



# Evidence and Information Value in R



**KIM LARSEN**

August 13, 2015 - San Francisco, CA



[Tweet this post!](#)



[Post on LinkedIn](#)

Binary classification models are perhaps the most common use-case in predictive analytics. The reason is that many key client actions across a wide range of industries are binary in nature, such as defaulting on a loan, clicking on an ad, or terminating a subscription.

Prior to building a binary classification model, a common step is to perform variable screening and exploratory data analysis. This is the step where we get to know the data and weed out variables that are either ill-conditioned or simply contain no information that will help us predict the action of interest. Note that the purpose of this step should not to be confused with that of multiple-variable selection techniques, such as stepwise regression and lasso, where the variables that go into the final model are selected. Rather, this is a precursory step designed to ensure that the approaches deployed during the final modeling phases are set up for success.

The *weight of evidence* (WOE) and *information value* (IV) provide a great framework for for exploratory analysis and variable screening for binary classifiers. WOE and IV have been used extensively in the credit risk world for several decades, and the underlying theory dates back to the 1950s. However, it is still not widely used outside the credit risk world and it is a somewhat underserved area in R.

WOE and IV enable one to:



- Visualize the correlations between the predictive variables and the binary outcome.
- Seamlessly compare the strength of continuous and categorical variables without creating dummy variables.
- Seamlessly handle missing values without imputation.
- Assess the predictive power of missing values.

In this paper we will provide a brief description of WOE and IV as well as the Information R package, which provides an easy way to perform variable screening and exploratory analysis with WOE and IV for traditional binary classifiers as well as uplift models.

# What Are WOE and IV?

## Background

Weight of evidence (WOE) and information value are closely related to concepts from *information theory* where one of the goals is to understand the uncertainty involved in predicting the outcome of random events given varying degrees of knowledge of other variables (see [2], [3], and [4]). Thus this is a perfect framework for variable screening and exploratory analysis for predictive modeling.

Information theory kicked off when Claude Shannon (1948) wrote down an expression for the *entropy* of a probability distribution. Entropy is often described as the level of “disorder,” but in the context of information theory it is better to think of it as one’s level of uncertainty. High entropy means high uncertainty (you don’t



isn't worth very much. This turns out to be the maximum amount of entropy for an event with two possible outcomes. Suppose instead that you know the coin is weighted so that it comes up heads 90 percent of the time. Now your job of making a prediction is easier – Shannon's formula tells you exactly how much information you've gained compared to the previous situation.

Mutual information is a way of summarizing much knowing the value of one random variable tells you about another random variable (Shannon + Weaver 1949). If  $X$  and  $Y$  are independent, the mutual information is zero. Any correlations (positive, negative, or nonlinear) will result in positive mutual information. It turns out the mutual information of  $X$  and  $Y$  is nicely expressed in terms of entropy ( $H$ ):

$$\text{MI}(X, Y) = H(X) + H(Y) - H(X, Y).$$

Mutual information is the difference in predictability that you get from knowing the joint distribution  $p(X, Y)$  compared to knowing only the two marginal distributions  $p(X)$  and  $p(Y)$ .

Information value (IV), which we will define below, is quite closely connected to mutual information (MI), and both are intended to do the same thing: summarize how much knowing the value of  $X$  in a given trial helps you predict the value of  $Y$  for the same trial.

## Setup

Let's say that we have a binary dependent variable  $Y$  and a set of predictive variables  $X_1, \dots, X_p$ . As mentioned above,  $Y$  can capture a wide range of outcomes, such as defaulting on a loan, clicking on an ad, or terminating a subscription.

WOE and IV play two distinct roles when analyzing data:

- WOE *describes the relationship* between a predictive variable and a binary target variable.



The WOE/IV framework is based on the following relationship:

$$\log \frac{P(Y = 1|X_j)}{P(Y = 0|X_j)} = \underbrace{\log \frac{P(Y = 1)}{P(Y = 0)}}_{\text{sample log-odds}} + \underbrace{\log \frac{f(X_j|Y = 1)}{f(X_j|Y = 0)}}_{\text{WOE}},$$

where  $f(x_j|Y)$  denotes the conditional probability density function (or a discrete probability distribution if  $X_j$  is categorical).

