

모델 학습 결과서

팀: 1조(김나현, 문창교, 이경현, 이승규, 정래원)

프로젝트명: 고객 이탈 예측 (Customer Churn Prediction)

데이터 출처: Kaggle – Bank Customer Churn Dataset

데이터 규모: 총 10,000개 행, 12개 열

1. baseline 설정

1-1) 파이프라인 구축

본 프로젝트에서 baseline 모델 선정에 고려한 분류 모델은 다음과 같이 총 8종이다. Logistic Regression / KNN / SVM / Decision Tree / Random Forest / Bagging / AdaBoost / Neural Network(MLPClassifier)

8종의 모델 성능을 일관된 방식으로 비교·평가하기 위해, Scikit-learn의 `ColumnTransformer` 와 `Pipeline` 을 활용한 통합 전처리 및 학습 구조를 설계했다.

먼저, 전처리기는 `ColumnTransformer` 를 이용해 연속형 변수와 범주형 변수에 서로 다른 변환 과정을 병렬적으로 적용하도록 구성했다.

- 연속형 변수: 모델의 수학적 특성에 따라 `StandardScaler` 또는 `MinMaxScaler` 를 적용
- 범주형 변수: `OneHotEncoder` 를 적용하여 비수치형 데이터를 머신러닝 모델이 처리할 수 있도록 변환
- 트리 기반 모델(RandomForest, DecisionTree 등): 스케일링이 불필요하므로, `passthrough` 옵션을 사용하여 원본 값을 그대로 전달하도록 설정

이러한 방식은 모델 특성에 따라 유연하게 전처리를 적용하면서도, 각 모델이 동일한 데이터 처리 흐름 내에서 학습되도록 보장한다.

이후 전처리기와 모델을 결합하기 위해 `Pipeline` 객체를 구성하였다. `Pipeline` 은 `preprocessor` 와 `model` 두 단계로 이루어져 있으며, [데이터 전처리 → 모델 학습 → 예측 과정]을 하나의 흐름으로 처리한다. 이를 통해 데이터 누수를 방지하고, 모델을 변경하더라도 동일한 구조에서 손쉽게 비교·평가를 수행할 수 있다.

1-2) basline 단계 공통 설정

모든 모델은 하이퍼파라미터를 지정하지 않고 기본값(default) 으로 학습했다.

또한 동일한 데이터 분할(`train_test_split(test_size=0.2, stratify=y, random_state=42)`)을 적용했다.

2. baseline 모델 선정

2-1) 성능 평가 기준: Recall & F1

본 프로젝트의 목적은 ‘이탈 고객 탐지’로, 단순 정확도(Accuracy)보다는 이탈 고객을 얼마나 잘 찾아내는가가 핵심이다. 따라서 다음 두 지표를 모델 성능 평가의 기준으로 삼았다.

- Recall(민감도) : 실제로 이탈(1)한 고객 중, 모델이 이탈(1)이라고 잘 예측한 비율
즉, Recall은 “이탈 고객을 놓치지 않고 잘 찾아냈는가?”를 보여주는 지표다.

- F1-score(조화 평균) : Precision(정밀도)와 Recall의 균형

Recall만 높이면 실제로 이탈하지 않은 고객도 이탈로 예측할 위험이 커진다. 따라서 Precision과 Recall 사이의 균형을 함께 고려하는 것이 중요하다

즉, F1-score은 이탈 고객을 많이 잡되 과도한 오탐(False Positive)을 최소화하는 균형 지표다.

2-2) 모델 학습 및 성능 측정

train/test 성능 차이(과적합 정도)를 함께 고려해, 일반화 성능이 안정적인 모델을 선별했다.

아래 표는 모델이 test 데이터에 대해 예측한 결과다.

표 1.1 Test 데이터에 대한 각 모델의 성능 측정 (정렬: F1 내림차순)

Model	ROC_AUC	Accuracy	F1	Recall
AdaBoost	0.8464	0.8634	0.6021	0.5103
NN	0.8547	0.8609	0.5864	0.4872
Bagging	0.8186	0.8505	0.5727	0.4949
LogisticRegression	0.8487	0.8598	0.5714	0.4615
RandomForest	0.8501	0.8577	0.5595	0.4462
SVM	NaN	0.8634	0.5504	0.4128
KNN	0.7995	0.8406	0.5165	0.4205
DecisionTree	0.6919	0.7970	0.5069	0.5154

사진 1.2 Test 데이터에 대한 각 모델의 Classification Report

● 모델: LogisticRegression				
	precision	recall	f1-score	support
0	0.88	0.96	0.92	1536
1	0.75	0.46	0.57	390
accuracy			0.86	1926
macro avg	0.81	0.71	0.74	1926
weighted avg	0.85	0.86	0.85	1926

● 모델: KNN				
	precision	recall	f1-score	support
0	0.87	0.95	0.90	1536
1	0.67	0.42	0.52	390
accuracy			0.84	1926
macro avg	0.77	0.68	0.71	1926
weighted avg	0.83	0.84	0.83	1926

● 모델: SVM				
	precision	recall	f1-score	support
0	0.87	0.98	0.92	1536
1	0.83	0.41	0.55	390
accuracy			0.86	1926
macro avg	0.85	0.70	0.73	1926
weighted avg	0.86	0.86	0.84	1926

● 모델: DecisionTree				
	precision	recall	f1-score	support
0	0.88	0.87	0.87	1536
1	0.50	0.52	0.51	390
accuracy			0.80	1926
macro avg	0.69	0.69	0.69	1926
weighted avg	0.80	0.80	0.80	1926

