재구매 확률 예측 모델링 보고서

응용통계학과

AI를 위한 머신러닝

1조

김동현, 김성연, 송민주, 이정은

목차

1.Introduction

1.1 Project overview

현재 AI 기술의 발전과 함께 기업들은 다양한 분야에서 AI 모델의 적용을 모색하고 있다. 이러한 흐름에 따라 학습 모델을 적용할 분야를 모색한 결과 재구매 고객 예측을 중요한 영역으로 선정하였다. 현대 사회의 전자 상거래 분야에서는 다양한 프로모션을 활용하여 일회성 거래 소비자를 충성 고객으로 유도하려는 노력을 하고 있다. 소비자들은 유튜브 알고리즘, 맞춤형 광고 등의 맞춤형 서비스에 익숙하다. 이에 따라 도/소매 분야에 학습 모델을 적용하여 고객이 구매할 물건, 브랜드를 예측하여 제공한다면 물건 탐색의 시간이 줄어들고 기업의 입장에서 이에 대비하여 마케팅 전략을 효과적으로 세울 수 있을 것이다. 본 프로젝트는 전자상거래 기업과 소비자의 경향을 기반으로 학습 모델을 적용하여 고객 행동을 예측하고, 이를 활용하여 향후 더욱 효과적인 마케팅 전략을 수립하고자 하는 목표를 설정하였다

전자 상거래 상인들은 대규모 프로모션(할인 또는 캐시 쿠폰 등)을 주로 "블랙 프라이데이 기간"이나 "Double 11 (11월 11일)"과 같은 특정 날짜에 진행하여 많은 신규 구매자를 유치한다. 하지만 단기성 프로모션의 경우 구매자 중 다수는 일회성 거래로 끝나게 되어 장기적인 매출에 큰 영향을 미치지 못하고 제한된 영향을 가진다는 한계를 갖는다. 본 프로젝트는 이에 대한 해결책으로 대회에서 주어진 상인과 소비자 정보 및 로그 데이터를 활용하여 미래에 충성 고객으로 전환될 가능성이 있는 구매자를 예측하는 것을 목표로 하였다.

1.2 Problem definition

본 프로젝트의 궁극적인 목표는 특정 기간(6개월) 내 특정 판매자의 신규 구매자들에 대해 반복 구매가 발생할 확률을 예측하는 것이다. 이를 통해 기업은 미래 매출을 더 정확하게 예측하고, 타겟 마케팅을 통해 프로모션 비용을 최적화하여 고객 이탈을 최소화함과 동시에 장기적인 매출 향상을 이룰 수 있는 전략을 수립할 수 있을 것이다.

1.3 Data Introduction

본 프로젝트는 Tmall.com의 'double11 day' 당일 프로모션 기간 동안 획득한 가맹점 세트와 해당 신규 구매자를 활용하여 분석 및 예측 모델링을 진행한다. 유저 정보 데이터, 유저 로그 데이터, 학습 데이터와 테스트 데이터, 이렇게 총 4개의 파일을 사용했으며 데이터의 형태는 다음과 같다.

1) 유저 정보 데이터

유저 정보 데이터에는 424170명의 유저 id와 유저의 인구통계학적 정보인 나이, 성별이 들어가 있다. gender는 여자는 0, 남자는 1로 분류된 더미변수이고, age는 나이대별로 1~8의 숫자로 분류되어 있다.

Data Fields	Definition
user_id	A unique id for the shopper.
age_range	User' s age range: 1 for <18; 2 for [18,24]; 3 for [25,29]; 4 for [30,34]; 5 for [35,39]; 6 for [40,49]; 7 and 8 for >= 50;0 and NULL for unknown.
gender	User' s gender: 0 for female, 1 for male, 2 and NULL for unknown.

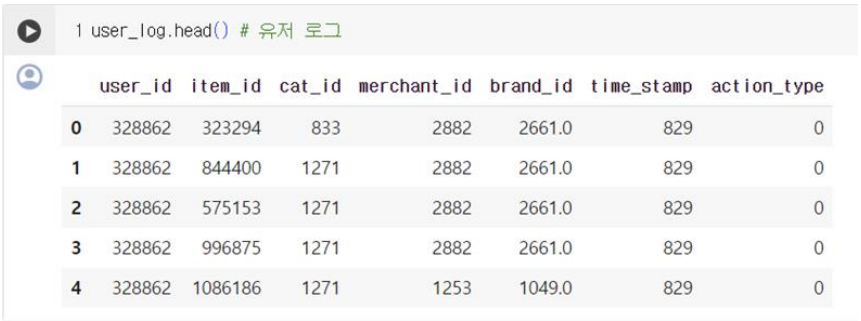
```
1 user_info.head() # 유저 정보
```

	user_id	age_range	gender
0	376517	6.0	1.0
1	234512	5.0	0.0
2	344532	5.0	0.0
3	186135	5.0	0.0
4	30230	5.0	0.0

