Netflix 고객 이탈률 예측 모델 산출물

II. 인공지능 모델 학습 결과서

목적

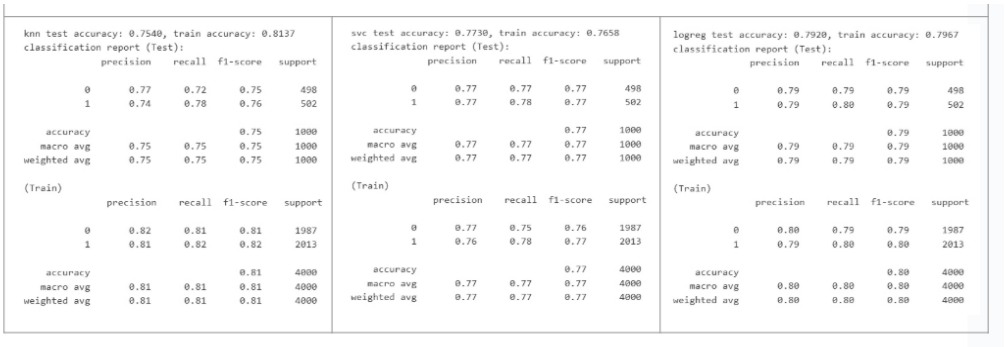
사용자 행동 데이터를 기반으로 고객 이탈 여부를 예측하고, 개인화된 구독 추천 서비스에 활용 가능한 모델 개발.

1. 모델 선택 및 설계

고객 이탈 예측은 이진 분류(Binary Classification)에 해당
이진 분류모델을 택하여 순차적으로 모델의 성능을 확인

- 기본 단일 분류 모델 실험

- 사용 모델: KNN, SVC, Logistic Regression
- 목적 : 기본분류 모델을 바탕으로 데이터 전처리 및 피처 엔지니어링에 따른 성능 비교(핵심 비교 칼럼 age: OneHot vs MinMax Scale)
- 결과 : 가) 정확도 80% 이하로 성능 개선 한계 확인
  나) KNN 에서 과적합 발생(Train – Test 격차 큼
  다) 편향성: 각 모델이 데이터의 특정 패턴에만 강점을 보임
- 결론 : 가장 성능이 안정적인 **Logistic Regression** 을 이후 앙상블 구성에 반영.



knn test accuracy: 0.7540, train accuracy: 0.8137
classification report (Test):

	precision	recall	f1-score	support
0	0.77	0.72	0.75	498
1	0.74	0.78	0.76	502
accuracy			0.75	1000
macro avg	0.75	0.75	0.75	1000
weighted avg	0.75	0.75	0.75	1000

(Train)

	precision	recall	f1-score	support
0	0.82	0.81	0.81	1987
1	0.81	0.82	0.82	2013
accuracy			0.81	4000
macro avg	0.81	0.81	0.81	4000
weighted avg	0.81	0.81	0.81	4000

svc test accuracy: 0.7730, train accuracy: 0.7658
classification report (Test):

	precision	recall	f1-score	support
0	0.77	0.77	0.77	498
1	0.77	0.78	0.77	502
accuracy			0.77	1000
macro avg	0.77	0.77	0.77	1000
weighted avg	0.77	0.77	0.77	1000

(Train)

	precision	recall	f1-score	support
0	0.77	0.75	0.76	1987
1	0.76	0.78	0.77	2013
accuracy			0.77	4000
macro avg	0.77	0.77	0.77	4000
weighted avg	0.77	0.77	0.77	4000

