Res. Lett. Inf. Math. Sci. (2002) **3**, 161-189
Available online at http://www.massey.ac.nz/~wwiims/research/letters/

Data Mining in the Survey Setting: Why do Children go off the Rails?

Judi Scheffer
Institute of Information and Mathematical Sciences,
Massey University at Albany,
P.O. Box 102904 N.S.M.C,
Auckland 1310,
New Zealand.
J.Scheffer@massey.ac.nz

Abstract
Data Mining is relatively new in the field of statistics, although widely used elsewhere. Is it a good idea to discard the model-based methods in favour of Data Driven methods? Data driven methods produce a high degree of accuracy, but very little interpretability. Model based methods are interpretable, but lack accuracy. Data mining techniques are commonly used where the data collection has been automated. I will show these methods are also useful in the large survey setting.

Introduction
The NASF is the national survey of American Families.
Within this very comprehensive survey, is the Focal Child Survey, Focus on the Child. This survey was first conducted in 1997, and repeated in 1999. It is the 1999 data I intend to focus on, as I am not interested in the longitudinal aspects, just the current data. In the original 1999 data there are 35938 cases and 316 variables. With such a comprehensive data set, it was interesting to see whether data mining techniques could be applied, and if any relationships could emerge from the data, describing what causes things to go wrong when bringing up children what are the positive aspects to prevent children getting into trouble? Generally data mining is an automated process. Central to this is model building. A representative model is created based on an existing data set which is useful for predicting trends, patterns, and correlations and provides predictions based on historical outcomes. (Groth, 1988)
The aim of the very extensive survey is to describe the American Family. The particular aim of this study is to identify children 'at risk' (also the aspects of family life which help prevent children becoming 'at risk').
There are two types of software used for this analysis, SAS Enterprise Miner, and Clementine 6.0. SAS Enterprise Miner had the advantage of being able to handle larger amounts of the data, that is the complete data was used, and could be carved up into training (50%), validation (30%), and test (20%) data sets. The output was largely more comprehensive, as is usual with SAS. Output with Enterprise Miner tends to be in the form of a HTML report, which makes extracting the appropriate bits difficult. There are a lot of secondary files to be searched to find useful outputs. In fairness it is better to have this, as some of the unrequired output may be useful in a different application.
There are positive aspects to Clementine, particularly the sensitivity analysis given as output from the neural network terminal node, which is very interpretable, and very useful. Another good feature of Clementine are the ability with Neural Network node, to be able to prune the model, and rerun that stream with a lesser number of inputs, and check thus the change in sensitivity, and accuracy. Clementine gives a rather terse output to its terminal nodes. (See glossary of terms)

Methodology
Classic Statistics will produce a top down standard Scientific Analysis. First a hypothesis is formed, then the Statistician/ Data Analyst will go about testing that hypothesis. Data mining will produce a bottom up analysis, looking purely at the data, and what information it may contain which may be of use, very often containing previously unsuspected relationships. This makes this type of analysis particularly suitable for data that has been collected automatically, e.g. banking, credit card transactions, telephone calls, swipe card access, supermarket shoppers, loyalty card programs etc.

1: Neural Networks

> "A neural network is a massively parallel-distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:
> Knowledge is acquired by the network through a learning process.
> Interneuron connection strengths known as synaptic weights are used to store the knowledge.
> "(Aleksander and Morton, 1990)

A biological neuron can be thought of as a cell that joins on to (transmits to) other neurons by means of synapses (like fibres). Neurons are said to be in an on/off state, when they fire they are activated. Neurons have a threshold level, above which they are on, below which they are off. The model neuron computes a weighted sum of its inputs from other neurons, and outputs a one or zero according to whether this sum is above or below the threshold.

$$net = \sum_{i=1}^{N} w_i x_i - \theta$$

Where x_1, x_2, x_3,…….x_n are the inputs to the neuron (this could also be inputs from other neurons), w_i (i=1, 2, 3,……., N), is a weight representing the strength of the synapse connecting neuron i to the current neuron, net is the net input into the current neuron, and θ is the threshold value.

$$\text{Now } y = f(\text{net}) = f\left(\sum_{i=1}^{N} w_i x_i - \theta\right), \text{ and } f(net) = \begin{cases} 1 & net \geq 0 \\ 0 & net < 0 \end{cases}$$

(The activation function). This lends itself very well to the logistic function.

$$f(net) = \frac{1}{1 + \exp\left(\frac{-net}{Q_0}\right)}$$

Another popular option is the hyperbolic tangent function, which is:

$$f(net) = \frac{1 - \exp(-net)}{1 + \exp(-net)}$$

where Q_0 is the 'temperature' of the neuron. The 'temperature' is merely the step function when close to 0, and the sigmoid curve when high. Both of these functions are available as options within SAS EM (among others), the hyperbolic tangent is the default activation function for the Neural Network node in SAS EM.

A perceptron is the simplest form of a neural network used for classification of two linearly distinguishable groups. Multi-layer perceptron networks as used here are trained by back propagation, and the knowledge required to map input layers into an appropriate classification is represented by the weights. Training the network is done to some predetermined error limit. These weights are frozen, and the validation data is run through the network, and the error rate is tested. Finally the test, or new data is run through the network, allowing prediction of new data. The mean squared error is used as a measure of how close the network is to establishing the desired result. To avoid the problems of getting false results due to local minima on the surface, it is a good idea to repeat the analysis using many different seeds (starting values), this way the true relationship may emerge. Genetic algorithms are a method of avoiding this problem.

If a neural network is applied, and there is a single continuous input and a single output, this is simple linear regression. If there are multiple inputs and a single output this is multiple linear regression. When hidden layers are added, an activation function is applied to the hidden layer. A multiplayer perceptron model has hidden layers that employ non-linear activation functions. (Westphal, Blaxton; 1998)

In the present study a single hidden layer was tried using different numbers of neurons in the hidden layer. The number that gave the smallest average error rate, and the smallest AIC1 was found to be the best model, this turned out to be 21 neurons and subsequent using different numbers of layers, each with 21 neurons in each hidden layer. Two hidden layers gave an even worse result than one, but three was significantly better. Next to be tried was changing the number of neurons in each layer, so that there would be progressively fewer neurons in each successive layer. Many models were tried before coming up with what seemed to be an optimal one, both in terms of average error, and AIC. This turned out to be 21, 14, 8 neurons successively, with 37 input variables and 1 output variable. When 4 hidden layers were tried, there was no improvement in the model; in fact it appeared to give a less accurate result. Many models were tried, but one representative of its type will be showed, to illustrate the point. Different results were gained with the two different software packages; I put this down to the use of random number seeds. This shows that there is a problem present of hitting local maxima on the surface being studied. The way around this would be to repeat these analyses many times using different seeds, and a pattern which is the global maxima is should soon emerge. SAS does not give a sensitivity analysis, which Clementine does.

2: Decision Trees (including Classification and Regression trees)

In SAS Enterprise Miner, trees are called a Decision Tree. Clementine provides a Classification and Regression Tree option that gives a similar output to SAS. This procedure uses both continuous and categorical dependent variables, and discriminates (classifies) for categorical variables and produces regression trees for continuous variables. There is an automated decision rule, which uses a nonparametric method that splits a node based on the data. Only binary splits are produced. Output from Clementine will produce a tree (and its rules), the statistics of each node and a gain chart or a risk chart. Output from SAS EM will include a non-portable tree, and English 'Rules' for splitting. Also a graph of the tree showing 'rings', with the input as its centre, with each level being a ring, and the 'leaves' are the outside layer. This shows where the splits are, and how many rows of data belong in each leaf. It is possible to correlate the output of the nodes with the dependent variable, to get an indication how much of the variation in the data is being described.

The advantage of this method is that it is quick and easy, and doesn't rely on normality of the data, or independence of observations. However if the distribution is known, particularly if it is normal, Regression will be a better option. A major disadvantage of decision trees is that the solution is non-unique, and there is no best tree solution, and sometimes the solution is intractable.

3: Regression Analysis

Regression Analysis is the cornerstone of traditional statistical analysis, particularly in the survey setting. To this end it is useful to compare results from older known methods with the results of newer techniques. In the data mining setting, regression is a tool applied using the same training, validation and testing procedures, which characterise this approach. While called regression, in the SAS EM data mining setting, it is in fact a generalised linear model as not only continuous outputs are used, but also binary and ordinal outputs are also available, by means of logistic regression. The method used here is maximum likelihood. Both Clementine, and Enterprise Miner have a comprehensive array of options available to be used. The default settings were not particularly helpful here, and require resetting for use with survey data. The stepwise option is useful for model selection, but invites a large amount of output. For those with a high competency in optimisation, there is a bewildering array of options available, some of which slow the process down unacceptably. The criterion for model selection is the smallest negative log-likelihood. Also given in the output are the AIC, the SBC, and many other choices also.

4: Factor Analysis

This is available in Clementine 6.0, as a Data Mining option. This again is a case of Classic Statistics being dressed up as a data-mining tool, as are many multivariate techniques. Factor Analysis is used with a principal components method. This is a very widely used multivariate technique. Some results are shown in an appendix for the purposes of comparison.