모델: AdaBoost					
	precision	recall	f1-score	support	
0	0.88	0.95	0.92	1536	
1	0.73	0.51	0.60	390	
accuracy			0.86	1926	
macro avg	0.81	0.73	0.76	1926	
weighted avg	0.85	0.86	0.85	1926	

모델: NN					
	precision	recall	f1-score	support	
0	0.88	0.96	0.92	1536	
1	0.74	0.49	0.59	390	
accuracy			0.86	1926	
macro avg	0.81	0.72	0.75	1926	
weighted avg	0.85	0.86	0.85	1926	

모델: RandomForest					
	precision	recall	f1-score	support	
0	0.87	0.96	0.92	1536	
1	0.75	0.45	0.56	390	
accuracy			0.86	1926	
macro avg	0.81	0.70	0.74	1926	
weighted avg	0.85	0.86	0.84	1926	

모델: Bagging					
	precision	recall	f1-score	support	
0	0.88	0.94	0.91	1536	
1	0.68	0.49	0.57	390	
accuracy			0.85	1926	
macro avg	0.78	0.72	0.74	1926	
weighted avg	0.84	0.85	0.84	1926	

F1, Recall이 모두 양호하면서, Train과 Test 점수 차이가 적었던 모델은 다음 3개였다:

- Logistic Regression
- AdaBoost
- Random Forest

이 세 모델을 baseline 모델로 선정했고, 성능을 높이기 위해 하이퍼파라미터 튜닝을 진행했다.

3. 하이퍼파라미터 튜닝

3-1) 하이퍼파라미터 튜닝 수행: RandomizedSearchCV

baseline 모델들의 일반화 성능을 극대화하기 위해 하이퍼파라미터 튜닝(Hyperparameter Tuning) 을 수행했다. 각 모델의 탐색 공간을 사전에 정의한 후, RandomizedSearchCV를 활용하여 최적 조합을 탐색하였다.

(튜닝 과정에서는 F1-score를 평가 지표로 사용함.이는 본 프로젝트의 핵심 목표가 단순 Accuracy가 아니라, 이탈 고객(소수 클래스)을 놓치지 않으면서도 오탐을 최소화하는 것이기 때문임.)

RandomizedSearchCV 의 옵션은 다음과 같다.

- RandomizedSearchCV(n_iter=30, cv=5, scoring='f1', random_state=42)
- 병렬 연산(n_jobs=-1)으로 탐색 효율화

3-2) 하이퍼파라미터 튜닝 결과

RandomizedSearchCV 결과, 각 모델의 최적 하이퍼파라미터는 아래와 같이 도출되었다.

모델	주요 최적 하이퍼파라미터
Logistic Regression	C=77.9693 max_iter=1544 penalty='l1' solver='saga'
Random Forest	max_depth=17 max_features='log2' min_samples_leaf=3 min_samples_split=13 n_estimators=104
AdaBoost	learning_rate=1.2272 n_estimators=285

최적 하이퍼파라미터 적용 후, 각 모델의 성능은 다음과 같다.

표 1.3 Test 데이터에 대한 baseline 모델의 성능 측정 (정렬: F1 내림차순)

Model	ROC_AUC	Accuracy	F1	Recall
RandomForest	0.856	0.858	0.649	0.649
AdaBoost	0.846	0.864	0.593	0.487
LogisticRegression	0.847	0.784	0.585	0.751

사진 1.4 Test 데이터에 대한 baseline 모델의 Classification Report

● 모델: LogisticRegression

```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
      precision    recall   f1-score   support
      0         0.93     0.79     0.85     1536
      1         0.48     0.75     0.58      390

      accuracy          0.78
      macro avg       0.70     0.77     0.72     1926
  weighted avg       0.84     0.78     0.80     1926
```

● 모델: RandomForest

```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
      precision    recall   f1-score   support
      0         0.91     0.91     0.91     1536
      1         0.65     0.65     0.65      390

      accuracy          0.86
      macro avg       0.78     0.78     0.78     1926
  weighted avg       0.86     0.86     0.86     1926
```

● 모델: AdaBoost

```
Fitting 5 folds for each of 30 candidates, totalling 150 fits
      precision    recall   f1-score   support
      0         0.88     0.96     0.92     1536
      1         0.76     0.49     0.59      390

      accuracy          0.86
      macro avg       0.82     0.72     0.76     1926
  weighted avg       0.86     0.86     0.85     1926
```

3-3) 최종 모델 선정 및 결론

최종적으로 선정된 모델은 **Random Forest**로,

하이퍼파라미터 튜닝을 거친 결과 **F1-score 0.649, Recall 0.649**로 두 지표 모두 높은 수준의 균형 성능을 보였다.

이는 모델이 단순히 정확도(Accuracy)에 치우치지 않고, 이탈 고객(1)을 놓치지 않으면서도 불필요한 오탐(False Positive)을 줄이는 방향으로 학습되었음을 의미한다.

또한, 모델의 깊이(max_depth)와 최소 분할 기준(min_samples_split, min_samples_leaf)을 적절히 조정함으로써 과적합을 완화하고 일반화 성능을 향상시킬 수 있었다.

베이스라인 모델 대비 F1-score는 $0.559 \rightarrow 0.649(+0.09)$ 로 개선되었으며, 특히 Recall 상승으로 이탈 고객 탐지력이 강화되었다.

Test 데이터에서도 성능이 안정적으로 유지되어, 튜닝 결과가 학습 데이터에 과도하게 의존하지 않고 일관된 예측력을 확보한 것으로 평가된다.