2) 유저 로그 데이터 (User Behaviour Logs)

유저 로그 데이터에는 54925330개로 이루어져 있으며 유저 고유 id와 유저가 구매한 제품과 제품의 카테고리, 제품을 파는 판매자, 브랜드의 고유 id 정보가 들어가 있다. 유저가 제품을 구매한 날짜와 유저가 취한 행동의 유형이 분류되어 있다. 0은 click, 1은 장바구니 담기, 2는 구매, 3은 찜하기를 나타낸다.

Data Fields	Definition
user_id	A unique id for the shopper.
item_id	A unique id for the item.
cat_id	A unique id for the category that the item belongs to.
merchant_id	A unique id for the merchant.
brand_id	A unique id for the brand of the item.
time_stamp	Date the action took place (format: mmdd)
action_type	It is an enumerated type {0, 1, 2, 3}, where 0 is for click, 1 is for add-to-cart, 2 is for purchase and 3 is for add-to-favourite.



```
1 user_log.head() # 유저 로그
```

	user_id	item_id	cat_id	merchant_id	brand_id	time_stamp	action_type
0	328862	323294	833	2882	2661.0	829	0
1	328862	844400	1271	2882	2661.0	829	0
2	328862	575153	1271	2882	2661.0	829	0
3	328862	996875	1271	2882	2661.0	829	0
4	328862	1086186	1271	1253	1049.0	829	0

3) 학습 데이터와 테스트 데이터

학습 데이터와 테스트 데이터는 동일한 형식이다. 유저 고유 id와 판매자의 고유 id, 그리고 구매자가 반복 구매자인지 아닌지에 대한 정보를 담인 label 변수가 들어 있다.

label의 값이 1이면 반복 구매한 사람, 0이면 반복 구매하지 않은 사람이다. 260864개의 학습데이터를 이용해서 261477개의 테스트 데이터를 예측할 예정이다.

Data Fields	Definition
user_id	A unique id for the shopper.
merchant_id	A unique id for the merchant.
label	It is an enumerated type {0, 1}, where 1 means repeat buyer, 0 is for non-repeat buyer. This field is empty for test data.

```
1 train_data.head() # training data
```

	user_id	merchant_id	label
0	34176	3906	0
1	34176	121	0
2	34176	4356	1
3	34176	2217	0
4	230784	4818	0

4) 제출 형태

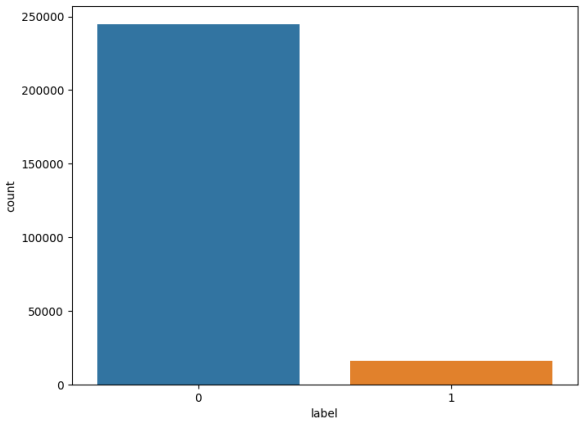
본 프로젝트는 단순히 재구매를 예측하는 classification 문제에서 좀 더 고도화되어 확률값을 예측하는 모델링을 진행한다. 이에 따라 제출 형태는 다음과 같이 유저 id와 판매자 id를 기준으로 재구매할 확률을 예측하는 형태이다.

Data Fields	Definition
user_id	A unique id for the shopper.
merchant_id	A unique id for the merchant.
prob	Predicted probability of the given user becoming a repeat buyer of the given merchant. Value should be between 0 and 1.

2. Method

2.1 데이터 전처리

데이터 전처리 과정에서 결측치와 레이블 데이터의 불균형을 처리하였다. 결측값이 있는 데이터는 데이터 분포를 왜곡시키고 모델의 부정확한 예측으로 이어질 수 있다. 따라서 고객 데이터 중 나이와 성별에 결측값이 있는 것을 확인하고 데이터의 특성이나 분석 목적에 따라 적절한 방법을 모색하여 결측값을 처리하고자 했다. 활용된 데이터셋 중 라벨(0,1) 간 불균형 정도가 크다는 것을 확인하였다. 이는 모델의 편향으로 이어질 수 있기 때문에 적절한 조정을 수행하여 모델 성능을 개선하고자 했다.



