Case Study Report - Wells Fargo QAP

by Shuangyan Wu

November 12, 2022

Contents

List of Figures

List of Tables

Summary

Financial banks reply on models to decide for credit product approval and decline for risk control. Whether a customer's account becomes defaulted or not after its approval and opening can be used to evaluate the risk of approving future products. So in this report, two models (logistic regression and XGBoost) were developed to predict the binary response (account defaulted or not), denoted as DI_Defind. The data sets were divided into training, validation, and testing sets. After data cleaning, the logistic regression model was built, and the variables were selected using step-wise selection, the final model includes 9 predictors as shown in the model equation below. It has accuracy of 0.90, AUC of 0.79, and F1 score of 0.35. The model indicates applicants holding accounts from the bank is predicted to have lower probabilities of having an defaulted account after approval. The final XGBoost model has accuracy of 0.87, AUC of 0.87, and F1 score of 0.50. Due to its weight control on the rare class in the imbalanced data, XGBoost model shows better recall ($0.66 > 0.26$), F1 score, and AUC than the logistic regression model. It has an overall better performance, especially for predicting the class with lower observations, however lower interpretability. A model should be selected based on its requirements such as high overall accuracy, high recall, or high interpretability.

$$
\begin{aligned}
log(\frac{\hat{p}}{1-\hat{p}}) = & -3.02 - 0.00014 * ABC - 0.0046 * CA + 1.08 * ND12 \\
& + 4.53e-01 * ND6 + 0.26 * NI12 + 0.158 * NC12 + 5.29 * UOC \\
& + 0.67 * PO50 - 0.24 IND * I_1.
\end{aligned} \tag{1}
$$

Where p is the probability of being in class 1 (defaulted account). I_1 is the indicator for the sample in ind_XYZ level 1 (Already having accounts).

1 Introduction

Financial institutions utilize statistical and machine learning models to determine whether they approve or decline applications for credit products such as credit cards and home loans. The factor of defaulted account after its approval and opening provides important information for risk control. If the applicant is more likely to have a defaulted account, then more strict requirements for new credit products approval may be used. Therefore, the objective of this case study is to develop two binary classification models for predicting the defaulted account response with 20 related explanatory variables. Here, logistic regression and XGBoost models were developed after data cleaning and exploration. The model performance was compared between the two final models using related criteria such as AUC, accuracy, F1 score, recall, and precision.

2 Exploratory data analysis

2.1 Data summary

The provided datasets includes 20 explanatory variables (2 categorical and 18 continuous) and 25000 observations from the training (20000) and testing (5000) sets. The response variable (Def_ind) is binary and categorical with input 1 indicating the account is defaulted after opening in last 18 mouths, 0 otherwise. Although some explanatory variables/features are integers, for instance num_acc_30d_past_due_12 months and num_inq_12_month, they are treated as continuous variables later for model fitting. Table 1 shows the variable names, their brief description, and their types. The additional short notations beside the variables names are created for easy reference. In order to apply the train-validation-test approach, the given training data is further spitted into training and validation sets with the ratio of 80/20. The data summary (statistics or frequency counts) for the final training data set is shown in the R output below. No extreme input or typos were found from the datasets. It is noticed the response variable is imbalanced with 14393 samples at level 0 and 1607 samples at level 1.

Table 1: Information for all provided variables

Notation and variable names	Brief Description	Type
TB.tot_balance	Total balance for all credit product	Continuous
ABC.avg_bal_cards	Average balance for all cards	Continuous
CA.credit_age	Age of the first obtained credit product	Continuous
CAG.credit_age_good_account	Age of the oldest good credit product	Continuous
CCA.credit_card_age	Age of the oldest credit card	Continuous
ND12.num_acc_30d_past_due_12_months	# account (≥30 days delinquent, 21 months)	Continuous
ND6.num_acc_30d_past_due_6_months	# account (≥30 days delinquent, 6 months)	Continuous
NMD.num_mortgage_currently_past_due	#mortgages (delinquent, 6 months)	Continuous
TAD.tot_amount_currently_past_due	Total amount past due (all credit account)	Continuous
NI12. num_inq_12_month	# of inquires in last 12 months	Continuous
NCI24.num_card_inq_24_month	# of credit card inquires, last 24 months	Continuous
NC12.num_card_12_month	# of credit cards opened, last 12 months	Continuous
NA36.num_auto_36_month	# of auto loans opened, last 36 months	Continuous
UOC.uti_open_card	Utilization on open credit card amounts	Continuous
PO50.pct_over_50_uti	% of account with >50% utilization	Continuous
UMC.uti_max_credit_line	Utilization of credit account (highest limit)	Continuous
PC50.pct_card_over_50_uti	% of credit cards with >50% utilization	Continuous
IND.ind_XYZ	Binary, already holding accounts	Categorical
RI. rep_income	Annual income	Continuous
RE.rep_education	Education level (4 levels)	Categorical
DI.Def_ind	Binary, account defaulted in past 18 months	Categorical

