Communications in Statistics - Simulation and Computation

ISSN: 0361-0918 (Print) 1532-4141 (Online) Journal homepage: http://www.tandfonline.com/loi/lssp20

A comparison study of computational methods of Kolmogorov–Smirnov statistic in credit scoring

Guoping Zeng

To cite this article: Guoping Zeng (2017) A comparison study of computational methods of Kolmogorov–Smirnov statistic in credit scoring, Communications in Statistics - Simulation and Computation, 46:10, 7744-7760, DOI: 10.1080/03610918.2016.1249883

To link to this article: https://doi.org/10.1080/03610918.2016.1249883

Accepted author version posted online: 02 Nov 2016.
Published online: 09 May 2017.

COMMUNICATIONS IN STATISTICS—SIMULATION AND COMPUTATION®
2017, VOL. 46, NO. 10, 7744–7760
https://doi.org/10.1080/03610918.2016.1249883

Taylor & Francis
Taylor & Francis Group

Check for updates

A comparison study of computational methods of Kolmogorov–Smirnov statistic in credit scoring

Guoping Zeng

Risk Management, Elevate, Fort Worth, TX, USA

ABSTRACT
Kolmogorov–Smirnov statistic (KS) is a standard measure in credit scoring. Currently, there are three computational methods of KS: method with equal-width binning, method with equal-size binning and method without binning. This paper compares the three methods in three aspects: Values, Rank Ordering of Scores and Geometrical Way. The computational results on the German Credit Data show that only the method without binning can produce a unique value of KS. It is further proved analytically that the method without binning yields the maximum value of KS among the three methods. The computational results also show that only the method with equal-size binning can be used to evaluate rank ordering of scores. Moreover, it is proved that all the three methods can be used to calculate KS in a geometric way.

ARTICLE HISTORY
Received 30 April 2016
Accepted 13 October 2016

KEYWORDS
Credit scoring; Equal-width binning; Equal-size binning; Kolmogorov–Smirnov statistic; Rank ordering; Lift table

MATHEMATICS SUBJECT CLASSIFICATION
Primary 62-07; Secondary 62G08

1. Introduction

Credit scoring is a process in which a statistic model is built to score loan applicants in terms of their good or bad status of returning loans. The main objective in credit scoring is to separate good applicants (or goods in short) from bad applicants (or bads in short) as much as possible. Albeit a short history, credit scoring has been recently extensively used to evaluate risk of a loan applicant (Bolton, 2009; Refaat, 2011; Siddiqi, 2006; Thomas et al., 2002).

Logistic regression is a statistic model to explain relationship between the binary dependent variable and one or more independent variable (Hosmer et al., 2013). It plays a central role in credit scoring due to the binary nature of the dependent variable–good or bad status of returning loans. Indeed, logistic regression was used in credit scoring as early as in 1980's (Srinivasan and Kim, 1987; Wiginton, 1980). Since then it has attracted major attentions in both academia and industry (Bolton, 2009; Hand and Henley 1997; Refaat, 2011; Zeng, 2013; Zeng, 2014a; Zeng, 2014b; Zeng 2015; Zeng, 2016). Recently, other types of models from machine learning (Khandani et al., 2010; Kennedy, 2013; Kraus, 2014; Li and Zhong, 2012) have been tried in credit scoring. Yet, logistic regression will continue to dominate credit scoring for years due to their accuracy (Abdou and Pointon, 2011; West, 2000), robustness and simplicity.

The Kolmogorov–Smirnov statistic KS is a standard measure of model strength or model performance in credit scoring. It measures the difference, distance or separation between goods and bads. In more details, KS measures the maximum difference across all scores in

CONTACT Guoping Zeng guopingtx@yahoo.com Risk Management, Elevate, 4150 International Plaza, Fort Worth, TX 76109, USA.
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/lssp.

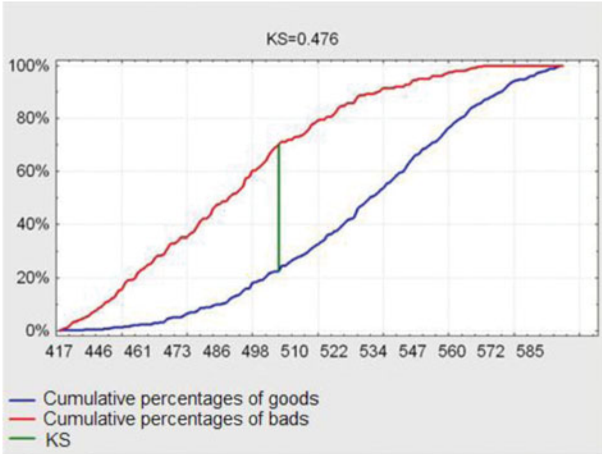

Figure 1. KS Graph.

the cumulative percentages of goods and bads. It is often explained as the maximum vertical distance between two curves in a KS graph as shown in Figure 1, in which the cumulative percentages of bads and the cumulative percentages of goods are plotted across the range of scores. Note that a KS graph can be drawn by using scores and the dependent variable.

KS in credit scoring was originally from the well-known Kolmogorov–Smirnov test in statistics. The Kolmogorov–Smirnov test was proposed by Smirnov in 1939 (Smirnov, 1939) to test the hypothesis that two cumulative distributions are statistically the same. This two-sample Kolmogorov–Smirnov test has been conventionally used in credit scoring to compare the cumulative distributions of goods and bads.