2.1.1 다양한 결측치 처리 방법

< Scikit-learn: IterativeImputer >

- Round-robin 형식으로 각 피처에 대해 회귀 분석을 진행해서 결측값을 예측한다.

```
from sklearn.impute import IterativeImputer

imputer = IterativeImputer(max_iter = 10, random_state = 0)
imputer.fit_transform(train_df)
```

< K-Nearest Neighbor 알고리즘을 사용해 가장 근접한 데이터를 k개 찾는 방식 >

- KDTree를 생성한 후 가장 가까운 이웃을 찾는다. K개의 NN을 찾은 뒤에 거리에 따라 가중 평균을 취한다
- 평균, 중앙값보다 정확한 경우가 있다는 장점이 있지만 이상치에 민감하다.

```
from impyute.imputation.cs import fast_knn

np_imputed = fast_knn(df.values, k=5)
df_imputed = pd.DataFrame(np_imputed)
```

< MICE(Multivariate Imputation by Chained Equation)>

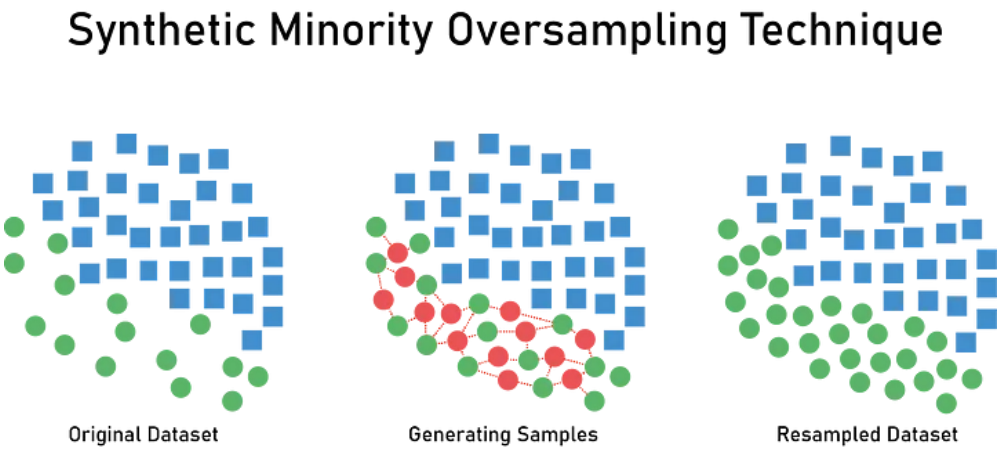
- 누락된 데이터를 여러번 채우는 방식으로 여러 결측치 대치 세트를 만들어 with함수로 특정 통계모델링을 수행하고 pool함수로 생성한 m개의 대치세트를 평균하여 결과를 도출한다.
- 연속형, 이진형, 범위형 패턴도 처리할 수 있다.

```
from impyute.imputation.cs import mice

np_imputed=mice(df.values)
df_imputed = pd.DataFrame(np_immputed)
```

2.1.2 Label 불균형 해소 방법

< SMOTE >



- SMOTE의 동작 방식은 데이터의 개수가 적은 클래스의 표본을 가져온 뒤 임의의 값을 추가하여 새로운 샘플을 만들어 데이터에 추가하는 오버샘플링 방식이다.

```
from imblearn.over_sampling import SMOTE


smote = SMOTE(random_state=0)
X_train_over,y_train_over = smote.fit_sample(X_train,y_train)
```

< Cost-sensitive learning >

- 상대적으로 적은 클래스를 잘못 분류하는 것에 대한 패널티를 부여해서 모델이 이탈 고객에 대한 성능이 더 높아지도록 도와준다.
- scale_pos_weight로 구현이 가능하다.
- 본 프로젝트에서는 대용량의 데이터를 처리하기위해 cost-sensitive learning을 사용하여 데이터 불균형을 해소하였다. 이에 따라 성능이 **Auc 기준 0.001정도** 향상 될 수 있었다.

