미래에셋 금융 빅데이터

산업공학종합설계

보험청구분류

2020. 10. 12

4조

김성진 서재완 한상훈 허진용

대회 개요

미래에셋 그룹 주관

금융 빅데이터 페스티벌에 참여
기간 : 08월 28일 09:00 ~ 10월 04일 23:59

주제 :

보험료 청구 건 분류

결과 :

2팀으로 진행
(허진용, 한상훈, 12등, F1 : 85.60%)
(김성진, 서재완, 14등, F1 : 85%)

62팀 참여, 7위(약 F1 : 86.7%)까지 본선

대회의 목적

1. 보험금 청구 건에 대해서 자동, 심사, 조사 등의 3가지 케이스를 Classification하는 문제
2. 자동과 비자동 사이의 분류가 가장 중요한 포인트

보험금 청구 프로세스 (자료: 당사 홈페이지)

0. 데이터 설명

범주형 변수

0	ID
1	kcd등급
2	target
3	가입금액구간
4	건강인 우대상품 여부
5	고객나이구분
6	발생지역 구분
7	보험료 구간코드
8	보험사기이력
9	부담보5년경과여부
10	부실판매자계약여부
11	요양병원 여부
12	의료기관 등급
13	접수년월
14	중도부가계약여부
15	질병경중등급
16	질병구분
17	청구일계약일기간구분
18	청구일부활일기간구분
19	치료행위코드

부담보5년경과여부 :

이전에 앓았던 질환에 대해 보장 X

0. 데이터 설명

연속형 변수

0	병원_수술
1	병원_입원
2	병원_진단
3	병원_통원
4	수술 청구 건수
5	입원일수
6	입원청구건수
7	질병_수술
8	질병_입원
9	질병_진단
10	질병_통원
11	청구보험금
12	통원청구건수
13	통원횟수

