An Investigation into the Use of Generalized Additive Neural Networks in Credit Scoring

DA de Waal
Centre for Business Mathematics and Informatics
North-West University, Potchefstroom Campus
Private Bag X6001, Potchefstroom, 2520, South Africa
+27 18 2992535
bwidadw@puk.ac.za

JV du Toit
School of Computer, Statistical and Mathematical Sciences
North-West University, Potchefstroom Campus
Private Bag X6001, Potchefstroom, 2520, South Africa
+27 18 2992548
rkwjvdt@puk.ac.za

T de la Rey
Centre for Business Mathematics and Informatics
North-West University, Potchefstroom Campus
Private Bag X6001, Potchefstroom, 2520, South Africa
+27 18 2992566
bwitdlr@puk.ac.za

ABSTRACT

Logistic regression occupies a central position in the field of credit scoring as it is relatively well understood and an explicit formula can be derived on which credit decisions may be based. Although artificial neural networks may be more powerful than logistic regression, it is not widely used in credit scoring because it is a black box with respect to interpretation and the absence of reasons why the neural network has reached its decisions may be unacceptable. In contrast to logistic regression and neural networks, generalized additive models is a compromise between inflexible, but docile linear models and flexible, but troublesome, universal approximators. In this study the performance of a generalized additive model (implemented as a neural network and therefore called a generalized additive neural network or GANN) is compared to that of a logistic regression model on a home equity data set, where the aim is to predict whether an applicant will eventually default or be seriously delinquent on a loan. Partial residuals are used to investigate the effect of the individual inputs, adjusted for the effect of the other inputs, where the *j*th partial residual is the deviation between the actual values and that portion of the fitted model that does not involve variable x_j. Using a GANN architecture therefore assists in alleviating the black box perception of neural networks with respect to interpretation as the effect of each input variable on the fitted model can be interpreted using a graphical method (partial residual plot). Another benefit of using a GANN is that the nonlinear effects of the inputs may be easier to learn in this constrained architecture than in the general artificial neural network or multiplayer perceptron setting.

Categories and Subject Descriptors

I [**Computing Methodologies**]: I.2 Artificial Intelligence I.2.6 Learning [Subjects: Connectionism and neural nets]; H [**Information Systems**]: H.2 Database Management H.2.8 Database applications [Subjects: Data mining]; J [**Computer Applications**]: J.1 Administrative Data Processing [Subjects: Financial]

General Terms

Algorithms, Performance, Design, Economics

Keywords

Additive models, neural networks, logistic regression, partial residuals, predictive modeling, credit scoring

1. INTRODUCTION

Logistic regression (Kleinbaum, 1994); (Hosmer & Lemeshow, 1989) occupies a central position in the field of credit scoring (Thomas, Edelman & Crook, 2002); (McNab & Wynn, 2000) as it is relatively well understood and an explicit formula can be derived on which credit decisions may be based. It is furthermore widely used in industry and has become the standard used by most companies.

Although artificial neural networks may be more powerful than logistic regression, it is not widely used in credit scoring because it is a black box with respect to interpretation and the absence of reasons why the neural network has reached its decisions may be unacceptable. Getting regulatory approval for the use of neural networks to make credit decisions may also be an important issue hindering the acceptance and wide use of neural networks in this environment.

In contrast to logistic regression and neural networks, generalized additive models is a compromise between inflexible, but docile linear models and flexible, but troublesome, universal approximators. In this study the performance of a generalized additive model (implemented as a neural network and therefore called a generalized additive neural network or GANN) is compared to that of a logistic regression model on a home equity data set, where the aim is to predict whether an applicant will eventually default or be seriously delinquent on a loan. A scorecard is then built using this technique and compared to a scorecard built using only logistic regression.

Partial residuals are used to investigate the effect of the individual inputs, adjusted for the effect of the other inputs, where the *j*th partial residual is the deviation between the actual values and that portion of the fitted model that does not involve variable x_j.

Using a GANN architecture therefore assists in alleviating the black box perception of neural networks with respect to interpretation as the effect of each input variable on the fitted model can be interpreted using a graphical method (partial residual plot). Another benefit of using a GANN is that the nonlinear effects of the inputs may be easier to learn in this constrained architecture than in the general artificial neural network or multilayer perceptron setting.

The rest of the paper is organized as follows. Generalized additive models are explained in Section 2. The home equity loan example is described in Section 3. A logistic regression model and a generalized additive model are built and compared. In Section 4 a scorecard is built using a generalized additive model. The last section contains conclusions and ideas for future work.

2. GENERALIZED ADDITIVE NEURAL NETWORKS

The generic supervised prediction problem consists of a data set of $i = 1, 2, ..., n$ cases (observations, examples or instances). Associated with each case is a vector of input variables (predictors, features), $x_1, x_2, ..., x_k$, and a target variable (response, outcome), y. A predictive model maps the inputs to the expected value of the target. The predictive model is built on a training data set where the target is known. The purpose is to apply the model to new data where the target is unknown.

Some examples of predictive models are presented next so that the GANN can be classified among other similar models.

2.1 EXAMPLES OF MODELS USED FOR PREDICTION

The first example is a linear model given by

$$E(y_i) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} \qquad (1)$$

In (1), the expected value of the target, $E(y_i)$, is expressed as a linear combination of the inputs. A slightly more complex model is the generalized linear model of the form

$$g_0^{-1}(E(y_i)) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} \qquad (2)$$

A link function, g_0^{-1}, is used in (2) to constrain the range of the predicted values and is the inverse of the activation function g_0. When the expected target is bounded between 0 and 1, such as probability, the logit link function given by

$$g_0^{-1}(E(y_i)) = \ln\left(\frac{E(y)}{1-E(y)}\right)$$

is appropriate. For an expected target that is bounded between *-1* and *1*, the hyperbolic tangent link function given by

$$g_0^{-1}(E(y_i)) = 1 - \frac{2}{1+\ln(2(E(y))}$$

can be used.