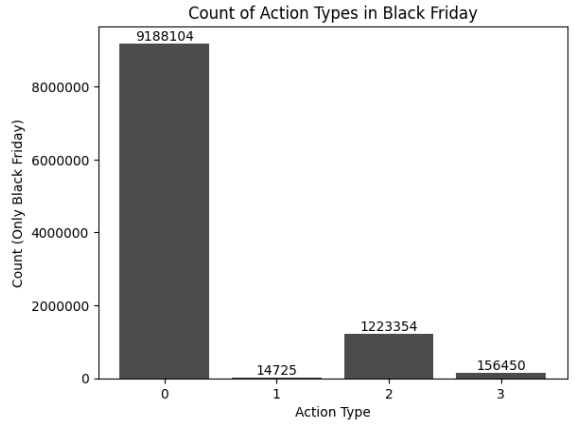
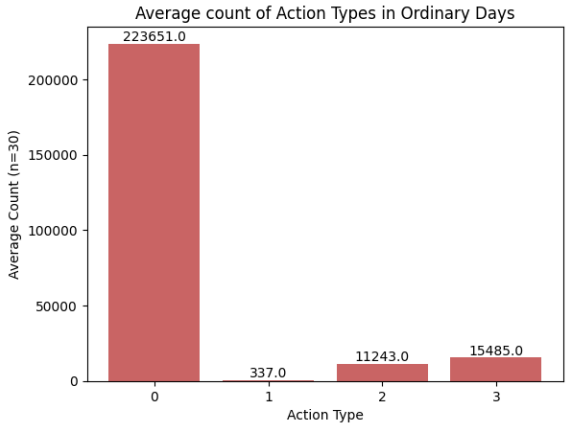
```
X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=VALID_SET_SIZE,
random_state=RANDOM_SEED)


dtrain = DMatrix(X_train, label=y_train)
dvalid = DMatrix(X_valid, label=y_valid)
watchlist = [(dvalid, 'valid')]
params = {
    'max_depth': 7,
    'min_child_weight': 200,
    'colsample_bytree': 0.8,
    'subsample': 0.8,
    'eta': 0.04,
    'seed': RANDOM_SEED,
    'eval_metric': 'auc',
    'scale_pos_weight': 5
}
booster = train(params, dtrain, num_boost_round=2000, evals=watchlist,
early_stopping_rounds=50, verbose_eval=True)
```

2.2 EDA & Feature engineering

2.2.1 EDA

EDA 단계에서는 변수 간의 상관 관계를 체계적으로 분석하였다. 주제와 관련하여 소비자 행동이나 판매 동향, 프로모션 효과를 살펴보기 위해 블랙 프라이데이 트렌드와 관련한 변수들의 관계를 살펴보고 데이터를 파악하고자 했다.

블랙프라이데이	블랙프라이데이 제외 일자의 평균
	

Action type의 변동률을 파악해봤더니
click은 4008%, add-to-cart는 4269%, purchase는 10781%, add-to-favorite는 910% 였다.
이는 실제로 구매까지 이어지는 일자는 Double11이었다는 것을 알게 되었으며, 시간과 action_type의 상호작용도 파악해봐야할 과제임을 알게 되었다.

2.2.2 Feature Engineering

변수가 어떻게 상호작용하며 데이터의 패턴을 어떻게 형성하는지를 살펴보고 적절한 피처 엔지니어링을 수행하여 새로운 파생 변수를 만들었다.

Feature Category	Feature Sub-Category	Feature examples
User-Related Feature	Dummy Feature	age_range
	User Count Feature	Items_user,categories_user, merchants_user, brands_user, dates_user, periods_user, action_types_users
	User Count Feature (per action_types)	clicks_user, carts_user, purchases_user, favourites_user
	User Ratio Feature	clicks_in_merchant_ratio_perspective, carts_in_merchant_ratio_perspective, purchases_in_merchant_ratio_perspective,
	User Ratio Feature (within action_types)	clicks_user_ratio, carts_user_ratio, purchases_user_ratio, favourites_user_ratio
Merchant-Related	Merchant Count Feature	Items_merchant, categories_merchant, users_merchant, brands_merchant, dates_merchant, periods_merchant, action_types_merchant
Feature	Merchant Count Feature (per action_types)	clicks_merchant, carts_merchant, purchases_merchant, fovourites_merchant
	Merchant Ratio Feature	clicks_by_user_ratio_perspective, carts_by_user_ratio_perspective, purchases_by_user_perspective, favourites_by_user_ratio_perspective
	Merchant Ratio Feature (within action_types)	clicks_merchant_ratio, carts_merchant_ratio, purchases_merchant_ratio, favourites_merchant_ratio
Interactive Feature	Interactive Count Feature	Items_user_merchant, categories_user_merchant, brands_user_merchant, dates_user_merchant, periods_user_merchant, action_types_user_merchant
	Interactive Count Feature (per action_types)	clicks_user_merchant, carts_user_merchant, purchases_user_merchant, favourites_user_merchant
	Interactive Ratio Feature (within action_types)	clicks_user_merchant_ratio, carts_user_merchant_ratio, purchases_user_merchant_ratio, favourites_user_merchant_ratio
Time-Related Feature	Categorical Version of time_stamp	time_period
	User Purchase Interval (max-min)	interval