Data summary for some continuous variables:

```
TB                ABC               CA               CAG              RI
Min.   :     0    Min.   :    0     Min.   :  0.0    Min.   :  2.0    Min.   : 20000
Median :107518    Median :12212     Median :281.0    Median :146.0    Median :166555
Mean   :107302    Mean   :12195     Mean   :280.8    Mean   :146.1    Mean   :166554
Max.   :200000    Max.   :23854     Max.   :560.0    Max.   :300.0    Max.   :300000
SD     :22428     SD     :306       SD     :73.1     SD     :38.5     SD     :33387

CCA               TAD               NI12             NCI24            PO50
Min.   :  0.0     Min.   :    0.0   Min.   :0.0      Min.   : 0.0     Min.   :0.00
Median :285.0     Median :    0.0   Median :0.0      Median : 0.0     Median :0.55
Mean   :285.1     Mean   :  357.2   Mean   :0.6      Mean   : 1.1     Mean   :0.55
Max.   :511.0     Max.   :35000.0   Max.   :8.0      Max.   :17.0     Max.   :1.00
SD     :64.1.     SD     :1807.6    SD     :1.2      SD     :2.0.     SD     :0.12
UOC               PO50              UMC
Min.   :0.00      Min.   :0.00      Min.   :0.00
Median :0.49      Median :0.49      Median :0.47
Mean   :0.49      Mean   :0.48      Mean   :0.47
Max.   :1.00      Max.   :1.00      Max.   :1.00
SD     :0.13.     SD     :0.12      SD     :0.13
```

Frequency count for categorical variables and continuous variables with small ranges of integers:

```
ND12        ND6.        NMD         NC12        NA36        IND          RE
```

```
0:14084    0:15538    0:15505    0:11963    0:13438    0:11973    college    :9719
1: 1409    1:  443    1:  495    1: 3728    1: 2526    1: 4027    graduate   :1919
2:  417    2:   19               2:  304    2:   36               high_school:4250
3:   80                          3:    5                         other      : 111
4:    8
5:    2

DI
0:14393
1: 1607
```

2.2 Data exploration

After exploring plots with different variables, correlation trends are found between some continuous variables. As an example, Figure 1 shows there is some positive correlation between the four included variables: UMC, UOC, PO50, and PC50. When either variable increases, other three variables tend to increase. Similarly, a similar trend can be found between the ABC and TB variables shown in the Appendix (Figure 8). Besides, some variables are found to possibly influence the response variable DI. As seen in Figure 2 and 3, the proportion of defaulted account (DI level 1, darker color) in applicants who already have accounts from XYZ bank (IND level 1) is smaller than it in the group without an account (IND level 0), which may indicate a smaller probability of defaulted account in applicants who already hold accounts. And increasing the NI12, UOC, and PO50 variables tend to increase the proportion of defaulted account.