0~4 병원별 평균 금액

7~10 질병별 평균 금액

목차

CHAPTER

1 EDA

2 데이터전처리

3 모델링

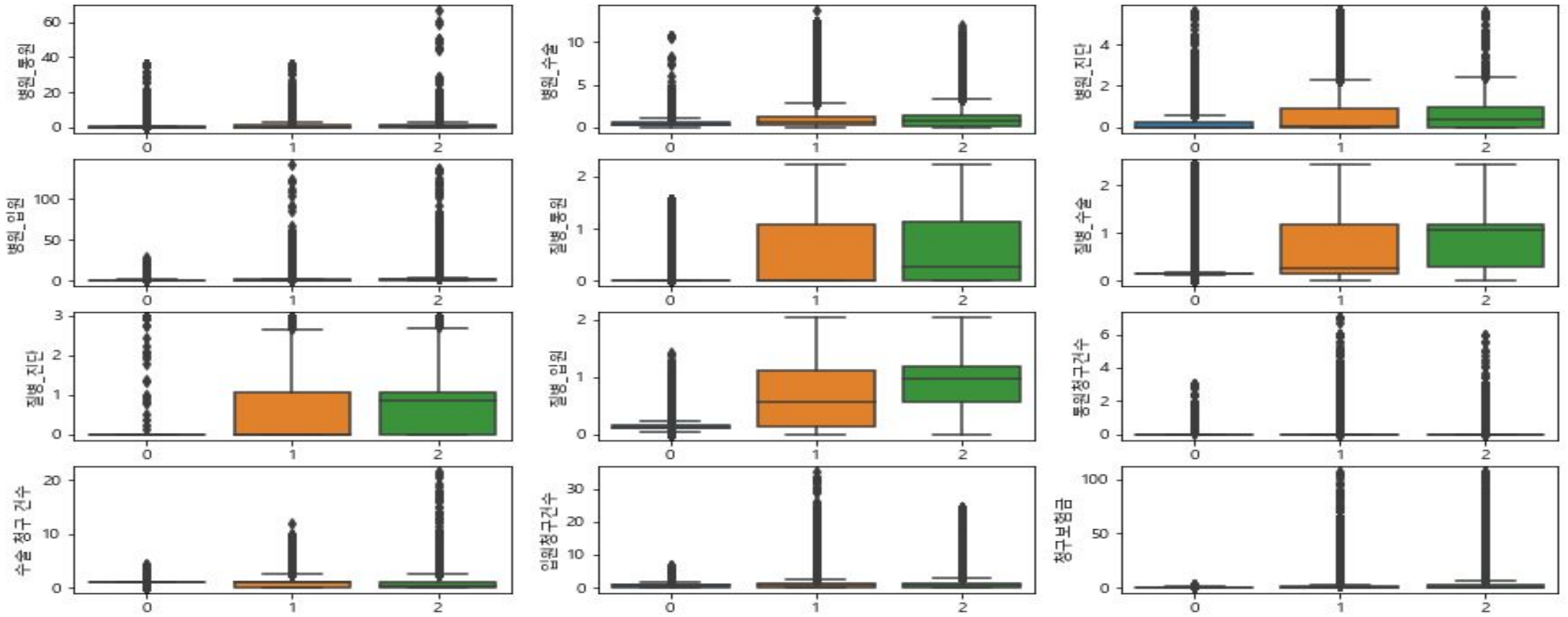
4 결론

CHAPTER

1 EDA

1. EDA

1) 연속형 데이터들의 데이터 boxplot통해 확인



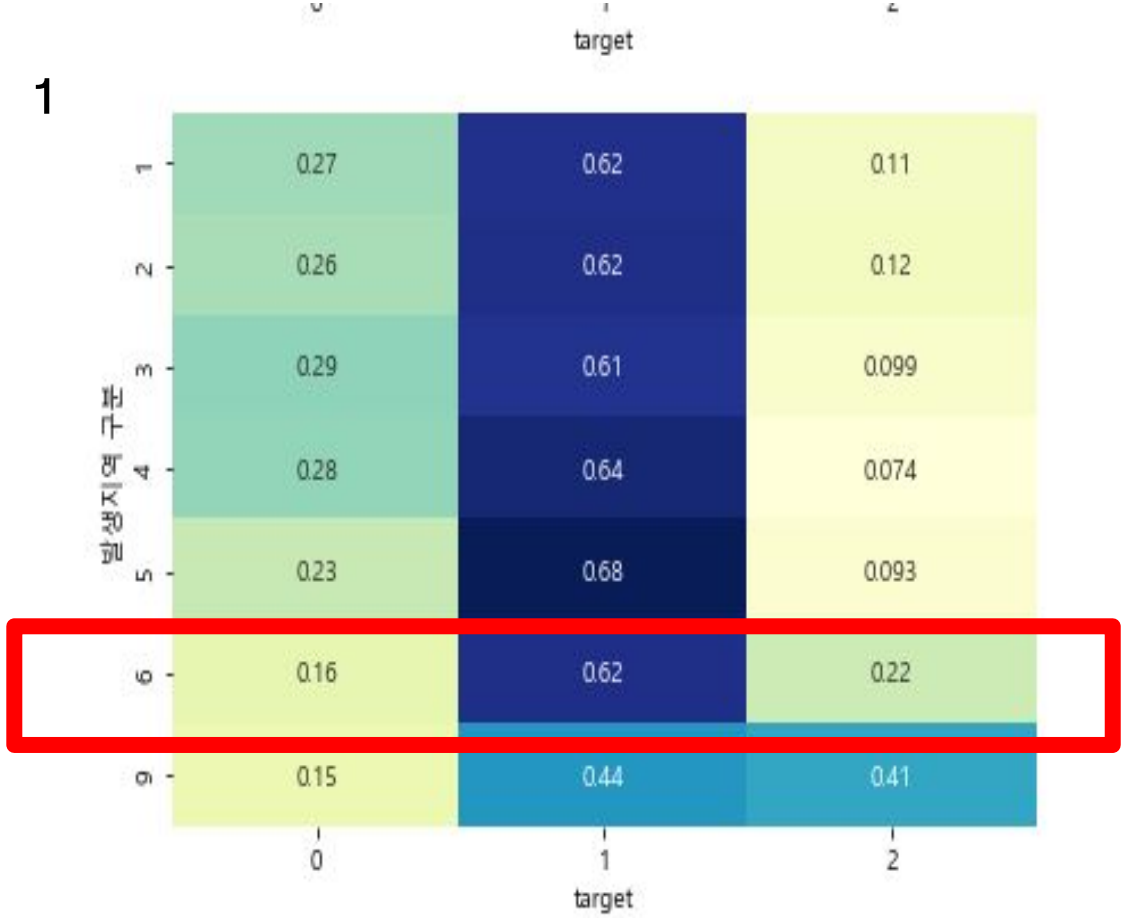
- 전체적으로 skewed 된 분포가 만들어 짐

- 이상치들이 굉장히 많기 때문에 처리 방안을 고민

1. EDA

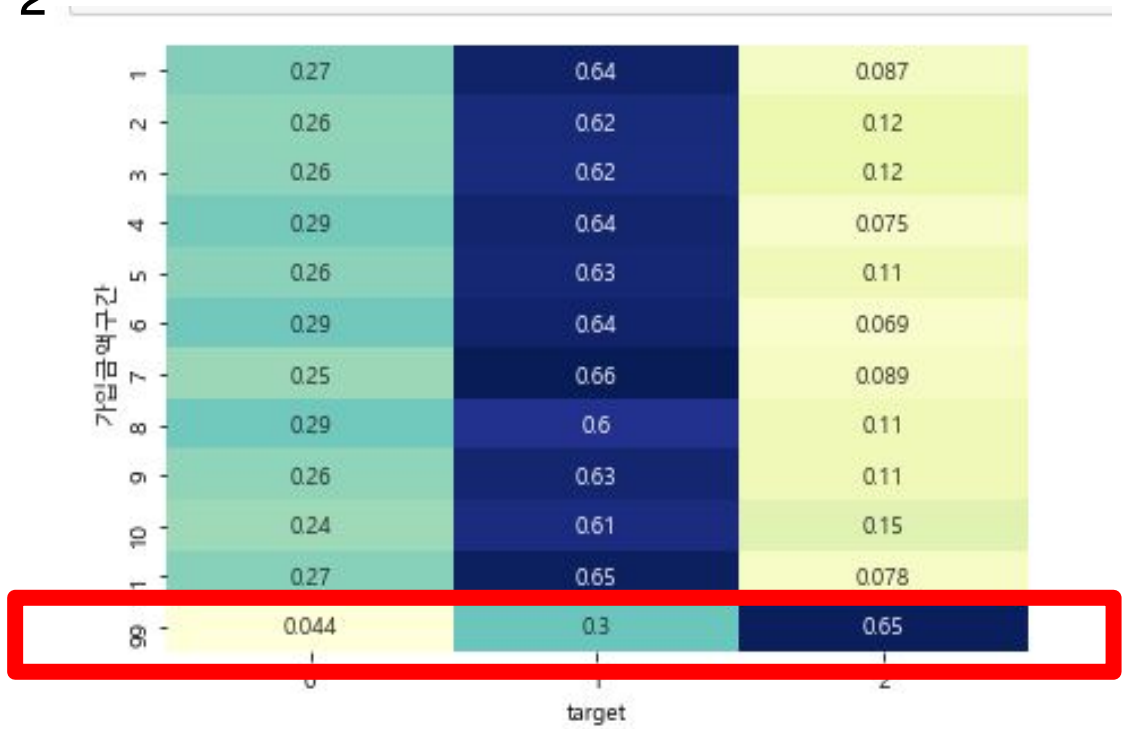
2) 범주형 데이터들의 데이터 crosstab을 통해확인

1



데이터 설명서상 존재하지 않는 6이 존재한다는 것을 확인

2



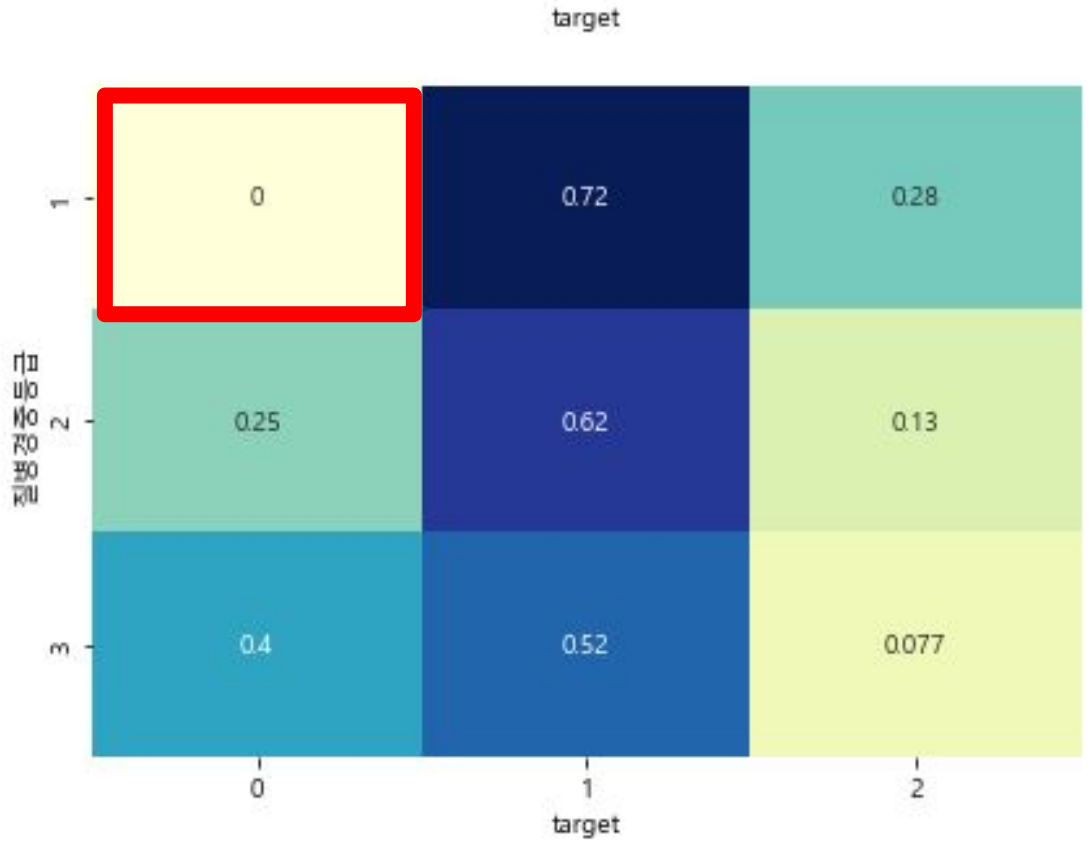
99의 경우 알 수 없는 구간이므로 도메인 지식을 이용해 해결

.