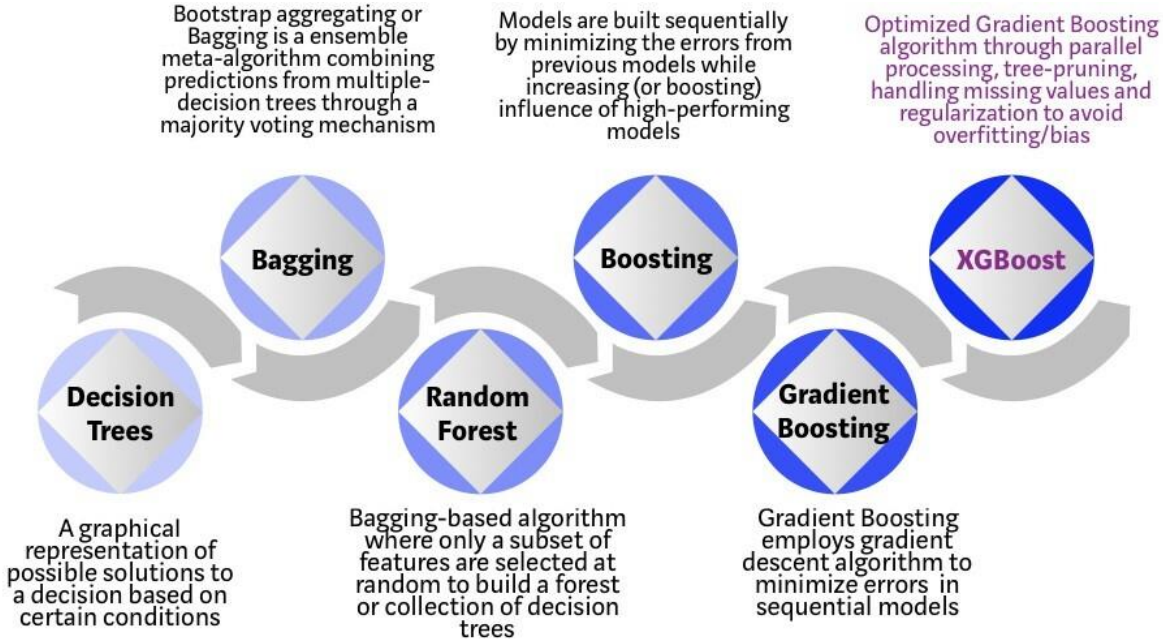
  - 범주형 변수만 존재했기에 정확한 상관관계 파악의 어려움이 있었다. 이에 따라 다양한 파생 변수 생성을 하였고 PCA를 통해 다중공선성을 없애고 유의미한 정보들만 담을 수 있도록 하였다.
  - 결국 필요한 정보는 '유저','판매자' 정보를 바탕으로 재구매를 예측할지의 여부이므로 '유저','판매자','유저*판매자'의 파생변수를 생성하였다.
  - 모두 범주형 데이터이므로 더미변수화 해주었다.
  - Double11시기의 데이터와 다른 시기 데이터가 상당히 다른 의미를 지닌다 보았기에 Double11파생변수도 만들었다.

- Feature engineering 부분에 대한 자세한 코드는 ipynb파일을 첨부하였다.

2.3 Modeling

2.3.1 XGBoost

- GBM을 더 발전시킨 모형으로 Decision Tree의 앙상블 모형이다.
- Image나 Text와 같은 비정형데이터에서는 Nerual Network 모델이 압도적인 성능을 보이고 있지만, 정형데이터에서는 XGBoost와 같은 tree based 알고리즘이 현재까지는 가장 좋은 알고리즘으로 평가받음.



2.3.2 Catboost

- Catergorical feature를 처리하는데 중점을 둔 알고리즘
- 기존 GBM기반 알고리즘들이 가지고 있는 target leakage 문제와 범주형 변수 처리 문제를 ordering principle과 새로운 범주형 변수 처리 방법으로 해결하고자 나왔다..

