SESUG 2022 Paper ###

A Two-Staged Local Regression Based Binning Method for Weight of Evidence Transformation in Credit Scoring Models

Hui Wang, Shirong Huang, Emma Zhou and Erin Martin, Federal Home Loan Bank of Atlanta

ABSTRACT

Weight of Evidence (WOE) variable transformation method is widely and commonly used in credit risk analysis. This work provides a two-staged, local regression based binning method to estimate WOE. Using the banking industry dataset as an example, this paper shows that when a sufficient number of bins are selected and appropriate smooth factor is chosen, the loss of information and prediction accuracy could be minimized. The proposed method performs well on imbalanced dataset. It can also handle either monotonic or U-shaped relationships between the transformed WOE and original variable by delivering results that have business soundness. The model created with this approach can enable the users to experience more smooth credit score migration when a financial ratio shifts from one bin to another. Considering the widely acknowledged advantages of using WOE method, such as the ease of handling missing values, and the good interpretability of the model after transformation, this approach is considered to have good performance compared to the existing variable transformation approaches and can meet the business needs specifically for credit risk analysis.

INTRODUCTION

In credit risk modeling process, the choice of a proper variable transformation method is always an important topic. Weight of Evidence (WOE) variable transformation method is widely and commonly used in credit risk analysis in the industry. The WOE method can be generalized to the method of transforming each independent variable into its WOE and then fit into the logistic regression framework, which is described as follows:

$$\log(\frac{p}{1-p}) = C + \sum_{i=1}^{j} \beta_i WOE(x_i)$$

where $WOE(x_i)$ is the WOE transformation of variable x_i.

Generally, the WOE method has lots of advantages such as the capability to handle missing values and the variable's non-linearity, as well as to maintain the interpretability of the model after the transformation. However, according to our experience, the performance of the current WOE estimation approaches was not satisfying on extremely imbalanced data. In addition, these algorithms could cause the large change in credit score when customer's financial ratio changes from one bin to adjacent ones.

This work provides a two-staged, local regression based binning method to estimate WOE. Using the banking industry dataset as an example, the paper shows that when sufficient number of bins were selected and appropriate smooth factor was chosen, the loss of information could be minimal. The LOESS smooth WOE method can also handle both monotonic and U-shaped relationship between the WOE and variable by delivering result that have business soundness. Considering the advantages of WOE method in handling the missing value as well as in interpreting the model after transformation, this approach is considered to have good performance among other variable transformation approaches and can serve as an applicable variable transformation method in credit risk modeling analysis.

LITERATURE REVIEW

In this part, the WOE framework is firstly introduced under continuous independent variable environment. The relationship between WOE framework and Naïve Bayes method is discussed. Subsequently, a variety of binning methods for the WOE component are introduced. The advantages and disadvantages of these approaches are reviewed. Finally, the objectives and expected outcome of new binning approach is summarized.

INTRODUCTION OF THE WOE FRAMEWORK

The weight of evidence (WOE) approach is a good framework for variable screening and exploratory analysis for credit risk modeling. WOE and information value (IV) are closely related to concepts from information theory where one of the goals is to understand the uncertainty involved in predicting the outcome of random events given varying degrees of knowledge of other variables (Hughes, 2015; Shannon, 1948; Singer & Kouda, 1999; Wod, 1985).

A credit risk scoring model contains a binary dependent variable Y (default/failure as 1, non-default/non-failure as 0) and a set of predictive variables (x_1, …, x_j). The WOE framework is based on the following relationship:

$$\ln\frac{P(Y=1|x_1,\dots,x_j)}{P(Y=0|x_1,\dots,x_j)} = \ln\frac{P(Y=1)}{P(Y=0)} + \sum_{j=1}^{p}\beta_j \ln\frac{f(x_j|Y=1)}{f(x_j|Y=0)}$$

where $P(Y=1|x_1,\dots,x_j)$ is the conditional probability of Y=1 given the observed set of independent variables (x_1, …, x_j), β_j is a set of coefficient denotes the weight of each independent variables and $\ln\frac{f(x_j|Y=1)}{f(x_j|Y=0)}$ is called the weight of evidence of variable x_j.

It is known that the WOE framework is closely related to the Naïve Bayes method:

$$\ln\frac{P(Y=1|x_1\dots x_j)}{P(Y=0|x_1\dots x_j)} = \ln\frac{P(Y=1)}{P(Y=0)} + \sum_{i=1}^{j}\ln\frac{P(x_i|Y=1)}{P(x_i|Y=0)}$$

The above equation is the Naive Bayes (NB) classifier (Friedman, Hastie, & Tibshirani, 2001). The Naive Bayes model essentially indicates that the logit(p) is equal to the direct summary of the individual weight of evidence variables. The word "naive" comes from the fact that this model relies on the assumption that all predictors are conditionally independent, which is a strong assumption.

The strong assumption of Naive Bayes model that all variables are conditionally independent from each other can be alleviated by various types of methods. A large number of literature proposes approaches to alleviate the conditional independence assumption. Such approaches can be placed into two general categories (Cortizo, Giraldez, & Gaya, 2007; Zaidi, Cerquides, Carman, & Webb, 2013; Zheng & Webb, 2008).

The first category is the Semi-Naive Bayes methods (Zheng & Webb, 2008). These methods are aimed at enhancing Naive Bayes' accuracy by relaxing the assumption of conditional independence between variables. In this category, the Semi-Naive Bayesian methods can be roughly subdivided into several groups (Zheng & Webb, 2008). The first group applies Naive Bayes to a subset of variables generated by deleting variables. The second group adds explicit interdependencies between variables. The third group applies Naive Bayes to a subset of training instances. The fourth group performs adjustments to the output of Naive Bayes without altering its direct operation. The fifth group introduces hidden variables to Naive Bayes.

In our modeling practice, considering the interpretability of the model after adjustment, the first approach of using a subset of variables of the dataset can be used by implementing

variable selection methods such as stepwise or lasso method. In the multivariable analysis part of the modeling process, correlation matrix and clustering analysis on variables can be implemented to reduce the number of strongly correlated variables. Thus the correlation between the final variables that entered into the WOE model could be small.

The second category comprises variable weighting methods (Zaidi et al., 2013). The attribute weighting can be viewed as a means of increasing the influence of highly predictive variables and discounting variables that have little predictive power. There are also various approaches on how to weight each variables (Zaidi et al., 2013). The primary value of variable weighting is its capacity to alleviate the assumption of conditional attribute independence. One of the commonly used weighting method is to set (Zaidi et al., 2013)

$$P(x_1 \dots x_j|Y=1) = \prod_{i=1}^{j} P(x_i|Y=1)^{\beta_i}$$

which add a weight term β_j to each variable's probability density function.

$$\ln\frac{P(Y=1|x_1 \dots x_j)}{P(Y=0|x_1 \dots x_j)} = \ln\frac{P(Y=1)}{P(Y=0)} + \sum_{i=1}^{j} \beta_i \ln\frac{P(x_i|Y=1)}{P(x_i|Y=0)}$$

The above equation is the credit score modeling field commonly used as 'WOE' framework that use logistic regression method with WOE transformation on independent variables. By using this approach, the assumption that all variables in the model are independent is alleviated. The underlying WOEs are still estimated under univariate environment.

Hence, one can use a logistic regression model to estimate the equation, just using a 'WOE' transformation on each independent variables x_j. In the credit scoring industry this "semi-naive" version of model is popular. The idea is to transform the variables into WOE variables and then use logistic regression to fit the model.

THE INFORMATION VALUE

We can leverage WOE to measure the predictive strength of x_j – i.e., how well it helps us to separate cases when Y=1 from cases when Y=0. This can be done through the information value (IV) which is defined by:

$$IV_j = \int \ln\frac{f(x_j|Y=1)}{f(x_j|Y=0)}\left(f(x_j|Y=1) - f(x_j|Y=0)\right) dx$$

Note that the IV is essentially a weighted "sum" of all the individual WOE values where the weights incorporate the absolute difference between the numerator and the denominator (WOE captures the relative difference). Generally, if IV is too small then the variable has very little predictive power and will not add much meaningful predictive power to the model.

ESTIMATION OF WOE USING BINNING APPROACH

The most commonly used approach to estimate the conditional probability density function is to bin x_j and then use a histogram-type estimate, which is also called the binning approach. Let B_1, …, B_j denotes the bins for x_j, the WOE for x_j for bin i can be written as,

$$WOE_{ij} = \ln\frac{P(x_j \in B_i|Y=1)}{P(x_j \in B_i|Y=0)}$$

And the IV for variable x_j can be calculated as,

$$IV_j = \sum_{i=1}^{k} \left(P(x_j \in B_i|Y=1) - P(x_j \in B_i|Y=0)\right) * WOE_{ij}$$

During implementation of WOE transformation on variables, several automatic binning methods can be the candidates, such as equal-width binning, equal-size binning, optimal binning, tree based binning, and monotonic regression binning.

COMMONLY-USED BINNING METHODS

In this part, a brief review for commonly used binning method is given together with our comments on them(Mironchyk & Tchistiakov, 2017).

