은행 고객의 정기 예금 가입 분석

정해성

Ⅰ. 개요

기업이 마케팅에 데이터 분석을 사용하는 목적 중 하나는 적은 비용으로 마케팅을 진행하여 최대한의 수익을 창출하기 위함이다. 이는 현재 진행 중이거나 완료한 마케팅의 성과와 원인을 이해하는 것으로부터 시작한다. 그 후 최적화, 고객 세분화. 예측 및 평가 단계를 거쳐 실질적으로 수익 창출을 달성할 수 있다.
이번 프로젝트에서는 통계 분석 방법론을 수강하며 익힌 기법들을 마케팅 관련 데이터에 적용해보고자 고객의 정기 예금 가입 여부에 대한 분석을 프로젝트의 주제로 선정하였다. 데이터는 UCI Machine Learning Repository의 Bank Marketing Data Set을 사용했으며 총 20개의 features와 1개의 binary target variable이 존재한다. 샘플은 총 41,188개이다. 모형의 성능을 평가하기 위하여 EDA 및 전처리 전 층화추출 방법으로 약 10%가량의 4,119개의 테스트 데이터를 별도로 준비하였다. 그 후 나머지 37,069개의 훈련용 데이터로 EDA 및 전처리, 그리고 모형 적합을 진행하였으며, 테스트 데이터를 사용해 분류 결과의 일반화에 대한 평가를 진행하였다. 또한, Oversampling 기법인 SMOTE의 효과를 먼저 살펴본 뒤, 최종 모형을 적합했다. 최종 앙상블 모형의 기초 모형으로 accuarcy가 높은 5가지의 모형들을 선택하고, 각 모형의 적절한 초모수를 recall 기준으로 GridSearchCV를 이용해 결정했다.

Ⅱ. 데이터의 형태

데이터는 포르투갈 은행 기관의 전화 통화를 통해 이루어진 마케팅 데이터이다. 이번 과제의 목표는 마케팅을 통해 수집한 데이터를 이용하여 고객이 정기 예금에 가입할 것인지 여부를 예측하는 모형을 만드는 것이다. 데이터는 크게 4가지의 정보가 들어있으며 각 범주마다의 특성 변수들은 다음과 같다.

<표 1> Bank Marketing Data Set의 변수 설명

범주	변수명	설명
Bank client data	Age	나이
	Job	직업
	Marital	결혼 여부
	Education	교육 수준
	Default	신용 불이행 여부
	Housing	주택 담보 대출 여부
	Loan	개인 융자 여부
Related with the last contact of the	Contact	연락 유형
	Month	마지막으로 연락한 월
	Day_of_week	마지막으로 연락한 요일

current campaign	Duration	마지막 통화 시간(단위:초); **마케팅 종료 전에는 알 수 없는 정보**
Other attributes	Campaign	이번 마케팅 기간동안 연락한 횟수
	Pdays	이전 마케팅에서 마지막 연락 후 경과 일수 (999:이전 연락이 없음)
	Previous	이전 마케팅에서 연락한 횟수
	Poutcome	이전 마케팅의 결과
Social and economic context attributes	Emp.var.rate	고용 변동률-분기별
	Cons.price.idx	소비자 물가지수-월간
	Cons.conf.idx	소비자 신뢰지수-월간
	Euribor3m	3개월 만기 유리보율-일간
	Nr.employed	고용자 수-분기별

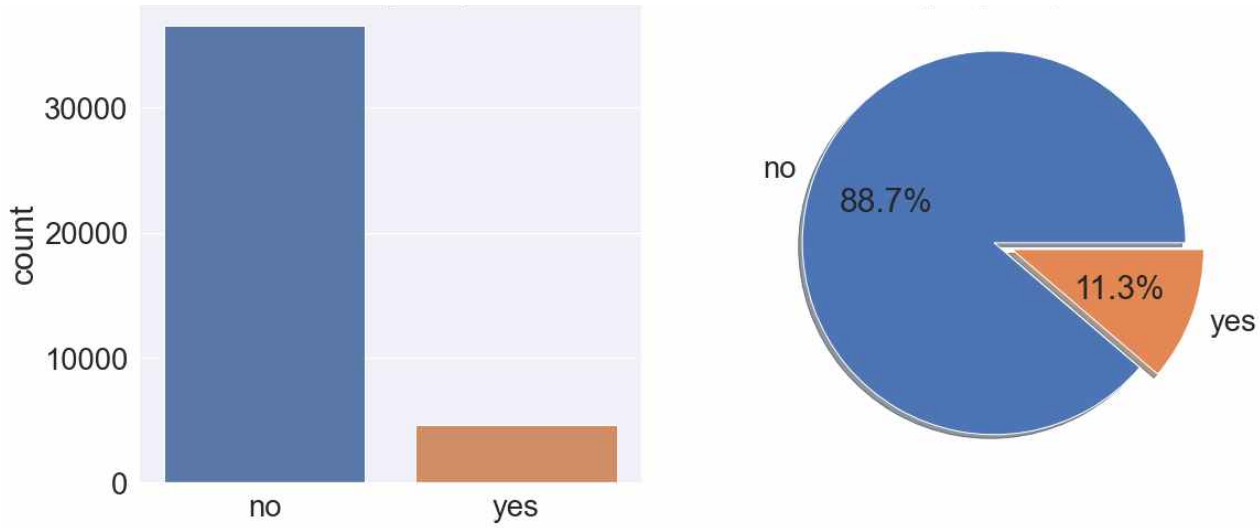
*유리보: 유료화를 단일통화로 하는 유럽연합 12개국의 시중은행 간 단기차입 금리

범주	변수명	설명
종속	y	고객의 정기 예금 가입 여부 ('yes', 'no')

대부분의 독립 변수는 범주형 또는 순서형으로 이루어져 있으며 종속 변수 y는 이항변수이므로 주어진 문제는 이항 분류 문제라고 정의 내릴 수 있다.

주목할 점은 이번 과제의 주요 관심사인 y의 값이 <그림 1>에 나타난 것처럼 'no'가 88.7%, 'yes'가 11.3%로 분포가 매우 불균형하다는 점이다. 그러므로 각 모형의 분류 결과를 평가할 때에도 이러한 불균형을 고려하여야 한다.

<그림 1> 종속 변수 y의 분포



이에 따라 우선 'no'를 0으로 'yes'를 1로 코딩한 뒤 훈련 데이터와 시험 데이터로 분리할 때, y의 0과 1의 비율에 맞게 층화추출을 진행했다. 즉, Train set의 차원은 (37069, 21), Test set의 차원은 (4119, 20)으로 분리했다. 따라서 이후의 분석에서 train set만을 사용해 EDA 및 모형 적합을 시도하였으며, test set은 모형 평가에만 사용하였다.

