Credit Card Default Prediction & Analysis

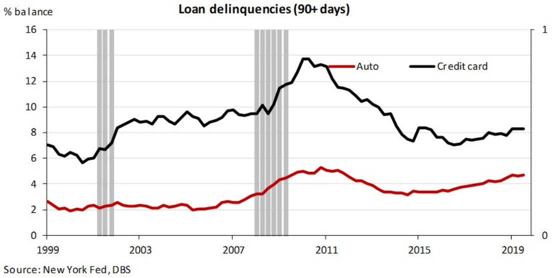
GROUP 1

NUS

March 24, 2021

Motivation

Credit risk plays a major role in the **banking industry** business. However, with the growing number of credit card users, banks have been facing an escalating credit card default and delinquency rate.



Motivation

Fortunately, large scale (behavior) data is naturally suitable for developing machine learning models. More economic researchers and data scientists are focusing on this area.

Research Papers:

- Forecasting and stress testing credit card default using dynamic models (2013)
- Credit Card Default Prediction using Machine Learning Techniques (2018)
- Enhanced Recurrent Neural Network For Combining Statistic and Dynamic Features for Credit Card Default Prediction (2019)
- ...

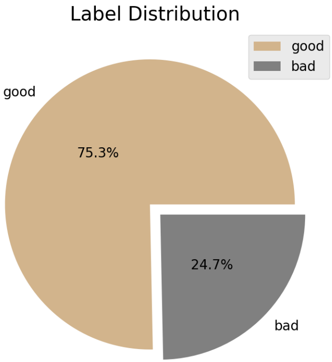
Data Science Competitions:

- Kaggle: Credit Card Default
- Kaggle: Default of Credit Card Clients
- "Magic Mirror Cup" Risk Management Algorithm Competition
- CMB Fintech Elite Training Camp (2020)
- **The 2nd Yipay Cup Big Data Modeling Competition**
- ...

Dataset

From "The 2nd Yipay Cup Big Data Modeling Competition"

- Supervised learning
- Imbalanced classification problem
- Real but desensitized data
- 47782 individuals
  - Train and Validation set: 80%
  - Test set 20%
- 3 data frames:
  - base
  - transaction
  - operation



Dataset: base.csv

DataFrame 1: base.csv → Static information

	user	sex	age	provider	level	...	product4_amount	product5_amount	product6_amount	product7_cnt	product7_fail_cnt
0	Train_12996	category 0	24889	category 0	category 2	...	level 0	level 0	level 1	24712	24712
1	Train_39734	category 0	24859	category 0	category 2	...	level 0	level 0	level 14	24712	24706
2	Train_09006	category 0	24931	category 0	category 2	...	level 0	level 0	level 20	24712	24706
3	Train_35097	category 1	24938	category 2	category 2	...	level 0	level 0	level 1	24712	24712
4	Train_47615	category 0	24853	category 0	category 1	...	level 0	level 0	level 1	24706	24706

5 rows × 46 columns

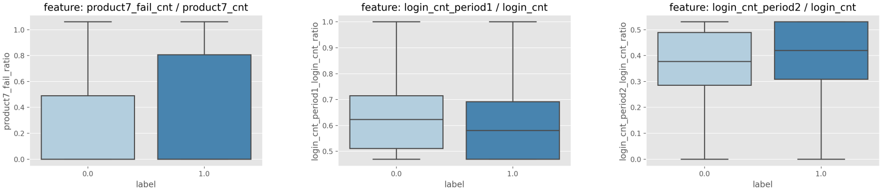
Encoding:

- nominal variables few values (e.g. sex): LabelEncoder
- nominal variables many values (e.g. city, province): CountEncoder
- ordinal or numerical variables: Transfer into integer

Feature Engineering:

- second order addition of numerical variables
- ratios from practical experience or economic intuitive

Dataset: base.csv



(a) (b) (c)

Figure: Box plots of generated feature

Now, We get 233 features.

Dataset: trans.csv

DataFrame 2 & 3: trans.csv & op.csv → Dynamic information

- Time series data starts from some time point
- Each user have different length of behavior

	user	platform	tunnel_in	tunnel_out	...	ip	type2	ip_3	tm_diff
0	Train_13770	46c69cbbce5f1568	b2e7fa260df4998d	6ee790756007e69a	...	NaN	11a213398ee0c623	NaN	19 days 09:02:45.000000000
1	Train_13770	46c69cbbce5f1568	b2e7fa260df4998d	6ee790756007e69a	...	NaN	11a213398ee0c623	NaN	19 days 09:03:58.000000000
2	Train_08351	46c69cbbce5f1568	b2e7fa260df4998d	6ee790756007e69a	...	f10a09fe9e522a47	11a213398ee0c623	ee386d6f9fe45d0d	18 days 11:06:49.000000000
3	Train_08351	42573d7287a8c9c2	NaN	6ee790756007e69a	...	NaN	NaN	NaN	26 days 09:52:51.000000000
4	Train_08351	42573d7287a8c9c2	NaN	6ee790756007e69a	...	NaN	NaN	NaN	26 days 07:50:05.000000000

