MWSUG 2019 - Paper IN-014

Comparison of Three Methods For Transforming Predictor Variables to Improve Model Fit Using SAS®

Doug Thompson, Rush Health, Chicago, IL

ABSTRACT

Continuous and ordinal predictor variables are common in predictive modeling (e.g., age in years, medical expenditures last year). Often, such variables are non-linearly related to the predictive modeling target. To maximize the accuracy of a predictive model, non-linear associations need to be taken into account and included in the final model when appropriate. There seems to be no consensus on how best to detect and quantify non-linear associations when building predictive models. Several methods have been proposed in the literature, including cubic splines and exploring a wide variety of functional forms then selecting the best-fitting via stepwise techniques. Although multivariate adaptive regression splines (MARS) and similar methods are often viewed as a stand-alone technique for predictive modeling, these techniques could also be used for exploring non-linear associations that are then included in a final model constructed using some other modeling technique (e.g., logistic regression or neural networks). The purpose of this paper is to illustrate three possible methods for exploring non-linear associations in predictive modeling using SAS: Cubic splines, MARS, and stepwise selection of the best fitting of exploratory functional forms. A SAS macro is described, facilitating easy implementation and evaluation of each of these techniques. The techniques illustrated require only SAS/STAT (particularly PROCs ADAPTIVEREG, LOGISTIC and SGPLOT). The audience is assumed to have intermediate familiarity with predictive modeling. The techniques are illustrated in a context that is common and important within the healthcare industry: Predicting which patients will have relatively high healthcare expenditures next year.

INTRODUCTION

Continuous and ordinal predictor variables (e.g., age in years, medical expenditures last year) are common in predictive modeling. Often, such variables are non-linearly related to the predictive modeling target. The slope of association may vary in different ranges of the predictor variable. For example, the association may be quadratic, cubic or some other complex form. (For convenience, “continuous” will be used to refer to such predictor variables throughout this paper, although ordinal numeric variables may not be truly continuous.)

The following example illustrates the issue. The data are from the Medical Expenditure Panel Survey (MEPS), Panel 19, 2014-2015 longitudinal data, limited to individuals younger than age 65 with commercial health insurance throughout 2014 (described further below in Methods). Suppose that the goal is to build a model that predicts high healthcare expenditures for individuals next year, based on information that is known about the individuals this year. Specifically, the proportion of MEPS respondents with healthcare expenditures above $20,000 in 2015 was modeled as a function of 2014 healthcare expenditures. Models were fit with logistic regression, modeling the log of the odds (logit) of having healthcare expenditures >$20,000 in 2015 as a function of healthcare expenditures in 2014 (in ten thousands). For illustration, linear as well as quadratic models were created. The quadratic model was chosen as a simple alternative, to estimate a possible non-linear association. The predictions of each model on the logit scale were transformed to the more intuitive probability scale by taking 1/(1+exp(-1*score on logit scale)). Model predictions were plotted against average actuals, grouped by $10k of spend to avoid having the plot look like a swarm of points. Results are shown in Figure 1 below. The quadratic model fit the data better than the linear model (areas under the ROC curve = 0.803 and 0.796, respectively). As shown in Figure 1, it appears that individuals with 2014 expenditures >$200,000 were actually less likely to have expenditures >$20K in 2015 than were individuals with 2014 expenditures in the $150,000-$200,000 range. One could speculate that the most expensive individuals in 2014 had an acute episode (e.g., expensive car accident) with expenses that did not carry over into 2015, but whatever the reason, the quadratic model fit the data better than the linear model did. Quadratic is a relatively simple non-linear form; perhaps a more complex functional relation (e.g., cubic, spline) would fit the data even better.

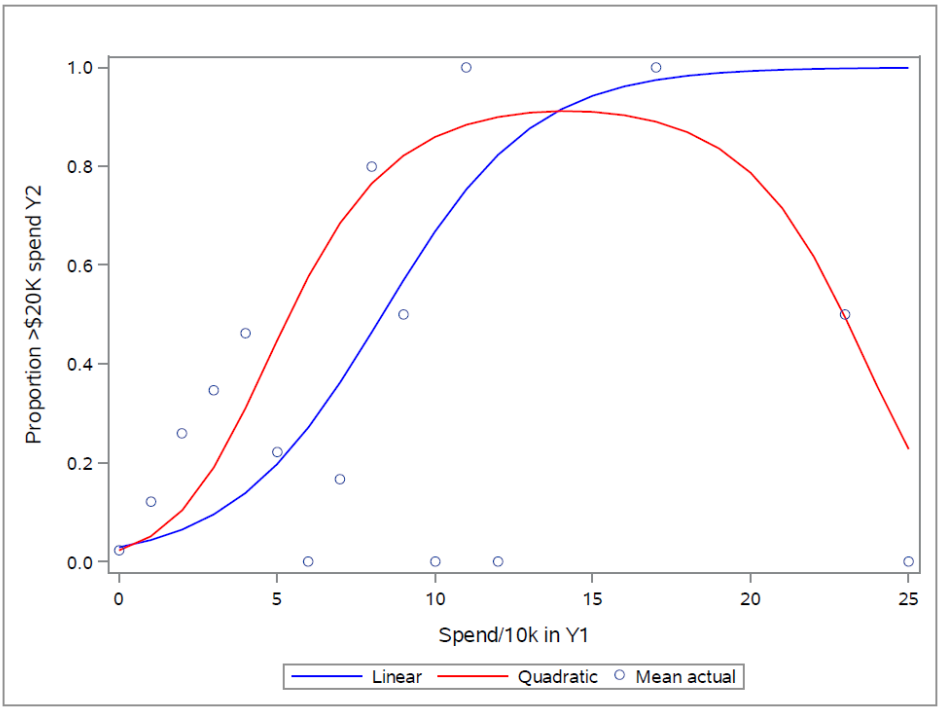

Figure 1. Model-predicted probability and actual proportion of MEPS respondents having healthcare expenditures >$20,000 in 2015 (Y2) by healthcare expenditures (spend) in 2014, in ten thousands (Y1)

As this example suggests, to maximize the accuracy of a predictive model, non-linear associations need to be taken into account when building the model and included in the final model when appropriate.