Although KS is widely used in credit scoring in industry, the literature is very limited. Indeed, even the literature about credit scoring is very limited, as pointed out by Abdou and Pointon (2011). Thrasher (1992) presented one of the first uses of KS in credit scoring. She explained KS geometrically, but didn't show how to calculate it algebraically. Mays (2001) calculated KS by binning (that is, grouping) scores with an equal width and then building a table. Thomas, et al. (2002) showed an example to calculate KS in credit scoring for attributes not for scores in Section 4.7.1. They did mention KS for scores in Section 7.5 but didn't show how to calculate its value. Wilkie (2004) formally defined KS in credit scoring but did not show how to calculate it either. Siddiqi (2006) and Bolton (2009) both used a KS graph to demonstrate how to calculate KS. Similar to Mays (2001), Thomas (2009) calculated KS by binning scores with an approximately equal width and building a table. Rezac and Rezac (2011) defined KS mathematically and used a KS graph to demonstrate how to calculate KS. Refaat (2011) as well as Gadidov and McBurnett (2015) calculated KS by binning scores with an equal size.

In summary, currently there are three computational methods of KS in terms of score binning: (1) the method with Equal-width binning (Mays, 2001), (2) the method with Equal-size binning (Gadidov and McBurnett 2015; Refaat, 2011), and (3) the method without binning (Bolton, 2009; Rezac and Rezac, 2011; Siddiqi, 2006; Thomas et al., 2002; Wilkie, 2004). Note that a KS graph essentially employs the method of without binning as it uses the range of scores. The three computational methods of KS coexist in credit scoring, although its definition is clear. Lack of a standard method to calculate the value of the standard measure KS leads to confusions in credit scoring. People are confused about which method to calculate KS and wonder about the relationship among these different computational methods. This paper aims at presenting a comparison study of the three methods in 3 aspects: Values, Rank Ordering of Scores and Geometrical Way.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the three computational methods of KS by summarizing them into computer algorithms. In Section 3, we present a unified numeric definition of the three computational methods of KS by treating the method without binning as a method with a kind of binning by making each score as a bin. In Section 4, we compare the three methods in three aspects: Values, Rank ordering and Geometric way by running logistic regression on the German Credit data. Finally, the paper is concluded in Section 5.

Throughout this paper, loans, records, customers, applications and accounts are all exchangeable terms.

2. Three computational methods of *KS*

From now on, we assume that a sample of N records have been scored by a credit scoring model. We also assume that the binary dependent variable y has N_b bads ($y = 1$) and N_g goods ($y = 0$) so that $N_b + N_g = N$.

2.1. Method with equal-width binning

The method with equal-width binning was proposed in Mays (2001). Essentially, it uses the following algorithm:

Step 1. Sort all records by score in ascending order.
Step 2. Divide all scores into bins with a fixed interval of scores.
Step 3. For each score bin, calculate the number of goods, the number of bads, the cumulative percentages of bads, the cumulative percentages of goods, and separation (the difference between the cumulative percentages of bads and the cumulative percentages of goods).
Step 4. Calculate KS by taking the maximum separation among all score bins.

2.2. Method with equal-size binning

The method with equal-size binning is similar to the one with equal-width binning. The only difference lies in Step 2. Instead of dividing all scores into bins with a fixed interval of scores, this method divides all scores into bins with an equal number of loans. In practice, it is rare to have an equal number of loans in all bins due to tied scores especially when scores are rounded into integers.

2.3. Method without binning

The method without binning is most commonly used in credit scoring. Unlike the method with equal-width binning or the method with equal-size binning, this method does not divide scores into bins. Thus, this method essentially uses only 3 steps:

Step 1. Sorted all records by score in ascending order.
Step 2. For each score, calculate the number of goods, the number of bads, the cumulative percentages of bads, the cumulative percentages of goods, and separation (the difference between the cumulative percentages of bads and the cumulative percentages of goods).

Step 3. Calculate KS by taking the maximum separation among all scores.

Remark 2.1. Since calculations at Step 2 are done at each score, this method essentially uses a kind of binning in which each score is treated as a bin, or equal-width binning with a width of 0.

3. A unified numerical definition

By treating the method without binning as a method with binning in Remark 2.1, we are now ready to give a unified numerical definition for all the three Computational Methods of KS.

To start with, let m be the number of bins such that scores in a lower bin are smaller than the scores in a higher bin. Let $n_g(j),\ n_b(j)$ be the numbers of goods and bads with scores in bin j. In other words, $n_g(j)$ and $n_b(j)$ are the number of records with scores in bin j and $y = 1$, and the number of records with scores in bin j and $y = 0$, respectively. Let $N_g(j),\ N_b(j)$ be the numbers of goods and bads with scores in bin j or lower bins so that

$$N_g(j) = \sum_{i=1}^{j} n_g(i) \tag{3.1}$$

and

$$N_b(j) = \sum_{i=1}^{j} n_b(i). \tag{3.2}$$

Let $p_g(j),\ p_b(j)$ be the proportions of goods in bin j and the proportions of bads in bin j, so that

$$p_g(j) = \frac{n_g(j)}{N_g} \tag{3.3}$$

and

$$p_b(j) = \frac{n_b(j)}{N_b}. \tag{3.4}$$

Let $P_g(j),\ P_b(j)$ be the proportions of goods and bads with scores in bin j or lower bins so that

$$P_g(j) = \frac{N_g(j)}{N_g} = \sum_{j=1}^{j} p_g(j) = \frac{\sum_{i=1}^{j} n_g(i)}{N_g} \tag{3.5}$$

and

$$P_b(j) = \frac{N_b(j)}{N_b} = \sum_{i=1}^{j} p_b(i) = \frac{\sum_{i=1}^{j} n_b(i)}{N_b}. \tag{3.6}$$

Then, the KS is given by

$$KS = \max_{1\le j\le m} |P_b(j) - P_g(j)|. \tag{3.7}$$

As mentioned before, for a good credit scoring model, $P_b(j)$ is generally greater than $P_g(j)$ except at bins with low scores and high scores where they are both 0 and 1, respectively. Hence, (3.7) can be simplified as

$$KS = \max_{1\le j\le m} (P_b(j) - P_g(j)). \tag{3.8}$$

In particular, if we make each score as a bin, then our numerical definition reduces to that in Wilkie (2004).

