고급통계적머신러닝

최종 과제

고려대학교 대학원

통계학과 응용통계학전공

2020021202 이혜원

목차

I. 서론

AI 인공지능의 시대가 도래하면서 방대한 데이터를 효율적으로 분석하기 위한 머신러닝의 중요성이 커지고 있다. 우리는 머신러닝을 통해 현실세계의 다양한 데이터를 분석하고 데이터로부터 결과를 예측하여 각 분야에서의 전략 수립 및 인사이트 발굴할 수 있다. 머신러닝을 제대로 활용하기 위해서는 통계적 개념을 정확히 이해하고, 각 머신러닝 알고리즘의 비교분석을 통해 알고리즘을 적재적소에 활용할 수 있는 역량을 키워야 한다. 특히 데이터 특성에 따라 각 데이터에 어떤 알고리즘을 이용해 예측을 하는 것이 적절한지 파악하는 것이 중요하다. 이러한 필요에 맞추어, 이 보고서에서는 실제 데이터에 기반한 분류(classification) 결과를 다양한 머신러닝 모델을 이용하여 예측하고 각 모델이 어떤 이유로 해당 결과를 가져왔는지 분석해볼 것이다.

II. 탐색적 데이터분석 및 특성공학

1. 분석 데이터

보고서에서 활용할 데이터는 UCI machine learning laboratory의 Bank Marketing Dataset이다. 이 데이터는 총 45211개의 관측값과 17개의 변수로 이루어져 있는데, 여기서 y 변수는 예측 대상인 target variable이다. 대략적인 변수에 대한 설명은 아래 첨부했다.

Bank marketing 데이터를 활용한 머신러닝 분석의 목적은 bank marketing을 했을 때의 고객이 정기예금(term deposit) 가입 여부를 예측하는 것이다. 즉, target 값인 y 가 yes/no를 예측하는 것으로 다양한 분류 알고리즘을 통해 예측을 진행해야 한다.

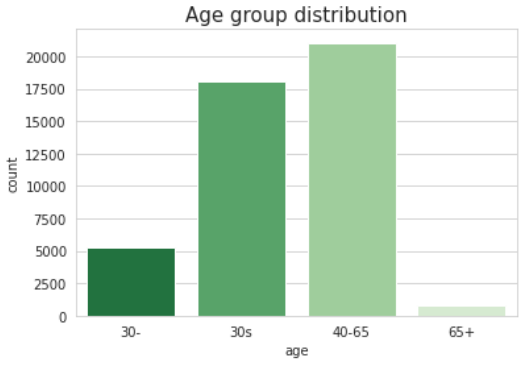
	Variable	Type	Description
Bank client data	age	integer	Client's age
	job	category	Type of job
	marital	category	Marital status
	education	category	Education level
	default	category	Credit in default(yes/no)
	balance	float	Balance amount
	housing	category	Housing in loan(yes/no)
	loan	cateogry	Personal loan(yes/no)

Last contact of Current campaign	contact	category	Contace communication type
	day	integer	Last contacted day
	month	category	Last contacted month of year
	duration	integer	Last contact duration
Other attributes	campaign	integer	Number of contacts during current campaign
	pdays	integer	Number of days passed after the client was Contacted from a previous campaign
	previous	integer	Number of contacts performed before this current campaign
	poutcome	category	Outcome of previous marketing campaign
target	y	category	Subscribe a term deposit(yest/no)

2. 데이터 전처리(Data Preprocessing)

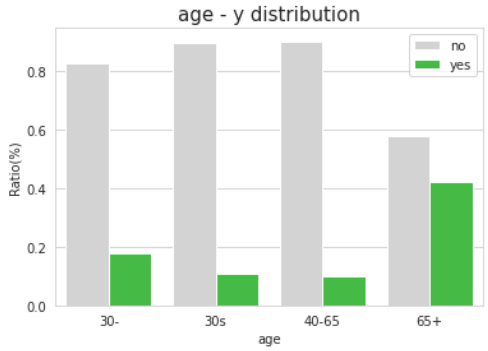
본격적인 분석에 앞서, 각 데이터에 대한 간단한 탐색과 함께 모델에 활용하기 위한 데이터 정제 및 파생변수를 생성한다.