There seems to be no consensus on how best to detect and quantify non-linear associations when building a predictive model, even though this is typically noted as an important task in the predictive modeling process (e.g., Harrell, 2001; Hastie et al, 2009; Kuhn and Johnson, 2016; Rud, 2001; Baessens et al, 2015). Several methods have been proposed, including cubic splines (Harrell, 2001; Hastie et al, 2009) and exploring a wide variety of functional forms before selecting the best-fitting via stepwise techniques (Rud, 2001). Although multivariate adaptive regression splines (MARS) is often viewed as a stand-alone technique for predictive modeling, it could also be used for exploring non-linear associations that are then included in a final model built using some other modeling technique (e.g., logistic regression, neural networks).

This paper illustrates three possible methods to identify and quantify non-linear associations in predictive modeling using SAS: 1) Linear splines, 2) MARS, and 3) stepwise selection of the best fitting of exploratory functional forms. A SAS macro is described, facilitating easy implementation and evaluation of each of these techniques.

DATA

To illustrate these methods, data from the Medical Expenditure Panel Survey (MEPS) were used. MEPS has been conducted annually since 1996 by the Agency for Healthcare Research and Quality, which is part of the U.S. Department of Health and Human Services. MEPS was designed to describe healthcare expenditures, healthcare utilization and health insurance among the U.S. non-institutionalized, non-military population. MEPS samples households. Information regarding each sampled household is collected for a 2-year period (“panel”) in 5 “rounds” of interviews spaced across 2.5 years. This enables longitudinal analysis of healthcare for individuals in the sampled households during the 2-year period covered in the panel. MEPS consists of a series of overlapping panels. The data are freely available for download. Instructions for importing the data into SAS are available on the MEPS website. Although the data are de-identified to protect respondent anonymity, an individual’s data can be tracked across time by tying it to an individual person ID (“DUPERSID”).

The analyses in this paper used MEPS Panel 19, covering 2014 ("Y1") and 2015 ("Y2"). Data were limited to Panel 19 participants under the age of 65 who had private health insurance throughout 2014, and who had data in both 2014 and 2015 (the latter condition was true of 95% of Panel 19 participants). A total of 5,848 MEPS survey respondents met these criteria. This subset of the data was selected to represent the realistic situation in which an organization responsible for managing the health of a commercially insured population (e.g., within a health insurance plan or commercial Accountable Care Organization) seeks to identify high-risk individuals to engage in healthcare management activities. The model described below was built using data for a randomly selected 2/3 of the individuals while the remaining 1/3 were left aside as a holdout sample to test the final model.

ANALYTIC METHODS

In the methods illustration, the goal was to predict high healthcare expenditures in 2015 (defined as an individual having healthcare expenditures above $20,000) based on information available about individuals in 2014, including demographics, usual healthcare provider relationship, health status, and healthcare expenditures. The model was logistic regression, modeling the log of the odds of having high healthcare expenditures in 2015 as a function of 2014 measures ("predictors"). Four of these predictors were continuous: Healthcare expenditures; household income; age in years; and family size, that is, the count of individuals in the household.

This paper illustrates three techniques to identify the optimal, possibly non-linear representations of these continuous variables in the predictive model. The goal was to use these methods to find and quantify "optimal transformations," meaning transformations of the original variables that maximize a metric of ability to predict the target outcome. The result may not be optimal, strictly speaking -- there may be methods other than these three that yield even better predictions. However, the intention was for the methods to be practical, relatively easy to implement, and to show a significant improvement over a simple linear representation when appropriate.

The first method was linear splines with a pre-specified set of knots. Spline models are piecewise regression models, allowing different segments to have different slopes. The segments are continuous, that is, one segment joins the next with no breaks in between. The join points are called "knots." Suppose that the x-axis is divided into intervals with knots at a and b. Then the linear spline model is: $F(X)=\beta_0+\beta_1X+\beta_2(X-a)_s+\beta_3(X-b)_s$. Where $(u)_s=u$, if $u>0$; $=0$, if $u<=0$. One alternative would be cubic splines, which allow segments to curve: $F(X)=\beta_0+\beta_1X+\beta_2X^2+\beta_3X^3+\beta_4(X-a)^3_s+\beta_5(X-b)^3_s$.

For predictive modeling of non-linear associations, Harrell (2001) proposed cubic splines with knots chosen based on subject matter knowledge. If there is no knowledge as a basis for choosing the knots, Harrell suggested to use four knots, at the 5th, 35th, 65th and 95th percentiles. Other authors have also recommend cubic splines for this purpose (e.g., Hastie et al, 2009).

To use splines in modeling requires some coding in SAS, but it is not too difficult once one gets the hang of it. The following example using simulated data illustrates how this can be done. The data was simulated using the code below. Simulated y will be modeled as a function of simulated x.

```
data simulated;
do x=1 to 100;
_x=x+2*rannor(38209);
if x<=30 then do;
   y=5+_x*0.5;
end;
else do;
   y=16+_x*0.2-0.002*_x**2;
end;
output;
end;
run;
```

The resulting data is plotted in Figure 2. It appears that the slope of the relationship between x and y changes when x is about 25 or 30 (as should be the case, given how the data were generated) and then

again at x of about 60. Therefore, these would be good places to have spline knots that allow the slope to change.

