VARIABLE CREATION
&
VARIABLE REDUCTION

Methodology Training Document (Module 3)

YEAR 2015

EXL Decision Analytics Methodology Snapshot

We apply a set of highly effective tools, techniques and best practices for the end-to-end model development cycle

Stage 1	Preliminary Data Exploration	Univariate Analysis (EDD*) Modeling and Validation Split Bivariate Analysis
Stage 2	Data Preparation	Outlier Treatment Missing Imputation Roll Ups and Data Merge
Stage 3	Variable Creation	Dummy Variable Creation Binning and Banding Transformations Interactions and Groupings
Stage 4	Variable Reduction	Variable Clustering Inter-Correlation Analysis Variance Inflation Factor Test
Stage 5	Modeling	Modeling Technique Selection Model Improvements Ensemble
Stage 6	Validation and Stabilization	In-Sample Validation Out-of-Time Validation Bootstrapping Coefficient Blasting

These stages demand lot of manual effort in analyzing and understanding each and every variable

These stages require business sense and out-of-box thinking for brainstorming on creating hypothesis-based variables and dropping redundant features

These stages require good knowledge of statistical techniques for providing high-end quality solutions

* Extended Data Dictionary

Objectives and Scope

Course Goals

- To provide a structured overview of variable creation and variable reduction techniques used during application of EXL DA methodology
- To introduce trainees to SAS syntax for implementation of such techniques and to explain interpretation of key SAS output
- To make trainees understand how to leverage the relevant MicroAnalytix™ macros
- To provide illustrations for better understanding

Beyond the Scope of this Training

- Comprehensive coaching on all possible techniques - there is great scope for innovation
- Derivation of statistical formulas or terms (unless required as part of methodology explanation)
- Variable reduction techniques that involve variable transformation (For example: Principal Component and Factor Analysis) – Such techniques are covered separately in Advanced Methodology Training

Self Study Goals

- Practice variable creation and variable reduction techniques covered during the training course
- Research on advanced parameters for implementation in SAS
- Discussion on advanced concepts can be taken up offline

Table of Contents

References

Chapter 1: Variable Creation

1.1 Need for Variable Creation

Two Key Reasons for Importance of Derived Variables

Better Model Performance

- Derived variables add to the predictive power of raw variables
- Model performance gets boosted

Insight Generation

- Derived variables provide the missing links in explaining the observed patterns
- Key driver analysis becomes more meaningful

1.2 Variable Creation Techniques

1.2.1. Dummy Coding

Meaning

- Creation of a binary indicator (flags) for each individual category

Usage

- Dummy coding converts a character variable structure into multiple numeric variables
- Numeric variables are easier to use for data analysis and modeling

Example

data.sas7bdat

	PHYSICIAN_ID	SPECIALTY	IND_SPC_HEART	IND_SPC_EYE	IND_SPC_NEURO
1	XX87601	HEART	1	0	0
2	XX87602	EYE	0	1	0
3	XX87603	NEURO	0	0	1
4	XX87604	NEURO	0	0	1
5	XX87605	EYE	0	1	0

Note: For model development purpose, in case of N categories, use only N-1 dummies

1.2.2. Bivariate Profiling

Meaning

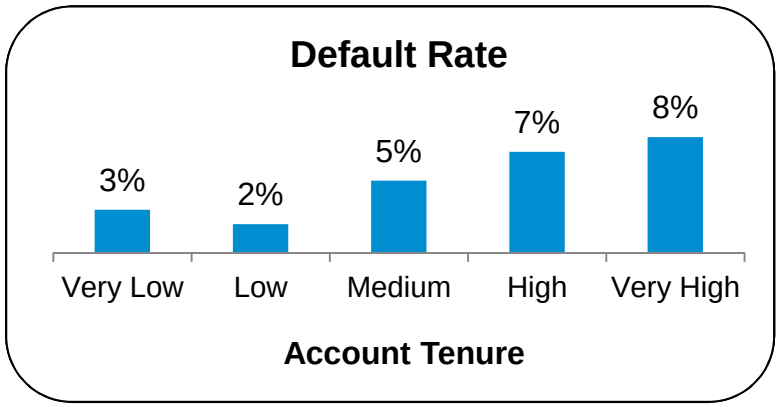
- Simultaneous analysis of two variables (a dependent and an independent variable)

Usage

- Useful indicator variables can be created based on trend in mean dependent variable value across the bins or categories of an independent variable

Example 1

- Target : Default Flag (IND_DEFAULT); Suppose overall default rate is 5%
- Independent Variable : Account tenure, taking 5 values (Very Low, Low, Medium, High, Very High)

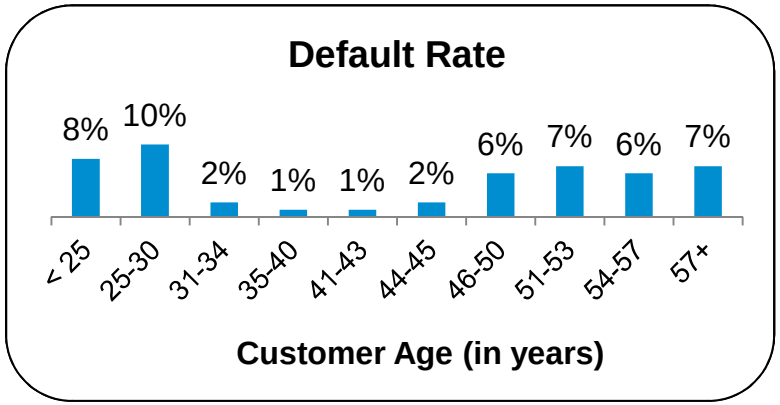


- Create a flag that takes value 1 if account tenure is 'Very Low' or 'Low' (IND_TENURE_VL_L). This will have fairly good negative correlation with IND_DEFAULT
- Create a flag that takes value 1 if account tenure is 'High' or 'Very High' (IND_TENURE_H_VH). This will have fairly good positive correlation with IND_DEFAULT
- Sizing of buckets (5 categories) also matters. In best case scenario, they should be evenly distributed

