GenIQ, Ltd

Genetic Modeling & Data Mining Software **BRUCE RATNER, PhD**

574 Flanders Drive North Woodmere, NY 11581 br@dmstat1.com
516.791.3544 ~ fax 516.791.5075 www.GenIQModel.com

The GenIQ Model: FAQs

1. **What is the GenIQ Model©?**

The GenIQ Model is a machine-learning (nonstatistical) alternative model to the statistical regression models for binary and continuous target variables, namely, Ordinary Least Squares (OLS) Regression Model, and Logistic Regression Model (LRM), respectively

The major difference between statistical and machine-learning methods is the model-building paradigm used.

- GenIQ uses the paradigm: "the data defines the model."

- Regression uses the converse paradigm: "fit the data to a model" under the assumption that the data analyst's pre-specified model generates the data at-hand (an untenable assumption, especially in big-data settings).

The second difference between statistical and machine-learning methods is the fitness function used, and how the fitness function is optimized.

- OLS: the fitness function is mean squared error (MSE), which is minimized by calculus. (Historical note: Historians generally regard calculus going back to the time of the ancient Greeks, circa 400 BC. Calculus started making great strides in Europe towards the end of the 18th century. Leibniz and Newton pulled their own "to-be-calculus" ideas together, and they are credited with the independent "invention" of calculus. The OLS regression model is celebrating 202 years of popularity, as the invention of the method of least squares was on March 6, 1805.)

- LRM: the fitness function is the joint probability function, which is maximized by calculus. (Historical note: The logistic function has its roots spread back to the 19th century, when the Belgian mathematician Verhulst invented the function, which he named logistic, to describe population growth. The rediscovery of the function in 1920 is due to Pearl and Reed, the survival of the term logistic to Yule, and the introduction of the function in statistics to Berkson. Berkson used the logistic as an alternative to the normal-probability probit model, usually credited to Bliss in 1934, and sometimes to Gaddum in 1933. (However, the probit can be first traced to Fechner in 1860.) As of 1944, Berkson's logistic was not accepted as a viable alternative to Bliss' probit. After the ideological debate about the logistic and probit had abated in the 1960s, Berkson's logistic gained

wide acceptance. Berkson was much derided for coining the term "logit" by analogy to the probit of Bliss, who coined the term probit for "probability unit."

- GenIQ: the fitness function is the decile table (**1**), which is optimized by the Darwinian inspired machine-learning genetic programming (GP). Operationally, "optimizing the decile table" is creating the best possible descending ranking of the target variable values; in other words, to fill the upper deciles with as many responses, or as much profit as possible. (Historical note #1: The decile table, which has its roots in the direct mail business in the 1950s, hallmarked by solicitations inside the cover of matchbooks, has transcended toward the universal measure of model performance. Historical note #2: The first experiments with GP were reported by Stephen F. Smith (1980) and Nichael L. Cramer (1985), as described in the famous book *Genetic Programming: On the Programming of Computers by Means of Natural Selection* by John Koza (1992), who is considered the inventor of GP.
  (1) Required read: Press Ctrl key + Click
  http://www.geniq.net/res/SmartDecileAnalysis.html .

The third difference is that the GenIQ Model is an unparallel data mining tool (discussed in Q10, below), while statistical regression has no data mining capabilities of any kind soever. GenIQ sits well in the work-ground of today's big-data setting because computers, which are necessary for handily housing big data, are also a necessity to strainlessly perform the required Darwinian-like evolutionary computation for mining data. Statistical regression, which has its roots in the small-data setting of the day, 202 years ago, is at-best optimal for the small-data of yesteryear without a loose theoretical thread to pull on to make it scaleable for today's big-data setting, or fashion it with some data mining potentiality. Suffice to say, GenIQ works equally well in both big-data settings and small-data settings (illustrated in Q5, below).

2. What is Genetic Programming?

Paraphrasing Arthur Samuel (1959), genetic programming is an automated methodology inspired by Darwinian evolution that assigns the computer the ability to program itself - to do what is needed to be done without being told (programmed) exactly how to do it!

Genetic modeling is based on the Darwinian ideas of "survival of the fittest" and the natural genetic operators of reproduction (copying), mating (crossover), and mutation (random alteration). The process begins with a fitness function (in GenIQ, the decile table) and a set of user-selectable mathematical and logical functions. A first generation of as many as 250 - 1000 models is randomly generated using the functions and variables available; the "fitness" of each model is evaluated using training data.

A second generation of models is then created through mating, reproduction, and mutation. When two models (parents) "mate" the offspring (children) are mixtures of the parents' genetic material. Thus, each parent probabilistically contributes good genetic material to the child. The frequency with which a model is mated, copied, or altered is a function of its fitness score - how well it fills the upper deciles appropriately. After a suitable number of generations (typically 50 - 100), the forces of natural selection yield the best-of-generation model superbly adapted to the model objective (optimizing the decile table).

For a technical discussion of GP: Press Ctrl key + Click
http://www.geniq.net/Koza_GPs.html

3. **How many variables and records can GenIQ accommodate**?

There are no limitations in terms of the number of variables and the number of records with respect to the GenIQ Software itself. The only limitation is that of the PC used with GenIQ. The more RAM, the more variables and records GenIQ can process.

Importing Data into GenIQ: Press Ctrl key + Click GenIQs 9-step Modeling Process

WARNING: You can not use variables names such as X1, X2, X3, …, Xn, because GenIQ uses these variable names for its coding of the GenIQ Model itself.

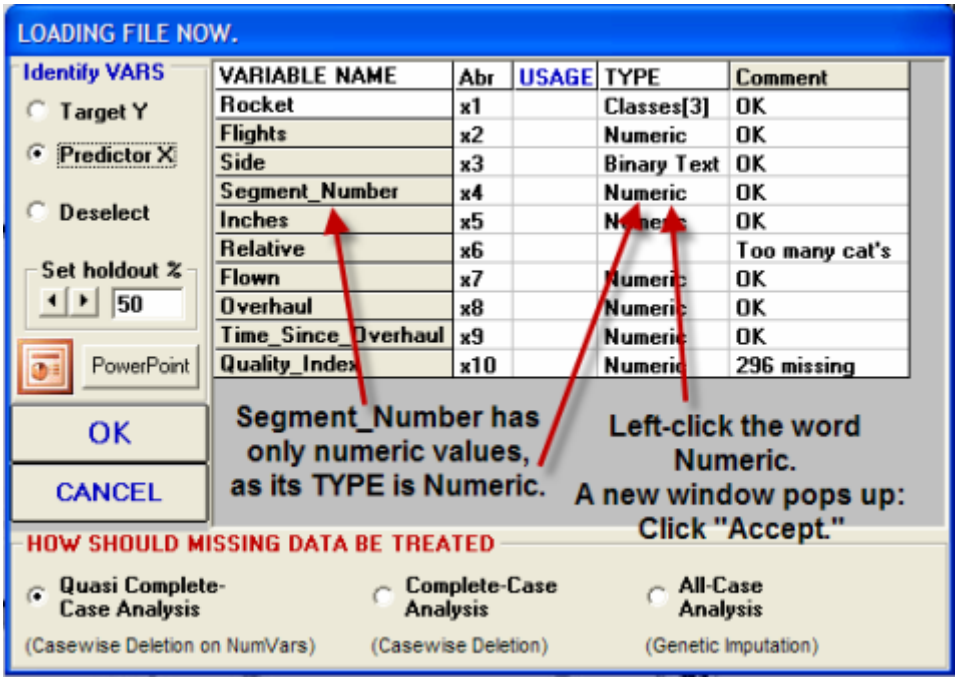
NOTE: GenIQ obviously inputs character variables. If a character variable assumes only non-numeric values, then there is no problem scoring the GenIQ Model code. If a character variable assumes character and numeric values or only numeric values, then scoring the GenIQ Model code is slightly cumbersome. *I recommend for such a character/numeric variable to recode the numeric values to character values*. Otherwise, for the character/numeric values, you have to manually add double quotes (“) before and after the numeric values. For example, consider Marital assumes: M, S, D and 9 (for missing). GenIQ produces something like this:

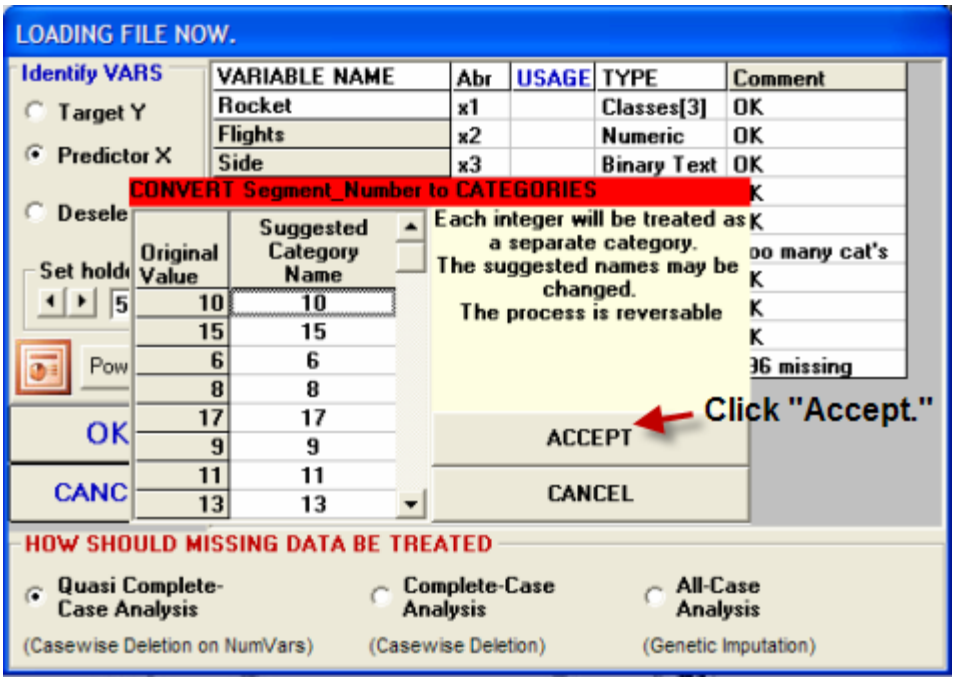
a. If Marital is M, then GenIQ code is: If Marital = “M” then …;
b. If Marital is missing, then GenIQ code is: If Marital = 9 then …; SAS cannot accommodate this syntax efficiently.
c. In case b above, you must add the double quotes. Thus, the coding in b changes to If Marital =”9” then …;

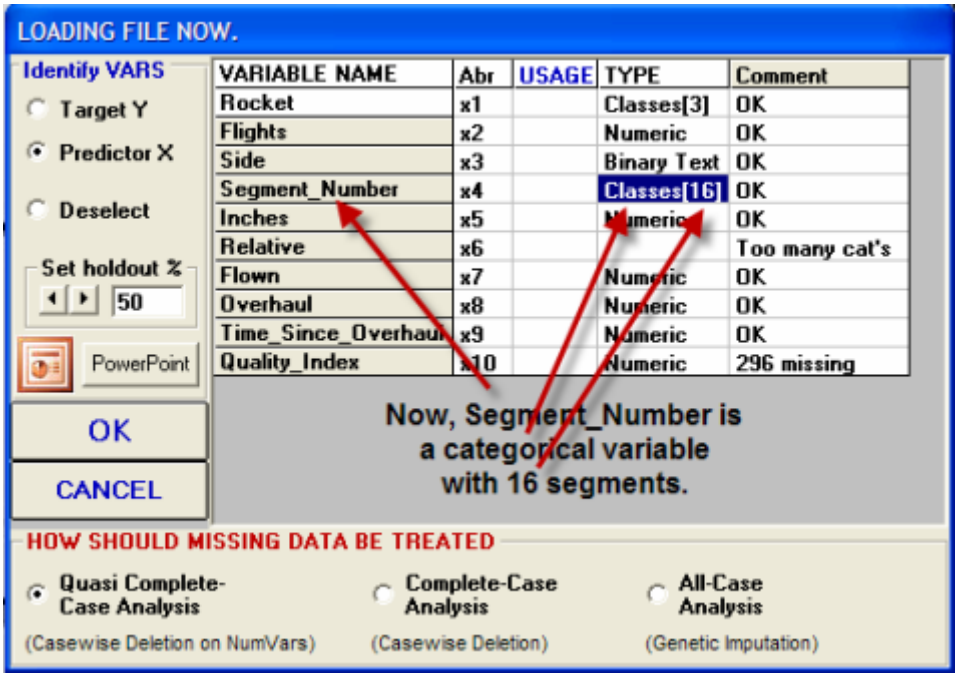
NOTE: GenIQ very conveniently handles categorical variables: GenIQ creates dummy variables for each level of a categorical variable. When the categorical variable has a missing value as a blank, then GenIQ uses the question mark “**?**” as the character to denote the missing/blank value. For example, consider Marital assumes: M, S, D and blank (for missing). GenIQ produces four dummy variables, respectively:

a. “Marital=M” such that if Marital=M then “Marital=M”=1; otherwise “Marital=M”=0
b. “Marital=S” such that if Marital=S then “Marital=S” =1; otherwise “Marital=S” =0
c. “Marital=D” such that if Marital=D then “Marital=D” =1; otherwise “Marital=D”=0
d. “Marital=?” such that if Marital=? then “Marital=?” =1; otherwise “Marital=?” =0

NOTE: If there is a numeric variable whose values represent nominal values, then GenIQ can easily convert the nominal values in class labels. The images below indicate what to do:







NOTE: An extreme event of its kind: GenIQ Off to a Bad Start.
If you notice that the first 5 – 10 generations indicate no improvement in model performance, then GenIQ got off to a bad start with a "bad time of day" seed. The *initial random population* is generated by a random seed, typically, based on the time of day of your computer's clock.

Remedy: Click PAUSE > MAIN MENU > Genetic > Restart GenIQ (at Generation 0).

TIP: Unless the position of the target variable (target_variable) is among the top, say, five variables in a dataset of many variables, scrolling down in the GenIQ input window to mark the target variable as such is somewhat tedious. Accordingly, to position the target variable in the first position, use the SAS data step *retain* statement, below:

```
Data my.data;
Retain target_variable;
Set   my.data;
Run;
```

4. **What kind of data preparation and exploratory data analysis (EDA) are required?**

GenIQ is a tool to be used virtually without data preparation – except for insuring there are no impossible or improbable values (e.g., age of 120 years, or a boy named Sue, respectively). If one considers outliers (unlikely values in a trend or pattern) as a separate data preparation issue from the two exceptions above, the issue is resolved: GenIQs fitness function of "rank-order" optimization moderates such values, rendering them without undue influence on the final GenIQ model. For more on outliers see Q11a. There is an excellent illustration of how GenIQ moderates outliers.

GenIQs inherent by-product of the genetic programming methodology, discussed in Q10, below, uniquely addresses the mandatory trinity of Tukey's EDA:

1. Symmetrizing original variables
2. Straightening pairs of original variables, and
3. Re-expressing two or more original variables to uncover a "new" variable (structure) with the use of relationship and symbolism of numbers and quantitative operations.

5. **What is the output of the GenIQ Model?**

The output of the GenIQ Model is two-fold: a graph known as a parse tree, and computer code that is the model equation. A parse tree is comprised of variables (leafs), which are connected to other variables with functions (e.g., arithmetic {+, -, /, x}, trigonometric {sine, tangent, cosine}, Boolean {and, or, xor}). See the GenIQ parse tree in Figure 1, below. Actually the GP-division is "%," to protect division by zero.

The GenIQ Response Model Tree, in Figure 1, reflects a model for predicting Response (1 = yes, 0 = no) based on two variables XX1 and XX2 in the data in Table

1, below. The GenIQ predicted Response variable, GenIQvar, reflects a unitless number whose interpretation is: the larger the GenIQvar value, the greater the responsiveness. GenIQ also converts GenIQvar values into probabilities of response (Prob_Response). Both of these variables are in Table 1. Note: The GenIQ Response Model produces a prefect ranking of responders and nonresponders, with a notable granularity of GenIQvar values that discriminate within both responders and nonresponders, and between responders and nonresponders. This is an indicator of a utile model. Additional Note: Granularity of any model score values is an indicator of a utile model. GenIQ has an option that performs a Quasi N-tile Analysis to assess its score granularity.

