Paper 2528-2018

Using Information Value, Information Gain and Gain Ratio for Detecting Two-way Interaction Effect

Alec Zhixiao Lin, Loan Depot, Lake Forest, CA

ABSTRACT

An interaction effect occurs when the impact of one attribute on a dependent variable depends on the value of another attribute. The presence of an interaction effect between two attributes often weakens, dissolves, or even distorts the predictive power of either or both attributes if used alone in predicting the outcome. Consequently, some valuable information is discarded or ignored. Examination of a limited number of attributes for an interaction effect can be done manually, but a similar task involving numerous attributes can be challenging. By employing information theory, this paper suggests a combinational use of Information Value, Information Gain, and Gain Ratio to detect a two-way interaction effect at the preliminary stage of variable screening and selection. We also expand the use of the methodology to continuous dependent variables. A SAS® process is introduced that automatically screens all attributes in pair with minimal manual handling by users. The SAS output ranks all pairs of attributes in terms of their magnitude of interaction effect and offers suggestions on variable treatment for downstream analysis or modeling.

INTRODUCTION

An interaction occurs when the differing effect of one predictor on an outcome depends on the value of another predictor. Its presence has important implications for the interpretation of variable behaviors. The following are two examples of interaction effect:

- A direct mail campaign sent out solicitations in two forms: large-size postcards and letters. Multiple tests have shown that neither form is a winner. However, a closer examination of the data reveals that young people were more likely to respond to postcards, while others were more likely to respond to letters. Mixing two groups of people together cancels out the differences in response and hence conceals the marketing effectiveness of either form of solicitation.

- Advertisements in multiple media channels for the same product often produce synergistic effects. That is, two factors can jointly reinforce the impact on the outcome.

Let's use a stylized example to illustrate interaction effects. X_1 and X_2 are two predictors for event Y. X_1 is a categorical or an ordinal variable with values A, B and C. X_2 is a continuous variable divided into four bins. We cross tabulate X_1 and X_2 in a 2×2 table with average probabilities computed for each cell. For simplicity, we assume that observations are almost evenly distributed across all cells.

		X_2				
		1	*2*	*3*	*4*	*total*
X_1	*A*	0.098	0.075	0.015	0.008	0.049
	B	0.047	0.055	0.045	0.052	0.050
	C	0.015	0.022	0.095	0.105	0.059
	total	0.053	0.051	0.052	0.055	0.053

Table 1. Example of interaction effect

In the above table, we can make the following observations of X_1 causing X_2 to impact Y differently:

- If X_1='A', X_2 ↑ causes Y ↓.
- If X_1='B', X_2 is not predictive for Y.
- If X_1='C', X_2 ↑ causes Y ↑.

We can also examine how X_2 causes X_1 to impact Y differently.

- If X_2 in (1, 2), Y ↓ as X_1 moves from A to B and to C.
- If X_2 in (3, 4), Y ↑ as X_1 moves from A to B and to C.

However, X_1 or X_2 alone shows very limited predictive power for Y.

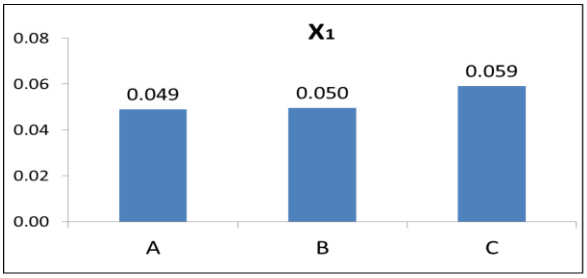

Figure 1. X_1 alone on Y

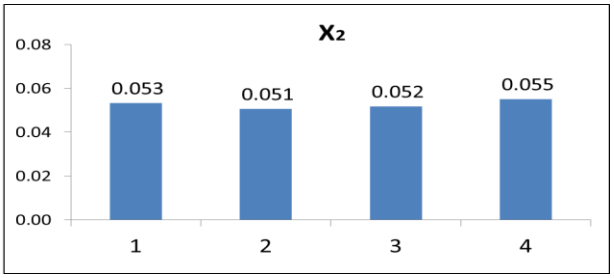

Figure 2. X_2 alone on Y

Here is an intuitive explanation for the interaction effect in our example: for X_2, the reverse of variable behavior from A to C by X_1 neutralizes the monotonic trend in each segment, and hence makes the X_2 an ineffective predictor when used alone. Similarly, X_2 neutralizes the predictive power of X_1. If we put both variables in a regression equation as follows, neither predictor is likely to show good predictive power.

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + ... + \varepsilon$$

For simplicity, let's assume X_2 is an ordinal variable here. We can include a multiplication term as follows:

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + ... + \varepsilon$$

The functional relationship between X_1 and X_2 is reflected in a transformation of the same equation:

$$Y = \alpha + \beta_1 X_1 + (\beta_2 + \beta_3 X_{1)}) X_2 + ... + \varepsilon$$

In practice, expert opinions and past experiences can be drawn to form a priori knowledge in possible interaction effects. If only a limited number of predictors are at play, it is not an arduous task to examine all pairs of variables for the purpose of detecting the presence of interactions. PROC LOGISTIC provides such an option. PROC TABULATE can also be used as shown in table 1. However, these practices might be inapplicable in the current era of big data with hundreds or even thousands of variables at play. Moreover, new information constantly emerges and new intelligence is constantly sought after.

This paper suggests the use of Information Value, Information Gain and Gain Ratio to at the early stage of data scrubbing and variable selection for detecting interaction effects between two attributes. These measures share some similarities, and using them in combination can offer best insights. Although they were originally developed for analyzing binary outcomes, we will explain how their usage can be expanded to continuous outcomes. Finally, the paper introduces an efficient SAS process that will automatically screen all potential predictors in pair with minimal manual handling by users.

INFORMATION VALUE, INFORMATION GAIN AND GAIN RATIO

Information Value, Information Gain and Gain Ratio are three commonly used measures based on Information Theory pioneered by Claude Shannon in late 1940s. Information Value is still widely used for building score cards and for modeling risk, response, etc. Information Gain and Gain Ratio are among core concepts of machine learning, especially for decision tree.

WEIGHT OF EVIDENCE (WOE) AND INFORMATION VALUE (IV)

Information Value (IV) is based on Weight of Evidence (WOE). The calculation of WOE requires the contrast between occurrence and non-occurrence (usually denoted by 1 and 0) across discrete bins. A categorical variable can use its original values (such as A, B and C for X_1) as bins or have multiple values collapsed into one bin. For a continuous variable, PROC RANK or some existing scheme can be used to recode it into a series of ordinal bins.

