[1반] 이재원-박원우-김상욱 소스코드 실행절차

[1반] 이재원-박원우-김상욱 (이재원우욱조)

<순서>

1. Data Preprocesssing & 파생변수 생성

2. Feature Set Setting

3. Feature Engineering

4. Feature Selection

4-a Feature Importance

5. Modeling

5-a. CatBoost, XGBM, LGBM + optuna + oof

6. Voting

Data Preprocessing에서 기존에 제공된 베이스라인 코드에서 STATUS를 활용하여 feature를 만드는 응답률, 응답횟수와 관련된 feature를 생성하였다. 예를 들어, userID 별 응답률( 유저아이디별 응답횟수 / 유저아이디별 전송횟수), surveyID별 응답률 (설문조사아이디별 응답횟수/ 설문조사아이디별 전송횟수) , 응답한 것들 중 userID 별 얻은 포인트 합 등등이 있다.

STATUS에 영향을 미치지 않게 하기 위해서 기존 train에 존재하는 수치형과 범주형을 조합하여 feature를 생성하였다. 수치형 + 수치형의 조합으로 만든 feature 예를 들어 IR/LOI, IR\*CPI, IR\*LOI 와 같은 변수를 만들고 transform\_columns 사용자 함수를 통해 제곱, 세제곱, 제곱근, 로그 등의 수치형 변수들에 대해서 조합을 만들었다. 또한 Birth와 같은 탄생연도를 AGE\_GROUP이라는 열을 생성하여 시니어 세대부터 Z세대까지 만들어주는 사용자함수를 통해 범주형으로 만들어 주기도 하였다. 수치형 변수에 대해 범주형 또는 수치형으로 구간화를 진행하였다.

위의 조합을 통해 만들어진 수치형 또는 범주형을 groupby를 통해 평균값 또는 합을 나타내는 열을 생성하거나 새롭게 만들어진 열에 수치형을 곱해줌으로 새로운 파생변수를 만들었다. 또한 범주형 + 범주형의 조합을 생성하기 위해 .apply(“\_”.join, axis = 1)을 통해 범주형과 범주형 feature를 ‘\_’ 기준으로 합치는 것으로 방법을 사용하였다 (TYPE\_TITLE과 같은 열). 또한 IR\_range와 같은 것은 수치로 구간화를 진행해주었는데 이를 apply 함수를 실행시키게 되면 오류를 발생하므로 .astype(str)을 통해 str값으로 변환한 뒤 활용하고 다시 drop하는 방식으로 진행하였다. 그리고 반대로 AGE\_GROUP과 같은 것은 다시 수치형으로 변환한 뒤에 다양한 수치형 파생변수 또는 기존의 변수들을 곱하면서 새로운 피쳐를 만드는 것에 초점을 맞추었다. 또한 train에 pannel과 survey에 대한 변수들 중에 주요한 것들만 대해 각 행별 유사도에 대한 열을 생성하기 위해 KMeansFeaturizer를 활용하여 CLUSTER Feature를 생성하였다. Clustering을 진행할 때는 수치형에 대해서는 StandardScaler를 사용하고, 범주형에 대해서는 get\_dummies를 활용하여 OneHotEncoder를 진행하였다. 이후 생성된 열을 merge하고 클러스터링을 통해 만든 변수들 별로 응답률에 해당하는 feature를 생성하기도 하였다.

Feature Set Setting에서는 target에 해당하는 STATUS를 제거하고 범주형, 수치형으로 변수를 분리하여 정의한다

Feature Engineering에서는 기존의 교수님의 베이스 코드를 활용하여 simpleimputer를 통해 mean과 most\_frequent와 같은 값으로 대체하고, 범주형에 대해서는 OrdinalEncoder를, 수치형에 대해서는 StandardScaler를 활용하였다.

Feature Selection에서는 train\_and\_evaluate라는 사용자 함수를 통해 개별 모델별로 Feature\_importance를 구하고 시각화하는 그래프를 생성하고 직접 값을 하나씩 넣어보며 다수의 실험을 통해 최적의 threshold를 찾고 각 모델별로 각각의 최적의 threshold를 적용하여 low\_importance\_feature에 대한 것은 drop하고 데이터를 실행하였다.

Modeling에서는 Cat, XGB, LGBM, RF를 사용하였는데 개별 모델에 대해 xgb\_importance\_feature, cat\_importance\_feature 등에 대해 다양한 학습을 진행해 본 결과 Cat모델에 대해서는 cat\_importance\_feature보다 xgb\_importance\_feature이 성능을 올리는 것에 있어서 더 좋다는 결과를 얻어 cat, xgb에는 xgb에 대한 importanc\_feature 나머지 2개는 각각에 대한 것을 진행하였다. 이후 objective 사용자함수를 통해 하이퍼파라미터 값 튜닝 구간을 지정하였다. 그리고 나온 best\_param에 대해 모델을 학습 시키고 oof\_prediction을 생성하였다. 그 결과 성능이 좋은 순서는 cat, lgbm, xgb, rf임을 알게 되었고, voting과정에서 성능이 좋은 순서대로 가중치를 0.4, 0.3, 0.2, 0.1 값을 주고 임계치에도 교수님이 tip으로 제공해준 threshold를 기준으로 0.495값을 주고 final\_predicitons를 정의해준다

이후 pd.Dataframe을 통해 final\_7.csv라는 submission 파일을 생성하고 생성된 csv를 제출한다