우선 age 변수의 은행 고객 나이는 18세부터 95세까지 다양한 연령대로 구성되어 있다. 연령대를 나타내는 변수는 일반적으로 개별로 보는 것보다 연령대로 그룹화하여 분석하는 것이 더 용이하기 때문에 총 4개의 그룹으로 그룹화했다. 10-20대는 '30-' 그룹, 30대는 '30s' 그룹, 40-65세는 '40-65' 그룹, 65세 이상의 노년층은 '65+' 그룹으로 지정했다. 각 그룹에 속하는 고객의 수는 아래 [그림1]에서 확인할 수 있다. 40-65세의 고객이 가장 많고, 그 다음은 30대 고객이 많다.



[그림] 1

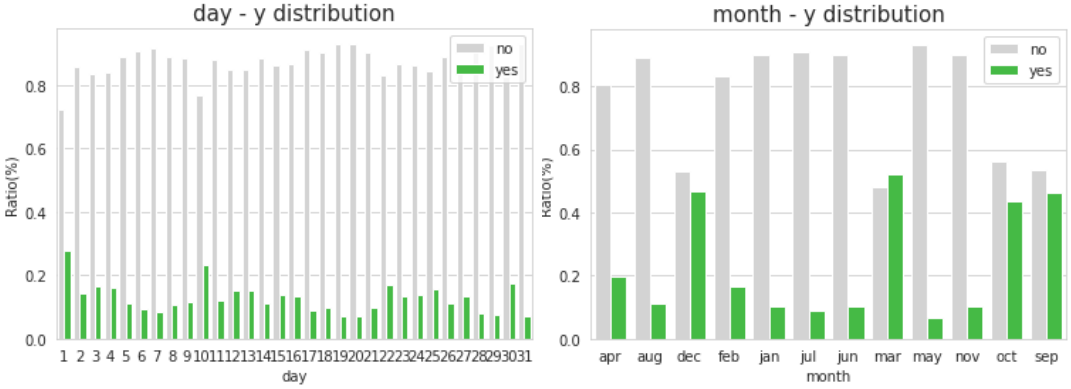
여기서 드는 의문점은 age를 그룹화한 것이 과연 분류 결과에 유의미한 영향을 끼칠까? 라는 것이다. 이를 확인하기 위해 각 연령대 그룹별로 y의 yes/no의 비율이 얼마나 되는지 파악했다. [그림2]를 보면 65세 이상의 노년층이 마케팅을 통해 정기예금에 가입하는 비율이 다른 연령대에 비해 높은 것을 확인할 수 있다. 그리고 연령대 그룹별로 yes/no 비율의 차이가 있기 때문에 이 그룹화 결과가 유의미하다고 보인다.



[그림] 2

또한 day와 month 변수의 설명에 따르면 이 두 변수는 각각 현재 진행중인 campaign에 대한 마지막 contact 날짜의 일과 월을 나타낸다. 위 age 변수와 마찬가지로 day와 month에 대해 yes/no 비율 그래프를 그린다. [그림3]을 보면 day(일)에 따른 target value의 비율 차이는 거의 없지만, month에 따라서는 target 비율 차이가 큰 것을 볼 수 있다. 따라서, day 변수는 예측에 유의미한 영향을 미치지 않을 것으로 예상되므로 제거한다.

[그림] 3



마지막으로 pdays 변수를 살펴보자. 이 변수는 이전 캠페인 당시 contact을 한 이후로 며칠이 지났는지를 나타내는 변수이다. Pdays 변수의 summary를 하면 -1 값을 가지는 데이터가 총 36954개로 전체의 81.74%를 차지한다. 일반적으로 '-1일이 지났다' 라는 표현은 쓰지 않기 때문에 -1 값이 이전 캠페인 때 contact을 받지 않았다는 것을 나타낸다는 것을 알 수 있다. 이전 캠페인 당시 contact을 받지 않은 고객이 전체의 80% 이상으로 매우 큰 비율을 차지하기 때문에 pdays 변수를 제거하는 것이 예측에 더 유리할 것으로 판단했다.