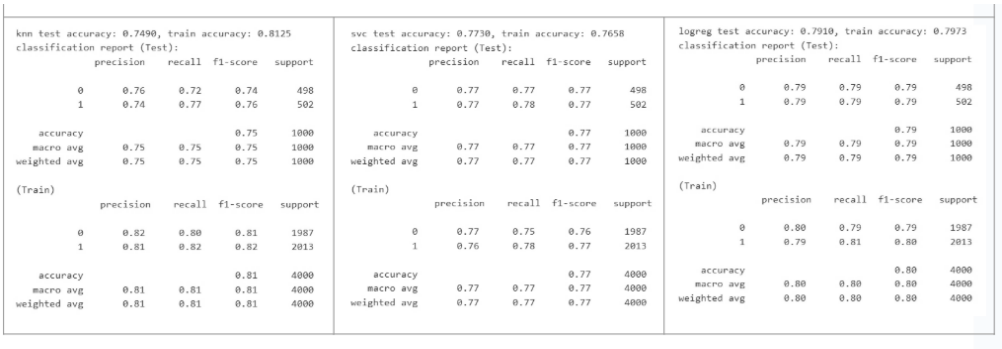
logreg test accuracy: 0.7920, train accuracy: 0.7967
classification report (Test):

	precision	recall	f1-score	support
0	0.79	0.79	0.79	498
1	0.79	0.80	0.79	502
accuracy			0.79	1000
macro avg	0.79	0.79	0.79	1000
weighted avg	0.79	0.79	0.79	1000

(Train)

	precision	recall	f1-score	support
0	0.80	0.79	0.79	1987
1	0.79	0.80	0.80	2013
accuracy			0.80	4000
macro avg	0.80	0.80	0.80	4000
weighted avg	0.80	0.80	0.80	4000

그림 1. 기본 분류 모델(Age One-Hot)



knn test accuracy: 0.7490, train accuracy: 0.8125
classification report (Test):

	precision	recall	f1-score	support
0	0.76	0.72	0.74	498
1	0.74	0.77	0.76	502
accuracy			0.75	1000
macro avg	0.75	0.75	0.75	1000
weighted avg	0.75	0.75	0.75	1000

(Train)

	precision	recall	f1-score	support
0	0.82	0.80	0.81	1987
1	0.81	0.82	0.82	2013
accuracy			0.81	4000
macro avg	0.81	0.81	0.81	4000
weighted avg	0.81	0.81	0.81	4000

svc test accuracy: 0.7730, train accuracy: 0.7658
classification report (Test):

	precision	recall	f1-score	support
0	0.77	0.77	0.77	498
1	0.77	0.78	0.77	502
accuracy			0.77	1000
macro avg	0.77	0.77	0.77	1000
weighted avg	0.77	0.77	0.77	1000

(Train)

	precision	recall	f1-score	support
0	0.77	0.75	0.76	1987
1	0.76	0.78	0.77	2013
accuracy			0.77	4000
macro avg	0.77	0.77	0.77	4000
weighted avg	0.77	0.77	0.77	4000

logreg test accuracy: 0.7910, train accuracy: 0.7973
classification report (Test):

	precision	recall	f1-score	support
0	0.79	0.79	0.79	498
1	0.79	0.79	0.79	502
accuracy			0.79	1000
macro avg	0.79	0.79	0.79	1000
weighted avg	0.79	0.79	0.79	1000

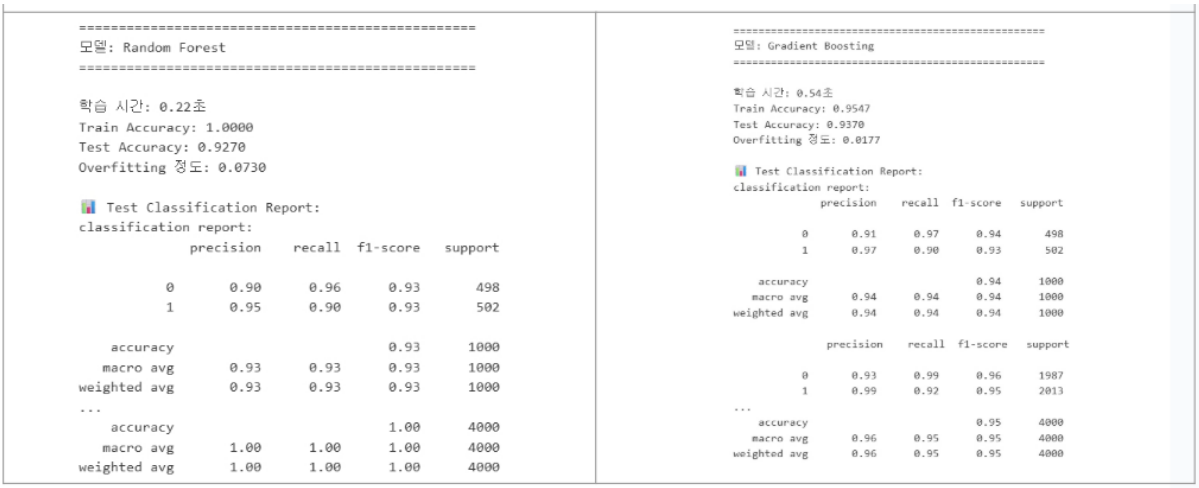
(Train)