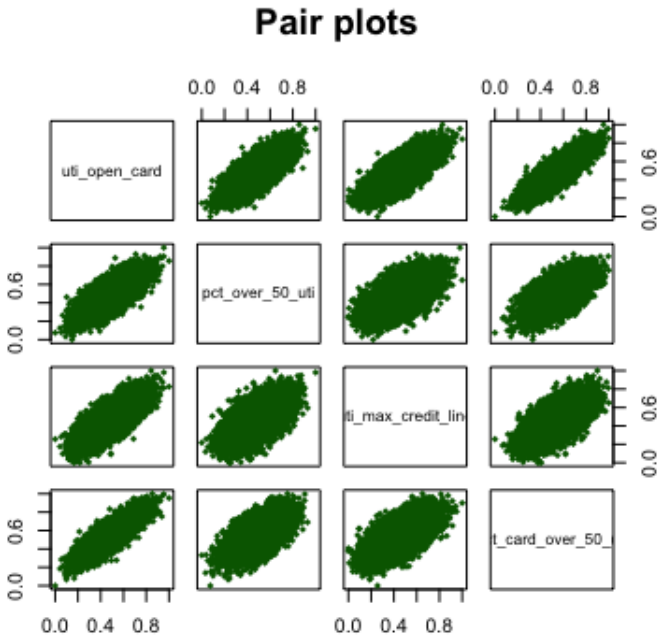

Figure 1: Matrix plots of some continuous variables

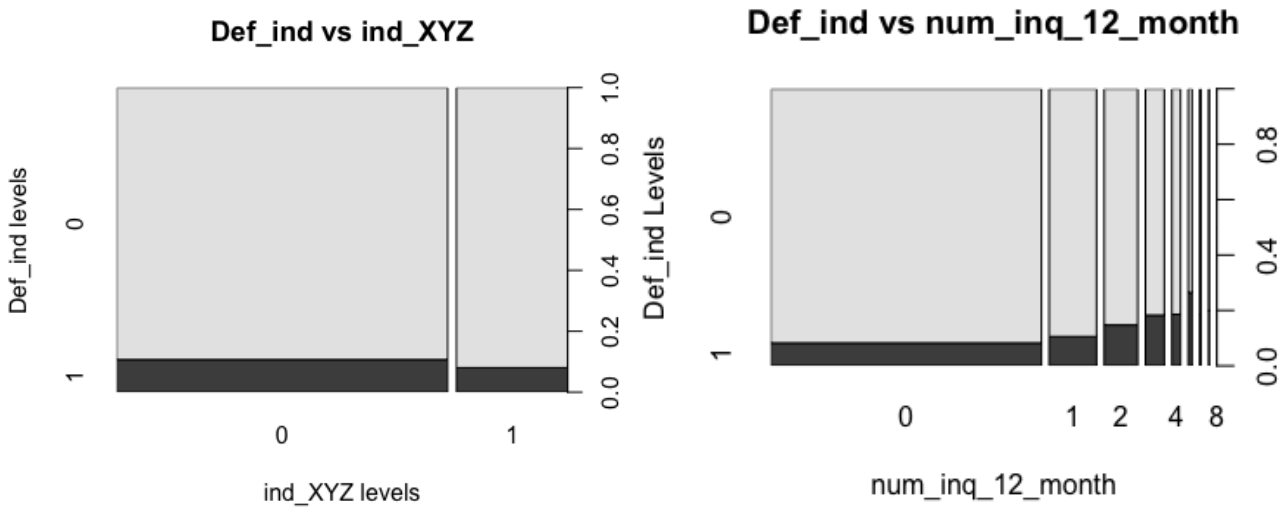

Figure 2: Plot of DI vs IND (left) and DI vs NI12 (right).

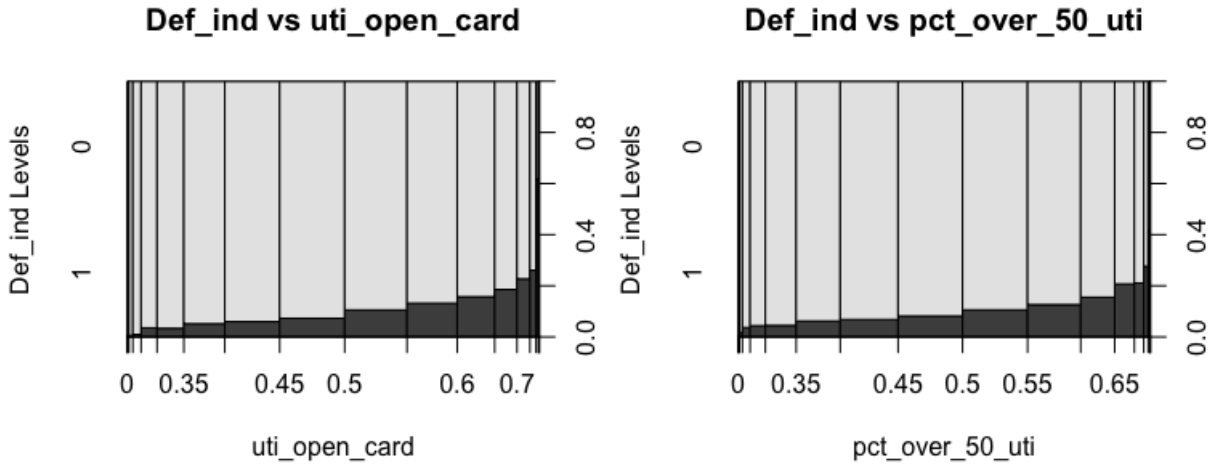

Figure 3: Plot of DI vs UOC (left) and DI vs PO50 (right).

2.3 Missing data imputation

Before analyzing the data, data cleaning was completed. As shown in Table 2, there are 9.7%, 7.7% and 0.00625% of the data missing in the training set for the explanatory variables: PC50, RI, and RE, respectively. Similar percentages of missing data are found in the validation and training sets. More information about the ratios (missing count/total) and distribution of missing data in the variables can be found in the Appendix (Figures 9-11). The missing values of continuous variables PC50 and RI in all data sets were imputed using the predictive mean matching (*pmm*) function from R MICE package, which imputes a missing data point by selecting the non-missing data of another sample which has the nearest predicted value to the predicted value of the missing one, here, other continuous variables excluding the response variables were used as predictors [1]. The missing values of categorical variables RE in all data sets were imputed with its mode in each dataset.

Table 2: Missing data percentages (counts).

Datasets	PC50	RI	RE
training set	9.7 (1554)	7.7 (1235)	0.006 (1)
validation set	10.1 (404)	8.1 (324)	-
testing set	9.8(489)	8.2 (410)	0.08 (4)

3 Statistical methods

To help decide who to approve and decline for credit product, two binary classification models were developed to predict the probability of an defaulted account after the account being approval and opened and classify samples into 1 (defaulted) and 0 (not defaulted). The models are built using logistic regression and XGBoost based on gradient boosting algorithm. The XGBoost is selected here mainly because it is better and easier to use for imbalanced data since the model will learn from errors and build new trees to minimize the errors made by previous tree. When it fails to predict a class with lower participation, it gives more weightage to it for the next iteration, therefore increase its ability to detect that class. In XGBoost, the weight given to the rare class can be scaled by tuning the scale_pos_weight booster parameter. In contrast, random forest and neuron network algorithms may not treat the class imbalance with proper process, which may result in low model performance for detecting the rare events. Therefore, to use random forest and neuron network, addition data preparation is often needed such as downsampling and upsampling, it increases the time cost and complexity of data analysis. In addition, XGBoost is in general more time efficient, interpretable, and easier to use than neuron network algorithm.