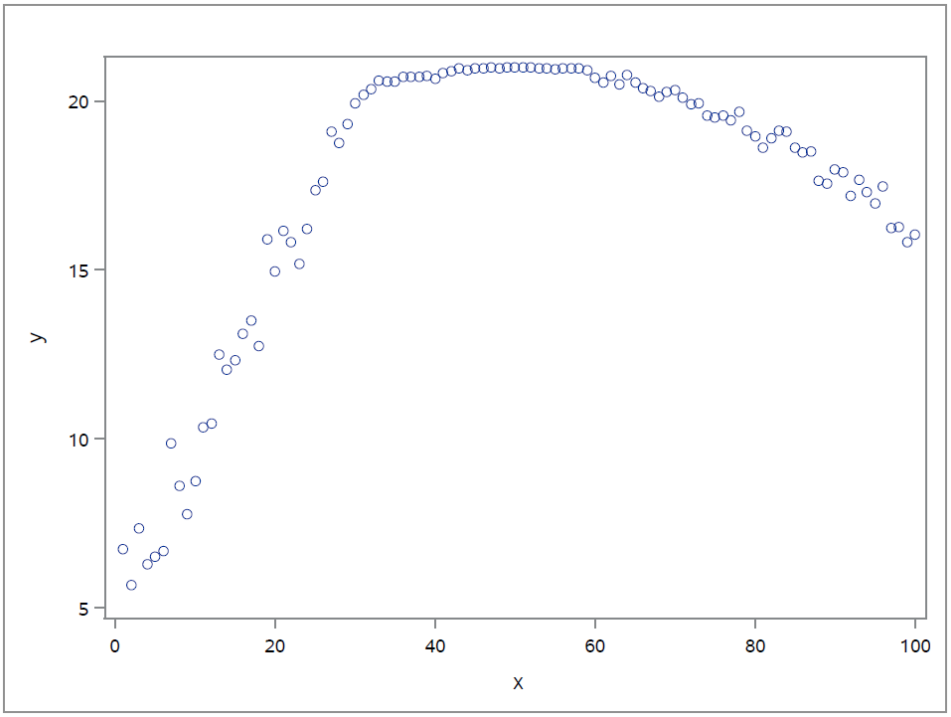

Figure 2. Plot of simulated x versus simulated y for spline illustration

To code the splines in SAS, segments can be defined as illustrated below. Because the relationship appears to have a rough inverse-U shape, a quadratic model was also created for comparison to the linear spline and cubic spline models.

```
data simulated2;
set simulated;
x2=x**2;
x3=x**3;
if x>25 then x4=(x-25); else x4=0;
if x>60 then x5=(x-60); else x5=0;

if x>25 then _x4=(x-25)**3; else _x4=0;
if x>60 then _x5=(x-60)**3; else _x5=0;
run;

* Quadratic;
proc genmod data=simulated2;
model y = x x2 / dist=normal link=identity;
ods output parameterestimates=parmb;
run;

* Linear splines;
proc genmod data=simulated2;
model y = x x4 x5 / dist=normal link=identity;
ods output parameterestimates=parmc;
run;
```

```
* Cubic splines;
proc genmod data=simulated2;
model y = x x2 x3 _x4 _x5 / dist=normal link=identity;
output pred=pred out=preds_spl;
ods output parameterestimates=parmd;
run;
```

The model results were plotted against the data, as shown in Figure 3.

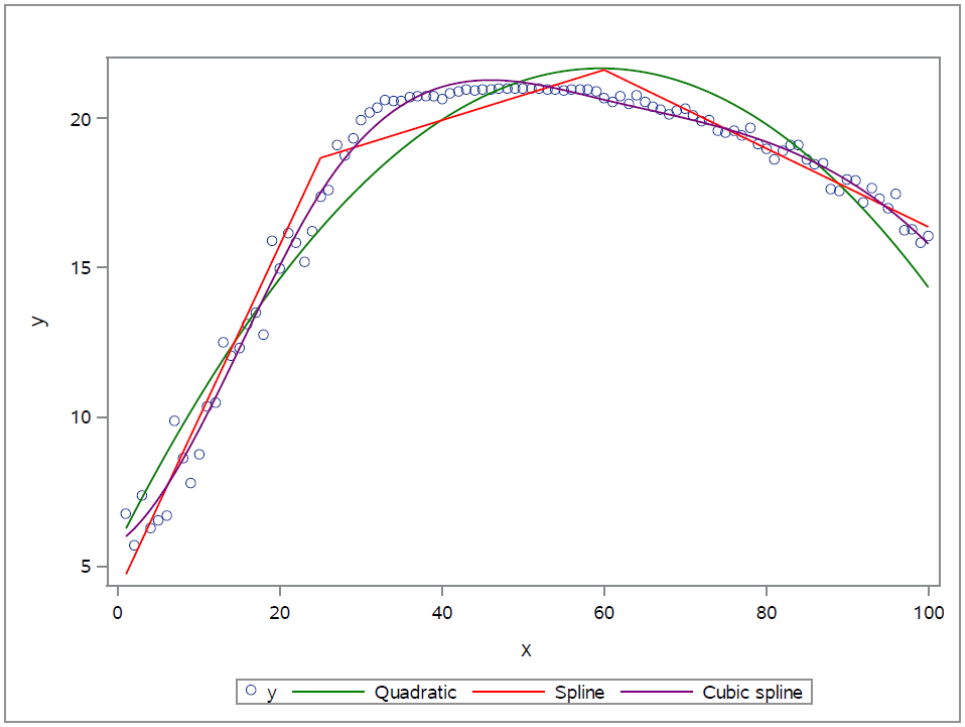

Figure 3. Results of 3 models (quadratic, linear spline, cubic spline) plotted against simulated data

The cubic spline model had the best fit with the data (R^2=0.986), followed by linear splines (R^2=0.968) and finally the quadratic model (R^2=0.936). Of course, the data were simulated in a way that would make this result likely. A more useful test would be to see how these functional forms work in more realistic predictive modeling situations, as illustrated later in this paper.