WOE is calculated as follows:

$$\text{WOE} = [\ln\left(\frac{\%\text{Occurence}_i}{\%\text{Nonoccurences}_i}\right)] \times 100$$

where

$$\%\text{Occurence}_i = \frac{\#\text{ Occurence in } i\text{th bin}}{\#\text{ Total Occurences}}$$

$$\%\text{Nonoccurence}_i = \frac{\#\text{ Nonoccurence in } i\text{th bin}}{\#\text{ Total NonOccurences}}$$

Natural log is the suggested log function for WOE. IV assesses the overall predictive power of the variable and can be used for evaluating competing variables. The following is how IV is calculated:

$$\text{IV} = \sum_{i=1}^{n} (\%\text{Occurence}_i - \%\text{Nonoccurence}_i) \times \ln(\frac{\%\text{Occurence}_i}{\%\text{Nonoccurence}_i})$$

In our example, we compute the IVs for X_1 and X_2 as follows. The last column sums up the Index for each bin to get the IV for the variable.

X_1	% Y	# Records	% Records	# Occurrence	# Non-occurrence	% Occurrence	% Non-Occurences	WOE	Index for IV
A	0.049	200,000	33%	9,800	190,200	31.06%	33.46%	-7.435	0.00178
B	0.050	200,000	33%	9,925	190,075	31.46%	33.44%	-6.102	0.00121
C	0.059	200,000	33%	11,825	188,175	37.48%	33.10%	12.418	0.00544
Total	0.053	600,000	100%	31,550	568,450	100.00%	100.00%		0.00843

X_2	% Y	# Records	% Records	# Occurrence	# Non-occurrence	% Occurrence	% Non-Occurences	WOE	Index for IV
1	0.053	150,000	25%	7,975	142,025	25.28%	24.98%	1.165	0.00003
2	0.051	150,000	25%	7,575	142,425	24.01%	25.05%	-4.262	0.00045
3	0.052	150,000	25%	7,750	142,250	24.56%	25.02%	-1.855	0.00009
4	0.055	150,000	25%	8,250	141,750	26.15%	24.94%	4.749	0.00058
Total	0.052	600,000	100%	31,550	568,450	100.00%	100.00%		0.00114

Table 2. WOE and IV for X_1 and X_2

For the interaction between X_1 and X_2, we calculate the Index for IV as follows:

		X_2				
		1	*2*	*3*	*4*	*total*
X_1	*A*	0.05103	0.01421	0.08130	0.14385	0.29040
	B	0.00132	0.00019	0.00208	0.00001	0.00360
	C	0.08130	0.04818	0.04522	0.06563	0.24033
	Total	0.13365	0.06258	0.12860	0.20949	0.53432

Table 3. IV for combining X_1 and X_2

The following rules of thumb are often applied for evaluating predictive power of a variable by using IV:

< 0.02: unpredictive
0.02 to 0.1: weak
0.1 to 0.3: medium
0.03 to 0.5: strong

> 0.5: very strong

In our example, X_1 or X_2 used alone is very unpredictive, while a combination of the two becomes strongly predictive. We can now quantify the improvement in IV in two ways: Information Value Increment (IVI) and Information Value Lift (IVL).

$$\text{IVI} = \text{IV}(X_1, X_2) - \max[\text{IV}(X_1), \text{IV}(X_2)] = 0.53432 - 0.00843 = 0.52589$$

$$\text{IVL} = \frac{\text{IV}(X_1, X_2)}{\max[\text{IV}(X_1), \text{IV}(X_2)]} = \frac{0.53432}{0.00843} = 63.1$$

INFORMATION GAIN (IG)

Information Gain is used to gain insights on how much more we can know about a relationship between a predictor and an outcome when the predictor is partitioned or segmented. To calculate Information Gain, we need to employ the concept of entropy first:

$$\text{Entropy} = -\sum_{i=1}^{n} P_i \text{Log}_2 P_i$$

where P_i is the probability of occurrence or nonoccurrence in a segment. Different from Information Value, base 2 is always used for the log function of entropy. Entropy is commonly understood as a measure of impurity of an attribute in relation to an outcome. Lower entropy suggests a better output. In an extreme case, highest entropy can be achieved when an outcome is evenly distributed across an attribute, in which case the attribute has no predictive power.

Let's use the sample example to illustrate the calculation of Information Gain.

Entropy (whole data) = $0.053 \times \log_2 0.053 + (1\text{-}0.053) \times \log_2(1\text{-}0.053) = 0.29727$

Entropy (X_1) = $- (1/3) \times [0.047 \times \log_2 0.049 + (1\text{-}0.049) \times \log_2(1\text{-}0.049)]$
$- (1/3) \times [0.050 \times \log_2 0.050 + (1\text{-}0.050) \times \log_2(1\text{-}0.050)]$
$- (1/3) \times [0.059 \times \log_2 0.059 + (1\text{-}0.062) \times \log_2(1\text{-}0.059)] = 0.29697$

Entropy (X_2) = $- (1/4) \times [0.053 \times \log_2 0.053 + (1\text{-}0.053) \times \log_2(1\text{-}0.053)]$
$- (1/4) \times [0.051 \times \log_2 0.051 + (1\text{-}0.053) \times \log_2(1\text{-}0.051)]$
$- (1/4) \times [0.052 \times \log_2 0.052 + (1\text{-}0.052) \times \log_2(1\text{-}0.052)]$
$- (1/4) \times [0.051 \times \log_2 0.051 + (1\text{-}0.053) \times \log_2(1\text{-}0.051)] = 0.29723$

Entropy (X_1, X_2) = $- (1/12) \times [0.098 \times \log_2 0.098 + (1\text{-}0.098) \times \log_2(1\text{-}0.098)]$
$- (1/12) \times [0.075 \times \log_2 0.075 + (1\text{-}0.075) \times \log_2(1\text{-}0.075)]$
$- (1/12) \times [0.015 \times \log_2 0.015 + (1\text{-}0.015) \times \log_2(1\text{-}0.015)]$
$- (1/12) \times [0.008 \times \log_2 0.008 + (1\text{-}0.008) \times \log_2(1\text{-}0.008)]$
$- (1/12) \times [0.047 \times \log_2 0.047 + (1\text{-}0.047) \times \log_2(1\text{-}0.047)]$
$- (1/12) \times [0.055 \times \log_2 0.055 + (1\text{-}0.055) \times \log_2(1\text{-}0.055)]$
$- (1/12) \times [0.045 \times \log_2 0.045 + (1\text{-}0.045) \times \log_2(1\text{-}0.045)]$
$- (1/12) \times [0.052 \times \log_2 0.052 + (1\text{-}0.052) \times \log_2(1\text{-}0.052)]$
$- (1/12) \times [0.015 \times \log_2 0.015 + (1\text{-}0.015) \times \log_2(1\text{-}0.015)]$
$- (1/12) \times [0.022 \times \log_2 0.022 + (1\text{-}0.022) \times \log_2(1\text{-}0.022)]$
$- (1/12) \times [0.095 \times \log_2 0.095 + (1\text{-}0.095) \times \log_2(1\text{-}0.095)]$
$- (1/12) \times [0.105 \times \log_2 0.105 + (1\text{-}0.105) \times \log_2(1\text{-}0.105)] = 0.28232$