1. EDA

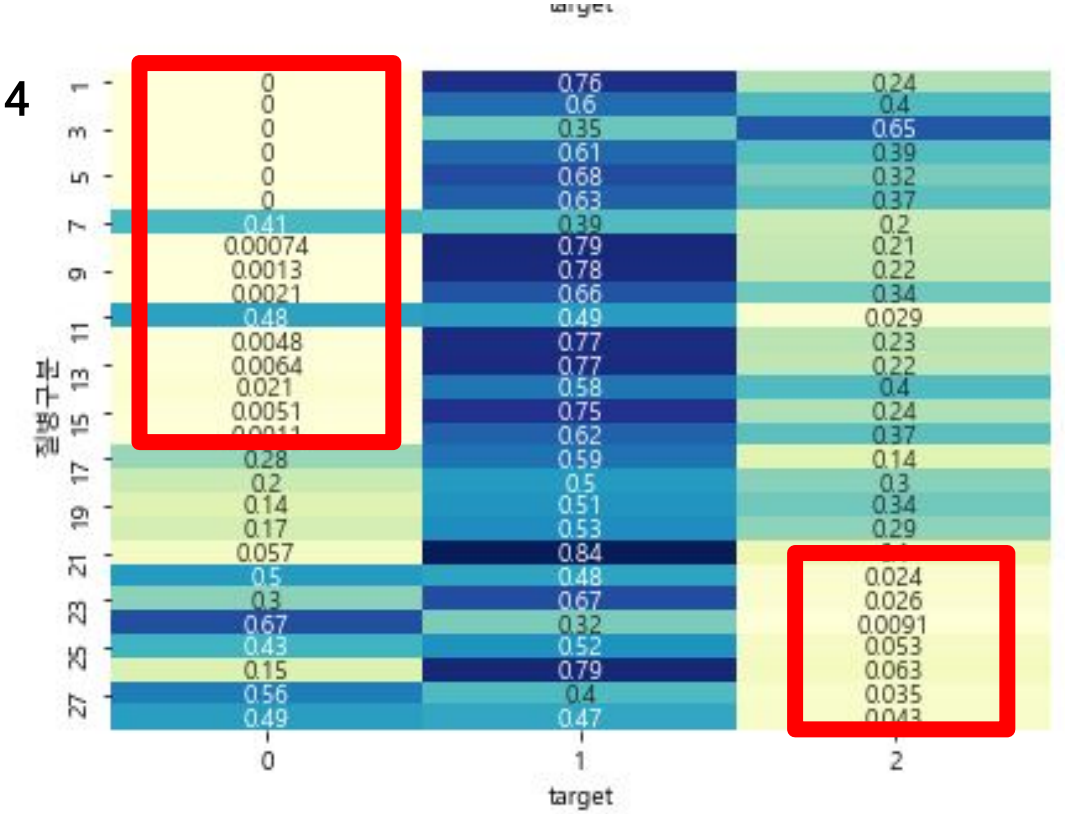
2) 범주형 데이터들의 데이터 crosstab을 통해확인

3



질병에서 1등급 중 자동인 경우가 없다는 것을 확인

4

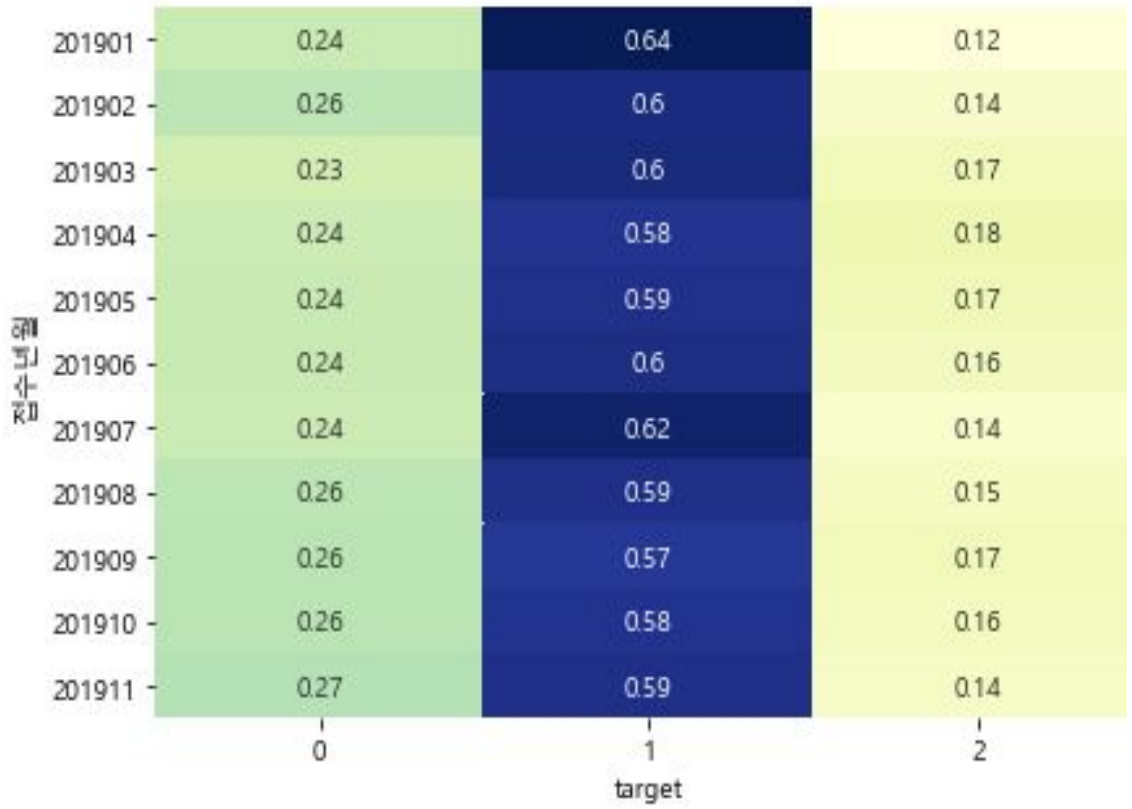


target이 0인경우나 2인경우와 같이 한쪽으로 쏠린 데이터 존재

1. EDA

2) 범주형 데이터들의 데이터 crosstab을 통해확인

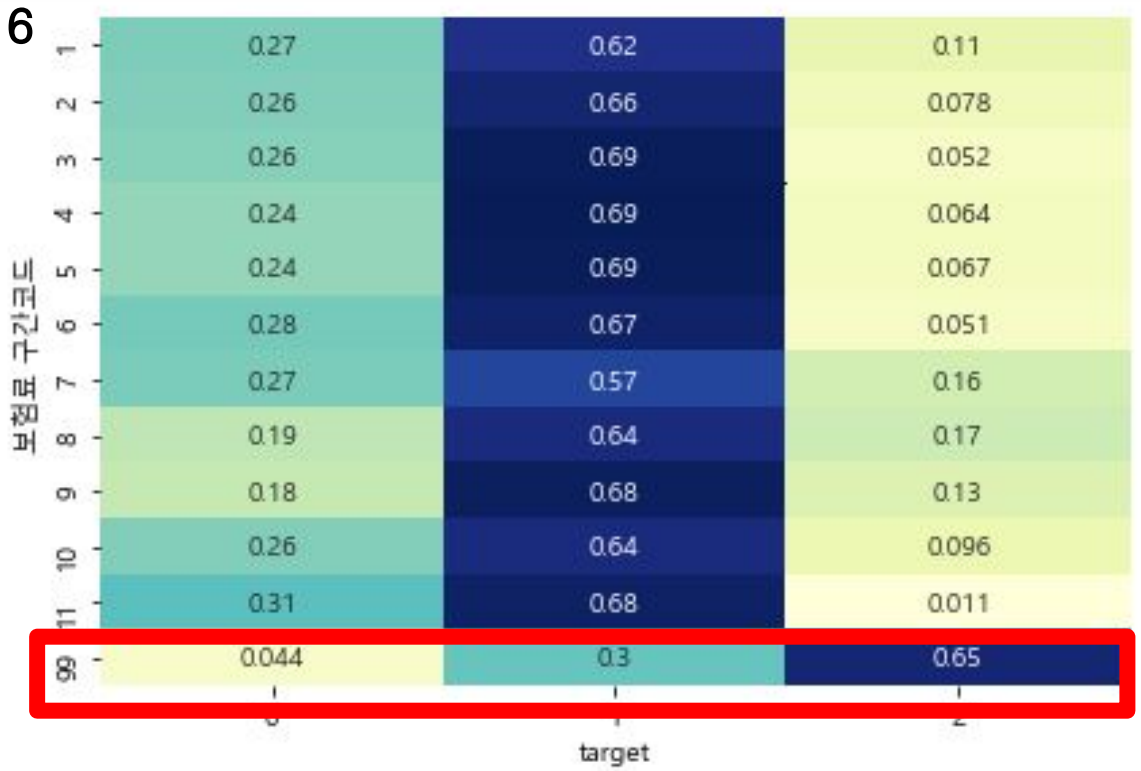
5



계절에 따른 변동성이 존재

금리와 관련이 있을것이라고 예측

6

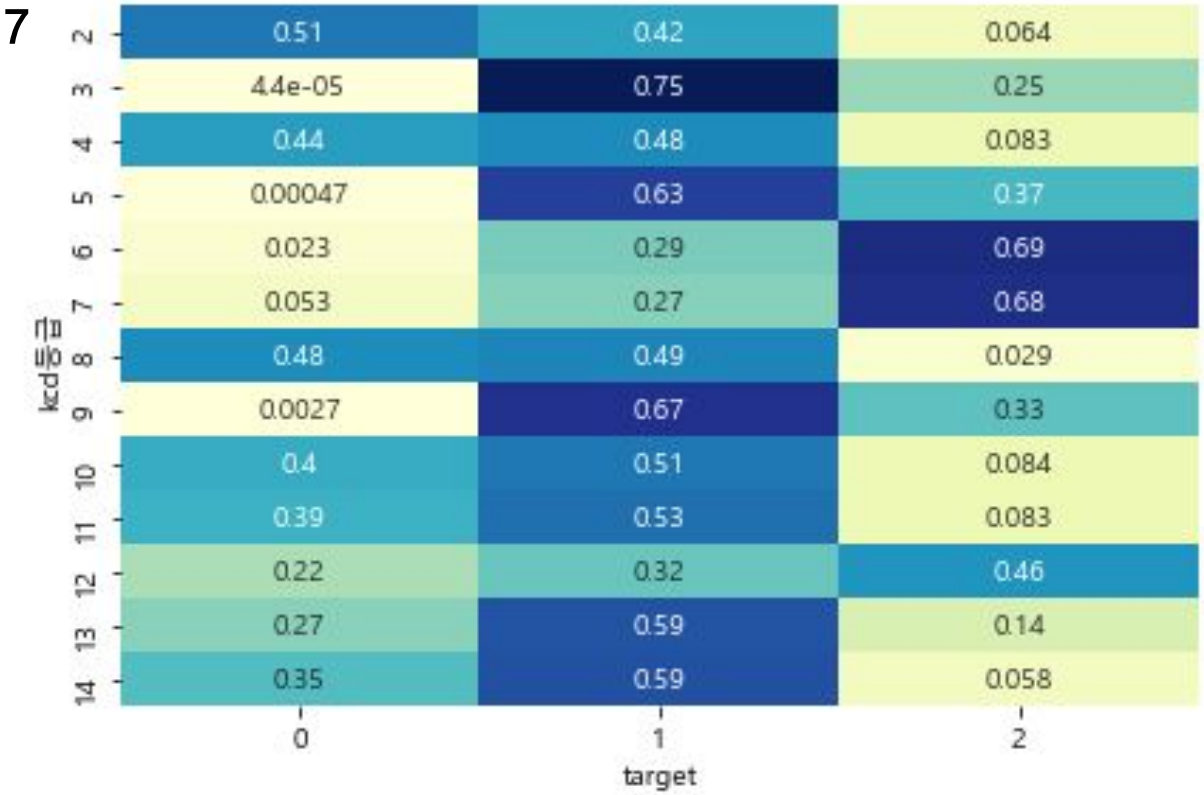


99라는 알 수 없는 구간이 존재

1. EDA

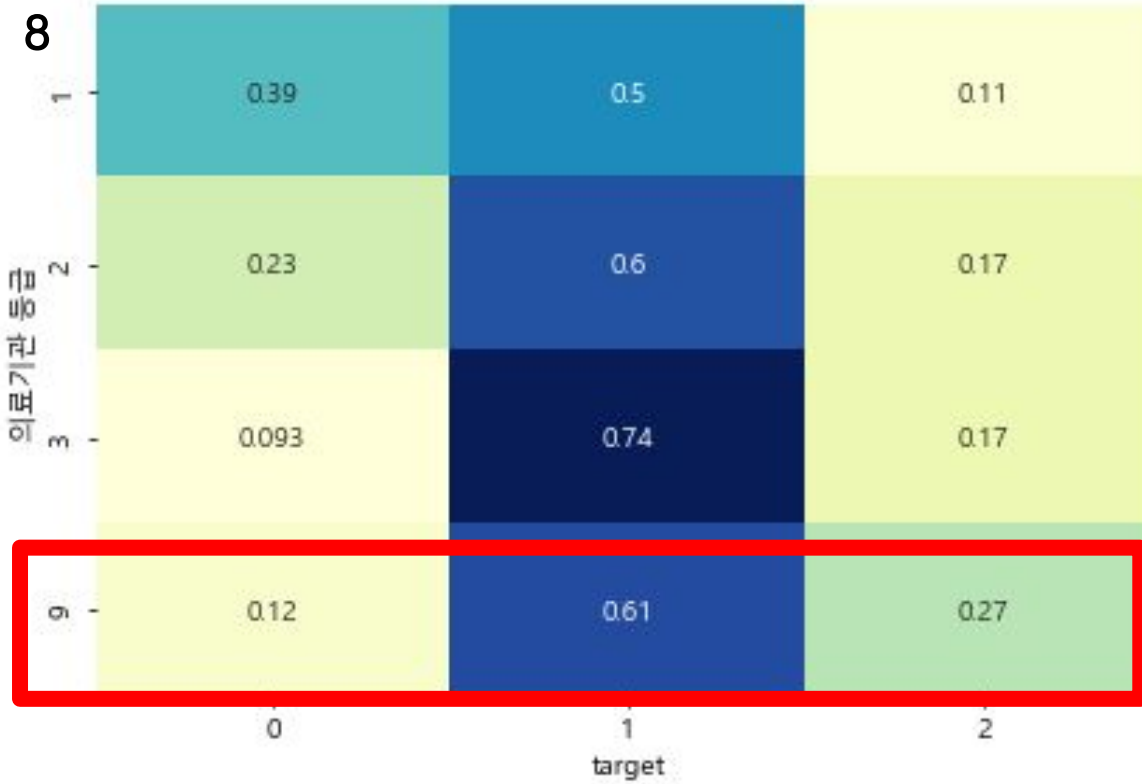
2) 범주형 데이터들의 데이터 crosstab을 통해확인

7



데이터 target별 치우침이 등급별로 다르다는 것을 확인

8

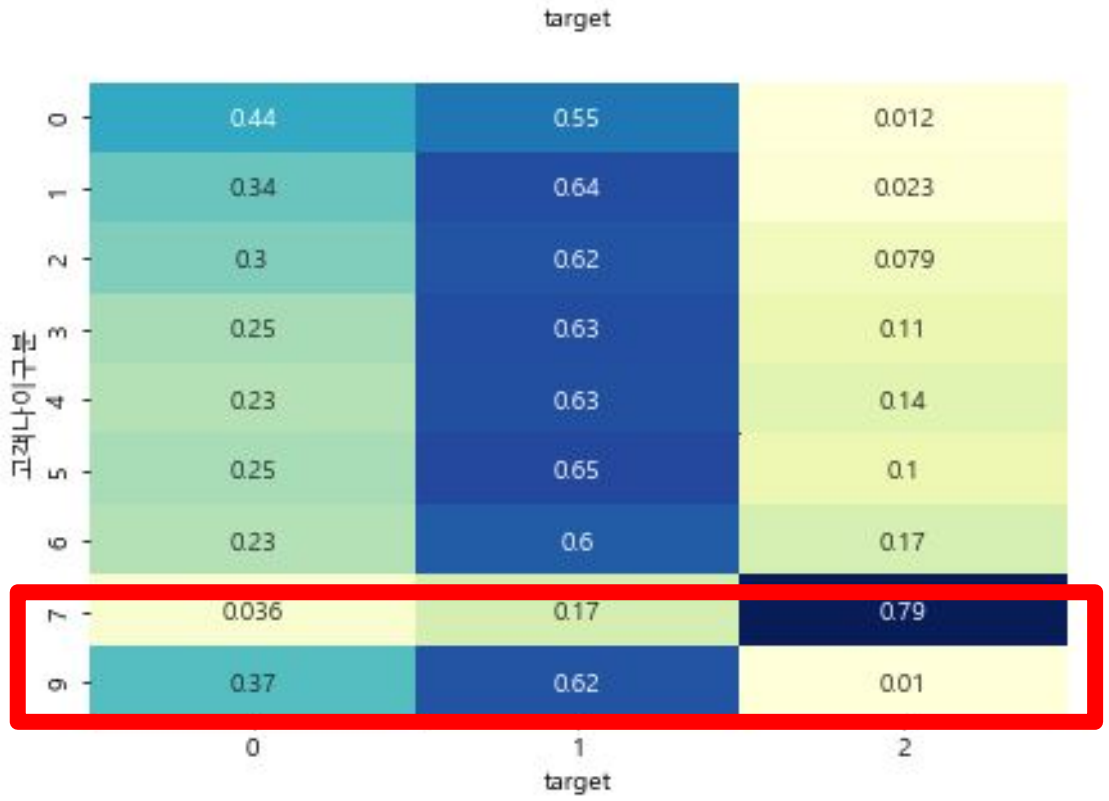


9라는 unknown 구간이 존재

1. EDA

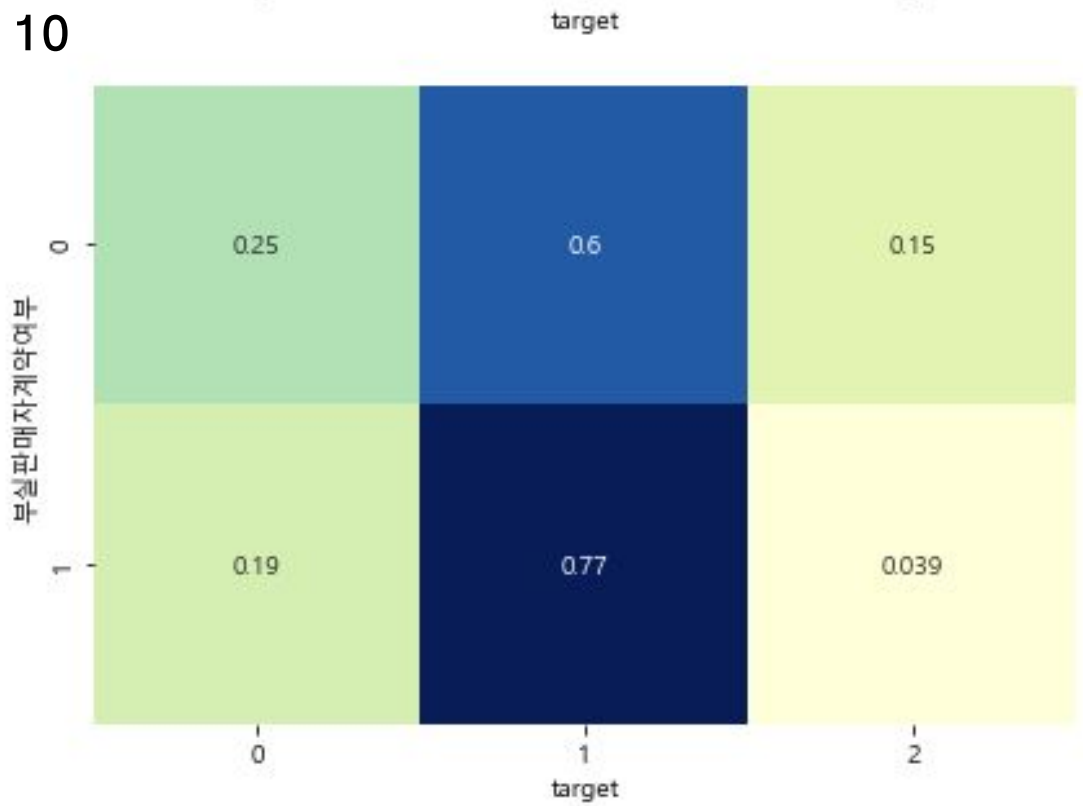
2) 범주형 데이터들의 데이터 crosstab을 통해확인

9



데이터 설명상에 없는 7, 9 값이 존재한다는 것을 확인

10

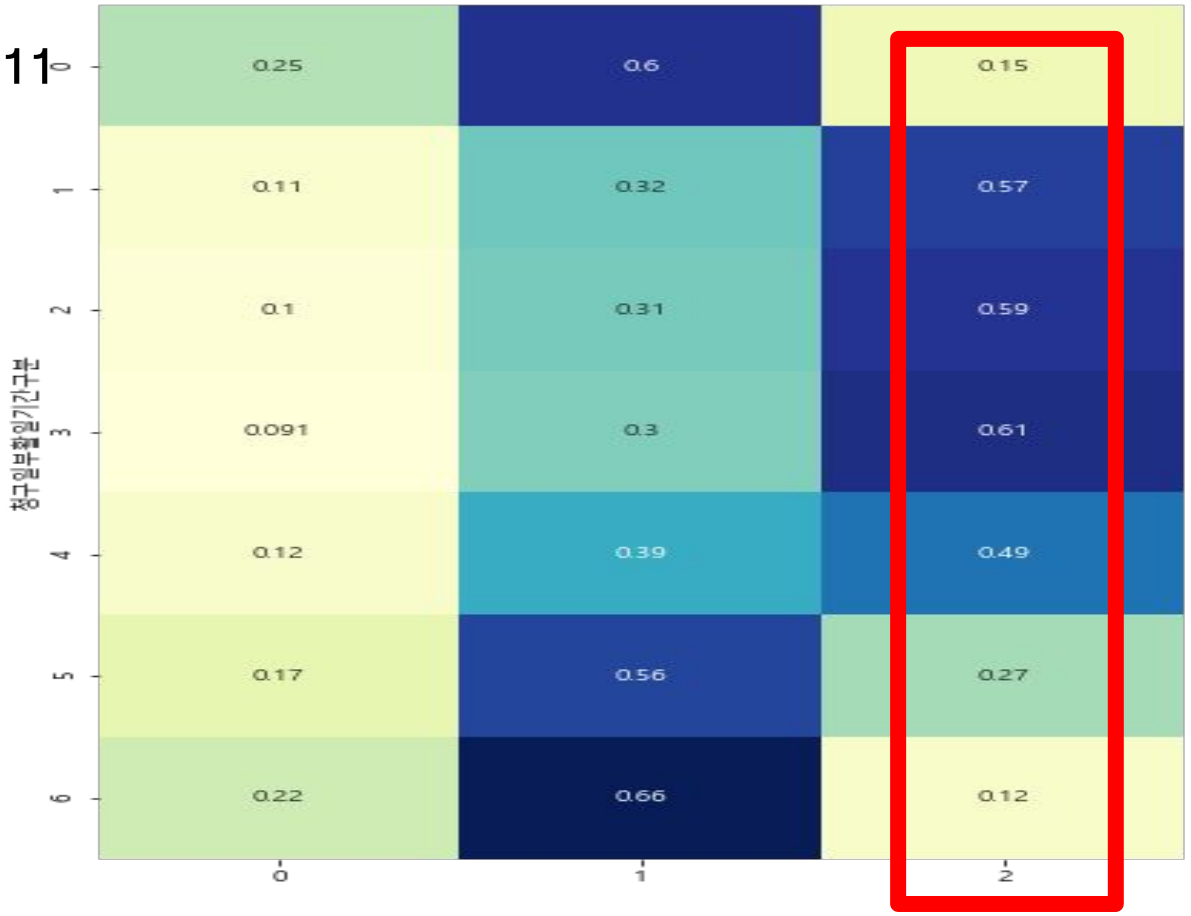


부실판매자계약여부에 따른 비율이 조금씩 다름을 확인

1. EDA

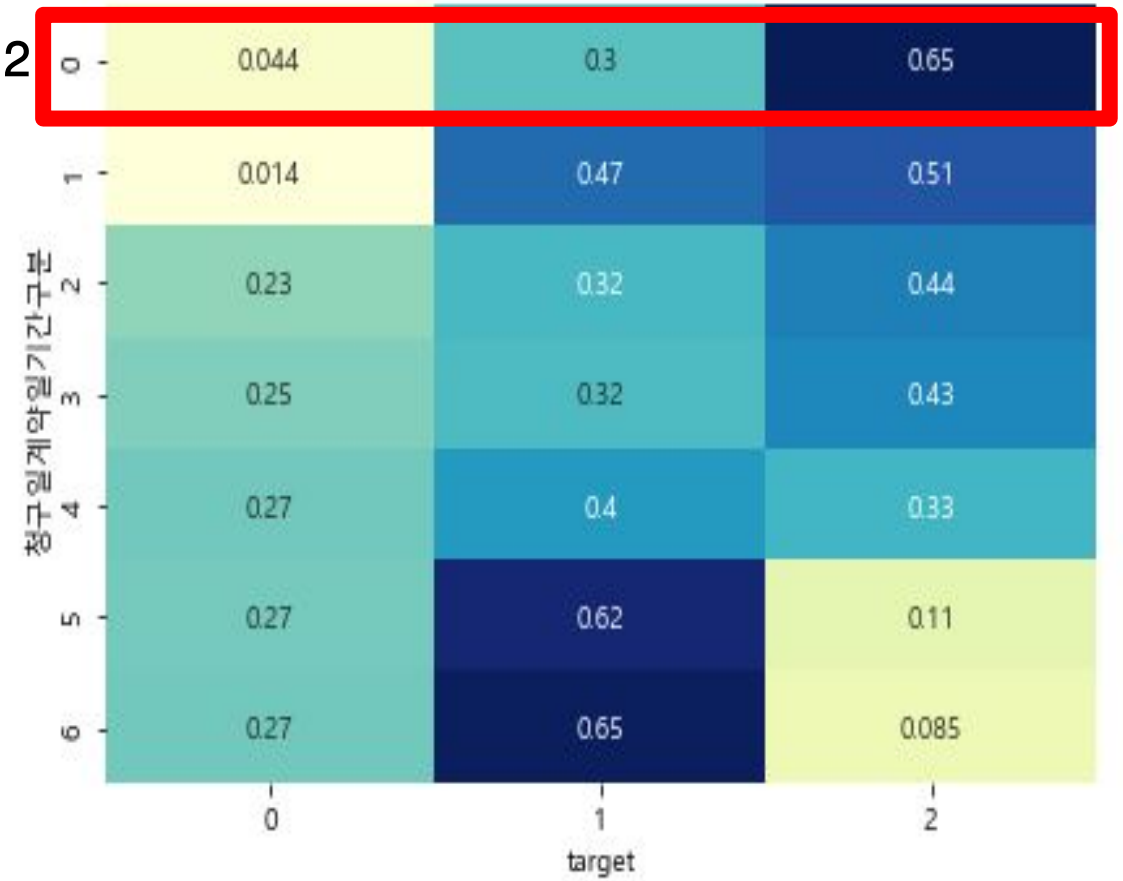
2) 범주형 데이터들의 데이터 crosstab을 통해확인

11



값이 작아질수록 target2에 많이 분포한다는 것을 확인

12



데이터 설명서상 unknwon인 0이 존재

2) 범주형 데이터들의 데이터 crosstab을 통해확인

```