3. 추가 전처리

2.3.1 범주형 변수

Object type의 범주형 변수들을 그대로 model의 input으로 넣으면 model이 이 변수들을 인

식하지 못한다. 따라서 범주형 변수를 숫자형으로 encoding하는 과정이 필요하다. 따라서 이 변수들에 대해 One-hot Encoding을 한다. One-hot encoding을 하면 특정 카테고리에 속하면 1, 그렇지 않으면 0으로 표시를 한다.

Target 변수인 y 또한 값이 yes/no로 object type이다. 모델 학습을 위해 y 변수도 숫자형으로 encoding을 해야 하는데, 이에 대해서는 one-hot encoding이 아닌 label encoding으로 진행한다. Label encoding을 적용하면 yes는 1로, no는 0으로 encoding이 된다.

2.3.2 연속형 변수

Bank marketing 데이터에서 balance, duration, previous 변수가 연속형 변수이다. 이 세변수에 대한 왜도(skew)를 계산하면 모두 3 이상으로 데이터의 분포가 왜곡된 것을 알 수 있다. 심하게 skewed된 데이터는 분류 예측 시 tail 부분의 적은 양의 데이터가 학습이 되지 않는 경우가 발생한다. 따라서 왜도가 큰 변수에 대해서는 스케일링(scaling)을 해주는 것이 중요한데, 본 분석에서는 가장 많이 쓰이는 scaling 방법 중 하나인 MinMaxScaler를 사용하였다.

III. 분류 모델링(Classification Model)

머신러닝 분류 모델은 '분류 결과 예측' 이라는 같은 목적을 갖고 있지만 그 방식이 각자 다르다. 각 모델은 배깅(bagging), 부스팅(boosting), KNN, 회귀 등등 다양한 알고리즘을 기반으로 한다. 따라서 본 보고서에서는 **bagging 알고리즘 기반의 Random Forest, boosting 기반의 LightGBM, 회귀 기반의 Logistic Regression** 모델의 결과를 분석하고 비교해볼 것이다.

1. 모델링 과정 및 세부 설명

모델링 과정은 다음과 같이 진행한다. 이 과정은 세 가지 모델에 모두 적용된다.

Step 1: feature set X 와 target set y 생성

Step 2: Nested cross validation을 통한 모델 성능 평가 및 파라미터 튜닝

Step 3: Step 2의 최적 파라미터로 세부 모델 성능 평가

Step 4: Smote oversampling을 적용한 후 Step 2, Step 3 반복

Step 1에서는 모델링의 input인 feature set X와 예측해야 할 target y를 생성하고 이 input을 이용해 Step 2에서 모델 학습을 진행한다. 이 과정에서 중요한 것은 과적합 (overfitting)을 방지하고 모델의 성능을 최대화할 수 있는 모델의 초모수(hyper parameter)를 찾는 것이다. 과

적합을 방지하기 위해 가장 많이 사용되는 방법은 교차검증(cross validation)이다. 일반적인 K-fold 교차검증 방법은 과적합 개선에 도움이 되지만, 학습 모델의 성능이 자료 분할 방식에 의존한다는 단점이 있다. 따라서 본 보고서에서는 기존 K-fold의 단점을 보완하고 파라미터 튜닝까지 할 수 있는 nested cross validation을 이용해 모델 학습을 진행한다.

3.1.1 Nested cross validation (Step 2)

Nested cross validation에서는 outer loop와 inner loop을 각각 정의한다. Outer loop에서는 기존의 K-fold처럼 train set을 K-1개의 fold로 나누어 돌아가면서 학습을 한다. Inner loop는 outer loop의 train fold에 적용되는 루프인데, inner loop에도 K-fold를 적용하여 파라미터를 tuning한다. 본 보고서에서 outer loop와 inner loop는 다음과 같이 설정했다. Nested cross validation을 통해 우리는 각 outer loop 5개에서 얻은 performance의 평균을 계산해 전체 모델의 최종 performance를 얻을 수 있다. 여기서 performance는 accuracy로 설정한다.