This relationship says that the conditional logit of  $P(Y = 1, \text{ given } X_j)$ , can be written as the overall log-odds (i.e., the “intercept”) plus the *log-density ratio* – also known as the *weight of evidence*.

Note that the weight of evidence and the conditional log odds of  $Y = 1$  are perfectly correlated since the “intercept” is constant. Hence, the greater the value of WOE, the higher the chance of observing  $Y = 1$ . In fact, when WOE is positive the chance of observing  $Y = 1$  is above average (for the sample), and vice versa when WOE is negative. When WOE equals 0 the odds are simply equal to the sample average.

## Ties to Naive Bayes and Logistic Regression

Notice that the left-hand-side of the equation above – i.e., the conditional log odds – is exactly what we are trying to predict in a *logistic regression* model. Hence, when building a logistic regression model – which is perhaps the most widely used technique for building binary classifiers – we are actually trying to estimate the weight of evidence.

Last, but not least, the WOE framework has ties to the well-known naive Bayes classifier, given by (see [1]):

$$\log \frac{P(Y = 1|x_1, \dots, x_p)}{P(Y = 0|x_1, \dots, x_p)} = \log \frac{P(Y = 1)}{P(Y = 0)} + \sum_{j=1}^p \log \frac{f(X_j|Y = 1)}{f(X_j|Y = 0)}.$$



In the credit scoring industry a “semi-Naive” version of this model is quite popular. The idea is to transform the data into WOE vectors and then use logistic regression to fit the model

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

thus partly relaxing the assumption that all predictors in the model are independent. It should be noted that the underlying WOE vectors are still estimated univariately and that the coefficients merely function as scalars. For a more general model, GAM is a great choice (see [5]).

## Measuring Univariate Variable Strength with the Information Value

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

$$IV_j = \int \log \frac{f(X_j|Y = 1)}{f(X_j|Y = 0)} (f(X_j|Y = 1) - f(X_j|Y = 0)) dx.$$

Note that the IV is essentially a weighted “sum” of all the individual WOE values where the weights incorporate the *absolute difference* between the numerator and the denominator (WOE captures the relative difference). Generally, if  $IV < 0.05$  the variable has very little predictive power and will not add any meaningful predictive power to your model.

## Estimating WOE

The most common approach to estimating the conditional densities needed to calculate WOE is to bin  $X_j$  and then use a histogram-type estimate.



densities are then obtained by calculating the column percentages from this table.

The typical number of bins used is 10-20. If  $X_j$  is categorical, no binning is needed and the histogram estimator can be used directly. Moreover, missing values are treated as a separate bin and thus handled seamlessly.

If  $B_1, \dots, B_k$  denote the bins for  $X_j$ , the WOE for  $X_j$  for bin  $i$  can be written as

$$\text{WOE}_{ij} = \log \frac{P(X_j \in B_i | Y = 1)}{P(X_j \in B_i | Y = 0)},$$

which means that the IV for variable  $X_j$  can be calculated as

$$\text{IV}_j = \sum_{i=1}^k (P(X_j \in B_i | Y = 1) - P(X_j \in B_i | Y = 0)) \times \text{WOE}_{ij}.$$

## Extensions to Exploratory Analysis for Uplift Models

Consider a direct marketing program where a *test group* received an offer of some sort and the *control group* did not receive anything. The test and control groups are based on a random split. The lift of the campaign is defined as the difference in success rates between the test and control groups. In other words, the program can only be deemed successful if the offer outperforms the “do nothing” (a.k.a baseline) scenario.