The second method also incorporates splines, but does not require a pre-specified set of knots or segments, thus it has more flexibility. Although multivariate adaptive regression splines (MARS) has not been proposed for exploring transformations of individual predictors to be incorporated in a subsequent "final" model (to the author's knowledge), it has attractive properties for this task. The method can be implemented using PROC ADAPTIVEREG in SAS/STAT. It involves piecewise linear regression, with splines allowing the slope of association to change in different ranges of the predictor variable. The splines are linear, not cubic. The technique involves searching through all possible knot values, in order to find "appropriate" ones. In practice, it uses a fast algorithm to do this efficiently (see SAS/STAT manual for details). The technique goes through a series of "growing" and "pruning" steps in model selection. Forward steps include selecting spline basis functions associated with greatest reduction in lack-of-fit (LOF) based on residual sum of squares, until a stopping criterion is reached, e.g., LOF change is minimal, or a maximum number of basis functions is selected. (A basis function defines the spline output relative to a specific knot, for example, "if x>25 then x4=(x-25), else x4=0" is a basis function relative to the knot of 25.) Backward steps, which use the criterion of a generalized cross-validation function (GCV), create reduced models, eliminating basis functions and then selecting the model with minimal GCV. The backward steps can result in an intercept-only model, so it may sometimes be necessary to use only the forward steps and then test the model fit on an independent validation dataset; whether or not this is

necessary will be dataset-specific.

The third method was proposed by Rud (2001). For convenience and due to lack of a better term, this will be called “Rud’s method” in this paper. The method involves exploring various functional forms (e.g., inverse, squared, cubed, trigonometric) – the “methodology is limited only by your imagination” (Rud, 2001, p. 89) – then select the best two by stepwise techniques. To implement the method, a set of transformed variables (e.g., square root, cubed, squared, inverse, trigonometric) is created from the original continuous variable. The original and all transformed variables are then fed into stepwise regression (e.g., PROC LOGISTIC), using maxstep=2 to get the two most predictive forms. An advantage of Rud’s method is its simplicity and ease of implementation, so if it is found to work reasonably well, it is worth considering for use in predictive modeling.

SAS MACRO

A SAS macro called %compare_transf (“compare transformations”) was created to facilitate implementation and comparison of the three methods. Full code of the macro is available from the author upon request.

The input dataset is named inscope2 and the target variable is totexpy2_gt20k; these remain constant throughout the macro (the user can modify these as needed for different analyses). A predictor variable, &var, is passed to the macro and run through each of the three methods.

The first method is linear splines. Per Harrell’s recommended defaults, the 5th, 35th, 65th and 95th percentiles are found via PROC UNIVARIATE and then used to define the knots:

```
data test_spline;
set inscope2;
x=&var;
if x>&p_5 then x2=(x-&p_5); else x2=0;
if x>&p_35 then x3=(x-&p_35); else x3=0;
if x>&p_65 then x4=(x-&p_65); else x4=0;
if x>&p_95 then x5=(x-&p_95); else x5=0;
run;
```

A logistic regression model is then created with the linear spline representation of &var and no other predictors, as shown below.

```
proc logistic data=test_spline(where=(training=1)) namelen=100;
model totexpy2_gt20k(event='1') = x x2 x3 x4 x5;
output predprobs=individual out=preds_spl;
ods output parameterestimates=parm1 association=assoc1;
run;
```

This is compared with a logistic regression in which a simple linear slope is estimated for &var. The area under the ROC curve (AUC, also known as the c-statistic) is used as the metric of model fit. In addition to creating this linear spline model and estimating fit, the macro creates scoring code for the linear spline model which can be executed in a SAS data step.

The second method uses PROC ADAPTIVEREG, which is similar to MARS (thus called “MARS” in this paper), to find an optimal transformation of &var. Unlike the linear spline method, this does not require the number of segments or knot values to be pre-specified. The model equation resulting from PROC ADAPTIVEREG is used to create a score which is the sole predictor in a subsequent logistic regression model (PROC LOGISTIC). The c-statistic from this model is compared to that from the model in which a simple linear slope was estimated for &var. Depending on the situation, it might be necessary to limit the technique to the forward steps only and limit the number of basis functions (for example, when an intercept-only model results from PROC ADAPTIVEREG and one is not satisfied with that result). As shown below, lines of code that can be used to do this are left in the macro, but commented out; these lines can be un-commented and run as the situation requires.

```
proc adaptivereg data=inscope2(where=(training=1)) plots=all
details=bases;
* model totexpy2_gt20k(event='1') = &var / forwardonly maxbasis=6 additive
dist=binomial;
model totexpy2_gt20k(event='1') = &var / additive dist=binomial;
* ods output bases=b FWDParams=p;
ods output bases=b BWDParams=p;
output out=preds_mars predicted=pred;
run;
```