Loop	Options	Role
Outer loop	Stratified 5-fold, shuffle True	Train with optimal parameter from inner loop
Inner loop	Stratified 2-fold, shuffle True	GridSearchCV - hyperparameter tuning

3.1.2 최적 파라미터 기반 모델 성능 평가 (Step 3)

Nested cross validation을 통해 각 outer loop에서 가장 우수한 파라미터와, 그 파라미터를 활용한 모델의 performance를 구할 수 있었다. 모델의 학습 결과 및 성능을 더 세부적으로 평가하기 위해 outer loop에서 구한 5개의 파라미터 중 가장 높은 성능을 내는 파라미터를 선택한 후, 이 파라미터를 적용한 모델을 이용해 세부 모델 평가를 진행한다.

2. 모델 3종류 학습 결과

3.2.1 종합 모델 성능 평가 및 비교

(1) 모델 최종 performance

다음 표는 세 가지 모델의 nested cross validation 최종 결과를 나타낸 자료이다. 즉, outer loop를 통해 얻은 각 loop의 최고 performance의 평균을 낸 것이다.

Model	Train Accuracy	Test Accuracy
Random Forest	0.9634	0.9033

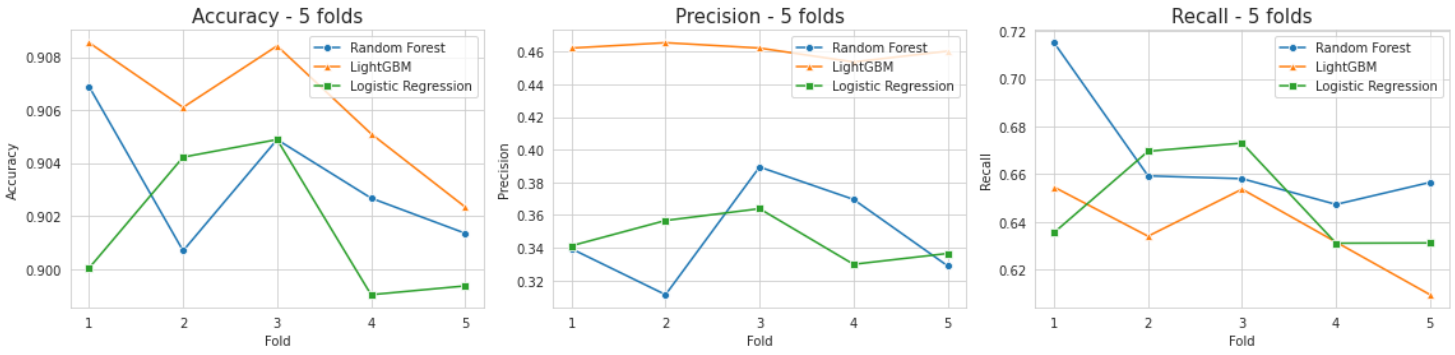
LightGBM	0.9259	0.9061
Logistic Regression	0.9016	0.9015

세 가지 모델의 최종 성능 평과 결과를 해석하면 다음과 같다.

모델 학습 과정에서 과적합이 발생했는지를 알아보자. 과적합(overfitting)은 모델이 train data를 과하게 학습하여 train data에 대해서는 performance가 좋지만, test data에 대해서는 performance가 떨어지는 현상을 말한다. 즉 과적합이 발생하면 train data의 accuracy는 거의 1에 수렴하는 반면, test data의 accuracy는 낮아져 그 차이가 매우 클 것이다. Nested cross validation 결과 세 모델 모두 train accuracy와 test accuracy의 차이가 모두 1 미만으로 거의 없기 때문에 모델 학습에서의 과적합 문제는 발생하지 않았다는 것을 알 수 있다.