3. Experiments

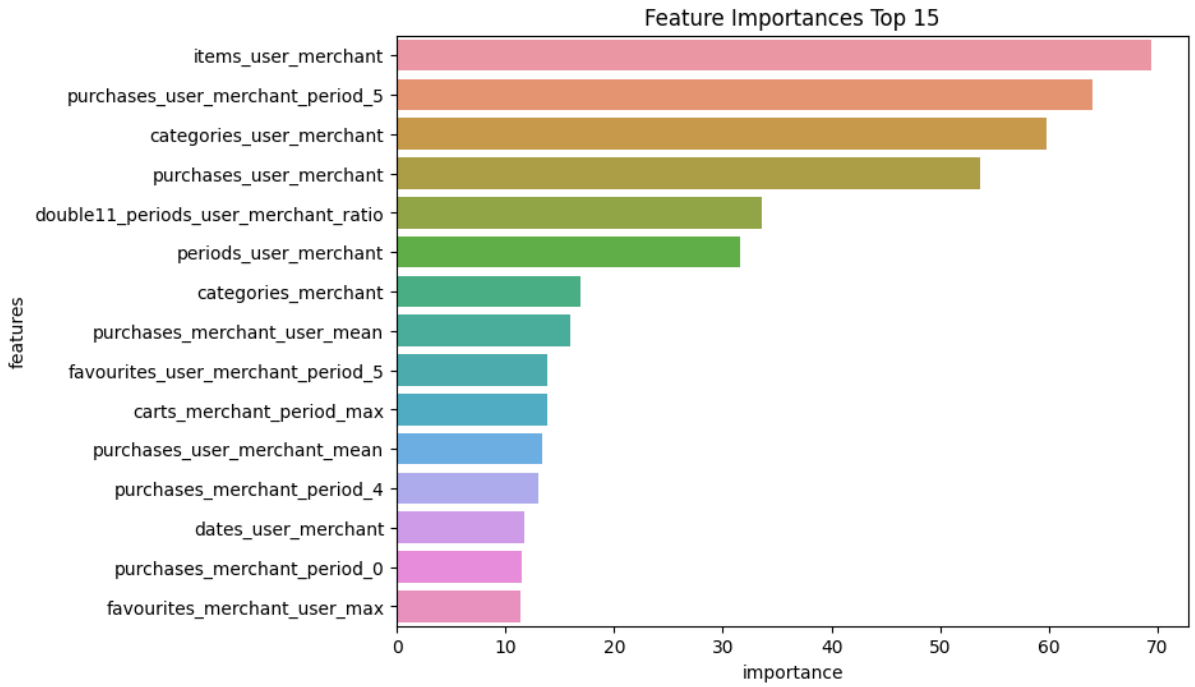
3.1 Experimental Setup

- 앞서 언급한 방식들을 바탕으로 파생변수를 생성한뒤 모델링을 진행하였다.
- 모델링 과정에서는 앙상블 모델을 활용하였다. 여러 다양한 모델을 결합함으로써 모델의 다양성을 확보하고 예측 성능을 극대화하고자 했다.
- 또한, 예측 성능을 최적화하기 위해 모델 튜닝과 최적화 작업을 수행하여 학습된 모델이 새로운 데이터에 대해 강건하고 정확한 예측을 수행할 수 있도록 조절하였다.
- 아래는 최종 모델의 구조이다.

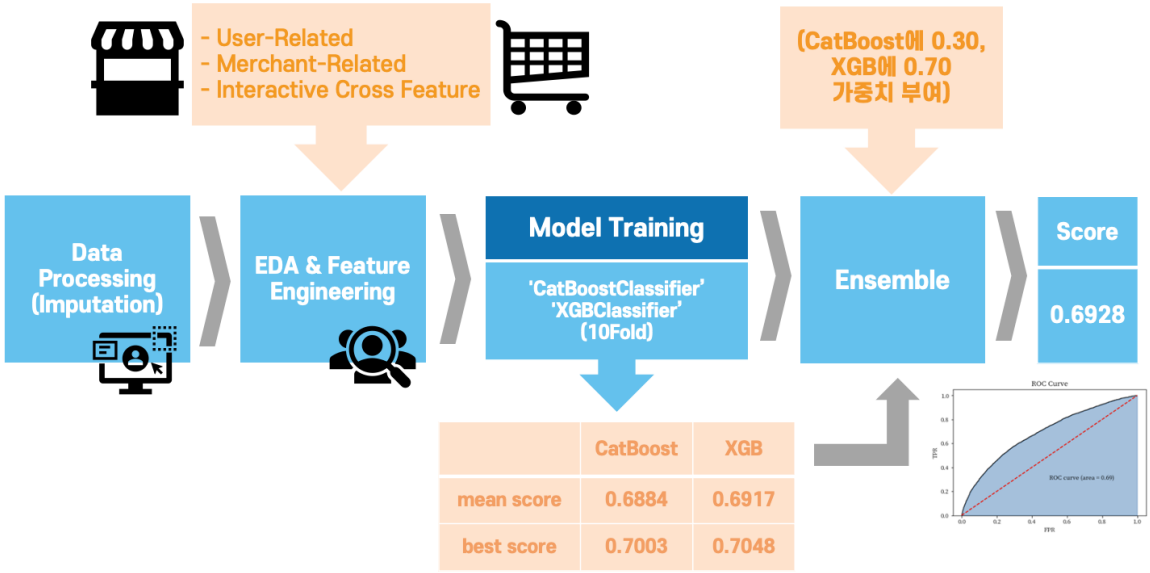
```
models = {
    'CatBoostClassifier': [CatBoostClassifier, {
        'depth': 6,
        'learning_rate': 0.05,
        'iterations': 1200,
        'eval_metric': 'AUC',
        'scale_pos_weight': 3,
        'random_state': RANDOM_SEED,
        'thread_count': 8,
        'silent': True
    }],
    'XGBClassifier': [XGBClassifier, {
        'max_depth': 7,
        'n_estimators': 1000,
        'min_child_weight': 200,
        'colsample_bytree': 0.8,
        'scale_pos_weight': 5,
        'subsample': 0.8,
        'eta': 0.04,
```

```
        'objective': 'binary:logistic',
        'use_label_encoder': False,
        'seed': RANDOM_SEED
    }]
}
```