Example 2

- Target : Default Flag (IND_DEFAULT); Suppose overall default rate is 5%
- Independent Variable : Customer Age (continuous variable)

Create 'Customer Age' deciles (that is, 10% population per bin) and plot 'Default Rate'



Young or Otherwise		
Age	**Sizing**	**Default Rate**
Age ≤ 30	20%	9.0%
Age > 30	80%	4.0%
Overall	**100%**	**5.0%**

Middle Aged or Otherwise		
Age	**Sizing**	**Default Rate**
30 < Age ≤ 45	40%	1.5%
Age ≤ 30 or Age > 45	60%	7.3%
Overall	**100%**	**5.0%**

Old or Otherwise		
Age	**Sizing**	**Default Rate**
Age ≤ 45	60%	4.0%
Age > 45	40%	6.5%
Overall	**100%**	**5.0%**

- Create flags for Young (<=30 years), Middle Aged (31-45 years) and Old (>45 years)
- 'Young' and 'Old' indicators would have moderate positive correlation with IND_DEFAULT
- 'Middle Aged' flag would have high negative correlation and is going to be a strong predictor

1.2.3. Interaction

An interaction variable is a multiplication of individual predictors

Interactions may add incremental value

Illustration: A Three-Way Interaction Variable

Target Variable: Whether an account is 30+ DPD at the end of 12 months of book

Predictor 1		Predictor 2		Predictor 3		
Indicator that FICO score < 700	AND	Number of times delinquent in past 3 months	AND	Utilization rate in past 6 months < 30%	→	**Interaction**
8.5%		**9.7%**		**8.3%**		**12.6%**

Absolute Correlation with Target Variable

Incremental Value = Absolute Correlation of Interaction **-** Max **(**Absolute Correlation of All Component Variables**)**
= 12.6% - 9.7%
= **2.9 percentage points**

Super Interactions Macro Syntax

```
%SUPER_INTERACTIONS
        (
        VERSION          = <Version number>,
        MOD_DATA         = <Library and name of modeling dataset>,
        VAL_DATA         = <Library and name of validation dataset> [1],
        SCR_DATA         = <Library and name of scoring dataset> [2],
        OUTLIB           = <Library of output dataset>,
        CSV_PATH         = <Location of CSV file>,
        DEP_VAR          = <Name of dependent variable>,
        IF_2WAY          = <Y or N [3]>,
        IF_3WAY          = <Y or N [4]>,
        IF_4WAY          = <Y or N [5]>,
        IF_5WAY          = <Y or N [6]>,
        PREDICTORS       = <List of predictors (separated by space)>
        );
```

[1] Optional. If there is no validation dataset, user should leave it blank.
[2] Optional. If there is no scoring dataset, user should leave it blank.
[3] **IF_2WAY = Y** if 2-way interactions are required
[4] **IF_3WAY = Y** if 3-way interactions are required
[5] **IF_4WAY = Y** if 4-way interactions are required
[6] **IF_5WAY = Y** if 5-way interactions are required

Super Interactions Macro Algorithm

1. If user sets 'IF_2WAY' = Y, create all possible two-way (a x b) interactions on the modeling dataset for the given list of independent variables
2. If user sets 'IF_3WAY' = Y, create all possible three-way (a x b x c) interactions on the modeling dataset for the given list of independent variables
3. If user sets 'IF_4WAY' = Y, create all possible four-way (a x b x c x d) interactions on the modeling dataset for the given list of independent variables
4. If user sets 'IF_5WAY' = Y, create all possible five-way (a x b x c x d x e) interactions on the modeling dataset for the given list of independent variables
5. Compute means of all interaction variables for the modeling dataset
6. Compute correlation of all interaction variables with the target variable for the modeling dataset
7. If user specifies a validation dataset (containing target and predictors), do steps 1-6 for validation data
8. If user specifies a scoring dataset (containing predictors), do steps 1-5 for the scoring dataset
9. At variable level, create an indicator (I_STRAIGHTAWAY_DROP) if
   - A variable takes single unique value or
   - The sign of correlation with target doesn't match across modeling and validation or
   - The sign of mean value doesn't match across modeling, validation and scoring
10. At variable level, as a measure of stability, compute absolute percentage change in
    - Absolute correlations across Modeling and Validation (Metric 1)
    - Absolute means across Modeling and Validation (Metric 2)
    - Absolute means across Validation and Scoring (Metric 3)
    - Absolute means across Modeling and Scoring (Metric 4)

1.2.4. Groupings

A grouped variable is a union of conditions (binary predictors)

Unlike binary variable interactions (intersection of conditions), the grouped variables tend to have better sizing and they often add good value

Illustration: A Three-Way Grouped Variable

Target Variable: Whether an account is 30+ DPD at the end of 12 months of book

Predictor 1		Predictor 2		Predictor 3		
Indicator that customer resides in South region	OR	Indicator that customer resides in West region	OR	Indicator that customer is a renter	→	**Grouping**
2.4%		**1.3%**		**3.6%**		**5.9%**

