Fraud Detection in Banking using XGBoost

Alexander Indrajaya L, Novan Dwi Atmaja, and Siti Fatimah

The following report describes the methodology and exploratory process in the prediction of fraud event in banking by using indicators gathered from historical transaction. We also use synthetic minority over sampling (smote) technique to solve the imbalanced data problem. We find that by using smote algorithm the recall value has increased significantly even though this technique caused a slight decrease in the area under curve (auc) and accuracy.

Keywords: credit scoring, xgboost, smote algorithm

Executive Summary

In this report, we will do the fraud detection modeling based on data that has been provided by FinHack 2018. We use the XGboost model as the base model. Previously we would do a features selection and solve unbalanced data problems using the smote algorithm. Next, we will measure the performance and stability of the model.

We divide the data to training and testing (80:20). By using the test data, we found that the smote process provided auc and accuracy reduction of 1% - 4% and a recall increase of more than 28%. The most important variable to predict the credit card default is machine type (ATM, EDC).

The Dataset

The data we use consists of 13,125 data. The data contains related history of transaction and the status of whether the transaction is good or not. Data details can be seen on the finhack dashboard.

Exploratory Data Analysis

Training data consists of 910 (6.93%) fraud and 12,215 (93.07%) non-fraud.

First, we will measure the predictive power of characteristic using information value (IV). One rule of thumb regarding IV is:

- Less than 0.02 : unpredictive
- 0.02 to 0.1 : weak
- 0.1 to 0.3 : medium
- More than 0.3 : strong

We measure IV using “library (smbinning)”. This algorithm will categorize numeric data into certain bins based on the Conditional Inferences Tree.

```
> iv_table
                        Char     IV               Process
12                id_channel 0.8676     Factor binning OK
11         kepemilikan_kartu 0.8484     Factor binning OK
```

```
13             nilai_transaksi 0.4193    Numeric binning OK
16    minimum_nilai_transaksi 0.2288    Numeric binning OK
17 rata_rata_jumlah_transaksi 0.1558    Numeric binning OK
14  rata_rata_nilai_transaksi 0.0775    Numeric binning OK
10          kuartal_transaksi 0.0254     Factor binning OK
1                  tipe_kartu     NA   Too many categories
2                 id_merchant     NA   Too many categories
3                  tipe_mesin     NA   Too many categories
4              tipe_transaksi     NA   Too many categories
5              nama_transaksi     NA   Too many categories
6                   id_negara     NA   Too many categories
7                   nama_kota     NA   Too many categories
8                lokasi_mesin     NA   Too many categories
9               pemilik_mesin     NA   Too many categories
15   maksimum_nilai_transaksi     NA No significant splits
```

Based on IV, it was found that there were 3 features: id_channel, kepemilikan_kartu, and nilai_transaksi were strong predictor to measure fraud.

From the table above also obtained that the maksimum_nilai_transaksi are significantly do not affect the fraud detection (the tree is not formed).

Predicting

The available data showed the existence of imbalanced data cases.

```
> table(data_train$flag_transaksi_fraud)
    0     1
12215   910
```

However, most of the existing state-of-the-art classification approaches are well developed by assuming the underlying training set is evenly distributed. Thus, they are faced with a severe bias problem when the training set is a highly imbalanced distribution. The resulting decision boundary is severely biased to the minority class, and thus leads to a poor performance according to the receiver operator characteristic (ROC) curve analysis. The synthetic minority oversampling technique (SMOTE) is an important approach by oversampling the positive class or the minority class.