4. Comparison of computational methods of *KS*

4.1. A logistic regression model for the German credit data

To compare the three computational methods of KS, we shall use the German Credit Data from a German bank. They contain 1000 records representing 1000 loan applicants, where each record has information about the credit status to classify if this applicant is Good or Bad. The dependent variable in the German Credit Data, also called the target variable or the response variable, is the one that classifies the credit status of Good or Bad.

The German Data have 20 independent variables, which provide basic information about the applicants:

- Account_Balance: Account Balance
- Duration_of_Credit: Duration of Credit (month)
- Payment_Status: Payment Status of Previous Credit
- Purpose: Purpose
- Credit_Amount: Credit Amount
- Value_Savings: Value Savings/Stocks
- Width_employment: Width of current employment in years
- Instalment_per_cent:Instalment per cent
- Sex_Marriage: Sex & Marital Status
- Guarantors: Guarantors
- Duration_Current_address: Duration in Current address in years
- Most_valuable_asset: Most valuable available asset
- Age: Age in years
- Concurrent_Credits: Concurrent Credits
- Type_apartment: Type of apartment
- No_Credits_this_Bank: No of Credits at this Bank
- Occupation: Occupation 0
- No_dependents: No of dependents
- Telephone: Telephone
- Foreign_Worker: Foreign Worker.

Note that some names are either shortened or hyphened for the modeling purpose.

The German Credit Data are widely used for research purposes. For instance, they are hosted by the UCI Machine Learning Repository (Lichman, 2013) and by the Penn State online course *STAT 897D: Applied Data Mining and Statistical Learning* ("Analysis of German Credit Data," n.d.). For convenience, we shall use the one by Penn State. Note that the original German Credit Data contain both numeric and categorical variables. For the convenience of modeling, all the categorical variables are converted into numeric ones.

Let's build a logistic regression model using 10 independent variables. As to the other 10 independent variables, they are either restricted by business logistics or have a small contribution to the model.

Since dependent variable Creditability has a value of 1 for credit-worthy and 0 otherwise, we shall define a new dependent variable called Default by reversing Creditability, that is, Default = 1 – Creditability.

Table 1. Output of logistic regression.

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Error	Standard Estimate	Wald Chi-Square	Pr > ChiSq
Intercept	1	2.1439	0.5901	13.1979	0.0003
Account_Balance	1	− 0.5656	0.0684	68.3692	<.0001
Duration_of_Credit	1	0.0356	0.00678	27.6574	<.0001
Payment_Status	1	− 0.3332	0.0772	18.6314	<.0001
Value_Savings	1	− 0.2267	0.0568	15.9240	<.0001
Most_valuable_asset	1	0.2327	0.0847	7.5470	0.006
Length_employment	1	− 0.158	0.0666	5.6177	0.0178
Instalment_per_cent	1	0.2179	0.0733	8.8420	0.0029
Sex_Marriage	1	− 0.2562	0.112	5.2296	0.0222
Concurrent_Credits	1	− 0.2261	0.1075	4.4184	0.0356
Type_apartment	1	− 0.3075	0.156	3.8870	0.0487

The following is the SAS code for logistic regression, where the output file is called out_0 and the bad probability prob.

```
proclogistic data = german_credit descending; model default
= Account_BalanceDuration_of_CreditPayment_Status
Value_SavingsMost_valuable_assetWidth_employment
Instalment_per_centSex_MarriageConcurrent_Credits
Type_apartment/MAXITER = 100; output out = out_0 p =
probxbeta = logit;
run;
```

The output in Table 1 shows that all the 10 independent variables have a small p-value(less than 0.05).

Now, let's convert the bad probability to a score ranging from 0 to 1000 rounded into integers:

```
datagerman_credit_scores;set out_0;
score =round(1000–prob*1000);run;
```

4.1.1. KS without binning

We may calculate KS without binning in two ways. The first way is to call a SAS macro *KS_no_bin* in Appendix A1. This macro calculates KS from a dataset containing the scores and the dependent variable. It will display the value of KS to the SAS Log window. Applying this macro to the score dataset german_credit_scores, we find KS to be 0.472857:

$$\%ks\_no\_bin(german\_credit\_scores, score, default);$$

The second way is from the D statistics in SAS procedure npar1way with option EDF:

```
procnpar1wayedf data=german_credit_scores;
class default;
var score;
run;
```

As shown in Table 2, the D statistic is 0.472857. Procedure npar1way with option EDF performs a Kolmogorov–Smirnov Two-Sample test on the null hypothesis: the distribution of goods is the same as the distribution of bads. The KS statistic (0.21669) in Table 2 is actually the KS statistic for Kolmogorov–Smirnov Two-Sample test, not the KS in credit scoring.

Unlike the method of without binning, methods of equal-width binning and equal-size binning find KS by means of a lift table. In a lift table, the first column lists the score bin.

Table 2. Output of: KS test.

Kolmogorov–Smirnov Two-Sample Test (Asymptotic)			
KS	0.21669	**D**	0.472857
KSa	6.852351	**Pr>KSa**	<.0001

The next 3 columns list the number of applicants, number of bads and number of goods in each bin. The following two columns are the cumulative percentages of goods and bads at that bin and bins with lower scores. The sep column is the difference between the cumulative bad percentage and cumulative good percentage. The bad_rate column is the percent of total loans in each bin that are bad. The bad_pcn column lists the percent of bads at that bin, which is simply the number of bads at that bin divided by the total number of bads in the population. The last column good_pcn lists the percent of goods at that bin, which is simply the number of goods at that bin divided by the total number of goods in the population. Finally, the KS can be found by taking the maximum value among the Separation column.

Usually, we don't build a lift table for the method without binning. On the one hand, there are usually many distinct scores which could make such a table too long to view. On the other hand, scores are usually unbalanced in that some scores may have only one record and others may have many records.