XGBoost 과정에서 feature importance를 확인해본 결과이다



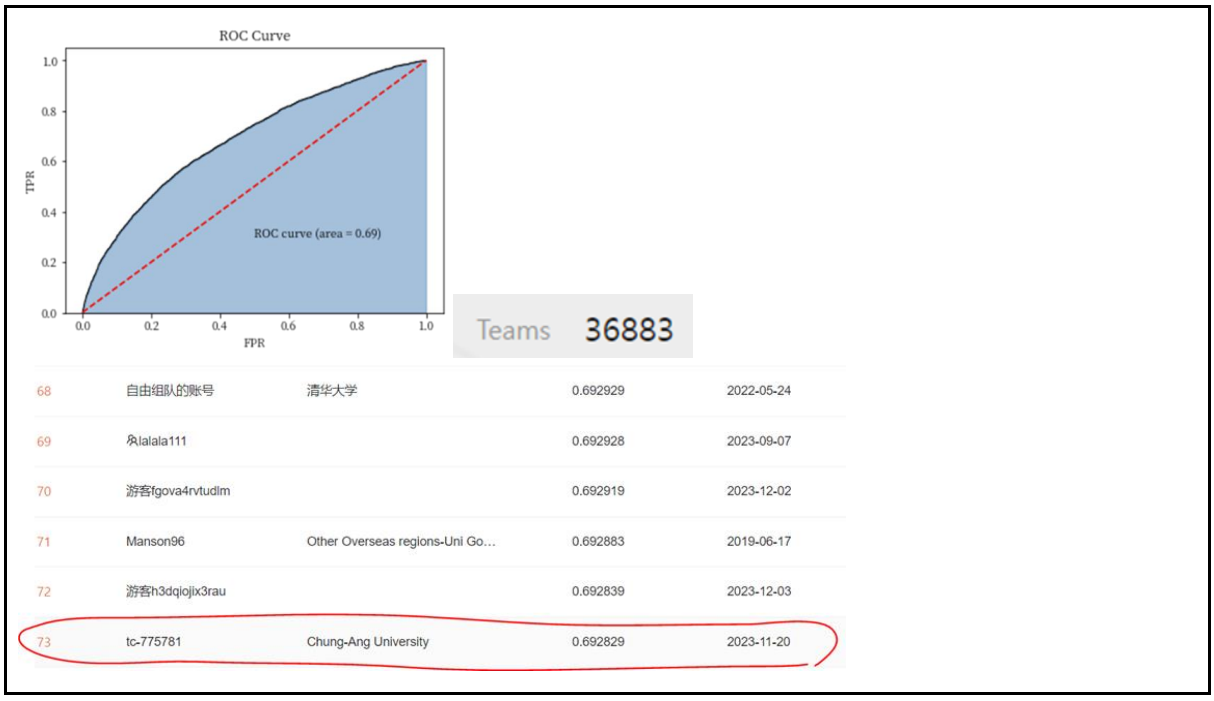
위의 모델로 cross validation 10fold를 진행하였고, 둘 중 성능이 더 좋게 나온 xgboost에 가중치를 부여하여 최종 결과를 내었다.



