AI #1 HPO 과제

목차

▼ 1. 실습 내용

> [HPO Method]
> 1. Grid Search
> 2. Random Search
> 3. Bayesian Search
> 4. Bayesian Search(TPE:Tree-structured Parzen Estimator) using Optuna
>
> [Data]
> Bank Marketing Data set
> https://archive.ics.uci.edu/dataset/222/bank+marketing
>
> [Note]
> HPO Example로 오류방지를 위한 최소한의 전처리만 수행
>
> [Task]
> 위 예시 사례를 활용하고 적용모델을 달리하여 하이퍼 파라메타 최적화(HPO)를 수행해보세요
> - HPO Method 중 2개의 Method를 선정하여 비교분석 해보세요.
> - 적용 모델을 다르게.. (XGBoost 제외)

▼ 2. 개요

- 목적
  - UCI Bank Marketing 데이터를 로드하고 전처리 후 LightGBM 모델을 구축 및 평가
  - 교차검증(`StratifiedKFold`, `KFold`)과 하이퍼파라미터 탐색 기법(`RandomizedSearchCV`, `Optuna`)을 비교하여 최적 성능을 도출
- 구현
  - 데이터 로드: `ucimlrepo` 라이브러리로 UCI 데이터셋을 가져와 `pandas` DataFrame으로 통합
  - 데이터 전처리: `LabelEncoder`로 범주형 인코딩, 타겟(y)을 이진화(`yes/no → 1/0`), `StandardScaler`로 스케일링 수행
  - 모델 구축: `lightgbm.LGBMClassifier`로 학습
  - 모델 평가: 정확도(`accuracy_score`), AUC(`roc_auc_score`), 분류리포트(`classification_report`), 혼동행렬(`confusion_matrix`)을 통해 성능 평가, `feature_importances_`로 중요한 변수 파악
  - 교차검증: `KFold`, `StratifiedKFold` 기반 `cross_val_score`로 모델 안정성 비교
  - 하이퍼파라미터 최적화
    - Random Search: `RandomizedSearchCV`로 지정된 하이퍼파라미터 후보군에서 랜덤 샘플링 기반 탐색 수행
    - Optuna 최적화: `optuna`의 `TPESampler`로 탐색 공간을 효율적으로 최적화하고 AUC 기준 최적 파라미터 도출

▼ 3. 결과

1. 기본 LightGBM

```
=== LightGBM (Default) ===
정확도 (Accuracy): 0.9104
AUC 점수: 0.9334

분류 리포트:
precision    recall  f1-score   support

0       0.93      0.97      0.95      7985
1       0.66      0.49      0.56      1058

accuracy                           0.91      9043
macro avg       0.80      0.73      0.75      9043
weighted avg       0.90      0.91      0.90      9043

혼동 행렬:
[[7719  266]
[ 544  514]]

교차검증 결과:
KFold (5) AUC: 0.9349 (±0.0058)
StratifiedKFold (5) AUC: 0.9341 (±0.0068)

상위 10개 중요한 변수:
        feature  importance
11     duration         507
10        month         498
9   day_of_week         368
5       balance         317
0           age         316
13        pdays         256
8       contact         125
12     campaign         115
1           job         105
15     poutcome          88
```

- 정확도 (0.9104)

  전체적으로 91%를 맞혔다는 의미이지만, 데이터의 불균형(클래스 0: 7985 vs 클래스 1: 1058)을 고려하면 "정상 고객만 맞히고 끝"이라도 높은 정확도가 나올 수 있다. 따라서 이 정확도는 모델이 실제로 소수 클래스(마케팅 캠페인에 응답한 고객)를 잘 분류했다고 보장하지 않는다.

- AUC (0.9334)

  ROC-AUC가 높게 나온 것은 임계치(threshold)를 다양하게 바꾸어도 양성과 음성을 잘 구분하는 경향이 있다는 뜻이다. 즉, threshold를 0.5로 고정했을 때는 recall이 낮지만, threshold를 낮춰주면 더 많은 양성을 검출할 수 있다는 가능성을 보여준다. AUC는 클래스 불균형 상황에서도 비교적 안정적이다.

- 분류 리포트
  - 클래스 0: precision 0.93, recall 0.97 → 정상 고객을 맞히는 능력은 탁월하다.
  - 클래스 1: precision 0.66, recall 0.49 → positive 클래스의 절반 이상을 놓치고 있으며, 맞힌 경우에도 34% 정도는 false positive다. 즉 recall이 낮아 "실제 캠페인에 응답한 고객"을 많이 놓치고 있다.
  - macro f1-score가 0.75로 떨어진 이유가 바로 이 클래스 불균형 때문이다.

- 혼동 행렬

```
[[7719  266]
 [ 544  514]]
```

- TN=7719, FP=266, FN=544, TP=514.
- FP보다 FN이 많다는 건, 실제 양성 고객을 놓치는 경우가 더 많다는 뜻이다.
- 마케팅에서는 recall이 중요한 경우가 많다. 응답 고객을 놓치면 매출 기회를 잃기 때문이다.