Absolute Correlation with Target Variable

Incremental Value = Absolute Correlation of Grouping **-** Max **(**Absolute Correlation of All Component Variables**)**
= 5.9% - 3.6%
= **2.3 percentage points**

Super Groups Macro Syntax

```
%SUPER_GROUPS
        (
        VERSION          = <Version number>,
        MOD_DATA         = <Library and name of modeling dataset>,
        VAL_DATA         = <Library and name of validation dataset> [1],
        SCR_DATA         = <Library and name of scoring dataset> [2],
        OUTLIB           = <Library of output dataset>,
        CSV_PATH         = <Location of CSV file>,
        DEP_VAR          = <Name of dependent variable>,
        IF_2WAY          = <Y or N [3]>,
        IF_3WAY          = <Y or N [4]>,
        IF_4WAY          = <Y or N [5]>,
        IF_5WAY          = <Y or N [6]>,
        PREDICTORS       = <List of predictors (separated by space)>
        );
```

[1] Optional. If there is no validation dataset, user should leave it blank.
[2] Optional. If there is no scoring dataset, user should leave it blank.
[3] **IF_2WAY = Y** if 2-way groupings are required
[4] **IF_3WAY = Y** if 3-way groupings are required
[5] **IF_4WAY = Y** if 4-way groupings are required
[6] **IF_5WAY = Y** if 5-way groupings are required

Super Groups Macro Algorithm

1. If user sets 'IF_2WAY' = Y, create all possible two-way (a=1 or b=1) groups on the modeling dataset for the given list of independent variables
2. If user sets 'IF_3WAY' = Y, create all possible three-way (a=1 or b=1 or c=1) groups on the modeling dataset for the given list of independent variables
3. If user sets 'IF_4WAY' = Y, create all possible four-way (a=1 or b=1 or c=1 or d=1) groups on the modeling dataset for the given list of independent variables
4. If user sets 'IF_5WAY' = Y, create all possible five-way (a=1 or b=1 or c=1 or d=1 or e=1) groups on the modeling dataset for the given list of independent variables
5. Compute means of all group variables for the modeling dataset
6. Compute correlation of all group variables with the target variable for the modeling dataset
7. If user specifies a validation dataset (containing target and predictors), do steps 1-6 for validation data
8. If user specifies a scoring dataset (containing predictors), do steps 1-5 for the scoring dataset
9. At variable level, create an indicator (I_STRAIGHTAWAY_DROP) if
   - A variable takes single unique value or
   - The sign of correlation with target doesn't match across modeling and validation or
   - The sign of mean value doesn't match across modeling, validation and scoring
10. At variable level, as a measure of stability, compute absolute percentage change in
    - Absolute correlations across Modeling and Validation (Metric 1)
    - Absolute means across Modeling and Validation (Metric 2)
    - Absolute means across Validation and Scoring (Metric 3)
    - Absolute means across Modeling and Scoring (Metric 4)

1.2.5. Mathematical Transformations

Usage

- Correlation between a dependent and an independent variable is a measure of the degree of linear association only
- Mathematical transformations of the independent variable capture the non-linear trend
- Six mathematical transforms, generally, used at EXL Decision Analytics comprise
  - Square
  - Square Root
  - Cube
  - Cube Root
  - Log
  - Inverse

Macro in MicroAnalytix™

- Variable transformation macro computes six types of transformations for a given set of variables and retains the transformation for each variable that has maximum correlation with the dependent variable

Variable Transformation Macro Syntax

```
%TRANSFORM_VARS
  (
  INLIB                 =  <Library of input dataset>,
  INDATA                =  <Name of input dataset>,
  OUTLIB                =  <Library of output dataset>,
  OUTDATA               =  <Name of output dataset>,
  VAR_LIST              =  <List of variables>,
  DEP_VAR               =  <Name of dependent Variable>,
  RETAIN_ALL_TRANSFORMS =  <Y or N, depending upon all transformations need to be retained or not>,
  RETAIN_BASE_VAR       =  <Y or N, depending upon base variables need to be retained or not>
  );
```

Note: **TRANSFORM_VARS** macro will not work for variables with name length greater than **27** bytes, as it adds suffix to raw variable names. It is advised to rename the variables accordingly before using this macro.

1.2.6. Variable Tenurization

Meaning and Usage

- Variable tenurization involves usage of time-series data to create a separate variable for each time period and to create summarized variables (e.g. min, max, range, mean, standard deviation)
- Without any loss of information, variable tenurization uses granular level information (e.g. transactional level information) and facilitates model development at a higher level (e.g. customer level)

Example

- Consider daily transaction amount information (for a week) for 2 customers C1 and C2

CUST_ID	DAY	AMT
C1	MON	40
C1	TUE	45
C1	WED	55
C1	THU	50
C1	FRI	42
C1	SAT	51
C1	SUN	46
C2	MON	47
C2	TUE	46
C2	WED	48
C2	THU	45
C2	FRI	49
C2	SAT	44
C2	SUN	50

CUST_ID	AMT_1	AMT_2	AMT_3	AMT_4	AMT_5	AMT_6	AMT_7
C1	40	45	55	50	42	51	46
C2	47	46	48	45	49	44	50

CUST_ID	AMT_MIN	AMT_MAX	AMT_RANGE	AMT_AVG	AMT_STDEV
C1	40	55	15	47	5.29
C2	44	50	6	47	2.16

1.2.7. Trend Variables

Delta Variables

- Difference variables to capture
  - Positive, negative or no change trend; and
  - Magnitude of change in case of positive or negative trend
- Example