1. **Click** "PAUSE." **Click** the "VIEW MODELS" button.
2. **Left-Click** the blue banner of the Decile Analysis in the top-left panel. The small-text option "Quasi Analysis" appears. **Click** "Quasi Analysis."
3. **Click** various "N-Tile" values to assess the granularity of the GenIQvar values.
4. Required read: Press Ctrl key + Click http://www.geniq.net/res/SmartDecileAnalysis.html .

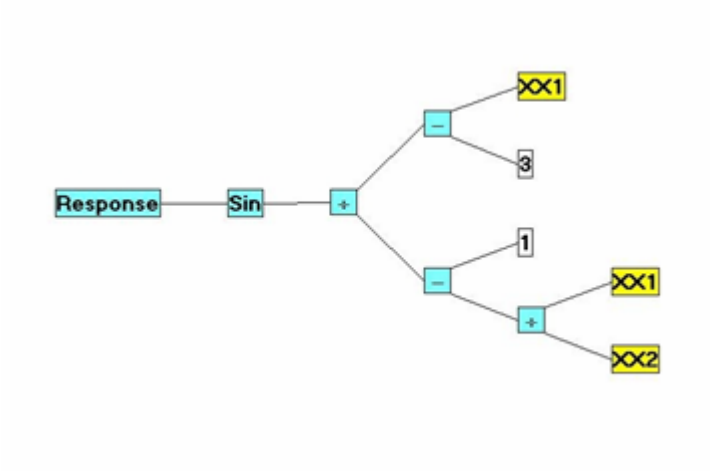


Figure 1. GenIQ Response Model

Table 1. Response Data with GenIQ Model Scores

Response	XX1	XX2	GenIQvar	Prob_Response
1	6	10	0.93800	1.000E+00
1	31	38	0.93332	1.000E+00
1	45	5	0.85893	1.000E+00
1	30	30	0.84147	9.999E-01
1	35	21	0.76825	9.827E-01
0	12	30	0.65029	1.488E-02
0	45	37	0.50445	5.749E-07
0	16	13	0.21367	8.862E-16
0	23	30	-0.77788	7.910E-46
0	30	10	-0.80378	1.297E-46

The GenIQ Response Model computer code (model equation) is in Table 2, below. Note the "Drop" statement, which GenIQ automatically generates, at the end of the code for the intermediate variables x1 – x3. The Drop statement is necessary if the data analyst "re-uses" the full GenIQ tree and/or any branch (i.e., mini-model) in subsequent GenIQ runs. (I discuss "Why the Drop Statement is Necessary," below.) Re-using the full GenIQ tree, and/or any of its branches, which are genetically data-mined structure (newly evolved candidate predictor variables), simply means to append these variables to the data at-hand. "Data Reuse" is discussed in the section "Why the Drop Statement is Necessary," and later in this section under the hybrid statistic-machine-learning paradigm, and in Q10, below.

Table 2. The GenIQ Model Computer Code (model equation)

```
  x1 = XX2;
      x2 = XX1;
   If x1 NE 0 Then x1 = x2 / x1; Else x1 = 1;
      x2 = 1;
   x1 = x2 - x1;
      x2 = 3;
          x3 = XX1;
      x2 = x3 - x2;
   If x1 NE 0 Then x1 = x2 / x1; Else x1 = 1;
   x1 = Sin(x1);
GenIQvar = x1;
Drop x1 – x3;
```

Unfortunately, GenIQ produces a tree with a Picasso-like abstractness, which is not very easy on the eyes, or friendly for interpretation. Nothing beats the coefficients (weights) in the sum of weighted predictor variables that defined a regression model for interpretability. But, regression cannot find structure to compete with GenIQ, as illustrated in the GenIQ Response Model Tree. So, there is a trade-off to be made:

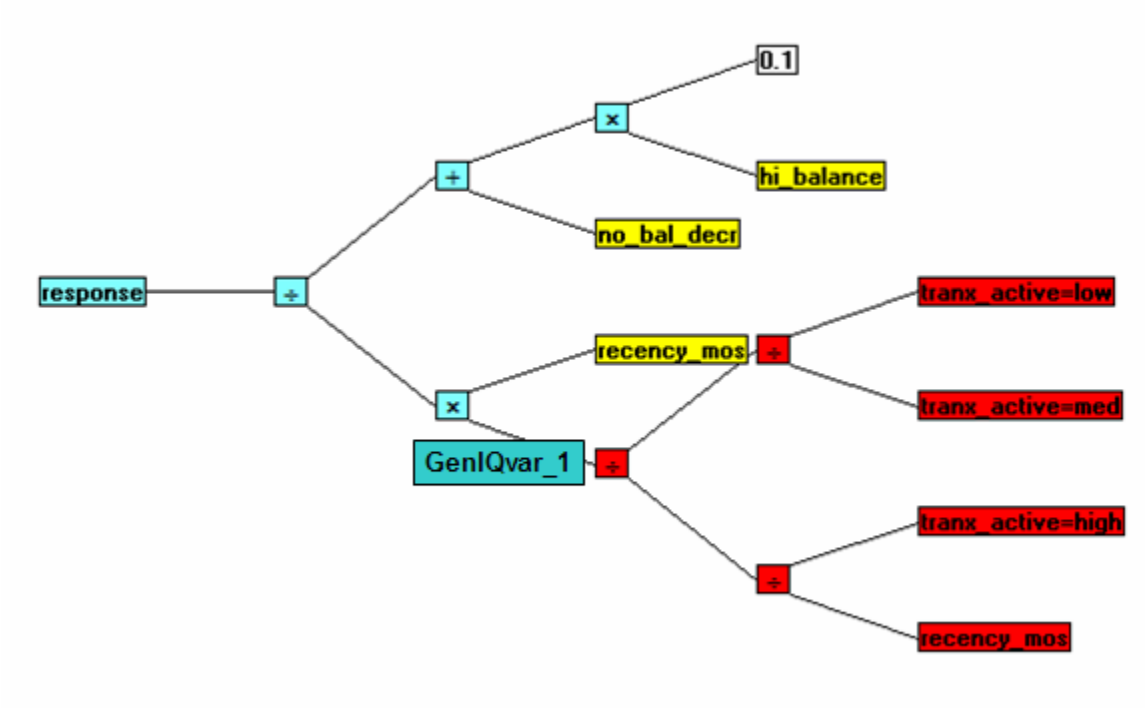
- Accept the "white-box" GenIQ Model with its primo predictiveness, and its unique data-mining capableness as indicated by the branches, which are defined at a "stem" function in the GenIQ tree (discussed in Q10). Admittedly, the GenIQ tree is only a visual comfort, as it allows the data analyst to see the innards of the GenIQ compute code/model equation. The GenIQ Model is difficult to interpret, in part, because it has no coefficients. Tyros and experienced analysts when interpreting a model unwittingly seek the regression coefficients, as they are the means to interpret the everyday logistic and ordinary regression models.

- Accept a highly interpretable regression model with the best subset of the original candidate predictor variables as determined by one of many statistical criterions, e.g., Rsquared. But, no newly constructed variables are possible.

- However, there is a compromise between the two above acceptances: A hybrid statistics-machine-learning paradigm that yields a utile alternative for modeling. The data analyst fits the data to the regression model with the original variables and the genetically-constructed variables (any of the predictive branches of the GenIQ tree). Thus, the hybrid regression-GP model includes a) the redoubtable regression coefficients, which provide the necessary comfort level for model acceptance, and b) the probably inclusion of powerful, genetically constructed variables. Any predictive branches of the GenIQ tree along with their corresponding computer codes should be "copied and pasted" into Power Point (which was opened at input screen #5).