Multilayer perceptrons are the most widely used type of neural network for supervised prediction. Theoretically, they are universal approximators that can model any continuous function (Ripley, 1996). A multilayer perceptron (MLP) that has a single layer with h hidden neurons has the form

$$\begin{aligned} g_0^{-1}(E(y_i)) &= w_0 + w_1 \tanh(w_{01} + \sum_{j=1}^{k} w_{j1} x_{ji}) + \\ &\cdots + w_h \tanh(w_{0h} + \sum_{j=1}^{k} w_{jh} x_{ji}) \end{aligned} \qquad (3)$$

The link-transformed expected value of the target is expressed as a linear combination of nonlinear functions of linear combinations of the inputs (3). The activation function used for the hidden layers in this case is the hyperbolic tangent function. This nonlinear regression model has $h(k + 1)+1$ unknown parameters, where h is the number of hidden nodes and k the number of inputs. The parameters are estimated by numerically optimizing some suitable measure of fit to the training data such as the negative log likelihood.

2.2 GENERALIZED ADDITIVE MODELS

Generalized additive models (Hastie & Tibshirani, 1990) are a compromise between the inflexible, but docile, linear models and the flexible, but troublesome, universal approximators. A generalized additive model (GAM) has the form

$$E(y_i) = \beta_0 + f_1(x_{1i}) + f_2(x_{2i}) + \cdots + f_k(x_{ki})$$

where the expected target on the link scale is expressed as the sum of individual unspecified univariate functions. Each univariate function can be interpreted as the effect of the corresponding input while holding the other inputs constant. GAMs allow more flexibility than linear models and are easy to interpret graphically.

GAMs are usually presented as extensions of linear models, but can also be presented as constrained forms of flexible universal approximators such as pursuit regression (Friedman & Stuetzle, 1981) and artificial neural networks (Sarle, 1994). In the latter case they are called GANNs. An example of a GANN with two inputs is shown in Figure 1. Each input has a direct connection (skip layer), the first input has two nodes in the hidden layer and the second input has three nodes in the hidden layer.

The basic architecture for a GANN has a separate MLP with a single hidden layer of h units for each input variable, given by

$$f_j(x_{ji}) = w_{1j}\tanh(w_{01j} + w_{11}x_{ji}) + \cdots + w_{hj}\tanh(w_{0hj} + w_{1hj}x_{ji})$$

The overall bias β_0 absorbs the individual bias terms. Each individual univariate function has *3h* parameters, where *h* could vary across inputs. The architecture can be enhanced to include an additional parameter for a skip layer

$$f_j(x_{ji}) = w_{oj}x_{ji} + w_{1j}\tanh(w_{01j} + w_{11}x_{ji}) + \cdots + w_{hj}\tanh(w_{0hj} + w_{1hj}x_{ji})$$

so that the generalized linear model is a special case.

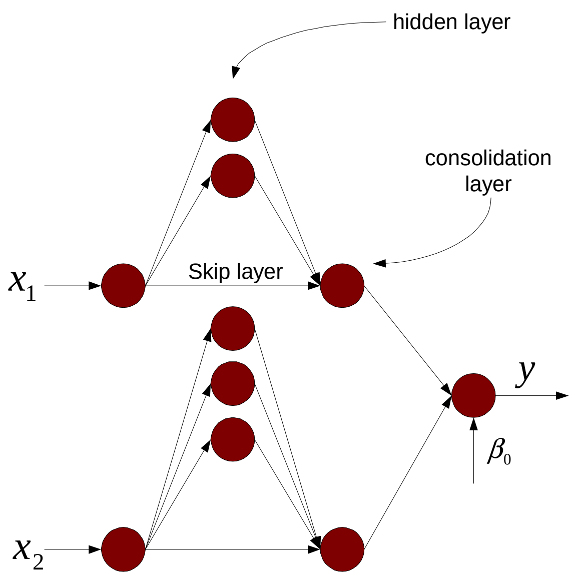

Figure 1: Example GANN

Any method suitable for fitting more general MLPs can be used to simultaneously estimate the parameters of GANN models. As a result, the usual optimization and model complexity issues also apply to GANN models.

The following set of instructions (Potts, 1999); (Potts, 2000) for the construction of a GANN interactively takes advantage of their constrained form to simplify optimization and model selection.

1. Construct a GANN with one neuron in the hidden layer and a skip layer for each input $f_j(x_{ji}) = w_{oj}x_{ji} + w_{1j}\tanh(w_{01j} + w_{11}x_{ji})$. This gives four parameters for each input. Binary inputs only have a direct connection.
2. Fit a generalized linear model to give initial estimates of β_0 and the w_{0j}.
3. Initialize the remaining three parameters in each hidden layer as random values from a normal distribution with mean zero and variance equal to 0.1.
4. Fit the full GANN model.
5. Examine each of the fitted univariate functions overlaid on their partial residuals.
6. Prune (remove neurons) the hidden layers with apparently linear effects and grow (add neurons) the hidden layers where the nonlinear trend appears to be underfitted. If this step is repeated, the final estimates from previous fits can be used as starting values.

For more than half a century, a variety of diagnostic plots have been used to assess nonlinear relationships between the target and input variables in multiple regression models. In general, there are two complementary approaches to examining the assumption of linearity: informal graphical methods (Cai & Tsai, 1999) and formal tests. Ezekiel (1924) introduced an informal graphical method that was termed the partial residual plot by Larsen & McCleary (1972). This method is still used frequently.