We can calculate Information Gain as follows:

$\mathrm{IG}(X_1) = \text{Entropy (whole data)} - \text{Entropy } (X_1) = 0.29727 - 0.29677 = 0.00031$
$\mathrm{IG}(X_2) = \text{Entropy (whole data)} - \text{Entropy } (X_2) = 0.29727 - 0.29723 = 0.00004$
$\mathrm{IG}(X_1, X_2) = \text{Entropy (whole data)} - \text{Entropy } (X_1, X_2) = 0.29727 - 0.28037 = 0.01690$

Similarly, we can calculate the increment of Information Gain in two ways: Information Gain Increment (IGI) and Information Gain Lift (IGL).

$$\mathrm{IGI} = \mathrm{IG}(X_1, X_2) - \max[\mathrm{IG}(X_1), \mathrm{IG}(X_2)] = 0.01690 - 0.00031 = 0.01659$$

$$\mathrm{IGL} = \frac{\mathrm{IG}(X_1,X_2)}{\max[\ \mathrm{IG}(X_1),\ \ \mathrm{IG}(X_2)]} = \frac{0.01690}{0.00031} = 411.8$$

GAIN RATIO (GR)

Information Gain is biased towards attributes with large number of values, so X_1 and X_2 combined are potentially more predictive than X_1 or X_2 used alone. In the most extreme case, we can maximize Information Gain by treating each observation as a partition. To overcome this bias, several remedial measures have been suggested. Split Information is the most commonly used one.

$$\text{SplitInfo} = \sum_{i=1}^{n} \frac{\text{\# Records in ith bin}}{\text{\# Total Records}} \mathrm{Log}_2 \frac{\text{\# Records in ith bin}}{\text{\# Total Records}}$$

Base 2 is always used for the log function.

Using the same example, we calculate Split Information as follows:

$\mathrm{SplitInfo}(X_1) = -(1/3)\times\log_2(1/3) - (1/3)\times\log_2(1/3) - (1/3)\times\log_2(1/3) = 1.58496$
$\mathrm{SplitInfo}(X_2) = -(1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4) - (1/4)\times\log_2(1/4) = 2$
$\mathrm{SplitInfo}(X_1, X_2) = -(1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12)$
$-(1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12)$
$-(1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12)$
$-(1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12) - (1/12)\times\log_2(1/12) = 3.58496$

The Gain Ratio is calculated as follows:

$$\mathrm{GR} = \frac{\mathrm{IG}}{\mathrm{SplitInfo}}$$

Now we have

$$\mathrm{GR}(X_1) = \frac{\mathrm{GR}(X1)}{\mathrm{SplitInfo}(X1)} = \frac{0.00031}{1.58496} = 0.00019$$

$$\mathrm{GR}(X_2) = \frac{\mathrm{GR}(X2)}{\mathrm{SplitInfo}(X2)} = \frac{0.00004}{2} = 0.00002$$

$$\mathrm{GR}(X_1, X_2) = \frac{\mathrm{IG}(X1,X2)}{\mathrm{SplitInfo}(X1,X2)} = \frac{0.01690}{1.58496} = 0.00471$$

Similarly, we can compute the incremental GR in the following two ways: Gain Ratio Increment (GRI) and Gain Ratio Lift (GRL).

$$GRI = GR(X_1, X_2) - \max[GR(X_1),\ GR(X_2)] = 0.00471 - 0.00019 = 0.00452$$

$$GRL = \frac{GR(X_1,X_2)}{\max[\ GR(X_1),\ \ GR(X_2)]} = \frac{0.00471}{0.00019} = 24.47$$

SEVERAL SITUATIONS

Before moving to the SAS process, we would like to go through several situations by using examples. Some insights will provide valuable implications on the SAS process we are going to introduce.

Situations 1: Weak or No Interaction

If we reverse the performance of X_2=3 and X_2=4, X_1 shows a strong predictive power from A to C. X_2 alone still shows no predictive power. The two variables no longer exhibit an interaction.

X_1 \ X_2	1	2	3	4	total
A	0.098	0.075	0.095	0.105	0.093
B	0.047	0.055	0.045	0.052	0.050
C	0.015	0.022	0.015	0.008	0.015
total	0.053	0.051	0.052	0.055	0.053

Table 4. Example of no or weak interaction

Information Values of X_1 and X_2 are computed as follows:

X_1	% Y	# Records	% Records	# Occurrence	# Non-occurrence	% Occurrence	% Non-Occurences	WOE	Index for IV
A	0.093	200,000	33%	18,650	181,350	59.11%	31.90%	61.676	0.16782
B	0.050	200,000	33%	9,925	190,075	31.46%	33.44%	-6.102	0.00121
C	0.015	200,000	33%	2,975	197,025	9.43%	34.66%	-130.175	0.32844
Total	0.053	600,000	100%	31,550	568,450	100.00%	100.00%		0.49747

X_2	% Y	# Records	% Records	# Occurrence	# Non-occurrence	% Occurrence	% Non-Occurences	WOE	Index for IV
1	0.053	150,000	25%	7,975	142,025	25.28%	24.98%	1.165	0.00003
2	0.051	150,000	25%	7,575	142,425	24.01%	25.05%	-4.262	0.00045
3	0.052	150,000	25%	7,750	142,250	24.56%	25.02%	-1.855	0.00009
4	0.055	150,000	25%	8,250	141,750	26.15%	24.94%	4.749	0.00058
Total	0.052	600,000	100%	31,550	568,450	100.00%	100.00%	0	0.00114

Table 5. WOE and IV for X_1 and X_2

Both IVI and IVL are very limited. Information Gain and Gain Ratio also show no or very limited gains.

$$IVI = IV(X_1, X_2) - \max[IV(X_1)\ , IV(X_2)\ ] = 0.53432 - 0.49747 = 0.03686$$

$$IVL = \frac{IV(X_1,X_2)}{\max[\ IV(X_1),IV(X_2)]} = \frac{0.53432}{0.49747} = 1.07$$

$$IGI = IG(X_1, X_2) - \max[IG(X_1), IG(X_1)] = 0.01690 - 0.0325 = -0.0156$$

$$IGI = \frac{IG(X_1,X_2)}{\max[\ IG(X_1),IG(X_2)]} = \frac{0.01690}{0.0325} = 0.52$$

$$GRI = GR(X_1, X_2) - \max[GR(X_1), GR(X_2)] = 0.00471 - 0.02049 = -0.01577$$

$$\text{GRL} = \frac{\text{GR}(X_1,X_2)}{\max[\ \text{GR}(X_1),\text{GR}(X_2)]} = \frac{0.00471}{0.02049} = 0.23$$

Situation 2: Levels of Probabilities Matter

In the business world, different events have different levels of probabilities[1]. For example, response rate of a direct mail marketing campaign is usually less than 1%. Credit cards usually have an activation rate of 40%-60% and a charge-off rate around 5%.

