**모델 하이퍼파라미터 튜닝**

**n\_estimators = default = 100 인데 300, 500 시도**

**learning rate가 점점 떨어지도록**

**0.1 -> 0.05 ->, 0.04,-> .0.03**

**boosting :**

**보통 GBDT (default)를 쓰고,**

**DART : tree dropout 적용 (수행시간 긴편)**

**[트리]**

**-Max\_depth : 5-12 (depth 조절 시 많이 바뀌므로 하이퍼파라미터 튜닝이 필요)**

**-min child\_sample은 크게 변화가 없음.**

**-num\_leaves(d=32~64)로 조절**

**[샘플링]**

* **sub\_sample : 0.5 -1 정도로 세팅**
* **Colsample\_bytree : 0.5 -1 정도로 세팅**

**[규제]**

**크게 효과를 못봄.**

**-grid search**

**-randomsearch**

**-베이지안 최적화**

**감을 잡고 그리드 서치 필요. cost 소요.**

**아나콘다 베이지안 옵티마이제이션 필요.**

**하이퍼파라미터 튜닝**

**조건 :**

**오버샘플링안하고,**

**Defaulf 값으로 xgboost,ada,rf 돌렸을때.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모델 | 정확도 | 정밀도 | 재현율 | F1점수 |
| XGB | 0.741 | 0.745 | 0.741 | 0.697 |
| ADABoost | 0.731 | 0.73 | 0.731 | 0.68 |
| RF | 0.734 | 0.729 | 0.735 | 0.69 |

**중요변수들로만 돌렸을때,**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모델 | 정확도 | 정밀도 | 재현율 | F1점수 |
| xgb | 0.69 | 0.66 | 0.69 | 0.617 |

**Lightgbm 돌렸을 때,**

정확도: 0.743023020159031

정밀도: 0.7491543919788156

재현율: 0.743023020159031

F1 점수: 0.6991004791457245

Minmax scale, default 값 : 모델 성능들 (standard scale 적용 칼럼 다시 확인)

1. xgb

정확도: 0.6934420561973225

정밀도: 0.6650198122605

재현율: 0.6934420561973225

F1 점수: 0.6181663707324057

Standard scale, default 값 : 모델 성능들 (Standard scale 적용 칼럼 다시 확인)

1. xgb

정확도: 0.7420812321924586

정밀도: 0.7469991624142157

재현율: 0.7420812321924586

F1 점수: 0.6983164569103107

그냥 모델만 하이퍼파라미터 튜닝. 2% 증가 추세를 보였음.

# 탐색할 하이퍼파라미터 범위 설정

param\_dist = {

'num\_leaves': range(32, 64),

'max\_depth': range(5, 12),

'learning\_rate': [0.1, 0.05, 0.04, 0.03],

'n\_estimators': [100, 300, 500],

'boosting\_type': ['gbdt', 'goss']

# 다른 하이퍼파라미터도 추가할 수 있습니다.

}

# RandomizedSearchCV를 사용하여 최적의 하이퍼파라미터 찾기

random\_search = RandomizedSearchCV(

estimator=lgb\_model,

param\_distributions=param\_dist,

n\_iter=100, # 시도할 조합의 횟수

cv=5, # 교차 검증 폴드 수

n\_jobs=-1

)

Default 모델값.

1. Lgb 74 -> 76%

(스케일링 없이) 오버샘플링 후, (하이퍼 파라미터 조정시 1-2% 증가)

1. Lightgbm : 76 -> 81-82%
2. Xgboost : 74,76 -> 80-81% 비슷하게 나왔음.
3. Adaboost 69% 로 많이 감소
4. 랜덤포레스트 78%

1 . xgboost

Smote + 원핫인코딩, Standardscale => 모델 성능

모델 정확도: 0.7406625895592671

그냥 데이터 스케일링 없이, 오버샘플링했을때,

정확도 80%

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

결론 :

단순히 오버샘플링 시 모델결과가 가장 좋았음.

스케일링시 오히려 모델성능이 떨어졌음.

최적의 변수 조합 및 하이퍼 파라미터 시도가 필요함