Equal-width binning and equal-size binning

These two approaches are straightforward approaches for binning (Thomas, Crook, & Edelman, 2017a). For equal-width binning, the user firstly decide the number of bins, then the whole range of predictor values is divided into a pre-specified number of equal-width intervals. Via this practice, each interval defines borders (minimum and maximum) for corresponding bin. The number of bins of the equal-width binning method is predefined. The problem with this method is that, after applying the method, the number of default (Y=1) or non-default (Y=0) in each bin is not determined. There are possibilities that one bin contains too many default observations while lacking of non-default observations, or vice versa. In addition, the business soundness of the transformation is unsupported.

For equal-size binning, the method split the range of predictor values into intervals that each bin contains equal number of observations. The width of bins varies depending on density of observations. The target number of bins of the equal-size binning is also predefined. The problem with this method is similar with the equal-width method. In addition, according to our experiments, when encountered imbalanced data, the first or last bin may not have enough default or non-default observations for an accurate WOE estimation.

Optimal binning and related algorithms

The optimal binning and its related algorithms mainly include the optimal binning (Siddiqi, 2012), multi-interval discretization (Fayyad & Irani, 1993), Chi-merge (Kerber, 1992) and conditional inference tree algorithms (Hothorn, Hornik, Strobl, & Zeileis, 2010; Mironchyk & Tchistiakov, 2017).

Optimal binning algorithm is deemed as an evolution of the previous two algorithms. It can be considered as an enhancement on top of the previous two methods, as in this case a predictor variable is used to define cutoff points for intervals. This algorithm aims to define bins to have sufficiently different statistical mean estimates of predictor value. It consists of the following steps: 1) start with the bins that are small in size (the number of observations in each bin is small) but sufficiently large in the total count of bins (large enough quantity of bins); 2) for each neighboring pair of bins, compute the p-value; 3) find the largest p-value of all pairs. If it is above some threshold, merge corresponding pair of bins then repeat step 1, otherwise exit.

Multi-interval discretization binning is based on entropy minimization heuristic search for recursively splitting of continuous range into sub-intervals, and recursively define the best bins. The purpose of this method is to separate classes based on observation frequencies. This algorithm maximized the test statistics related to entropy to discover the cutoff point for each bin.

Chi-merge algorithm is similar to optimal binning merging. But this method substitutes p-value with Chi-square to test similarity of adjacent bins. The algorithm consists of the following steps: 1) the input range is initialized by splitting it into sub-intervals with each sample getting own interval; 2) for every pair of adjacent sub-intervals a Chi-square value is computed; 3) merge pair with lowest Chi-square into single bin; 4) repeat step 1 and 2 until the maximum Chi-square is less than some predefined threshold.

Conditional inference trees such as multidimensional formulation of algorithm are also based on exhaustive search of partition scheme that would maximize some test statistics.

The common drawbacks of these methods are that, all of these methods optimized some test statistics in the process. In literature, it is said that categorizing variables using optimized test statistics with insufficient number of categories, for example dichotomizing variables (setting bin number to 2) based on optimized test statistics such as p-value, will lead to several problems (Altman & Royston, 2006). Firstly, information is lost, so the statistical power to detect a relationship between the variable and outcome is reduced. Secondly, observations close to but on opposite sides of the cutoff point are characterized as being very different rather than very similar. This will be more critical especially when not enough bins were generated.

It is also noticed that when applying these methods to imbalanced dataset, the results are mostly not desired. Firstly, we observe it from the outputs that not enough bins were generated. Secondly, the output trend of WOE is not smooth & monotonic or smooth & U-shaped, which makes the result to have less business sense. Thus, the new method should be able to handle imbalanced dataset as well as generate output that make business sense.

Monotonic binning approaches

Monotonic binning approaches guarantee a monotonic relationship between the WOE of the variable and variable's value after transformation. It is expected that after applying the binning algorithm, if one walks from one bin to another in the same direction, there is a monotonic change of credit risk indicator. One example of the approaches is provided below. Maximum-likelihood monotone coarse classifier is one example of the monotonic binning approaches (Thomas, Crook, & Edelman, 2017b). It is also known as "Monotone Adjacent Pooling Algorithm". Here is a brief summary about how to implement this method. Assume that bad rate is going down as characteristic value increases. Start at the lowest characteristic value and keep adding values until the cumulative bad rate hits its maximum. This is the first coarse classification split point. Start calculating the cumulative bad rate from this point until it again hits maximum. This is the second split point. Repeat the process until all the split points are obtained (Thomas et al., 2017b).

However, in credit risk modelling process for corporates and firms, it is observed that for some specific category of variables firm's growth, a too low or too high value will increase the firm's default rate. This have business soundness because a lack of growth or too aggressive growth of the firm will result in a higher default risk. Thus, the new binning algorithm should also be able to analyze a variable non-monotonic WOE trend, while maintaining most of the variables' transformation to be monotonic when these variables are supposed to have a monotonic relationship with the default trend.

There are other binning method such as smoothed WOE based on modified WOE definition (Garla, Chakraborty, & Cathie, 2013). Unfortunately, neither of them can generate desired smooth, monotonic or U-shaped WOE result when handling imbalanced dataset.

ADVANTAGES AND CURRENT PROBLEMS OF WOE BINNING ALGORITHMS

The WOE framework has lots of advantages. Generally, WOE can help the modeler to detect linear and non-linear relationships between the independent variables and the response. It can also visualize the correlations between the predictive variables and the binary outcome, providing the capability for users to visually check the economic soundness of the transformation. The method standardizes the scale of each variable, making it easier to compare with each other in term of 'weight' and it can handle missing values without additional steps.

However, there are also problems in the current binning methods to estimate the weight of evidence. Firstly, according to our experiments, most of the algorithms do not perform well

on extremely imbalanced data as discussed above. Secondly, these algorithm either cannot handle the U-shaped WOE variables or the output generated was not smooth thus make no business sense. In such case when the financial ratio migrate from one bin to another, the corresponding credit score will deteriorate or improve towards the opposite direction compared to what user expected. Last, those algorithms generate insufficient number of bins which cause a sudden PD change when the customer's financial ratio jumps from one bin to its adjacent bin.

BINNING ALGORITHM OBJECTIVES

In previous literature, some specific requirements for a binning algorithm were introduced. These requirements can be derived into the following three requirements (Mironchyk & Tchistiakov, 2017; Zeng, 2014):

- Monotonicity: The algorithm should be able to generate WOE that has a monotonic relationship with the original variable. However, in credit risk modelling process for financial corporates, we've observed that for some specific category of variables such as firm's growth, there is a U-shaped relationship and it make business sense.
- Representativeness: The algorithm should reflect maximized correlation between the dependent and independent variable. The loss in either the information or the prediction accuracy should be minimal.
- Constraints: The algorithm should produce number of bins within particular constraints. For example, sufficient number of bins should be generated so that when the variable's value switched from one be to another, not too dramatically PD change will occur. This enables the model user to obtain a smoother PD transition during model usage, which means they will be able to explain the credit score change to customers easily.

PROPOSED BINNING ALGORITHM'S OBJECTIVES

In general, based on the literature review and the discussion above, the objectives of new binning algorithm are:

- Business soundness and monotonicity: The result of the binning algorithm should make business sense. The new binning algorithm should be able to incorporate non-monotonic relationship variable default trend into the model, while maintaining most of the variables' transformation to be monotonic when these variables are supposed to have a monotonic relationship with the default trend.
- Minimal loss of information (representativeness): There should be no significant decrease in prediction accuracy on the transformed variables compared to the original untransformed variable. The information loss from the transformation step should be small.
- Handle imbalanced data: The algorithm should be able to handle imbalanced data.
- Specific requirement: The algorithm should generate sufficient bins to facilitate the model's implementation. The WOE difference between adjacent bins shouldn't be too large. The model should be easy to transfer from statistical software to other environments, for example from SAS or Python to Excel software.