To develop the models, a formal train-validation-testing process is applied, in which, the training set is used for developing model, validation set is used to check model performance and tune model parameters, and the testing set is used for report the performance of final models. Variable selection is completed with step-wise selection in the logistic regression model. Possible collinearity is detected using condition number and variance inflation factor (VIF) which measures the correlation and strength of correlation between explanatory variables in regression models [2-3]. And the model parameters are tuned based on the model performance criteria such as accuracy, AUC, F score, precision, and recall. The final model performance is also reported with these related values. R studio software (version 2022.02.3) is used to analyze the data and generate plots.

4 Results and discussion

4.1 Logistic Regression

4.1.1 Model building

In this section, a logistic regression model is built for the binary response variable (Def_ind) using the steps: (1) checking multicollinearity, (2) fit the model with all variables, and (3) step-wise variable/model selection using Akaike information criterion (AIC). The model equation is shown in equation (2). Table 3 shows the VIF values of all continuous variables, as seen, all values are below 10, which indicates the correlation between the variables is not high enough to be concerned. This is also confirmed by small condition number $5.91 < 30$, another indicator for multicollinearity, it is the ratio of maximum eigenvalue and minimum eigenvalue of the $X^T X$ matrix.

$$log(\frac{p}{1-p}) = X\beta \tag{2}$$

Where X is the matrix for predictors, and β is the matrix for all coefficients. And p is the probability of a defaulted account.

$$AIC = 2K – 2ln(\hat{L}) \tag{3}$$

Where K is the number of model parameters and $\hat{L}$ is the maximum value of the likelihood function for the model.

Table 3: VIF values for the continuous variables.

Variables	TB	ABC	CA	CAG	CCA	ND12	ND6
VIFs	1.97	1.97	5.39	2.76	3.58	3.23	2.98
Variables	NMD	TAD	NI12	NCI 24	NCI12	NA36	UOC
VIFs	3.00	4.24	5.43	5.43	1.02	1.01	6.09
Variables	PO50	UMC	PC50	RI			
VIFs	2.31	2.29	3.50	1.00			

The R output below shows the model summary for the full logistic regression model. It is seen 10 variables including ABC, CA, etc. are significant at $\alpha = 0.1$ while other variables show no significance. Also noticed, dummy variables for the IND and RE categorical variables are automatically created by the logistic regression model with the IND-level 0 and RE-level college as the baseline.

```
Coefficients for the full model:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -2.800e+00  2.881e-01  -9.719  < 2e-16 ***
tot_balance                   -2.448e-06  1.817e-06  -1.347 0.177836
avg_bal_cards                 -1.235e-04  1.353e-05  -9.127  < 2e-16 ***
credit_age                    -5.109e-03  9.220e-04  -5.541 3.00e-08 ***
credit_age_good_account        1.552e-03  1.249e-03   1.242 0.214314
credit_card_age               -1.887e-04  8.516e-04  -0.222 0.824654
num_acc_30d_past_due_12_months  1.010e+00  8.412e-02  12.001  < 2e-16 ***
```

```
num_acc_30d_past_due_6_months    3.267e-01  1.806e-01   1.809 0.070405 .
num_mortgage_currently_past_due  1.988e-01  1.939e-01   1.026 0.305020
tot_amount_currently_past_due    1.151e-05  2.282e-05   0.505 0.613906
num_inq_12_month                 3.222e-01  5.342e-02   6.031 1.63e-09 ***
num_card_inq_24_month           -4.111e-02  3.027e-02  -1.358 0.174499
num_card_12_month                1.555e-01  5.699e-02   2.729 0.006352 **
num_auto_36_month                3.879e-02  7.670e-02   0.506 0.613052
uti_open_card                    5.710e+00  5.610e-01  10.179  < 2e-16 ***
pct_over_50_uti                  6.774e-01  3.582e-01   1.891 0.058570 .
uti_max_credit_line             -4.592e-02  3.347e-01  -0.137 0.890879
pct_card_over_50_uti            -4.528e-01  4.290e-01  -1.055 0.291223
rep_income                      -5.370e-07  8.625e-07  -0.623 0.533573
ind_XYZ1                        -2.398e-01  7.144e-02  -3.357 0.000788 ***
rep_educationgraduate           -4.302e-02  9.672e-02  -0.445 0.656450
rep_educationhigh_school         1.211e-01  6.551e-02   1.848 0.064596 .
rep_educationother              -3.294e-01  3.729e-01  -0.883 0.377049
AIC: 8450
```