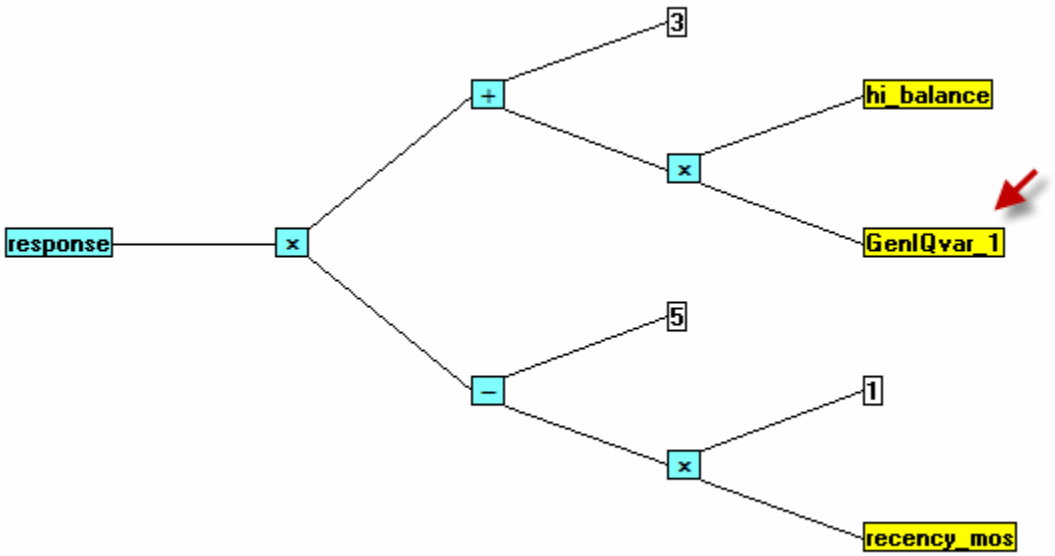
5a. Why the Drop Statement (DS) is necessary. How the DS is related to Data Reuse.

I illustrate why it is necessary to drop the intermediate variables. Consider the GenIQ Model tree, below, from which I identify a genetically data-mined structure, highlighted in red, labeled GenIQvar_1, and whose computer code follows the tree. I chose GenIQvar_1 for its predictive power (not shown).



```
x1 = recency_mos;
      If tranx_active = "high" Then x2 = 1; Else x2 = 0;
   If x1 NE 0 Then x1 = x2 / x1; Else x1 = 1;
      If tranx_active = "med" Then x2 = 1; Else x2 = 0;
         If tranx_active = "low" Then x3 = 1; Else x3 = 0;
      If x2 NE 0 Then x2 = x3 / x2; Else x2 = 1;
   If x1 NE 0 Then x1 = x2 / x1; Else x1 = 1;
GenIQvar_1 = x1;
drop x1 - x3;
```

Now, I re-run GenIQ with the original dataset now appended with the new variable, GenIQvar_1. (This is an example of Data Reuse.) The GenIQ Model I select for its predictive power (not shown), includes variable GenIQvar_1, whose tree and computer code are displayed, below. Note that the computer code for GenIQvar_1 has the intermediate variables x1 – x3. If these latter intermediate variables are not dropped from the code, then the computer code for the final GenIQ Model, which has its own intermediate variables x1 – x3, would be using the intermediate variables x1 – x3 from GenIQvar_1. The results of calculating GenIQvar for the final GenIQ Model would be meaningless. Thus, when re-using genetically data-mined variables from previous GenIQ runs, the dropping of the intermediate variables is clearly necessary.



```
x1 = recency_mos;
     x2 =  1;
  x1 = x1 * x2;
     x2 =  5;
  x1 = x2 - x1;
     x2 = GenIQvar_1;
        x3 = hi_balance;
     x2 = x2 * x3;
        x3 =  3;
     x2 = x2 + x3;
  x1 = x1 * x2;
GenIQvar = x1;
drop x1 - x3;
```

NOTE about Copy and Paste: Upon exporting the computer code (see Q15), the Notepad pops up. Highlight the code, then copy and paste onto a Power Point slide. Before copying the tree, actually the entire four-window GenIQ screen, **take note** to see if any of the yellow-boxed variables and/or turquoise-boxed functions overlaps. If so, then **resize** the tree window such that all variables and function are not overlapping. If resizing is not removing the overlap, the "FONT" buttons located on the header of the Tree window will be helpful. Click the "FONT+" button once or twice, and then click the "FONT– **"**button once or twice, and if necessary resize again. You may have to repeat resizing and clicking the FONT buttons several times before the entire tree is without any overlap.

To copy the resized tree: press the Print Screen key. Then, paste the resized tree (which is on the clipboard) onto a new Power Point slide. Lastly, annotate the code and tree slides, **as you should be collecting several predictive branches and the entire tree as you monitor the evolution of GenIQ Model during the early, say, generations #10, #20 and #30.** If you have a lot of data, a generation could take, say, 50 minutes. If a generation takes 5 minutes, then each tenth-generation takes 50 minutes! So, instead of "watching paint dry," set the number of generations to stop at the tenth, twentieth, and thirtieth generations, for data mining at the potentially predictive 10-generation intervals. After the GenIQ session, you can format data mining codes and trees in Power Point as required.

6. **How does GenIQ handle missing data?**

- GenIQ provides three methods of handling missing data. One, the traditional complete-case analysis: deletes any record for which a candidate predictor variable has a missing value.

- Quasi Complete-case Analysis is identical to complete-case analysis, except if a categorical candidate predictor variable has a missing value then the record is not deleted. GenIQ "dummifies" the categorical variable, say, with k values;

i.e., k dummy variables are created, and uses all k dummy variables. Note: GenIQ uses the "?" for the dummy variable with a missing.

In contrast, all k dummy variables cannot be used in regression, because the set of k dummy variables creates a perfect multicollinearity condition (i.e., one variable can be determined by a linear relationship among other variables), which literally prevents the calculation of the regression model. In the case of dummy variables, any one dummy variable is uniquely determined by the remaining (k – 1) dummy variables. The data analyst has to remove one of the dummy variables in order for the regression model to be derived.

- All-case Analysis genetically imputes missing values for all candidate predictor variables with missing values. Missing-data guru Rubin warns: "All imputation methods are seductive and dangerous." GenIQs works well when missingness is moderate. Must read: Press Ctrl key + Click http://www.dmstat1.com/res/DataPrepSampleSize_.html .

7. **How does GenIQ handle multicollinearity?**

Multicollinearity is not an issue for GenIQ. From the discussion in Q6 above regarding the way GenIQ handles dummy variables, suffice to say multicollinearity, whether a perfect multicollinearity condition (such as with dummy variables), or a near multicollinearity condition, is not an issue for GenIQ. Multicollinearity is a "data" problem for regression models, as it affects the standard errors of the regression coefficients. GenIQ has no coefficients. As well, the discussion of Data Reuse in Q11a, below, renders the multicollinearity issues a nonissue for GenIQ.

7a. **How does GenIQ handle outliers?**

In statistics an outlier is an observation that lies outside the overall pattern of the rest of the data. Outliers can also occur when comparing relationships between two or more variables. Outliers of this type can be easily identified on a scatterplot. When performing regression modeling a single outlier will often render the resultant model misleading. Discarding outliers is a controversial practice frowned on by many statisticians and data analysts. While mathematical criteria provide an objective and quantitative method for data rejection, they do not make the practice more scientifically or methodologically sound. The GenIQ moderates outliers, instead of discarding them, by transforms the independent variables to straighten a given relationship. For an illustration of how GenIQ handles outliers, Press Ctrl key + Click: http://www.geniq.net/res/AMethodForModeratingOutliersInsteadDiscardingThem.pdf

8. **How does GenIQ handle overfitting?**

GenIQ is just as susceptible to overfitting as any other modeling technique, which seeks a solution by optimization. However, GenIQ is potentially less prone for overfitting as its fitness function has a component to moderate overfitting. If your GenIQ Model validation produces unacceptable overfitting, Press Ctrl key + Click: http://www.geniq.net/res/overfitting-old-problem-new-solution.html .

To put in order the issues of overfitting I discuss "What is an overfitted model?" An overfitted model is one that approaches reproducing the data on which it is built (training data), capturing the idiosyncrasy of the data by including unnecessary predictor variables (as indicated by their large p-values). When such a model is applied to new representative data (hold-out, or test data) of the population from which the training data was drawn, the predictions will have immoderate variability (error variance). This is because the model is applied to test data, producing predictions based on the spurious contributions of the unnecessary variables. Symptomatically, an overfitted model shows deterioration in model performance on test data vis-à-vis model performance on training data. In other words, if the test error increases while the training error steadily decreases then a situation of overfitting has probably occurred.

In contrast, a well-built model is one that represents the training data, capturing overall trends and patterns in the data by including only necessary variables (as indicated by their equivalent small p-values). When such a model is applied to test data, which is (assumingly) representative of the population, the predictions will be with acceptable bias and variability. This is because the model is applied to test data, producing predictions based on reliable contributions of only the necessary variables.