The final method executed in the macro is Rud's method. About 20 functional forms are defined in the macro, as shown below. More functional forms could be considered ("limited only by your imagination" per Rud), but the macro uses the basic set described by Rud (2001).

```
data transform;
set inscope2;
rawvar=&var;

var_sq=rawvar**2;
var_cu=rawvar**3;
var_sqrt=sqrt(rawvar);
var_curt=rawvar**0.33;
var_log=log(max(0.0001,rawvar));
var_exp=exp(max(0.0001,min(700,rawvar)));
var_tan=tan(rawvar);
var_sin=sin(rawvar);
var_cos=cos(rawvar);
var_inv=1/max(0.0001,rawvar);
var_sqi=1/max(0.0001,rawvar**2);
var_cui=1/max(0.0001,rawvar**3);
var_sqri=1/max(0.0001,sqrt(rawvar**3));
var_curi=1/max(0.0001,sqrt(rawvar**0.33));
var_logi=1/max(0.0001,log(max(0.0001,rawvar)));
var_expi=1/max(0.0001,exp(max(0.0001,min(700,rawvar))));
var_tani=1/max(0.0001,tan(rawvar));
var_sini=1/max(0.0001,sin(rawvar));
var_cosi=1/max(0.0001,cos(rawvar));
run;
```

PROC LOGISTIC with selection=stepwise and maxstep=2 is then used to find the two best-fitting functional forms. As with the other methods, scoring code is created and AUC of the model is output.

After all three methods are executed, a final step in the macro is to plot the resulting equations against the original data using PROC SGPLOT (code shown below). Each equation is plotted using a different color. The original data are grouped by taking the mean within a user-specified interval and limited to a user-specified range, to avoid having the data appear as a swarm of points and to avoid having the visual representation be overly affected by data points at the extreme ends of the range.

```
ods pdf file="C:\projects\MWSUG19\Optimal_transformations\&var..pdf";
proc sgplot data=examine_function4;
series x=i y=score / legendlabel='Splines' lineattrs=(color=blue);
series x=i y=score2 / legendlabel='MARS' lineattrs=(color=red);
series x=i y=score3 / legendlabel='Rud' lineattrs=(color=green);
scatter x=i y=mean_totexpy2_gt20k / legendlabel='Mean actual';
xaxis label="&xlabel";
yaxis label='Proportion >$20K spend Y2';
run;
```

```
ods pdf close;
```

Note that all of the above are assumed to be executed on the predictive model's training dataset. This is assumed to be part of the exploratory phase of modeling prior to constructing a final model and testing the fit on a holdout sample. Those steps, which are described below, are executed outside of the %compare_transf macro.

FINAL MODEL: APPROACH

The final model was constructed by going through the predictive model build process using all predictor variables, most of which were binary indicators. This process involved bivariate screening, variable clustering to weed out redundancies, and selection of predictive variable subsets using backward and stepwise selection, as well as selecting the top candidate predictors from a fixed model. After executing these steps, transformations of continuous variables were considered by entering the scores resulting from all three methods described above, for the continuous variables still remaining under consideration at this step. As a final step in model build process, interactions (i.e., combinations of predictors) were considered. To evaluate the lift achieved from transformations of the continuous variables, an alternative model was built without considering such transformations, representing the result that might have been achieved without using the techniques described above (essentially, considering only simple linear slopes for continuous variables). After the final predictive model was built on the training sample via this process, fit was examined in an independent holdout sample. MEPS survey weights were not used at any point in this process.

RESULTS

Persistent problems were encountered in estimating quadratic and cubic spline models with knot values at the selected percentiles (e.g., convergence problems, resulting models that yielded very implausible predictions), therefore linear splines were used as they seemed to be less vulnerable to these problems. Linear spline models converged to reasonable estimates with no difficulty, unlike the quadratic and cubic spline models. This is possibly in part because data at the extremes (>95th percentile, <5th percentile) were influential and complex, which may be more problematic for quadratic and cubic spline models than for linear spline models. Restricted cubic spline models might provide a solution, but that was left for future research.

Table 1 below shows the results produced by the %compare_transf macro, examining transformations of continuous predictors (&var) individually in relation to the predictive modeling target. No single type of transformation was best across all of the predictor variables. MARS produced the best fit for two variables, linear splines for one variable, and Rud's method for one variable. For 3 of the 4 variables (all except family size), the simple linear model achieved the poorest fit, underscoring the importance of examining non-linear transformations of continuous predictors in predictive models.

Transformation	Predictors (continuous variables measured in 2014)			
	Expenditures	**Income**	**Age**	**Family size**
None (linear)	0.7962	0.5257	0.7068	0.6176
Linear splines	0.8102	0.5447	**0.7089**	0.6174
MARS	**0.8219**	**0.6065**	0.7055	0.6296
Rud's method	0.8095	0.5521	0.7077	**0.6297**

Table 1. AUC by transformation for each predictor variable; the best-fitting transformation for each variable is shown in bold font.

For 2014 healthcare expenditures, the MARS method achieved the best fit. As shown in Figure 4, all of the models depicted an increasing slope between $0 and about $25,000 in 2014 healthcare expenditures, but the MARS model was unique in depicting a relatively flat slope beyond $50,000 (which may work well because the pattern beyond that point looks noisy and it is probably difficult to outperform a flat-line

slope).