4.1.2. KS with equal-width binning

In the method of equal-width binning for KS, we first do equal-width binning for scores and then build a lift table to calculate KS. The width of scores is determined by the number of bins. For 10 bins, the score width can be calculated as (990 – 98)/10 = 89.2 and the 9 bin boundaries as 187.2, 276.4, 365.6, 454.8, 544, 633.2, 722.4, 811.6, 900.8. Indeed, a call to macro EqWBinn in (Refaat, 2001) will yield the same result. Since our scores are all integers, we can do equal-width binning as follows:

```
data equal_width_bin_10;
setgerman_credit_scores;
IF98 <= score <= 187 THEN Score_Bin = '[98, 187]';
IF188 <= score <= 276 THEN Score_Bin = '[188, 276]';
IF277 <= score <= 365 THEN Score_Bin = '[277, 365]';
IF 366 <= score <= 454 THEN Score_Bin = '[366, 454]';
IF 455 <= score <= 544 THEN Score_Bin = '[455, 544]';
IF545 <= score <= 633 THEN Score_Bin = '[545, 633]';
IF 634 <= score <=722 THEN Score_Bin = '[634, 722]';
IF 723 <= score <= 811 THEN Score_Bin = '[723, 811]';
IF 812 <= score <= 900 THEN Score_Bin = '[812, 900]';
IF 901 <= score <= 990 THEN Score_Bin = '[901, 990]';
run;
```

A call to SAS macro ***lift_table*** in Appendix A3 will generate a lift table shown in Table 3 and find KS to be 0.469048. Note that column good_pcn is not displayed in Table 3 and the remaining three lift tables since it does not affect our analysis.

$$\%lift\_table(equal\_width\_bin\_10, default, score\_bin);$$

Note that macro ***lift_table*** calculates KS from a dataset with score bins and dependent variable, displays the value of KS in the SAS Log window, and output a lift table.

Table 3. Equal-width Binning with 10 bins and width = 89.2.

score_bin	cnt	bads	goods	cum_bad_pcn	cum_good_pcn	sep	bad_rate	bad_pcn
[98, 187]	16	16	0	5.33%	0.00%	5.33%	100.00%	5.33%
[188, 276]	48	35	13	17.00%	1.86%	15.14%	72.92%	11.67%
[277, 365]	45	26	19	25.67%	4.57%	21.10%	57.78%	8.67%
[366, 454]	56	31	25	36.00%	8.14%	27.86%	55.36%	10.33%
[455, 544]	86	54	32	54.00%	12.71%	41.29%	62.79%	18.00%
[545, 633]	84	37	47	66.33%	19.43%	46.90%	44.05%	12.33%
[634, 722]	100	24	76	74.33%	30.29%	44.05%	24.00%	8.00%
[723, 811]	159	36	123	86.33%	47.86%	38.48%	22.64%	12.00%
[812, 900]	193	24	169	94.33%	72.00%	22.33%	12.44%	8.00%
[901, 990]	213	17	196	100.00%	100.00%	0.00%	7.98%	5.67%
	1,000	300	700					100.00%

Next, we use 9 bins. The score width can be calculated as (990 − 98)/9 = 99.11111 and bin boundaries as 197.11111111, 296.22222222, 395.33333333, 494.44444444, 593.55555556, 692.66666667, 791.77777778, 890.88888889. Since our scores are all integers, we can do equal-width binning as follows:

data equal_width_bin_9;
setgerman_credit_scores;
IF98<= score <= 197 THEN Score_Bin = '[98, 197]';
IF198<= score <= 296 THEN Score_Bin = '[198, 296]';
IF 297<= score <= 395 THEN Score_Bin = '[297, 395]';
IF 396<= score <= 494 THEN Score_Bin = '[396, 494]';
IF 495<= score <= 593 THEN Score_Bin = '[495, 593]';
IF 594<= score <= 692 THEN Score_Bin = '[594, 692]';
IF 693<= score <=791 THEN Score_Bin = '[693, 791]';
IF792<= score <= 890 THEN Score_Bin = '[792, 890]';
IF 891<= score <= 990 THEN Score_Bin = '[891, 900]';
run;

A call to macro ***lift_table*** will generate a lift table shown in Table 5 and find KS to be 0.459048:

$$\%lift\_table(equal\_width\_bin\_9, default, score\_bin);$$

Remark 4.1. In practice, people often multiply KS by 100 to have a better scale.

4.1.3. KS with equal-size binning

In the method of equal-size binning, we first do equal-size binning for scores and then build a lift table to calculate KS. Again, we consider two cases: 10 bins and 9 bins. For 10 bins, we call macro ***equal_size_bin*** in Appendix A2 to do equal-size binning with 10 bins for the scores:

$$\%equal\_size\_bin(german\_credit\_scores, score, 10);$$

Note that macro *equal_size_bin* finds the binning rules from a dataset with scores and outputs the binning rules in the last column.

Next, let's copy and paste the last column of the result (the IF-ELSE statements) to generate a new dataset equal_size_bin_10:

data equal_size_bin_10;
setgerman_credit_scores;
IF98<= score <= 340 THEN Score_Bin = '[98, 340]';

Table 4. Equal-size binning with 10 bins.

score_bin	cnt	bads	Goods	cum_bad_pcn	cum_good_pcn	sep	bad_rate	bad_pcn
[98, 340]	100	72	28	24.00%	4.00%	20.00%	72.00%	24.00%
[342, 488]	99	60	39	44.00%	9.57%	34.43%	60.61%	20.00%
[490, 601]	101	56	45	62.67%	16.00%	**46.67%**	55.45%	18.67%
[602, 694]	100	27	73	71.67%	26.43%	45.24%	27.00%	9.00%
[695, 763]	100	26	74	80.33%	37.00%	43.33%	26.00%	8.67%
[765, 814]	101	19	82	86.67%	48.71%	37.95%	18.81%	6.33%
[816, 868]	99	13	86	91.00%	61.00%	30.00%	13.13%	4.33%
[870, 906]	102	14	88	95.67%	73.57%	22.10%	13.73%	4.67%
[907, 941]	100	9	91	98.67%	86.57%	12.10%	9.00%	3.00%
[942, 990]	98	4	94	100.00%	100.00%	0.00%	4.08%	1.33%
	1,000	300	700					100.00%