The visual diagnostics used to aid the model selection process in this study are plots of the fitted univariate functions, $\hat{f}(x_{ji})$, overlaid on the partial residuals

$$pr_{ji} = g_0^{-1}(y_i) - \beta_0 - \sum_{l \neq j} \hat{f}_l(x_{li}) = (g_0^{-1}(y_i) - g_0^{-1}(\hat{y}_i)) + \hat{f}_j(x_{ji})$$

versus the corresponding *j*th input.

With partial residuals the effect of the individual inputs, adjusted for the effect of the other inputs, can be investigated. The *j*th partial residual is the deviation between the actual values and that portion of the fitted model that does not involve x_j. When g_0^{-1} is nonlinear, a first order approximation is usually used

$$pr_{ji} = \frac{\partial g_0^{-1}(\hat{y}_i)}{\partial y}(y_i - \hat{y}_i) + \hat{f}_j(x_{ji})$$

The growing and pruning process is started with a single neuron and a skip layer instead of the linear model. The linear fit is only used for initialization. The effectiveness of partial residual plots for visualizing the underlying curve is discussed by Berk & Booth (1995). They demonstrated that the partial residuals based on a GAM fit are more reliable than those based on a linear fit. Also, starting with four parameters is common practice with GAM estimation.

3. HOME EQUITY EXAMPLE

In this section, two models are built on a data set containing baseline and loan performance information for 5,960 recent home equity loans (Wielenga, Lucas & Georges, 1999). The binary target variable (BAD) indicates whether an applicant eventually defaulted or was seriously delinquent and occurred in 1,189 cases (approx 20%). The are 12 input variables: REASON (home improvement or debt consolidation), JOB (six occupational categories), LOAN (loan amount requested), MORTDUE (amount due to existing mortgage), VALUE (value of current property), DEBTINC (debt to income ratio), YOJ (years at present job), DEROG (number of derogatory reports), CLNO (number of trade lines), DELINQ (number of delinquent trade lines), CLAGE (age of oldest trade line in months) and NINQ (number of recent credit enquiries). The data were collected from recent applicants granted credit.

The first model is built using standard modeling practices using logistic regression. Over sampling might be considered for this example as the number of BADs is substantially less than the number of GOODs. This is not done as our main motivation for writing this paper is to compare two modeling techniques on a high level and not to tweak an existing model.

The data set is first split into training (67%) and validation (33%) data sets. As there are a high percentage of missing values for DEBTINC (20%), some method is needed to handle the missing values in the data set: the standard mean value imputation method for interval targets is used in this example. Other variables with missing values are handled in a similarly way. The usual variable transformation step (to handle nonlinear associations between inputs and the target) is deliberately omitted to illustrate the main difference between a logistic regression model and a generalized additive model.

The second model is built using the same modeling practice just described, except for the replacement of the logistic regression model by a generalized additive model.

3.1 LOGICTIC REGRESSION

Stepwise logistic regression is performed on the training data set with missing values imputed. Standard significant levels of 0.05 are used. Two variables are deleted from the model, REASON and YOJ, and JOB is transformed into 5 variables using N-1 dummy coding. The events classification information given in Table 1 gives an indication of how well the model discriminates between bad and good applicants.

Data role	Target	False positive	False negative	True positive	True negative
Train	BAD	95	549	247	3101
Validate	BAD	43	282	111	1532

Table 1: Event Classification

The model has 15 degrees of freedom (14 variables plus an intercept).

3.2 GENERALIZED ADDITIVE MODEL

A generalized additive model is built using the developed AutoGANN node (Du Toit & De Waal, 2005) in SAS® Enterprise Miner. For illustrative purposes we give a specific model that is not too complex to highlight some important issues that we will discuss in the next sections. The resulting model is summarized in Table 2.

	Variable	Sub-architecture	Interpretation
1	LOAN	1	*Linear*
2	IMP_JOB	1	*Linear*
3	IMP_REASON	0	*Deleted*
4	IMP_CLAGE	1	*Linear*
5	IMP_CLNO	1	*Linear*
6	IMP_DEBTINC	4	*Nonlinear*
7	IMP_DELINQ	1	*Linear*
8	IMP_DEROG	2	*Nonlinear*
9	IMP_MORTDUE	1	*Linear*
10	IMP_NINQ	1	*Linear*
11	IMP_VALUE	1	*Linear*
12	IMP_YOJ	0	*Deleted*

Table 2: Generalized Additive Model Results

The sub-architecture gives an indication of the complexity of the neural network used to approximate the univariate function (Du Toit & De Waal, 2003). *Deleted* indicates that the variable is deleted from the model, *Linear* indicates that the relationship with the target is linear and *Nonlinear* indicates that the relationship with the target is nonlinear. The events classification information for this model is given in Table 3.

Data role	Target	False positive	False negative	True positive	True negative
Train	BAD	165	292	504	3101
Validate	BAD	80	140	253	1495

Table 3: Event Classification

The resulting model has 23 degrees of freedom. Figures 2 to 10 contain the partial residual plots for the variables in the model. Note that IMP_REASON and IMP_YOJ are deleted from the model. Their partial residual plots are not given.

