Learning Outcomes

Objective

ELO has ML models to understand the importance of aspects and preferences for their customer's lifecycle.

We use **Root Mean Square Error (RMSE)** to predict the customer loyalty.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2},$$

So we can give better opportunities to most loyalty customers (i.e. promotions)

Steps

First thing

We reduce the memory size with **reduce_men_usage** user defined function

After reducing, filled all the missing values in historical and new_merchant data.

Feature Extracting for Historical and New_Merchant data

- Purchase Date, Authorized Flag columns from historical and new_merchant data
- Calculated sum, min, max, variance from the features we extracted
- Created new columns for these calculated features, group by them with card_id (submission purpose), merge these columns with train data and test data.
- After adding these data into train and test, we deleted historical and new_merchant.

Outliers

We added a new column in training data set for outliers

We assigned 1 to the outliers that are below -30, otherwise 0.

We trained the Light GBM model with stratified k-fold with 5 folds, draw feature_importances and calculated root mean square errors.

Tuning the hyper-parameters

```
params = {'boosting': 'gbdt',    ( to tune the hyper-parameters)
        'objective':'regression',
        'metric': 'rmse',
        'learning_rate': 0.01, # 0.003! #0.005 #0.006
        'num_leaves': 110, #110 #100 #150 large, but over-fitting
        'max_bin': 66,  #60 #50 # large,but slower,over-fitting
        'max_depth': 10, # deal with over-fitting
        'min_data_in_leaf': 30, # deal with over-fitting
        'min_child_samples': 20,
        'feature_fraction': 0.5, #0.5 #0.6 #0.8
        'bagging_fraction': 0.8,
        'bagging_freq': 40,#5
        'bagging_seed': 11,
        'lambda_l1': 2, #1.3! #5 #1.2 #
         'lambda_l2': 0.1 #0.1
```

Trial	Early_stopping _rounds	fold	max_depth	Loss error	Rank on Kaggle
1	200	5	Default -1	3.6542	
2	200	10	Default -1	3.6514	
3	100	10	Default -1	3.6526	
4	100	5	6 (trick)	3.650 (better)	425 / 1850
5	200	5	6	3.6504	
6	100	10	6	3.6488 (best result so far)	
7	300	10	6	3.6505	
8	100	5	6	3.650536	Note: Verbose_Eval = 20, Verbose_Eval = 500, 100
					Each testing took 20 ~ 25 mins

In Progress

- CATBoost Algorithm works a bit better Validation which is 3.6280198

  (Just did use CATBoost model implemented which score 23 %

  Before was 25% (accomplishment)
- Currently running with the XGBoost with RepeatKFold

What next ???

- Trying to manipulate outliers more efficiently from training dataset
- Trying to run the model with using categorical features cause it already has built-in categorical feature parameter
- Could change to repeated k-fold

.