IF 342<= score <= 488 THEN Score_Bin = '[342, 488]';
IF 490<= score <= 601 THEN Score_Bin = '[490, 601]';
IF 602<= score <= 694 THEN Score_Bin = '[602, 694]';
IF 695<= score <= 763 THEN Score_Bin = '[695, 763]';
IF 765<= score <= 814 THEN Score_Bin = '[765, 814]';
IF 816<= score <= 868 THEN Score_Bin = '[816, 868]';
IF 870<= score <= 906 THEN Score_Bin = '[870, 906]';
IF 907<= score <= 941 THEN Score_Bin = '[907, 941]';
IF 942<= score <= 990 THEN Score_Bin = '[942, 990]';
run;
A call to macro ***lift_table*** will generate Table 4 and find the KS to be 0.466667.

$$\%\text{lift\_table}(equal\_size\_bin\_10, default, score\_bin);$$

For 9 bins, we call SAS macro equal_size_bin with bin_size = 9:

$$\%equal\_size\_bin(german\_credit\_scores, score, 9);$$

data equal_size_bin_9;
setgerman_credit_scores;
IF98 <= score <= 369 THEN Score_Bin = '[98, 369]';
IF 370 <= score <= 513 THEN Score_Bin = '[370, 513]';
IF 515 <= score <= 629 THEN Score_Bin = '[515, 629]';
IF 632 <= score <= 727 THEN Score_Bin = '[632, 727]';
IF 728 <= score <= 790 THEN Score_Bin = '[728, 790]';
IF 791 <= score <= 854 THEN Score_Bin = '[791, 854]';
IF 855 <= score <= 898 THEN Score_Bin = '[855, 898]';
IF 899 <= score <= 936 THEN Score_Bin = '[899, 936]';
IF 937 <= score <= 990 THEN Score_Bin = '[937, 990]';
run;

$$\%lift\_table(equal\_size\_bin\_9, default, score\_bin);$$

We'll see $KS = 0.467143$ and a lift table in Table 6.

Remark 4.2. All the 3 SAS macros only need a dataset with scores and dependent variable. Hence, they apply to all kinds of credit scoring models, no matter whether they are from a logistic regression model with SAS or from a nonlinear model with R or Python. These SAS macros can be easily converted into R functions.

Table 5. Equal-width Binning with 9 bins and width = 99.11111.

score_bin	cnt	bads	Goods	cum_bad_pcn	cum_good_pcn	sep	bad_rate	bad_pcn
[98, 197]	18	18	0	6.00%	0.00%	6.00%	100.00%	6.00%
[198, 296]	56	39	17	19.00%	2.43%	16.57%	69.64%	13.00%
[297, 395]	58	32	26	29.67%	6.14%	23.52%	55.17%	10.67%
[396, 494]	74	48	26	45.67%	9.86%	35.81%	64.86%	16.00%
[495, 593]	86	47	39	61.33%	15.43%	**45.90%**	54.65%	15.67%
[594, 692]	103	30	73	71.33%	25.86%	45.48%	29.13%	10.00%
[693, 791]	163	37	126	83.67%	43.86%	39.81%	22.70%	12.33%
[792, 890]	193	28	165	93.00%	67.43%	25.57%	14.51%	9.33%
[891, 900]	249	21	228	100.00%	100.00%	0.00%	8.43%	7.00%
	1,000	300	700					100.00%

4.2. Comparison of KS values

Let us first compare the uniqueness of the KS values. Since the method without binning does not depend upon any binning, it yields a unique value of KS.

The method with equal-width binning does not produce a unique value of KS. Different widths may result in different values of KS. Table 3 shows KS of 0.469048 with a width of 89.2 points. However, Table 5 shows KS of 0.459048 with a width of 99.11111 points. The method with equal-size binning does not produce a unique value of KS either. Different sizes may result in different values of KS. Table 4 shows KS of 0.466667with a size of 100 records. Yet, Table 6 shows KS of 0.467143 with a size of 111 records.

Next, let us compare the KS values between the method with equal-width binning and the method with equal-size binning. To make an apple to apple comparison, we consider the same number of bins. We see from Tables 3 and 4 that for 10 bins, the KS by equal-width binning is 0.469048, which is larger than $KS = 0.466667$ by equal-size binning. However, with 9 bins, as shown in Tables 5 and 6, the KS by equal-width binning is 0.459048, which is smaller than $KS = 0.467143$ by equal-size binning.

Now, let us compare the KS values of all the three methods. We summarize the result in the following theorem.

Table 6. Equal-size Binning with 9 bins.

score_bin	cnt	bads	Goods	cum_bad_pcn	cum_good_pcn	sep	bad_rate	bad_pcn
[98, 369]	111	78	33	26.00%	4.71%	21.29%	70.27%	26.00%
[370, 513]	111	67	44	48.33%	11.00%	37.33%	60.36%	22.33%
[515, 629]	111	53	58	66.00%	19.29%	**46.71%**	47.75%	17.67%
[632, 727]	112	27	85	75.00%	31.43%	43.57%	24.11%	9.00%
[728, 790]	110	25	85	83.33%	43.57%	39.76%	22.73%	8.33%
[791, 854]	111	18	93	89.33%	56.86%	32.48%	16.22%	6.00%
[855, 898]	112	15	97	94.33%	70.71%	23.62%	13.39%	5.00%
[899, 936]	114	11	103	98.00%	85.43%	12.57%	9.65%	3.67%
[937, 990]	108	6	102	100.00%	100.00%	0.00%	5.56%	2.00%
	1,000	300	700					100.00%