  DELTA_AVG_AMT_3M_6M = AVG_AMT_3M – AVG_AMT_6M

Ratio Variables

- Ratio variables to capture percentage change
- Example

  RATIO_AVG_AMT_3M_6M = AVG_AMT_3M / AVG_AMT_6M

1.2.8. Weight of Evidence Approach

Usage

- Weight of Evidence (WOE) analyzes the predictive power of a variable in relation to the event
- For each continuous variable, create decile bins and for each categorical variable, use actual categories
- For every bin or category of a predictor, calculate number of events, number of non-events and hence the weight of evidence using following formula

$$WOE = \left[LN\left(\frac{E_i / E}{NE_i / NE} \right) \right] \times 100\%$$

where

E_i = Count of events in bin or category i

NE_i = Count of non events in bin or category i

E = Total count of events in the sample

NE = Total count of non events in the sample

- Instead of raw variable, use its WOE values as the derived variable

A Related Concept: Information Value

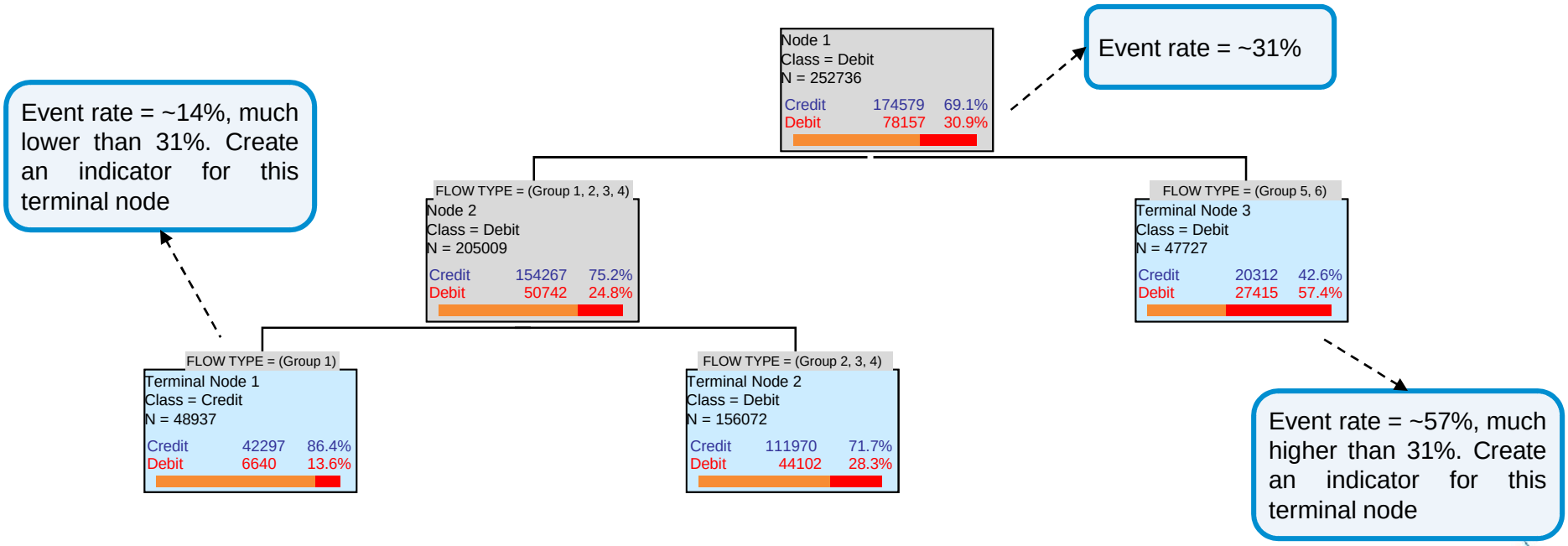
- Used for variable selection

$$IV = \sum_{i=i}^{n} \left[\left(\frac{E_i}{E} - \frac{NE_i}{NE} \right) \times LN\left(\frac{E_i / E}{NE_i / NE} \right) \right]$$

1.2.9. CART Nodes

CART is a non-linear, non-parametric technique that finds a cut-point in the predictor variable to classify the records into different levels of the response variable

Illustrative Example



1.2.10. MARS Basis Functions

MARS (Multivariate Adaptive Regression Splines) is a multivariate non-parametric regression procedure which builds flexible regression models by fitting separate Splines (or basis functions) to distinct intervals of the predictors

Example of a Basis Function:

BF1 = max(0, AMT – 5)

Where

AMT is an independent variable

5 is a constant value (also called knot), auto-generated by MARS algorithm

Drawbacks

- Not intuitive; they generally don't have any business meaning
- Tend to be unstable as they tend to overfit to the train data

Advantages

- Automated process based on optimization; fairly easy way to boost model performance (given that there is no major stability issue)

1.2.11. Hypothesis Based

Hypothesis based variables are very useful from the perspective of explanation and understanding

Hypotheses are generally driven by

- Internal factors
  - Intuition / common sense
  - Business sense
  - Domain knowledge
  - Experience
- External factors
  - Existing model equation
  - Study of research papers
  - Discussion with experts

Chapter 2: Variable Reduction

2.1 Need for Variable Reduction

A Typical Scenario

- A target variable Y
- N relevant predictors X_1, X_2 …X_N

Two Common Cases

Omission of Relevant Effects

$$Y = f(X_1, X_2, X_4 \ldots X_{N-1}, X_N)$$

X_3, despite being relevant, is missing from the model equation

Inclusion of Irrelevant Effects

$$Y = f(X_1, X_2, Z, X_3, X_4 \ldots X_{N-1}, X_N)$$

Z, despite being irrelevant, forms a part of the model equation

Need for Appropriate Application of Variable Selection / Variable Reduction Methodology