Ⅲ. EDA 및 전처리

훈련 데이터의 각 범주에 속한 변수들의 분포와 특징을 살펴보고 간단한 전처리를 시행한다. 각 변수를 살펴보기 전, 주어진 데이터의 가장 특이한 점은 <표 2>에서 보듯 몇 개의 변수가 unknown이라는 값을 가진다는 것이다. 즉, 주어진 데이터는 결측값이 'unknown'으로 코딩되어 있다.

<표 2> unknown을 포함하는 변수와 unknown의 개수, 비율 (train)

변수명	unknown의 개수	비율
job	296	0.80%
marital	69	0.19%
education	1564	4.22%
default	7773	20.97%
housing	893	2.41%
loan	893	2.41%

총 6개의 변수가 1개 이상의 'unknown' 값을 가지고 있다. 추후 각 변수에 대해 이 값을 어떻게 처리할지 EDA를 통해 결정하기로 한다. 특히, default의 경우 7773개로 전체 데이터의 약 21%에 달하는 값이 'unknown'이기 때문에 단순히 제거하는 방식은 지양한다.

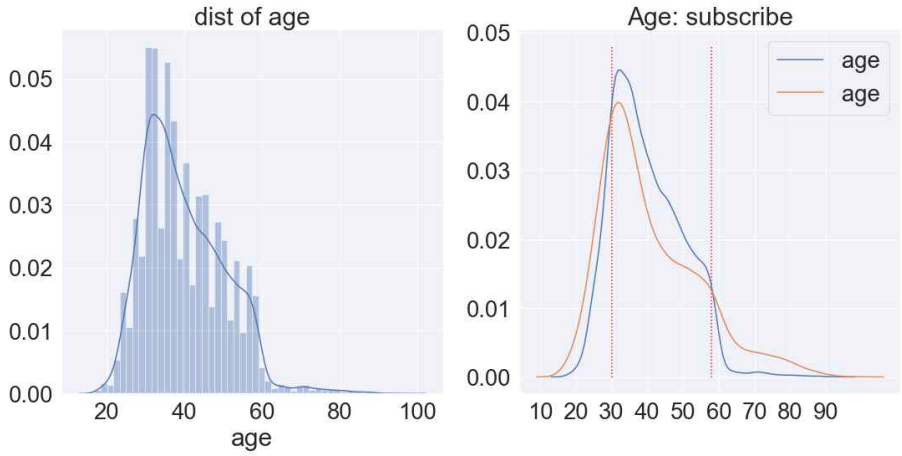
또한, 대부분의 독립 변수가 범주형이기 때문에 범주형 변수인 독립 변수에 대해서는 피어슨의 카이제곱 검정을 통해 종속변수 y와의 연관성을 검정하기로 한다. 특히, unknown이 큰 영향을 미치거나, 독립 변수의 수준에 따라 종속 변수 y와 연관성이 보이지 않는 변수는 피어슨의 카이제곱 통계량을 이용해 변수 선택의 일환으로 제거하기로 한다. 이때 검정을 시행하는 변수 A에 대하여 귀무가설 H_0와 대립가설 H_1은 다음과 같다.

H_0 : A와 y는 독립이다.
H_1 : A와 y는 독립이 아니다.

1. Bank Client Data

1.1 age

<그림 2> age 변수의 분포와 y별 분포



age 변수를 살펴보면 30세 미만과 58세 이상에서 정기 예금에 가입한 사람이 더 많다. 이처럼 나이 분포에 따른 정기 예금 가입 여부가 차이가 나기 때문에 age는 y를 예측하는 데 중요한 변수라고 생각할 수 있으며, age를 30세 이하, 30 ~ 58세 이하, 59세 이상으로 이루어진 age_cat이라는 범주형 변수를 생성했다. age_cat의 각 수준에 따른 정기 예금 상품 가입 여부 비율은 <표 3>과 같다.

<표 3> age_cat과 y의 교차분석표 및 y 값별 비율

age_cat	y		
	no	yes	
0	5640 (84.6%)	1026 (15.4%)	6666
1	26251 (90.8%)	2660 (9.2%)	28911
2	1002 (67.2%)	490 (32.8%)	1492
	32893	4176	37069

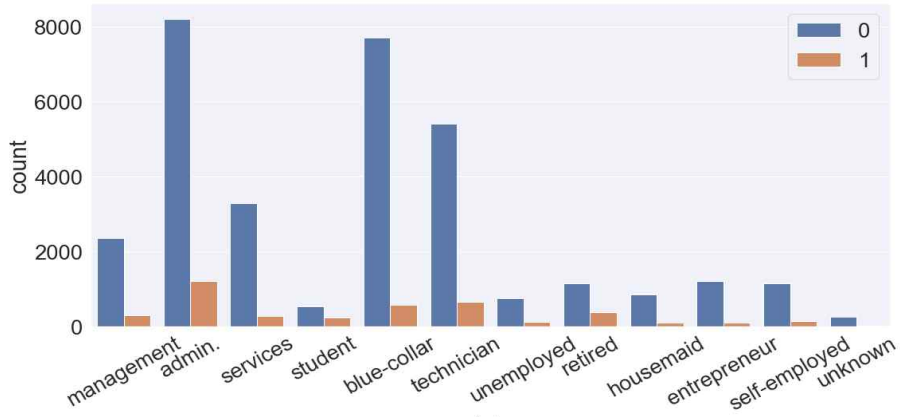
$\chi^2 = 931.7,\ df = 2,$ p<0.001

59세 이상인 사람의 33%가 정기 예금에 가입했으며, 30세 이하는 15%, 31세~58세는 9%로 범주별로 정기 예금 가입률에 차이가 크게 난다. 또한, age_cat과 y의 피어슨 카이제곱 검정의 결과는 p 값이 유의수준 0.05 보다 매우 작으므로 'H_0: age_cat과 y는 독립이다.'를 기각한다.

1.2 job

<그림 3> 직업별 정기 예금 가입률과 그래프

job	student	retired	unemployed	admin.	unknown	management	technician	self-employed	housemaid	entrepreneur	services	blue-collar
y	0.317164	0.253255	0.145856	0.129966	0.118243	0.111867	0.107979	0.105723	0.098517	0.085431	0.080785	0.068187



학생(student)과 은퇴자(retired)의 정기 예금 가입률은 각각 31.7%와 25.3%로 가장 높았으며, 그 외 직업들은 6.8%(blue-collar) ~ 14.6%(unemployed) 사이에 머물러있다. 또한, 'job'은 296개의 'unknown' 값을 갖는데, 총 37,069개의 샘플 중 약 0.8%에 불과하므로 큰 영향을 미치지 않고 있다. 그러므로, job 변수의 값이 unknown인 296개의 관측치를 제거하기로 한다.