In [19]: df_categorical[df_categorical['보험료 구간코드']==99]['가입금액구간'].unique()
Out[19]: array([99], dtype=int64)

In [21]: df_categorical[df_categorical['보험료 구간코드']==99]['청구일계약일기간구분'].value_counts()
Out[21]: 0    34436
         Name: 청구일계약일기간구분, dtype: int64

In [22]: df_categorical[df_categorical['청구일계약일기간구분']==0]['가입금액구간'].value_counts()
Out[22]: 99    34436
         4         7
         5         2
         3         2
         Name: 가입금액구간, dtype: int64
```

- 보험료 구간코드가 99인 경우에는 가입금액구간 모두 99
- 보험료 구간코드가 99인경우에는 청구일 계약일기간구분 모두 0
- 청구일 계약일기간구분이 0 이면 11개의 데이터를 제외하고는 가입금액구간 모두 99
- 즉 11개의 데이터를 제외하고 청구일계약일기간구분, 보험료구간코드, 가입금액구간의 unknown 인덱스가 같다

1. EDA

결론적으로

- 의료기관 코드가 9인값
- 고객 나이 구분 7와 9인값
- 보험료 구분 코드 99인 값
- 가입금액 구간 99인 값
- 청구일계약일기간구분 0인 값

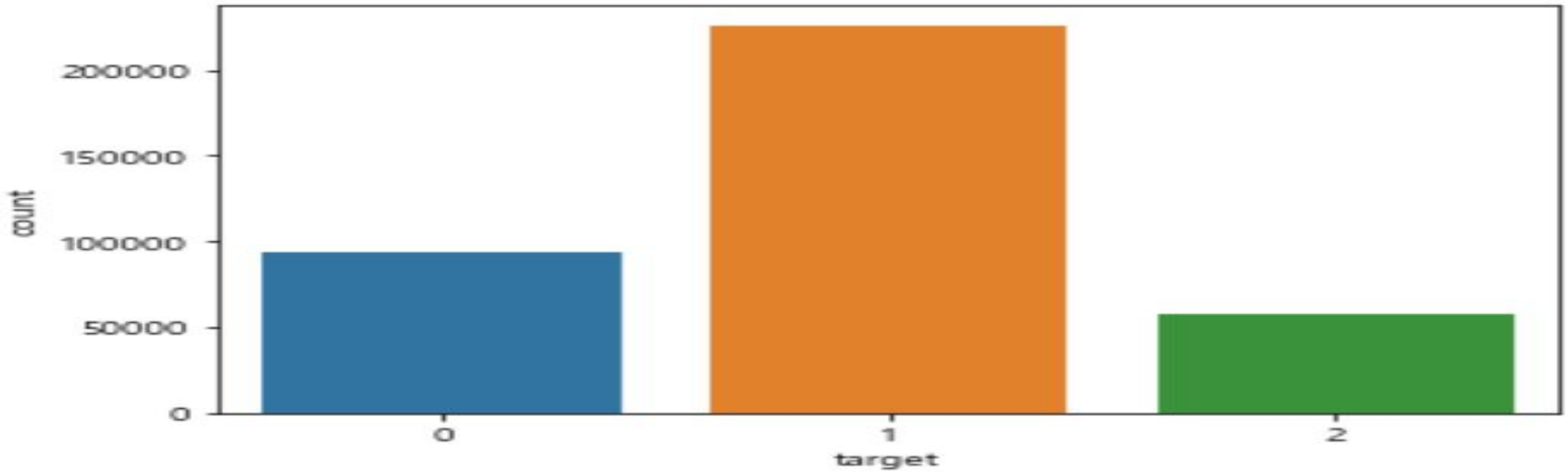
이 값들은 보험청구 분류에 중요한 데이터라고 판단되고 이 값들을 채우기 위하여 노력

CHAPTER

2 데이터 전처리

2. 데이터 전처리

1) 전체 데이터



총 377928개의 Train data
0 : 자동
1 : 심사
2 : 조사
데이터 불균형이 심각하게 존재하여 이를 업샘플링을 시도하여 보았지만 모델의 성능은 변화하지 않았음.