The purpose of uplift models is to estimate the difference between the test and control groups and then using the resulting model to target *persuadables* – i.e., potential or existing clients that are on the fence and need some sort of offer to buy a product. Thus, when preparing to build an uplift model, we cannot only focus on the log odds of  $Y = 1$ , we also need to consider the *log odds ratio* of  $Y = 1$  for the test group versus the control group. This leads to the *net weight of evidence* (NWOE) and the *net information value* (NIV). (Actual uplift modeling techniques are beyond the scope of this paper; we will stick to model exploration and variable screening.)



$$\text{NWOE}_j = \log \frac{f(x_j|Y=1)_t f(x_j|Y=0)_c}{f(x_j|Y=1)_c f(x_j|Y=0)_t}.$$

The net information value for variable  $X_j$  is then defined as

$$\int (f(X_j|Y=1)_t f(X_j|Y=0)_c - f(X_j|Y=1)_c f(X_j|Y=0)_t) \times \log \frac{f(X_j|Y=1)_t f(X_j|Y=0)_c}{f(X_j|Y=1)_c f(X_j|Y=0)_t} dx_j.$$

NWOE and NIV work just like IV and WOE, just from an uplift perspective. NIV measures the strength of a given variable while NWOE describes the pattern of the relationship. Specifically, the higher the value of NIV, the better the given variable is at separating *self-selectors* – i.e., people who are self-motivated to buy – and *persuadables* that need to be motivated.

## The Information R Package

The information package is designed to perform exploratory data analysis and variable screening for binary classification models using WOE and IV. To make the package as efficient as possible aggregations are done in data.table and creation of WOE vectors can be distributed across multiple cores.

The package can be downloaded from <https://github.com/klarsen1/Information>.

There are a number of R packages available that support WOE and IV (see a partial list below), but they are either primarily designed to build WOE-based models – e.g., naive Bayes classifiers – or they do not support the uplift use-case.

## Date Used in Examples

The data is from an historical marketing campaign from the insurance industry and is automatically downloaded when you install the Information package. The data is



- **PURCHASE** This variable equals 1 if the client accepted the offer, and 0 otherwise
- **TREATMENT** This variable equals 1 if the client was in the test group (received the offer), and 0 otherwise.

## Key Functions

- `createTables()` creates WOE tables and IVs for all variables in the input dataframe.
- `PlotWOE()` plots the WOE vector for one variable.
- `MultiPlotWOE()` plots more than one WOE vector on one page for comparisons.

The same functions work for NWOE and NIV.

The examples below show how to call these functions

## External Cross Validation

The information package supports external cross validation to check that the WOE and NWOE vectors are stable.

Let's assume that we have a training and a validation dataset, and we calculate WOE for both datasets using the same bin cutoffs. The cross validation penalty for bin  $B_i$  is given by

$$\text{penalty}_i = |\text{WOE}_{\text{train}} - \text{WOE}_{\text{valid}}|,$$

and the total cross validation (CV) penalty is

$$\sum_{i=1}^k |P(X_j \in B_i | Y = 1) - P(X_j \in B_i | Y = 0)| \times \text{penalty}_i.$$





# Example for a Traditional Binary Classification Problem

## Ranking All Variables Using Adjusted IV

```
library(Information)
library(gridExtra)

options(scipen=10)

### Loading the data
data(train, package="Information")
data(valid, package="Information")

### Exclude the control group
train <- subset(train, TREATMENT==1)
valid <- subset(valid, TREATMENT==1)

### Ranking variables using penalized IV
IV <- CreateTables(data=train,
                   valid=valid,
                   y="PURCHASE")

grid.table(head(IV$Summary), rows=NULL)
```

The strongest six variables:

Variable	IV	PENALTY	AdjIV
N_OPEN_REV_ACTS	1.0107695	0.08385690	0.9269126
TOT_HI_CRDT_CRDT_LMT	0.9345902	0.10269736	0.8318929
RATIO_BAL_TO_HI_CRDT	0.8232539	0.06544355	0.7578104
D_NA_M_SNC_MST_RCNT_ACT_OPN	0.6355466	0.07667477	0.5588718
M_SNC_OLDST_RETAIL_ACT_OPN	0.5573438	0.07840106	0.4789427
M_SNC_MST_RCNT_ACT_OPN	0.5026402	0.06044698	0.4421932