5: Kohonen Self-organising Maps

These are a form of a two dimensional unsupervised neural network. As the data is trained, a density 'map' is shown (Clementine 6.0), changing as the pattern is trained. The object of this is to discover which observations should be clustered together. When two input patterns predict the same output, then it can be assumed they belong to the same output cluster. Output from the node is somewhere between statistical clustering and neural networks. Clustering divides data into groups according to the characteristics within the actual data, and classifies the observations into groups according to the inputs. Usually the default for both Clementine and SAS EM is the Euclidean distance between two points X and Y

$$\|X - Y\| = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

Where X and Y are N-dimensional input patterns. (This is based on Pythagoras' theorem).

However it is possible in SAS to specify K-means Clustering. The resulting 'Map' which is shown of the clusters is really an aid to view the data, which started out multi-dimensional, in say two dimensions, shown in its natural clustering. This becomes a form of pattern recognition.
The SOFM (Self Organising Feature Map) Algorithm:
The Initial weights are set to small random values; make the 'neighbourhood' size large.
Calculates the distance between the current input and each neuron, for each observation.
The neuron with the minimum distance from input to 'weight' of neuron is the winner, and the algorithm updates the weights connecting the input layer to this neuron.

$$w_{ji}(t+1) = w_{ji}(t) + c\lfloor x - w_{ji}(t) \rfloor$$

Where $c = \alpha(t)\exp\left(- \|r_i - r_m\| \big/ \sigma^2(t) \right)$ for all neurons j in $N_m(t)$, r_i-r_m is the distance between neuron i and the winning neuron m (Smith, 1999), and where $\alpha(t)$ and $\sigma^2(t)$ are the two functions controlling the rate of learning. The algorithm iterates until the weights have stabilised.
Output from SOM's can then be correlated with the target variable, to look for relationships, if true ones exist. SAS will provide a cluster map, showing the circles of each cluster. Relating this to the outcome of interest via correlation will show whether this is a useful tool in this case.

Data Preparation
In the original 1999 data there are 35938 cases and 316 variables. A good number of these variables were flags for imputation. The 'public use' imputed data, turned out to be unusable in the data mining setting. This needed to be transformed into a more 'Data Mining Friendly Format', as manual checking was impossible given the large size of the data set.
1: Raw Data
The raw data (non-imputed) was tested on both Data Mining packages used. In both cases the software was unable to handle even small subsets of the data due to missing values. The decision was made to impute the data, to provide a single complete dataset, using regression imputation, with an added random error component. This was done in SAS using the PRINQUAL procedure, this gave rise to a data set that included original variables. However in the case of the categorical variables, some of the results were a little odd, due to the addition of the added random component. The data had been numericised, and the imputed continuous dependent variable had a few negative values, which is not really possible. As this came about due to the uncertainty due to imputation, this anomaly was allowed to remain, so as to preserve correct relationships within the data. Single imputation was used, and while it is not usually the best form of imputation, it probably is the best option in this case, as recombining results using multiple imputations would be very difficult to interpret, when using techniques such as neural networks. Would the standard means of estimates, variance (between, within datasets) be valid after applying neural networks? A team of researchers in Finland is currently researching this, and it is better to leave this until the results of that research is known. The single imputation provided not only a set of the original data that was complete (although imputed), but also a base to move forward and do a principal components analysis to reduce the dimensionality of the data.
2: Imputed Data
The imputation had to be done by carving the dataset into like type (similar topic questions) variables, for imputation. The reason for doing this was the SAS PROC PRINQUAL was unable to perform this task when the data set was entire, a singular matrix was returned each time, and the data set was simply too large. The imputed data was gathered together in one dataset, leaving out the original imputation flags. This resulted in a data set of 151 input variables, and one dependent variable. However when carved into eighteen different datasets, then the PRINQUAL procedure was able to be applied, and then recombined to give the dataset FCIMPQ. Even so data mining was difficult, error ridden, and the entire data set was too large for the software. It was found that all variables were not able to be included at once, and some form of variable selection was needed. Preliminary analysis showed these 151 variables to be far too many in number for the more sophisticated analyses, particularly Neural Networks. This was however attempted with different combinations of the variables, but the average error rate remained high, as did the AIC, SBC and there was a constant question over whether the right subset of variables had been chosen. All the time there was the question also of the architecture of neural networks, how can one decide the best architecture, without knowing the best subset of variables to use?

To further reduce the dimensionality in the data, and to look for the more obvious relationships, it was decided to run the data through stepwise regression, to pick out the best subsets. The 'best' model as predicted by forwards and backwards stepwise, turned out to be a problem when run through the PROC REG procedure. There were 56 predictors, all highly significant. The R^2 was 79.4%, and the mallows C_p was negative. It was decided to look at the Variance Inflation Factors (VIF's) to see if multicolinearity was a problem. It most certainly was, and some very significant predictor variables were dropped from the model. Fortunately as they were dropped other variables did not become insignificant as often happens with correlated data. Eventually, a model was settled upon with acceptable VIF's (nothing above six, the last one to be dropped was fourteen), and an R^2 of 72.1% (the amount of variation in the data described by the model), and 37 predictor variables. In Clementine this could be used after sampling (SRS) the data by 25%. Some of the more simple techniques could be used after 50% sampling within Clementine. SAS Enterprise Miner could use the complete data, although both packages carved the data up in to training, validation and test datasets. This is the dataset FCIMPQ described in appendix 2.

3: Pre-Processed Data

As an alternative method of constructing the input variables, the 151 variables were pre-processed by means of principal components, generally discarding those components with eigenvalues of less than 2 (for each of the eighteen imputed datasets). Correlation between the original variables and the principal components provided interpretability. These retained variables were then put back together along with the constructed dependent variable. While 'putting together' a pile of principal components from different datasets is unusual, it provided dimensionality reduction from 151 independent variables down to 37. The dataset that was finally created; using the first two principal components from each of the eighteen datasets (one data set with many variables provided three), which the original variables were carved into. This provided thirty-seven predictor variables, with around 72% of the variation in the data represented by these. The variables in this dataset differ from those in the original variable data set. This is explained by the multicolinearity present. The dataset is shown in Appendix 1. A check for correlation among the constructed variables showed those coming from the same dataset to be orthogonal (as expected), and those coming from different datasets to only slightly have a problem, so analysis with these variables would not be violating independence.
This formed the dataset FCIMPC.

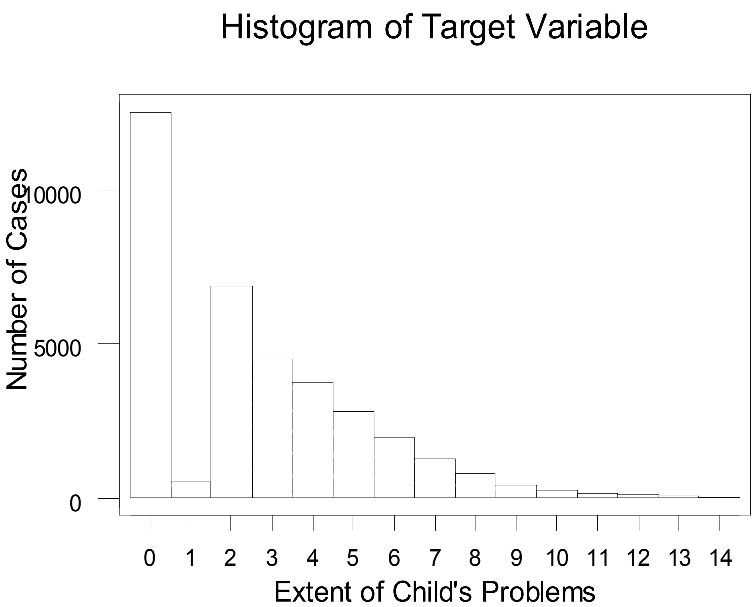


Fig 1: Histogram of Target Variable

4: The Target (Dependent) Variable
The dependent variable was constructed by first adding together the scores for the amount of difficulty children between the ages of six and eighteen experience (getting into trouble). The lower the score, the more difficult the child. Children under age six and over age eighteen were given scores of twenty, the maximum, as this was outside the range of interest. This score was subtracted from twenty to give zero, no problems up to a total of fourteen, maximum problems.
The original variables were themselves indexes, constructed from other variables. However by making no problems a zero, and maximum problems fourteen, this index becomes a linear scale, which is more intuitively interpretable. This is used as the dependent variable for both data sets.