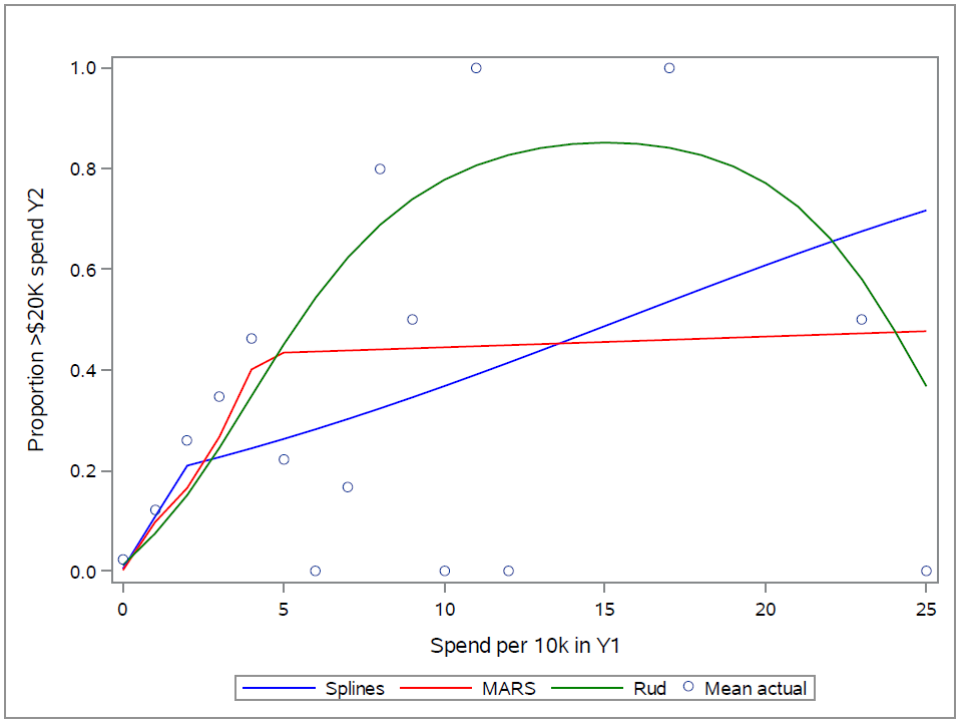

Figure 4. Model predictions (lines) vs. average actuals (circles) for healthcare expenditures in 2014.

MARS also produced the best fit for 2014 income. However, none of the transformations worked very well. The MARS model depicted a jagged pattern up to about $120,000 income, with a slightly upward curving slope afterward (Figure 5).

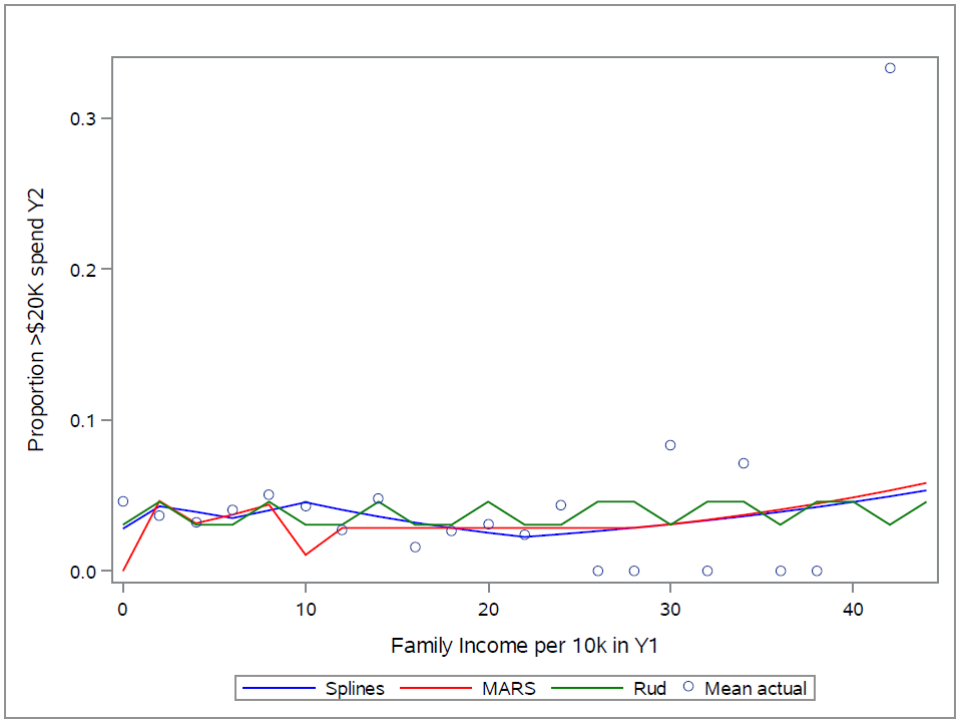

Figure 5. Model predictions (lines) vs. average actuals (circles) for income in 2014.

Linear splines worked best for age. As shown in Figure 6, linear splines were the only transformation that captured decreasing risk of high spend in 2015 after the late 50s.