2. HPO - Random Search

```
=== LightGBM (Random Search) ===

정확도 (Accuracy): 0.9092
AUC 점수: 0.9333

분류 리포트:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      7985
           1       0.65      0.48      0.56      1058

    accuracy                           0.91      9043
   macro avg       0.79      0.72      0.75      9043
weighted avg       0.90      0.91      0.90      9043

혼동 행렬:
[[7710  275]
 [ 546  512]]

교차검증 결과:
KFold (5) AUC: 0.9352 (±0.0060)
StratifiedKFold (5) AUC: 0.9356 (±0.0065)

상위 10개 중요한 변수:
        feature  importance
11     duration        1317
10        month        1242
9   day_of_week        1127
0           age        1099
5       balance        1085
13        pdays         805
1           job         404
12     campaign         391
8       contact         292
15     poutcome         247
```

- 정확도 (0.9092)

  기본 모델과 거의 비슷하다. 약간의 튜닝에도 불구하고 정확도 자체는 큰 변화가 없다. 이는 정확도가 불균형 데이터에서 한계가 있음을 다시 보여준다.

- AUC (0.9333)

  KFold 0.9352, StratifiedKFold 0.9356으로 기본 모델보다 아주 약간 상승했다. threshold를 조정하면서 분류할 때는 기본보다 조금 더 안정적으로 성능을 낼 가능성이 있다.

- 분류 리포트

- 클래스 0: 여전히 precision과 recall이 0.93, 0.97로 매우 높다.
- 클래스 1: precision 0.65, recall 0.48로 기본과 사실상 동일하다. 즉 소수 클래스의 검출 능력 개선은 거의 없었다.
- 이 결과는 Random Search가 탐색한 하이퍼파라미터 범위에서 모델의 capacity가 클래스 1 recall 개선 쪽으로 최적화되지 않았음을 시사한다.

- 혼동 행렬

```
[[7710  275]
 [ 546  512]]
```

- FN=546, TP=512 → Recall이 낮은 원인은 여전히 많은 FN 때문이다.
- FP=275로 늘어난 것도 특징인데, precision도 약간 떨어지는 원인이 된다.
- 따라서 Random Search로 얻은 모델은 recall을 개선하지 못했고, precision과 recall의 균형도 기본과 비슷하다.

3. HPO - Optuna

```
=== LightGBM (Optuna) ===
정확도 (Accuracy): 0.9095
AUC 점수: 0.9330

분류 리포트:
              precision    recall  f1-score   support

           0       0.93      0.97      0.95      7985
           1       0.66      0.47      0.55      1058

    accuracy                           0.91      9043
   macro avg       0.80      0.72      0.75      9043
weighted avg       0.90      0.91      0.90      9043

혼동 행렬:
[[7732  253]
 [ 565  493]]

교차검증 결과:
KFold (5) AUC: 0.9349 (±0.0056)
StratifiedKFold (5) AUC: 0.9348 (±0.0066)

상위 10개 중요한 변수:
        feature  importance
10        month        1088
11     duration         814
9   day_of_week         773
5       balance         552
0           age         543
13        pdays         518
8       contact         278
12     campaign         221
6       housing         186
15     poutcome         185
```

- 정확도 (0.9095)

Random Search와 거의 동일하다. 정확도가 유지된다는 건 클래스 불균형 상황에서 정확도 지표만 보는 것이 무의미하다는 점을 다시 보여준다.

- AUC (0.9330)

  KFold 0.9349, StratifiedKFold 0.9348 → 세 모델 중 가장 근소하게 낮지만 사실상 동일한 수준. Optuna가 효율적으로 탐색했지만, 이 데이터셋과 LightGBM에서는 큰 개선을 만들지 못했다.

- 분류 리포트
  - 클래스 1 recall이 0.47로 Random Search(0.48)보다 조금 더 낮아졌다. 즉, 양성을 놓치는 비율이 더 커졌다.
  - precision은 0.66으로 유지되어 "양성이라고 한 경우 2/3는 맞혔다" 수준을 유지.
  - recall이 낮아졌다는 건 Optuna가 찾은 파라미터 조합이 더 보수적으로 분류한 경향이 있다는 뜻이다.
- 혼동 행렬

```
[[7732  253]
 [ 565  493]]
```

  - FN=565, TP=493 → 놓친 양성이 더 많다.
  - FP=253 → 불필요하게 양성이라고 찍은 정상 고객은 줄었지만, 그만큼 recall을 희생한 구조다.
  - 즉 Optuna 모델은 "맞히는 건 좀 더 신중히 하자" 쪽으로 최적화되었다고 볼 수 있다. precision은 유지하면서 recall은 떨어졌다는 점이 이를 보여준다.

▼ 4. 3 case 결과 비교 및 해석

- 정확도
  - 3 case 모두 ~91%로 거의 동일하게 나왔다.
    - 최적화에 따른 정확도 성능 향상이 없었는데 데이터 불균형 상황에선 정확도가 한계적인 지표이기 때문일것으로 생각된다.
- AUC는 세 모델 모두 ~0.933으로 높다.
  - threshold 조정을 통한 잠재적 구분 능력은 충분하고 생각된다.
- 분류 리포트와 혼동 행렬을 보면 recall이 공통적으로 낮다.
  - 기본: recall 0.49
  - Random Search: recall 0.48
  - Optuna: recall 0.47
- 즉, 세 모델 모두 positive 클래스(캠페인 응답 고객)를 제대로 잡아내지 못한다.
- Random Search vs Optuna의 차이는 precision-recall 균형 방식이다.
  - Random Search는 recall과 precision 모두 기본 모델 수준에서 유지.
  - Optuna는 FP를 줄이는 대신 FN이 늘어 recall을 더 낮췄다.
- 비즈니스 관점에서는 recall(coverage)이 더 중요하다면 Optuna보다는 Random Search 또는 threshold 조정이 유리하다. 반대로 false positive를 줄이는 게 중요하다면 Optuna 결과를 선호할 수도 있다.