PROPOSED METHOD

Based on the binning algorithm objectives discussed above, an algorithm is proposed to perform the WOE smoothing and calculation. The general process of this algorithm is shown below:

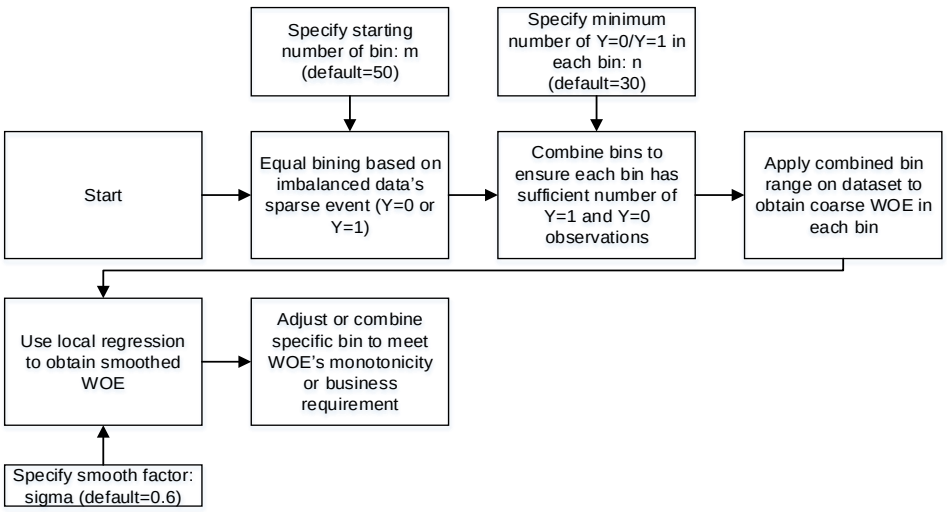

Figure 1. Process of the WOE smoothing algorithm

In this algorithm, the variable in the dataset are transformed by the following procedure:

- Step 1: Based on the training dataset, select the rare event observation first. Since the default event (Y=1) is rare here, select all Y=1 first.
- Step 2: Evenly distribute the Y=1 observations by number to m bins via the ranking of the target predictor. Specifically, the observations were assigned to the designated bin via $\mathrm{floor}(\mathrm{rank} * \mathrm{m}/(\mathrm{n\_def}+1))$, here $\mathrm{n\_def}$ refers to the total number of Y=1 observations in the training data. In the model development, this process was noted as original binning.
- Step 3: Apply the bin range to the training dataset. Obtain each bin's number of Y=1 and number of Y=0.
- Step 4: Starting from the 1st bin, if the specific bin contains Y=1 observations less than specified number n, combine it with the next bin. Otherwise proceed to the next bin. Do this until the last bin. Repeat this process with Y=0 observations. This is to ensure that each bin has number of both Y=1 and Y=0 observations larger than specified number n.
- Step 5: Apply new combined bin's range to obtain coarse WOE in each bin by the equation

$$\mathrm{WOE}_{ij} = \ln \frac{\mathrm{P}(x_j \in B_i | Y=1)}{\mathrm{P}(x_j \in B_i | Y=0)}$$

  where $B_1, \ldots, B_k$ denotes the bins for variable x_j.
- Step 6: Apply local regression on coarse WOE with specified smooth factor to obtain the smoothed WOE of each bin.
- Step 7: Perform visual check and minimal manual adjustment on the bins to ensure the variable's business soundness. When necessary, ensure the transformed variable to have a monotonic relationship between the variable and its WOE by adjusting smooth factor or manually combining bins.

Specifically in the local regression step, according to the SAS supporting document (SAS), the mechanism of the LOESS was:

- Set 'degree' parameter as 1, which means the regression is a local linear fitting.
- Set 'smooth' factor as the specified number, for example 0.6, which means 60% of the observations is used to perform local regression estimates, specifically the number of

points in the local neighborhoods would be $q = \text{floor}(0.6 * k)$, where k is the number of bins (Wicklin, 2016).

- Calling parameter 'direct', which meant the local regression would be done at every point in the input data set.
- Set 'alpha=0.05', which was the significant level for the confidence interval of each fitted data point. The value 0.05 was the default and commonly used.
- Then for each element of the raw WOE_{ij}, found the q nearest neighbors, and denoted their distances to raw WOE_{ij} as $d_1, d_2, \ldots, d_q$, assign their weights as

$$w_i = \frac{32}{5}\left(1 - \left(\frac{d_i}{d_q}\right)^3\right)^3$$

- The q points in the local neighborhood of raw WOE_{ij} are used to fit and score a local weighted regression model at raw WOE_{ij}. The score is the WOE_LOESS value.

In this approach, since the coarse WOE is smoothed, the IV of the smoothed WOE is then revised as:

$$IV_j = \sum_{i=1}^{k}\left(P(x_j \in B_i | Y = 1) - P(x_j \in B_i | Y = 0)\right) * \text{LOESS\_WOE}_{ij}$$

Where coarse weight of evidence "WOE" is replaced by smoothed weight of evidence LOESS_WOE.

DATA AND SETTINGS

In this paper, the data used are the quarterly financial statement data with 46 predictor variables obtained from US banks on a nation-wide basis over 15 years period from 2005 to 2019. The variables are from capital, asset quality, earning, liquidity, size and macroeconomic indicator categories. The default data are collected from FDIC failure list (Federal Deposit Insurance Corporation). The data were cleaned and combined firstly and then divided into 80% training dataset and 20% testing dataset. The training dataset contains 370,634 observations with 2,820 default observations. The testing dataset contains 41,226 observation with 327 default observations.

In the result and discussion part, a minimum of 30 observations for both Y=1 and Y=0 observations in each bin is used. For algorithm and result demonstration, starting number of bins for each variable was selected to be 50 and smooth factor selected to be 0.6. For the discussion of initial bin size, smooth factor was set to be 0.6 while initial bin size varied from 2 to 100. For the discussion of smooth factor, initial bin size was selected to be 50 and smooth factor varied from 0.1 to 0.9.

For the discussion of initial bin size and smooth factor in multivariate environment, a benchmark model was implemented. The 46 variables in the training dataset went through a variable selection process. Firstly, the variables were analyzed under univariate environment and variables with too low AR/IV were discarded. Subsequently the remaining variables were selected under multivariate environment with lasso method. The collinearity between variables were also considered during the process. The VIF for the benchmark model for each variable is less than 1.5. The final benchmark model is guided by the CAMEL framework and contains six variables (Federal Financial Institutions Examination Council, 1996). Two variables are from asset quality category (variable 1 and 10), one variable from capital category (variable 15), one variable from earning category (variable 26), one variable from liquidity category (variable 35) and one variable from macroeconomic indicator category (variable 39).

RESULT AND DISCUSSION

EXAMPLE OF PROPOSED METHOD'S OUTPUT

After applying the proposed algorithm on the dataset, a set of results were obtained and demonstrated below.

Bin	Average_x	Minimum_x	Maximum_x	Cnt_Observation	Cnt_Bad	Cnt_Good	Bad%	WOE_Coarse	WOE_LOESS
0	1346.4	725.7	.	204	169	35	82.84%	6.445	6.401
1	653.5	607.5	725.7	106	56	50	52.83%	4.984	5.139
2	569.0	529.3	607.5	126	57	69	45.24%	4.680	5.058
3	503.9	481.6	529.3	97	56	41	57.73%	5.183	4.980
4	458.6	440.4	481.6	104	56	48	53.85%	5.025	4.928
5	425.8	412.0	440.4	101	57	44	56.44%	5.130	4.902
6	398.9	389.4	412.0	102	56	46	54.90%	5.068	4.875
				.....					
40	52.6	49.2	56.5	4829	57	4772	1.18%	0.443	0.039
41	46.1	43.3	49.2	5653	56	5597	0.99%	0.266	-0.228
42	38.8	35.0	43.3	11442	56	11386	0.49%	-0.444	-0.528
43	31.5	28.5	43.3	14336	57	14279	0.40%	-0.653	-0.834
44	24.6	21.3	28.5	24853	56	24797	0.23%	-1.222	-1.127
45	18.5	16.0	21.3	29492	57	29435	0.19%	-1.376	-1.386
46	11.5	7.9	16.0	79596	56	79540	0.07%	-2.388	-1.684
47	3.0	.	7.9	179105	56	179049	0.03%	-3.199	-2.055

Table 1. Example of final output table on a single variable of proposed WOE smoothing algorithm

Table 1 shows the final output of the WOE smoothing algorithm on one single variable (Texas ratio in this example). In the table, typical results that other WOE binning algorithms reported were provided. Those results include bin number, minimum, maximum and average value of independent variable x in each bin, count of total observation, Y=1 observation and Y=0 observation in each bin, bad ratio and raw WOE. Additionally, the smoothed WOE value of each bin is also reported in this table.

The algorithm will automatically analyze all the independent variables in the dataset. For each analyzed variable, the results in Table 1 and Table 2 will be reported. In addition, accuracy ratio (AR) will be reported for both coarse and smoothed transformation of the variable. IV and LOESS modified IV will also be provided for coarse and smoothed WOE transformed variable. This will help the users to determine whether the smoothing procedure cause information loss. In addition, whether the transformed variable is monotonic is also reported in the output.

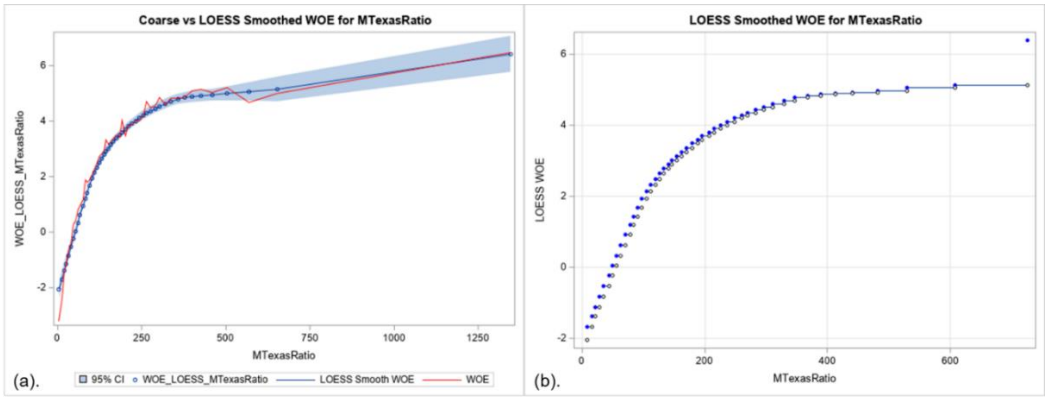

Figure 2. (a). Smoothed WOE vs. coarse WOE. (b). Final smoothed WOE example

Figure 2 shows the variable transformation output graph. Figure 2(a) shows the comparison between smoothed WOE and coarse WOE. Figure 2(b) shows the final transformation result. These graphs can help the user to determine whether the smoothed WOE has significant deviation from coarse WOE. It can also help the users to determine whether the transformation has business soundness.