Let's experiment with different levels of probabilities without changing the relative predictive power of each variable or cell. The purpose is to see how they can have differing implications on Information Value Gain, Information Gain and Gain Ratio. Our experiment sets probabilities from 0.0053 to 0.4733[2], with some interesting observations:

- Information Value Increment (IVI) and (IVL) remains more stable across most scales of probabilities till reaching a very high probability such as > 0.1.

- IVL and IGL in general look more stable than their counterparts IVI and IGI. In modeling or analytics, an increase of IV from 0.02 to 0.05 is more meaningful than an increase from 0.0002 to 0.001 as the letter still shows little use despite a very high IVL. Therefore, we should consider IVI as an important measure.

- When Gain Ratio for a predictor is very low, a good Gain Ratio Increment (GRI) by interaction will cause substantial Gain Ratio Lift (GRL) because the latter uses a very low number for division.

	Information Value (IV)		Information Gain (IG)		Gain Ratio (GR)	
avg Y	IVI	IVL	IGI	IGL	GRI	GRL
0.00526	0.4880	472.6	0.0016	413.3	0.0004	9,969,269
0.01052	0.4927	472.1	0.0032	413.1	0.0009	4,955,759
0.01578	0.4975	471.6	0.0049	412.9	0.0014	3,284,612
0.02103	0.5023	471.1	0.0065	412.8	0.0018	2,449,056
0.02629	0.5073	470.7	0.0082	412.6	0.0023	1,947,736
0.03155	0.5123	470.2	0.0099	412.4	0.0028	1,613,535
0.03681	0.5174	469.7	0.0116	412.3	0.0032	1,374,831
0.04207	0.5226	469.2	0.0133	412.1	0.0037	1,195,812
0.04733	0.5278	468.8	0.0151	412.0	0.0042	1,056,583
0.05258	0.5332	468.3	0.0169	411.8	0.0047	945,207
0.05784	0.5386	467.8	0.0186	411.6	0.0052	854,089
0.06310	0.5442	467.4	0.0204	411.5	0.0057	778,163
0.06836	0.5498	466.9	0.0223	411.3	0.0062	713,925
0.07362	0.5555	466.5	0.0241	411.2	0.0067	658,869
0.07888	0.5613	466.0	0.0260	411.0	0.0072	611,159
0.08939	0.5733	465.1	0.0297	410.7	0.0083	532,594
0.10517	0.5921	463.8	0.0356	410.3	0.0099	444,240
0.13146	0.6257	461.7	0.0457	409.7	0.0127	344,178
0.15775	0.6627	459.8	0.0565	409.1	0.0157	277,553
0.21033	0.7487	456.6	0.0803	408.3	0.0223	194,483
0.26292	0.8559	454.6	0.1075	408.2	0.0299	144,915
0.31550	0.9934	454.9	0.1392	409.1	0.0387	112,170
0.36808	1.1777	459.3	0.1772	412.0	0.0493	89,148
0.42067	1.4436	472.9	0.2245	418.4	0.0624	72,405
0.47325	1.9067	515.9	0.2878	433.3	0.0801	60,385

Table 6. Different levels of probabilities

We can experiment with multiplication of IVI with IVL, IGI with IGL and GRI with GRL as follows:

$$\text{IV\_gain} = \{\text{IV}(X_1, X_2) - \max[\text{IV}(X_1)\ ,\ \text{IV}(X_2)\ ]\} \times \frac{\text{IV}(X_1, X_2)}{\max[\ \text{IV}(X_1),\ \ \text{IV}(X_2)]}$$

[1] Information Gain and Gain Ratio in many online tutorials or in textbooks are much higher, due to their use of examples of very high probabilities, such as Titanic Survival data or Fisher's Iris flower data.

[2] We applied a series of ascending multipliers to the same data as shown in Table 1. The simulation by multipliers stopped when any individual cell showed a probability > 1 or when overall average probability exceeded 0.5. For higher overall average probabilities, we can always redefine the outcome as 1-y in order to limit the overall probability below 0.5.

$$\text{IG\_gain} = \{\text{IG}(X_1, X_2) - \max[\text{IG}(X_1), \text{IG}(X_2)]\} \times \frac{\text{IG}(X_1, X_2)}{\max[\text{IG}(X_1), \text{IG}(X_2)]}$$

$$\text{GR\_gain} = \{\text{GR}(X_1, X_2) - \max[\text{GR}(X_1), \text{GR}(X_2)]\} \frac{\text{GR}(X_1, X_2)}{\max[\text{GR}(X_1), \text{GR}(X_2)]}$$

The first two in general are more useful than GR_gain as GRI allows negative values and could make the results difficult to be explained. All pairs of variables will be ranked for each measure from highest to lowest. A composite gain is computed by averaging selected ranks as follows:

$$\text{composite\_gain\_rank} = \text{mean}(\text{IVI\_rank}, \text{IV\_gain\_rank}, \text{GRL\_rank}, \text{IG\_gain\_rank})$$

You can experiment with any combinations of measures to form your own composite gain.

Situation 3: Expanding the methodology to continuous outcomes

Strictly speaking, Information Value, Information Gain and Gain Ratio were developed for classified outcomes. However, since all comparisons are based on aggregated data, we can creatively extend the same methodology to analysis of continuous outcomes.

The following is a highly stylized example of sales volume by three teams, each with 10 members.

Section A		Section B		Section C	
Salesperson ID	Sales Volume	Salesperson ID	Sales Volume	Salesperson ID	Sales Volume
1	$50,000	20	$43,000	10	$13,000
27	$62,000	2	$35,000	21	$24,000
13	$45,000	19	$25,000	11	$21,000
17	$50,000	8	$25,000	28	$5,000
25	$34,000	16	$28,000	5	$8,000
6	$23,000	3	$24,000	22	$5,000
7	$20,000	15	$13,000	12	$11,000
24	$16,000	9	$24,000	30	$23,000
14	$54,000	23	$33,000	29	$12,000
18	$22,000	26	$13,000	4	$9,000
total	$376,000	total	$263,000	total	$131,000

Table 8. Example of a continuous outcome

We suggest transforming a continuous outcome to a binary- like one in the following way:

- Take a percentile (98th, 95th, 90th, 75th, mean or median) of a continuous outcome as the full realization of the outcome. In our example, we choose $50,000 as the cap and consider Sales Volume ≥ $50,000 as a full achiever. The cap is applied to suppress the impact by outliers.
- Divide all outcomes by the chosen high cap ($50,000 in our example). For each salesperson, we get two fractions: % achieved and % unachieved.