2.2 Variable Reduction Techniques

2.2.1. Constant Value Variables

Case of Straightaway Drop

- A variable taking single unique value in the modeling dataset adds no value

Example

- Data for 2012 batch of10 students of XI standard

Variables 'STANDARD' and 'YEAR' take same value across all observations and hence add no value. **They should be dropped straightaway**

train_sample.sas7bdat

	ROLL_NO	STANDARD	YEAR	MARKS	ATTENDANCE	HEIGHT	WEIGHT
1	100501	XI	2012	80	0.70	173	55
2	100502	XI	2012	96	0.82	165	59
3	100503	XI	2012	78	0.69	162	64
4	100504	XI	2012	88	0.78	175	68
5	100505	XI	2012	82	0.93	170	70
6	100506	XI	2012	36	0.45	168	66
7	100507	XI	2012	41	0.58	166	56
8	100508	XI	2012	29	0.66	171	69
9	100509	XI	2012	97	0.75	173	73
10	100510	XI	2012	65	0.72	170	70

2.2.2. Correlation with Target

Features with 'close to zero' correlation with the target variable can be safely dropped

Illustration

Target variable: EXPENSE

Predictors:

- Individual's household income (HH_INCOME)
- Residence at outskirts (OUTSKIRTS)
- Number of kids in the household (NUM_KIDS)
- Individual's favorite color is red (FAV_COLOR_RED)

SAS Code

```
PROC CORR
 DATA = INPUT_DATA
 OUTP = OUTPUT_DATA;
 VAR EXPENSE;
 WITH HH_INCOME
      OUTSKIRTS
      NUM_KIDS
      FAV_COLOR_RED;
RUN;
```

Output Interpretation

	A	B	C
1	**Variable**	**Correlation**	**Absolute Correlation**
2	HH_INCOME	38.5%	38.5%
3	OUTSKIRTS	-15.7%	15.7%
4	NUM_KIDS	11.2%	11.2%
5	FAV_COLOR_RED	0.1%	0.1%

- As expected, an individual's spending is much less likely to be related to his favorite color than his household income
- Variable 'FAV_COLOR_RED' does not pass the 0.5% cut off and it is safe to exclude it from the model development process

2.2.3. Variable Clustering

VARCLUS procedure classifies numeric variables into disjoint or hierarchical clusters based on a similarity measure

Illustration and SAS Code

Six Predictors:

A, B, C, D, E and F

```
PROC VARCLUS
 DATA = INPUT_DATA
 MAXEIGEN = 0.7
 MAXCLUSTERS = 6
 SHORT
 HI;
 VAR A B C D E F;
 ODS OUTPUT
 RSQUARE = OUTPUT_DATA;
RUN;
```

Variable Selection Criterion

A variable selected from each cluster should have a high correlation with its own cluster and a low correlation with the other clusters

$$\text{R Square Ratio} = \frac{1 - \text{R Square Own Cluster}}{1 - \text{R Square Next Closest}}$$

A cluster's best representative is the variable with minimum R Square Ratio

Output Interpretation

Cluster	Variable	Own Cluster	Next Closest	R-Square Ratio
1	A	0.9124	0.1236	0.1000
	B	**0.9156**	**0.1167**	**0.0956**
	C	0.5461	0.1791	0.5529
2	D	0.8242	0.0694	0.1889
	E	**0.8338**	**0.0862**	**0.1819**
	F	0.6362	0.0347	0.3768

- Variables B and E are the best representatives of Clusters 1 and 2 respectively and hence they are selected for further consideration in the model

Variable Clustering Macro Syntax

```
%VARCLUSMACRO          (
                       VERSION              = <Version number>,
                       IN_DATA              = <Library and name of input dataset>,
                       OUTLIB               = <Library of output dataset>,
                       DEP_VAR              = <Name of dependent variable>,
                       INDEPVARLIST_1       = <Independent variables (separated by space)>,
                       INDEPVARLIST_2       = <Independent variables (separated by space)>,
                       INDEPVARLIST_3       = <Independent variables (separated by space)>,
                       INDEPVARLIST_4       = <Independent variables (separated by space)>,
                       INDEPVARLIST_5       = <Independent variables (separated by space)>,
                       NUM_MAXCLUSTERS      = <Maximum number of clusters desired>,
                       MAXEIGEN_VALUE       = <Maximum Eigen value>,
                       MODULEWISE           = <Y or N [1] >
                       );
[1] MODULEWISE = Y if of all modules in a cluster, at least 1 variable from each module is to be kept.
Variables of same module should have same prefix (e.g. names of all demographics variables may start
with 'demo_', those of geography variables may start with 'geo_' , and so forth)]
```

This macro is used to automate variable selection from variable clustering output.