	precision	recall	f1-score	support
0	0.80	0.79	0.79	1987
1	0.79	0.81	0.80	2013
accuracy			0.80	4000
macro avg	0.80	0.80	0.80	4000
weighted avg	0.80	0.80	0.80	4000

그림 2. 기본 분류 모델(Age Min-Max Scale)

● 트리 기반 모델

- 사용 모델: Random Forest, Gradient Boosting
- 목적 : 예측의 안정성 확보, 과적합 위험 감소, 변수 중요도 분석 가능
- 결과 : Gradient Boosting 모델은 특정 컬럼 제거가 필요해 정보 손실 발생
  Random Forest 모델이 더욱 적합한 모델로 선정,
  단일 모델 대비 향상되었으나 과적합 문제 발생으로 인해 추가 모델 탐색



==========
모델: Random Forest
==========

학습 시간: 0.22초
Train Accuracy: 1.0000
Test Accuracy: 0.9270
Overfitting 정도: 0.0730

📊 Test Classification Report:
classification report:

	precision	recall	f1-score	support
0	0.90	0.96	0.93	498
1	0.95	0.90	0.93	502
accuracy			0.93	1000
macro avg	0.93	0.93	0.93	1000
weighted avg	0.93	0.93	0.93	1000
...				
accuracy			1.00	4000
macro avg	1.00	1.00	1.00	4000
weighted avg	1.00	1.00	1.00	4000

==========
모델: Gradient Boosting
==========

학습 시간: 0.54초
Train Accuracy: 0.9547
Test Accuracy: 0.9370
Overfitting 정도: 0.0177

📊 Test Classification Report:
classification report:

	precision	recall	f1-score	support
0	0.91	0.97	0.94	498
1	0.97	0.90	0.93	502
accuracy			0.94	1000
macro avg	0.94	0.94	0.94	1000
weighted avg	0.94	0.94	0.94	1000

	precision	recall	f1-score	support
0	0.93	0.99	0.96	1987
1	0.99	0.92	0.95	2013
...				
accuracy			0.95	4000
macro avg	0.96	0.95	0.95	4000
weighted avg	0.96	0.95	0.95	4000

그림 3. 트리 기반 모델

- 앙상블 모델
  - 사용 모델: Bagging, Adaboost, Voting(Hard), Voting(soft)
  - 목적 : 앞선 트리 기반 모델의 문제점을 보안 가능한 새로운 앙상블 모델 탐색
  - 결과 : 1 차 테스트 결과 AdaBoost, Voting(Soft) 후보 선정
    하이퍼 파라미터 튜닝 결과 Voting(soft) 성능 하락
    최종 모델은 AdaBoost 로 결정 (정확도, 안정성, 과적합 제어 모두 우수)