From the results, for all 46 variables, it is observed that after applying the algorithm, more than 90% of the transformed variables have already met its desired monotonicity/non-monotonicity properties. And the rest of the variables only require minimal manual adjustment on the bins, typically combining 1-2 bins.

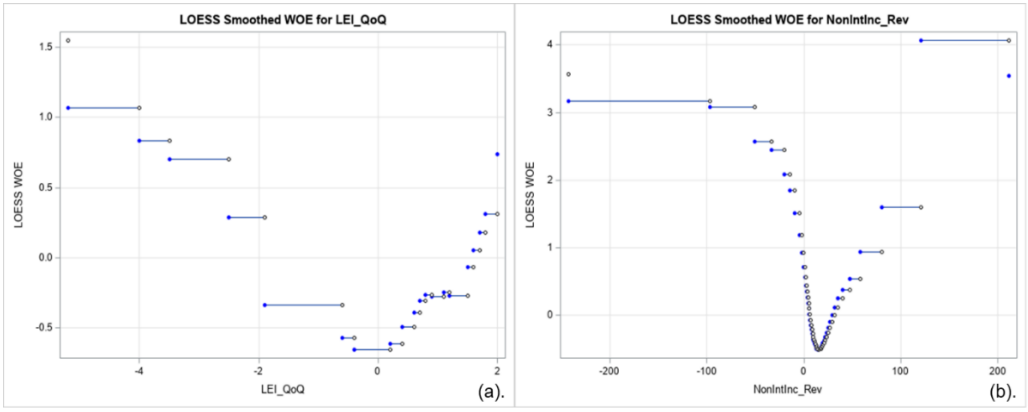

Figure 3. (a) (b). Example of non-linear relationship between independent variable and WOE generated by proposed WOE smoothing algorithm

The algorithm is also capable of handling the variables that has non-monotonic WOE relationships. As can be seen from Figure 3(a) (b), the non-linear relationship on the WOE is correctly captured. Such relationship could make business sense. For example, Leading-Economic-Index quarter over quarter (LEI_QoQ) could have non-linear WOE relationship because when the economy shrinks or expands too fast, the default risk may increase.

Table 2 shows the summary of the analysis result of all variables on the training dataset. AR and IV of coarse and smoothed WOE were reported. It can be seen that most variables have comparable AR or IV between coarse and smoothed WOE, indicating no significant information loss on the smoothing step.

Var	Name	Category	AR	AR_LOESS	IV	IV_LOESS
1	Variable 1	Asset Quality	90.77%	90.76%	5.19	4.41
2	Variable 2	Asset Quality	87.69%	87.67%	4.27	3.82
3	Variable 3	Asset Quality	86.15%	86.11%	3.94	3.57
4	Variable 4	Asset Quality	86.65%	86.63%	4.05	3.73
5	Variable 5	Asset Quality	85.97%	85.81%	3.98	3.69
6	Variable 6	Asset Quality	72.17%	72.10%	2.31	1.92
7	Variable 7	Asset Quality	75.95%	75.90%	2.49	2.40
8	Variable 8	Asset Quality	70.08%	69.66%	2.08	2.00
9	Variable 9	Asset Quality	66.35%	66.08%	1.84	1.79
10	Variable 10	Asset Quality	62.56%	62.36%	1.55	1.52
11	Variable 11	Asset Quality	21.14%	12.50%	0.21	0.22
12	Variable 12	Asset Quality	67.06%	65.35%	2.02	1.86
13	Variable 13	Asset Quality	65.71%	62.42%	1.89	1.59
14	Variable 14	Asset Quality	52.75%	51.34%	0.99	0.91
15	Variable 15	Capital	89.31%	89.29%	5.22	5.37
16	Variable 16	Capital	71.27%	71.16%	2.73	2.74
17	Variable 17	Capital	83.38%	83.36%	4.14	4.27
18	Variable 18	Capital	83.23%	83.22%	4.30	4.36
19	Variable 19	Capital	83.26%	83.23%	4.12	4.23
20	Variable 20	Capital	82.63%	82.59%	4.06	4.14
21	Variable 21	Capital	85.23%	85.20%	4.61	4.73
22	Variable 22	Capital	83.58%	83.57%	4.32	4.43
23	Variable 23	Capital	82.84%	82.78%	4.23	4.31
24	Variable 24	Earnings	31.58%	30.00%	0.37	0.34
25	Variable 25	Earnings	87.48%	87.19%	4.33	3.89
26	Variable 26	Earnings	87.42%	87.24%	4.36	4.03
27	Variable 27	Earnings	86.81%	86.53%	4.23	3.96
28	Variable 28	Earnings	76.32%	72.22%	2.77	2.34
29	Variable 29	Earnings	38.45%	37.58%	0.80	0.73
30	Variable 30	Earnings	49.31%	48.74%	0.99	0.93
31	Variable 31	Earnings	62.02%	61.71%	1.62	1.57
32	Variable 32	Earnings	15.60%	13.80%	0.08	0.06
33	Variable 33	Liquidity	18.80%	14.72%	0.12	0.06
34	Variable 34	Liquidity	46.04%	44.82%	0.86	0.85
35	Variable 35	Liquidity	36.98%	35.49%	0.62	0.61
36	Variable 36	Liquidity	32.71%	30.05%	0.47	0.45
37	Variable 37	Size	18.93%	16.05%	0.11	0.08
38	Variable 38	Size	18.93%	16.28%	0.11	0.09
39	Variable 39	Macroeconomic Indicator	61.39%	60.00%	1.62	1.47
40	Variable 40	Macroeconomic Indicator	47.71%	36.52%	0.80	0.40
41	Variable 41	Macroeconomic Indicator	54.17%	43.40%	1.13	0.90
42	Variable 42	Macroeconomic Indicator	57.79%	54.71%	1.43	1.10
43	Variable 43	Macroeconomic Indicator	48.89%	41.06%	0.89	0.69
44	Variable 44	Macroeconomic Indicator	42.48%	24.42%	0.61	0.23
45	Variable 45	Macroeconomic Indicator	55.65%	49.54%	1.16	0.98
46	Variable 46	Macroeconomic Indicator	36.88%	32.62%	0.49	0.36

Table 2. AR and IV of coarse and smoothed WOE on training dataset

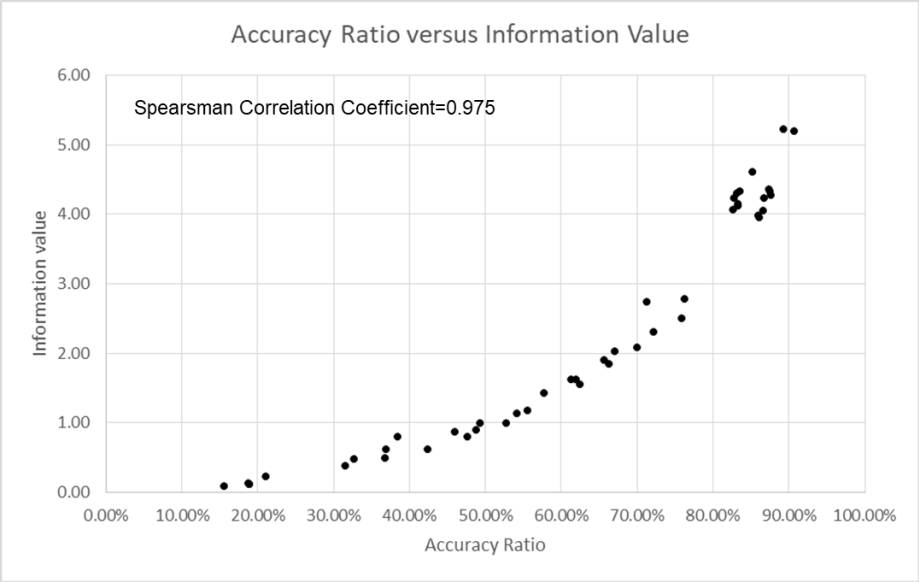

Figure 4. Relationship between IV and AR under univariate condition

It is known that both AR and IV can be used to measure prediction accuracy and information loss of the variables. However, IV is derived under univariate environment and AR can be used in both univariate and multivariate environment. In order to compare the performance of these two metrics, the relationship between IV and AR of variables was plotted as shown in Figure 4. From the result, it can be seen that there is an approximately non-linear monotonic relationship between IV and AR. This is also confirmed by calculating the Spearman correlation coefficient to be 0.975, which shows strong relationship between these two metrics. This indicates that in most cases when IV is high, AR will be high too. For convenience purpose, AR will be used as the metric for prediction accuracy in the following discussion.