## Analyzing WOE Patterns



```
grid.table(IV$Tables$N_OPEN_REV_ACTS, rows=NULL)
```

N_OPEN_REV_ACTS	N	Percent	WOE	IV	PENALTY
[0,0]	1469	0.29545455	-2.0465968	0.6401443	0.05703080
[1,2]	958	0.19267900	-0.5900120	0.6958705	0.06226262
[3,3]	310	0.06234916	0.2033085	0.6986029	0.06514553
[4,5]	583	0.11725664	0.4419768	0.7244762	0.06767437
[6,8]	632	0.12711183	0.6148243	0.7810611	0.07159274
[9,11]	453	0.09111022	0.8815772	0.8692672	0.07683238
[12,48]	567	0.11403862	0.9883818	1.0107695	0.08385690

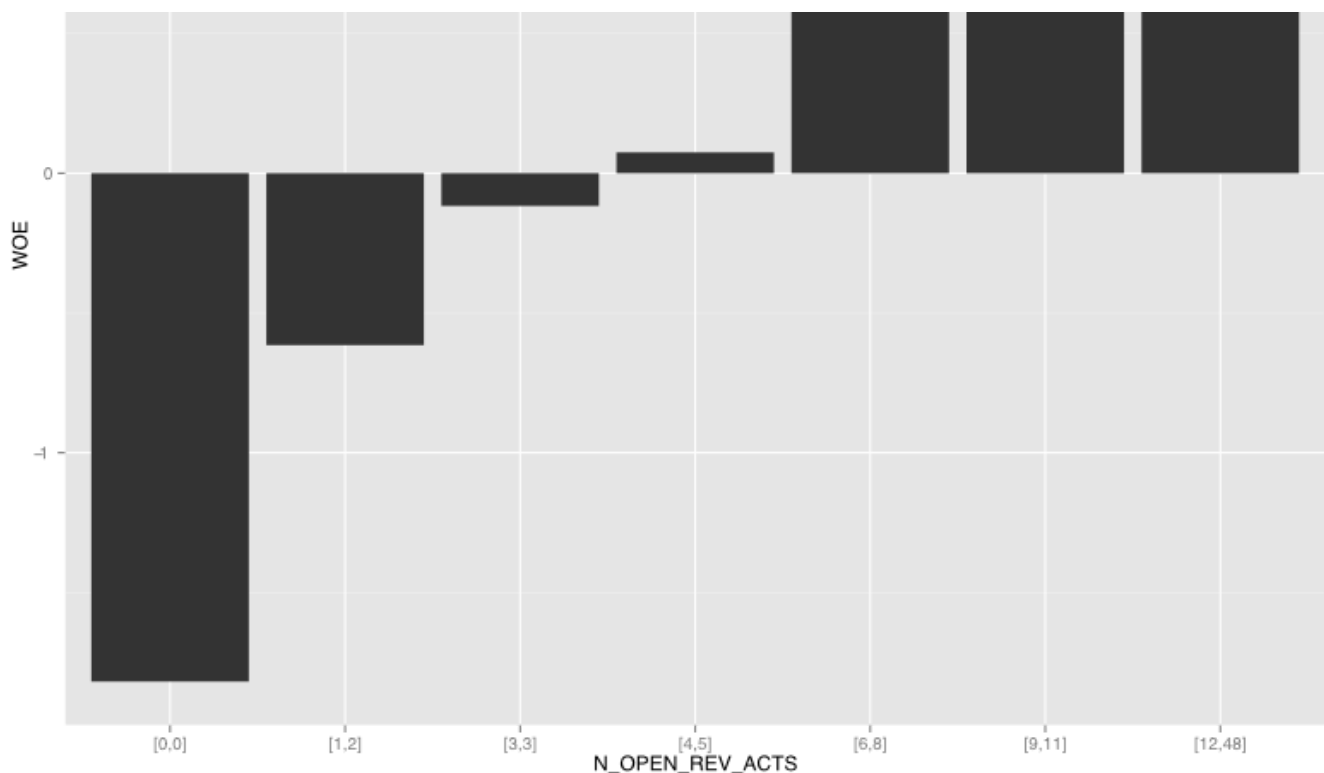
The table shows that the odds of PURCHASE=1 increases as N\_OPEN\_REV\_ACTS increases, although the relationship is not linear.