과적합이 발생하지 않은 이유는 무엇일까? 가장 기본적인 이유는 교차검증을 했기 때문이다. 교차검증 외에도 우리는 nested cross validation을 통해 초모수를 적합했다. 트리 기반 모델(Random Forest, LightGBM)은 트리가 과도하게 깊어지면 train data를 깊고 세밀하게 학습하기 때문에 과적합 위험이 있지만, 트리 최대 깊이인 max_depth를 제한하면 과적합을 막을 수 있다. 로지스틱 회귀의 파라미터에서는 C 파라미터를 조정했다. C는 규제 강도를 나타내는 λ 의 역수이다. 적절한 λ 값, 즉 C값을 찾으면 로지스틱 회귀의 손실함수를 최소화하면서 과적합을 방지할 수 있다.

(2) 모델 3종류 5-folds performance 비교



위 그래프는 nested cross validation의 총 5 fold에서 세 가지 모델의 accuracy, precision, recall이 어떻게 변화하는지를 비교한 결과이다. Performance 측정 기준인 accuracy는 LightGBM 이 다른 두 모델보다 높다.

1) LightGBM은 boosting 기반 알고리즘이다. Boosting 알고리즘의 핵심은 한 개의 train data

에 weak learner을 순차적으로 결합하는 것이다. Weak learner을 여러 개 결합함으로써 오분류된 train data의 가중치는 높이고, 제대로 분류된 train data의 가중치를 줄이는데, 이 과정을 통해 모델의 performance가 높아지는 것이다.

2) Bagging 기반의 Random Forest의 경우, 여러 개의 sampling된 데이터에 각각 모델을 적용하여 예측 결과의 평균을 내서 최종 결과를 산출한다. 이러한 알고리즘의 차이로 일반적으로 boosting 알고리즘이 bagging보다 좋은 성능을 보이는 경우가 많다.

3) 로지스틱 회귀는 기본적으로 선형 모형을 가정하기 때문에 데이터가 선형을 띄지 않는 경우 성능이 다른 모델모다 낮을 수 있다. 이러한 이유로 accuracy만 봤을 때, 대체적으로 로지스틱 회귀의 performance가 제일 낮다.

또한 precision과 recall을 비교하면 흥미로운 결과를 볼 수 있다. Precision을 비교하면 LightGBM이 다른 두 모델보다 더 높지만, recall은 가장 낮다. 이는 precision과 recall 사이에 trade-off가 존재하기 때문에 나타나는 현상으로, 일반적으로 precision기 증가하면 recall은 감소한다.

3.2.2 모델 세부 성능 평가

섹션 3.2.1에서는 nested cross validation에서 얻은 최종 모델 성능 평가 및 5-fold 성능 비교를 다루었다. 섹션 3.2.2에서는 nested cross validation 과정에서 얻은 5개의 초모수 중 가장 높은 accuracy를 산출한 초모수를 이용하여 모델 세부 성능평가를 한다. 이 파트에서는 각 target에 대한 모델의 precision, recall 성능을 중심으로 살펴볼 것이다.

(1) Random Forest with hyper parameter

Random Forest	Predicted 0 (no)	Predicted 1 (yes)
Actual 0 (no)	7840	145
Actual 1 (yes)	724	334

Random Forest의 5개의 최적 파라미터 중 fold accuracy가 0.9069로 가장 높았던 파라미터는 n_estimators:200, max_depth: 15 이다. Target을 예측 시 0(no) 예측에 대한 precision은 0.92, recall은 0.98로 매우 높았으나, 1(yes) 예측에 대한 precision은 0.7, recall은 0.32로 recall이 매우 낮았다. Recall은 실제로 1인 데이터 중 1로 예측된 데이터의 비율을 나타내는데, recall이 0.32라는 것은 Random Forest 모델이 1(yes)를 잘 예측하지 못한다는 것을 의미한다.

(2) LighGBM with hyper parameter

LightGBM	Predicted 0 (no)	Predicted 1 (yes)

Actual 0 (no)	7709	276
Actual 1 (yes)	574	484

LightGBM의 accuracy가 가장 높았던 파라미터는 n_estimators:200, max_depth:10, num_leaves=50 이다. LightGBM 역시 0(no) 값은 잘 예측한다. 1(yes)에 대한 recall 값은 0.46으로 Random Forest나 Logistic Regression보다는 1(yes) 예측 성능이 더 우수하다.