DEFAULT EVENTS IN EACH BIN

For estimating WOE precisely, minimal of 5 default events in each bin is commonly required for each bin. Too few default events in each bin will cause the estimation of WOE to be too coarse, resulting in difficulties in smoothing the WOE later. Too many default events in each bin will decrease the number of bins that are used in the variable transformation process. Thus, the WOE difference between different bins will be relatively large. In the model implementation stage, it will cause the probability of default (PD) to jump too dramatically when a variable's value change from one bin to the adjacent bin, which means a small change in variable value could possibly cause relatively large PD change.

In this model development, since a single default event will result in a maximum number of 6 default observations, 30 default and non-default observations were set as minimum requirement for each bin. Similar value on this criteria can also been seen on different fields in literatures (Boston University School of Public Health, 2016). And considering that particularly for the training dataset used in this paper, which has 2,820 default observations, this is a sufficient small number to start with. This will result in approximately 100 bins to be the maximum bins allowed for each variable, which is a sufficient large bin number.

INITIAL NUMBER OF BINS

Figure 5(a) shows the AR of the benchmark model with different initial number of bins from 2-100. Since the dataset is imbalanced, additional metric is provided to measure the prediction accuracy of the benchmark model. False positive rate and false negative rate

(FPR and FNR, or Type I and Type II error rate) can be used in this environment. Figure 1(b) shows the total error rate (FPR+FNR) when varying the initial bin number. It is observed that, when the number of bins is less than 6, the information started to lose significantly. Thus, to avoid losing information, it is considered the number of bins should be larger than 10-20. However, it is also considered that the number of bins shouldn't be excessively large, because when too many bins were assigned, some bins won't have default (Y=1) observations. This will create missing value in certain bins. Consecutive missing value in bins will bring problems to the implementation of the WOE smoothing algorithms. It also should be noticed that when the number of bins was set to a proper value (20-50 for example), the training or testing AR is slightly higher than the original model without transformation and smoothing. One possible explanation is that the noise in the dataset is smoothed during the WOE transformation, creating a better and clear illustration on the relationship between variable and response. In this paper, a number of 50 bins is considered appropriate to ensure minimal information loss and thus is used in the following analysis.

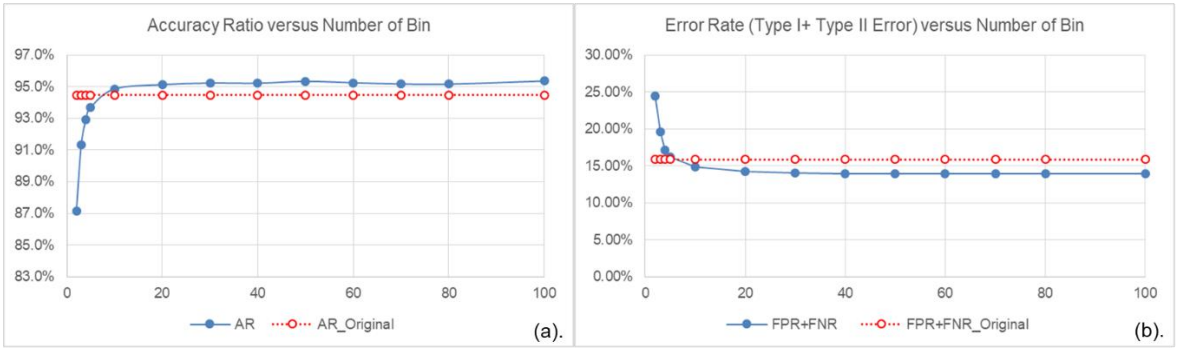

Figure 5. AR and total error rate (false positive rate + false negative rate) of model based on different starting number of bins

SMOOTH FACTOR

Smooth factor between 0.1-0.9 were applied to obtain the results. Figure 6 shows the coarse WOE versus smoothed WOE with respect to different smoothing factor. In the figure, the red line shows the coarse WOE and the blue line shows the smoothed WOE. As can be seen in Figure 6(b)-(d), when applying the smooth factor between 0.3-0.7, the noise within the coarse WOE is effectively smoothed and the smoothed WOE can effectively reflect the true value of the coarse WOE, showing no significant deviance from the true WOE value.

Applying a high value of smooth factor will cause the variable's WOE being smoothed too much and potentially lose significant relationship characteristics to the coarse ones in the variable. For example, when applying factor value of 0.9 or even larger to the WOE of independent variable x, the relationship between x and its WOE will be close to a straight line. On the other hand, applying a small factor will not smooth the coarse WOE enough. The smoothed WOE will still have too much noise and will not be monotonic or have business soundness. It will also possibly cause the overfitting problem because the WOE are trying to fit the detailed features of the training data, instead of capturing a general default trend. This will also bring difficulties to the model implementation process since the probability of default change doesn't make business sense when the independent variable value changes. In this work, a smooth factor of 0.6 is deemed to be able to obtain a desirable result.

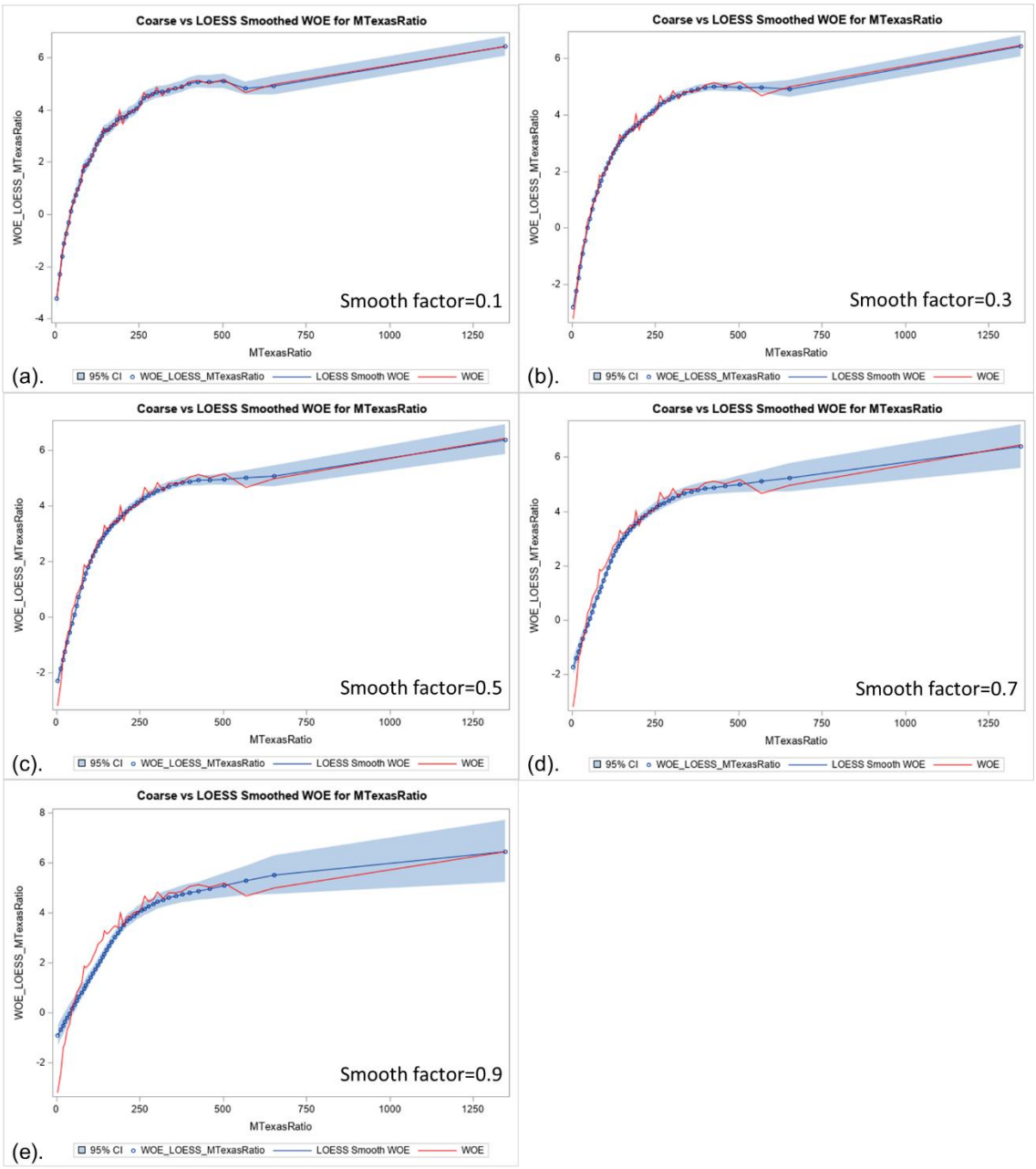

Figure 6. (a)-(e). WOE smoothing result with smooth factor of 0.1, 0.3, 0.5, 0.7, 0.9, respectively (using Texas ratio as example).