2. 데이터 전처리

2) 의료기관구분 9 처리

7 의료기관구분코드

코드	의료기관구분
1	1차병원(일반병원)
2	2차병원(종합병원)
3	3차병원(상급종합병원)
9	Unknown

	병원_통원	병원_수술	병원_진단	병원_입원	의료기관 등급
0	0.1303	0.0000	0.0000	1.6021	2
1	0.0000	0.5770	0.0000	0.0000	1
2	0.0000	0.2885	0.0000	0.0000	1
3	0.0197	0.0680	0.0587	0.5638	2

병원과 관련된 결측치가 183개 정도 존재하였음.
처리하기 위해 classification 문제로 재정의
RandomForest 모형을 사용한 결과 validation set에서 82% 정도의 정확도
라는 점을 고려하여 183개의 데이터는 **삭제하기로 하였음**.

2. 데이터 전처리

3) 고객나이구분 7, 9 처리

5 연령구분코드

코드	연령
0	0~9세
1	10~19세
2	20~29세
3	30~39세
4	40~49세
5	50~59세
6	60세 이상

```
1 sorted(df['고객나이구분'].unique())
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 9]
```

```
1 len(df[df['고객나이구분']==7])
```

```
26577
```

```
1 len(df[df['고객나이구분']==9])
```

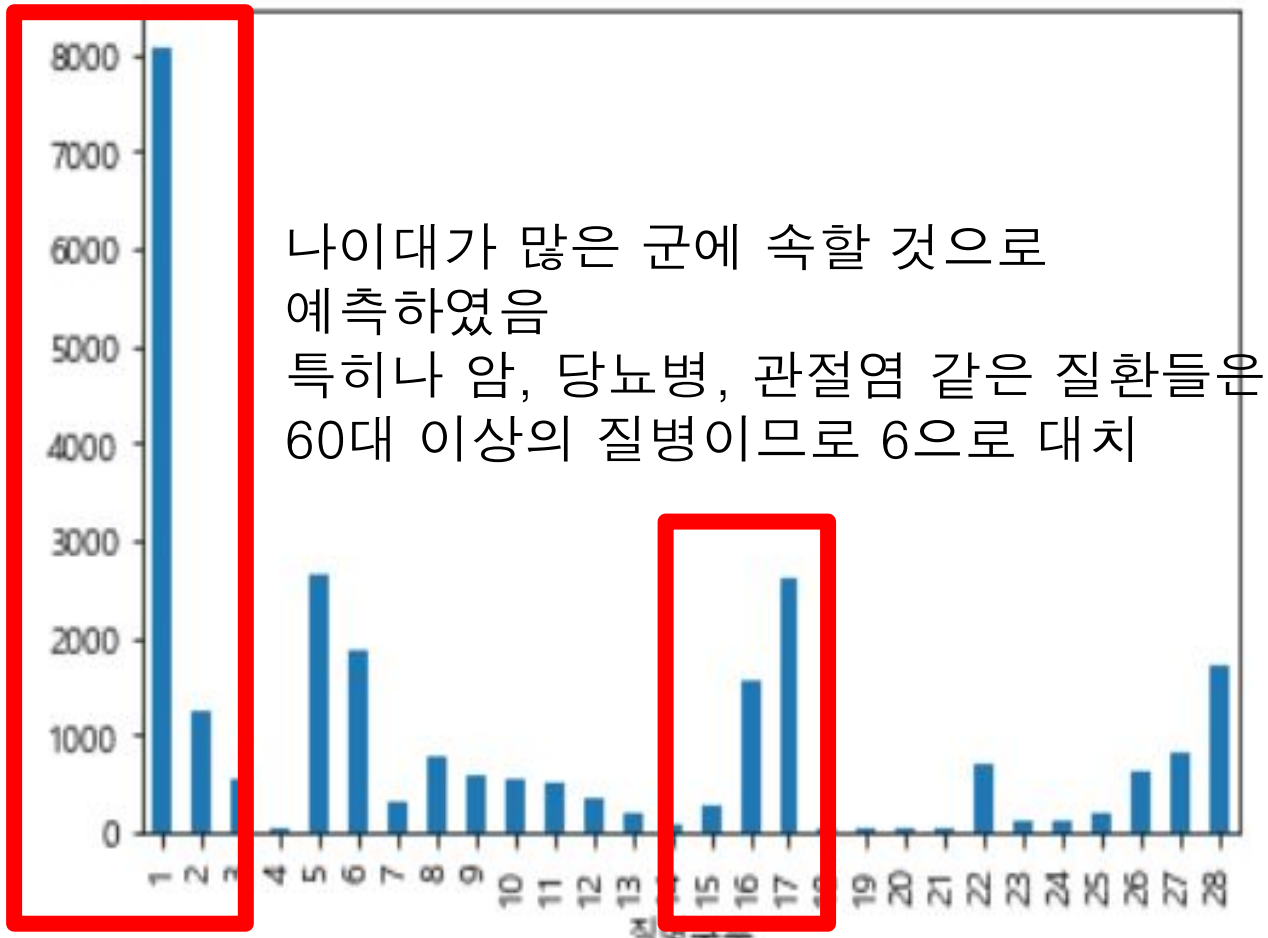
```
12459
```

나이 7과 9가 데이터에 존재하였으며
TEST 데이터에도 존재하였기에
이를 처리하기 해야 하였음.

2. 데이터 전처리

3 – 1) 고객나이구분 7 처리

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



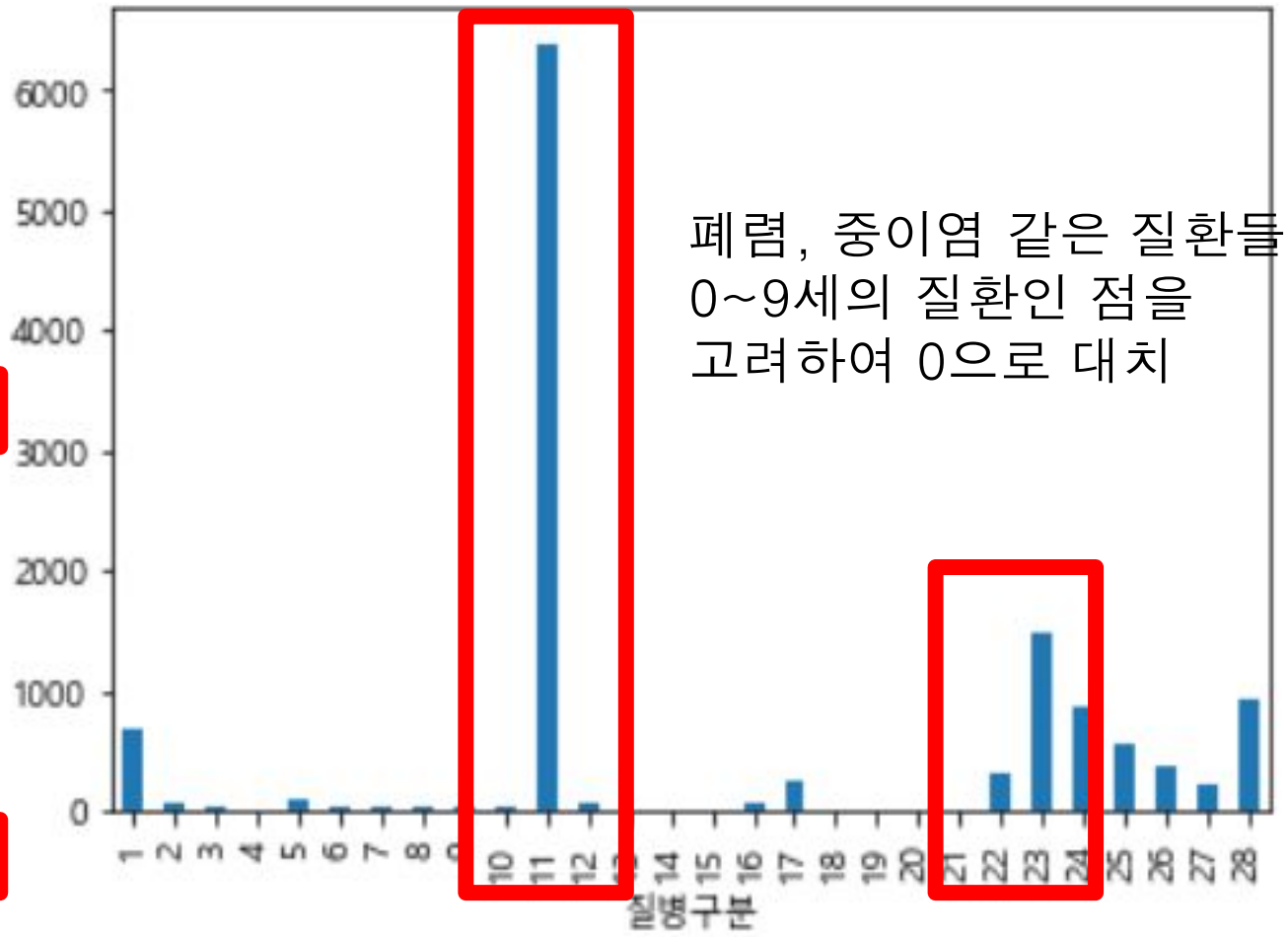
나이대가 많은 군에 속할 것으로
예측하였음
특히나 암, 당뇨병, 관절염 같은 질환들은
60대 이상의 질병이므로 6으로 대치

참고자료 : http://health.cdc.go.kr/health/mobileweb/content/group_view.jsp?CID=Y3UH9E64HN

2. 데이터 전처리

3 – 2) 고객나이구분 9 처리

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



폐렴, 중이염 같은 질환들은
0~9세의 질환인 점을
고려하여 0으로 대치

참고자료 : https://m.health.chosun.com/svc/news_view.html?contid=2020021402276

2. 데이터 전처리

4) 가입금액구간 99인 경우 전처리

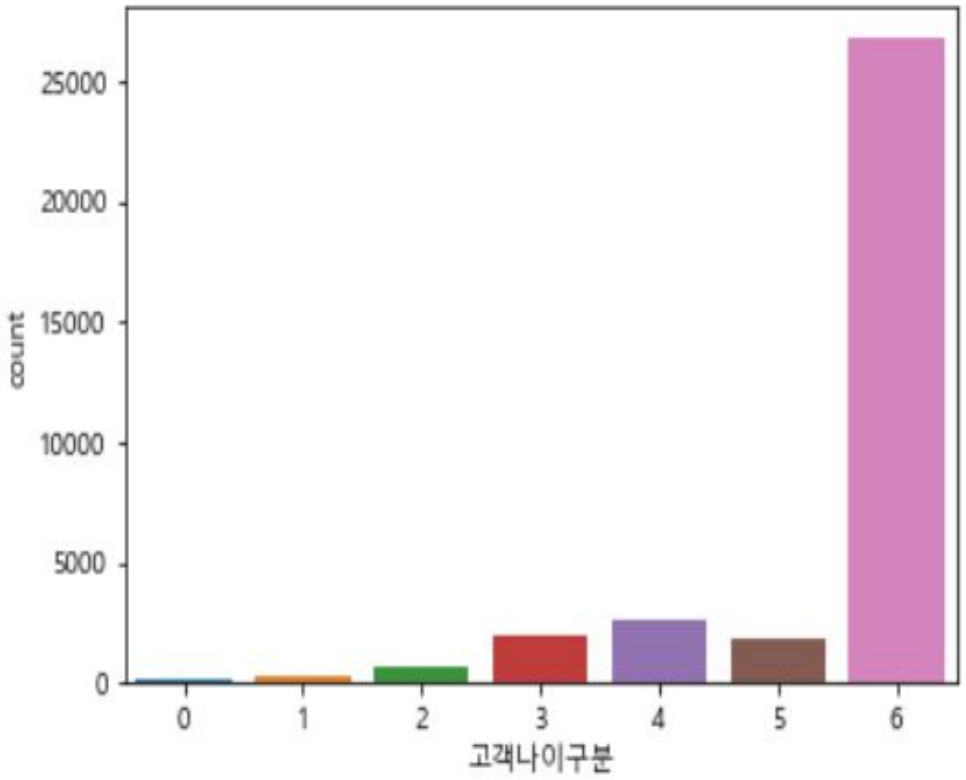
코드	가입금액구간
1	0~1000만원 미만
2	1000만~2000만원 미만
3	2000만~3000만원 미만
4	3000만~4000만원 미만
5	4000만~5000만원 미만
6	5000만~6000만원 미만
7	6000만~7000만원 미만
8	7000만~8000만원 미만
9	8000만~9000만원 미만
10	9000만~1억원 미만
11	1억원 이상
99	Unknown