Theorem 4.3. *Among the 3 computational methods of KS, the method without binning always gives the largest value of KS.*

Proof. To start with, we assume there are$K$ distinct scores by $s_1 < s_2 < \cdots < s_K$ under the N records. For the method without binning, let us treat each score as a bin. As before, we denote

$n_b(i)$ and $n_g(i)$ as the numbers of bads and goods at bin i, that is, under score s_i. We have

$$KS = \max_{1 \le j \le K} |P_b(j) - P_g(j)| = \max_{1 \le j \le K} \left| \frac{\sum_{i=1}^{j} n_b(i)}{N_b} - \frac{\sum_{i=1}^{j} n_g(i)}{N_g} \right|. \tag{4.1}$$

Now assume there are m equal-width bins or equal-size bins such that scores $s_1,\ s_2, \ldots,\ s_{l_1}$ belong to bin 1, scores $s_{l_1+1},\ s_{l_1+2}, \ldots s_{l_2}$ belong to bin 2, ..., scores $s_{l_{m-1}+1},\ s_{l_{m-1}+2}, \ldots s_{l_m}$ belong to bin m, where $1 \le l_1 < l_2 < \cdots < l_m = K$. Without loss of generality, we may assume $m < K$. Then the numbers of bads and goods at bin j $(1 \le j \le m)$ are $n_b(l_{j-1}+1) + n_b(l_{j-1}+2) + \cdots + n_b(l_j)$ and $n_g(l_{j-1}+1) + n_g(l_{j-1}+2) + \cdots + n_g(l_j)$. Therefore, the numbers of bads and goods at bin j and lower bins are $\sum_{i=1}^{l_j} n_b(i)$ and $\sum_{i=1}^{l_j} n_g(i)$. So, the KS can be calculated as

$$KS = \max_{1 \le j \le m} \left| \frac{\sum_{i=1}^{l_j} n_b(i)}{N_b} - \frac{\sum_{i=1}^{l_j} n_g(i)}{N_g} \right| \tag{4.2}$$

Since $\{l_1,\ l_2,\ \ldots,\ l_m\}$ is a subset of $\{1,\ 2,\ \ldots,\ K\}$, the KS from the method with equal-width binning or from the method with equal-size binning in (4.2) is not larger than the KS without binning in (4.1) for any $m < K$. □

4.3. Comparison of rank ordering

A good credit scoring model should place bad customers at the bottom of the score range and good customers at the top of score range. This is called rank ordering. Since the first column score_bin in a lift table is in ascending order of scores, rank ordering means that bad customers are on the top of the table and good customers are at the bottom of the table.

We first consider the method with equal-size binning. If all scores are distinct, all the score bins in the lift table except the last one will have the same number of counts. In case of tied scores, not all score bins may have the same number of counts but approximately the same number of counts. The number of bads and hence the bad rates should be decreasing from top to bottom for a good credit scoring model. With the same denominator, that is, the number of bads in the population, the bad percentages should also be decreasing from top to bottom for a good credit scoring model. Thus, the method with equal-size binning can be conveniently used to evaluate rank ordering of a credit scoring model.

Next, let's move to the method with equal-width binning. Since credit scores turn to have a normal distribution or skewed normal distribution (Wilkie, 2004), not uniformly distributed. The number of bads and hence the bad rates and bad percentage may not be decreasing from top to bottom for a good credit scoring model. Hence, the method with equal-width binning cannot be used to evaluate rank ordering of scores.

Tables 4 and 6 for the method with equal-size binning both show rank ordering with a little shift from the 7th score bin to the 8th score bin in Table 4. However, neither Table 3 nor Table 5 for the method with equal-width binning shows good rank ordering. Rather, there are significant shifts in the number of bads, bad rates and bad percentage.

4.3. Comparison of geometric ways

In a credit scoring model, the scores of goods and bads have two typical (though irregular) bell-shaped distributions. Moreover, the KS from the method without binning can be found geometrically at the score at which the two distributions cross (Wilkie, 2004). However, an analytical proof is not provided.

Figure 5–2 in Mays (2001) compares the score distribution for a group of good loans and a group of bad loans. Mays points out that for a perfect credit scoring model, there would be no overlapping at all in the distributions. Although not mentioned, the scores of goods and bads show bell-shaped distributions with equal-width binning. Nevertheless, the *KS* from the method with equal-width binning is not linked to the score distributions of good loans and bad loans in Mays (2001).

To the best knowledge of the author, the *KS* from the method with equal-size binning has not been linked to the score distributions of goods and bads.

In this subsection, we shall link the KS from the three methods to the score distributions of goods and bads and provide an analytical proof. Let us first state a lemma about the score distributions of bads and goods.

Lemma 4.4. *The score distribution of bads among the m bins is* $\{p_b(j)\}_{j=1}^{m}$, *while the score distribution of goods among the m bins is* $\{p_g(j)\}_{j=1}^{m}$.

Proof. We only prove the first part which is for the bads. The second part, which is for the goods can be proved similarly. With notation in Section 3, there are N_b records for bads, that is, N_b records with $y = 1$. Among the N_b records for bads, let us denote the number of records with scores in bin j by $n_b(j)$ for $j = 1, 2, \ldots, m$. Then $\{\frac{n_b(j)}{N_b}\}_{j=1}^{m}$ is the score distribution of bads among the m bins. Since $n_b(j)$ is the number of records with scores in bin j and $y = 1$, $n_b(j) \geq n_b(j)$ for $j = 1, 2, \ldots, m$.