Section A			Section B			Section C		
Salesperson ID	% Achieved	% Unachieved	Salesperson ID	% Achieved	% Unachieved	Salesperson ID	% Achieved	% Unachieved
1	1	0	20	0.86	0.14	10	0.26	0.74
27	1	0	2	0.7	0.3	21	0.48	0.52
13	0.9	0.1	19	0.5	0.5	11	0.42	0.58
17	1	0	8	0.5	0.5	28	0.1	0.9
25	0.68	0.32	16	0.56	0.44	5	0.16	0.84
6	0.46	0.54	3	0.48	0.52	22	0.1	0.9
7	0.4	0.6	15	0.26	0.74	12	0.22	0.78
24	0.32	0.68	9	0.48	0.52	30	0.46	0.54
14	1	0	23	0.66	0.34	29	0.24	0.76
18	0.44	0.56	26	0.26	0.74	4	0.18	0.82
total	0.72	0.28	total	0.53	0.47	total	0.26	0.74

Table 9. Converting a continuous outcome to a binary-like outcome

- Aggregate both fractions by segments (or cells), similar to aggregating occurrence and non-occurrence for a binary outcome.

Situation 4: High-order interaction
A three-way interaction is an interaction among three variables, i.e., a two-way interaction differs depending on the level of a third variable. A four-way interaction can be understood similarly. Even though programming for uncovering interactions exceeding two dimensions is beyond what this paper intends, interested users can apply the same methodology to explore high-order interactions.

THE SAS PROCESS

This paper introduces a SAS process that will calculate the Information Value, Information Gain and Gain Ratio for each pair of attributes in relation to the outcome and compare them to the same measures calculated for each variable in the pair.

Once a data set has been prepared, you only need to define the macro values at the beginning of the program in order to run the process.

```
libname yourlib "H:\project for modeling\data";                  /* libname */
%let datalib=yourlib;                                         /* library name */
%let inset=term36;                                        /* data set to be used */
%let y=pp;                                                   /* Target variable */
%let yformat=10.2;                                  /* format of target variables */
%let ytype=binary;   /* binary or continue for binary or contiuous outcome */
%let ycap=95;            /* percentile of high cap for continuous outcomes */
%let vartxt=charx1 charx2;                /* list of all character variables */
%let varnum=numx1 numx2;                    /* list of all numeric variables */
%let binnum=6;                       /* number of bins for numeric variables */
%let graphfolder=H:\project for modeling\output; /* folder for output file */
%let graphname=check_overlay_pair;        /* name of the output file in pdf */
%let missingnum=-9999999999;            /* filler for missing numeric value */
%let missingchar=_MISSING_;           /* filler for missing character value */
%let heaty=(green yellow orange red);/* color pattern for performance data */
%let heatdist=TwoColorRamp;        /* color pattern for record distribution */
```

You can experiment with above macro values to find a preferred set of bins, color patterns, etc. The following tips might help you to understand the codes better.

- You do not need to impute missing values for running the program. They are treated as a separate category in each variable.
- If only numeric variables or character variables are included for analysis, the macro values for `vartxt` or `varnum` can be left blank. The SAS program will make an automatic skip.
- We suggest limiting the number of bins for continuous variables, such as ≤ 10, in order to reduce bias towards a large number of bins.
- The SAS process will generate (N-1)×N/2 pairs of graphs, where N is the number of variables. For example, 3 attributes will generate 3 graphs, while 50 attributes will generate 1225 graphs. Increasing the number of variables will exponentially increase the processing time. We suggest reducing number of variables that highly overlap in business meanings in order to shorten processing time without losing value of the data.

THE SAS OUTPUT

The SAS process will generate two outputs for you to review.

- A table that lists all pairs of variables examined, ranked by composite_gain_rank.

x1	x2	IV_x1	IV_x2	IV_x1_x2	IVI	IVL	IG_x1	IG_x2	IG_x1_x2	IGI	IGL	GR_x1	GR_x2	GR_x1_x2	GRI	GRL	composite_gain_rank
numx2	charx1	0.02874	0.0223	0.08386	0.05512	2.91788	0.00037	0.00029	0.00109	0.00072	2.94595	0.000143	0.000138	0.0002329	8.98E-05	1.62702	2
numx2	numx3	0.02874	0.20213	0.29806	0.09593	1.4746	0.00037	0.00265	0.00387	0.00122	1.46038	0.000143	0.001025	0.0007515	-0.00027	0.73289	2
numx2	charx2	0.02874	0.0073	0.07424	0.0455	2.58316	0.00037	9.37E-05	0.00114	0.00077	3.08108	0.000143	0.0001	0.0003248	0.000182	2.26877	2.6667
numx3	charx2	0.20213	0.0073	0.26637	0.06424	1.31782	0.00265	9.37E-05	0.00362	0.00097	1.36604	0.001025	0.0001	0.0010756	5.02E-05	1.04898	3.3333
numx3	charx1	0.20213	0.0223	0.2462	0.04407	1.21803	0.00265	0.00029	0.00321	0.00056	1.21132	0.001025	0.000138	0.0006927	-0.00033	0.67556	5
numx1	charx1	0.00745	0.0223	0.0413	0.019	1.85202	9.76E-05	0.00029	0.00054	0.00025	1.86207	0.000151	0.000138	0.000197	4.56E-05	1.30097	6.6667
numx1	numx3	0.00745	0.20213	0.23016	0.02803	1.13867	9.76E-05	0.00265	0.00301	0.00036	1.13585	0.000151	0.001025	0.0009359	-9E-05	0.91272	7.3333
charx1	charx2	0.0223	0.0073	0.03885	0.01655	1.74215	0.00029	9.37E-05	0.00057	0.00028	1.96552	0.000138	0.0001	0.000191	5.28E-05	1.38214	7.6667
numx1	charx2	0.00745	0.0073	0.01977	0.01232	2.65369	9.76E-05	9.37E-05	0.00025	0.000152	2.56253	0.000151	0.0001	0.000159	7.61E-06	1.05026	8.3333
numx1	numx2	0.00745	0.02874	0.02874	0	1	9.76E-05	0.00037	0.00037	0	1	0.000151	0.000143	0.0001431	-8.3E-06	0.9455	10

Table 10. Ranking interaction effects

- A collage of 2x2 tables and heat maps that show distribution of observations and patterns of interaction effects. The macro value `graphfolder` specifies where the pdf graphs are stored.