1.3 marital

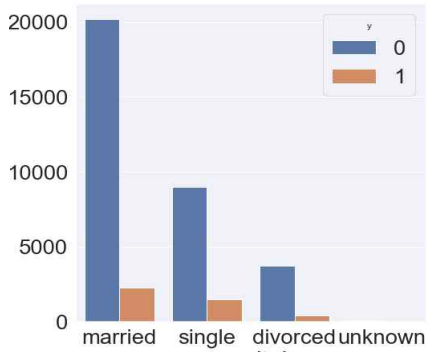
- marital 변수는 총 69개의 unknown을 갖고 있다.

<표 4> marital과 y의 교차분석표 및 y 값별 비율

marital	y		
	no	yes	
divorced	3712 (89.7%)	425 (10.3%)	4137
married	20143 (89.8%)	2281 (10.2%)	22424
single	8980 (86.0%)	1459 (14.0%)	10439
unknown	58 (84.1%)	11 (15.9%)	69
	32893	4176	37069

$\chi^2 = 109.1,\ df = 3,$ p-value<0.001

<그림 4> y별 marital의 분포



또한, 'yes'의 비율 역시 다른 수준과 큰 차이를 보이지 않으므로 'unknown'은 y에 큰 영향을 미치지 않는 것으로 보이므로 marital이 unknown인 관측치를 제거하기로 한다. 또한, marital이 'single'일 때 14.1%로 divorced 혹은 married보다 예금 가입률이 소폭 높다. 또한, marital과 y의 피어슨 카이제곱 검정의 결과는 p 값이 유의수준 0.05 보다 매우 작으므로 'H_0: marital과 y는 독립이다.'를 기각한다. 즉, marital과 y는 연관성이 있다.

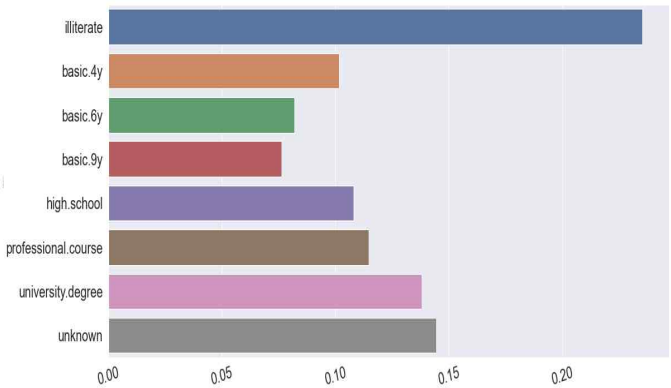
1.4 education

<표 5> education과 y의 교차분석표

education	y		
	no	yes	
illiterate	13	4	17
basic.4y	3367	381	3748
basic.6y	1901	170	2071
basic.9y	5021	415	5436
high.school	7625	924	8549
professional.course	4176	541	4717
university.degree	9452	1515	10967
unknown	1338	226	1564

$\chi^2 = 187.3,\ df = 7,$ p-value<0.001

<그림 5> education 별 정기 예금 가입 비율



순서형 변수인 교육 수준과 정기 예금 가입률은 감소 후 증가하는 특정 패턴이 보인다. 또한, unknown은 전체 훈련 데이터의 약 2%에 해당하는 69개의 값이지만, 정기 예금 가입률이 약 15.9%로 매우 높은 특징을 보이기 있어 쉽게 제거할 수 없다. 따라서, 가장 비슷한 비율을 갖는 'university.degree'로 다시 코딩하는 것이 합리적이라고 판단했다.

1.5 default, housing & loan

- default 변수는 yes가 오직 3개의 값만 존재하며 이는 전체 훈련 데이터의 0.009%에 불과하고 모두 y=0의 값을 갖고 있다. 또한, unknown 값이 7773개나 존재하기 때문에 실질적으로 이 변수는 y에 대한 정보를 주지 않고 있다고 볼 수 있다. 따라서, 모형을 적합할 때 이 변수를 사용하지 않기로 한다.

- 또한, 주택 담보 대출 여부(housing)와 개인 융자 여부(loan)은 각각 893개의 unknown을 갖고 있다. 또한, 'yes', no, unknown 별 y=1의 비율의 차이가 없는 것처럼 보이기 때문에 피어슨의 카이제곱 검정을 진행했다.

<표 6> housing과 y의 교차분석표 및 카이제곱검정 결과

housing	y		
	no	yes	
no	14919	1825	16744
unknown	794	99	893
yes	17180	2252	19432

$\chi^2 = 4.31,\ df = 2,\ p-value = 0.12$

<표 7> loan과 y의 교차분석표 및 카이제곱검정 결과

loan	y		
	no	yes	
no	27065	3453	30518
unknown	794	99	893
yes	5034	624	5658

$\chi^2 = 0.42,\ df = 2,\ p-value = 0.81$

카이제곱 검정결과 housing과 y, loan과 y의 검정 통계량은 각각 4.31, 0.42로 유의수준 0.05 하에서 'H_0 : housing과 y는 독립이다.'와 'H_0 : loan과 y는 독립이다.'라는 귀무가설을 기각하지 못한다. 그러므로, housing과 loan 역시 모델링에 사용하지 않는다.

2. Related with the last contact of the current campaign

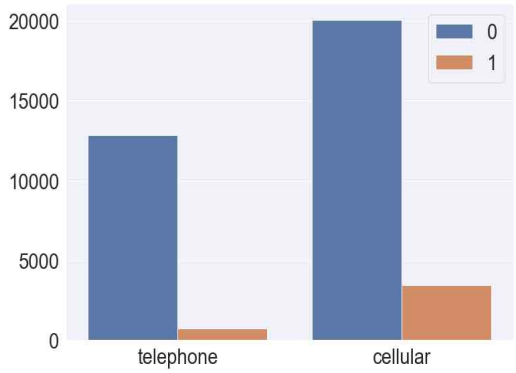
2.1 contact

<표 8> contact와 y의 교차분석표 및 y 값별 비율

contact	y		
	no	yes	
cellular	20053 (85.3%)	3462 (14.7%)	23315
telephone	12840 (94.7%)	714 (5.3%)	13554

$\chi^2 = 767.9,\ df = 1,$ p-value<0.001

<그림 6> y 수준별 contact의 분포



<표 8>과 <그림 6>을 통해 Contact의 수준에 따라 정기 예금 가입률이 크게 달라지는 것을 확인할 수 있다. 이는 카이제곱 검정의 결과와도 일치한다. 즉, 카이제곱 검정 통계량이 767.9로 유의수준 0.05 하에서 기각역인 3.841보다 매우 크므로 'H_0 : contact와 y는 독립이다.'라는 귀무가설을 기각한다. 즉, contact와 y는 연관성이 있다.