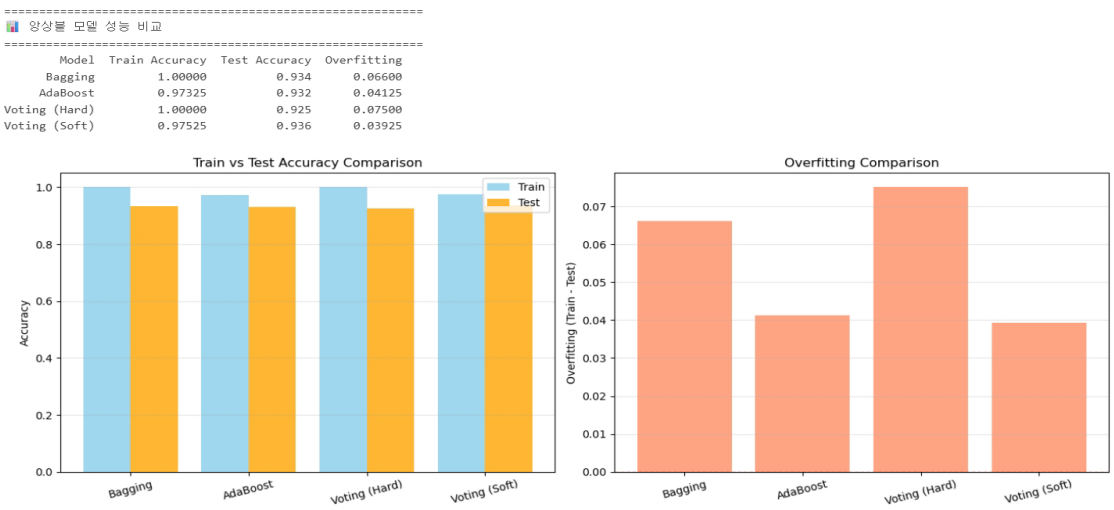


그림 4: 앙상블 1 차 테스트 Baggin, Adaboost, Voting(hard, soft) 비교

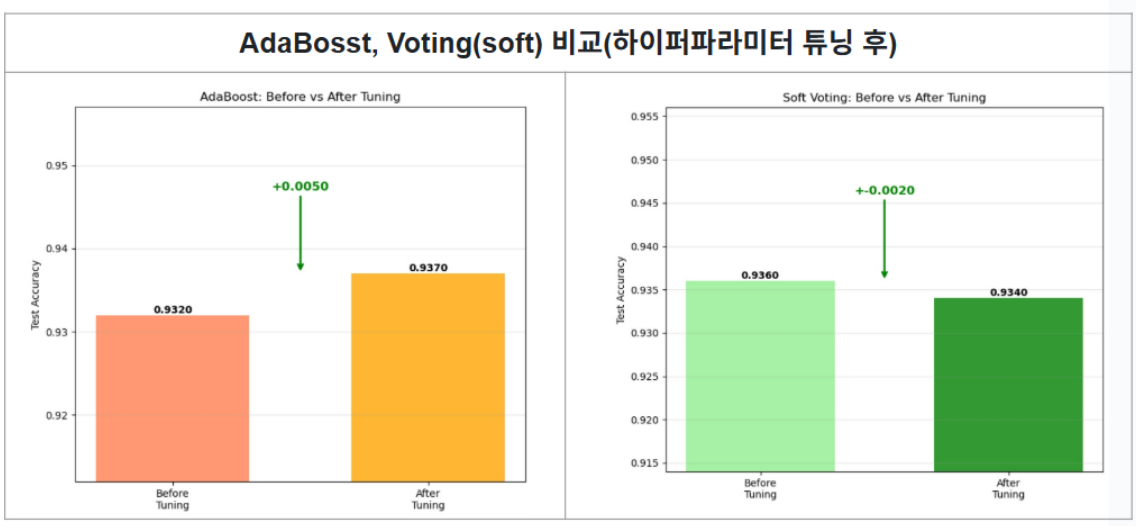


표 1. AdaBoost, Voting(soft) 비교(하이퍼파라미터 튜닝 후)

```
================================================================================
1. AdaBoost Hyper Parameter Tuning
================================================================================
성능 평가:
   Train Accuracy: 0.9487
   Test Accuracy: 0.9370
   Overfitting: 0.0117

 Test Classification Report:
              precision    recall  f1-score   support

 Not Churned       0.91      0.97      0.94       498
     Churned       0.97      0.90      0.93       502

    accuracy                           0.94      1000
   macro avg       0.94      0.94      0.94      1000
weighted avg       0.94      0.94      0.94      1000
```

그림 5. AdaBoost 최종 결과

● CNN 모델 - 데이터 규모 대비 학습 효율성과 성능 향상 폭이 낮아 제외.

```
Test Accuracy: 77.40%

Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.81      0.79       520
           1       0.78      0.74      0.76       480

    accuracy                           0.77      1000
   macro avg       0.77      0.77      0.77      1000
weighted avg       0.77      0.77      0.77      1000

Loss : 0.0971
```