CONCLUSION

Interaction effects conceal many hidden treasures in data. Uncovering these treasures will improve the quality of models and add strengths to analytics. This paper introduces a useful process for SAS users who are interested in this exploration.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at

Alec Zhixiao Lin
VP of Modeling
Loan Depot
26642 Towne Center Drive
Foothill Ranch, CA 92610
Email: alecindc@gmail.com
Web: www.linkedin.com/pub/alec-zhixiao-lin/25/708/261/

APPENDIX

```
      ** erase all stored macro values so that it will interfere with rerun;
proc datasets lib=work memtype=data nolist;
delete _: ; quit;

libname yourlib "H:\project for modeling\data";                          /* libname */
%let datalib=yourlib;                                                 /* library name */
%let inset=term36;                                                 /* data set to be used */
%let y=pp;                                                            /* Target variable */
%let yformat=10.2;                                              /* format of target variables */
%let ytype=binary;           /* binary or continu for binary or contiuous outcome */
%let ycap=95;                   /* percentile of high cap for continuous outcomes */
%let vartxt=charx1 charx2;                          /* list of all character variables */
%let varnum=numx1 numx2;                              /* list of all numeric variables */
%let binnum=6;                                /* number of bins for numeric variables */
%let graphfolder=H:\project for modeling\output;        /* folder for output file */
%let graphname=check_overlay_pair;                  /* name of the output file in pdf */
%let missingnum=-9999999999;                      /* filler for missing numeric value */
%let missingchar=_MISSING_;                    /* filler for missing character value */
```

```
%let heaty=(green yellow orange red);       /* color pattern for performance data */
%let heatdist=TwoColorRamp;              /* color pattern for record distribution */

                    ** No need to change the codes below;
data check_contents;
retain &varnum;
set &datalib..&inset(keep=&varnum obs=1); run;

proc contents data=check_contents varnum out=check_contents2 noprint; run;
proc sort data=check_contents2(keep=name varnum)
out=checkfreq(rename=(name=tablevar)); by varnum; run;

data varcnt; set checkfreq; varcnt+1; run;

proc sql; create table vcnt as select count(*) as vcnt from varcnt; quit;
data _null_; set vcnt; call symputx('numcnt', vcnt); run;

proc sql noprint; select tablevar into :vnum1-:vnum&numcnt from varcnt; quit;
proc sql noprint; select tablevar into :wnum1-:wnum&numcnt from varcnt; quit;

data check_contents; retain &vartxt;
set &datalib..&inset(keep=&vartxt obs=1); run;

proc contents data=check_contents varnum out=check_contents2 noprint; run;
proc sort data=check_contents2(keep=name varnum)
out=checkfreq(rename=(name=tablevar)); by varnum; run;

data varcnt; set checkfreq; varcnt+1; run;

proc sql; create table vcnt as select count(*) as vcnt from varcnt; quit;
data _null_; set vcnt; call symputx('txtcnt', vcnt); run;

proc sql noprint; select tablevar into :vtxt1-:vtxt&txtcnt from varcnt; quit;
proc sql noprint; select tablevar into :wtxt1-:wtxt&txtcnt from varcnt; quit;

%macro locglobe;
data _null_;
set macrotemp;
call symputx('tot_bad', tot_bad);
call symputx('tot_good', tot_good);
call symputx('tot_both', tot_both);
call symputx('avgy', avgy);
call symputx('entropytemp', entropytemp);
run;
%global entropybase;
%let entropybase=&entropytemp;
%mend;

%macro binary;
proc sql noprint;
create table macrotemp as select
       sum(case when &y=1 then 1 else 0 end) as tot_bad,
       sum(case when &y=0 then 1 else 0 end) as tot_good,
       count(*) as tot_both,
       sum(case when &y=1 then 1 else 0 end)/count(*) as avgy,
       -(sum(case when &y=1 then 1 else 0 end)/count(*))*log2(sum(case when &y=1 then
1 else 0 end)/count(*))
       -(sum(case when &y=0 then 1 else 0 end)/count(*))*log2(sum(case when &y=0 then
1 else 0 end)/count(*)) as entropytemp
from &datalib..&inset; quit;

%locglobe;
%macro ivbinary;
```

```
%macro woe(xlabel, xrank);
proc sql;
create table &xlabel as select
        &xrank as tier,
        count(*) as cnt,
             count(*)/&tot_both as cnt_pct,
        sum(case when &y=0 then 1 else 0 end) as sum_good,
        sum(case when &y=0 then 1 else 0 end)/&tot_good as dist_good,
        sum(case when &y=1 then 1 else 0 end) as sum_bad,
        sum(case when &y=1 then 1 else 0 end)/&tot_bad as dist_bad,
        log((sum(case when &y=0 then 1 else 0 end)/&tot_good)/(sum(case when &y=1 then
1 else 0 end)/&tot_bad))*100 as woe,
        ((sum(case when &y=0 then 1 else 0 end)/&tot_good)-(sum(case when &y=1 then 1
else 0 end)/&tot_bad))
                    *log((sum(case when &y=0 then 1 else 0 end)/&tot_good)/(sum(case
when &y=1 then 1 else 0 end)/&tot_bad)) as pre_iv,
             sum(case when &y=1 then 1 else 0 end)/count(*) as outcomesum,
        -(count(*)/&tot_both)*(sum(case when &y=1 then 1 else 0
end)/count(*)*log2(sum(case when &y=1 then 1 else 0 end)/count(*))
                              +sum(case when &y=0 then 1 else 0
end)/count(*)*log2(sum(case when &y=0 then 1 else 0 end)/count(*))) as entropy,
        -(count(*)/&tot_both)*log2(count(*)/&tot_both) as splitinfo
   from xdata
   group by &xrank; quit;

proc sql noprint; select min(woe) into :minwoe from &xlabel where woe ne .; quit;
proc sql noprint; select max(woe) into :maxwoe from &xlabel where woe ne .; quit;
proc sql noprint; select min(outcomesum) into :minpery from &xlabel where woe ne .;
quit;
proc sql noprint; select max(outcomesum) into :maxpery from &xlabel where woe ne .;
quit;

data &xlabel; set &xlabel;