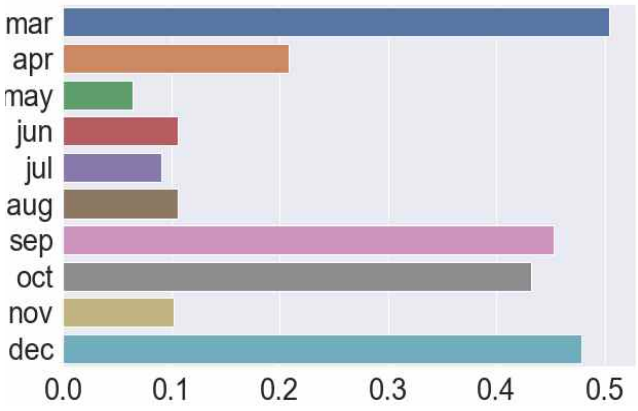
2.2 month

<표 9> month와 y의 교차분석표

month	y		
	no	yes	
mar	246	250	496
apr	1851	486	2337
may	11632	792	12424
jun	4268	503	4771
jul	5909	591	6500
aug	4957	586	5543
sep	288	238	526
oct	366	278	644
nov	3291	374	3665
dec	85	78	163

$\chi^2 = 2791.2,\ df = 9,$ p-value<0.001

<그림 7> month 별 정기 예금 가입 비율



month 변수에서 주목할 점은 1월과 2월에는 데이터가 없다는 것이다. 즉, 마케팅은 모두 3월~12월에 진행되었다. 또한, 대부분의 연락이 may(5월)에 몰려있다. 하지만 가장 적은 비율 가입률을 보이며, 오히려 mar(3월), sep(9월), oct(10월), dec(12월)은 상대적으로 적은 통화량이나 마케팅이 성공(y=1)일 확률이 훨씬 높다. 또한, month와 y의 피어슨 카이제곱 검정의 결과는 p 값이 유의수준 0.05 보다 매우 작으므로 'H_0 : month와 y는 독립이다.'를 기각한다. 즉, month와 y는 연관성이 있다.

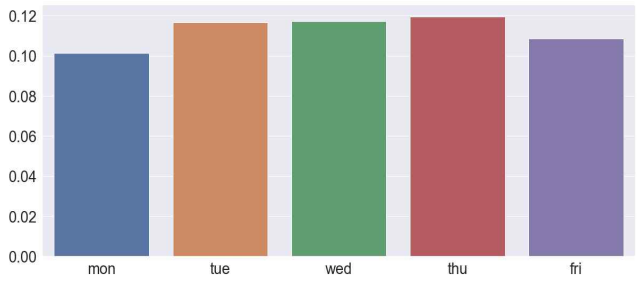
2.3 day_of_week

<표 10> day_of_week와 y의 교차분석표

day_of_week	y		
	no	yes	
mon	6882	777	7659
tue	6411	847	7258
wed	6482	861	7343
thu	6811	924	7735
fri	6307	767	7074

$\chi^2 = 17.21,\ df = 4,$ p-value=0.002

<그림 8> day_of_week 별 정기 예금 가입 비율



day_of_week 변수의 주목할만한 점은 주말에는 마케팅을 진행하지 않았다는 것이다. 마케팅을 목요일에 진행할 때 정기 예금 가입률이 가장 높았다. 또한, day_of_week와 y의 피어슨 카이제곱 검정의 결과는 p 값이 유의수준 0.05 보다 작으므로 'H_0 : day_of_week와 y는 독립이다.'를 기각한다.

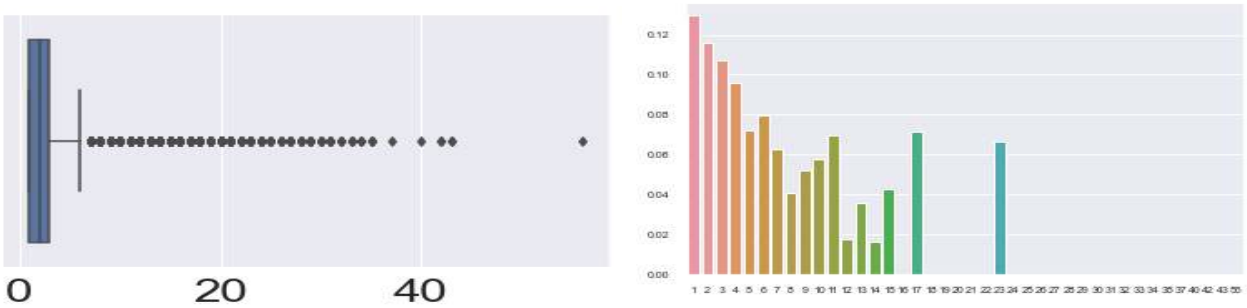
2.4 duration

이번 과제의 목표는 정기 예금 상품에 가입할 확률이 높은 고객을 찾는 것이다. 그러나 duration은 마케팅 통화가 이루어지기 전에는 알 수 없는 변수이므로 제거 후 예측 모형을 만들어야 한다. 예를 들어, durtaion이 0인 고객은 이번 마케팅 기간동안 연락을 취하지 않은 고객이므로 정기 예금 가입을 하지 않았을 것이다. 이처럼 UCI Machine Learning Repository의 데이터 명세서에도 마케팅 수행 전에는 알 수 없는 정보이므로 제거해야 한다고 명시하고 있다. 그러므로 duration 변수는 제거한다.

3. Other Attributes

3.1 campaign

<그림 9> campaign의 상자 그림과 campaign 횟수별 정기 예금 가입 비율



이번 마케팅 기간 동안 고객에게 연락한 횟수를 뜻하는 campaign은 대부분이 10보다 작은 값을 갖는 매우 편향된 분포를 갖는다. 또한, y=1인 비율 역시 campaign이 증가할수록 감소하는 추세가 보이는데 상식적으로 한 번의 마케팅에서 수차례 전화를 하는 것은 상품 가입률을 떨어뜨릴 것이기 때문이다. 따라서 boxplot의 수염(1.5*IQR)을 벗어나는 데이터(8 초과) 1,239개를 이상값으로 여기고 제거한다.