Since

$$\sum_{j=1}^{m} n_b(j) = N_b = \sum_{j=1}^{m} n_b(j),$$

we have $n_b(j) = n_b(j)$ for all $j = 1, 2, \ldots, m$. Thus,

$$\frac{n_b(j)}{N_b} = \frac{n_b(j)}{N_b} = p_b(j)$$

for $j = 1, 2, \ldots, m$. Hence, the score distribution of bads among the m bins is $\{p_b(j)\}_{j=1}^{m}$.□

For the method with equal-width binning, if the number of bins is large, the scores of bads and goods are bell-shaped as demonstrated in Figure 5–2 in Mays (2001). However, if the number of bins is not large enough, the scores of bads and goods may not be bell-shaped as Tables 3 and 5 show.

As to the method with equal-size binning, $\{p_b(j)\}_{j=1}^{m}$ is decreasing and $\{p_g(j)\}_{j=1}^{m}$ is increasing. Therefore, the scores of bads and goods are not bell-shaped.

Nevertheless, for a good credit scoring model, at the beginning the score distribution of bads is on the top of the score distribution of goods and then moves to the bottom. With this observation, we are now ready for our main result.

Theorem 4.5. *The KS from all the 3 methods (with equal-width binning, with equal-size binning and without binning) can be found geometrically at the bin at which the two distributions cross.*

Proof. Assume there are m bins and the score distributions of goods and bads cross at bin i_0 where $1 < i_0 < m$. We note that the following hold: $p_b(i) > p_g(i)$, if $i < i_0$, $p_b(i) \geq p_g(i)$ if $i = i_0$ and $p_b(i) < p_g(i)$ if $i > i_0$. Here, by $p_b(i_0) \geq p_g(i_0)$ we mean that that cross each other if $p_b(i_0) = p_g(i_0)$ and just miss each other if $p_b(i_0) > p_g(i_0)$.

$$P_b(i) - P_g(i) = \sum_{j=1}^{i} p_b(i) - \sum_{j=1}^{i} p_g(i) = \sum_{j=1}^{i} (p_b(i) - p_g(i))$$

Since

$$(P_b(i) - P_g(i)) - (P_b(i-1) - P_g(i-1)) = p_b(i) - p_g(i) \geq 0 \; for \; i \leq i_0,$$

and $(P_b(i) - P_g(i)) - (P_b(i-1) - P_g(i-1)) = p_b(i) - p_g(i) < 0$ for $i > i_0$, sequence $\{P_b(i) - P_g(i)\}_{i=1}^{m}$ is increasing from $i = 1$ to $i = i_0$ and decreasing from $i = i_0 + 1$ to $i = m$. Hence, $P_b(i) - P_g(i)$ attains its maximum value at $i = i_0$, that is, the KS can be found to be $P_b(i_0) - P_g(i_0)$. □

In Chapter 7, "Measuring Scorecard Performance" of Thomas, et al. (2002), the probability density functions of goods and bads are used and the KS from the method without binning is geometrically linked to the intersection of the 2 density functions without analytical proofs. However, the scores of bads and the scores of bads are discrete random variables and hence density functions do not exist. Even if the scores of bads and the scores of goods are treated as continuous random variables, their density functions cannot be analytically found. After all, they are not standard distributed albeit with irregular bell shapes.

Note that the density function of goods and the density function of bads stand side by side in Figure 7.1 in Thomas, et al. (2002). Their areas should be both 1. However, the area of the density function of goods looks significantly larger than that of bads.

5. Conclusions

KS is a standard measure in credit scoring. However, there is no standard method to calculate the value of KS. Currently, there are three methods to compute KS: method with equal-width binning, method with equal-size binning and method without binning. These 3 methods coexist and causes confusions in credit scoring. This paper presents a comparison study on the 3 methods in 3 aspects: Values, Rank Ordering of Scores and Geometrical Way. Through computational results and analytical proofs, we reveal the relationship among the 3 methods. The method without binning yields a unique value of KS, which is also the maximum value of KS among the 3 methods. The KS value from the method of equal-size binning is not unique and is dependent on the (equal) size of the score bins. The KS value from the method of equal-width binning is not unique either and is dependent on the (equal) width of the score bins. The method of equal-size binning can be used to evaluate rank ordering of scores. All the three methods can be used to calculate KS in a geometric way by the intersection of distributions of goods and bads.

In conclusion, the method without binning can serve as the standard method to calculate the value of KS. The method with equal-size binning can be used to evaluate rank ordering of scores, in which the KS value can provide a quick and approximate value of the standard KS value. The method of equal-width is not recommended to calculate the value of KS.

Appendix. SAS macros

A1. Macro KS_no_bin

```
%macro ks_no_bin(InData, score, dept_var);
proc sort data = &InData;
```

```
by&score;
data ks_temp1;
set&InData end=last;
cumgood+1-&dept_var;
cumbad+&dept_var;
if last then do;
callsymput('tgoods',left(cumgood));
callsymput('tbads',left(cumbad));
end;
run;
data ks_temp2 (keep=&score pctdiff);
set ks_temp1;
by&score;
pctdiff = abs((cumgood/&tgoods-cumbad/&tbads)*100);
iflast.&score;
run;
procsqlnoprint;
select max(pctdiff) into: KS
from ks_temp2;
quit;
%put ... KS = &ks;
%mend;
```

A2. Macro equal_size_bin

```
%macroequal_size_bin(inData, score, bin_size);
proc rank data=&inData out = &inData._temp group =
&bin_size;
ranksxrank;
var&score;
run;
procsql;
create table &inData._bins as
selectxrank, min(&score) as min_score, max(&score) as
max_score
from&inData._temp
group by xrank;
quit;
/* Print out Binning Rule. */
dataBinning_Rule;
set&inData._bins;
widthbining_item $200;
bining_item = "IF " || put(min_score, 3.) || " <= " ||
"&score. <= "
|| put(max_score, 3.) || " THEN Score_Bin = '["
|| put(min_score, 3.) || ", " || put(max_score, 3.) ||
"]';";
run;
%mend;
```