8a. **How does GenIQ show validation results based on the hold-out data, selected at the GenIQ setup?**

1. Note: GenIQ upon importing the entire dataset first randomizes it. Then, GenIQ creates the training and hold-out datasets after selecting "% for hold-out." The implication is comparing GenIQ results with a competing model is tricky, because you have no way of getting the randomized versions of the training and hold-out datasets. The best approach of comparing GenIQs competitive performance is to apply a new hold-out dataset to both the final GenIQ Model and the competing model, and then assess the two resultant decile analyses.
2. **Run** the GenIQ Model Software.
3. When you are satisfied with the evolved GenIQ Model, **click** the "PAUSE' button.
4. **Click** the "VIEW MODELS" button.

5. **Left-Click** the blue banner of the Decile Analysis panel in the top-left of the screen. The small-text option "Apply to ... "appears between the large "CONTINUE" and "PAUSE" rectangular option buttons.
6. **Left-Click** "Apply to ... "A drop-down menu appears: "Training data" is greyed-out (because you are building a GenIQ model with these data). "HoldOut data" is blackened.
7. **Click** "HoldOut data." The Decile Analysis changes to show the GenIQ Model under consideration with the hold-out data.
8. **Assess** validation results, **after which DO NOT forget to return to the training data**.
    a. **Left-Click** the blue banner of the Decile Analysis panel.
    b. **Left-Click** "Apply to ... "A drop-down menu appears: "HoldOut data" is greyed-out (because you are assessing the GenIQ model with these data). "Training data" is blackened.
    c. **Click** "Training data."
9. **Click** "CONTINUE" if you want to resume building the GenIQ Model.

9. **How does GenIQ perform variable selection?**

The GenIQ Model provides a unique variable selection of important predictor variables, as it provides the ranking of the relationship between each predictor variable with the target variable – accounting for the presence of the other predictor variables jointly considered. The statistic used is the MEAN FREQUENCY of a predictor variable within the top twenty-five best models. This is in stark contrast to the statistical correlation coefficient, which provides the ranking of the linear-relationship between each predictor variable with the target variable – without considering the other predictor variables.

The GenIQ variable selection process is automated regardless of the number of candidate predictor variables. However, when there are hundreds to thousands of candidate predictor variables, like with any tool, there are know-hows to getting the most out of the tool. The same applies to the GenIQ tool for variable selection with umpteen variables. The recommended procedure is:

1. With GenIQ launched and a GenIQ tree model in the top-right panel, note the "VARIABLE IMPORTANCE" panel in the lower-right of the screen.
2. The variables in this panel are ranked in terms of their predictive importance as per GenIQs statistic MEAN FREQUENCY. Note the magnitudes of the MEAN FREQUENCY.
3. **Find** the variable that displays a sudden drop in the MEAN FREQUENCY values. That variable defines the cut-off point, above which all variables are declared the most important. **Say**, the cut-off variable is in rank position 76.
4. **Click** "MAIN MENU" button. Small-text options will appear above the large rectangular option buttons. Note the "Statistics" option above the greyed-out "PAUSE" button.

5. **Left-Click** "Statistics." A drop-down menu is displayed.
6. **Select** "Variable Selection." A pop-up window appears asking how many important variables to do want to keep.
7. **Input** the value from step #3: 75, one less than the cut-off variable's rank position.
8. **Click** "OK." GenIQ starts to re-run, this time with only the important variables, avoiding the creeping-in of spurious variables that increases the likelihood of an overfitted GenIQ model.

At step #3, a genetic data reduction has taken place that can be used in another application.

9a. **How does GenIQ perform "function" selection?**

GenIQ has "eliminated" the problem of variable selection, but has created another problem: Which functions to select among those in the GENETIC ALPHABET SELECTOR? No. GenIQ has not created another problem. The reason is based on the following recommended procedure.

1. Use the default function setting, which includes addition, subtraction, division, and multiplication, along with numerical material (0.1, 1, 3, 5, and Rand {random numbers}).
2. **Run** GenIQ as discussed above.
3. To select other functions, **click** "PAUSE."
4. The small-text option "Genetics" appears above the large "CONTINUE" rectangular option button.
5. **Click** "Genetics." A drop-down menu appears with two options, one of which is "Resign Genetic Alphabet."
6. **Left-Click** "Resign Genetic Alphabet," after which the GENETIC ALPHABET SELECTOR screen appears.
7. **Use** the rule-of-thumb:
    - **a.** If you have dollar-unit variables, **select** Log.
    - **b.** If you have discrete variables, **select** Logicals (AND, OR, and XOR).
    - **c.** If you have continuous variables, **select** Circular functions (Sine, Cosine, and maybe Tangent).
    - **d.** If you are "moved-to-mine" the data," **select** other functions.
8. After selecting functions, **click** "OK."
9. The GenIQ panels appear. But, this time the VARIABLE IMPORTANCE panel is replaced with "FUNCTION IMPORTANCE" panel.
10. The FUNCTION IMPORTANCE panel displays a bar chart for all the functions, the original default ones, and the newly selected one.
11. **Run** GenIQ for 25 - 50 generations. Assess the bar chart to determine which functions are important: functions with short bars are not important.
12. **De-select** the unimportant functions by following steps 3 – step 8, replacing "select" with "de-select."

13. **Run** GenIQ until you are satisfied with the evolved GenIQ Model.
14. To restore the VARIABLE IMPORTANCE panel, **left-click** once in the middle of the FUNCTION IMPORTANCE panel.

10. **How does GenIQ perform data mining?**

The GenIQ Model provides automatic data mining – an inherent by-product of the genetic programming methodology. GenIQ genetically evolves (data mines) predictive structures as indicated by the branch, which is defined at a "stem-function" in the GenIQ tree. Although a branch is defined at a single stem-function, it can have more than one function within the branch itself.

With GenIQ launched and a GenIQ tree model in the top-right panel, **left-click** each function, which highlights the branch in red. For example, I left-click the division function, at the right-side bottom in the GenIQ Response Model Tree, which highlights the XX1/XX2-branch in red, in Figure 2, below. Because a branch is a mini-model, it has its own computer code (mini-model equation). The computer code for the XX1/XX2-branch is in Table 3, below.

To determine the branch's predictiveness compare the changes in the CumLifts in a
branch's decile table vis-à-vis the CumLifts in the full GenIQ tree's decile table.

The most predictive branches can be exported (discussed in Q12, below) for either a hybrid regression-GP model, or re-use in the GenIQ modeling process (discussed in Q11, below).

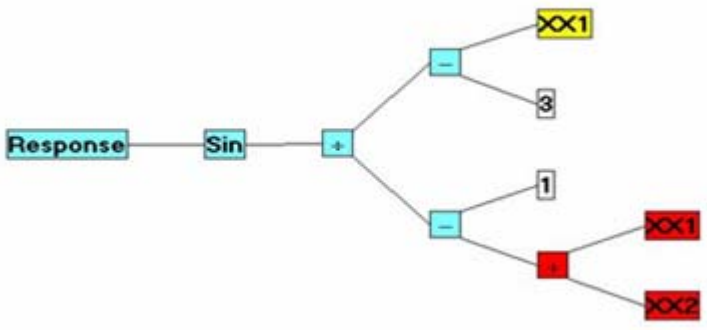


Figure 2. Genetic-evolved (data-mined) Structure

Table 3. GenIQ Branch Computer Code (mini-model equation)

```
x1 = XX2;
      x2 = XX1;
   If x1 NE 0 Then x1 = x2 / x1; Else x1 = 1;
      x2 = 1;
   x1 = x2 - x1;
GenIQvar_Branch = x1;
Drop x1 - x2;
```

10a. When Data Mining results in a "flipped" decile table.

"FLIPPED" VARIABLE-BRANCH HOT KEY: If a highlighted variable or branch produces a "flipped" decile table, i.e., the top decile has a CumLift value less than 100, or equivalently, the quantity of responders/profits is decreasing from top to bottom deciles, then the variable/branch is negatively correlated to the target variable. To assess the predictive power of a negatively correlated variable/branch, the flipped decile table has to be "un-flipped," which is done by multiplying the variable or branch-code by negative one (-1), and then recalculating the decile table. GenIQ has a *Flipped Variable-Branch Hot Key* that automatically un-flips the decile table. The procedure to un-flip the decile table consists of the three steps described and illustrated below:

1. **Left-click** a variable or branch (actually, the function defining the branch) of interest. If the corresponding decile table is "up-right," namely, the quantity of responders/profits is increasing from top to bottom deciles, there is nothing to do. The corresponding decile table is correctly indicating the predictive power of the variable/branch of interest.
2. If the variable/branch of interest is **flipped:**
    a. **Remove** the left-click from the variable/branch (Note: the variable/branch is still highlighted in red).
3. **Press and hold-down** the **Ctrl** key and then **left-click** the variable/branch. The decile table is now un-flipped. The affected, correct decile table now indicates the predictive power of the variable/branch of interest.