3.2 pdays

pdays는 이전 마케팅에서 고객에게 마지막 연락 이후의 경과 일수지만, 대부분의 값이 999로 다수는 이전 마케팅 대상 고객이 아니다. 그러므로 999면 0 아니면 1의 값을 갖는 이항 변수 pdays_cat으로 변환한다. 변환 후의 교차분석표는 <표 11>에 나와있다. pdays_cat과 y의 피어슨 카이제곱 검정의 결과는 p값이 유의수준 0.05 보다 작으므로 'H_0: pdays_cat과 y는 독립이다.'를 기각한다. 즉, pdays_cat과 y는 연관성이 있다.

<표 11> pdays_cat과 y의 교차분석표

pdays_cat	y		
	no	yes	
0	31152	3241	34393
1	499	875	1374

$\chi^2 = 3814.33$, $df = 1$, p-value<0.001

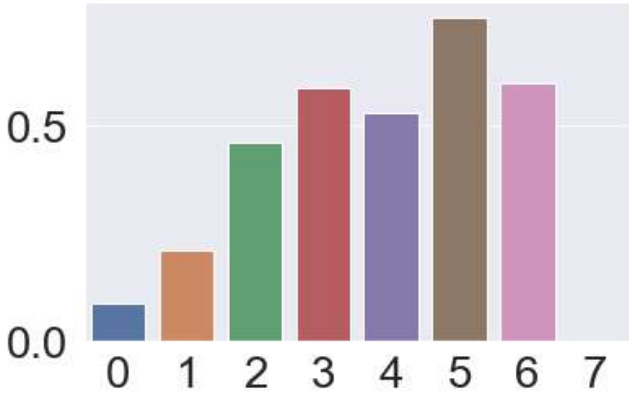
3.3 previous

<표 12> precious와 y의 교차분석표

previous	y		
	no	yes	
0	29165	2824	31989
1	3242	871	4113
2	367	316	683
3	81	115	196
4	31	35	66
5	4	12	16
6	2	3	5
7	1	0	1

$\chi^2 = 1987.5$, $df = 7$, p-value<0.001

<그림 10> previous 별 정기 예금 가입 비율



previous는 순서형 변수로 이전 마케팅에서 고객에게 연락한 횟수를 뜻하며 이전에 한 번이라도 연락을 취했던 고객이라면 정기 예금 가입률이 높아지는 경향이 보인다. 하지만, 피어슨의 카이제곱 검정 통계량은 5보다 작은 cell이 25% 이상이므로 믿을 만하지 못하다. 또한, 3번 이상의 연락을 한 고객의 수가 급격하게 감소하므로 previous가 2 이상이면 1, 아니면 0을 갖는 변수 previous_cat을 생성하여 추후 분석에 사용하기로 결정했다.

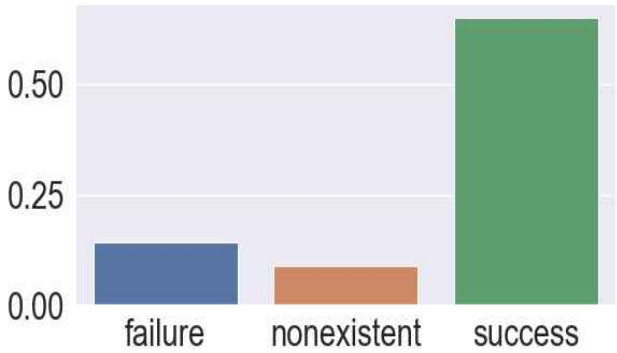
3.4 poutcome

<표 13> poutcome과 y의 교차분석표

poutcome	y		
	no	yes	
failure	3291	543	3834
nonexistent	29165	2824	31989
success	437	809	1246

$\chi^2 = 3690.9$, $df = 2$, p-value<0.001

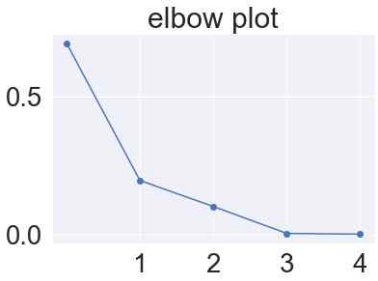
<그림 11> poutcome 별 정기 예금 가입 비율



이전 마케팅의 결과를 나타내는 변수로, success의 값에서 y=1의 비율이 매우 높다. 즉, 이전 마케팅이 성공적으로 끝났다면, 이번 마케팅도 성공했을 가능성이 높은 모습을 보인다. 뿐만 아니라, 이전에 실패했던 고객이라도 이전에 마케팅을 하지 않은 고객에 비해 정기 예금 상품에 가입하는 경향성을 보인다. 그러므로 이전 마케팅을 실패한 고객에게 다시 연락하는 것이 중요해 보인다. 또한, 피어슨 카이제곱 검정의 결과는 p 값이 유의수준 0.05 보다 작으므로 'H_0: poutcome과 y는 독립이다.'를 기각한다. 그러므로 nonexistent와 failure면 0, 성공이면 1의 값을 갖는 변수로 변환했다.

4. Social and Economic Context Attributes

이 범주에 속하는 총 5개의 변수들은 모두 연속형 변수들이며 사회 및 경제 지표를 나타내는 변수들이다. 이러한 변수들의 피어슨 상관계수는 <그림 13>과 같다. 상관계수를 살펴보면, 0.97, 0.91, 0.95 등으로 매우 높은 것을 확인할 수 있다. 현실적으로 은행이 마케팅을 진행할 때 사회 및 경제 지표를 나타내는 변수들을 통제하는 것은 불가능한 일이다. 따라서 5개 변수를 표준화한 후, PCA를 통해서 차원 축소를 고려했다. 3개의 주성분만으로도 99%의 분산을 설명하고 있으며, <그림 12>의 elbow plot을 고려해서 3개의 주성분을 사용하기로 했다.