99인 사람의 청구보험금 평균 : 0.701033122
99가 아닌 사람의 청구보험금 평균 : 1.8573

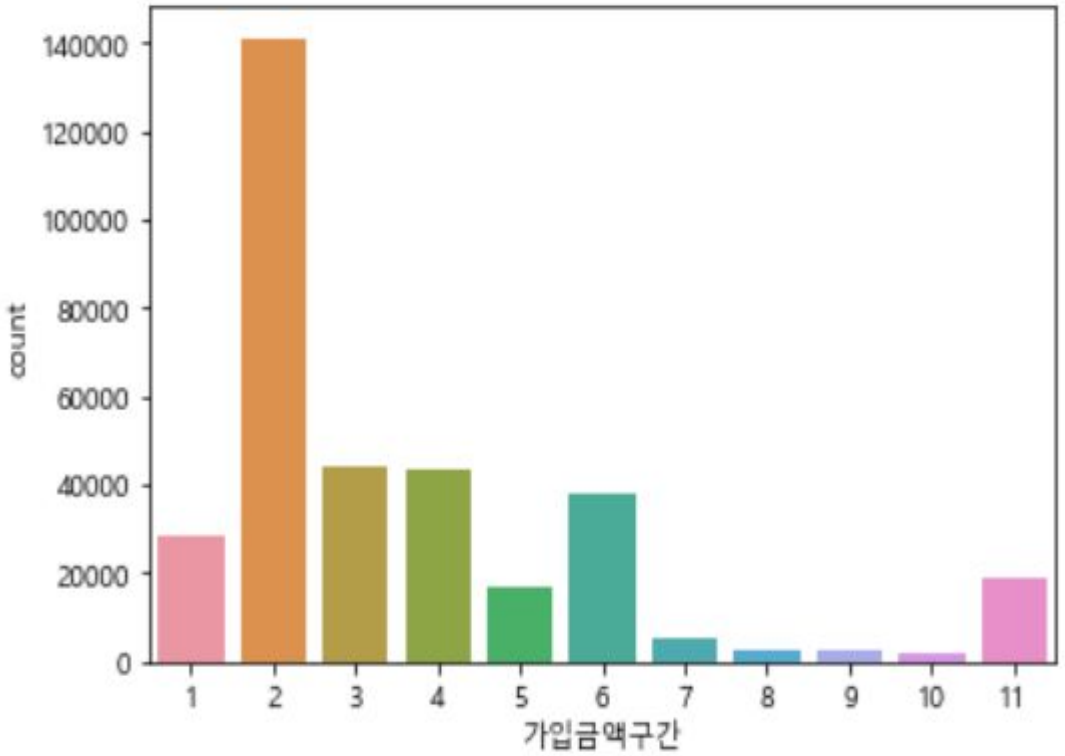
평균보다 현저하게 낮다는 것을 볼 때 **낮은 보험금 구간이라는 것을 추론**

2. 데이터 전처리

4) 가입금액구간 99 전처리



가입금액구간이 99 제외한 경우



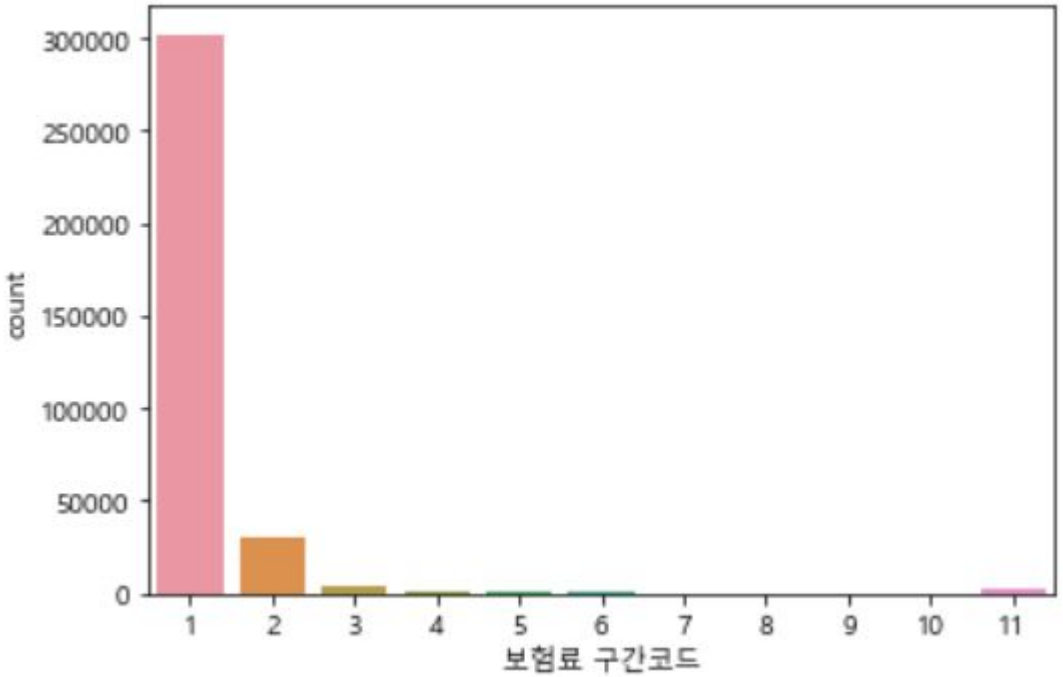
1) 나이가 크다는 점, 2) 평균이 작다는 점, 3) 99 제외한 경우 가입금액구간 2번에 많이 쏠려있다는 점을 고려할 때

2로 대치시키는 것이 합리적이라는 판단

2. 데이터 전처리

5) 보험료구간 99 전처리

코드	보험료구간
1	0~10만원 미만
2	10만~20만원 미만
3	20만~30만원 미만
4	30만~40만원 미만
5	40만~50만원 미만
6	50만~60만원 미만
7	60만~70만원 미만
8	70만~80만원 미만
9	80만~90만원 미만
10	90만~100만원 미만
11	100만원 이상
99	Unknown



일반적인 상황을 가정하고 **1으로 대치** 하는 것이 합리적이라는 판단

2. 데이터 전처리

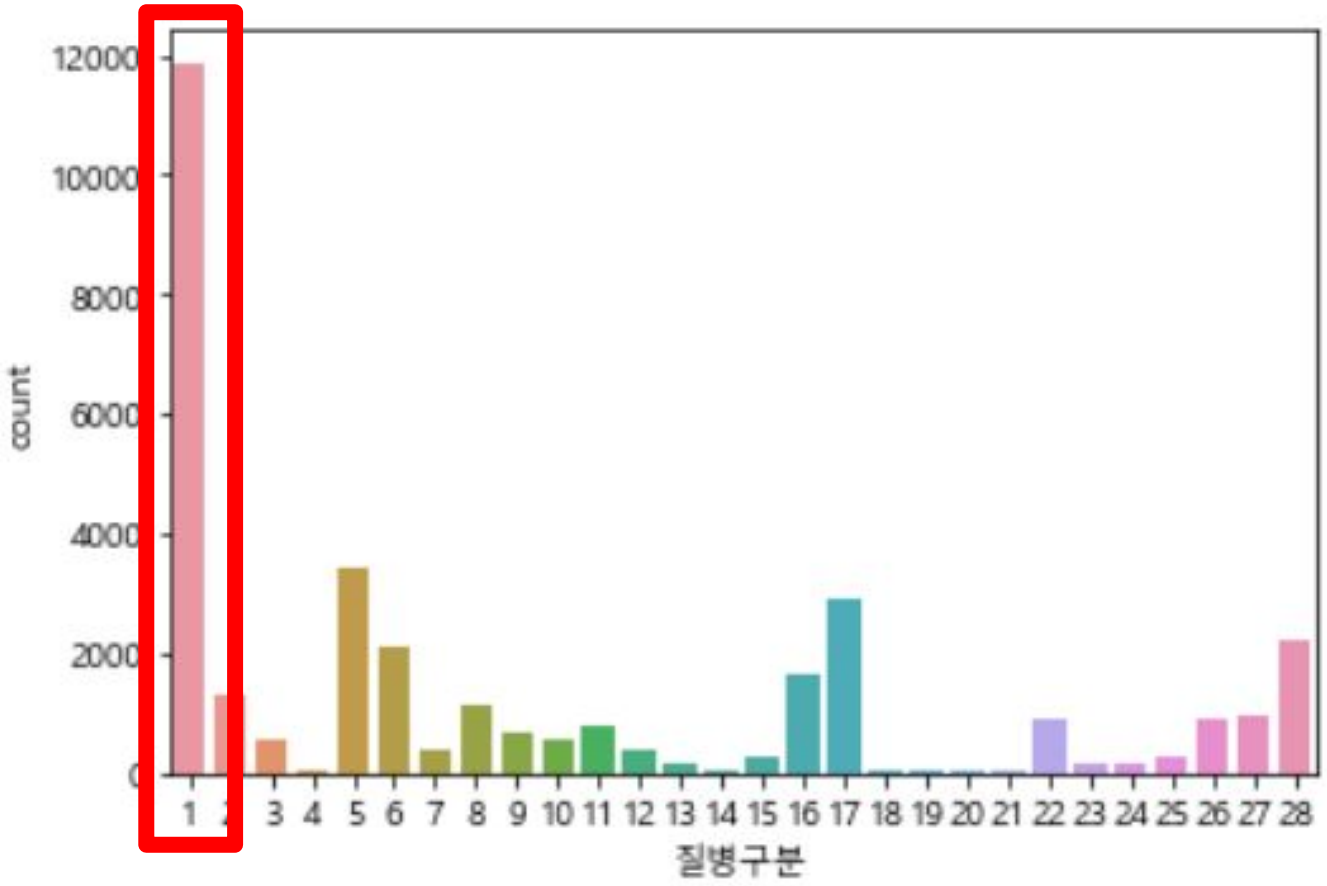
6) 청구계약일 구분 0 처리

코드	기간	
0	청구일계약일간기간구분코드	Unknown
1	1일 이상 3개월 이하	
2	3개월 초과 6개월 이하	
3	6개월 초과 1년 이하	
4	1년 초과 2년 이하	
5	2년 초과 5년 이하	
6	5년 초과	