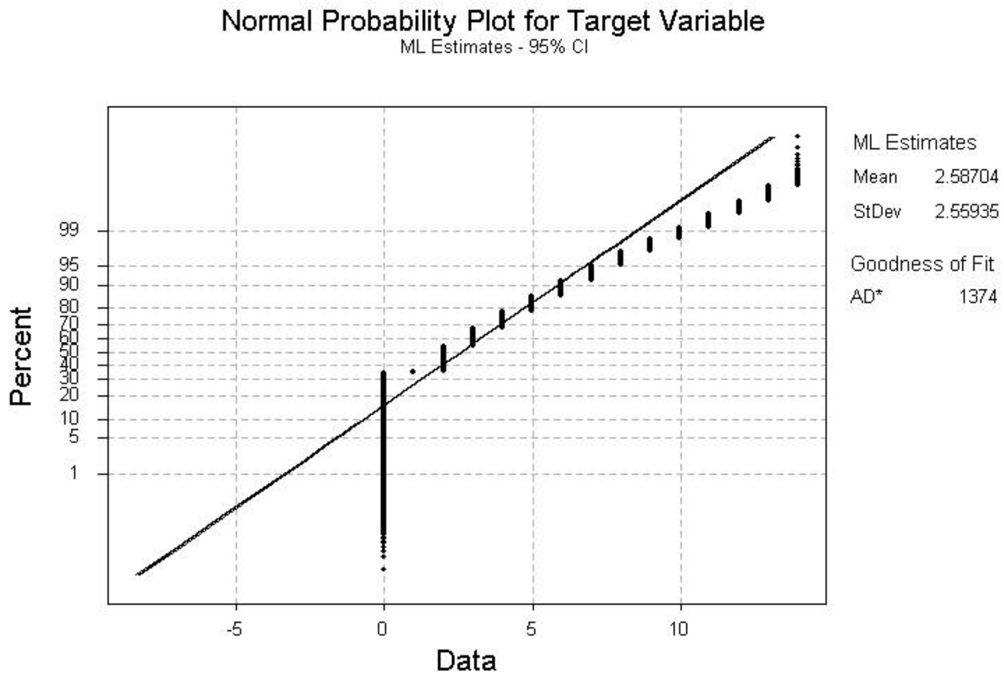


Fig 2: Normal Probability plot of Target Variable

This clearly shows that the assumption of normality used for most Statistical Analyses is not upheld. However normality in the target variable is not required for NN or SOM or trees. What is called the 'Regression procedure' in data mining is in fact a Generalised Linear Model using maximum likelihood.

Results for FCIMPC data set
Analysis was using both SAS EM and Clementine 6.0.
An example of a SAS EM diagram for FCIMPC is as follows as shown in Fig. 3i, and Fig. 3ii:
This is a simple diagram with one neural network and one regression, follows a diagram with 8 different neural networks. These diagrams show the nodes being run.
The diagrams required running many times, as each time the nodes (when the options within SAS were changed), took sometimes a little while other times a long time. Running these diagrams often took at least a week, sometimes longer. A major problem with running this software in a student lab environment, if students are to use this software) was that as soon as the screen saver was activated, the neural network training / validation graph would cease to operate, and essentially shut the whole program down. So delays processing this data were greater than was necessary. In the computer lab environment, the screen saver must be disabled before running SAS EM with any large amount of data - With the small well-behaved datasets this is not a problem. With the SAS EM neural networks the progress graph displayed was average error (See Fig 7), and this was reasonably close to zero. Clementine on the other hand gave a progress graph, which was the predictive accuracy, but this was unfortunately unable to be saved (This graph is not given as part of the available output).

Table 2: Comparison of three different models. (Regression, Neural Networks and Decision tree)

Model (Tool)	Error rate (T) (Average Error)	Error rate (V) (Average Error)	Error rate (Test (Average Error)	AIC	SBC
Regression	2.303	2.155	2.370	15035.16	15206.68
Neural network	0.01	0.01	0.02	-84091	-788883
Decision Tree	0.17	*	*	*	*

Table 2 gives a comparison of the average error (for training) for three types of modelling, and in the case of NN and Regression, the average error for validation and testing, as well as the comparative SBC and AIC.

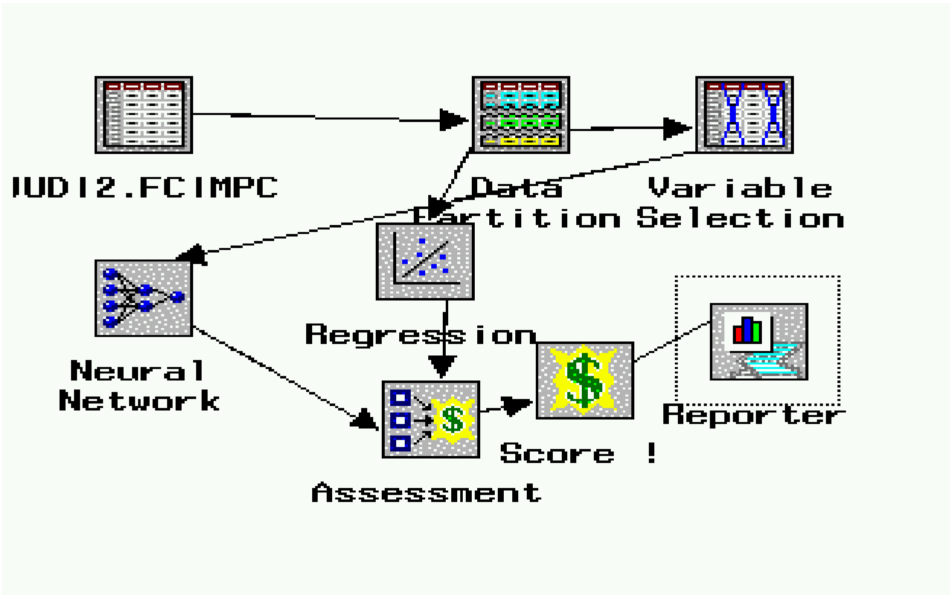


Fig 3i: SAS EM Diagram for FCIMPC, using different models.

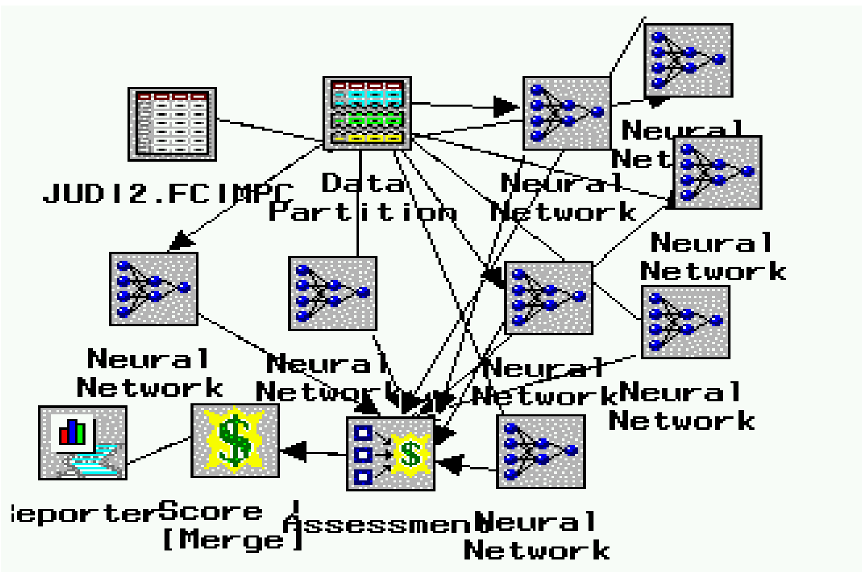


Fig 3ii: SAS EM diagram for FCIMPC, assessing different NN architectures.

Table 3: Table of Estimates and T-scores for DM Reg, for FCIMPC.

Variable	**Estimates (se)**	**T-scores**	**Pr > |t|**
Intercept	2.6553 (0.0013)	234.282	< .0001
actpsen	-0.66154 (0.0948)	-6.976	<.0001
amochc	-1.23756 (0.0161)	-77.060	<.0001
argtrub	-0.14249 (0.00632)	-22.531	<.0001
ccarr	0.04530 (0.00491)	9.234	<.0001
cdepwls	-1.03652 (0.0313)	-33.116	<.0001
chlivar	-0.07038 (0.00897)	-7.848	<.0001
cnotsch	-0.04147 (0.00524)	-7.917	<.0001
cpsfam	1.02913 (0.0908)	11.338	<.0001
defgby	-0.10229 (0.0376)	-2.721	0.0065
deviach	1.45685 (0.0245)	59.448	<.0001
homalon	-0.22135 (0.00667)	-33.179	<.0001
hwkgcc	0.02168 (0.00565)	3.837	<.0001
mendhth	0.06387 (0.00923)	6.916	<.0001
mkares	-0.08291 (0.00555)	-14.939	<.0001
negpagg	0.01774 (0.00866)	2.048	0.0405
pmhelp	0.03513 (0.0110)	3.189	0.0014
poverty	-0.07238 (0.00877)	-8.253	<.0001
pparagg	0.14504 (0.0301)	4.819	<.0001
sibsact	0.41786 (0.0381)	10.955	<.0001
sumsch	0.02130 (0.00670)	3.179	0.0015
suspwk	-0.12927 (0.0132)	-9.783	<.0001