A3. Macro lift_table

```
%macro lift_table(InData, dept_var, bin_var);
dataInData_New;
set&InData;
good_var = 1 - &dept_var;
run;
procsqlnoprint;
select sum(&dept_var), sum(good_var) into: total_bads,
:total_goods
fromInData_New;
quit;
procsql;
create table temp1 as
select&bin_var,
count(*) as cnt,
sum(&dept_var) as bads,
(calculatedcnt - calculated bads) as goods,
calculatedbads / &total_bads. asbad_pcn format =
percent8.2,
calculated goods / &total_goods. asgood_pcn format =
percent8.2,
calculatedbads /calculated cnt as bad_rate format =
percent8.2
fromInData_New
group by &bin_var
order by &bin_var;
quit;
data Temp2;
set Temp1;
retaincum_bad_pcncum_good_pcn;
cum_bad_pcn + bad_pcn;
cum_good_pcn + good_pcn;
sep = cum_bad_pcn - cum_good_pcn;
run;
proc print data=Temp2 noobs;
formatcnt comma8. bads comma8. goods comma8. cum_bad_pcn
percent8.2
cum_good_pcn percent8.2 sep percent8.2 bad_rate percent8.2
bad_pcn percent8.2 good_pcn percent8.2;
varscore_bincntbads goods cum_bad_pcncum_good_pcnsep
bad_ratebad_pcngood_pcn;
sumcntbads goods bad_pcngood_pcn;
run;
procsqlnoprint;
select max(sep) into: KS
from temp2;
quit;
%put ... KS = &ks;
%mend;
```

References

Abdou, H., Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: a review of the literature. *Intelligent Systems in Accounting, Finance & Management* 18(2, 3):59–88.

Analysis of German Credit Data. (n.d.). *Penn State online course STAT 897D: Applied Data Mining and Statistical Learning*. Retrieved from https://onlinecourses.science.psu.edu/stat857/sites/onlinecourses.science.psu.edu.stat857/files/german_credit.csv

Bolton, C. (2009). *Logistic Regression and Its Application in Credit Scoring* (doctoral dissertation). University of Pretoria, South Africa.

GadidovB., McBurnett, B. (2015). Paper 132. Population stability and model performance metrics replication for business model at SunTrust Bank. SESUG 2015.

Hand, D. J., Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A Review. *Journal of the Royal Statistical Society. Series A (Statistics in Society)* 160(3): 523–541.

Hosmer, D. W., Lemeshow, S., Sturdivant, R.X. (2013). *Applied Logistic Regression* (3rd edition). New York: John Wiley and Sons.

Kennedy, K. (2013). *Credit Scoring Using Machine Learning Scoring* (doctoral dissertation). School of Computing, Dublin Institute of Technology, Dublin, Ireland.

Khandani, A. E., Kim, A. J., LoA. W. (2010). Consumer credit risk models via machine-learning algorithms. *Journal of Banking and Finance* 34(11):2767–2787.

Kraus, A. (2014). *Recent Methods from Statistics and Machine Learning for Credit Scoring* (doctoral dissertation). Ludwig-Maximilians-Universitat Munchen, Germany.

Li, X. L., Zhong, Y. (2012). An Overview of Personal Credit Scoring: Techniques and Future Work. *International Journal of Intelligence Science* 2:181–189.

Lichman, M. (2013). *UCI Machine Learning Repository* [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Mays, E. (2001). *Handbook of Credit Scoring*. Chicago, Illinois, USA: Glenlake Publishing Company, Ltd.

Refaat, M. (2011). *Credit Risk Scorecards: Development and Implementation Using SAS*. Raleigh, North Carolina, USA: Lulu Press, Inc.

Rezac, M., Rezac, F. (2011). How to measure the quality of credit scoring models. *Czech Journal of Economics and Finance* 61(5):486–507.

Siddiqi, N. (2006). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Hoboken, New Jersey, USA: John Wiley & Sons, Inc.

Smirnov, N. (1939). On the estimation of the discrepancy between empirical curves of distribution for two independent samples.(Russian). *Bulletin of Moscow University* 2(2):3–14.

Srinivasan, V., Kim, Y. H. (1987). Credit granting: A comparative analysis of classification procedures. *The Journal of Finance* 42:665–683.

Thomas, L. C. (2009). *Consumer Credit Models: Pricing, Profit and Portfolios*. 1st Edition. Oxford: Oxford University Press.

Thomas, L. C., Edelman, D. B., Crook, J. N. (2002). *Credit Scoring and Its Applications*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Thrasher, R. P. (1992). A note on the use of the K-S statistic as a measure of model strength. *Journal Direct Marketing* 6(4):40–47.

West, D. (2000). Neural network credit scoring models. *Computers & Operations Research* 27(11, 12):1131–1152.

Wiginton, J. C. (1980). A note on the comparison of logit and discriminant models of consumer credit behaviour. *Journal of Financial and Quantitative Analysis* 15:757–770.

Wilkie, A. D. (2004, reprinted 2007) Measures for comparing scoring systems. In: L. C. Thomas, Edelman, D. B., Crook, J. N., eds. *Readings in Credit Scoring*. Oxford: Oxford University Press. pp. 51–62.

Zeng, G. (2013). Metric divergence measures and information value in credit scoring. *Journal of Mathematics* 2013:1–10

Zeng, G. (2014a). A rule of thumb for reject inference in credit scoring. *Mathematical Finance Letters* 2014, article 2.

Zeng, G. (2014b). A necessary condition for a good binning algorithm in credit scoring. *Applied Mathematical Sciences* 8(65):3229–3242. doi: 10.12988/ams.2014.44300

Zeng, G. (2015). A unified definition of mutual information with applications in machine learning. *Mathematical Problems in Engineering* 2015, Article ID 201874. doi: 10.1155/2015/201874

Zeng, G. (2016). Invariant properties of logistic regression model in credit scoring under monotonic transformations. *Communications in Statistics: Theory and Methods*. doi: 10.1080/03610926.2016.1193200