2. 데이터 전처리

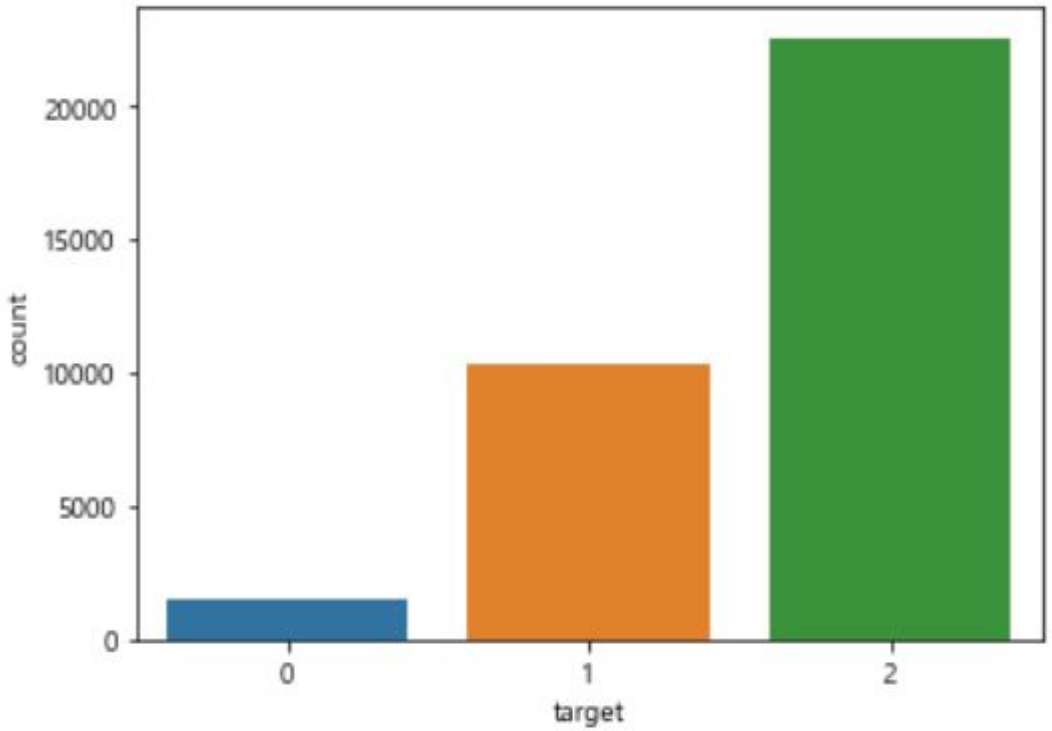
6) 청구계약일 구분 0 처리

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

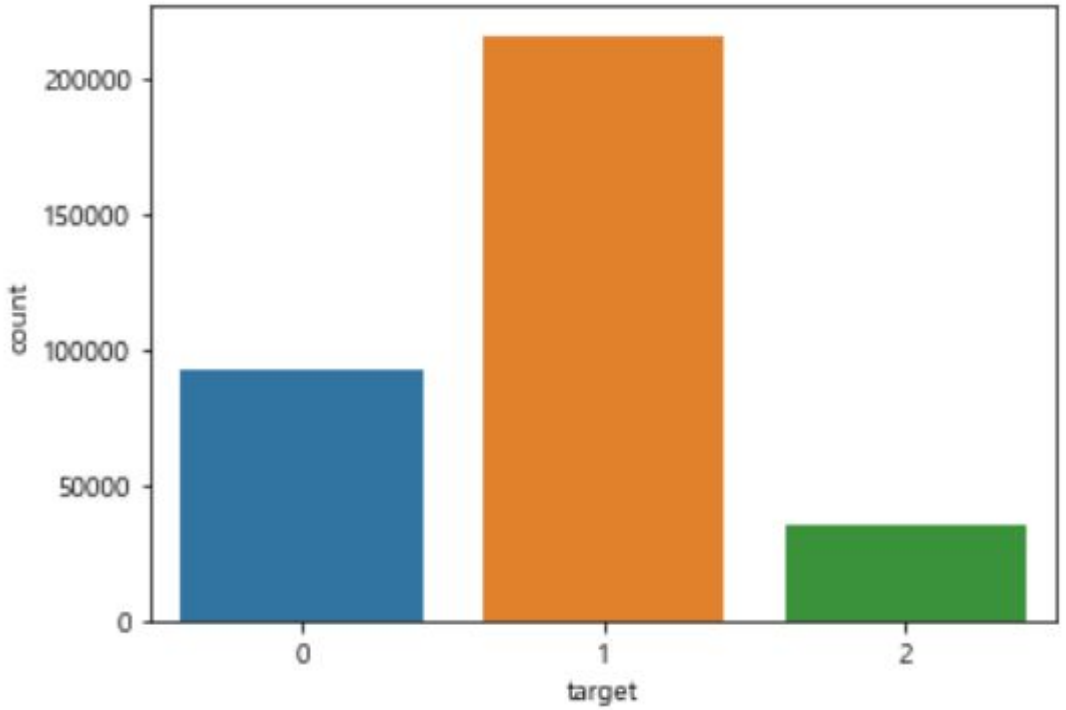


2. 데이터 전처리

6) 청구계약일 구분 0 처리



0인 경우



0이 아닌 경우

0의 경우 비율상 타겟이 자동인 경우가 굉장히 적었고 심사가 굉장히 많았음, 0이 아닌 경우에는 조사가 가장 많았음

보험사의 경우 **2년이라는 기간이 가장 중요한 구간**이라는 것을 기사를 통해 알게 되었음.

참고자료 : http://www.dybionews.com/news/articleView.html?idxno=1959

2. 데이터 전처리

6) 청구계약일 구분 0 처리

1	1일 이상 3개월 이하
2	3개월 초과 6개월 이하
3	6개월 초과 1년 이하
4	1년 초과 2년 이하

보험사는 **2년**을 굉장히 중요시 한다는 점을 고려하였고, 암과 같은 질병이 있다면 애초에 가입이 안되었을 것이므로 **3,4로 대치시킨 후 모델링 해본 결과 4가 조금 더 나은 결과**를 보여줌

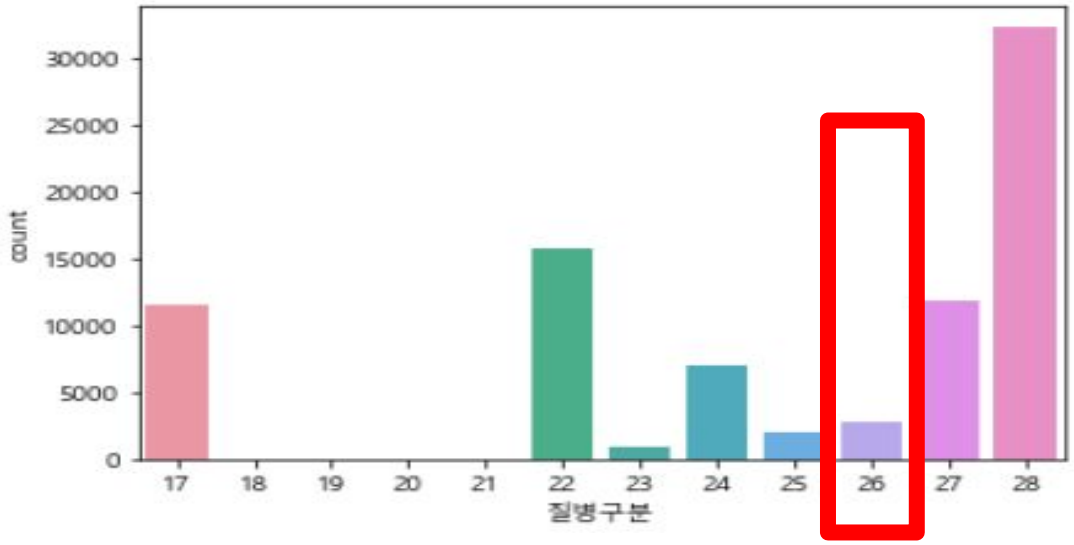
2. 데이터 전처리

7) 의미 있던 **파생 변수**

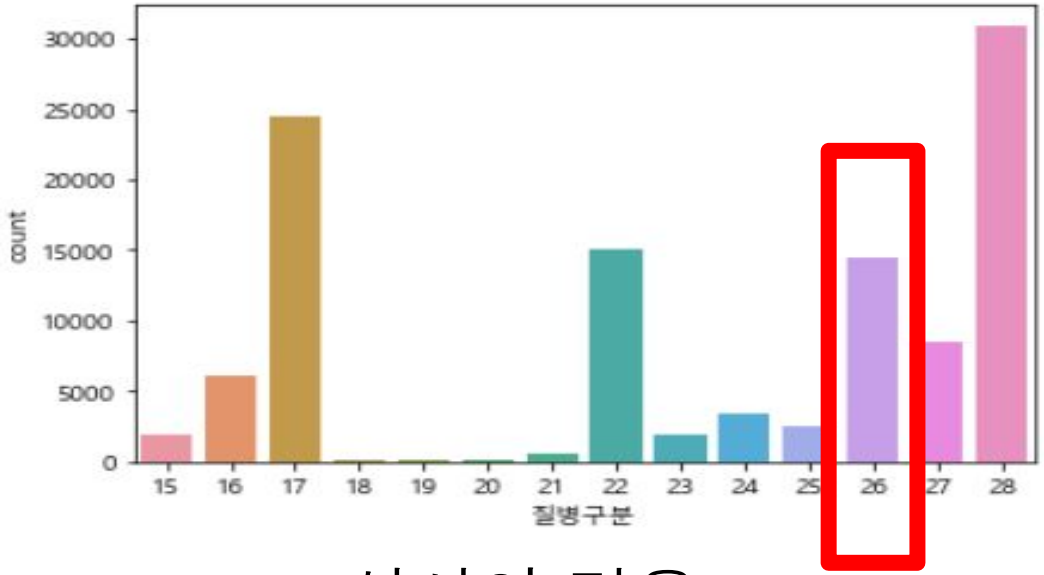
1. 거시적인 변수 추가 : **금리**

(2019.1~6 : 1.75%), (2019.7~9 : 1.70%), (2019.10~12 , 1.25%), 출처 : 한국은행

2. EDA를 통한 **비뇨기과**의 **현저한 차이**를 반영



자동인 경우



심사인 경우

8) 시도하였지만 성능 높이기에 실패한 파생변수들

1) **질병 세분화** –> 질병 평균 가격 변수(4개 feature)를 가지고 질병을 clustering **하는 방식**으로 (기존 질병 28개) 군집수를 늘려가며 **변수 생성**

2) **자동 / 심사, 조사** 두가지를 구분 지어주는 변수

3) **남/녀**를 구분해주는 변수 (비뇨기과, 산부인과를 기반)

CHAPTER

3 모델링

3.모델링

아무런 전처리를 하지 않고 **Auto ML을 활용**하여 **가장 성능이 좋은 모델** 선택

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	Extra Trees Classifier	0.9498	0.0000	0.9431	0.9499	0.9498	0.9097	0.9097	70.9211
1	Random Forest Classifier	0.9239	0.0000	0.9160	0.9257	0.9242	0.8637	0.8641	5.8420
2	K Neighbors Classifier	0.8938	0.0000	0.8986	0.8965	0.8946	0.8125	0.8133	12.2247
3	Decision Tree Classifier	0.8690	0.0000	0.8615	0.8695	0.8692	0.7660	0.7660	9.7601
4	CatBoost Classifier	0.8222	0.0000	0.7990	0.8263	0.8228	0.6808	0.6817	178.1256
5	Light Gradient Boosting Machine	0.7915	0.0000	0.7576	0.7965	0.7915	0.6244	0.6258	10.4570
6	Gradient Boosting Classifier	0.7536	0.0000	0.6835	0.7572	0.7490	0.5398	0.5433	416.0847
7	Extreme Gradient Boosting	0.7452	0.0000	0.6666	0.7493	0.7391	0.5196	0.5243	151.7314
8	Logistic Regression	0.7211	0.0000	0.6480	0.7220	0.7162	0.4758	0.4792	23.2429
9	Linear Discriminant Analysis	0.7060	0.0000	0.6335	0.7069	0.7013	0.4471	0.4504	8.7381
10	Ridge Classifier	0.7055	0.0000	0.6040	0.7060	0.6935	0.4239	0.4348	0.7543
11	SVM - Linear Kernel	0.7021	0.0000	0.5969	0.7098	0.6799	0.4069	0.4332	19.5573
12	Ada Boost Classifier	0.6873	0.0000	0.6498	0.6891	0.6875	0.4415	0.4420	30.5706
13	Quadratic Discriminant Analysis	0.5904	0.0000	0.6123	0.7031	0.5862	0.3537	0.4063	5.0679
14	Naive Bayes	0.5150	0.0000	0.6648	0.6967	0.4762	0.3215	0.3989	0.4456