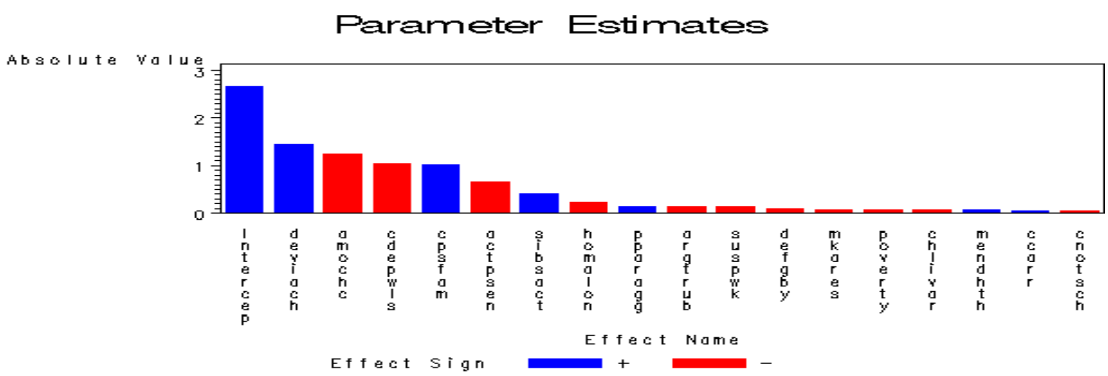


Fig 4: Parameter estimates for DM Reg., for FCIMPC

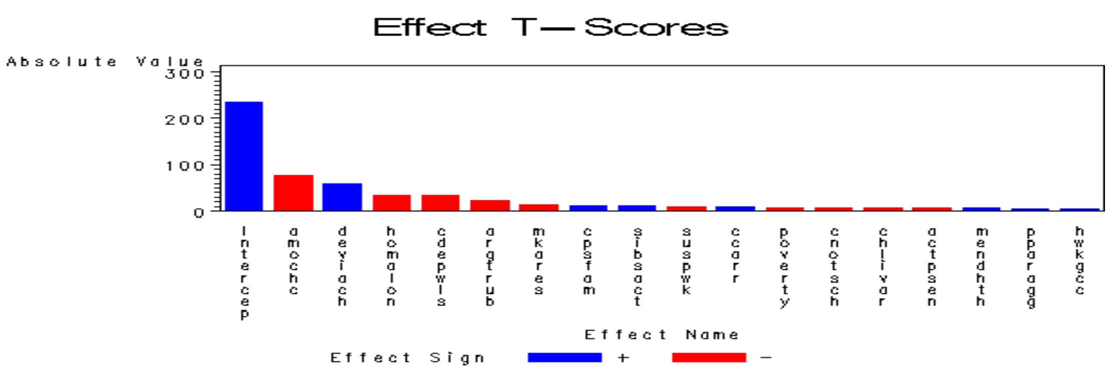


Figure 5: Effect of the T-Scores (Regression)

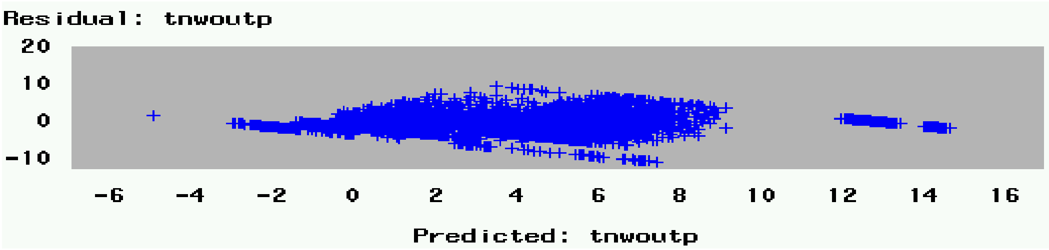


Fig 6: Residual vs. Fits plot for Regression Model.

The Regression Model: See Table 3, and Figs. 4, 5 and 6. Fig. 6 the residual vs. predicted by the model shows a reasonably cloud like pattern, so the assumptions underlying the model would appear to be reasonable. Figs 4 and 5 show the output graphs given by SAS EM DM Reg. Table 3 gives a table of coefficients, Amount of Child care is the most significant, with a highly negative score, but as this is a surrogate for age, this possibly is not the best predictor. Next is deviach (child lies cheats does poorly at school, and doesn't sleep well), contributing strongly to increasing the score. This is discussed further in the following section.

Neural Networks: See Fig 7 and Table 4.

The neural network output shows the average error rate to be close to zero, there is no divergence between the two lines, therefore the model is not over fitted. The model: 37 input layer (variables), 21 hidden layer 1, and 14 hidden layer 2, and 8 hidden layer 3, with 1 output layer has the lowest error rate, AIC, and SBC.

Fig 7: A Neural Network error plot for the FCIMPC dataset.

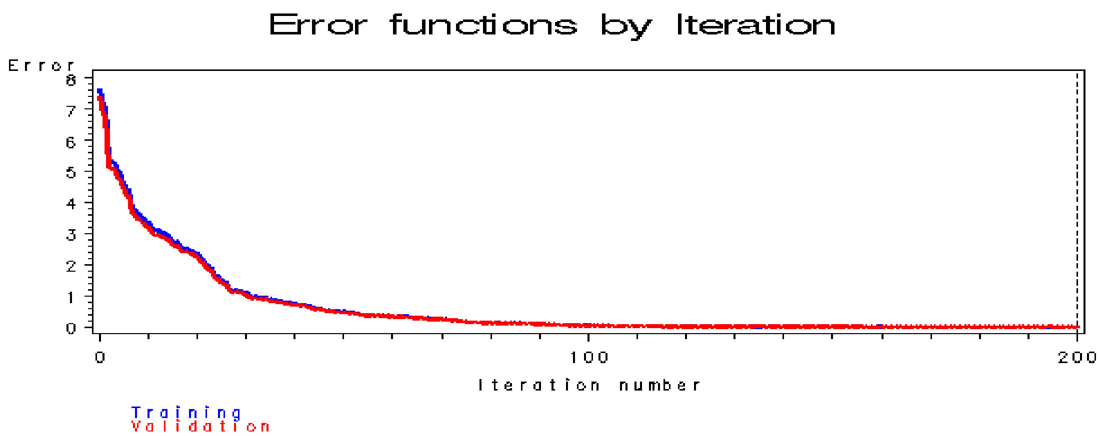


Table 4: A comparison of different NN architectures.

Architecture (Hidden layers)	Error rate (T) (Average Error)	Error rate (V) (Average Error)	Error rate (Test (Average Error)	AIC	SBC
9	0.09	0.12	0.12	-35223	-29013
13	0.08	0.11	0.12	-35223	-29013
21	0.11	0.14	0.12	-30596	-24386
21, 21	0.12	0.16	0.13	-28433	-18565
21, 21, 21	0.22	0.28	0.25	-18139	-4923
21, 9, 15	0.21	0.24	0.24	-19898	-10189
21, 14, 8	**0.01**	**0.01**	**0.02**	**-84091**	**-78883**
21, 14, 9	0.07	0.09	0.10	-34840	-25366
21, 14, 11	0.28	0.32	0.28	-15838	-6121
21, 15, 10	0.29	0.33	0.31	-15189	-5109
21, 15, 11	0.06	0.09	0.08	-37150	-27206

21, 16, 9	0.13	0.16	0.15	-26850	-16634
21, 16, 10	0.08	0.10	0.09	-33606	-23768
21, 16, 11	0.14	0.17	0.15	-25967	-16001
21,8,14,5	0.20	0.24	0.24	-20313	-10642
21 14 8 4	0.02	0.04	0.04	-52298	-42703
21 14 9 3	0.05	0.09	0.13	-39290	-29634
21 14 9 4	0.03	0.05	0.05	-48392	-38653
21 14 9 6	0.04	0.06	0.06	-43917	-34011

SOM clustering proximities for FCIMPC: See Fig. 8 and Table 5.
Figure 8 shows the clustering for children, based on Euclidean distance. Table 5 gives the relative importance of each input variable.
Fig 8: A SOM cluster Proximities map from SAS for FCIMPC

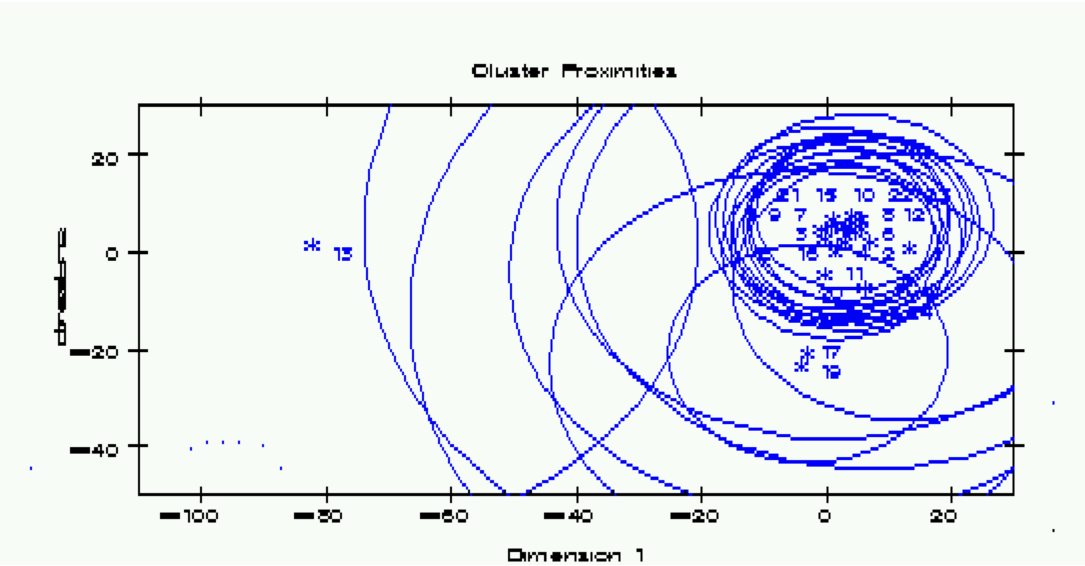


Table 5: Importance of input variables for FCIMPC SOM clustering

Variable	Order of Importance	Value	Description
UNHAPPY	1	1	Unhappy Child, doesn't socialise well, feels sad depressed, worthless and inferior, acts young for his/her age
HWKGCC	2	0.58375	Hours per week in group Child Care
HOMALON	3	0.49476	Child Home Alone whilst Parent Works
ATTSS	4	0.35739	Attended Summer School
PMHELP	5	0.27085	Child knows a place they can get help
SUMSCH	6	0.17716	Child attended Summer program
CNOTSCH	7	0.13418	Child Elsewhere, not at School
DEFGBY	8	0.11622	Does enough Homework to get by when Forced
NWELAT	9	0.10612	Negative attitude to welfare
PPARAGG	10	0.09494	Positive parent aggravation

The Decision Tree

The decision tree procedure within SAS EM gave 21 leaves, as shown in Fig. 9. The tree is given in Appendix 3.