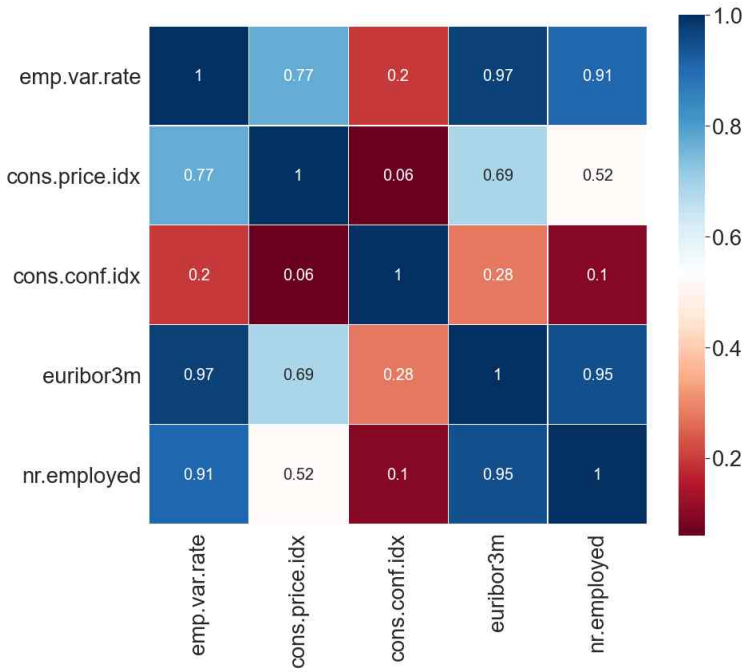


<그림 12> Elbow plot

<표 14>에는 여태까지 논의한 독립 변수들에 대한 전처리 결과가 정리되어 있다. 'age', 'pdays', 'previous' 변수는 각각 'age_cat', 'pdays_cat', 'previous_cat'으로 범주형 변수로 변환하였다. 또한, 데

이터 명세서와 카이제곱 검정의 결과를 참고하여 'default', 'housing', 'loan' 및 'duration' 변수를 제거하였다. 또한 피어슨의 상관계수를 활용해 상관계수가 매우 높게 나온 사회 및 경제 지표 변수들을 PCA를 통해 3개의 주성분을 선택했다. 'campaign' 변수는 상자그림을 활용해 이상치를 제거하였다.

<그림 13> 사회 및 경제 지표 변수들의 피어슨 상관계수



<표 14> 변수별 전처리 결과

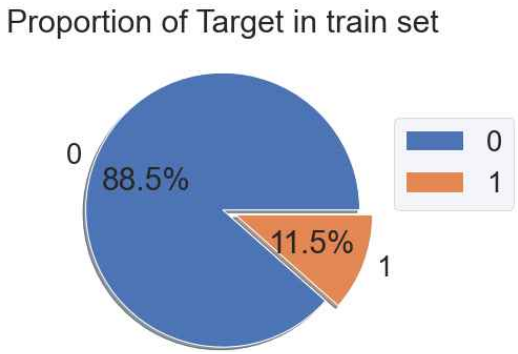
범주	특성 변수	전처리
Bank client data	Age	age_cat(0: $x \leq 30$, 1: $30<x<58$, 2: $60 \leq x$)
	Job	더미 변수화
	Marital	더미 변수화
	Education	수치화(순서형 변수)
	Default	제거
	Housing	제거
	Loan	제거
Related with the last contact of the current campaign	Contact	더미 변수화
	Month	더미 변수화
	Day_of_week	더미 변수화
	Duration	제거
Other attributes	Campaign	8 이상인 이상치 제거
	Pdays	pdays_cat(0: 999, 1: o/w)
	Previous	previous_cat(0: $x=0,1$, 1: $2 \leq x$)
	Poutcome	poutcome_cat (0:failure|nonexistent, 1:success)
Social and economic context attributes	Emp.var.rate	PCA(n_component=3)
	Cons.price.idx	
	Cons.conf.idx	
	Euribor3m	
	Nr.employed	

5. 최종 데이터의 형태

최종적으로 완성된 훈련 데이터 X(타겟 y)와 시험 데이터 X_test(타겟 y_test)는 각각 (35821, 38), (3990, 38)의 shape을 갖는다. SMOTE의 효과를 알아보기 위해 train_test_split함수로 훈련 데이터를 다시 각각 (28656, 38), (7165, 39)의 shape을 갖는 훈련 데이터 X_tr와 검증용 데이터 X_vld로 분할하였다. 이 데이터는 최종 모델링 과정에서는 사용하지 않았다. 최종 모델링에는 전체 훈련 데이터 X와 y를 사용하여 GridSearchCV 함수를 적용했기 때문이다.

Ⅳ. SMOTE의 효과

앞서 보았듯이 훈련 데이터의 y의 값은 <그림 1>과 같이 'no'가 88.7%, 'yes'가 11.3%로 분포가 매우 불균형하다. 이러한 경우 모형 훈련 시 모형 역시 불균형한 학습이 이루어질 가능성이 높다. 단적인 예로 훈련 데이터의 모든 샘플에 대해서 타겟을 0('no')으로 예측한다면 Accuracy는 88.7%를 기록할 것이다. 그러므로 모형 평가의 기준이 Accuracy만으로는 부족하다.



<그림 14> target

이번 과제의 목표는 고객이 정기 예금에 가입할지 여부를 예측하는 것이다. 이러한 목표 하에서 다음과 같은 2가지의 오류가 발생할 수 있다.

1. 고객이 정기 예금에 가입하지 않았는데 가입했다고 예측하는 False Positive
2. 고객이 정기 예금에 가입했는데 가입하지 않았다고 예측하는 False Negative

주어진 상황에서 더 줄여야 하는 오류를 골라야 한다면, False Negative라고 할 수 있다. 즉, y가 'no'인 경우보다 y가 'yes'인 것을 잘 분류해내는 것이 중요하다. 이미 가입한 고객을 미가입자로 분류한 뒤 마케팅을 진행하는 것은 가입하지 않은 고객에게 마케팅을 진행하지 않는 것보다 비용 측면에서 손해이기 때문에 실제 가입한 고객들을 제대로 분류하는 것이 더 중요하다. 따라서, 모형 평가 시 이를 더 잘 반영할 수 있는 Recall과 Accuracy를 함께 비교하기로 한다. 여기서 $(\text{Recall}) = \text{TP}/(\text{TP}+\text{FN})$이다. 즉 FN가 작다는 말은 Recall이 크다는 것과 동일하다.

그러므로 최종모형은 다음과 같은 단계로 생성하기로 결정했다. 우선 accuarcy가 높은 5가지의 모형들을 앙상블의 기초 모형으로 사용하고, 각 모형의 적절한 초모수를 recall 기준으로 GridSearchCV를 이용해 결정한다. 그 후, 적절한 초모수를 선택하는 GridSearchCV 시 가장 우선 시 되는 것은 Recall을 높이는 것이며, 이를 모형 훈련 시 scoring에 전달하기로 했다.