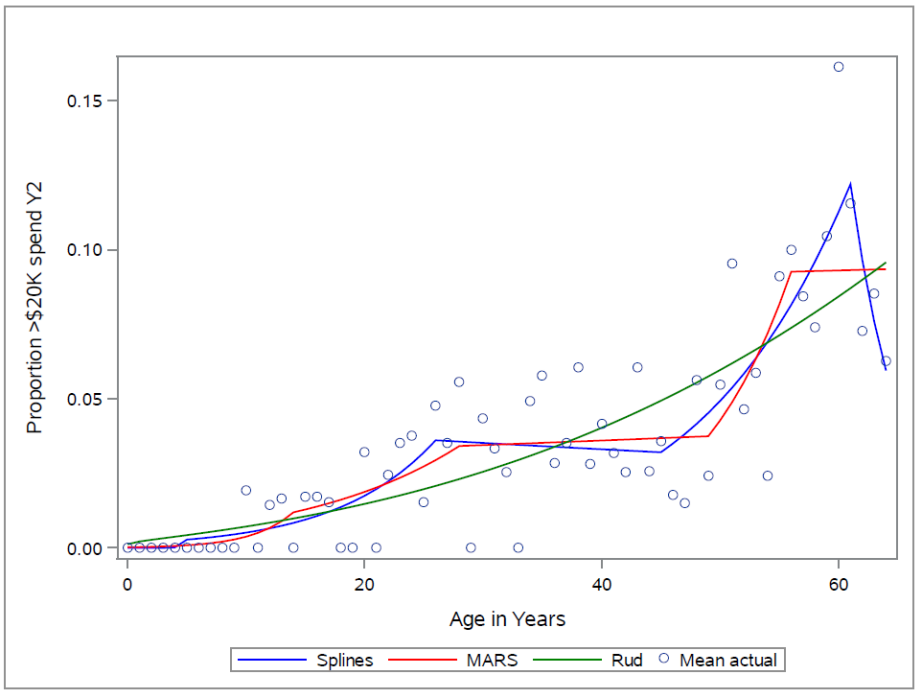

Figure 6. Model predictions (lines) vs. average actuals (circles) for age in 2014.

Finally, Rud's method achieved the best fit for family size. Compared to MARS (a close runner-up), which yielded fairly similar predictions, the Rud transformation predicted slightly lower risk for small families and slightly higher risk for larger families (Figure 7).

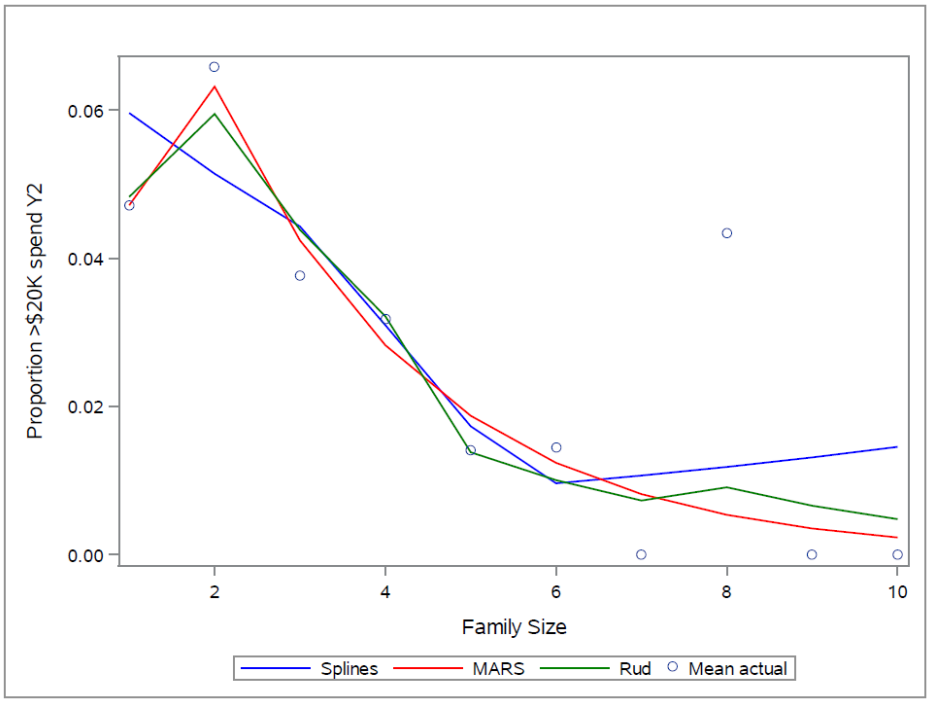

Figure 7. Model predictions (lines) vs. average actuals (circles) for income in 2014.

Income and family size dropped out of the candidate predictor set prior to fitting the final model. This is perhaps not surprising, given the relatively low AUCs for these variables in relation to the predictive model target (see Table 1). In the final model, the linear spline representation was selected for age (age_score) and the MARS transformation was selected for 2014 healthcare expenditures (totexp_score2).

The age_score linear spline transformation used in the final model was defined as follows, where x=age in years:

```
if x>5 then x2=(x-5); else x2=0;
if x>26 then x3=(x-26); else x3=0;
if x>45 then x4=(x-45); else x4=0;
if x>61 then x5=(x-61); else x5=0;
yhat=(-19.31460159 + x*2.68518169 + x2*-2.56121991 + x3*-0.13041829 +
x4*0.09605896 + x5*0.35198385);
if -699<=yhat<=699 then age_score=1/(1+exp(-1*yhat));
else if -699<=yhat then age_score=0;
else if 699>=yhat then age_score=1;
```

The totexp_score2 transformation used in the final model was defined as follows:

```
Basis0=1;
Yhat = 0;
yhat + 1394.37361508725 * 1;
yhat + 331.737288425125 * Basis0*MAX(totexpy1_per10k -  4.2148,0);
yhat + 653.213587486712 * Basis0*MAX(totexpy1_per10k -  0.1589,0);
yhat + -337.79257086367 * Basis0*MAX(totexpy1_per10k -  0.0061,0);
yhat + 26.5188105474582 * Basis0*MAX(totexpy1_per10k -  0.0817,0);
yhat + -1414.2380918162 * Basis0*MAX(totexpy1_per10k -  0.1387,0);
yhat + 1094.14243929054 * Basis0*MAX(totexpy1_per10k -  0.1367,0);
yhat + -332.33737055093 * Basis0*MAX(4.2148 - totexpy1_per10k,0);
yhat + -3.0284779401348 * Basis0*MAX(totexpy1_per10k -  0.5905,0);
yhat + -4819.7355515228 * Basis0*MAX(totexpy1_per10k -  0.1551,0);
yhat + 13.3168236867483 * Basis0*MAX(totexpy1_per10k -  0.2384,0);
yhat + 4455.87432982679 * Basis0*MAX(totexpy1_per10k -  0.1538,0);
if -699<=yhat<=699 then totexp_score2=1/(1+exp(-1*yhat));
else if -699<=yhat then totexp_score2=0;
else if 699>=yhat then totexp_score2=1;
```