To refresh the highlighted variable/branch, and decile table, **left-click** the target variable (response) in the tree.

Consider the following GenIQ output in Figure 3, below. Top decile CumLift is 203, along with 38 and 16 responses in the top and bottom deciles, respectively. I am interested in variable **recency_mos**. I **left-clicked recency_mos**: The corresponding decile table is flipped, in Figure 4, below: Top decile CumLift is 80, along with 15 and 24 responses in the top and bottom deciles, respectively.

Figure 3. GenIQ Model Output of a Given Model

RESPONSE DECILE ANALYSIS

TRAINING SET weighted sample size = 3740
Population response = 5.00%

DECILE	Predicted Response	Resp Rate%	Cum RespR%	Cum Lift
Top	38	10.16	10.16	**203**
2nd	24	6.42	8.29	**166**
3rd	20	5.35	7.31	**146**
4th	15	4.01	6.48	**130**
5th	12	3.21	5.83	**117**
6th	19	5.08	5.70	**114**
7th	13	3.48	5.39	**108**
8th	17	4.55	5.28	**106**
9th	13	3.48	5.08	**102**
Bottom	16	4.28	5.00	**100**

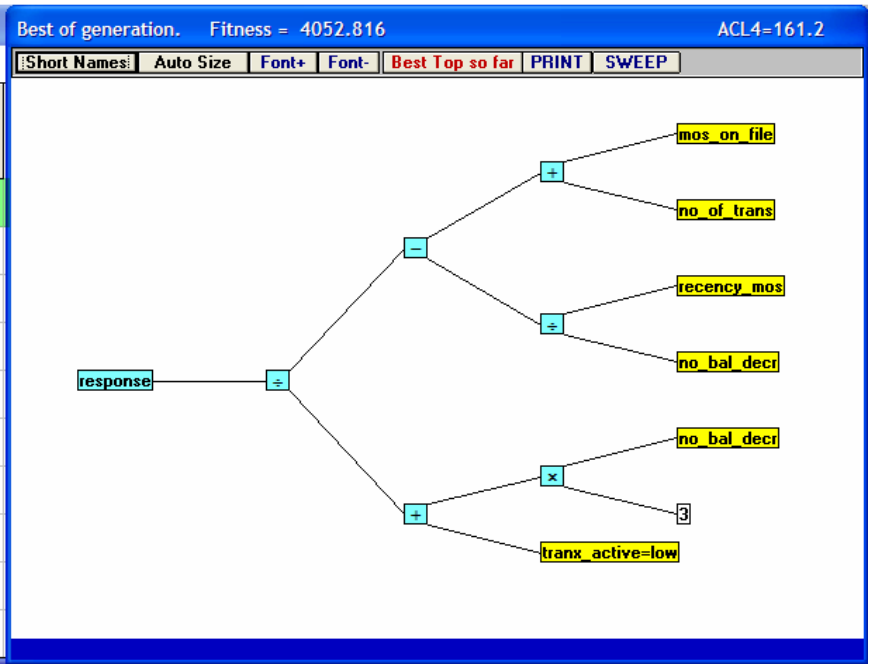
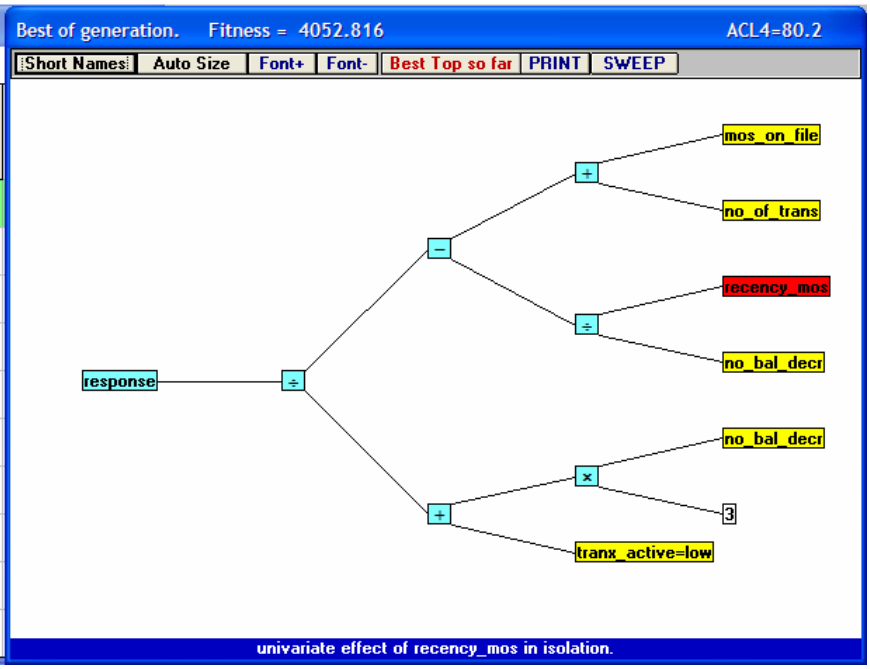


Figure 4. Flipped Decile Table Due to Negatively Correlated Variable "recency_mos"

RESPONSE DECILE ANALYSIS

TRAINING SET weighted sample size = 3740
Population response = 5.00%

DECILE	Predicted Response	Resp Rate%	Cum RespR%	Cum Lift
Top	15	4.01	4.01	**80**
2nd	14	3.74	3.88	**78**
3rd	16	4.28	4.01	**80**
4th	17	4.55	4.14	**83**
5th	20	5.35	4.39	**88**
6th	16	4.28	4.37	**87**
7th	27	7.22	4.77	**95**
8th	22	5.88	4.91	**98**
9th	16	4.28	4.84	**97**
Bottom	24	6.42	5.00	**100**



To determine the predictive power of **recency_mos** I follow the three steps, above, to un-flip the decile table, in Figure 5, below. Top decile CumLift is 128, along with 24 and 15 responses in the top and bottom deciles, respectively. Thus, the predictive power of the

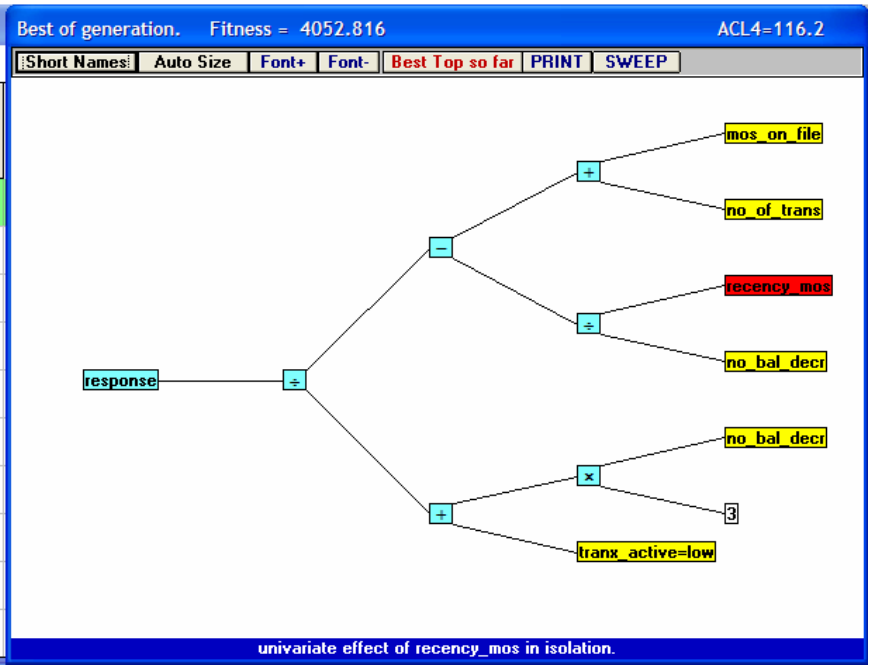
individual variable **recency_mos** at the top decile is subjectively declared moderate with a value of 63.05%. The 63.05% is the percentage of the **recency_mos** top decile CumLift (128) with respect to the GenIQ Model (entire tree) top decile CumLift (203): 63.05% (={128/203}*100.

