

2023 ML Competiton

분석보고서

AI빅데이터융합경영학과

20202662 조현식

20202656 이정현

20212563 우서연

Feature 생성

```
def assign_group(title):  
    if 'DR' in title and title.endswith('L'):  
        return 1  
    elif 'DR' in title and title.endswith('S'):  
        return 2  
    elif 'G' in title:  
        return 3  
    elif 'DN' in title:  
        return 4  
    elif 'IM' in title:  
        return 5  
    elif 'BL' in title:  
        return 6  
    elif '소비자' in title or '소비사' in title:  
        return 7  
    elif '주류' in title:  
        return 8  
    elif '일반국민' in title:  
        return 9  
    elif '국민' in title:  
        return 10  
    elif '광고' in title:  
        return 11  
    elif '서비스' in title:  
        return 12  
    elif '외국어' in title:  
        return 13  
    elif '해외' not in title and '일반인' in title:  
        return 14  
    else:  
        return 99
```

TITLE 피처를 다른 방식으로 분류하여 TITLE2 피처 생성



이를 다시 세부 분류하여 TITLE1 생성,
TITLE1별 응답률 피처 만든 후 TITLE1 삭제



기존 피처를 다르게 분류하고 사용함으로써
Feature 활용도 높임

그 밖에도 설문별 응답률 같은 새로운 피처 생성으로
성능 향상 도모

Silhouette 계수 활용

KMeansFeaturizer = 데이터 포인트 기준으로 k개 군집을 형성

```
kmf_hint = KMeansFeaturizer(k=10, target_scale=5, one_hot=False, random_state=0).fit(X_train)

train_cluster_features = kmf_hint.transform(X_train)
test_cluster_features = kmf_hint.transform(X_test)

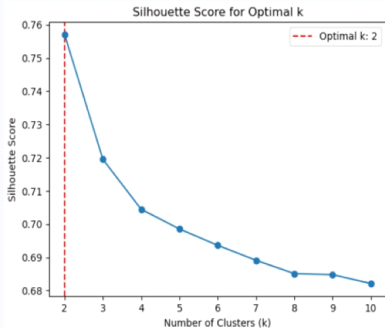
X_train = pd.concat([X_train, pd.Series(train_cluster_features, name='CLUSTER')], axis=1)
X_test = pd.concat([X_test, pd.Series(test_cluster_features, name='CLUSTER')], axis=1)
```



feature 각각을 군집화 할 수 있을까?

=> Silhouette 계수 활용하여 최적의 군집 수를 찾는다

Silhouette 계수 활용



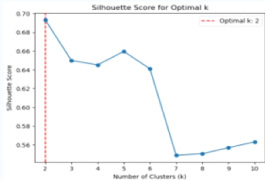
userID 최적의 군집 계수 = 2



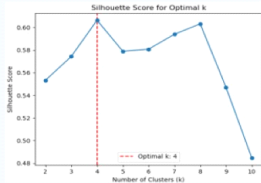
	userID	cluster_2
0	p00000	1
1	p00001	1
2	p00002	1
3	p00003	0
4	p00004	1
...
15150	p16046	0
15151	p16047	1
15152	p16048	0
15153	p16049	1
15154	p16050	1

userID를 2개로 군집화(0,1)

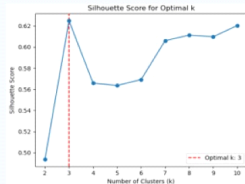
Silhouette 계수 활용



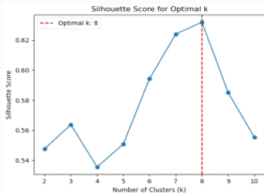
surveyID 2개



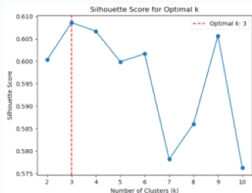
REGION 4개



IR 3개

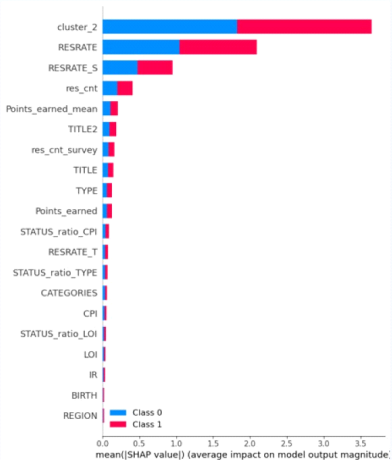


LOI 8개



CPI 3개

군집화 결과



SHAP를 활용한 Feature Selection

cluster_2 (userID) importance = 1.822246

=> userID 군집화가 성능향상에 기여

OOF [Kfold =3]