In addition, the effect of applying smoothing LOESS regression on model prediction accuracy is also discussed. Figure 7(a) shows the AR of the benchmark model before and after smoothing on training dataset. No significant AR drop was observed from the result. Figure 7 shows the total error rate, which is false positive rate + false negative rate (FPR+FNR) before and after applying the LOESS regression. No significant increase of total error rate was observed. The same analysis was also performed on the testing data. Similar result was obtained for the testing dataset, suggesting no significant overfitting problem after smoothing. It indicates that the algorithm is successful and the loss of information is minimal.

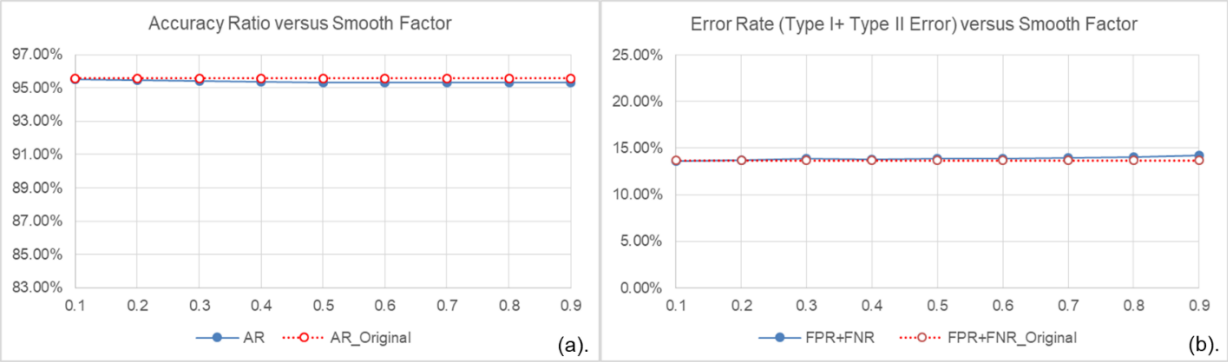

Figure 7. AR and total error rate (false positive rate + false negative rate) of model based on different smooth factor

EXAMINE OF THE TRANSFORMATION NECESSARY MONOTONIC RELATIONSHIP AND BUSINESS SOUNDNESS

After transformation, the WOE of variables are examined graphically to check its business soundness. For example, from business perspective, the probability of default should increase as Texas ratio (used in Figure 2 and Figure 6) increase because the higher the Texas ratio is the worse the asset quality will be. In addition, the marginal effect of Texas ratio on increasing the probability of default decreases when Texas ratio is large enough. This is also consistent with the business judgement. Figure 2(b) shows an example for the transformed variables. The relationship between original variable and transformed variables (WOE) is clearly illustrated.

CONCLUSION

This work provides a two-staged, local regression based binning method to estimate WOE. A banking industry dataset was used as an example to show that when initial number of bins were selected to be appropriately large (>20) and appropriate smooth factor was chosen (0.3-0.7), the loss of information and prediction accuracy could be minimal. The LOESS smooth WOE method can also handle both monotonic and U-shaped relationship between the WOE and variable by delivering result that have business soundness. The proposed approach can serve as a variable transformation method in credit risk modeling analysis for both variable screening and actual production model variable transformation purpose.

From the discussion above, it is known that the binning approach actually used a step function to estimate WOE. In the future, the continuous functions can also be applied to estimate WOE. This will fully eliminate the credit score jumping problem at bin boundary caused by the binning method. One idea could be using a piecewise linear function to "connect each bin's WOE" after the applying the current method. Another idea could be directly using a continuous function to estimate WOE. However, the choosing of function family and the necessary steps to make the transformation having business sense remains as a problem that needs further study.

REFERENCES

Altman, D. G., & Royston, P. (2006). The cost of dichotomising continuous variables. Bmj, 332(7549), 1080.

Boston University School of Public Health. (2016). Tests of Single Proportions. Retrieved from https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R6_CategoricalDataAnalysis/R6_CategoricalDataAnalysis5.html

Cortizo, J. C., Giraldez, I., & Gaya, M. C. (2007). Wrapping the naive bayes classifier to relax the effect of dependences. Paper presented at the International Conference on Intelligent Data Engineering and Automated Learning.
Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning.
Federal Deposit Insurance Corporation. Failed Bank List. Retrieved from https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/
Federal Financial Institutions Examination Council. (1996). Uniform financial institutions rating system. Federal Register, 61(245), 67021-67029.
Friedman, J., Hastie, T., & Tibshirani, R. (2001). 6. Kernel Smoothing Methods The elements of statistical learning (Vol. 1, pp. 211): Springer series in statistics New York, NY, USA.
Garla, S., Chakraborty, G., & Cathie, A. (2013). Extension Node to the Rescue of the Curse of Dimensionality via Weight of Evidence (WOE) Recoding. Paper presented at the SAS Global Forum.
Hothorn, T., Hornik, K., Strobl, C., & Zeileis, A. (2010). Party: A laboratory for recursive partytioning.
Hughes, G. (2015). Youden's Index and the Weight of Evidence. Methods Inf Med, 54(2), 198-199.
Kerber, R. (1992). Chimerge: Discretization of numeric attributes. Paper presented at the Proceedings of the tenth national conference on Artificial intelligence.
Mironchyk, P., & Tchistiakov, V. (2017). Monotone optimal binning algorithm for credit risk modeling. Электронный ресурс, 1-15.
SAS. The LOESS Procedure. Retrieved from https://support.sas.com/documentation/onlinedoc/stat/131/loess.pdf
Shannon, C. E. (1948). A mathematical theory of communication. Bell system technical journal, 27(3), 379-423.
Siddiqi, N. (2012). Credit risk scorecards: developing and implementing intelligent credit scoring (Vol. 3): John Wiley & Sons.
Singer, D. A., & Kouda, R. (1999). A comparison of the weights-of-evidence method and probabilistic neural networks. Natural Resources Research, 8(4), 287-298.
Thomas, L., Crook, J., & Edelman, D. (2017a). Credit scoring and its applications: SIAM.
Thomas, L., Crook, J., & Edelman, D. (2017b). Maximum likelihood monotone coarse classifier Credit scoring and its applications (pp. 138): SIAM.
Wicklin, R. (2016). What is loess regression? Retrieved from https://blogs.sas.com/content/iml/2016/10/17/what-is-loess-regression.html
Wod, I. (1985). Weight of evidence: A brief survey. Bayesian statistics, 2, 249-270.
Zaidi, N. A., Cerquides, J., Carman, M. J., & Webb, G. I. (2013). Alleviating naive Bayes attribute independence assumption by attribute weighting. The Journal of Machine Learning Research, 14(1), 1947-1988.
Zeng, G. (2014). A necessary condition for a good binning algorithm in credit scoring. Applied Mathematical Sciences, 8(65), 3229-3242.
Zheng, F., & Webb, G. I. (2008). Semi-naive Bayesian classification: Monash University.

CONTACT INFORMATION

Hui Wang
Federal Home Loan Bank of Atlanta
404-888-5628
hwang@fhlbatl.com

CODE

Before running the following program, please save the two SAS code into directory first. Appendix 2 contains the macros that combine bins according to parameter setting and the

macro that check the monotonicity of WOE. Please load Appendix 2 code first before running the main code in Appendix 1.