At this stage, we will predict credit default by using XGBoost. We will compare the effect of smote on training data on model performance. The following are the steps we did:

1. Create new data training using SMOTE algorithm
2. convert categorical factor into one-hot encoding
3. construct XGBoost object dengan xgb.DMatrix
4. Construct XGBoost model

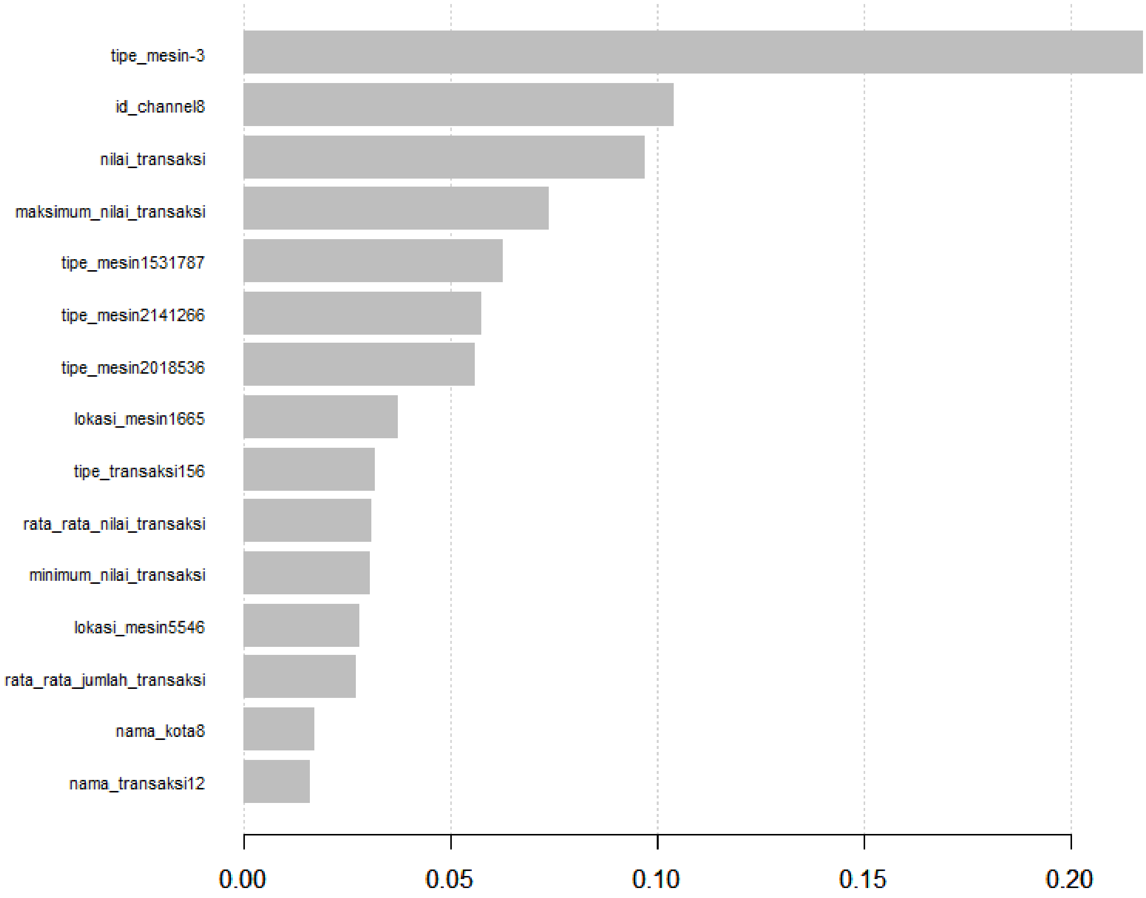
5. Predict credit default by using model above
6. Model evaluation

The following is the model comparison:

```
> table_akurasi
          model                data       auc  accuracy    recall
1       xgboost       train (model) 0.9103861 0.9580912 0.4485597
2       xgboost                test 0.8958366 0.9440213 0.3314917
3 smote-xgboost smote_train (model) 0.9538842 0.9207438 0.8006401
4 smote-xgboost                test 0.8577229 0.9310739 0.4254144
```

The smote process slightly reduces the level of accuracy, but can improve recall. By considering that the accuracy is still large, the model with smote will be chosen as the prediction model.

Here the important variables of smote-xgboost model:



The most important variable is machine type, it contributes more than 20%.