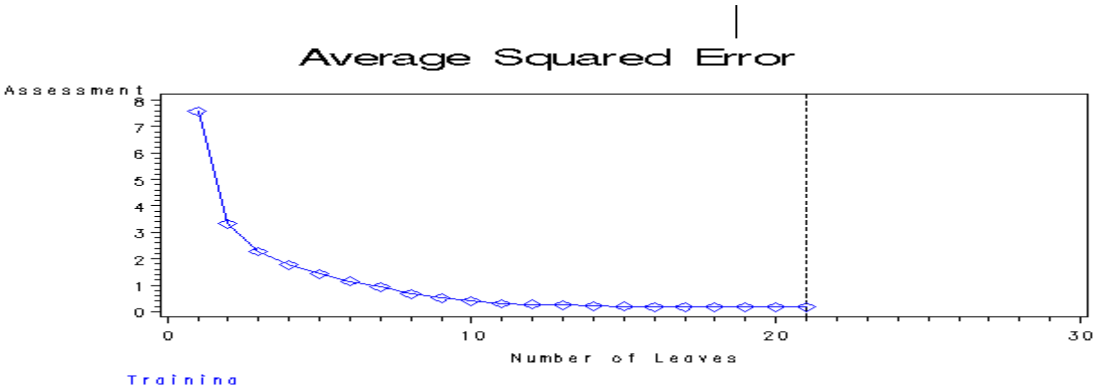


Fig 9: The number of leaves for FCIMPC.

Clementine Output. The Diagram from Clementine is as follows:

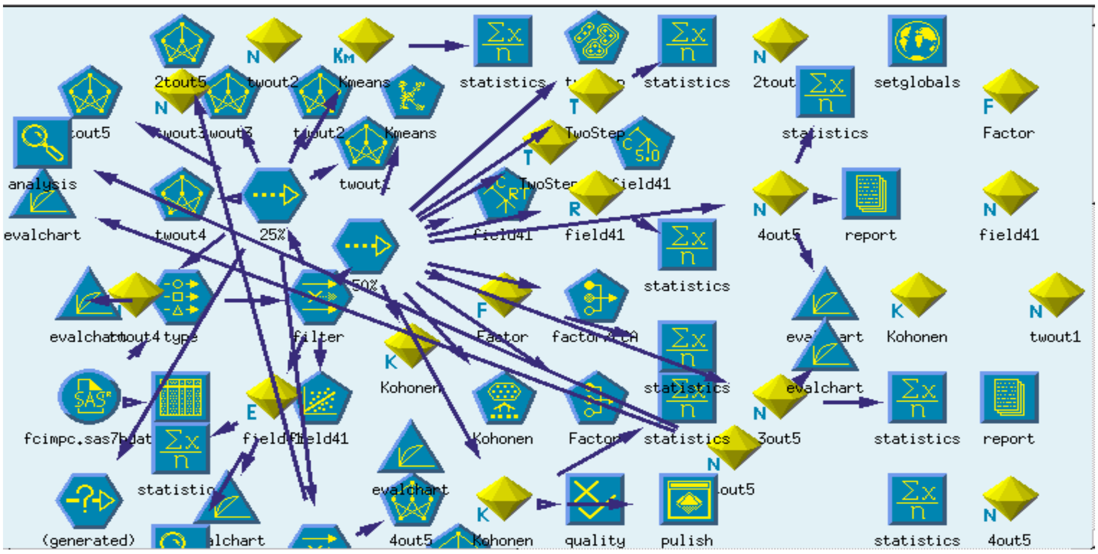


Fig 10: The Clementine diagram for FCIMPC

Clementine runs as slow as does SAS EM, with the exception that the screen saver did not interfere with the program. The probable reason for this is that with the neural networks, and the SOM's, both have rapidly changing screen output that prevents the screen saver from engaging. Some neural network nodes took more than a week to run, and this was using only 25% of the data. The SOM's took a couple of days to run, while regression, factor analysis, CART, and two-step methods took a matter of hours.

Table 6i: Regression from Clementine 6.0 for FCIMPC; model summary.

Regression Model Summary					
R	**Rsq**	**Rsqadj**	**SE of estimate**	**F**	**Sig**
.823(a)	.677	.677	1.5870	2077.806	.000

Table 6ii: ANOVA table for Regression for FCIMPC

ANOVA (b)					
Model.	**Sum of Squares**	**df**	**Mean Square**	**F**	**Pr > F**
Regression	94200.855	18.0	5233.4	2077.81	.000(a)
Residual	44903.5	17828.0	2.52		
Total	139104.338	17846.0			

Tables 6i, and 6ii show that the regression model is a significant one, 675 of the variation in he data has been described, and the F value of 2077.8 shows that the null hypothesis that all slopes are zero, is to be rejected, the model is significant. A discussion of the coefficient table is given in the next section.

Table 6iii: Regression Coefficients for FCIMPC

Regression Coefficients (a)					
Model	**B**	**Std. Error**	**t-Value**	**p > |t|**	**VIF**
(Constant)	2.682	.012	225.647	.000	*
field6	-5.656E-02	.006	-10.096	.000	1.043
field7	6.266E-02	.010	6.439	.000	1.585
field9	-.138	.007	-20.468	.000	1.079
field10	1.628E-02	.007	2.475	.013	1.019
field12	4.421E-02	.005	8.492	.000	1.261
field14	2.807E-02	.007	4.018	.000	1.505
field15	-4.116E-02	.005	-8.160	.000	1.014
field16	-.266	.007	-39.031	.000	1.477
field17	1.686E-02	.010	2.461	.014	1.002
field19	-2.509E-02	.006	-4.275	.000	1.025
field24	-.113	.014	-8.269	.000	4.976
field29	.753	.015	51.467	.000	6.089
field31	.769	.013	61.458	.000	3.368
field34	.188	.011	16.692	.000	1.755
field35	-9.510E-02	.009	-10.239	.000	1.832
field37	-1.325E-02	.007	-1.897	.058	1.034
field40	-7.802E-02	.009	-8.235	.000	1.287
(a) Dependent Variable: field41					

Comparison of Clementine models: See Tables 6i and 7.

Table 7: Comparison of output from Clementine for NN and Regression

Model	Architecture	Occurrences	S.D.	Correlation to target variable	Predicted accuracy %
Regression	Maximum Likelihood	18000	1.573	0.823	67.7
Neural Network	37 input 21 HL1 14 HL2 8 HL3 1 output	17973	2.7508	1.000	99.95

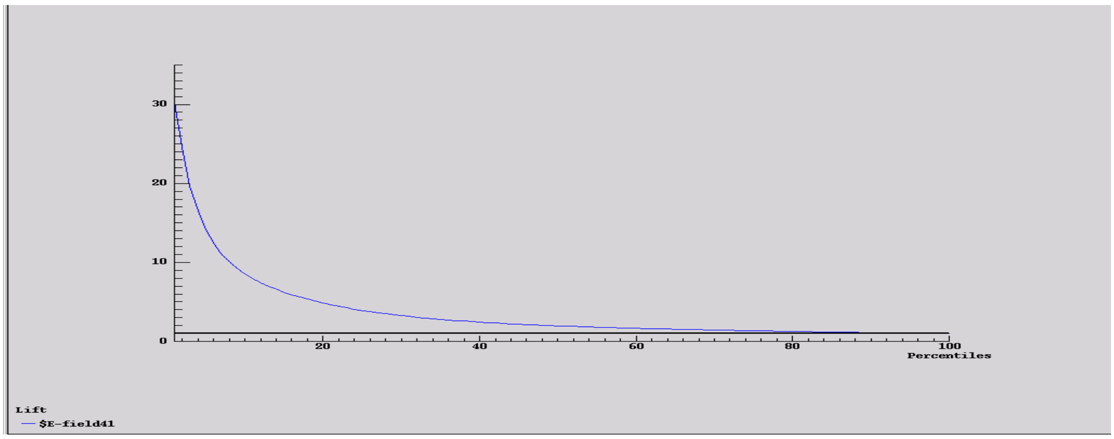


Fig 11: Lift Chart for Regression FCIMPC The Lift Chart show the amount of error in the model left, by percentiles of cases trained.