if woe=. and outcomesum < &minpery then woe=&maxwoe;
if woe=. and outcomesum > &maxpery then woe=&minwoe;
run;
%mend;
%woe(x1, rankyaxis);
%woe(x2, rankxaxis);
%woe(x1_x2, x1_x2_rank_seq);
%mend;
%ivbinary;
%mend;

%macro continu;
proc univariate data=&datalib..&inset noprint;
var equifaxficoscore;
output out=ycapped PCTLPTS=1 &ycap PCTLPRE=ycapped; run;

proc sql noprint; select ycapped&ycap into :ycapped from ycapped; quit;
proc sql noprint; select min(&y) into :miny from &datalib..&inset; quit;

data continuetemp;
set xdata;
tempy=min(0.98, (&y-&miny)/(&ycapped-&miny)); run;

proc sql noprint;
create table macrotemp as select
       sum(tempy) as tot_bad,
       sum(1-tempy) as tot_good,
       count(*) as tot_both,
       sum(tempy)/count(*) as avgy,
       -(sum(tempy)/count(*))*log2(sum(tempy)/count(*))
```

```
          -(sum(1-tempy)/count(*))*log2(sum(1-tempy)/count(*)) as entropytemp
from continuetemp; quit;

%locglobe;
%macro ivconti;
%macro woe(xlabel, xrank);
proc sql;
create table &xlabel as select
        &xrank as tier,
        count(*) as cnt,
             count(*)/&tot_both as cnt_pct,
        sum(1-tempy) as sum_good,
        sum(1-tempy)/&tot_good as dist_good,
        sum(tempy) as sum_bad,
        sum(tempy)/&tot_bad as dist_bad,
        log((sum(1-tempy)/&tot_good)/(sum(tempy)/&tot_bad))*100 as woe,
        ((sum(1-tempy)/&tot_good)-(sum(tempy)/&tot_bad))
                    *log((sum(1-tempy)/&tot_good)/(sum(tempy)/&tot_bad)) as pre_iv,
             sum(tempy)/count(*) as outcomesum,
        -(count(*)/&tot_both)*(sum(tempy)/count(*)*log2(sum(tempy)/count(*))
                              +sum(1-tempy)/count(*)*log2(sum(1-tempy)/count(*))) as
entropy,
        -(count(*)/&tot_both)*log2(count(*)/&tot_both) as splitinfo
   from continuetemp
   group by &xrank; quit;

proc sql noprint; select min(woe) into :minwoe from &xlabel where woe ne .; quit;
proc sql noprint; select max(woe) into :maxwoe from &xlabel where woe ne .; quit;
proc sql noprint; select min(outcomesum) into :minpery from &xlabel where woe ne .;
quit;
proc sql noprint; select max(outcomesum) into :maxpery from &xlabel where woe ne .;
quit;

data &xlabel; set &xlabel;
if woe=. and outcomesum < &minpery then woe=&maxwoe;
if woe=. and outcomesum > &maxpery then woe=&minwoe;
run;
%mend;
%woe(x1, rankyaxis);
%woe(x2, rankxaxis);
%woe(x1_x2, x1_x2_rank_seq);
%mend;
%ivconti;
%mend;

%macro xtabnum(x1, x2);
proc rank data=&datalib..&inset groups=&binnum out=xdata;
var &x1 &x2;
ranks rankyaxis rankxaxis; run;

data xdata; set xdata;
rankyaxis=rankyaxis+1;
rankxaxis=rankxaxis+1;
if rankyaxis=. then do; rankyaxis=0; &x1=&missingnum; end;
if rankxaxis=. then do; rankxaxis=0; &x2=&missingnum; end;
x1_x2_rank_seq=1000*(rankyaxis+1)+rankxaxis; run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum; run;
```

```
proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;

data check_mean_cnt;
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;

format y_mean &yformat;
informat y_mean &yformat; run;

proc means data=xdata median min max nway noprint;
class rankyaxis;
var &x1;
output out=check_x1(drop=_type_ _freq_)
    median=&x1 min=min_yaxis max=max_yaxis; run;

proc means data=xdata median min max nway noprint;
class rankxaxis;
var &x2;
output out=check_x2(drop=_type_ _freq_)
    median=&x2 min=min_xaxis max=max_xaxis; run;

proc sql;
create table graph_data_num
as select a.*,
          b.&x1, min_yaxis, max_yaxis,
             c.&x2, min_xaxis, max_xaxis
from check_mean_cnt a,
     check_x1 b,
     check_x2 c
where a.rankyaxis=b.rankyaxis
  and a.rankxaxis=c.rankxaxis; quit;

proc sort data=graph_data_num; by &x2 &x1; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=graph_data_num NOAUTOLEGEND ;
XAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title; footnote; run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=graph_data_num NOAUTOLEGEND ;
XAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title; footnote; run; quit;
ods layout end;

%&ytype;

proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1, :ig_x1, :si_x1 from x1; quit;
```

```
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x2, :ig_x2, :si_x2 from x2; quit;
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1_x2, :ig_x1_x2, :si_x1_x2 from x1_x2; quit;

data num_iv_&i._&j;
length x1 $32.;
length x2 $32.;
x1="&&vnum&i";
x2="&&wnum&j";
IV_x1=&iv_x1;
IV_x2=&iv_x2;
IV_x1_x2=&iv_x1_x2;
IVI=iv_x1_x2-max(iv_x1, iv_x2);
IVL=iv_x1_x2/max(iv_x1, iv_x2);

IG_x1=&ig_x1;
IG_x2=&ig_x2;
IG_x1_x2=&ig_x1_x2;
IGI=IG_x1_x2-max(IG_x1, IG_x2);
IGL=IG_x1_x2/max(IG_x1, IG_x2);

si_x1=&si_x1;
si_x2=&si_x2;
si_x1_x2=&si_x1_x2;

GR_x1=ig_x1/&si_x1;
GR_x2=ig_x2/&si_x2;
GR_x1_x2=ig_x1_x2/&si_x1_x2;
GRI=GR_x1_x2-max(GR_x1, GR_x2);
GRL=GR_x1_x2/max(GR_x1, GR_x2); run;
%mend;

%macro xtabtxt(x1, x2);
data xdata; set &datalib..&inset;
if compress(&x1)=' ' then &x1="&missingchar";
if compress(&x2)=' ' then &x2="&missingchar";
rankyaxis=&x1;
rankxaxis=&x2;
x1_x2_rank_seq=compress(rankyaxis||rankxaxis);
run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum;run;

proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;

data gragh_data_txt(rename=(rankyaxis=&x1 rankxaxis=&x2));
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;
format y_mean &yformat;
informat y_mean &yformat; run;
```

```
proc sort data=gragh_data_txt; by y_mean; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txt NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title; footnote; run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txt NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title; footnote; run; quit;
ods layout end;

%&ytype;

proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1, :ig_x1, :si_x1 from x1; quit;