After model selection, 9 predictors are selected as shown in the result below, they all are significant at $\alpha = 0.1$. And the final model is shown in equation (4). The coefficient for ind_XYZ1 is -0.24, which indicates applicants who already have accounts from the bank XYZ are predicted to have a lower probability of having defaulted accounts after account approval and opening. And max(VIF) for these selected variables is 2.17 <10, indicating they are not strongly correlated.

```
Coefficients for the final model:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -3.020e+00  2.112e-01 -14.297  < 2e-16 ***
avg_bal_cards                 -1.359e-04  9.719e-06 -13.986  < 2e-16 ***
credit_age                    -4.608e-03  4.022e-04 -11.456  < 2e-16 ***
num_acc_30d_past_due_12_months  1.081e+00  6.394e-02  16.912  < 2e-16 ***
num_acc_30d_past_due_6_months   4.531e-01  1.564e-01   2.897 0.003772 **
num_inq_12_month                2.560e-01  2.068e-02  12.375  < 2e-16 ***
num_card_12_month               1.577e-01  5.642e-02   2.795 0.005195 **
uti_open_card                   5.293e+00  3.506e-01  15.098  < 2e-16 ***
pct_over_50_uti                 6.994e-01  3.577e-01   1.956 0.050510 .
ind_XYZ1                       -2.412e-01  7.138e-02  -3.379 0.000729 ***
---
AIC: 8431.9
```

$$
\begin{aligned}
log(\frac{\hat{p}}{1-\hat{p}}) = &-3.02 - 0.00014 * ABC - 0.0046 * CA + 1.08 * ND12 \\
&+ 4.53e-01 * ND6 + 0.26 * NI12 + 0.158 * NC12 + 5.29 * UOC \\
&+ 0.67 * PO50 - 0.24IND * I_1.
\end{aligned} \tag{4}
$$

Where p is the probability of being in class 1 (defaulted account). I_1 is the indicator for the sample

in ind_XYZ level 1 (Already having accounts).

To check the model generalization ability and performance, it is used to predict the response IND on both training and validation sets. Some of the confusion matrix and statistical information from both sets are shown below. As seen, the results including accuracy, precision, recall, and F1 score are close for both sets meaning the final model selected using the training set can be generalized and it does not have an over-fitting problem.

```
Confusion Matrix and Statistics for training set:
          Reference
Prediction     0     1
         0 14280  1327
         1   112   280

               Accuracy : 0.9101

            Sensitivity : 0.1742
            Specificity : 0.9922
              Precision : 0.7143
                 Recall : 0.1742
                     F1 : 0.2801
      Balanced Accuracy : 0.5832
```

```
Confusion Matrix and Statistics for validation set:
          Reference
Prediction    0    1
         0 3578  325
         1   29   68

               Accuracy : 0.9115

            Sensitivity : 0.17303
            Specificity : 0.99196
              Precision : 0.70103
                 Recall : 0.17303
                     F1 : 0.27755
      Balanced Accuracy : 0.58249
```

4.1.2 Model performance

To report the model performance, the prediction on the testing set is completed using the final model. The results shown below are close to those of both training and testing set. And the model accuracy is as high as 0.9026, precision is 0.5261, and the recall is 0.2620, and the F1 score is 0.3498. The model has a better performance for predicting 0 than predicting 1 due to the imbalanced data.

```
Confusion Matrix and Statistics for testing set:

          Reference
```

```
Prediction    0    1
         0 4382  369
         1  118  131

               Accuracy : 0.9026

            Sensitivity : 0.2620
            Specificity : 0.9738
              Precision : 0.5261
                 Recall : 0.2620
                     F1 : 0.3498
      Balanced Accuracy : 0.6179
                    AUC : 0.7877
```

And the distribution of predicted probability values are shown below. It is right skewed due to the much higher number of level 0 in the testing set. The predicted value range is [0, 0.99] and the median is 0.067.

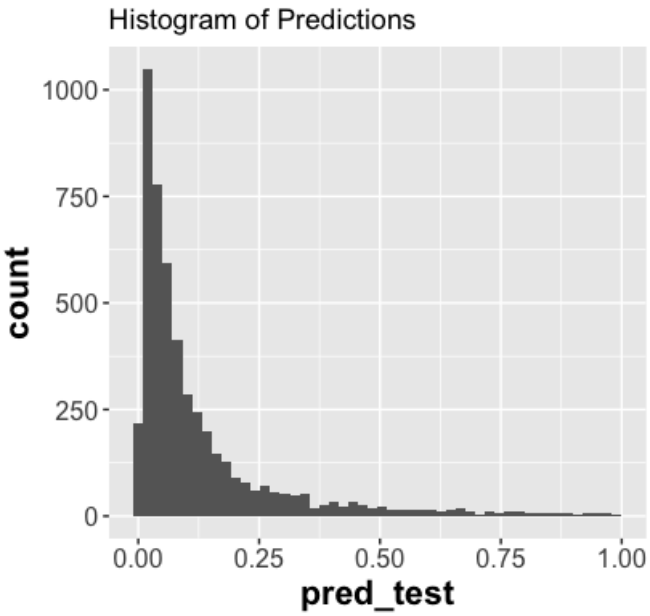

Figure 4: Distribution of predicted values.

