Reviewer: WONG, Hoi Ming 20641276

Project # reviewed: 18

Author of the project: XIA Yiqiao

Overall Summary

In the project report, the author discussed i) how datasets are aggregated and merged, ii) what features are engineered and iii) how a predictive model is fitted. Key features of the predictive model were analyzed, visualized, and discussed in a concise manner.

Evaluation & Comments on Clarity & Quality of Writing: 4/5

Comments: The report is overall well-organized, clear and easy-to-understand, with only minor typos or grammatical errors in the analysis & conclusion part. Charts and concrete numbers are well used as supplements to explain the results. One suggestion to improve the report is to elaborate more on the author's feature engineering on the two datasets applications_train and applications_test, where 8 new features have been added while the rationale behind (eg. domain knowledge or empirical evidence) has not been discussed explicitly.

Evaluation Technical Part: 3/5

Comments: The feature engineering and aggregation steps are generally reasonable in different datasets. Several new features added turned out to carry relatively high importance as shown in the feature importance plots. Some areas of clarification and recommendation are discussed below.

- *Exponentially weighted mean/sum*: In bureau and bureau balance datasets, the author aggregated the records by taking the exponentially weighted sum of all the dummy variables for the different status, where we understand the rationale of taking the time factor in account but somehow think this will become more difficult to interpret. We recommend trying out other easy-to-interpret functions like simple sum, mean or max and see which would work better in prediction. If we really stick to exp-weighted sum, we recommend doing some adjustment on the discounting factor (eg. dividing months by 12, ie. weight = exp(months/12)) applied to each status value. In the current code, we note that the weights come down very quickly (as shown in the output next page) so that the contribution from months_balance from 6 months ago is already almost none. Re-scaling of the exponential weights for this engineered feature may be considered. The similar issue appears in handling the credit card balance.

	SK_ID_BUREAU	MONTHS_BALANCE	STATUS	months_exp	0	1	2	3	4	5
26079753	5001709	0	C	1.000000e+00	0	0	0	0	0	0
26079754	5001709	-1	C	3.678794e-01	0	0	0	0	0	0
26079755	5001709	-2	C	1.353353e-01	0	0	0	0	0	0
26079756	5001709	-3	C	4.978707e-02	0	0	0	0	0	0
26079757	5001709	-4	C	1.831564e-02	0	0	0	0	0	0
26079758	5001709	-5	C	6.737947e-03	0	0	0	0	0	0
26079759	5001709	-6	C	2.478752e-03	0	0	0	0	0	0
26079760	5001709	-7	C	9.118820e-04	0	0	0	0	0	0
26079761	5001709	-8	C	3.354626e-04	0	0	0	0	0	0
26079762	5001709	-9	C	1.234098e-04	0	0	0	0	0	0

- *Handling of Data*: The author removed bureau data with >1000 days ago in bureau datasets, and filtered out credit card data with "months_balance" >-3 as active. The treatment sounds slightly arbitrary and lacks theoretical or experimental support.

- *Approached the problem decently with multiple models:* I think the author is objective and open-minded in the sense that the author incorporated all the available datasets for analysis and tried out multiple model settings. He/she did not presume which would work better. The drawbacks are that it could consume much more computing power and that in some cases the conclusion could not be easily drawn. For example, in this project, the author concluded LGBM worked best by comparing AUC scores under the python default parameter settings (eg. max-depth of trees, learning rate, etc) for each model (except for KNN, which the author assigned K = 8) and the single test-train split with random state = 0. To be prudent in drawing conclusions on which model works better, it is recommended that the author could try different parameters or use a 10-fold cross validation with a comparison of the average AUC scores achieved by each model in the testing folds.

- *Replication of results*: When I ran through the codes to replicate the results, I encountered some points that may need the author's clarification. In the data merging part, the program printed the data sets merged. The dataset "credit_card_by_current_feature.csv" was merged and used but the code does not show how it was created (should be in part 1.2?). I would like to clarify if it is equal to "credit_card_by_current.csv". I would also like the author to revisit the part for handling the dataset pos_cash_balance, as the field 'NAME_CONTRACT_STATUS_Active' seems absent. Overall, I was able to largely replicate the author's results, reaching a similar AUC score at 0.771 for LGBM while not using the 'pos_cash_balance' dataset.

```
for file in os.listdir('./working/'):
    if 'feature' in file:
        print(file)
        feature = pd.read_csv('./working/{}'.format(file))
        train = pd.merge(train,feature,on='SK_ID_CURR',how='left',suffixes = ('','_y'))
        test = pd.merge(test,feature,on='SK_ID_CURR',how='left',suffixes = ('','_y'))
variables = train.dtypes
category = variables[variables==np.dtype('O')].index.to_list()
numerical = variables[variables!=np.dtype('O')].index.to_list()
dummy = pd.get_dummies(train[category])
train = pd.concat([train[numerical],dummy],axis=1)

variables = test.dtypes
category = variables[variables==np.dtype('O')].index.to_list()
numerical = variables[variables!=np.dtype('O')].index.to_list()
dummy = pd.get_dummies(test[category])
test = pd.concat([test[numerical],dummy],axis=1)
del dummy
shared = list(set(train.columns).intersection(test.columns))
train = train[['TARGET'] + shared]
test = test[shared]
```

```
bu_features.csv
credit_card_by_current_feature.csv
installment_feature.csv
POS_CASH_by_current_feature.csv
previous_application_feature.csv
```

Extracts from 1.3 Pos_cash_balance

In [93]:
```
by_previous['SK_ID_CURR'].unique().shape
by_current.columns

```

Out[93]:
```
Index(['SK_ID_PREV', 'SK_ID_CURR', 'CNT_INSTALMENT', 'SK_DPD', 'SK_DPD_DEF',
       'MONTHS_BALANCE'],
      dtype='object')
```

In [94]:
```
by_current = pd.read_csv('POS_CASH_by_previous.csv')
active = by_current.loc[(by_current['MONTHS_BALANCE']>-3)&(by_current['NAME_CONTRACT_STATUS_Active']==1)]
numerical = active.columns[2:]
grouped = by_current.groupby('SK_ID_CURR')[numerical].sum().reset_index()
grouped.drop(columns='MONTHS_BALANCE',inplace=True)
grouped.to_csv('POS_CASH_by_current.csv',index=False)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\Anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2656             try:
-> 2657                 return self._engine.get_loc(key)
   2658             except KeyError:
```

Confidence/Details of Assessment: 3/3 – I have carefully read the paper and checked the results.