우선, SMOTE를 사용했을 때의 효과를 살펴보자. 이를 실험해보기 위해 35821개의 샘플이 있는 훈련 데이터를 train_test_split(startify=y)로 각각 (28656, 38)과 (7165, 38)의 shape을 갖는 X_tr, X_vld 그리고 (28656,)과 (7165,)의 shape을 갖는 y_tr, y_vld로 나눴다. SMOTE를 적용한 결과는 다음과 같다.

```
print('SMOTE 적용 전 Train data set: ', X_tr.shape, y_tr.shape)
print('SMOTE 적용 후 Train data set: ', X_tr_smote.shape, y_tr_smote.shape)
print('SMOTE 적용 전 레이블 값 분포: ₩n', pd.Series(y_tr).value_counts())
print('SMOTE 적용 후 레이블 값 분포: ₩n', pd.Series(y_tr_smote).value_counts())
```

```
SMOTE 적용 전 Train data set:  (28656, 38) (28656,)
SMOTE 적용 후 Train data set:  (50706, 38) (50706,)
SMOTE 적용 전 레이블 값 분포:
 0    25353
1     3303
Name: y, dtype: int64
SMOTE 적용 후 레이블 값 분포:
 1    25353
0    25353
Name: y, dtype: int64
```

그 후, X_tr과 y_tr로 모형을 훈련하고 X_vld, y_vld로 평가한 결과와, X_tr_smote, y_tr_smote로 모형을 적합하고 X_vld, y_vld로 평가한 결과를 비교했다. 이때 사용한 모형들은 KNN, Logistic Regression, SVM(kernel='rbf'), Decision Tree, Random Forest, XGBoost, LightGBM 등 7개의 모형이며, 각 초모수는 기본적으로 정해져있는 디폴트 값을 사용했다.

	accuracy		precision		recall		f1	
	original	sm	original	sm	original	sm	original	sm
knn	0.903	0.851	0.624	0.422	0.406	0.788	0.492	0.550
logit	0.905	0.863	0.637	0.444	0.406	0.754	0.496	0.559
svm	0.906	0.865	0.712	0.455	0.311	0.849	0.433	0.592
decision tree	0.894	0.881	0.538	0.488	0.553	0.643	0.545	0.555
rf	0.911	0.898	0.645	0.548	0.501	0.679	0.564	0.606
xgb	0.905	0.896	0.600	0.535	0.519	0.736	0.557	0.619
lgbm	0.912	0.892	0.636	0.520	0.542	0.820	0.586	0.636

<표 15> SMOTE 전·후의 평가지표 값

비교한 7개 모형 모두에서 약간의 accuracy와 precision은 감소하지만, 정기 예금 가입자를 가입했다고 올바르게 예측하는 recall, 그리고 precision과 recall의 조화평균인 f1 score가 모두 증가한 것으로 보아 SMOTE를 사용하는 것이 더 좋다고 판단할 수 있다. 그러므로 SMOTE를 훈련 데이터에 사용해 최종 모형을 만들기로 했다.

Ⅴ. 앙상블의 기초 모형 선정

앞서 언급했듯 앙상블의 기초 모형의 기준은 accuracy를 통해 5개의 상위 모델을 결정한 뒤, 이들에 대해 recall을 기준으로 최적의 초모수를 찾기로 했다. 따라서, 전체 훈련 데이터에 SMOTE를 적용한 뒤 모

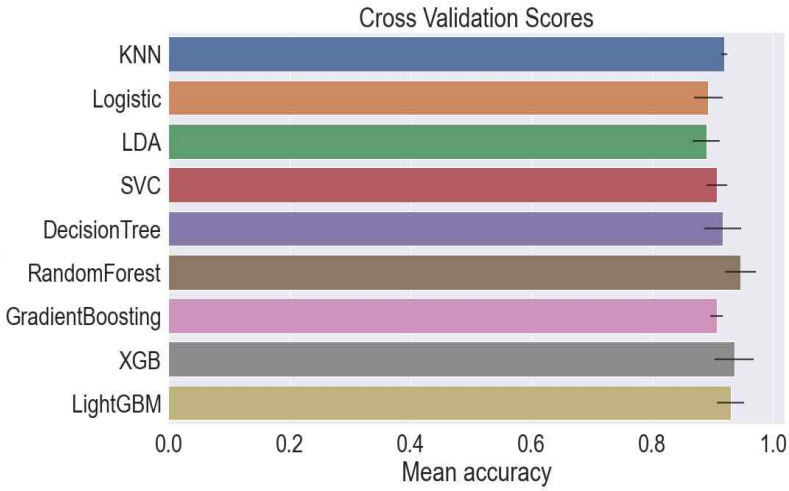
형을 훈련시키기 위해, 다음과 같이 X_smote와 y_smote를 생성했다.

<표 16> SMOTE 적용 전,후의 Train data

SMOTE	# of sample	y labels	
		0	1
Before	35,821	31,692	4,129
After	63,384	31,692	31692

그 후, cross_val_score함수에 모형과 X_smote, y_smote를 훈련 데이터로, scoring='accuracy'로 설정한 뒤, StratifiedKFold(n_splits=10)으로 생성한 객체를 인자로 넣어 10-fold CV를 수행했다. 그런 다음 KNN, Logitstic Regression, LDA, SVC(kerenl='rbf'), Decision Tree, Random Forest, Graident Boosting, XGBoost, LightGBM 등 9개의 모형에 대해 교차 검증 평균과 표준편차를 구했다. 결과는 다음과 같다.

<그림 15> 9개 모형의 CV Score 결과



<표 17> 9개 모형의 CV Score 결과

Algorithm	CV mean	CV std
RandomForest	0.9455	0.0255
XGBoost	0.9357	0.0322
LightGBM	0.9300	0.0225
KNN	0.9192	0.0050
Decision Tree	0.9173	0.0395
SVC	0.9075	0.0169
Gradient Boosting	0.9068	0.0103
Logistic Regression	0.8928	0.0236
LDA	0.8896	0.0223

이 결과를 바탕으로, CV accuracy 평균이 높은 Random Forest, XGBoost, LightGBM, KNN과 Decision Tree를 앙상블 모형의 기초 모형으로 결정했다.