4.2 XGBoost Model

4.2.1 Model building

With the same training, validation, and testing sets, a XGBoost model is developed using the steps: (1) one-hot emcoding, (2) define watchlist, (3) set hyper parameters, build model, and tune the parameters. The one-hot em-coding is used to crete dummy variables for the two categorical variables, which was automatically applied in the logistic regression. When tuning the parameters, parameters such as eta (learning rate), max_depth (of the tree), and nrounds were tuned to obtained a balanced high accuracy, AUC, recall, and precision for both training and validation sets. Early.stop.round was also used to prevent overfitting. In addition, scale_pos_weight (0.35*(negative_cases/postive_cases)) is

added and tuned to for the imbalanced class. The result of the final tuning cycle presented below show no obvious overfitting on the training data since the AUC values (selected criteria) are close.

```
[1] train-auc:0.671221 validation-auc:0.671433
Multiple eval metrics are present. Will use validation_auc for early stopping.
Will train until validation_auc hasn't improved in 50 rounds.


[101] train-auc:0.832368 validation-auc:0.832753
[201] train-auc:0.850512 validation-auc:0.849560
[301] train-auc:0.858278 validation-auc:0.857198
[401] train-auc:0.863854 validation-auc:0.863641
[501] train-auc:0.867180 validation-auc:0.866777
[601] train-auc:0.869651 validation-auc:0.868680
[701] train-auc:0.871846 validation-auc:0.869981
[800] train-auc:0.873934 validation-auc:0.870450
```

And the final parameters settings are:

```
eta         max_depth         subsample    colsample_bytree
0.02              2.00              0.95               0.30
min_child_weight    num_parallel_tree     nrounds    early.stop.round
            2.00                 1.00         800                  50
```

Then the final model is used to predict the IND response and the model results on both training and validation sets are shown below. The results are close for both sets indicating the model was not trained properly. And balanced precision and recall values around 0.5 are obtained. Much higher F1 scores around 0.5 than that from logistic regression are received.

```
Confusion Matrix and Statistics for training


          Reference
Prediction     0     1
         0 13666   797
         1   727   810


               Accuracy : 0.9048
            Sensitivity : 0.50404
            Specificity : 0.94949
              Precision : 0.52700
                 Recall : 0.50404
                     F1 : 0.51527
      Balanced Accuracy : 0.72677
                    AUC : 0.8739
```

```
Confusion Matrix and Statistics for validation set:


          Reference
Prediction    0    1
```

```
          0 3417  193
          1  190  200

               Accuracy : 0.9042
             Sensitivity : 0.50891
             Specificity : 0.94732
               Precision : 0.51282
                  Recall : 0.50891
                      F1 : 0.51086
      Balanced Accuracy : 0.72812
                     AUC : 0.8704
```

4.2.2 Model performance

After finalizing the XGBoost hyperparameters, the model is used to predict y repsonse on the testing data for performance report. As seen from the model performance results below for the testing set. The F1 score and AUC are close to them from both training and validation sets. Although the model accuracy dropped slightly compared to the training and validation results. It is as high as 0.87. By comparing the results of both XGBoost and logistic regression below, it is seen the XGBoost algorithm worked better for the imbalanced data, since it improved the sensitivity from 0.26 to 0.66, recall from 0.26 to 0.66, F1 score from 0.35 to 0.50, balanced accuracy from 0.62 to 0.78 and AUC from 0.79 to 0.87. Although the accuracy dropped slightly, the overall performance of the XGBoost improved, especially for predicting the IND class with rare cases.

```
Confusion Matrix and Statistics for the testing set :
            XGBoost                logistic regression
          Reference
Prediction    0    1            Prediction    0    1
         0 4012  169                     0 4382  369
         1  488  331                     1  118  131

         Accuracy : 0.8686               Accuracy : 0.9026
      Sensitivity : 0.6620            Sensitivity : 0.2620
      Specificity : 0.8916            Specificity : 0.9738
        Precision : 0.4042              Precision : 0.5261
           Recall : 0.6620                 Recall : 0.2620
               F1 : 0.5019                     F1 : 0.3498
Balanced Accuracy : 0.7768      Balanced Accuracy : 0.6179
              AUC : 0.8706                    AUC  : 0.7877
```

A presentation of the combination of all XGBoost trees is shown below. From left to right are the roots, internal nodes and leaves. The numbers beside the roots and internal nodes are the "quality" representing their feature importance across all tress. And the numbers beside the leaves show the log-odds of the class 1 for observations ended up in those leaves. So the probability of class 1 for the top leave is: $exp(0.18222)/(1 + exp(0.18222) = 0.545$. Similarly, the probabilities for other leaves can be calculated. The feature importance based on feature gain is also plotted below. It is seen the

features ABC (avg_bal_cards), UOC(uti_open_card), and ND12 (num_acc_30d_past_due_12_months) are the three most important features in this case. The predicted values for the testing set is also shown below. The median value is 0.172, which is higher than that (0.067) from the logistic regression model due to its better performance in predicting the rare class: 1.