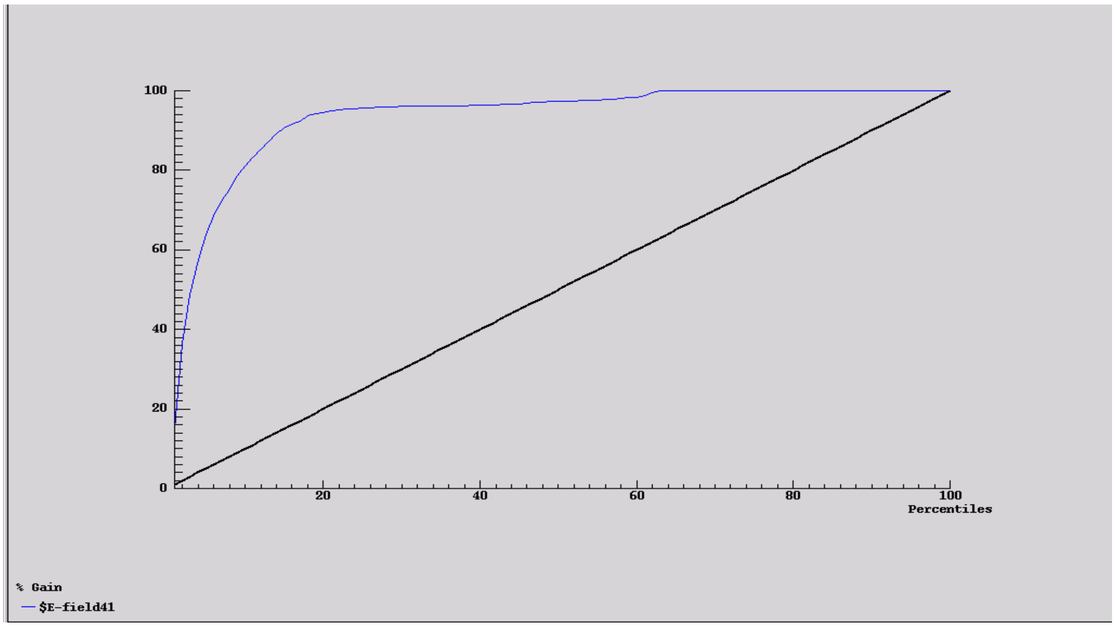


Fig 12: Gain Chart for FCIMPC. The gain chart shows the amount of variation in the data modelled for by percentiles of data trained.

Regression analysis gave interesting results. See tables 2, 3, and 6i, 6ii, 6iii. Also figures 4, 5, 6, 11 and 12. A check on the residuals vs. fits plot shows that the residuals were near enough to 'cloud like', so the model was reasonable. What possibly is not reasonable is the fact that the target variable is very skewed, to the extent that it would appear to have say a gamma distribution. (A transformation would possibly cure this problem, however interpretability, and being able to compare models is what is important here.)

That aside, it would appear that SAS EM gives a different output to Clementine 6.0. A look at the VIF's confirms that multicolinearity is indeed a problem, and the model was rerun several times dropping out the variables which were so obviously surrogates for other variables. Ultimately eleven of the input variables gave similar, although not the same results. The reason also for the discrepancy is that SAS EM DM Reg. uses an initial starting seed, and the 'regression' is done numerically (iteratively). This accounts for the slightly different results each time this is run (with a different seed).

Neural Networks

Output shows the greatest prediction accuracy for the neural network model with architecture 37 inputs, 3 hidden layers (21, 14, 8), and 1 output layer.

This gives an error rate of 0.01, in the case of SAS EM, or 99.95% predicted accuracy in the case of Clementine 6.0 (using the same architecture as SAS EM). Interestingly Clementine also offers the use of a filter based on the relative importance of inputs (sensitivity), and the user can prune these to a given percentage of the importance of the inputs, or a given number of the inputs, say the first 10 inputs. Rerunning the data after this filtering, still gave an accuracy rate of 99.94%, so very little was lost dropping off 27 of the input variables. However so comparable analysis could be made, the 37 input model was reported. Obviously prediction for new data would be very accurate from this model. Sensitivity analysis shows that the surrogate variable for age, amount of child care, scores highest but after that all the variables which relate to an unhappy childhood, and the parent (or caregivers) resenting the child, and stressing the family, not doing well at school all feature on this list.

However Clementine 6.0 gives a sensitivity analysis:

Table 8: Sensitivity (relative importance of inputs) For Neural Network model. (Clementine 6.0)

Field Number	Relative Importance	Name of Variable
field42	0.91102	Amount of child care
field29	0.37349	Does poorly at School, lies, cheats and doesn't sleep well
field27	0.29594	Contrast between feeling sad and inferior, and not getting along well with others, has no concentration

field26	0.26490	Unhappy Child, doesn't socialise well, feels sad depressed, worthless and inferior, acts young for his/her age
field6	0.20241	MKA resents child, feel they give up a lot for the child, Angry with child and that the child is difficult
field5	0.08364	Child knows a place they can get help
field31	0.03670	CPS family
field33	0.02658	Negative parent aggravation
field14	0.01146	Hours per week in group Child Care
field36	0.00346	Child mental health score, parent Aggravation

This output is useful as it gives some indication of what inputs are important, in training this data, for use with new data, later on.

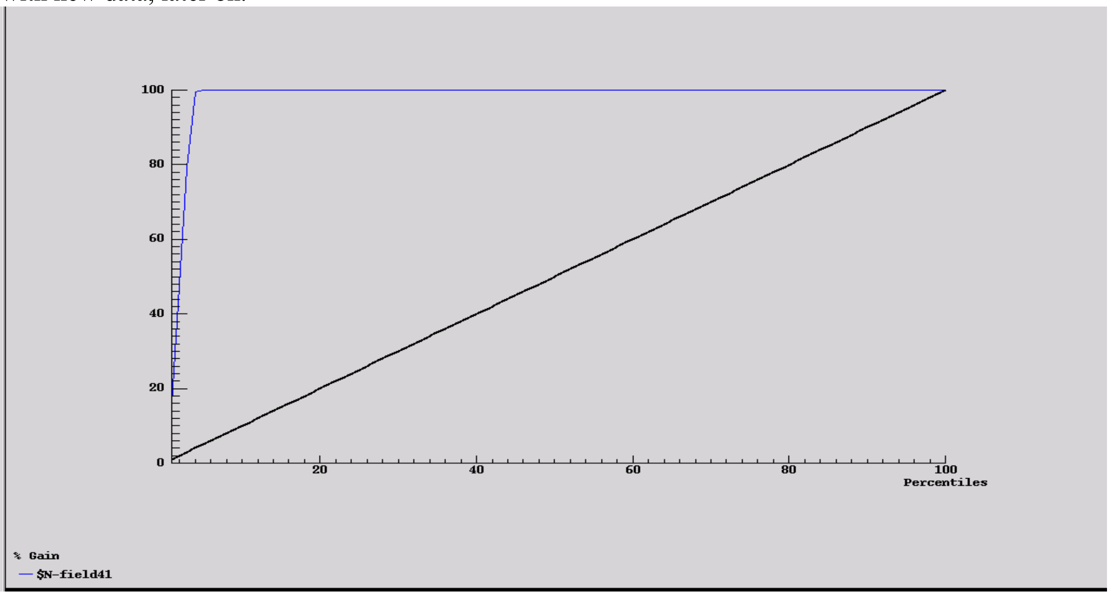


Fig 13: Gain Chart for NN FCIMPC the gain chart shows the model to be fitted after about 5 % of the data is trained.

Neural Networks

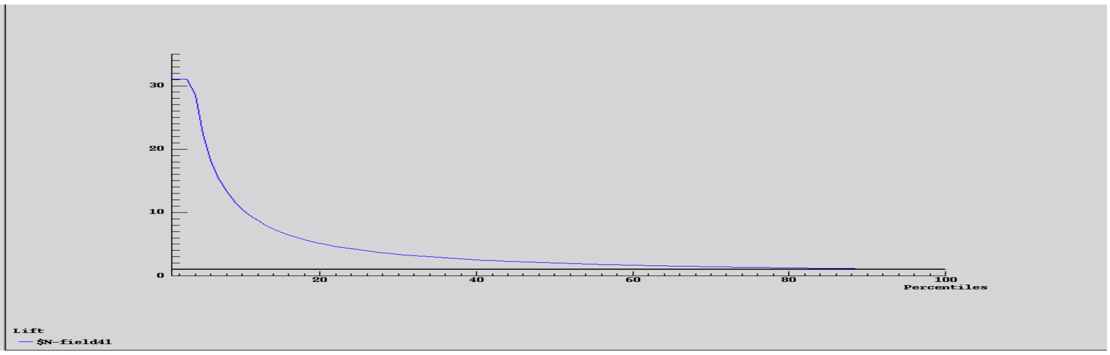


Fig 14: Lift Chart for NN FCIMPC the lift chart shows that 95% of the variation is explained after only 20% of the data is trained.

The tree models whilst very interpretable, give little in terms of prediction, and the Clementine tree carved the data up by amount of childcare only. This is essentially a surrogate for age, (as was stated before), and so not very helpful. The SAS version gave a little more detail, but again relied heavily on the variable AMOCHC, amount of childcare. Tree models included the child's feelings of being worthless and inferior, not getting along with others, feeling sad and depressed, and acting young for age, and having no concentration. It could be useful to prune the tree at this point. The tree is shown in Appendix 3.