Figure 5. Un-flipped Decile Table to Determine the Predictive Power of "recency_mos"

RESPONSE DECILE ANALYSIS

TRAINING SET weighted sample size = 3740
Population response = 5.00%

DECILE	Predicted Response	Resp Rate%	Cum RespR%	Cum Lift
Top	24	6.42	6.42	**128**
2nd	16	4.28	5.35	**107**
3rd	22	5.88	5.53	**111**
4th	27	7.22	5.95	**119**
5th	16	4.28	5.61	**112**
6th	20	5.35	5.57	**111**
7th	17	4.55	5.42	**108**
8th	16	4.28	5.28	**106**
9th	14	3.74	5.11	**102**
Bottom	15	4.01	5.00	**100**



I want to clarify what to do in step #1, above, if a branch is of interest, say, **recency_mos** divided by **no_bal_decr**. Let's call this branch the "Red Branch." Step #1 indicates that the function defining the branch is **left-clicked**. In this case, the **left-click** is on the division function, as indicated in Figure 6, below. This branch has a corresponding flipped decile table. The unflipped branch decile table is in Figure 7, below. Parenthetically, the negatively correlated Red Branch has 95.07% (= {193/203}*100) of the predictive power of the GenIQ Model, at the top decile. I declare that the predictive power of the Red Branch is very powerful, and should be re-used in a subsequent re-run of GenIQ.

Figure 6. Flipped Decile Table Due to Negatively Correlated "Red"

RESPONSE DECILE ANALYSIS

TRAINING SET weighted sample size = 3740
Population response = 5.00%

DECILE	Predicted Response	Resp Rate%	Cum RespR%	Cum Lift
Top	16	4.28	4.28	**86**
2nd	17	4.55	4.41	**88**
3rd	14	3.74	4.19	**84**
4th	12	3.21	3.94	**79**
5th	16	4.28	4.01	**80**
6th	19	5.08	4.19	**84**
7th	15	4.01	4.16	**83**
8th	19	5.08	4.28	**86**
9th	23	6.15	4.49	**90**
Bottom	36	9.63	5.00	**100**

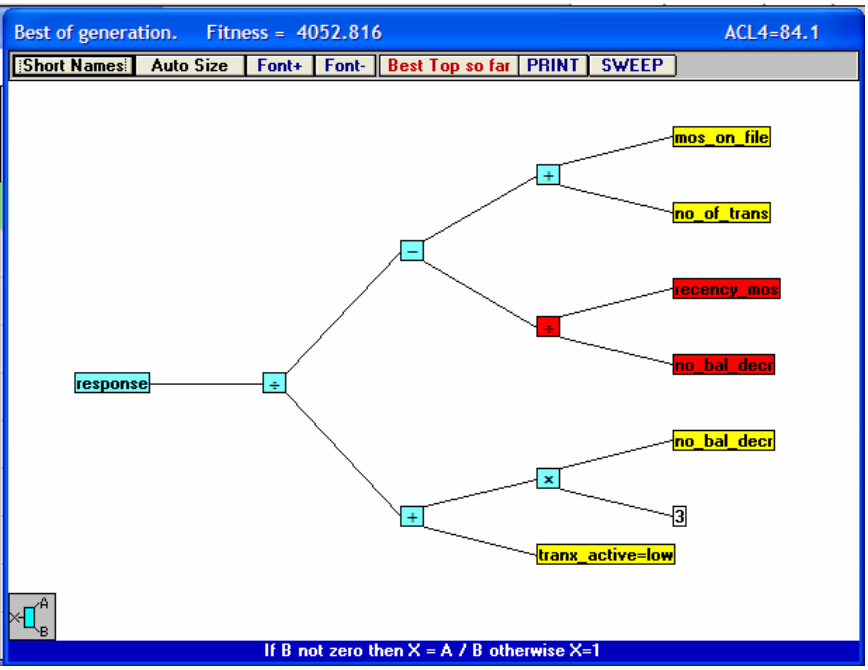
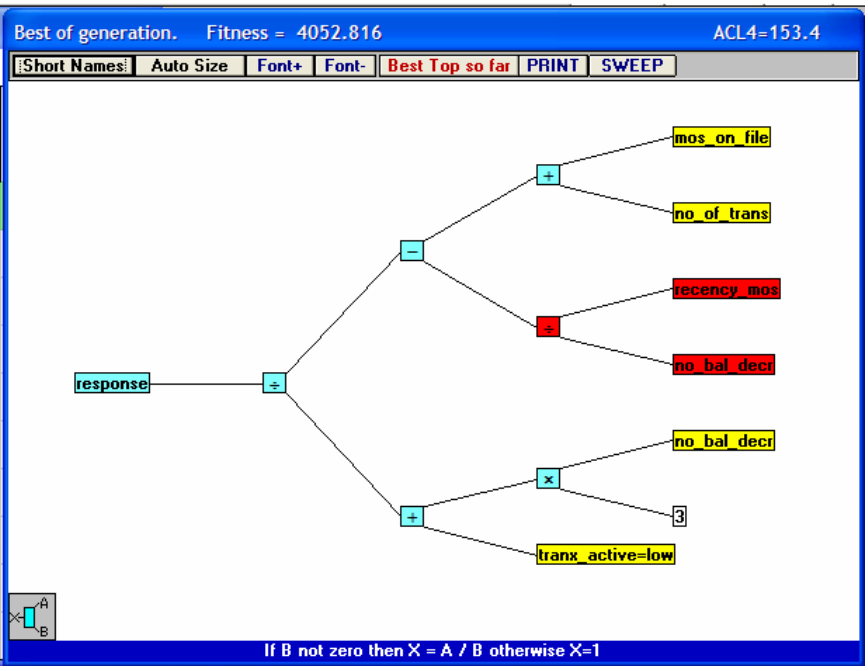


Figure 7. Unflipped Decile Table to Determine the Predictive Power of the Red Branch

RESPONSE DECILE ANALYSIS

TRAINING SET weighted sample size = 3740
Population response = 5.00%

DECILE	Predicted Response	Resp Rate%	Cum RespR%	Cum Lift
Top	36	9.63	9.63	**193**
2nd	23	6.15	7.89	**158**
3rd	19	5.08	6.95	**139**
4th	15	4.01	6.22	**124**
5th	19	5.08	5.99	**120**
6th	16	4.28	5.70	**114**
7th	12	3.21	5.35	**107**
8th	14	3.74	5.15	**103**
9th	17	4.55	5.08	**102**
Bottom	16	4.28	5.00	**100**



CAPTURE PREDICTIVE BRANCHES: Let's assume that the XX1/XX2-branch in red is determined to have good predictive power. Whether you use the branch for either a hybrid regression-GP model, or re-use in the GenIQ modeling process, you must **CAPTURE** both the GenIQ tree and GenIQ computer code for the branch; otherwise, you will not have a "paper trail" for final model definition to include in the GenIQ presentation deck. To capture is easy: Any predictive branches of the GenIQ tree along with their corresponding computer codes should be copied and pasted into Power Point (which was opened at input screen #5). See **NOTE about Copy and Paste** in Question #5.

11. **What are the best values for the GenIQ population, breeding, and fitness controls?**

GenIQ has been optimized by the fixed default settings for the breeding and fitness controls.
The genetic population size (GPS) has varying default values, which is automatically set based on the number of candidate predictor variables.

After gaining familiarity with GenIQ tree, it is farsighted for the data analyst to manually test for the optimal GPS. The recommended approach, which immediately follows, should also incorporate the concept of "Data Resuse." Data Reuse in Q11a, below, further helps determine the optimal GPS, as well as whether the number of generations run was too small.