Note that the Information package attempts to create evenly-sized bins in terms of the number of subjects in each group. However, this is not always possible due to ties in the data, as with N\_OPEN\_REV\_ACTS which has ties at 0. If the variable under consideration is categorical, its distinct categories will show up as rows in the WOE table. Moreover, if the variable has missing values, the WOE table will contain a separate “NA” row which can be used to gauge the impact of missing values. Thus, the framework seamlessly handles missing values and categorical variables without any dummy-coding or imputation .

## Plotting WOE Patterns

We can also plot this pattern for better visualization:

```
PlotWOE(IV, "N_OPEN_REV_ACTS")
```

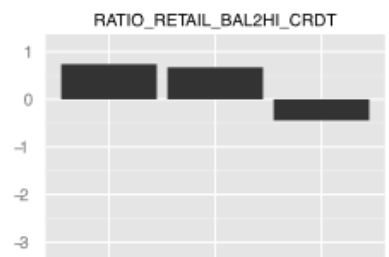
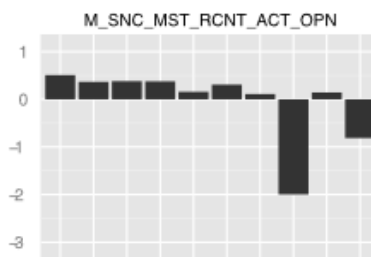
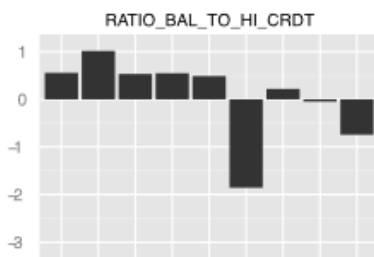
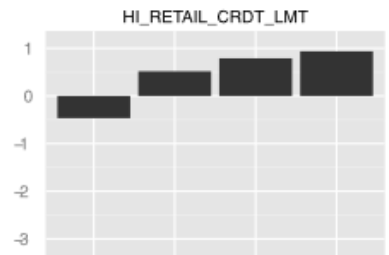
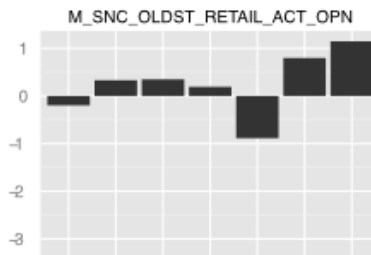
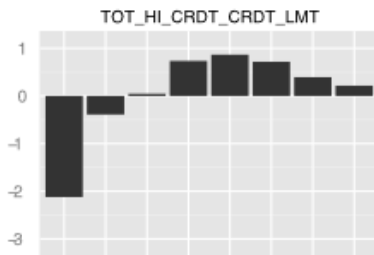
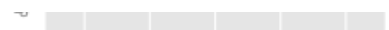
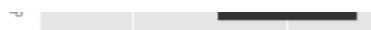


If we want to plot all variables on separate pages, we can loop through all variables:

```
n <- names(IV$Tables)
for (i in 1:length(n)){
  PlotWOE(IV, n[i])
}
```

For better visualization we can do a multiplot to compare WOE patterns. Here we plot the first nine variables on one page. Note that we can plot as many variables as we want; MultiPlotWOE will simply spread the plots over multiple pages (max of nine plots per page).

```
MultiPlotWOE(IV, IV$Summary$Variable[1:9])
```