- Version number is to avoid overwriting of datasets. E.g. if 'version' is specified as 01, output SAS dataset would be 'varclus_output_v01'
- If independent variables aren't too many, they can be specified in 'indepvarlist_1' and remaining lists can be left blank. However, if there are thousands of independent variables, they can be split into 5 different lists
- Default value of 'NUM_MAXCLUSTERS' is 10000. User may leave it blank, if there is no need to keep maximum limit on the # clusters
- Default value of 'MAXEIGEN_VALUE' is 0.7. User may leave it blank.
- Output dataset and excel file would contain all independent variables along with four keep-list indicators
    a. **VARSELECT_BY_CORR** (based on maximum absolute correlation with the dependent variable within a cluster)
    b. **VARSELECT_BY_RSQRATIO** (based on minimum 1-RSquare Ratio within a cluster)
    c. **VARSELECT_BY_CORR_OR_RSQRATIO** (Conservative Approach: Union of a and b above)
    d. **VARSELECT_BY_CORR_AND_RSQRATIO** (Aggressive Approach: Intersection of a and b above)
- It is recommended to go with conservative approach in general

Variable Clustering macro of MicroAnalytix™ Toolkit clusters variables and selects best representatives from each cluster

ILLUSTRATIVE

Illustrative Variable Clustering Output

Sequence	# Clusters	Cluster	Variable	1-R-Square Ratio	Absolute Correlation	Select by R-Square Ratio	Select by Correlation
1	6	Cluster 1	AVG_NUM_PRM_CALLS_3M	0.0430	13.6%	1	1
2	6	Cluster 1	AVG_NUM_PRM_CALLS_6M	0.0543	12.6%	0	0
3	6	Cluster 2	IND_CHANGE_IN_ADDRESS	0.0001	5.8%	1	1
4	6	Cluster 2	IND_CHANGE_IN_CITY	0.0088	4.7%	0	0
5	6	Cluster 3	REBATE_AMT_6M	0.0012	1.8%	1	0
6	6	Cluster 3	REBATE_AMT_3M	0.0015	2.3%	0	1
7	6	Cluster 4	IND_DO_NOT_CALL	0.0085	7.5%	1	1
8	6	Cluster 4	IND_DO_NOT_MAIL	0.0199	7.0%	0	0
9	6	Cluster 5	NUM_COMPLAINTS	0.0436	19.7%	1	1
10	6	Cluster 5	IND_POOR_FEEDBACK	0.0746	11.9%	0	0
11	6	Cluster 6	NUM_ACTIVE_SERVICES	0.0146	4.8%	1	0
12	6	Cluster 6	DAYS_TO_CONTRACT_EXPIRY	0.0443	7.3%	0	1
13	6	Cluster 6	IND_HOMEOWNER	0.0301	2.1%	0	0

This may be used for

- Grouping variables into different clusters
- Selection of Cluster's Best Representative based on minimum 1-R square Ratio within cluster
- Selection of Strong Predictor based on maximum absolute correlation with the target variable

2.2.4. Inter-Correlation Analysis

Inter-correlation among predictors is analyzed to eliminate highly correlated variables

Algorithm

1. For a given list of independent variables, compute pair-wise absolute correlation (Pearson's absolute correlation of each variable with all other variables)
2. Identify the pair with maximum absolute inter-correlation
3. If this maximum value of absolute inter-correlation is greater than the specified correlation cut-off, check absolute correlation of the two variables with the target variable
4. Drop the variable with lesser absolute correlation with the target
5. Repeat Steps 2-4, as long as the maximum value of absolute inter-correlation between any two variables is greater than the specified correlation cut-off

Inter-Correlation Analysis Macro Syntax

```
%INTERCORR_ANALYSIS
          (
          VERSION              = <Version number>,
          IN_DATA              = <Library and name of modeling dataset>,
          OUTLIB               = <Library of output dataset>,
          DEP_VAR              = <Name of dependent variable>,
          CORR_CUTOFF          = <Correlation Cut-Off (should be a fraction) [1] >,
          PREDICTORS           = <List of predictors (separated by space)>
          );
[1] Default value is 0.7
```

Features

Very fast in processing (takes only few minutes to run even with 1000 variables)

Choice of any inter-correlation cut-off

Generates two sets of variables as output

- List of Shortlisted Variables
- List of Eliminated Variables

2.2.5. Variance Inflation Factor Test

Multicollinearity issue arises when predictors are highly correlated with each other. VIF test may be used to identify and resolve the problem

Illustration and SAS Code

Target variable: EXPENSE

Predictors:

- Individual's household income (HH_INCOME)
- Flag for medical profession (MEDICAL_PROFESSION)
- Number of kids in the household (NUM_KIDS)
- Individual's age (AGE)

```
PROC REG
 DATA = INPUT_DATA;
 MODEL EXPENSE = HH_INCOME
                 MEDICAL_PROFESSION
                 NUM_KIDS
                 AGE
                 / VIF;
 ODS OUTPUT
 PARAMETERESTIMATES = OUTPUT_DATA;
QUIT;
```

Variance Inflation Factor (VIF)

$$VIF = \frac{1}{1 - R_i^2}$$

For each predictor i, R-square is defined as the coefficient of determination in a regression model where predictor i is considered as target variable and all other predictors are explanatory variables. Higher R-square results in higher VIF and indicates high correlation between the target (i.e. predictor i) and all other predictors.

Output Interpretation

Variable	VIF
HH_INCOME	9.2342
MEDICAL_PROFESSION	8.5621
NUM_KIDS	1.1596
AGE	1.1258

- Household Income and Medical Profession Indicator have high VIF values. One of these two variables should be dropped out of the model equation

Variance Inflation Factor (VIF) Macro Syntax

```
%VIF
 (
 MOD_DAT              = <Library and name of input dataset>,
 OUT_DAT              = <Library and name of output dataset containing stats of short-listed variables>,
 ELIM_SUM             = <Library and name of output dataset containing summary of eliminated variables>,
 VAR_LIST             = <List of variables>,
 WEIGHT_VAR           = <Weight variable>,
 DP_VAR               = <Name of dependent variable>,
 MAX_VIF_LIMIT        = <Maximum value of VIF permitted>,
 IF_CORR              = <Y or N [1] >
 );

 [1] IF_CORR = N, if VIF only is sufficient
     IF_CORR = Y, if correlation technique should be applied for variable reduction
```