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x2, :ig_x2, :si_x2 from x2; quit;
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1_x2, :ig_x1_x2, :si_x1_x2 from x1_x2; quit;

data txt_iv_&m._&n;
length x1 $32.;
length x2 $32.;
x1="&&vtxt&m";
x2="&&vtxt&n";
IV_x1=&iv_x1;
IV_x2=&iv_x2;
IV_x1_x2=&iv_x1_x2;
IVI=iv_x1_x2-max(iv_x1, iv_x2);
IVL=iv_x1_x2/max(iv_x1, iv_x2);

IG_x1=&ig_x1;
IG_x2=&ig_x2;
IG_x1_x2=&ig_x1_x2;
IGI=IG_x1_x2-max(IG_x1, IG_x2);
IGL=IG_x1_x2/max(IG_x1, IG_x2);

si_x1=&si_x1;
si_x2=&si_x2;
si_x1_x2=&si_x1_x2;

GR_x1=ig_x1/&si_x1;
GR_x2=ig_x2/&si_x2;
GR_x1_x2=ig_x1_x2/&si_x1_x2;
GRI=GR_x1_x2-max(GR_x1, GR_x2);
GRL=GR_x1_x2/max(GR_x1, GR_x2); run;
%mend;

%macro xtabtxtnum(x1, x2);
data xdata; set &datalib..&inset;
if compress(&x2)=' ' then &x2="&missingchar";
rankxaxis=&x2; run;

proc rank data=xdata groups=&binnum out=xdata;
```

```
var &x1;
ranks rankyaxis; run;

data xdata; set xdata;
rankyaxis=rankyaxis+1;
if rankyaxis=. then do; rankyaxis=0; &x1=&missingnum; end;
x1_x2_rank_seq=compress(rankxaxis||'_'||rankyaxis); run;

proc summary data=xdata nway;
var &y;
class rankyaxis rankxaxis/ missing order=data;
output out=tempcheckdata
       sum=y_sum; run;

proc sql noprint;
select case when min(y_sum/_freq_) ge 0 and max(y_sum/_freq_) le 1 then
       int(1000/sum(max(y_sum/_freq_/2), min(y_sum/_freq_/2))) else 1000 end into
:multiplier
from tempcheckdata; quit;

data check_mean_cnt(rename=(rankxaxis=&x2));
set tempcheckdata;
y_mean=y_sum/_freq_;
y_n=_freq_;
y_mean_int=int(y_mean*&multiplier);
if y_mean_int in (0, 1) then y_mean_int=1;

format y_mean &yformat;
informat y_mean &yformat; run;

proc means data=xdata median min max nway noprint;
class rankyaxis;
var &x1;
output out=check_x1(drop=_type_ _freq_)
    median=&x1 min=min_yaxis max=max_yaxis; run;

proc sql;
create table gragh_data_txtnum
as select a.*,
          b.&x1, min_yaxis, max_yaxis
from check_mean_cnt a,
     check_x1 b
where a.rankyaxis=b.rankyaxis; quit;

proc sort data=gragh_data_txtnum; by &x1 y_mean; run;

ods layout Start width=10in height=8in;
ods region x=0% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txtnum NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_n discretex discretey colormodel=&heatdist outline;
text x=&x2 y=&x1 text=y_n / textattrs=(size=5pt);
title;
footnote; run;

ods region x=50% y=0% width=50% height=30%;
proc sgplot data=gragh_data_txtnum NOAUTOLEGEND ;
XAXIS DISCRETEORDER=formatted labelattrs=(size=8pt) valueattrs=(size=5pt);
YAXIS TYPE=discrete DISCRETEORDER=data labelattrs=(size=8pt) valueattrs=(size=5pt);
heatmap x=&x2 y=&x1 / freq=y_mean_int discretex  discretey colormodel=&heaty outline;
text x=&x2 y=&x1 text=y_mean / textattrs=(size=5pt);
title;
```

```
footnote; run; quit;
ods layout end;

%&ytype;

proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1, :ig_x1, :si_x1 from x1; quit;
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x2, :ig_x2, :si_x2 from x2; quit;
proc sql noprint; select sum(pre_iv), &entropybase - sum(entropy), sum(splitinfo) into
:iv_x1_x2, :ig_x1_x2, :si_x1_x2 from x1_x2; quit;

data txtnum_iv_&a._&b;
length x1 $32.;
length x2 $32.;
x1="&&vnum&a";
x2="&&wtxt&b";
IV_x1=&iv_x1;
IV_x2=&iv_x2;
IV_x1_x2=&iv_x1_x2;
IVI=iv_x1_x2-max(iv_x1, iv_x2);
IVL=iv_x1_x2/max(iv_x1, iv_x2);

IG_x1=&ig_x1;
IG_x2=&ig_x2;
IG_x1_x2=&ig_x1_x2;
IGI=IG_x1_x2-max(IG_x1, IG_x2);
IGL=IG_x1_x2/max(IG_x1, IG_x2);

si_x1=&si_x1;
si_x2=&si_x2;
si_x1_x2=&si_x1_x2;

GR_x1=ig_x1/&si_x1;
GR_x2=ig_x2/&si_x2;
GR_x1_x2=ig_x1_x2/&si_x1_x2;
GRI=GR_x1_x2-max(GR_x1, GR_x2);
GRL=GR_x1_x2/max(GR_x1, GR_x2); run;
%mend;

ods pdf file="&graphfolder\&graphname..pdf" style=myfont;
%macro dealtxt;
%if %sysfunc(countw(&vartxt dummymiss)) > 1 %then %do;
     %macro overlaytxt;
     %do m=1 %to &txtcnt;
       %do n=1 %to &txtcnt;
         %if &m < &n %then %do;
          %xtabtxt(&&vtxt&m, &&wtxt&n);
         %end;
      %end;
    %end;
    %mend overlaytxt;
     %overlaytxt;
%end;
%mend;
%dealtxt;

%macro dealnum;
%if %sysfunc(countw(&varnum dummymiss)) > 1 %then %do;
     %macro overlaynum;
     %do i=1 %to &numcnt;
       %do j=1 %to &numcnt;
        %if &i < &j %then %do;
```

```
          %xtabnum(&&vnum&i, &&wnum&j);
        %end;
      %end;
    %end;
    %mend overlaynum;
    %overlaynum;
%end;
%mend;
%dealnum;

%macro dealtxtnum;
%if %sysfunc(countw(&vartxt dummymiss)) > 1 and %sysfunc(countw(&varnum dummymiss)) >
1 %then %do;
     %macro overlaytxtnum;
     %do a=1 %to &numcnt;
       %do b=1 %to &txtcnt;
          %xtabtxtnum(&&vnum&a, &&wtxt&b);
      %end;
    %end;
    %mend overlaytxtnum;
     %overlaytxtnum;
%end;
%mend;
%dealtxtnum;
ods pdf close;

%macro numiv; %if %sysfunc(exist(num_iv_1_2)) %then %do; num_iv_: %end; %mend;
%macro txtiv; %if %sysfunc(exist(txt_iv_1_2)) %then %do; txt_iv_: %end; %mend;
%macro txtnum; %if %sysfunc(exist(txtnum_iv_1_1)) %then %do; txtnum_iv_: %end; %mend;

data all_iv; set %numiv %txtiv %txtnum; run;
proc sort data=all_iv; by descending IVL; run;
data all_iv; set all_iv; IVL_rank+1; run;
proc sort data=all_iv; by descending IGL; run;
data all_iv; set all_iv; IGL_rank+1; run;
proc sort data=all_iv; by descending GRL; run;
data all_iv; set all_iv; GRL_rank+1; run;
proc sort data=all_iv; by descending ivi; run;
data all_iv; set all_iv; ivi_rank+1; IVI_IVL=IVI*IVL; IGI_IGL=IGI*IGL; run;
proc sort data=all_iv; by descending IVI_IVL; run;
data all_iv; set all_iv; IV_gain_rank+1; run;
proc sort data=all_iv; by descending IGI_IGL; run;
data all_iv; set all_iv; IG_gain_rank+1;
composite_gain_rank=mean(IVI_rank, GRL_rank, IV_gain_rank, IG_gain_rank); run;
proc sort data=all_iv; by composite_gain_rank; run;
proc print data=all_iv;
var x1 x2 IV_x1 IV_x2 IV_x1_x2 IVI IVL IG_x1 IG_x2 IG_x1_x2 IGI IGL GR_x1 GR_x2
GR_x1_x2 GRI GRL composite_gain_rank;
run;
```