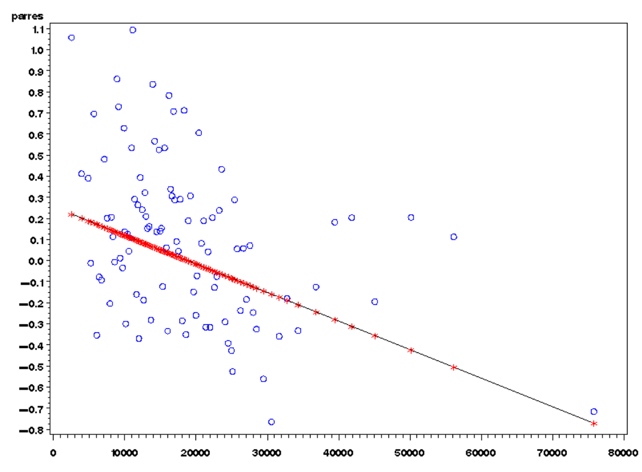

Figure 2: Partial residual Plot for LOAN

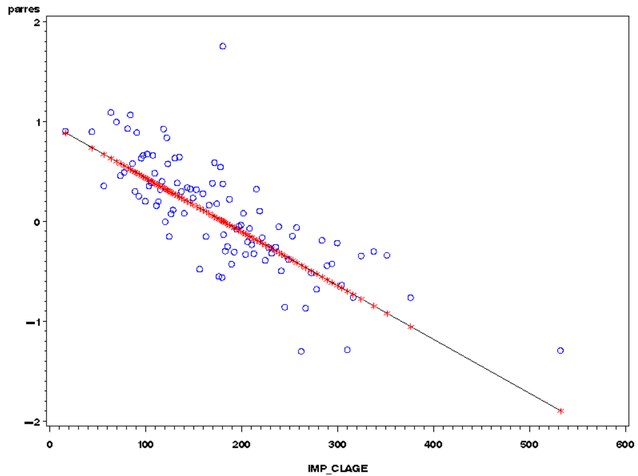

Figure 3: Partial Residual Plot for IMP_CLAGE

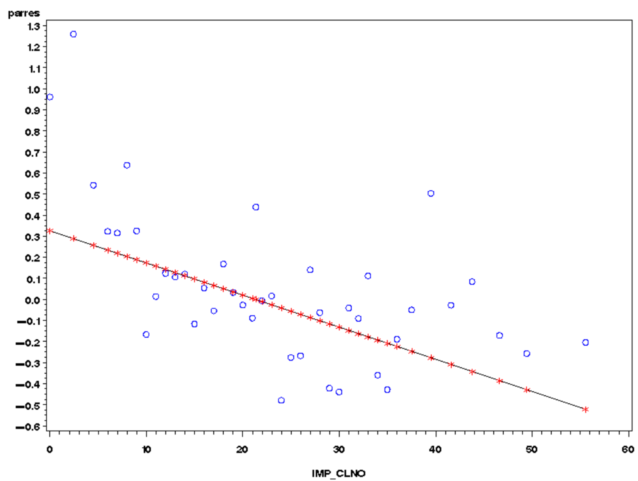

Figure 4: Partial Residual Plot for IMP_CLNO

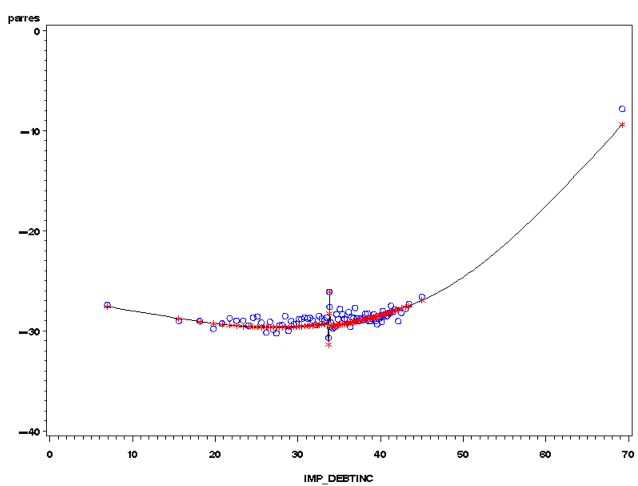

Figure 5: Partial Residual Plot for IMP_DEBTINC

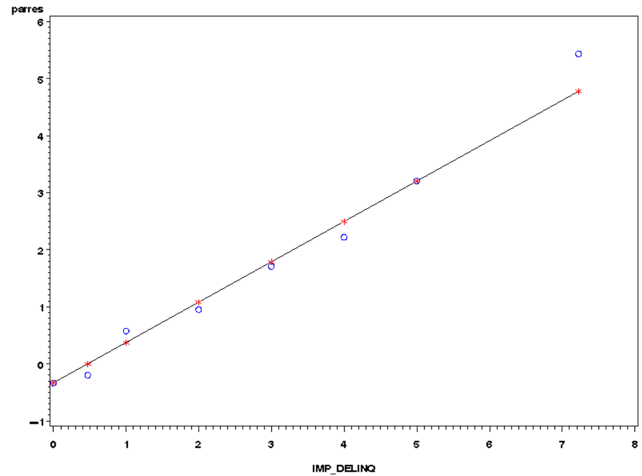

Figure 6: Partial Residual Plot for IMP_DELINQ

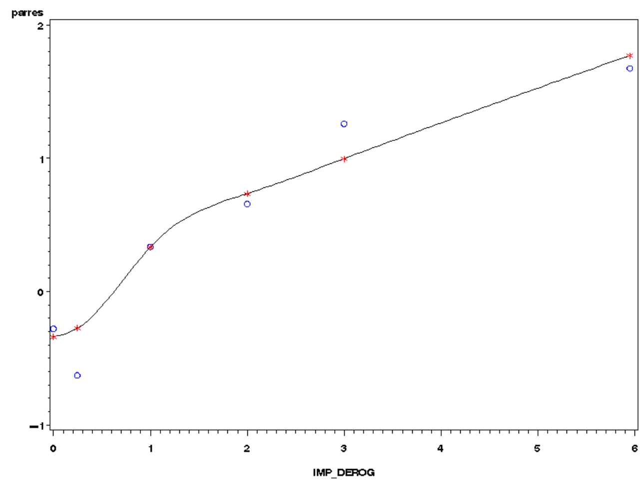

Figure 7: Partial Residual Plot for IMP_DEROG

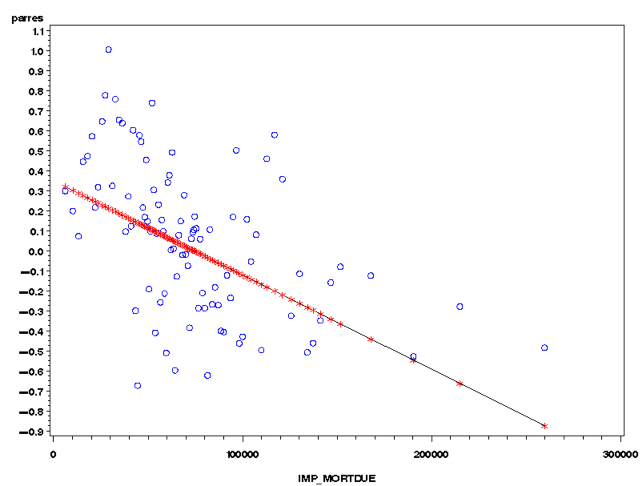

Figure 8: Partial Residual Pot for IMP_MORTDUE

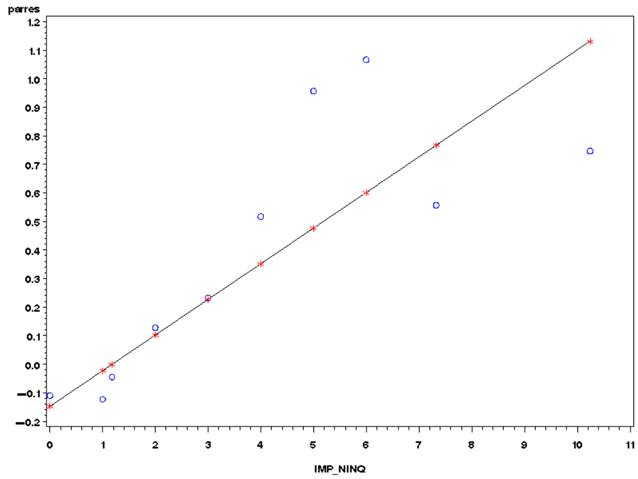

Figure 9: Partial Residual Plot for IMP_NINQ

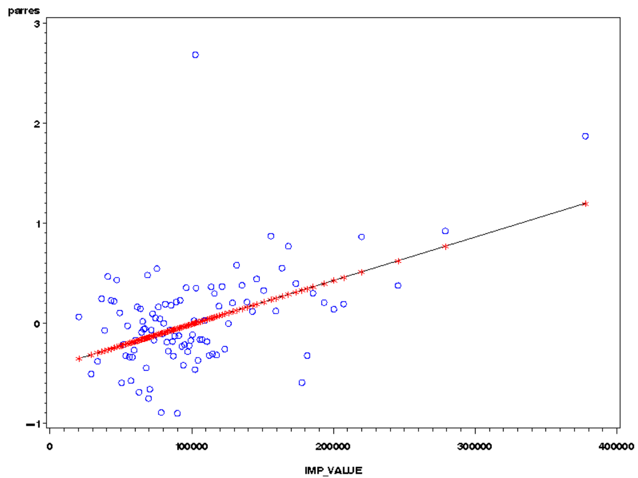

Figure 10: Partial Residual Plot for IMP_VALUE

It is clear from Figures 5 and 7 that IMP_DEBTINC and IMP_DEROG exhibit nonlinear relationships with the target. An important question that needs answering is whether the complexity of the nonlinear effects exhibited by the two variables is satisfactory given our domain knowledge of the given problem.

The following insights can be gained from closer inspection of the partial residuals and fitted curve in Figure 5:

- It seems that there exist at least three extreme points that dramatically influence the complexity of the curve: these points need further investigation as they may be regarded as extreme points;
- The fitted curve is probably too complex and may need simplification (so that the model will be able to generalize better on new data).

Inspecting Figure 7 provides the following insights:

- The curve is slightly nonlinear and it is worth investigating whether the added complexity of the nonlinear curve is really needed.
- One partial residual seems out of place, as it is very unlikely that there can be 0.25 derogatory reports (the value 0.25 is the result of substituting the missing values with the mean value). This partial residual influences the fitted curve and should be further investigated.

The remaining variables exhibit linear or slight nonlinear relationships with the target. These plots should be further investigated and the complexity of the univariate function adjusted so as to achieve a reasonable fit.

3.3 COMPARISON

Inspection of the ROC Chart (Figure 11) reveals that the generalized additive model (indicated as *ImportModel* in the following charts) does significantly better than the logistic regression model. This is to be expected as two nonlinear trends have been incorporated into the model.