APPENDIX 1

Code in Appendix 1 © 2022 Federal Home Loan Bank of Atlanta, All Rights Reserved

```
exclude none;

%let nvar=%sysfunc(countw(&var_list));

proc sql noprint;
select TableVar
into :var_1-:var_&nvar
from checkfreq; quit;

%macro Gen_Group;
%do i = 1 %to &nvar;
%Global group_&i.;
%let group_&i.=%qscan(%bquote(&group_list),&i);
%end;
%mend Gen_Group;
%Gen_Group;

/*****Calculation of WOE*****/
%Macro DOLoop;
%DO i=1 %to &nvar;

%let var_x=&&var_&i.;
%let j=%eval(&i.+1);

/*Output default training dataset*/
data newtrain_def_&i.;
set newtrain_&i.;
if &Dep_Var=1 then output;
run;

/*Sort and group observation by varible x_i*/
PROC RANK DATA=newtrain_def_&i. groups=&&group_&i. out = newtrain_def_&j.
Ties = low descending;
var &var_x.;
ranks Rnk_&var_x.;
run;

PROC MEANS data=newtrain_def_&j. noprint nway;
Class Rnk_&var_x.;
output out=range min(&var_x.)=min_&var_x. max(&var_x.)=max_&var_x.;
run;

/*Making the upper bound of each bin connected to each other*/
data range;
set range;
max2_&var_x.=lag(min_&var_x.);
run;
```

```
/*If the bin is the last (max) bin then upper bound changed to
99999999999999*/
data range;
set range;
if max2_&var_x.=. then max2_&var_x.=99999999999999;
run;

/*Update first bin minimum value (Y=1's lower limit not necessary to be
dataset lower limit)*/
proc means data=work.range noprint min;
var min_&var_x.;
output out=minrange min=minvar;
run;

data _Null_;
set minrange;
call symput('minvar',minvar);
run;

data range;
set range;
if min_&var_x. EQ &minvar then min_&var_x.=-99999999999999;
run;

data range;
set range;
Rnk_2_&var_x.=_N_-1;
run;

/*Output for bin range before bin combination*/
proc sql;
create table newtrain_&j. as select newtrain_&i..*,range.Rnk_2_&var_x. from
newtrain_&i.
join range
on newtrain_&i..&var_x. >= range.min_&var_x. and newtrain_&i..&var_x. <
range.max2_&var_x.;
quit;

proc sql;
create table newtest_&j. as select newtest_&i..*,range.Rnk_2_&var_x. from
newtest_&i.
join range
on newtest_&i..&var_x. >= range.min_&var_x. and newtest_&i..&var_x. <
range.max2_&var_x.;
quit;

/*obtain average x, average y (&Dep_Var), average PD for each group*/
PROC MEANS data=newtrain_&j. noprint nway;
Class Rnk_2_&var_x.;
var &Dep_Var;
output out=Result_Table mean(&var_x.)=avg_&var_x. sum(&Dep_Var)=num_def
N=num_obs;
run;

/*Displaying the raw result*/
data Result_Table_display;
retain Rnk_2_&var_x. avg_&var_x. num_obs num_def;
```

```
set Result_Table;
rename Rnk_2_&var_x.=Raw_Bin_Rank_&var_x. avg_&var_x.=Average_&var_x.
num_obs=Num_Observation num_def=Num_Bad;
drop _FREQ_ _type_;
run;

title "Original Bin before Combine: &var_x.";
proc print data=Result_Table_display;
run;
title;

/*----------Call Combine Bin Macro-----------------*/
%Combinebin(input_data=range,var_x=&var_x ,Min_Bin_Obs=&Min_Bin_Num_Obs);

/*----------Regenerate Result_Table----------------*/
/*reinitiate variable*/
data range;
set range_out;
run;

proc sql;
create table newtrain_&j. as select newtrain_&i..*,range.Rnk_2_&var_x. from
newtrain_&i.
join range
on newtrain_&i..&var_x. >= range.min_&var_x. and newtrain_&i..&var_x. <
range.max2_&var_x.;
quit;

proc sql;
create table newtest_&j. as select newtest_&i..*,range.Rnk_2_&var_x. from
newtest_&i.
join range
on newtest_&i..&var_x. >= range.min_&var_x. and newtest_&i..&var_x. <
range.max2_&var_x.;
quit;

/*obtain average x, average y (&Dep_Var), average PD for each group*/
PROC MEANS data=newtrain_&j. noprint nway;
Class Rnk_2_&var_x.;
var &Dep_Var;
output out=Result_Table mean(&var_x.)=avg_&var_x. sum(&Dep_Var)=num_def
N=num_obs;
run;

/*----------End of Regenerate Result_Table--------------*/
/*Calculate WOE*/
data Result_Table_out;
set Result_Table;
/*treat computation problem, if phat=0 then WOE will be nonmeaningful, thus
replace 0 with a small number, 0.0001 (0.01% PD) */
if num_def=0 then WOE_&var_x.=.;
else WOE_&var_x.=-1*log(((num_obs-
num_def)/&num_tot_nondef)/(num_def/&num_tot_def));
run;

/*Output 1, join table 2&3*/
proc sql;
create table Output_1 as select * from Result_Table_out
```

```
left join (select badrate, min_&var_x., max2_&var_x., goods from range_out)
on Result_Table_out.Rnk_2_&var_x.=range_out.Rnk_2_&var_x. order by
Rnk_2_&var_x.;
quit;

/*Obtain loess smoothed WOE*/
ods exclude all;
proc loess data=Result_Table_out;
model WOE_&var_x. = avg_&var_x. / degree=1 smooth = &Smooth_F
direct alpha=.05 all details;/*95% CI*/
ods output OutputStatistics = LOESSResult;
run;
/*select=AICC(steps); (Used for determining smooth parameter of 0.6)*/
ods exclude none;

data LOESSResult;
set LOESSResult;
WOE_LOESS_&var_x.=Pred;
drop Pred;
run;

/*Merge dataset and smoothed WOE*/
proc sql;
create table variableplot as select * from LOESSResult
left join (select WOE_&var_x. from Result_Table_out) on
Result_Table_out.avg_&var_x.=LOESSResult.avg_&var_x. order by obs;
quit;

/*Output 2*/
proc sql;
create table Output_&var_x. as select * from Output_1
left join (select obs, LowerCL, UpperCL, WOE_LOESS_&var_x. from variableplot)
on Output_1.avg_&var_x.=variableplot.avg_&var_x. order by Rnk_2_&var_x.;
quit;

/*Display the final result*/
data data.Output_&var_x._display;
retain Rnk_2_&var_x. avg_&var_x. min_&var_x. max2_&var_x. num_obs num_def
goods badrate WOE_&var_x. WOE_LOESS_&var_x. LowerCL UpperCL;
set Output_&var_x.;
drop _FREQ_ _TYPE_ LowerCL UpperCL;
rename Rnk_2_&var_x.=Combined_Bin_Rank_&var_x. avg_&var_x.=Average_&var_x.
min_&var_x.=Minimum_&var_x. max2_&var_x.=Maximum_&var_x.
num_obs=Num_Observation num_def=Num_Bad goods=Num_Good badrate=Bad_Rate
WOE_&var_x.=WOE_Raw_&var_x.;
run;

title "LOESS Smoothed WOE Result Summary: &var_x.";
proc print data=data.Output_&var_x._display;
run;
title;

/*Obtain IV and loess IV*/
data Output_&var_x._display_2;
set data.Output_&var_x._display;
Pre_IV_&var_x.= -1*((Num_Good/&num_tot_nondef)-(Num_Bad/&num_tot_def))
*WOE_Raw_&var_x.;
Pre_IV_LOESS_&var_x.= -1*((Num_Good/&num_tot_nondef)-(Num_Bad/&num_tot_def))
```

```
*WOE_LOESS_&var_x.;
run;

proc sql;
create table iv_&var_x. as select sum(Pre_IV_&var_x.) as iv,
sum(Pre_IV_LOESS_&var_x.) as iv_LOESS from Output_&var_x._display_2; quit;

data _null_;
set iv_&var_x.;
call symputx('IV_raw', iv);
call symputx('IV_LOESS', iv_LOESS);
run;

data Output_&var_x._graph;
set Output_&var_x.;
if min_&var_x.=-99999999999999 then min_&var_x.=.;
if max2_&var_x.=99999999999999 then max2_&var_x.=.;
run;

/*Plot Coarse and LOESS Smoothed WOE*/
title "Coarse vs LOESS Smoothed WOE for &var_x.";
proc sgplot data=Output_&var_x._graph;
band x=avg_&var_x. lower=LowerCL upper=UpperCL /legendlabel="95%
CI";/*Confidence band*/
scatter x=avg_&var_x. y=WOE_LOESS_&var_x./ markerattrs=(size=5px
);/*Scatterplot*/
series x=avg_&var_x. y=WOE_LOESS_&var_x./ legendlabel="LOESS Smooth
WOE";/*Smoothed scatterplot*/
series x=avg_&var_x. y=WOE_&var_x./ legendlabel="WOE" lineattrs=(Color=Red);
run;
title;

/*LOESS Smoothed WOE*/
title "LOESS Smoothed WOE for &var_x.";
proc sgplot data=Output_&var_x._graph noautolegend;
   vector x=max2_&var_x. y=WOE_LOESS_&var_x. / xorigin=min_&var_x.
yorigin=WOE_LOESS_&var_x. noarrowheads;
   scatter x=min_&var_x. y=WOE_LOESS_&var_x. /
markerattrs=(symbol=CircleFilled color=blue size=5px);      /* closed */
   scatter x=max2_&var_x. y=WOE_LOESS_&var_x. / filledoutlinedmarkers
markerfillattrs=(color=white) /* open */
    markerattrs=(symbol=CircleFilled color=blue size=5px);
   xaxis grid label="&var_x.";
   yaxis grid label="LOESS WOE";
run;
title;

%Check_Monotonic(input_data=data.Output_&var_x._display,var_x=&var_x.);

data var_merge;
set Output_&var_x.;
rnk=Obs-1;
run;

proc sql;
create table newtrain_&j. as select * from newtrain_&j
left join (select WOE_&var_x. , WOE_LOESS_&var_x. from var_merge) on
newtrain_&j..Rnk_2_&var_x.=var_merge.rnk /*order by obs*/;
```

```
quit;

proc sql;
create table newtest_&j. as select * from newtest_&j
left join (select WOE_&var_x. , WOE_LOESS_&var_x. from var_merge) on
newtest_&j..Rnk_2_&var_x.=var_merge.rnk /*order by obs*/;
quit;

/*Accuracy Ratio*/
ods exclude all;
proc logistic data=newtrain_&j.;
Model &Dep_Var(event='1')=WOE_LOESS_&var_x./firth clparm=wald clodds=pl;
Roc;
ODS output ROCAssociation=AR_&var_x.;
run;
ods exclude none;

data AR_&var_x.;
set AR_&var_x.;
if ROCModel='ROC1' then delete;
run;

proc sql noprint;
select Area, SomersD
 into :AUC_out separated by ',',
 :AR_out separated by ','
 from AR_&var_x.;
quit;

title "Summary for Analyis of Variable: &var_x.";
proc odstext;
p "&var_x. is: &Mono_or_not";
p "Area Under Curve of &var_x.: &AUC_out";
p "Accuracy Ratio of &var_x.: &AR_out";
p "IV of &var_x. is: &IV_raw";
p "LOESS IV of &var_x. is: &IV_LOESS";
run;
title;

/*Output final dataset*/
%if &i=&nvar %then %do;
data data.new_training_final;
set newtrain_&j.;
run;

data data.new_testing_final;
set newtest_&j.;
run;
%end;

%end;
%mend DOLoop;
%DOLoop;
```