The final model included the spline representation of age (age_score), the MARS representation of 2014 healthcare expenditures (totexp_score2), and self-assessed health status in 2014 (_RTHLTH2). The model equation was:

```
fnl_logit=-6.01751090+
totexp_score2*7.65772905+
age_score*15.38177763+
_RTHLTH2*0.62209167;
fnl_score=1/(1+exp(-1*fnl_logit));
```

The alternative model built without considering the optimal transformations on the training data, estimating simple linear slopes for continuous predictors was:

```
alt_logit=-7.12389587+
TOTEXPY1*0.00002578+
_AGE2X*0.03471384+
_RTHLTH2*0.72528551+
born_in_US*0.61367493;
alt_score=1/(1+exp(-1*alt_logit));
```

On the holdout sample, the c-statistic of the model including non-linear transformations (fnl_score) was 0.812, a noticeable improvement over 0.793 for the model with no transformations (alt_score). Improved performance appeared to come largely from the middle score deciles, as shown in Figure 8.

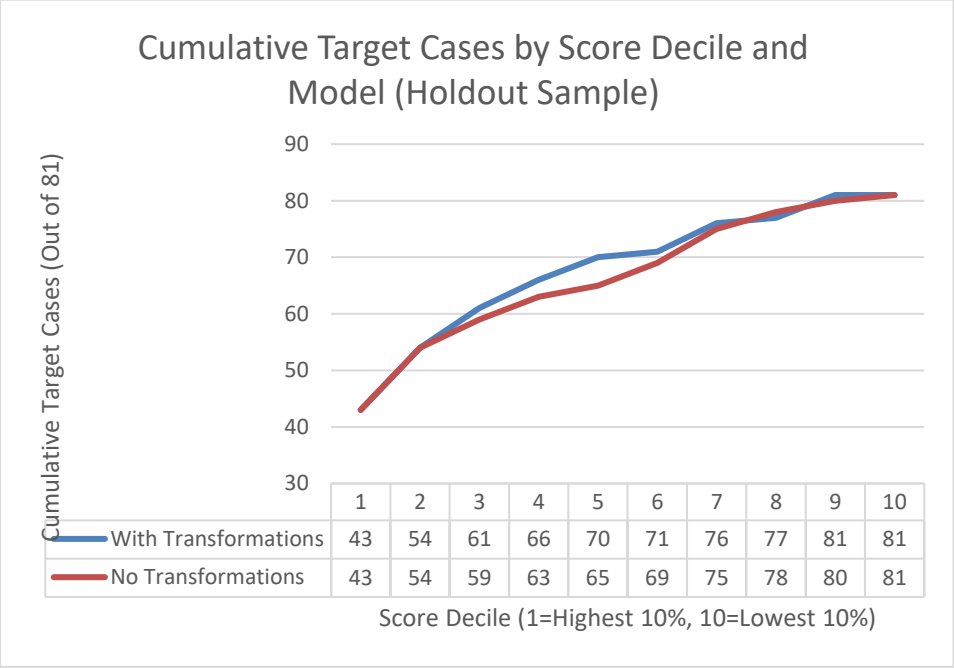

Figure 8. Cumulative target cases by score decile for the models with vs. without transformations of continuous variables (holdout sample).

CONCLUSION

Three methods for exploring optimal transformations of continuous variables in predictive modeling were described. A SAS macro for implementing the methods was described and illustrated. In an example predictive model (predicting the likelihood of high healthcare expenditures in 2015 as a function of information known about individuals in 2014), a linear representation of continuous variables typically provided the poorest fit. MARS provided the best fit for two variables, linear splines was best for one variable, and Rud's method was best for one. These results are not necessarily representative of all predictive modeling situations, but in the author's experience, it would not be surprising to find complex functions like those fit with MARS in many healthcare predictive models. Thus the MARS approach might be best if it were necessary to choose only one, but the other approaches seemed to work reasonably well also.

It is a little surprising that no method was clearly superior to the others. For example, given the flexibility of MARS, one might guess that it would consistently perform the best. However, this was not the case. Each of the three methods examined yielded the best-fitting solution for at least one variable in the illustrative predictive model.

REFERENCES

Baesens B, Vlasselaer VV, Verbeke W. 2015. Fraud analytics, using descriptive, predictive, and social network techniques: A guide to data science for fraud detection. Hoboken, NJ: Wiley.

Harrell FE. 2001. Regression modeling strategies. New York: Springer.

Hastie T, Tibshirani R, Friedman J. 2009. The elements of statistical learning: Data mining, inference and prediction. New York: Springer.

Kuhn M, Johnson K. 2016. Applied predictive modeling. New York: Springer.

Rud OP. 2001. Data mining cookbook: Modeling data for marketing, risk, and customer relationship management. New York: Wiley.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Doug Thompson, PhD
Director of Advanced Analytics
Rush Health
1645 W. Jackson Blvd
Chicago, IL 60612
Doug_Thompson@rush.edu