1. Start with a GPS of, say, 250, for 25 – 50 generations; note the model performance.
2. Increase GPS, say, to 500, for 25 – 50 generations; note the model performance.
3. Compare the results of both runs in steps #1 and #2:
    a. If model performance is not improved with the larger GPS (500), then the GPS (250) in step #1 is optimal.
    b. If model performance is improved with the larger GPS (500), then increase GPS to, say, 1000, and re-run GenIQ.
4. Compare the results of the new runs in steps #3b with GPSs 500 and 1000:
    a. If model performance is not improved with the larger GPS (1000), then the GPS (500) in step #2 is optimal.
    b. If model performance is further improved with GPS of 1000, then continue to increase the GPS until no further improvement is obtained. The last GPS, which yields no improvement, is not-yet declared optimal, and not-yet produces the best GenIQ model.
5. Lastly, the data analyst re-uses the full GenIQ tree and the GPS in Step 4b, and re-runs GenIQ one last time to see if the GSP is in fact optimal.
    a. If no improvement is achieved, then that declared GPS is in fact optimal, and the GenIQ model is best.
    b. If improvement in model performance is achieved, then re-run GenIQ with additional increases in the GPS, left to the carefulness of the data

analyst. The resultant run indicates the optimal GPS, and the best GenIQ model.

11a. **Data Reuse: Adjusting for setting too small: number of generations and/or population size.**

When building a GenIQ Model, it is recommended to append the new "genetically data-mined" variables (the branches of the GenIQ tree, including the model itself) to the original dataset, and then re-run GenIQ. **NOTE #1:** When appending GenIQvar_branches, do not forget the drop the GenIQ intermediate x variables that define the branches: Use the (SAS) drop statement Drop x1 — xn; after each GenIQvar_branch. Otherwise only the first GenIQvar_branch will be coded correctly, and all subsequence branches will be meaningless. Re-run of GenIQ is repeated until no better model is evolved. **NOTE #2:** The implication of reusing variables that are statistically correlated, yet not a problem for GenIQ is that multicollinearity is not an issue for GenIQ.

1. First GenIQ Model is complete.
2. Second GenIQ Model: Re-run GenIQ with appending the new variables from the first model, producing the second GenIQ Model.
    a. If the second model is not better than the first model, then you have, in effect, validated the stability of the first model, which is thusly considered the best model.
    b. If the second model is better than the first model, then it indicates that the number of generations run was too small and/or the GPS was set too small. In this case, re-run GenIQ is suggested.
3. Third GenIQ Model: Re-run GenIQ with appending the new variables from the second model, producing a third GenIQ Model.
    a. If the third model is not better than the second model, then you have, in effect, validated the stability of the second model, which is thusly considered the best model.
    b. If the third model is better than the second model, then it indicates that the number of generations run was too small and/or the GPS was set too small. In this case, re-run GenIQ (for a four model) is suggested.
4. This process of re-running GenIQ with appending of the new variables from the latest model is continued until the resultant model is not better than the previous model. In such case, the previous model is validated as stability, and is considered the best model.
5. Re-running GenIQ more than three times is quite rare. For an interesting example of Data Reuse, Press Ctrl key + Click: http://www.geniq.net/HowToUseGenIQ.pdf

12. **How does GenIQ export the computer code?**

1. **Run** the GenIQ Model Software.
2. When you are satisfied with the evolved GenIQ Model, **click** the "PAUSE' button.
3. **Click** the "VIEW MODELS" button. Small-text options will appear above the larger rectangular option buttons. The last one, furthest to the right is "Export."
    a. **Exporting a branch: Left-click** the desired branch, then proceed to step 4.
4. **Click** "Export." A drop-down menu appears.
5. **Click** "Export as shown." A pop-up window appears at the upper left corner.
6. **Click** the radial button "SAS style." Note: "APPEND TO FILE" is checked "on" by default. Until you become acquainted with this procedure, click off this option. Later, you would want this feature "on" when you are testing several GenIQ Models. The feature allows you to annotate the code (in the Notepad which pops up, in the next step) so you would not lose track of which models performed better than others.
7. **Click** "OK" button. A pop-up window appears, indicating "The (SAS) code has been written to file **C:\Program File\GenIQ** *filename* **.txt**." The path is the same as where the training data (say, *filename*) resides. **Click** "OK." The Notepad opens with the computer code/model equation selected in step 3.
8. **Right-click** in the middle of the Notepad. **Choose** "Select all.' **Right-click** again in the middle of the Notepad. **Choose** "Copy."
9. **Close** down the Notepad.
10. **Paste** the model equation in your (SAS) application. Next, **close** down GenIQ by either of two approaches:
11. **Click** the "QUIT" button. The project session is retrievable under the notation: "Last Used dd-month-yy hh:mm:ss".
12. **Click** the "MAIN MENU" button, then **click** on small-text option "File," the first one, furthest to the left, which results in a drop-down menu.
13. **Choose** "Save Project" A "Save project: add memo first" window appears.
14. In the "Notes:" line, **choose** a description for the GenIQ project session.

13. **What is a GenIQ-enhanced Regression Model?**

The GenIQ Model can be used on a final regression model to let GenIQs data mining prowess enhance, data permitting, the results of the final model. GenIQs Utility: Enhance a final regression model by running GenIQ with only one predictor: the final regression equation score. See extra-GenIQ applications:

1. Overfitting: Old Problems, New Solution (Press Ctrl key + Click http://www.geniq.net/overfitting-old-problem-new-solution.html)
2. How to Make the Best Credit Score Even Better (Press Ctrl key + Click http://www.geniq.net/res/how-to-make-the-best-credit-score-even-better.html)

14. **Scoring GenIQ Models with Excel**

1. **Prepare** the dataset for GenIQ in an excel format (xls).
2. **Run** GenIQ Model Software as usual, using the excel dataset.
3. When you are satisfied with the evolved GenIQ Model, **click** the "PAUSE' button.
4. **Click** the "VIEW MODELS" button. Small-text options will appear above the larger rectangular option buttons. The last one, furthest to the right is "Export."
5. **Click** "Export." A drop-down menu appears.
6. **Click** "Export as shown." A pop-up window appears at the upper left corner.
7. **Click** the radial button "VB for Excel." Note: "APPEND TO FILE" is checked "on" by default. Until you become acquainted with this procedure, Click off this option. Later, you would want this feature "on" when you are testing several GenIQ Models. The feature allows you to annotate the code (in the Notepad which pops up, in the next step) so you would not lose track of which models performed better than others.
8. **Click** "OK" button. A pop-up window appears, indicating "The VB code for Excel has been written to the text file name and its path of the **GenIQ Model Equation Code**" (as per step 3). The path is the same as where the excel dataset resides. Click "OK." The Notepad opens with the code of the model selected in step 3.
9. **Right-click** in the middle of the Notepad. **Choose** "Select all.' **Right-click** again in the middle of the Notepad. **Choose** "Copy."
10. **Close** down the Notepad.
11. Next, **close** down GenIQ by either of two approaches:
    a. **Click** the "QUIT" button. The project session is retrievable under the notation: "Last Used dd-month-yy hh:mm:ss".
    b. **Click** the "MAIN MENU" button, then **click** on small-text option "File," the first one, furthest to the left, which results in a drop-down menu.
        i. **Choose** "Save Project" A "Save project: add memo first" window appears.
        ii. In the "Notes:" line, **choose** a description for the GenIQ project session.
12. **Launch** Excel, and then **Open** the excel dataset in use.
13. **Click** "Tools" > "Marco" > "Visual Basic Editor."
14. **Click** "Insert" > "Module." A "Module" window appears.
15. In the Module window, **Right-click** and **choose** "Paste." You have now imported the GenIQ Model "Equation" Code of step 8.
16. **Choose** small-text option "Run." Wait until the Excel macro processing is complete (i.e., the scoring of the GenIQ Model is finished).
17. **Select** the Excel sheet of the dataset at hand. New columns are added to the end of the sheet: Three columns for a GenIQ Response Model, and two columns for a Profit GenIQ Model:

a. The first new column is “Dependent Y1=target variable,” where the “target variable” is the variable selected when defining the target and predictor variables during the GenIQ data input screen. (This column/variable is a duplicate of the column of the target variable, adding here for convenience).
b. The second new column is the **GenIQ Model Score.** Recall, the GenIQ score is a unitless number: the larger its value the greater the responsiveness for a response model, and greater the contribution of profit for a profit model.
c. For the GenIQ Response Model Only: The third new column is the probability score, “Prob (target variable),” which are derived from the unitless GenIQ Model Score.

18. **Save** the Excel sheet with the newly appended GenIQ Model scores.
19. **Proceed** as your desire dictates.