APPENDIX 2

Code in Appendix 2, © 2022 Federal Home Loan Bank of Atlanta, All Rights Reserved

```
%Macro Check_Monotonic(input_data=,var_x=);
```

```
proc sort data=&input_data out=try;
by WOE_LOESS_&var_x.;
run;

data try2;
set try;
diff=dif(Combined_Bin_Rank_&var_x.);
If diff = . then delete;
run;

proc sql noprint;
   create table try3 as
   select distinct diff
   from try2;
quit;

proc sql noprint;
create table try4 as
 select count(*) as nrow from try3;
quit;

data try5;
set try4;
if nrow eq 1 then WOE_LOESS_&var_x.='Monotonic';
else WOE_LOESS_&var_x.='Non-Monotonic';
drop nrow;
run;

%Global Mono_or_not;

data _null_;
set try5;
call symputx('Mono_or_not', WOE_LOESS_&var_x.);
run;

%mend Check_Monotonic(input_data=,var_x=);

%Macro CombineBin(input_data=,var_x=,Min_Bin_Obs=);

%let BinNumLim=&Min_Bin_Obs;
%let varx=&var_x;

proc sql;
create table table as select * from &input_data
left join (select num_obs from Result_Table) on
&input_data..Rnk_2_&varx.=Result_Table.Rnk_2_&varx. /*order by obs*/;
quit;

data table;
set table;
keep min_&varx max2_&varx _freq_ num_obs;
run;

data table;
set table;
Badrate=9999;
WoE=9999;
run;
```

```
data table;
set table;
rename min_&varx=Lower;
rename max2_&varx=Upper;
rename _freq_=Bads;
rename num_obs=Freq;
run;

proc sort data = table;
by Lower;
run;

/*******************************************/
/*   y=1 (default, combine bin < &BinNumLim) */
/*******************************************/

Data table1;
set table;
If bads<&BinNumLim then lt30=1;
else lt30=0;
retain sum_lt30;
sum_lt30+lt30;
lower=lag(upper);
run;

data _Null_;
Set table1;
call symputx('m',sum_lt30);
run;

%put &=m;

data table;
set table1;
drop lt30
sum_lt30;
run;

%do %while(&m NE 0);

Data table1;
set table;
If bads<&BinNumLim then lt30=1;
else lt30=0;
retain sum_lt30;
sum_lt30+lt30;
lower=lag(upper);
run;

proc sql;
create table sum as select max(sum_lt30) as sum_lt30 from table1;
quit;

data _Null_;
Set sum;
call symputx('m',sum_lt30);
run;
```

```
data table2;
set table1;
x + 1;
run;

proc sort data = table2;
by descending x;
run;

data table3;
set table2;
lead_bads = lag(bads);
lead_freq = lag(freq);
lead_upper = lag(upper);
lead_lt30 = lag(lt30);
run;

proc sort data = table3;
by x;
run;

data table4;
set table3;
lag_x=lag(x);
lag_lower = lag(lower);
lag_bads = lag(bads);
lag_freq = lag(freq);
run;

/*Generate number of rows variable*/
data _null_;
if 0 then set table4 nobs=n;
call symputx('nrows',n);
stop;
run;

%put nobs=&nrows;
/*-End-*/

proc sql;
create table row as select * from table4 where sum_lt30=1;
quit;

proc sql;
create table minrow as select * from row where x=(select min(x) from row);
quit;

Data minrow1;
set minrow;
if x ne &nrows then bads=bads+lead_bads;
else  bads=bads+lag_bads;
if x ne &nrows then freq=freq+lead_freq;
else  freq=freq+lag_freq;
if x ne &nrows then upper=lead_upper;
else  upper = upper;
if x ne &nrows then lower=lower;
else  lower = lag_lower;
```

```
Run;

proc sql;
update table4
set bads=(select bads from minrow1 where x=table4.x),
freq=(select freq from minrow1 where x=table4.x),
upper=(select upper from minrow1 where x=table4.x),
lower=(select lower from minrow1 where x=table4.x)
where exists (select * from minrow1 where x=table4.x);
quit;

proc sql;
create table table5 as select * from table4 left join (select x as delete
from minrow1)
on minrow1.x = table4.lag_x;
quit;

data _Null_;
Set minrow1;
call symputx('current_row',x);
run;

data table6;
set table5;
if delete NE . then delete;
/*Delete last row if last row criteria less than min_obs*/
if x=&nrows-1 and lead_lt30=1 and &current_row = &nrows then delete;
drop lt30
sum_lt30
lag_bads
x
lead_bads
lead_freq
lead_upper
lag_x
lag_bads
lag_freq
lag_lower
woe
delete
lead_lt30;
run;

data table;
set table6;
run;

%end;

data table;
set table;
badrate=bads/freq;
run;

/*******************************************/
/* Y=0 (non-default combine bin < &BinNumLim)*/
/*******************************************/
proc sort data=table out=table ;
```

```
  by descending upper ;
run ;

Data table1;
set table;
goods = freq - bads;
If goods<&BinNumLim then lt30=1;
else lt30=0;
retain sum_lt30;
sum_lt30+lt30;
run;

data _Null_;
Set table1;
call symputx('m',sum_lt30);
run;

%put &=m;

data table;
set table1;
drop lt30
sum_lt30;
run;

%do %while(&m NE 0);

proc sort data=table out=table1 ;
  by descending upper ;
run;

Data table2;
set table1;
goods = freq - bads;
If goods <&BinNumLim then lt30=1;
else lt30=0;
retain sum_lt30;
sum_lt30+lt30;
run;

proc sql;
create table sum as select max(sum_lt30) as sum_lt30 from table2;
quit;

data _Null_;
Set sum;
call symputx('m',sum_lt30);
run;


data table3;
set table2;
x + 1;
run;

proc sort data = table3;
by descending x;
run;
```

```
data table4;
set table3;
lead_goods = lag(goods);
lead_bads = lag(bads);
lead_freq = lag(freq);
lead_lower = lag(lower);
lead_lt30 = lag(lt30);
run;

proc sort data = table4;
by x;
run;

data table5;
set table4;
lag_x = lag(x);
lag_lower = lag(lower);
lag_bads = lag(bads);
lag_freq = lag(freq);
lag_upper = lag(upper);
run;

/*---generate number of rows variable---*/
data _null_;
if 0 then set table5 nobs=n;
call symputx('nrows',n);
stop;
run;

%put nobs=&nrows;

proc sql;
create table row as select * from table5 where sum_lt30=1;
quit;

proc sql;
create table minrow as select * from row where x=(select min(x) from row);
quit;

data minrow1;
set minrow;
if x ne &nrows then bads=bads+lead_bads;
else  bads=bads+lag_bads;
if x ne &nrows then goodss=goods+lead_goods;
else  goods=goods+lag_goods;
if x ne &nrows then freq=freq+lead_freq;
else  freq=freq+lag_freq;
if x ne &nrows then lower=lead_lower;
else  lower = lower;
if x ne &nrows then upper=upper;
else  upper = lag_upper;
Run;

proc sql;
update table5
set bads=(select bads from minrow1 where x=table5.x),
goods=(select goods from minrow1 where x=table5.x),
```

```
freq=(select freq from minrow1 where x=table5.x),
upper=(select upper from minrow1 where x=table5.x),
lower=(select lower from minrow1 where x=table5.x)
where exists (select * from minrow1 where x=table5.x);
quit;

proc sql;
create table table6 as select * from table5 left join (select x as delete
from minrow1)
on minrow1.x = table5.lag_x;
quit;

data _Null_;
Set minrow1;
call symputx('current_row',x);
run;

data table7;
set table6;
if delete NE . then delete;
if x=&nrows-1 and lead_lt30=1 and &current_row = &nrows then delete;
drop
lt30
sum_lt30
lag_bads
lag_goods
lag_woe
lag_lower
lag_upper
woe
lag_freq
goods
x
lead_bads
lead_goods
lead_freq
lead_lower
lag_x
delete
badrate
lead_lt30;
run;

data table;
set table7;
run;

%end;

data table;
set table;
badrate=bads/freq;
run;

proc sort data=table out=table ;
  by descending upper;
run ;
```

```
data table;
set table;
goods=freq-bads;
run;

data range_out;
set table;
rename Lower=min_&varx;
rename Upper=max2_&varx;
rename Bads=_freq_;
rename Freq=num_obs;
run;

data range_out;
set range_out;
if min_&varx. EQ . then min_&varx.=-99999999999999;
run;

data range_out;
set range_out;
Rnk_2_&varx.=_N_-1;
run;

%mend CombineBin(input_data=,var_x=,Min_Bin_Obs=);
```