그림 6. CNN 결과

2. 모델 학습 및 튜닝

- 기본 분류 모델을 통해 데이터셋에 맞는 모델의 방향성을 탐색하여 이진 분류에 적합한 모델 선정. 선정된 모델의 성능 최적화를 위한 튜닝
- 튜닝 목표: 모델의 정확도를 극대화하는 동시에 과적합을 방지
- 학습 데이터 구성: Train/Test = 8:2 비율,
  데이터 불균형을 고려하여 Stratified Split 을 적용.
  Train/Test 셋 모두 원본 데이터의 클래스 비율이
  동일하게 유지되도록 만듬.
- GridSearchCV 를 사용한 하이퍼파라미터 튜닝: 5-fold 교차 검증으로 최적의 파라미터 조합을 탐색
- Cross-validation 을 통해 모델 안정성 검증: 오버피팅 문제 해결(수치 0.0117)

```
======================================================================
1. AdaBoost Hyper Parameter Tuning
======================================================================

탐색할 파라미터:
   • n_estimators: [10, 150, 200, 250, 300]
   • learning_rate: [0.05, 0.1, 0.15, 0.2]
   • estimator__max_depth: [2, 3, 4]

   총 조합 개수: 60
AdaBoost GridSearch:   0%|          | 0/1 [00:00<?, ?it/s]
Fitting 5 folds for each of 60 candidates, totalling 300 fits
AdaBoost GridSearch: 100%|██████████| 1/1 [01:15<00:00, 75.40s/it]

 Best Parameters:
   • estimator__max_depth: 4
   • learning_rate: 0.05
   • n_estimators: 250

 Best Cross-Validation Score: 0.9377

성능 평가:
   Train Accuracy: 0.9487
   Test Accuracy: 0.9370
   Overfitting: 0.0117

 Test Classification Report:
              precision    recall  f1-score   support

 Not Churned       0.91      0.97      0.94       498
     Churned       0.97      0.90      0.93       502

    accuracy                           0.94      1000
   macro avg       0.94      0.94      0.94      1000
weighted avg       0.94      0.94      0.94      1000
```

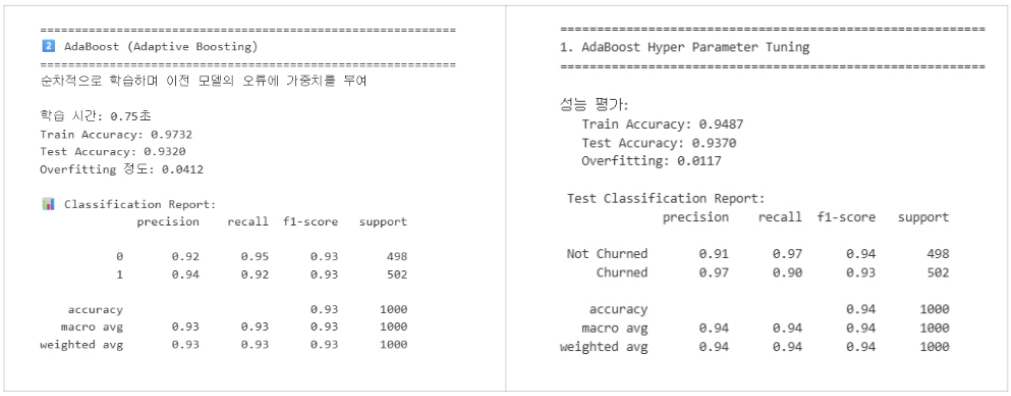
그림 7. AdaBoost 하이퍼 파라미터 튜닝

- AdaBoost(최종모델) 최적의 파라미터
  - max_depth = 4
  - learning_rate = 0.05
  - n_estimators = 250