(3) Logistic Regression with hyper parameter

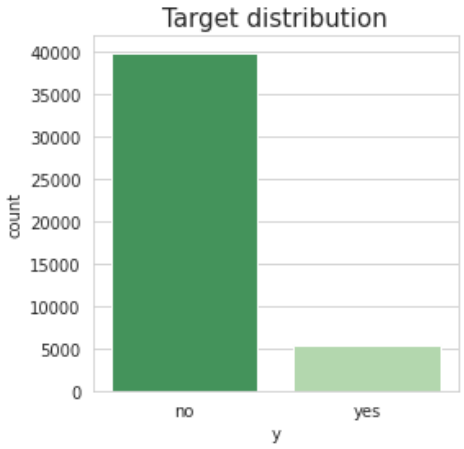
Logistic Regression	Predicted 0 (no)	Predicted 1 (yes)
Actual 0 (no)	7783	202
Actual 1 (yes)	704	354

Logistic Regression의 최고 accuracy 파라미터는 'L2 규제' 및 C=5 이다. Logistic Regression 예측 결과는 Random Forest와 비슷하다. Target 1에 대한 예측력이 0에 대한 예측력보다 현저히 떨어진다.

3. Oversampling 모델 3종류 학습 결과

3.3.1 Oversampling의 필요성

모델 3종류의 세부 성능 평가에서 알 수 있듯이, 세 모델 모두 target 0(yes)에 대한 예측 성능은 매우 좋으나 1을 제대로 예측하지 못한다는 단점이 있다. 그 원인을 bank marketing 데이터의 y 분포에서 찾을 수 있다.



왼쪽의 Target distribution plot을 보면 target 값의 수가 매우 불균형하다는 것을 알 수 있다. 이런 경우, 0(no) 데이터가 대부분이므로 이에 대한 예측력은 높아지지만 상대적으로 적은 수를 차지하는 1(yes)에 대한 예측력은 감소하게 된다. 이런 문제를 방지하기 위해 수가 적은 target 값의 수를 늘려주는 알고리즘인 oversampling 기법을 적용한다. Oversampling 기법 중 가장 많이 사용되는 SMOTE oversampling 알고리즘을 사용해 데이터를 oversampling 한 후, 앞의 3.2.2와 3.2.3 과정을 반복한다. Undersampling을 하면 데이터의 손실이 매우 많기 때문에 오히려 모델의 성능을 떨어뜨릴 수 있다고 판단하여 하지 않았다.

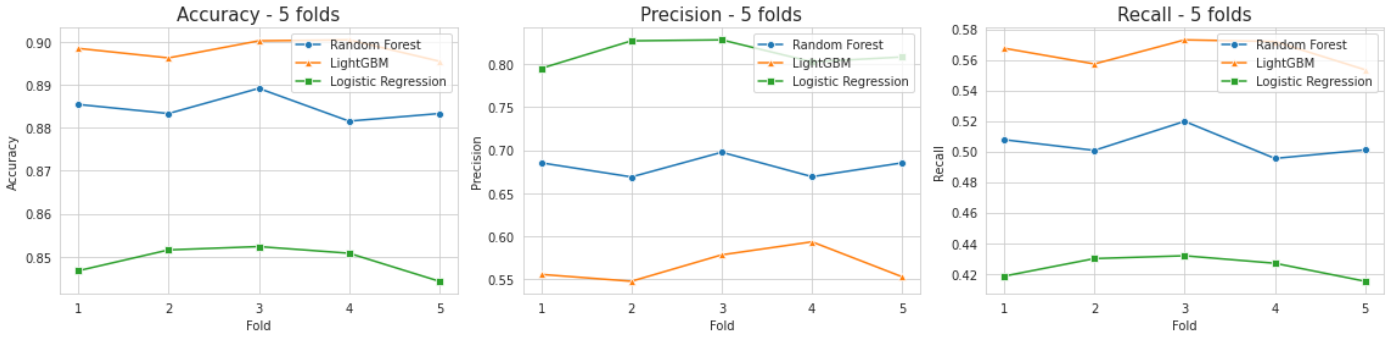
3.3.2 SMOTE 적용 후 모델 3종류 종합 평가

(1) SMOTE 모델 최종 performance

Model	Train Accuracy	Test Accuracy
Random Forest	0.9754	0.8846
LightGBM	0.9792	0.8982
Logistic Regression	0.8536	0.8492