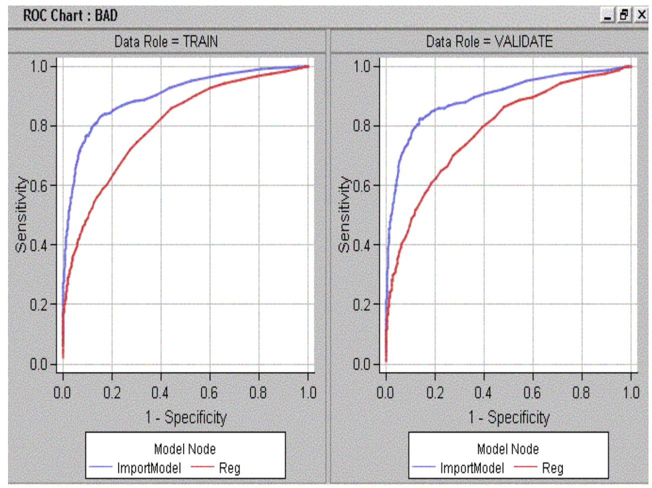

Figure 11: ROC Chart: BAD

The Score Rankings (Figure 12) also indicates that the generalized additive model is significantly better at discriminating between the BAD and GOOD applicants.

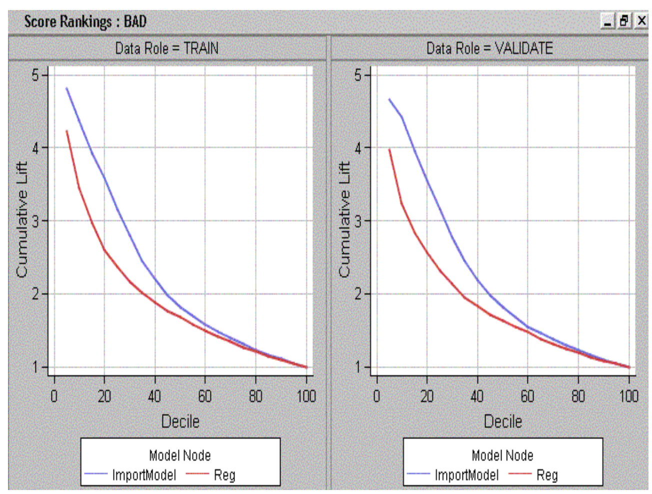

Figure 12: Score Rankings: BAD

The Classification Chart (Figure 13) further shows that the generalized additive model correctly identifies a significantly larger number of BAD customers (see Tables 1 and 3 for further details).

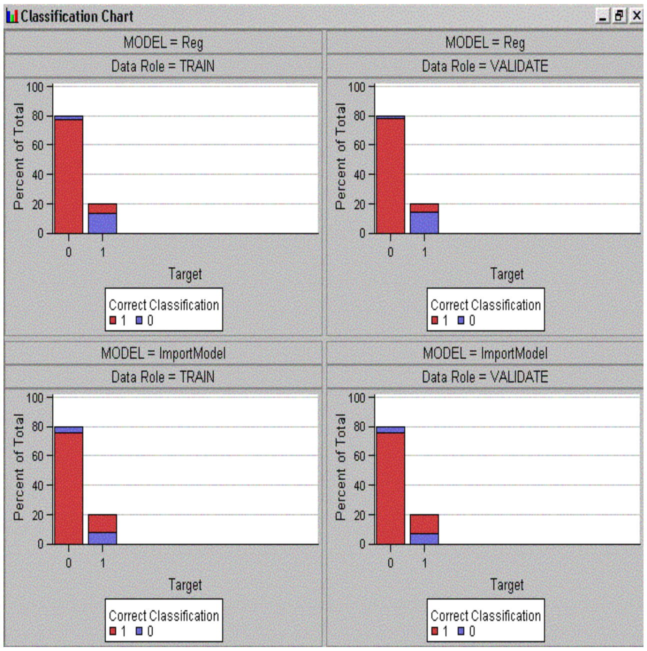

Figure 13: Classification Chart

The difference between the two models needs further investigation.

From the description of the logistic regression model (Section 3.1), it is clear that at least one important step was omitted from the modelling process: variable transformation was not done. This is usually done to model nonlinear trends in a more satisfactory way. From Figure 5 and Figure 7 it is clear that the generalized additive model incorporated these nonlinear trends into the computed model in a transparent way. It is furthermore not immediately clear what transformations must be done on the imputed variables to improve the logistic regression results so that similar results to that of the generalized additive model can be obtained.

The modeller now has a choice:

1. Use the more accurate generalized additive model, or
2. Search for transformations to improve the logistic regression model.

If the latter option is exercised, information from the generalized additive model may be used to improve the regression model:

1. Variables deleted from the generalized additive model may be investigated for deletion from the logistic regression model (it is interesting to note that in this specific example the variables excluded from the logistic regression are exactly the same as those excluded from the generalized additive model);
2. Variables exhibiting linear relationships with the target may be kept unchanged; and
3. Variables exhibiting nonlinear relationships may be candidates for transformation: suitable transformations should be searched for to model the nonlinear relationships.

The diligent reader may have realised that quite some effort may be needed to arrive at the given generalized additive model presented earlier. This effort may negate the stated advantages as it may have been spent on searching for suitable transformations for the logistic regression model giving similar results.

If the generalized additive model is constructed using an interactive approach relying on the inspection of partial residual plots, this argument may be valid. However, it has recently been shown by du Toit and de Waal (2003, 2005) that the construction of generalized additive models may be automated. No user interaction is required except for specifying the maximum complexity of the univariate functions. This approach has been implemented as a modelling node in SAS® Enterprise Miner (called AutoGANN) and may be used to compute the generalized additive model automatically giving substantial time savings while still resulting in an accurate model. The generalized additive model presented in this paper was constructed using this modelling node, but with limited time allowed for the construction of the model. The model is therefore not optimal and may be further improved upon.

4. BUILDING A SCORECARD USING A GENERALIZED ADDITIVE MODEL

The building of scorecards with logistic regression is well-understood and standard practice in many companies (Anonymous). Given the benefits of constructing a generalized additive model, the process of building a scorecard with this model is explained in the rest of this section.