This macro is used to reduce the correlated variables in a model using Variance Inflation Factor

The macro works in two ways:

CASE I (**IF_CORR** = **N**) ---> Macro will

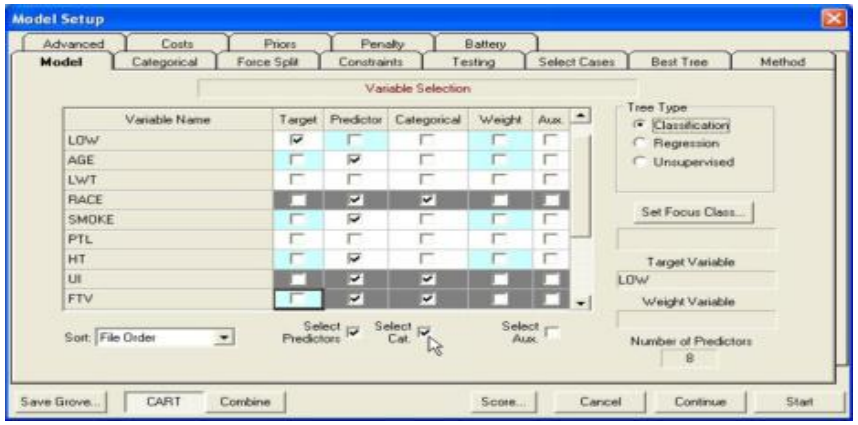
i. delete the variable having maximum VIF value
ii. run iteratively till all the variables have VIF value < MAX_VIF_LIMIT

CASE II (**IF_CORR** = **Y**) ---> Macro will

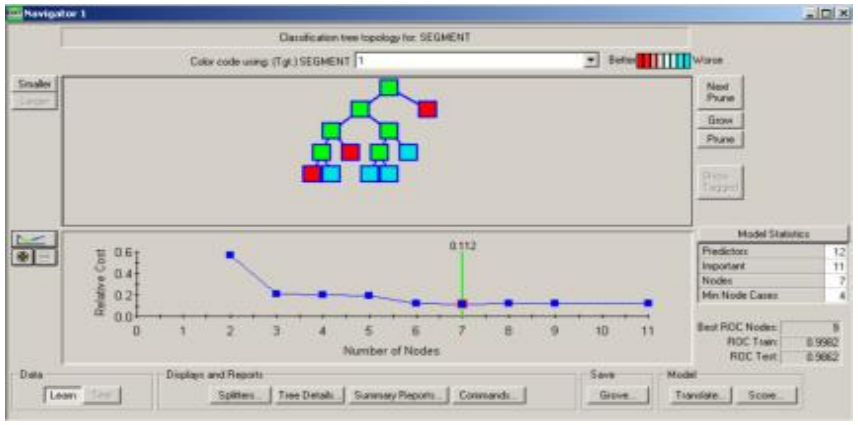
i. identify the variable with highest VIF in each iteration,
ii. using the COLLIN table, find the corresponding variable in the given variable list responsible for the high VIF of the first variable.
iii. check the correlation of both the variables with the dependent variable
iv. retain the variable having higher correlation with the dependent variable, and drop the other one from the variable list
v. run iteratively till all the variables have VIF value < MAX_VIF_LIMIT

2.2.6. Variable Importance List from Decision Trees

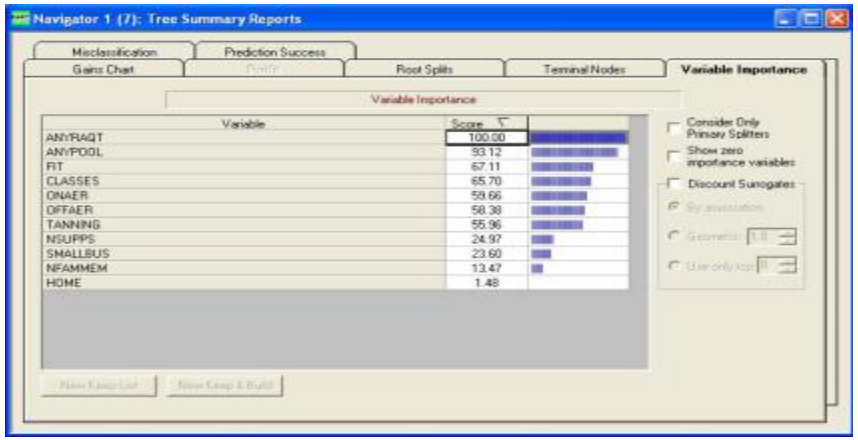
STEP 1 Set Up a CART Model



STEP 2 Grow / Prune CART Tree (Navigator)



STEP 3 Save Variable Importance Scores



STEP 4 Shortlist Variables

- ✓ Keep aside the variables with high or positive scores
- ✓ Re-run CART on remaining variables
- ✓ Repeat the process to shortlist more variables

2.2.7. MAX STEP = 1 Technique

Usage

- Univariate predictive power of each continuous variable is analyzed

How to Implement?