SMOTE 적용 후Train accuracy와 test accuracy의 차이가 증가했다. Oversampling 후 Random Forest와 LightGBM의 train test accuracy의 차이가 커졌다. 이는oversampling을 적용한 모델이 적용하지 않은 모델보다 과적합 취약하다는 것을 의미한다. Oversampling을 하면 기존의 데이터를 단순 복제해서 데이터의 양을 늘리기 때문에 과적합 위험이 더 커지게 된다.

(2) 모델 3종류 5-folds performance 비교



SMOTE 적용 후 nested cross validation에서 모델 performance가 어떻게 변화하는지를 확인했다.

1) Accuracy는 SMOTE 적용 후가 전보다 살짝 낮았으나, 유의미한 차이를 보이지는 않았다. SMOTE 적용 전과 마찬가지로 accuracy는 LightGBM>Random Forest> Logistic Regression 순으로 높았다. 이유는 섹션 3.2.1의 (2)와 동일하다.

2) 전체적인 Precision과 recall은 역시 trade-off를 보였다. LightGBM이 recall은 가장 높지만 precision은 가장 낮았다. 각 target 값에 대한 세부적인 recall과 precision은 다음 장에서 설명한다.

(3) 모델 세부 성능 평가

Random Forest	Predicted 0 (no)	Predicted 1 (yes)
Actual 0 (no)	7239	692
Actual 1 (yes)	328	730

LightGBM	Predicted 0 (no)	Predicted 1 (yes)
Actual 0 (no)	7501	484
Actual 1 (yes)	432	626

Logistic Regression	Predicted 0 (no)	Predicted 1 (yes)
Actual 0 (no)	6792	1193
Actual 1 (yes)	189	869

위 표를 보면 실제 target이 1(yes)인 데이터를 예측을 1(yes)로 한 비율, 즉 1(yes)에 대한 recall이 oversampling 전에 비해 증가했다. 이는 oversampling을 통해 target 0과 1에 해당하는 데이터 수를 맞춤으로써 1에 대한 예측 성능이 더 증가했다는 것을 의미한다. 또한 0(no)에 대한 recall, precision도 모두 0.9 이상으로 0에 대한 예측 성능을 유지하였다.

IV. 결론 및 의의

본 보고서에서는 분류 모델을 이용해 bank marketing data의 정기예금 가입 여부를 예측하였다. Nested cross validation을 통해 과적합 방지와 파라미터 튜닝을 동시에 진행하고, 이 과정에서 구한 초모수를 이용해 모델에 대한 세부 평가를 할 수 있었다.

또한 oversampling 기법 적용 여부에 따른 모델링 결과를 비교하는데 의의가 있었다. 클래스가 불균형한 데이터셋에 oversampling을 적용하면 과적합 위험이 증가하고 최종 accuracy도 미세하게 감소하지만, oversampling 전 모델보다 소수의 클래스 1(yes)에 대한 recall은 증가했다. 필자의 관점에서는 모델 3종류의 최종 performance가 oversampling 전과 거의 차이가 없고, 0(no)에 대한 precision, recall를 유지함과 동시에 1(yes)에 대한 recall을 증가했다는 점에서 oversampling이 의의가 있다고 판단했다. 하지만, oversampling 기법은 항상 과적합 위험을 수반하기 때문에 이를 방지하기 위한 다른 기법에 대한 공부가 더 필요하다고 생각했다.

Appendix

https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

고려대학교 통계학과 대학원 고급통계적머신러닝 강의자료 Ch 5, 10, 12