If it is possible to get access to the outputs of the univariate functions computed by the generalized additive model, these outputs may be used to build a scorecard in the traditional way: the outputs of the univariate functions become the inputs to a classical scorecard building process that includes variable grouping and logistic regression on the weights of evidence. However, in building the scorecard, no variable transformations are considered. The generalized additive model computed the transformations and is now considered a preprocessing step.

The building of the scorecard is completed using the outputs of the generalized additive model (univariate functions) computed in Section 3.2 as inputs. The scorecard built the traditional way with only logistic regression is given in Table 4 and the new scorecard built exploiting the generalized additive model is given in Table 5.

The scorecards are very similar, except for DEBTINC that now has a wider range of scorecards points. This is to be expected as DEBTINC was identified as the variable with the most complex nonlinear relationship with the target. The scorecard points have also been changed dramatically for this variable.

Split values for the new scorecard are computed by running the original split values through the univariate functions giving "transformed" split values that are then used to built the scorecard. Note that the original variable groupings have been preserved and are transferred to the new scorecard, although the scorecard was actually built on the output of the generalized additive neural network. Keeping variable groupings unchanged is not optimal and groupings for variables exhibiting nonlinear relationships with the target should be reevaluated.

Characteristic Name	Attribute	Scorecard Points
Clage	.->120	40
Clage	120->180	50
Clage	180->240	57
Clage	240->	74
Clno	.->9	34
Clno	9->13	60
Clno	13->18	59
Clno	18->24	55
Clno	24->28	54
Clno	28->.	49
Debtinc	.->30	96
Debtinc	30->35	29
Debtinc	35->40	82
Debtinc	40->.	54
Delinq	.->0.9	65
Delinq	0.9->1.9	34
Delinq	1.9->.	4
Derog	.->1.9	55
Derog	1.9->.	15
Job	Sales	35
Job	Self	44
Job	Mgr	49
Job	Other	50
Job	Profexe	58
Job	Office	62
Mortdue	.->49999	44
Mortdue	49999->69999	52
Mortdue	69999->.	58
Ninq	.->0.9	61
Ninq	0.9->3.9	51
Ninq	3.9->.	29
Value	.->74332	55
Value	74332->107824	54
Value	107824->.	49
Loan	.->5999	19
Loan	5999->9999	50
Loan	9999->14999	53
Loan	14999->19999	55
Loan	19999->24999	61
Loan	24999->29999	59
Loan	29999->.	56

Table 4: Scorecard Built with Logistic Regression

Characteristic Name	Attribute	Scorecard Points
Clage	.->120	38
Clage	120->180	49
Clage	180->240	58
Clage	240->	74
Clno	.->9	34
Clno	9->13	60
Clno	13->18	59
Clno	18->24	55
Clno	24->28	54
Clno	28->.	49
Debtinc	.->30	106
Debtinc	30->35	95
Debtinc	35->40	84
Debtinc	40->.	22
Delinq	.->0.9	65
Delinq	0.9->1.9	34
Delinq	1.9->.	4
Derog	.->1.9	55
Derog	1.9->.	17
Job	Sales	36
Job	Self	44
Job	Mgr	49
Job	Other	50
Job	Profexe	58
Job	Office	61
Mortdue	.->49999	41
Mortdue	49999->69999	53
Mortdue	69999->.	60
Ninq	.->0.9	60
Ninq	0.9->3.9	51
Ninq	3.9->.	32
Value	.->74332	55
Value	74332->107824	55
Value	107824->.	48
Loan	.->5999	27
Loan	5999->9999	51
Loan	9999->14999	50
Loan	14999->19999	60
Loan	19999->24999	55
Loan	24999->29999	65
Loan	29999->.	55

Table 5: Scorecard Built with GANN

Score Range	Cumulative Count	Cumulative Number of Goods	Cumulative Number of Bads	Marginal Badrate	Cumulative Badrate	Approv Rate
626<=Score<641	14	14	0	0.00	0.00	0.71
612<=Score<626	101	99	2	2.30	1.98	5.13
597<=Score<612	240	236	4	1.44	1.67	12.20
583<=Score<597	472	465	7	1.29	1.48	23.98
569<=Score<583	651	635	16	5.03	2.46	33.08
554<=Score<569	850	813	37	10.55	4.35	43.19
540<=Score<554	1065	1007	58	9.77	5.45	54.12
525<=Score<540	1337	1244	93	12.87	6.96	67.94
511<=Score<525	1516	1369	147	30.17	9.70	77.03
497<=Score<511	1652	1465	187	29.41	11.32	83.94
482<=Score<497	1765	1525	240	46.90	13.60	89.68
468<=Score<482	1833	1550	283	63.24	15.44	93.14
453<=Score<468	1896	1569	327	69.84	17.25	96.34
439<=Score<453	1925	1575	350	79.31	18.18	97.82
425<=Score<439	1943	1575	368	100.0	18.94	98.73
410<=Score<425	1958	1575	383	100.0	19.56	99.49
396<=Score<410	1961	1575	386	100.0	19.68	99.64
381<=Score<396	1964	1575	389	100.0	19.81	99.80
367<=Score<381	1967	1575	392	100.0	19.93	99.95
353<=Score<367	1968	1575	393	100.0	19.97	100.0

Table 6: Gains Table (Logistics Regression)