LGBM (threshold = 0.495) => 5-fold 평균 Accuracy는 0.867762551465618

CatBoost (threshold = 0.495) => 5-fold 평균 Accuracy는 0.8678006513857308

*Boost 계열 모델 중 LGBM, CatBoost 성능이 제일 좋게 나옴

But, OOF에만 의지하기에는 성능이 불안전



lgbm_submission_12192226.csv

Complete · hyeonsik827 · 1d ago

0.85997

0.8605



(Public: 0.8605 Private: 0.85997)

=> 다른 계열 모델인 DNN과 앙상블하여 성능의 안정성을 높이하고자 함


DNN

* DNN의 평균적인 성능을 얻고자 seed ensemble을 함 (DNN_MODELS = 20)

```
RANDOM SEEDS RESET: 1549
validation accuracy = 0.9015793204307556
16934/16934 [=====] - 16s 917us/step
RANDOM SEEDS RESET: 8947
validation accuracy = 0.9015547037124634
16934/16934 [=====] - 16s 930us/step
RANDOM SEEDS RESET: 3521
validation accuracy = 0.9015485644340515
16934/16934 [=====] - 16s 949us/step
RANDOM SEEDS RESET: 9715
validation accuracy = 0.9017022252082825
16934/16934 [=====] - 17s 980us/step
RANDOM SEEDS RESET: 96
validation accuracy = 0.9029005169868469
16934/16934 [=====] - 16s 922us/step
RANDOM SEEDS RESET: 9639
validation accuracy = 0.9012535810470581
16934/16934 [=====] - 17s 997us/step
RANDOM SEEDS RESET: 4686
validation accuracy = 0.8926565647125244
16934/16934 [=====] - 17s 1ms/step
RANDOM SEEDS RESET: 4944
validation accuracy = 0.900073766708374
16934/16934 [=====] - 51s 3ms/step
RANDOM SEEDS RESET: 513
...
16934/16934 [=====] - 22s 1ms/step
RANDOM SEEDS RESET: 3674
validation accuracy = 0.9005223512649536
16934/16934 [=====] - 17s 982us/step
```

validation accuracy
0.9 극 초반대로 매우 좋은 성능을 보임

But


 dnn_submission_1220_1600.csv 0.85147 0.85134 
Complete · hycroak827 · 8h ago

public score: 0.85134


과적합 현상이 발생

Ensemble

모델 성능 및 안정성을 높이고자 **lgbm, catboost, DNN** 모델

 cat_submission_12201615	2023-12-20 오후 4:15	한컴오피스 한셀 ...
 dnn_submission_1220_1600	2023-12-20 오후 4:00	한컴오피스 한셀 ...
 lgbm_submission_12201615	2023-12-20 오후 4:15	한컴오피스 한셀 ...

앙상블 총 3개의 모델 submission을 가지는 **Power Mean 앙상블**

 p1mean_submission_1220_1620

* 결과 (public: 0.86032, private:0.86009)



p1mean_submission_1220_1620.csv

Complete · hyeonsik827 · 9h ago

0.86009

0.86032



배운 점

여러 종류의 모델을 앙상블 하면 일반적으로 성능이 올라가는 것이 맞지만 성능 차이가 많이 난다면 오히려 여러 모델을 합치는 것이 해가 될 수 있다.
자신만의 판단 기준을 잘 세워서 결정해야 한다.

모델 튜닝을 통해 best parameter를 찾아내어 단일모델의 성능이 잘 나오더라도 그 값이 과적합이라면 k-fold(oof)에서 허점이 그대로 드러난다.

feature 양보다 질이 중요한 것 같다.

모델 성능을 높이는데 있어 EDA가 큰 비중을 차지하고 있다고 느꼈다.

이론적으로 맞다고 생각되는 것과 실제 결과는 많이 다르다.