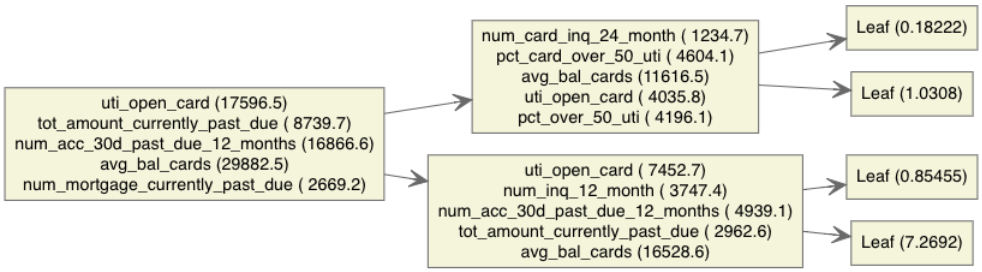

Figure 5: Presentation of the tree combination.

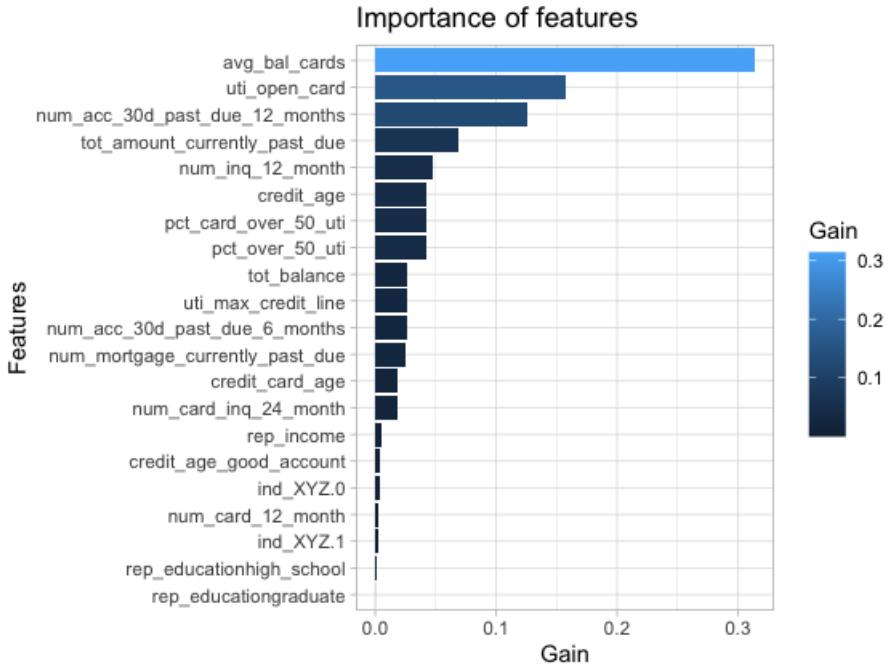

Figure 6: Plot of the feature importance based on Gain.

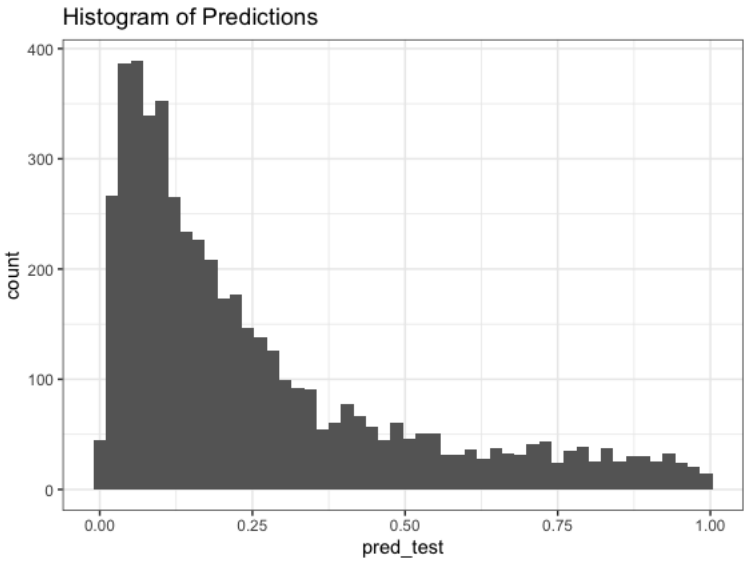

Figure 7: Plot of the predicted probability values.

5 Conclusion

In this report, logistic regression and XGBoost algorithm were used to build binary classification models for predicting the class/probability of defaulted account. The 20 predictors with 18 continuous and 2 categorical were explored. The performance of the XGBoost is better in predicting the class with much lower observations (class 1). Although the accuracy dropped slightly, the overall performance of the XGBoost is better than that of the logistic regression in terms of the AUC, F1 score, recall, and balanced accuracy. For XGBoost, these values are 0.87, 0.50, 0.66, 0.78, respectively. And for the logistic regression model, these values are 0.79, 0.35, 0.26, 0.62, respectively. However, the process of applying logistic regression is much simpler than using XGBoost since the latter one involves tuning many parameters. And the logistic regression model can be better interpreted with the effects/coefficients of each predictor. For example, from the final model shown below, it is known, applicants who already have accounts in the bank are less likely to have defaulted accounts than their counterparts.

$$\begin{aligned} log(\frac{\hat{p}}{1-\hat{p}}) = & -3.02 - 0.00014 * ABC - 0.0046 * CA + 1.08 * ND12 \\ & + 4.53e-01 * ND6 + 0.26 * NI12 + 0.158 * NC12 + 5.29 * UOC \\ & + 0.67 * PO50 - 0.24 IND * I_1. \end{aligned} \tag{5}$$

Where p is the probability of being in class 1 (defaulted account). I_1 is the indicator for the sample in ind_XYZ level 1 (Already having accounts).