3.2 Experimental Result

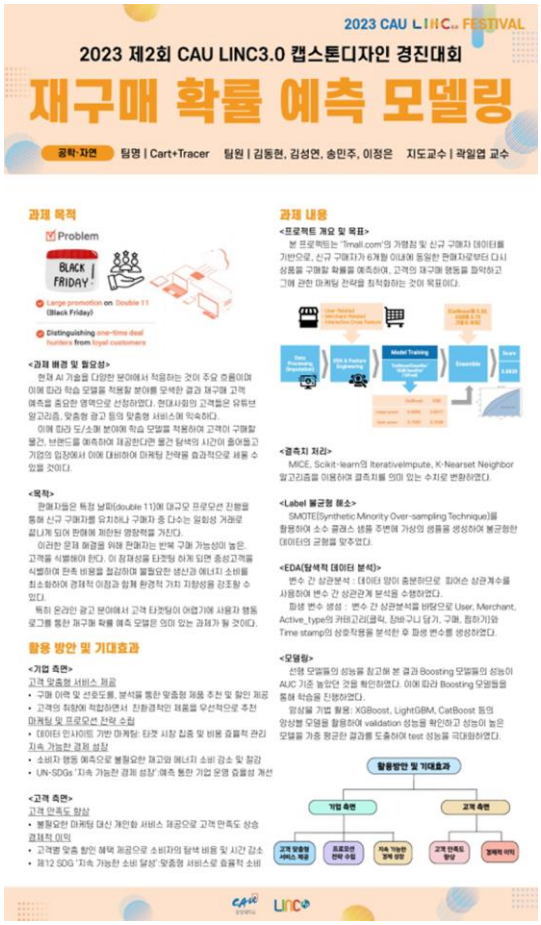
3.1.1 AUC score & 순위

가장 최적화된 모델로 학습 및 테스트를 진행한 결과 AUC score가 0.692829로 나타났다. 또한 대회를 진행한 Tianchi 사이트에서 전체 36,883명의 팀 중 73등을 달성했다.



3.1.2 CAU LINC Capstone design competition 참가

이뿐만 아니라, 해당 주제로 CAU LINC 캡스톤 대회도 참가하여 예선을 통과하는 성과를 이뤘다.

<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
</table>

4. Conclusion

4.1 모델의 장단점(Pros and Cons)

4.1.1 장점

이번 분석을 진행하면서 만든 모델은 다음과 같은 장점이 있었다. 첫째, 부스팅 모델의 사용을 통해 높은 분석 성능을 제공하여, 새로운 고객의 재구매 확률을 효과적으로 예측할 수 있었다. 둘째, 부스팅 모델의 로버스트한 통계량을 활용함으로써, 데이터의 다양한 패턴을 효과적으로 학습할 수 있었다. 이러한 모델의 강점은 신속하고 정확한 예측을 가능케 하며, 고객의 재구매에 대한 풍부한 통찰력을 제공했다.

4.1.2 단점

한편, 이 모델에는 다음과 같은 단점들이 존재했다. 첫째, 앙상블 모델의 특성상 모델 해석이 어려워, 설명력이 좋더라도 고객의 재구매에 주요한 영향을 미치는 요인을 명확히 이해하는 데 어려움이 있었다. 이로써 모델이 도출한 결과의 심도 있는 해석이 어려워지는 문제가 발생했다. 둘째, T-mall에서 제공된 데이터셋의 양이 너무 방대해 분석에 상당한 시간을 소모하게 되었으며, 실질적인 결과 도출까지의 시간이 너무 길어지는 문제가 있었다. 이는 분석의 효율성과 실용성에 영향을 미치는 중요한 제약으로 작용했다.

4.2 분석의 향후 발전 방향(Future Direction)

마지막으로, 위와같이 파악한 모델의 장단점을 바탕으로, 분석의 정확성을 향상시킬 수 있고 향후 해당 분석을 활용할만한 부분을 다음과 같이 탐색했다.

- **Feature Engineering 개선** : 추가적인 특성과 도메인 지식을 찾아 활용하여 분석 모델이 재구매 행동을 더 잘 예측하도록 한다.
- **모델 해석력 개선** : 모델의 각 요인에 대한 해석력을 향상시킬 방법을 찾아 재구매에 영향을 끼치는 요인들을 좀 더 명확하게 파악하도록 한다.

- **실시간 예측 구현 가능성** : 차후 이 모델을 실시간 예측에 적용, 새로운 고객의 재구매 행동을 신속하게 식별하고 대응할 수 있는 시스템으로 발전시킬 가능성을 찾는다.
- **계산 효율성 최적화** : 대규모 데이터셋을 다룰 때 계산 효율성을 최적화하여 분석에 걸리는 시간을 최소화 할 수 있는 방법을 찾는다.