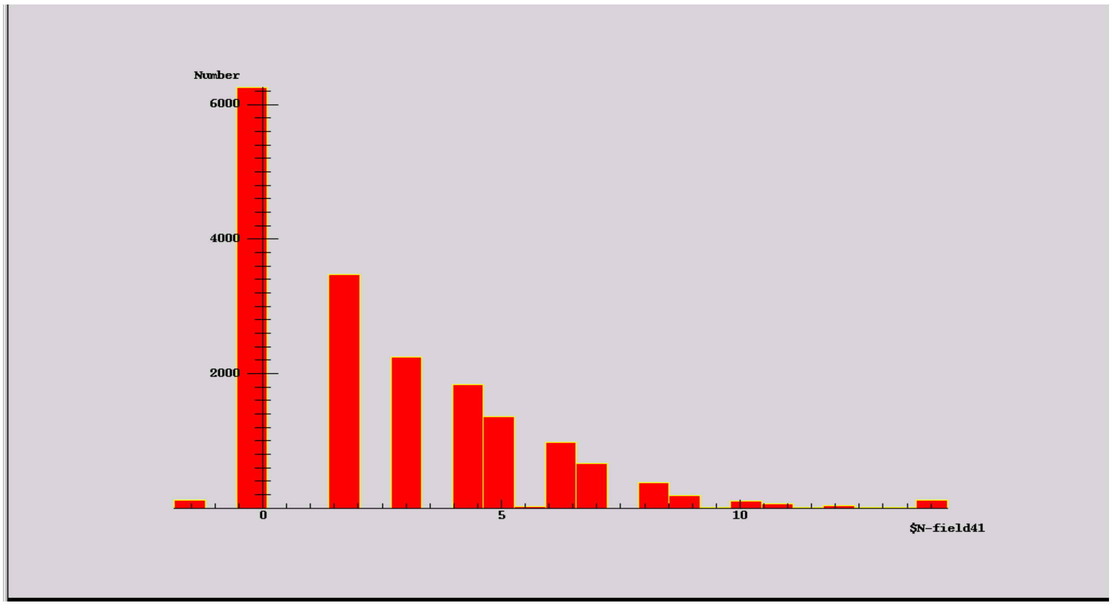


Fig 15: Predicted output by NN procedure

Clementine on the other hand cannot handle such a large data set (on this computer PII, with 256 Mg RAM). The regression node can only be run when the data has been sampled by 50%. Random sampling was selected here, so each time a slightly different result was obtained. This could have been sampled many times to give a 'bootstrap' effect, and probably given more time this would be a good idea. However regression was not overly accurate, (the error rate quoted is 2.303 (MSE), for SAS EM DM Reg., and 2.52 from Clementine). The Clementine model had a higher MSE, and a lower R^2 (amount of variation in the data described by the model) 67.7% -Clementine, and 72% for SAS EM. This is because variables with high VIF's were discarded from the model. Regression has however the advantage of describing how the input variables relate to the target variable. Common to both models were four input variables that contributed positively to the target score (increasing the likelihood of the child being a problem). These were in order of importance, Child does poorly at school, lies cheats and doesn't sleep well, followed by Child cared for in relatives home (rather than parents), Child had mental and dental health visits last year, and hours per week in group child care. There were 7 common negative influences (Influences negative only to the model; likely to make the child less of a problem), the variables are Child home alone whilst parent works, Family argues a lot and need help to get out of trouble, extent of poverty, MKA is angry with child and resents the amount of time spent with child, child's living arrangements, child elsewhere; not at school (This could be a surrogate for age), and child attended summer program.

The SOM's From SAS EM were in fact a clustering, giving 24 clusters. Most were overlaid on top of each other with differing amounts of variance. One cluster was very different to the rest. See fig. 8. A sensitivity (order of importance of inputs) is shown giving the 10 most important inputs. This is given in table 5. This does not say which variables increase the Child's problem score and which decrease it, but certainly give an indication of how strongly they affect it, with the variable UNHAPPY scoring much higher than anything else. The Clementine output proved uninterpretable with 137 clusters, while it did give an interesting pattern of output, was not very useful.

A comparison of the different models is given in Table 2. Gain and Lift Charts for regression and NN models, figs. 11, 12, 13, 14 show that 100 % gain was after the 5-percentile mark for NN, this is very accurate for prediction. Regression showed a 95% gain after the 20^{th} percentile. Lift charts show a remarkably similar graph, by the 20^{th} percentile, there is only 5% lift, as it is 95% trained. Figure 15 shows fitted target values from a NN, compare with Figure 1.

Factor Analysis, (see Appendix 5) The first 8 factors had eigenvalues greater than 1, and showed all but the first factor to be uncorrelated with the target variable. The first factor had a significant correlation with the target variable of -.682. This is comparable with regression analysis. So Factor 1 is a useful description of the input variables, which contrasts (The factor score decreases as the target variable increases) with the target variable.

The Independent Variables (FCIMPQ) (pruned)

Dataset 1 FCIMPQ SAS EM analysis included: neural networks, Kohonen SOM (This in both SAS EM, and Clementine), also a decision tree, a SOM (the output here being in the for of clustering) and various Neural Networks, with slightly different architecture. The SAS EM output is not altogether interpretable. Interpretation is done by means of an overall diagram, which links together different analyses, and then combines them into a report. However the most useful aspects are not always in the report, and it is important to follow links to get to the required information. Output is enormously copious.

An example of a SAS EM diagram for FCIMPQ is as follows:

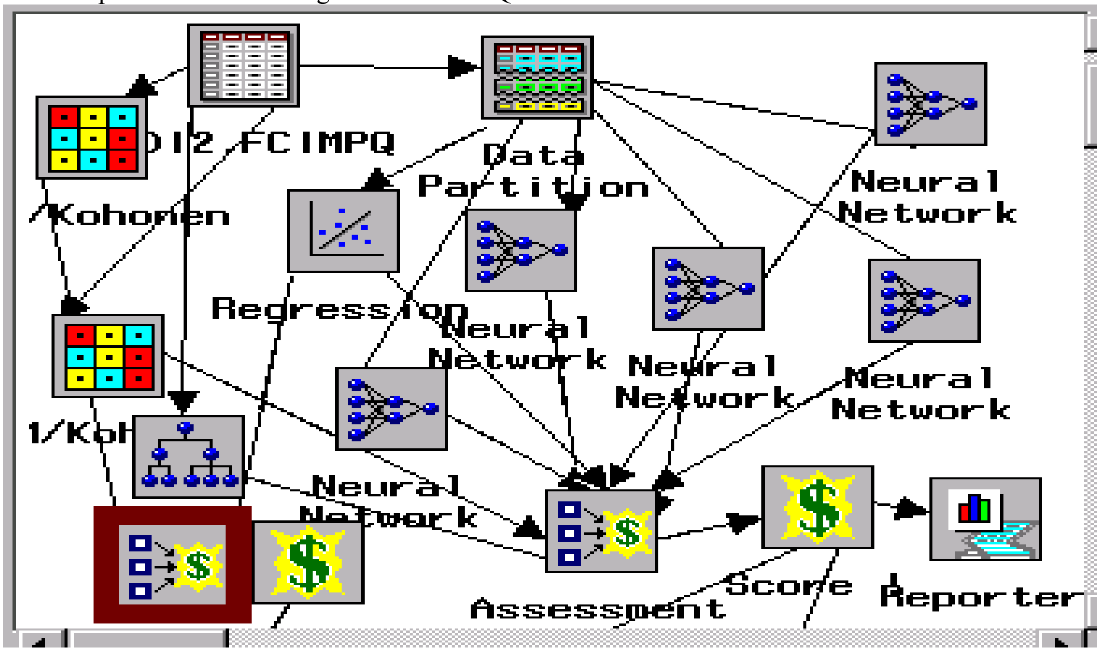


Fig 16: SAS EM Diagram for FCIMPQ. Each of the nodes was run as part of an overall analysis.

An example of a Clementine diagram for the FCIMPQ (pruned) dataset is:

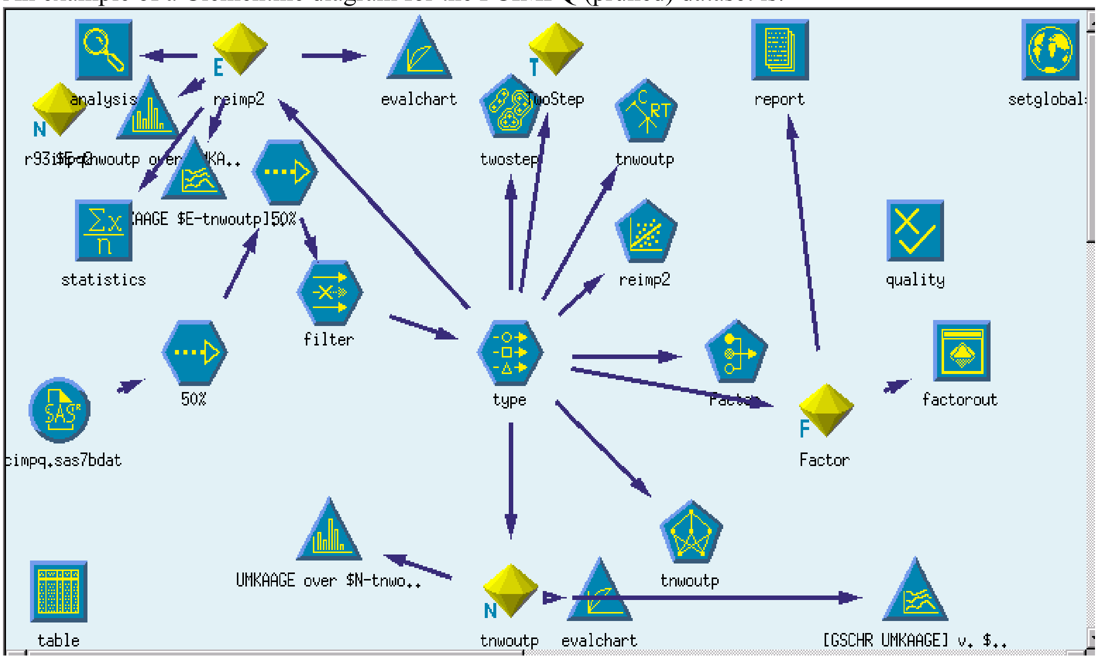


Fig 17: Clementine diagram for FCIMPQ.