- Run PROC LOGISTIC with SELECTION = STEPWISE MAXSTEP = 1 and DETAILS

SAS Code

```
PROC LOGISTIC
 DATA = INPUT_DATA DESCENDING;
 MODEL Y = X1 X2 X3 X4 X5 X6
           / SELECTION = STEPWISE
             MAXSTEP = 1
             DETAILS;
RUN;
```

Rule of Thumb

- Eliminate all variables with probability of chi-square > 0.5

Relevance

- Relevant for continuous variables only
- Particularly useful in case of 1000+ modeling variables

2.2.8. Dry Run

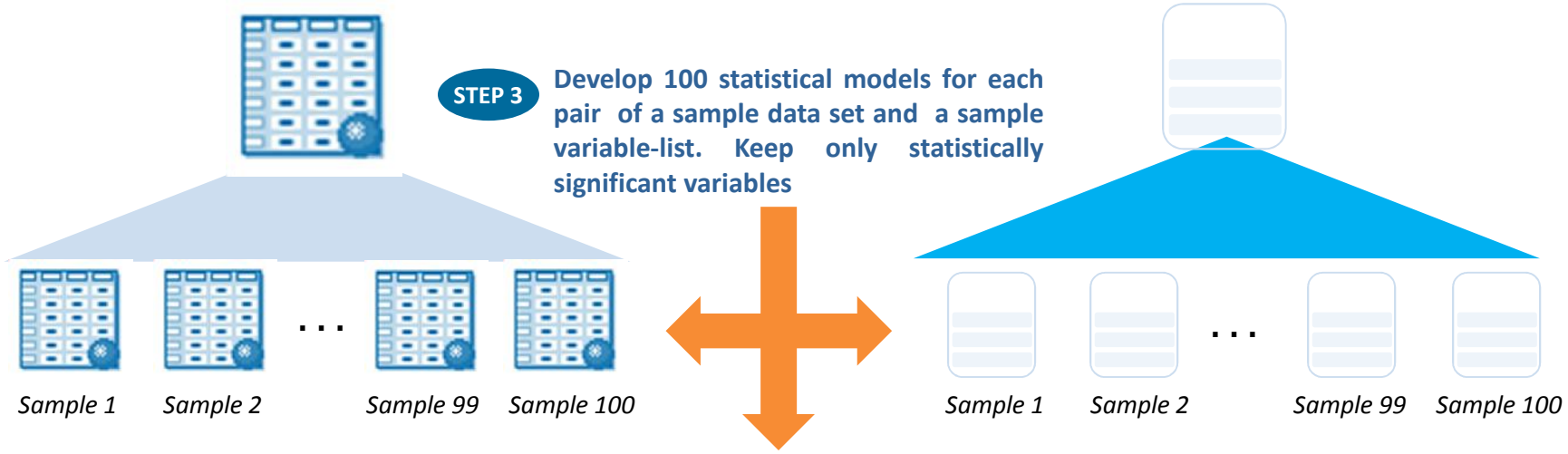
Usage

- A few dry logistic regressions (after variable clustering) to get a list of variables turning up in first cut models
- This is one stage ahead of MAX STEP = 1 as here a smaller list of variables is used and also no restriction is put on the number of steps in modeling process
- It has manual selection biases and so is one step behind the automated random forest method

2.2.9. Random Feature Selection

STEP 1 Draw 100 random samples of a fixed proportion (70%) of the modeling data set where records are drawn with replacement

STEP 2 Draw 100 random samples of a fixed proportion (80%) from the specified list of n variables, with replacement



STEP 3 Develop 100 statistical models for each pair of a sample data set and a sample variable-list. Keep only statistically significant variables

Things to Remember

This is a variant of bootstrapping approach, where idea of random sampling of features is borrowed from Random Forest algorithm

STEP 4 For each variable, compute the percentage occurrence in 100 models

STEP 5 Select variables with 'high' percentage occurrence (say, greater than 80%)

Variable (A)	#Times Variable was Randomly Picked (B)	#Models where Variable was present (C)	% Occurrence (C) / (B)
V1	65	63	97%
V2	75	71	95%
V3	66	61	92%
V4	76	70	92%
V5	66	60	91%
V6	64	58	91%
V7	78	69	88%
V9	69	58	84%
V8	73	61	84%
V10	76	62	82%
V11	62	44	71%
V12	67	47	70%
V13	76	50	66%
V14	73	48	66%
V15	64	40	63%
V16	70	39	56%
V17	77	38	49%
V18	72	35	49%
V19	70	30	43%
V20	62	26	42%
V21	68	27	40%
V22	67	24	36%
V23	77	20	26%
V24	67	17	25%
V25	71	18	25%
V26	75	13	17%
V27	74	11	15%
V28	72	10	14%
V29	65	5	8%
V30	75	0	0%

ILLUSTRATIVE

- 30 Input Variables
- Picked up 100 random samples
- Example:
  - V1 was picked up for building 65 models
  - Out of 65 models, V1 turned up in 63 models; hence its percentage occurrence is 97%
- 10 Variables out of 30 get selected if we apply the criterion of percentage occurrence > 80%

References

1. **Feature Selection and Dimension Reduction Techniques in SAS**
   by Varun Aggarwal and Sassoon Kosian
   Presented at NESUG Conference (Sep 2011), Portland, ME (US)

2. **PAKDD 2007 Data Mining Competition: Cross-Selling Problem**
   by Alok Rustagi, Ankur Jain, Krishna Mehta, Mansi Shingla and Mayank Rustagi

3. **Proc Logistic Plus: The Power of Creative Variable Interactions and Transformations**
   by Krishna K. Mehta, Nitin Kumar Jain and Varun Aggarwal
   Presented at NESUG Conference (Sep 2008), Pittsburg, PA (US)

4. **Wikipedia** (http://www.wikipedia.org)

Thanks

For queries, contact Varun Aggarwal at Varun.Aggarwal@exlservice.com