## Omitting Cross Validation

To run IVs without external cross validation, simply omit the validation dataset:

```
IV <- CreateTables(data=train, y="PURCHASE")
```

## Changing the Number of Bins

The default number of bins is 10 but we can choose a different number if we desire more granularity. Note that the IV formula is fairly invariant to the number of bins. Also, note that the bins are selected such that the bins are evenly sized, to the extent that it is possible (depending on the number of ties in the data).

```
IV <- CreateTables(data=train,  
                   valid=valid,  
                   y="PURCHASE",  
                   bins=20)
```



[0,0]	1469	0.29545455	-2.0465968	0.6401443	0.05703080
[1,1]	619	0.12449718	-0.6965280	0.6886015	0.07175526
[2,2]	339	0.06818182	-0.4149854	0.6989262	0.08042539
[3,3]	310	0.06234916	0.2033085	0.7016586	0.08330830
[4,4]	321	0.06456154	0.3578370	0.7107970	0.08389512
[5,5]	262	0.05269509	0.5410952	0.7286543	0.08628066
[6,6]	241	0.04847144	0.6636975	0.7540782	0.09051947
[7,8]	391	0.07864039	0.5842726	0.7854698	0.09062331
[9,9]	176	0.03539823	1.0188057	0.8323823	0.10326863
[10,11]	277	0.05571199	0.7920957	0.8751621	0.10810564
[12,15]	308	0.06194690	0.7752886	0.9205752	0.10825364
[16,48]	259	0.05209171	1.2316121	1.0247571	0.11745814

## Uplift Example

For an uplift model we have to include both the test group and the control group in our dataset:

```
data(train, package="Information")
data(valid, package="Information")
```

When calling the createTables function, all we have to do is specify the variable that identifies the test and control groups.

```
NIV <- CreateTables(data=train,
                    valid=valid,
                    y="PURCHASE",
                    trt="TREATMENT")

grid.table(head(NIV$Summary),
            rows=NULL,
            gp=gpar(fontsize=12))
```