- 성능 향상

  AdaBoost(튜닝 전) Accuracy = 0.9320, Overfitting = 0.0412

  AdaBoost(튜닝 후) Accuracy = 0.9370, Overfitting = 0.0117



```
============================================================
2️⃣ AdaBoost (Adaptive Boosting)
============================================================
순차적으로 학습하며 이전 모델의 오류에 가중치를 부여

학습 시간: 0.75초
Train Accuracy: 0.9732
Test Accuracy: 0.9320
Overfitting 정도: 0.0412

📊 Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.95      0.93       498
           1       0.94      0.92      0.93       502

    accuracy                           0.93      1000
   macro avg       0.93      0.93      0.93      1000
weighted avg       0.93      0.93      0.93      1000
```

```
============================================================
1. AdaBoost Hyper Parameter Tuning
============================================================

성능 평가:
   Train Accuracy: 0.9487
   Test Accuracy: 0.9370
   Overfitting: 0.0117

 Test Classification Report:
              precision    recall  f1-score   support

 Not Churned       0.91      0.97      0.94       498
     Churned       0.97      0.90      0.93       502

    accuracy                           0.94      1000
   macro avg       0.94      0.94      0.94      1000
weighted avg       0.94      0.94      0.94      1000
```

그림 8. AdaBoost 튜닝 전,후 성능 비교

- 모델 학습 및 튜닝 결론:

  현제 데이터셋에 적합한 튜닝으로는 AdaBoost 모델의 학습률(learning_rate)을 낮추고 트리 깊이(max_depth)를 제한하는 튜닝을 통해, 과적합을 방지하고 최적의 하이퍼 파라미터 찾고 성능 향상 달성.

3. 성능 평가 및 비교

모델	Test Accuracy	Train Accuracy	Train-Test 격차	개선폭
KNN	75.4%	81.8%	6.4%	-
SVC	77.3%	76.6%	0.7%	-
Logistic Regression	79.2%	79.6%	0.4%	-
Random Forest	92.7%	100%	8.3%	+15.4%
Soft Voting Ensemble	93.6%	97.5%	3.9%	+0.9%
AdaBoost (기본)	93.2%	97.3%	4.1%	+0.5%
Tuned AdaBoost (최종)	**93.7%**	**94.9%**	**1.2%**	**+0.5%**

표 1. 모델 성능 및 개선

* 개선폭: 이전 모델의 테스트 정확도 대비 증가 폭

** 랜덤포레스트는 베이스라인과, Soft Voting 과 AdaBoost(기본)은 랜덤포레스트와, 최종은 기본

- 정확도 향상
  - 단일 모델(Logistic Regression) 최고치: 79.2% → 최종 앙상블(AdaBoost) 93.7%
  - 절대 개선: +14.5%
  - 상대 개선(직전모델): +0.5%
- 과적합 제어
  - Random Forest 의 심각한 과적합를 AdaBoost 로 해결
  - 모델의 일반화 능력 대폭 향상
- 안정성 확보
  - StratifiedKFold 교차검증으로 신뢰성 높은 평가
  - 다양한 평가 지표(Precision, Recall, F1-Score)에서 균형잡힌 성능

4. 과정 문제점 및 해결

- AdaBoost 하이퍼 파라미터 최적화
  - 문제: AdaBoost 하이퍼 파라미터 튜닝을 수행하는 과정에서 오히려 accuracy 값이 하락하는 현상이 발생
  - 해결과정:
    ① 과적합 방지를 위해 앙상블 개수(n_estimators)를 늘리고, 학습률(learning_rate)을 낮추어 모델의 일반화 성능을 향상
    ② 데이터의 복잡도가 높지 않아, max_depth 는 기존 설정을 유지
    ③ 가장 중요한 교차 검증 시 시 StratifiedKFold 를 적용해 각 폴드마다 클래스 비율이 일정하도록 설정
    ④ 5-Fold 교차검증(n_splits=5)과 데이터 셔플(shuffle=True)을 사용하여, 최적의 하이퍼파라미터 조합을 탐색한 결과 accuracy 가 향상된 최적 모델을 도출함.

5. 최종 결론

- 최종 선정된 **AdaBoost** 모델은 성능(Accuracy 93.7%), 과적합 제어, 안정성 측면에서 가장 우수한 결과를 보임. 해당 모델은 고객 이탈 조기 탐지, 개인화 추천, 마케팅 타깃 최적화 등 실제 서비스 적용 가능성이 높음.