Score Range	Cumulative Count	Cumulative Number of Goods	Cumulative Number of Bads	Marginal Badrate	Cumulative Badrate	Approva Rate
632<=Score<648	23	22	1	4.35	4.35	1.17
617<=Score<632	129	126	3	1.89	2.33	6.55
602<=Score<617	316	310	6	1.60	1.90	16.06
586<=Score<602	619	605	14	2.64	2.26	31.45
571<=Score<586	891	861	30	5.88	3.37	45.27
556<=Score<571	1044	1003	41	7.19	3.93	53.05
540<=Score<556	1195	1140	55	9.27	4.60	60.72
525<=Score<540	1342	1268	74	12.93	5.51	68.19
510<=Score<525	1483	1370	113	27.66	7.62	75.36
495<=Score<510	1613	1452	161	36.92	9.98	81.96
479<=Score<495	1740	1523	217	44.09	12.47	88.41
464<=Score<479	1819	1545	274	72.15	15.06	92.43
449<=Score<464	1879	1563	316	70.00	16.82	95.48
433<=Score<449	1921	1573	348	76.19	18.12	97.61
418<=Score<433	1940	1575	365	89.47	18.81	98.58
403<=Score<418	1957	1575	382	100.0	19.52	99.44
387<=Score<403	1963	1575	388	100.0	19.77	99.75
372<=Score<387	1967	1575	392	100.0	19.93	99.95
342<=Score<357	1968	1575	393	100.0	19.97	100.00

Table 7: Gains Table (GANN)

The change in scorecard points for the variable LOAN is due to the fact that only four decimal places were allowed for split values (the "transformed" split values are on a completely different scale compared to original split values). Closer correlation of the groupings (and therefore scorecard points) can be achieved by allowing more precision for the "transformed" split values.

Two extracts from the gains tables for the two developed scorecards are given in Tables 6 and 7. The difference in accuracy between the two scorecards is obvious. From the 9th grouping (highlighted) the following results can be inferred:

- With an approval rate of approximately 77%, the bad rate for the improved scorecard is more than 2% lower than that of the original scorecard.
- The approval rate using the new scorecard can be increased from 77% to nealy 82% giving a similar bad rate to that of the old scorecard.

These improvements are significant and may provide huge monetary benefits to companies willing and able to exploit the power of generalized additive models.

5. CONCLUSIONS AND FUTURE WORK

As nonlinear models are better understood and become available in statistical and data mining systems, the move from linear models to nonlinear models is inevitable (progress waits for no one). There are however external constraints, such as the need for regulatory approval, that may hinder or temporarily delay the replacement of linear models by yet unproven, but potentially more powerful, nonlinear models such as generalized linear models and neural networks.

In this paper a way forward is sketched. The standard theory of scorecard building is not tampered with, but a preprosessing step is introduced to arrive at a more accurate scorecard that discriminates better between good and bad applicants. The preprocessing step exploits generalized additive models (implemented as generalized additive neural networks) to achieve significant reductions in marginal and cumulative bad rates.

Future work includes a detailed comparison between classical variable transformations as done in regression modeling and the univariate functions computed by the neural network.

Acknowledgements

The authors wish to thank SAS Institute for providing them with Base SAS® and SAS® Enterprise Miner software used in computing all the results presented in this paper.

This work forms part of the research done at the North-West University within the TELKOM CoE research programme, funded by TELKOM, GRINTEK TELECOM and THRIP.

Bibliography

Anonymous, *Building Consumer Credit Scoring Models with Enterprise Miner White Paper*. SAS Institute Inc., Cary, NC, United States of America.

Berk, K. N. and Booth, D. E. (1995), Seeing a curve in multiple regression. *Technometrics*, **37**(4):385-398.

Cai, Z. and Tsai, C. (1999), Diagnostics for nonlinearity in generalized linear models. *Computational Statistics and Data Analysis*, **29**:445-469.

Du Toit, J. V. and De Waal, D. A. (2003), Automated construction of generalized additive neural networks for predictive data mining. In *Proceedings of the 33rd annual conference of the South African Computer lecturer's Association*, J. Mende and I. Sanders, eds.

Du Toit, J. V. and De Waal, D. A. (2005), The AutoGANN node in SAS® Enterprise Miner (in preparation).

Ezekiel, M. (1924), A method for handling curvilinear correlation for any number of variables. *Journal of the American Statistical Association*, **19**(148):431-453.

Friedman, J. H. and Stuetzle, W. (1981), Projection persuit regression. *Journal of the American Statistical Association*, **76**(376):817-823.

Hastie, T. J. and Tibshirani, R. J. (1990), Generalized Additive Models, Vol. 43 of *Monographs on Statistics and Applied Probability*, Chapman and Hall, London.

Hosmer, D. W. & Lemeshow, S. (1989), *Applied logistic regression*, Wiley, New York.

Kleinbaum, D. G. (1994), *Logistic regression: a self-learning text*, Springer, New York.

Larsen, W. A. and McCleary, S. J. (1972), The use of partial residual plots in regression analysis. *Technometrics*, **14**(3):781-790.

McNab, H. & Wynn, A. (2000), *Principles and Practice of Consumer Credit Risk Management*, Financial World Publishing, Canterbury.

Potts, W. J. E. (1999), Generalized additive neural networks. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 194-200.

Potts, W. J. E. (2000), *Neural Network Modeling Course Notes*, SAS Institute Inc., Cary, NC, United States of America.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge, United Kingdom.

Sarle, W. S. (1994), Neural networks and statistical models. In *Proceedings of the Nineteenth Annual SAS Users Group International Conference.*

Thomas, L. C., Edelman, D. B. & Crook, J. N. (2002), *Credit Scoring and Its Applications*, SIAM, Philadelphia.

Wielenga, D. and Lucas, B. & Georges, J. (1999), *Enterprise Miner: Applying Data Mining Techniques Course Notes.* SAS Institute Inc., Cary, NC, United States of America.