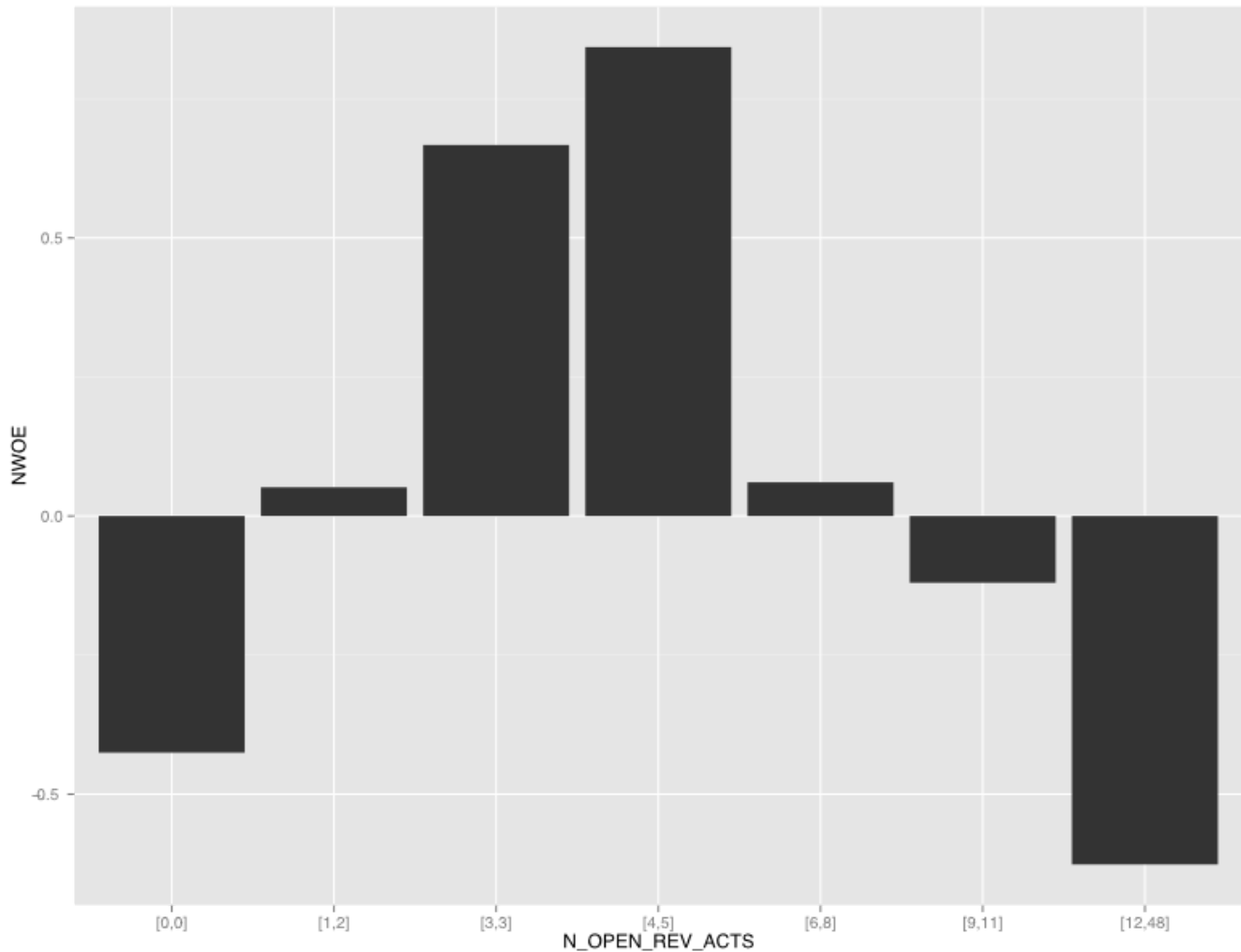
M_SNC_MSTRCNT_MRTG_ACT_UPD	0.09072817	0.003556086	0.08717209
M_SNC_MSTREC_INSTL_TRD_OPN	0.09622970	0.025260288	0.07096942
MRTG_2_CURRENT_BAL	0.07030452	0.005079736	0.06522478
MRTG_1_MONTHLY_PAYMENT	0.13837290	0.084556887	0.05381601

Note that we cannot compare the scales of NIV and IV. Moreover, there is no rule-of-thumb cutoff for the NIV. Hence, we have to use it solely as a ranking statistic and make a judgement call.

Interestingly, `N_OPEN_REV_ACTS` is also the most predictive variable from an uplift perspective. However, the NWOE pattern is quite different from the WOE pattern and suggests a u-shaped pattern. This illustrates how the story can be very different when modeling the incremental effect as opposed to simply building a model to estimate the chance of  $Y = 1$  following a treatment.



[9,11]	0.0885	453	432	-0.11986332	0.8815772	1.0014406	0.15679026	0.05698007
[12,48]	0.1129	567	562	-0.62619770	0.9883818	1.6145795	0.23453611	0.07976614



## Combining IV Analysis With Variable Clustering

Variable clustering divides a set of numeric variables into mutually exclusive clusters. The algorithm attempts to generate clusters such that

- the correlations between variables assigned to the same cluster are maximized.
- the correlations between variables in different clusters are minimized.



correlation with variables outside the cluster. A more meaningful choice for a predictive modeling is to choose the variable that has the highest information value. The following code shows how to do this.