5 rows × 10 columns

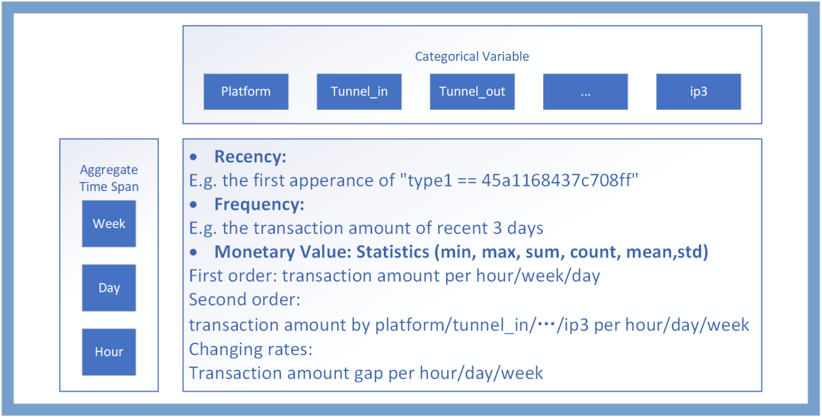
Main Issue

Represent the vector of each user's behavior record (N $\times$ 10) into an aggregation vector (1 $\times$ M), then we can concatenate them with features from base.csv, and feed into model.

Dataset: trans.csv

Feature Engineering:

Tracking user's **Recency**, **Frequency**, **Monetary Value**



Dataset: trans.csv

Feature Engineering: (mainly focusing on amount)

- Manually written functions

```
# define a function to calculate users' transcation amount linked with feature 'hours'
def gen_user_window_amount_features(df, window):
    group_df = df[df['hour']>window].groupby('user')['amount'].agg([
        ('user_amount_mean_{}h'.format(window), 'mean'),
        ('user_amount_std_{}h'.format(window),'std'),
        ('user_amount_max_{}h'.format(window),'max'),
        ('user_amount_min_{}h'.format(window),'min'),
        ('user_amount_sum_{}h'.format(window), 'sum'),
        ('user_amount_med_{}h'.format(window), 'median'),
        ('user_amount_cnt_{}h'.format(window), 'count')
    ]).reset_index()
    return group_df
```

- Word Embedding: word2vec

Now, we get 671 features.

Dataset: op.csv

Feature Engineering: (no amount here)

- Manually written functions
- Word Embedding: word2vec
- Truncated svd on TF-IDF (n_components=10)
- Truncated svd on CountVectorizer (n_components=10)

Now, we get 994 features.

Baseline Model: LightGBM (5-fold CV)

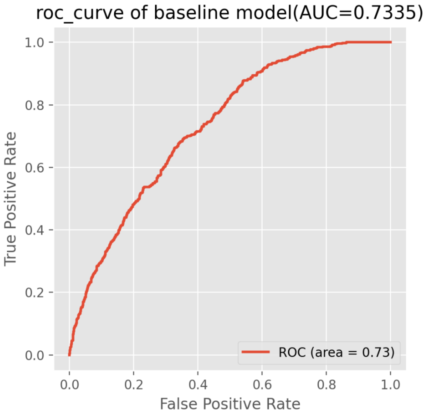


Figure: ROC curve of baseline model

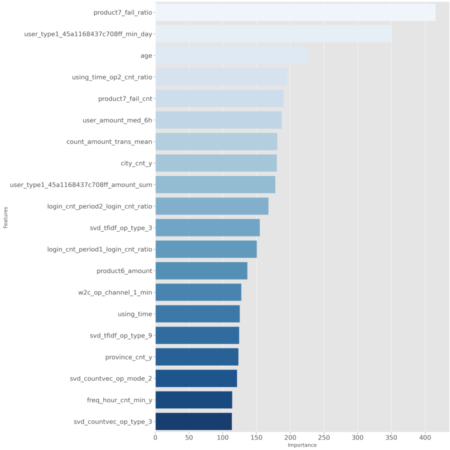


Figure: Feature importance

Working On

- Model Tuning
- Try other models, e.g. XGBoost
- Stacking
- Oversampling (to handle imbalanced learning issue), e.g. SMOTE