Reference

1. https://datascienceplus.com/imputing-missing-data-with-r-mice-package/.
2. https://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/.
3. https://www.statology.org/variance-inflation-factor-r/.

Appendix -Figures

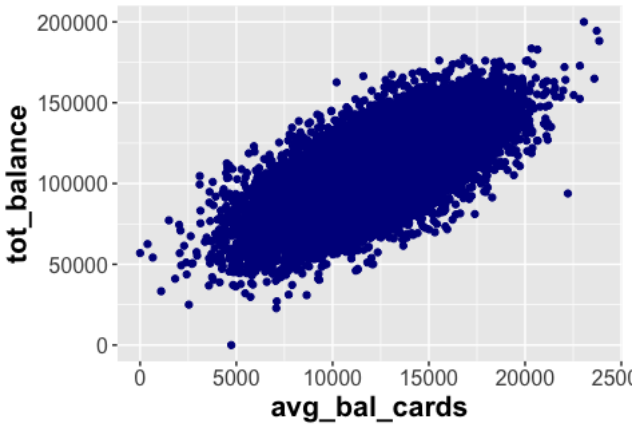

Figure 8: Plot of TB vs ABC.

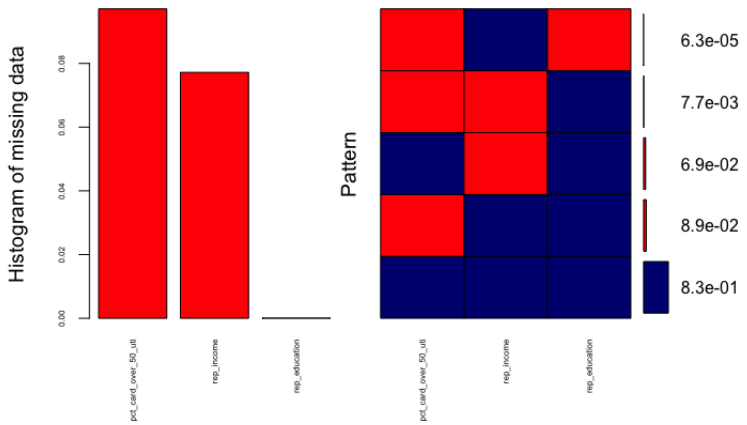

Figure 9: Ratios and distribution of missing data in training set.

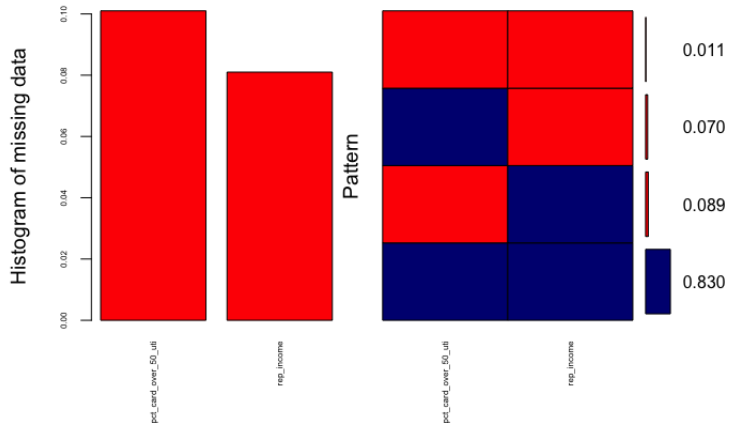

Figure 10: Ratios and distribution of missing data in validation set.

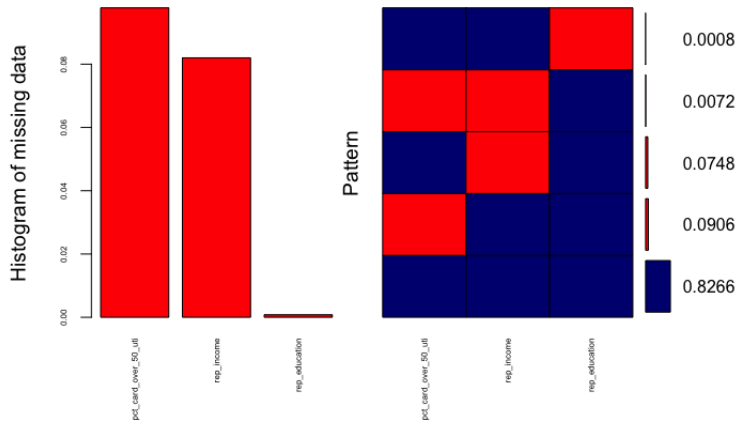

Figure 11: Ratios and distribution of missing data in testing set.