In R, we can do this with the `clustOfVar` package.

```
library(ClustOfVar)
library(reshape2)
library(plyr)

data(train, package="Information")
data(valid, package="Information")
train <- subset(train, TREATMENT==1)
valid <- subset(valid, TREATMENT==1)

tree <- hclustvar(train[,!(names(train) %in% c("PURCHASE", "TREATMENT"))])
nvars <- length(tree[tree$height<0.7])
part_init<-cutreevar(tree,nvars)$cluster
kmeans<-kmeansvar(X.quantile=train[,!(names(train) %in% c("PURCHASE", "TREATMENT"))],
clusters <- cbind.data.frame(melt(kmeans$cluster), row.names(melt(kmeans$cluster)))
names(clusters) <- c("Cluster", "Variable")
clusters <- join(clusters, IV$Summary, by="Variable", type="left")
clusters <- clusters[order(clusters$Cluster),]
clusters$Rank <- ave(-clusters$AdjIV, clusters$Cluster, FUN=rank)
selected_members <- subset(clusters, Rank==1)
selected_members$Rank <- NULL
```

Using variable clustering in combination with IV cuts the number of variables from 68 to 21:

```
nrow(selected_members)
> 21
nrow(clusters)
> 69
```

Here are the first rows from the resulting variable ranking table:

```
grid.table(head(selected_members),
            rows=NULL,
```





3	INFO_BAL_TO_MORTG	0.0202009	0.00044000	0.7370104
4	AGRGT_BAL_ALL_XCLD_MRTG	0.2718228	0.05601568	0.2158071
5	N_OF_SATISFY_FNC_REV_ACTS	0.2883329	0.04703612	0.2412968
6	D_NA_M_SNC_MST_RCNT_ACT_OPN	0.6355466	0.07667477	0.5588718

## Summary

- The purpose of exploratory analysis and variable screening is to get to know the data and assess “univariate” predictive strength, *before* we deploy more sophisticated variable selection approaches.
- The *weight of evidence* (WOE) and *information value* (IV) provide a great framework for performing exploratory analysis and variable screening prior to building a binary classifier (e.g., logistic regression). It seamlessly handles missing values and character variables, and the output is easy to interpret.
- The information value originates from information theory and is closely related to the concept of *mutual information* (see [3], [4]).
- The information package is specifically written to perform this type of analysis using parallel processing. It also supports exploratory analysis for uplift models, a growing area within marketing analytics. The information package is not designed to transfer data into WOE vectors for Naive Bayes models, although this feature could be added later.

## References

[1] Hastie, Trevor , Tibshirani, Robert and Friedman, Jerome. (1986), Elements of Statistical Learning, Second Edition, Springer, 2009.

[Engineering](#)[Algorithms](#)[Algorithms](#)[Careers](#)[Blog](#)[Tour](#)[MultiThreaded](#)[Engineering](#)[Algorithms](#)[Algorithms Tour](#)[Careers](#)[Blog](#)

[4] Shannon, C.E. and Weaver, W. The Mathematical Theory of Communication. Univ of Illinois Press, 1949.

[5] GAM: the Predictive Modeling Silver Bullet,  
<http://multithreaded.stitchfix.com/blog/2015/07/30/gam/>.

## Other Packages that support WOE/IV

woe package, <https://cran.r-project.org/web/packages/woe/woe.pdf>.

riv package (not to be confused with the riv package in CRAN), <http://www.r-bloggers.com/r-credit-scoring-woe-information-value-in-woe-package/>.

klaR package, <https://cran.r-project.org/web/packages/klaR/klaR.pdf>.

[Tweet this post!](#)[Post on LinkedIn](#)

## COME WORK WITH US!

We're a diverse team dedicated to building great products, and we'd love your help.  
Do you want to build amazing products with amazing peers? Join us!

[All Careers at Stitch Fix](#)



Engineering

Algorithms

Algorithms

Careers

Blog

Tour



MultiThreaded

Engineering

Algorithms

Algorithms Tour

Careers

Blog

Your partner in personal style

Stitch Fix and Fix are  
trademarks of Stitch Fix, Inc.

Home

FAQ

Press

Big and tall

Jeans

Business  
Casual

Petite

Plus

Tech Careers

Terms of Use

Privacy Policy



Follow Us!



---

TECH BLOG

---

TECH CAREERS

---

STITCH FIX HOME

---

FAQ

---

PRESS

---

TERMS OF USE

---

PRIVACY POLICY

---

Stitch Fix and Fix are trademarks of Stitch Fix, Inc.