Ⅵ. 각 모형의 초모수 결정

SMOTE를 사용해서 교차검증을 통한 초모수 결정을 시행할 때의 가장 중요한 점은, oversampling이 교차검증 이전에 이루어져서는 안된다는 것이다. 그 이유는 교차검증을 수행하기 전에 SMOTE를 사용했다면, 이를 평가하는 Validation Fold가 Unseen이라는 가정을 만족하지 못하기 때문이다. 그러므로 SMOTE기법은 교차검증 전이 아니라, 교차검증을 수행하면서 이루어져야 한다. 예를 들어 5-fold cv를 진행할 때, 훈련 데이터로 사용되는 4개의 fold에 대해서 SMOTE를 적용한 뒤 모형을 훈련해야 한다는 것이다. 그 후 validation set에 대해 예측하고 평가해야 한다. 이러한 과정을 다음과 같이 for 반복문과 imblearn 패키지 내의 make_pipeline을 사용해 진행했다.

[KNN의 초모수 결정 - 교차검증 수행 중 oversampling(SMOTE)]

```
from imblearn.pipeline import make_pipeline as imb_pipeline

skf = StratifiedKFold(n_splits=5, random_state=0)
accuracy_lst_knn = []
precision_lst_knn = []
recall_lst_knn = []
f1_lst_knn = []
auc_lst_knn = []

knn_sm2 = KNeighborsClassifier()

knn_param = {"n_neighbors":list(range(1,20))}

gs_knn_sm2 = GridSearchCV(knn_sm2,param_grid = knn_param, cv=skf, scoring="recall", n_jobs= -1, verbose = 1)

for train, test in skf.split(X_train_original, y_train_original):
    pipe = imb_pipeline(SMOTE(sampling_strategy='minority'), gs_knn_sm2)
    model = pipe.fit(X_train_original[train], y_train_original[train])
    best_est_knn = gs_knn_sm2.best_estimator_
    pred = best_est_knn.predict(X_train_original[test])
    accuracy_lst_knn.append(pipe.score(X_train_original[test], y_train_original[test]))
    precision_lst_knn.append(precision_score(y_train_original[test], pred))
    recall_lst_knn.append(recall_score(y_train_original[test], pred))
    f1_lst_knn.append(f1_score(y_train_original[test], pred))
    auc_lst_knn.append(roc_auc_score(y_train_original[test], pred))

print("accuracy: {}".format(np.mean(accuracy_lst_knn)))
print("precision: {}".format(np.mean(precision_lst_knn)))
print("recall: {}".format(np.mean(recall_lst_knn)))
print("f1: {}".format(np.mean(f1_lst_knn)))
```

이 과정을 거쳐 얻은 앙상블 모형의 최종 베이스 모형들은 다음과 같다.

모형	hyperparameters
RandomFoestClassifier	n_neighbors=5
XGBClassifier	base_score=0.5, booster='gbtree', colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1, gamma=0, learning_rate=0.05, max_depth=9, min_child_weight=1, n_estimators=50
LGBMClassifier	learning_rate=0.05, n_estimators=50, num_leaves=35
KNeighborsClassifier	criterion='entropy', max_features=10, min_samples_leaf=10, n_estimators=200
DecisionTreeClassifier	criterion='entropy', max_depth=3

Ⅶ. 앙상블 모형(최종 모형)

예측력을 높이고자 앞에서 구한 5개의 모형들을 활용해 앙상블 모형 즉, hard 방식의 VotingClassifier를 생성하기로 한다. 과정은 VotingClassifier에 앞서 구한 모형들과 함께 voting='hard'를 전달하고, 전체 훈련 데이터에 적용한 X_smote, y_smote를 활용해 훈련시킨다. 그리고 이 모형을 최종모형으로 정했다.

```
vote_smote = VotingClassifier(estimators=[('knn', best_est_knn),('rf',best_est_rf),('dt', best_est_dt),('lgbm',best_est_lgbm),
                                          ('xgb',best_est_xgb)], voting='hard', n_jobs=-1)
vote_smote.fit(X_smote, y_smote)
```

최종 모형을 사용해 구한 훈련 데이터와 시험 데이터에 대한 예측 결과는 다음과 같다.

Confusion Matrix: Train set			
		Prediction	
		0	1
True	0	28045	3647
	1	630	31062

Confusion Matrix: test set			
		Prediction	
		0	1
True	0	3088	447
	1	43	412

Classification Report: Train set				
	Precision	Recall	F1_score	Support
0	0.98	0.88	0.93	31692
1	0.89	0.98	0.94	31692
accuracy			0.93	63384
macro avg	0.94	0.93	0.93	63384
weighted avg	0.94	0.93	0.93	63384

Classification Report: Test set				
	Precision	Recall	F1_score	Support
0	0.99	0.87	0.93	3535
1	0.48	0.91	0.63	455
accuracy			0.88	3990
macro avg	0.73	0.89	0.78	3990
weighted avg	0.93	0.88	0.89	3990

훈련 데이터에 대한 Accuracy는 0.93이고 시험 데이터에 대한 Accuracy는 0.88이다. 그러나 Target variable y가 불균형한 자료이기 때문에 Recall을 살펴봐야 한다. 우리의 관심은 y=1에 대한 예측이므로 각 Classification Report의 1이 있는 행을 살펴보면, 훈련 데이터에 대한 Recall은 0.98, 시험 데이터에 대한 Recall은 0.91이다. 훈련 데이터에 대한 것만큼은 아니지만 시험 데이터에 대해서도 어느 정도 실제 정기 예금 상품에 가입한 고객들을 모형이 올바르게 예측하는 비율이 꽤 높다는 것을 확인할 수 있다. 그러나 과대적합의 문제가 보인다는 점이 아쉬움으로 남는다.

Ⅷ. 한계점 및 느낀점

배운 이론을 데이터에 응용해보니 많은 것을 배웠다. 우선, 불균형 데이터이다 보니 모형 훈련 시 편향이 생길 수 있어 resampling 기법을 사용했다. 그중에서도 일반적으로 통계적으로 유용한 과대표집 방법 중 SMOTE 방법을 사용했다. 그러나 처음에는 각 모형의 초모수를 결정하는 과정에서 교차검증 수행 전에 SMOTE를 적용하고 심한 과대적합의 결과를 얻었다. 구글링을 통해 이유를 찾아보니 oversamling이나 undersampling 기법은 교차검증을 진행하는 과정에서 사용해야 한다는 것을 배웠다. 교차검증을 수행할 때 validation set은 unseen data여야 하는데, 만약 미리 SMOTE를 사용한다면 그 의미가 퇴색되기 때문이다. 그러나 oversampling 기법 중 과대적합을 피하는 SMOTE 방법과 이를 교차검증 내에서 수행했지만, 여전히 시험 데이터에 대해 과대적합이 보였다는 점은 추후 개선해야 할 점이라고 생각한다.