앙상블 모델이 가장 성능이 좋게 나왔음.
아무런 전처리를 하지 않고,
RF : F1 score 81%, ETC : F1 score : 82.5%

3.모델링

1) **전략1 – 질병 경중**에 따라 **모델링**

1.1 중증
1.2 성인
2.1 생활

총 **3단계**가 존재했고 **각각을 모델링** 하는 전략

3.모델링

1) 질병 경중에 따라 모델링 – 중증

1	중증	1	암
		2	상피내암
		3	경계성
		4	심장질환
		5	
		6	뇌혈관질환
2	성인	7	간질환
		8	
		9	신장질환
		10	갑상선질환
		11	폐렴
		12	천식
		13	위궤양
		14	십이지장궤양

Extra Tree Classifier 기준
depth : 500
Accuracy : 99%
F1 Score : 99%

3.모델링

2) 질병 경중에 따라 모델링 – 성인

		15	고혈압
		16	당뇨병
		17	관절염
		18	
		19	
		20	
3	생활	21	골다공증
		22	백내장
		23	중이염
		24	충수염
		25	남성비뇨기계
		26	
		27	부인과
		28	

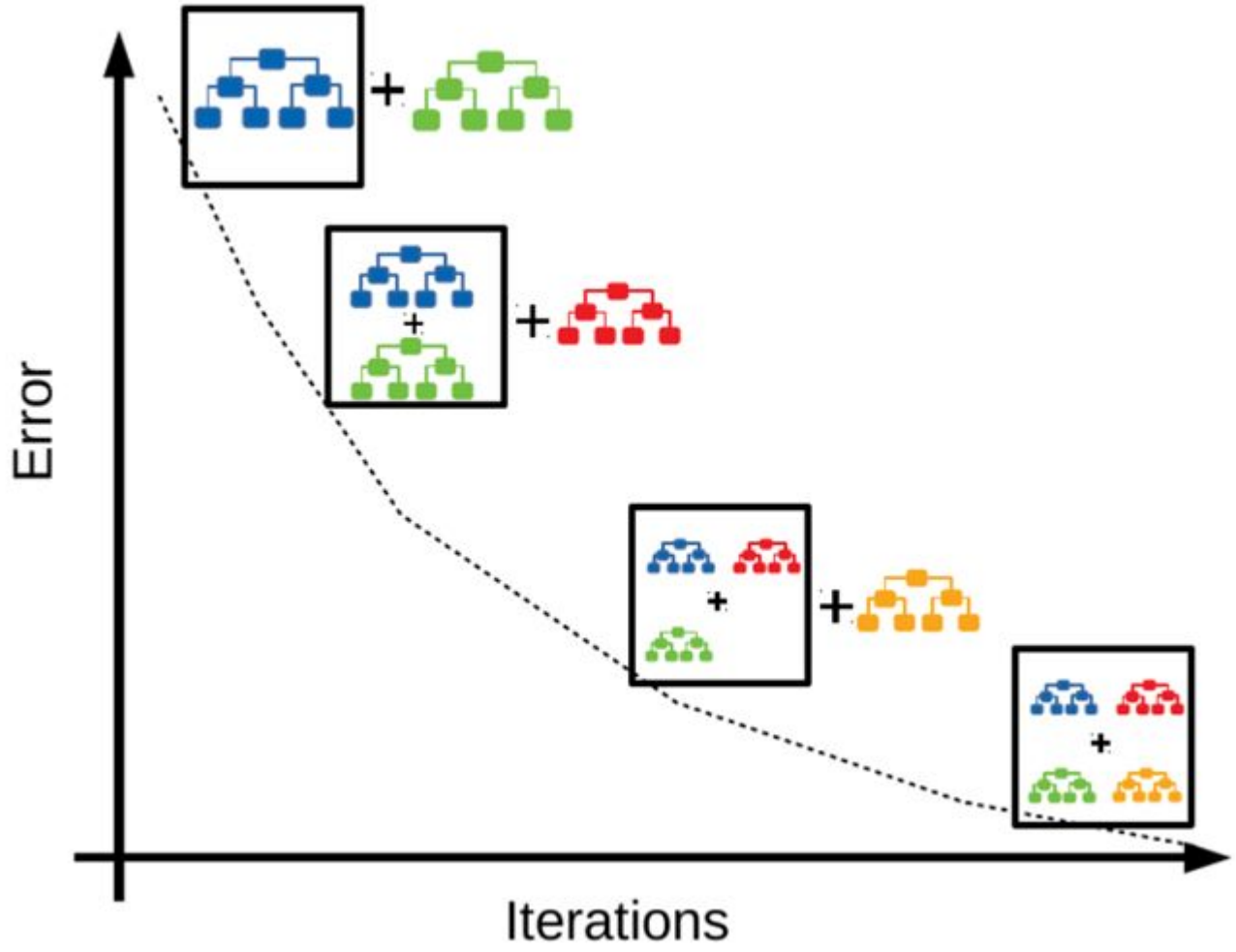
Extra Tree Classifier 기준
depth : 500

Accuracy : 97.4%
F1 Score : 97.3%

생활 질병의 경우에 잘 맞추지 못하여 이를 해결하기 위한 노력을 주로 함

3.모델링

2) 전략 2- Boosting 기법 이용



⇒ Random forest(rf) + extra trees(et)
⇒ Norm / add = Unknown데이터를 다른 열로 분석하여 예측하여 채운 Train_data
⇒ Dis = 파생변수 만들어준 data

- 조합1

0.1 원본 (rf) + 0.2 원본 (et) + 0.3 norm (et) + 0.4 add(et)

F1 score = 85.246

- 조합2

0.2 norm(et) + 0.2add (et) + 0.25 조합1 (et) + 0.35 dis (et)

F1 score = 85.5077

- 조합3

0.2 add (et) + 0.2 조합1 (et) + 0.25dis (et) + 0.35 조합2 (et)

F1 score = 85.5543

-> 각 가중치에 대해서는 모델링을 돌려가며 최적의 값을 찾아냄

3.모델링

```
result = pd.DataFrame()
a = 0.35
b = 0.2
c = 0.25
d = 0.2
# e = 0.1
result['result'] =  (d*result1.target) + (c*result3.target) + (b*result3.target) + (result4.target*a)


for i in range(len(result)):
  a = result.iloc[i,0]
  if a >= 1.35:
    result.iloc[i,0] = 2
  elif 1.35> a >= 0.5:
    result.iloc[i,0] = 1
  else:
    result.iloc[i,0] = 0

data_test['target'] = result.result

dff = data_test[['ID','target']]
dff = dff.astype('int')

dff
```

3.모델링

result1

	ID	target
0	0	0
1	1	1
2	2	1
3	3	1
4	4	0
...	...	...
22067	22067	1
22068	22068	1
22069	22069	0
22070	22070	2
22071	22071	0

22072 rows × 2 columns

전처리 + RF

result2

	ID	target
0	0	0
1	1	1
2	2	1
3	3	1
4	4	0
...	...	...
22067	22067	1
22068	22068	1
22069	22069	0
22070	22070	2
22071	22071	0

22072 rows × 2 columns

전처리 + ETC

result3

	ID	target
0	0	0
1	1	1
2	2	1
3	3	1
4	4	0
...	...	...
22067	22067	1
22068	22068	1
22069	22069	0
22070	22070	2
22071	22071	0

22072 rows × 2 columns

기본 데이터 + RF

result4

	ID	target
0	0	0
1	1	1
2	2	1
3	3	1
4	4	0
...	...	...
22067	22067	1
22068	22068	1
22069	22069	0
22070	22070	2
22071	22071	0

22072 rows × 2 columns

기본 데이터 + ETC

result

	ID	target
0	0	0
1	1	1
2	2	1
3	3	1
4	4	0
...	...	...
22067	22067	1
22068	22068	1
22069	22069	0
22070	22070	2
22071	22071	0

22072 rows × 2 columns

3.모델링

모델링 결과

Auto ML을 통해 앙상블 모델이 가장 성능이 좋게 나왔음을 알 수 있었음

주어진 데이터 전부를 학습 시킨 결과

아무런 전처리를 하지 않고,
RF : F1 score 81%, Extra Tree Classifier : F1 score : 82.5%

전처리 + Boosting + 증상 분리 한 이후
: ***RF : F1 score 83%, Extra Tree Classifier: F1 score : 85.6%***

CHAPTER

4 결론

4. 결과

1) 추가적으로 성능 향상 방법

"데이터분석"은 더욱더 많고 다양한 시도를 해봐야 한다.

① 데이터 업샘플링 적용
② 다양한 파생변수 생성
③ 좀 더 세밀한 파라미터 튜닝
④ 조금 더 다양한 모델조합의 Stacking

데이터 정밀도와 재현률 동시 향상

4. 결과

2) 아쉬운 점

데이터에 대한 도메인 지식 부족	보험 데이터에 관한 도메인 지식이 부족하여 성능 향상에 영향을 주는**추가적인 파생 변수** 형성에 **실패**함
청구 계약일 기간 구분 **결측치 해결의 한계**	청구 계약일 기간이 0인 데이터가 중요한 변수임이기에 **EDA를 통한 대치 법과, MissForest 알고리즘**등을 사용하여 결측치를 채워주었지만, **결과에 대한 향상이 미미**했음
심사(1)과 자동 지급(0)의 구분	Confusion Matrix를 통해 error를 확인하였지만,**심사(1)과 자동 지급(0)을 구분**하는데 모델이 **명확한 해결 방법 제시 못함**
트레인셋 데이터에서 만든 모델의 성능과 테스트셋 에서 만든 모델의 성능 차이	**Train set**에서는 Accuracy : **97%, F1 score : 97%**을 보였으나 **Test set**에선 **F1 score가 85.68%로 10% 이상 차이**가 있었다. 이에 대한 **명확한 해결책**을 찾지 못함. 특히, 오버피팅에 대한 이슈 해결 문제 부족
좀 더 다양한 모델을 stacking 하는 대신 경험적으로 모델 마다 가중치를 주어 boosting 하여 모델의 성능 올렸음	**Stacking 기법을 활용**하여 랜덤 포레스트와 extra tree 알고리즘 이외의**light-gbm, XGBoost 등을 Stacking 하여 사용할 수 있었지만 전처리에 오랜 시간을 할애하여 많은 시도를 하지 못함..**

4. 결과

3) 교수님께 피드백 받고 싶은 부분

① 중요한 피처들을 뽑아 모델을 만들어 보았으나 오히려 성능이 더 떨어졌습니다. 중요 피처를 뽑는 방법에 대한 방법과 중요성에 대해 피드백 받고 싶습니다.

② 결측치의 양이 많으나 이 결측치가 성능에 큰 영향을 줄 때 어떻게 처리해 주어야 하는지 궁금합니다.

③ 도메인 지식이 부족할 때 추가적인 파생 변수 생성 방법에 대해 궁금합니다.

④ 알고리즘을 얼마나 이해하고 사용해야 하는지 궁금합니다. 그리고 이에 대해 추가적으로 어떤 방향으로 공부해 나가야 하는지 궁금합니다.

감사합니다 !

데이터 출처

미래에셋생명보험
한국은행