Neural Networks

Many different architectures of neural networks were tried, and their error rates and the AIC, SBC, are listed in the table below.

Table 9: Error rates, AIC, SBC, for FCIMPQ: Comparison of Different Neural Network Architectures.

Architecture (Hidden layers)	Error rate (T) (Average Error)	Error rate (V) (Average Error)	Error rate (Test (Average Error)	AIC	SBC
21 14 5 (NN1)	2.80	2.77	2.64	24091.39	45788.78
21 14 6 (NN2)	2.79	2.75	2.62	24034.60	45856.73
21 14 7 (NN3)	3.01	3.00	2.87	25451.21	47398.08
21 14 8 (NN4)	**2.74**	**2.72**	**2.56**	**23779.13**	**45850.75**
21 14 9 (NN5)	2.80	2.78	2.63	24227.00	46423.36

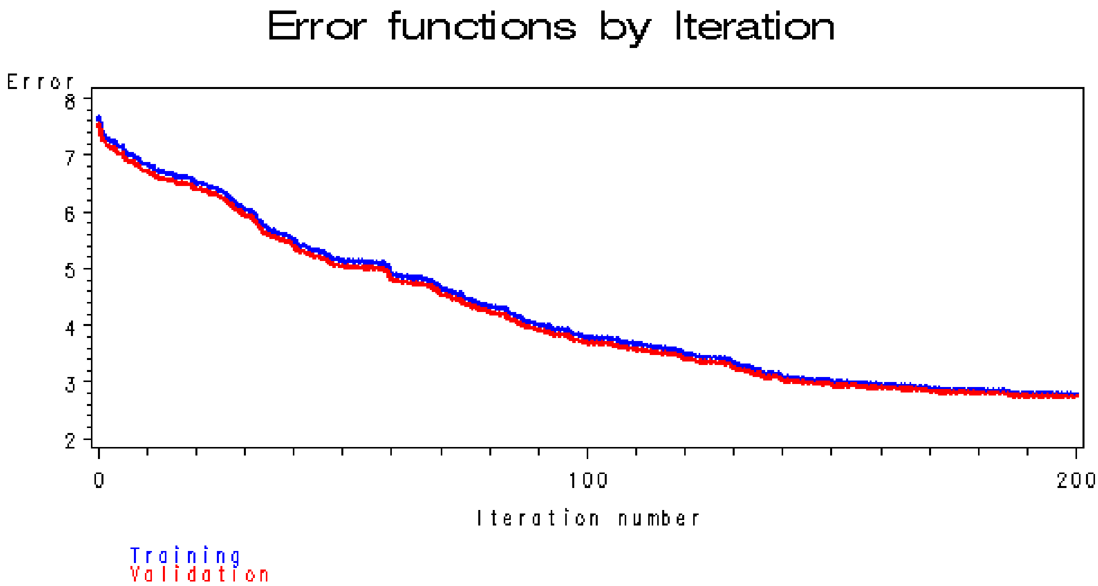


Fig 18: Neural Net 21, 14, 8 Architecture, for FCIMPQ Error rate by iteration. This shows that it took about 200 iterations to arrive at a stable NN model with minimum error. Also the two lines do not diverge, so the model is not over trained.

SOM For FCIMPQ

Table 10: SOM sensitivity output From SAS EM

Variable	Order of Importance	Value	Description
UMKAAGE	1	1	Age of most knowledgeable adult
UMEDULEV	2	0.64039	Most knowledgeable adult's highest level of education
UENG	3	0.60718	Child's engagement in school scale.

Table 10 gives the sensitivity from SOM modelling. Here the age of the MKA, and the level of education, as well as the Childs engagement in school are the most important inputs.

Table 11: Comparison of different models from SAS EM.

Model (Tool)	Error rate (T) (Average Error)	Error rate (V) (Average Error)	Error rate (Test (Average Error)	AIC	SBC
Neural Network	2.74	2.72	2.56	23779.13	45850.75
Regression	2.2990	2.3858	2.2044	15036.85	15644.97
Decision Tree	2.49	*	*	*	*

The comparison of models in Table 11, this time gives Regression analysis as the model with the smallest error, and the lowest AIC and SBC. Given the interpretability of Regression analysis, this clearly is the better model with this data.

Regression

Table 12: Regression Analysis of Effects (Type III SS)

Effect	DF	Type III SS	F-Value	Pr > F
BDISABL	1	1481.0002	800.7698	<.0001
BHLTHN	4	373.8538	40.6533	<.0001
BHLTHP	4	48.2666	5.2486	0.0003
CLGRAD	20	206.0079	4.4803	<.0001
FWELL	2	51.5107	11.2027	<.0001
FWHDEN	1	74.1371	32.2470	<.0001`
FWHMED	1	11.0604	4.8109	0.0283
GHCAR	5	65.8928	5.7322	<.0001
GHEADS	5	13679.3573	1190.007	<.0001
N4CPROBA	3	3452.2063	500.5291	<.0001
NARGUE	3	923.0732	133.8347	<.0001
NOACT	4	65.0878	7.0777	<.0001
NPCINTB	3	298.4293	32.4515	<.0001
NERVC	1	170.4822	74.1537	<.0001
NWORRYA	4	298.4293	32.4515	<.0001
NWORRYB	2	63.3478	13.7770	<.0001
UENG	1	6774.6396	2946.728	<.0001
UFAMSTR	4	160.8674	17.4929	<.0001
UMKAETH	1	59.5749	25.913	<.0001
USOURCE	8	38.0892	2.0709	<.0001

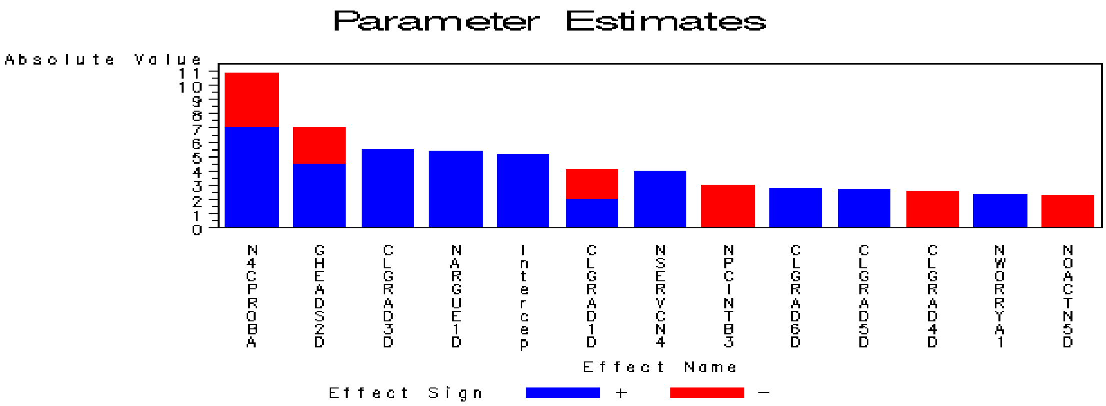


Fig 19: DM Reg. Estimates for FCIMPQ

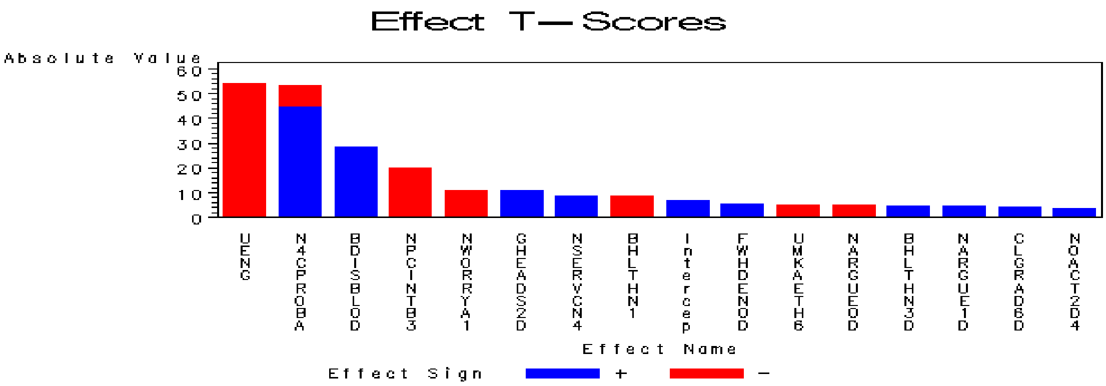


Fig 20: The Effect of the T scores

Table 12, Figures 19 and 20, give 1: Feels worthless or inferior, 2: Attended Head Start, 3: Know a place family can go if fighting, 4: Has a health condition that limits activity; as those increasing the likelihood of the child's problem score. On the other hand 1: Child's engagement in school scale, 2: Child really bothers MKA a lot, 3: Worry about keeping out of trouble; all prevent these kinds of problems.

Classification and Regression Trees

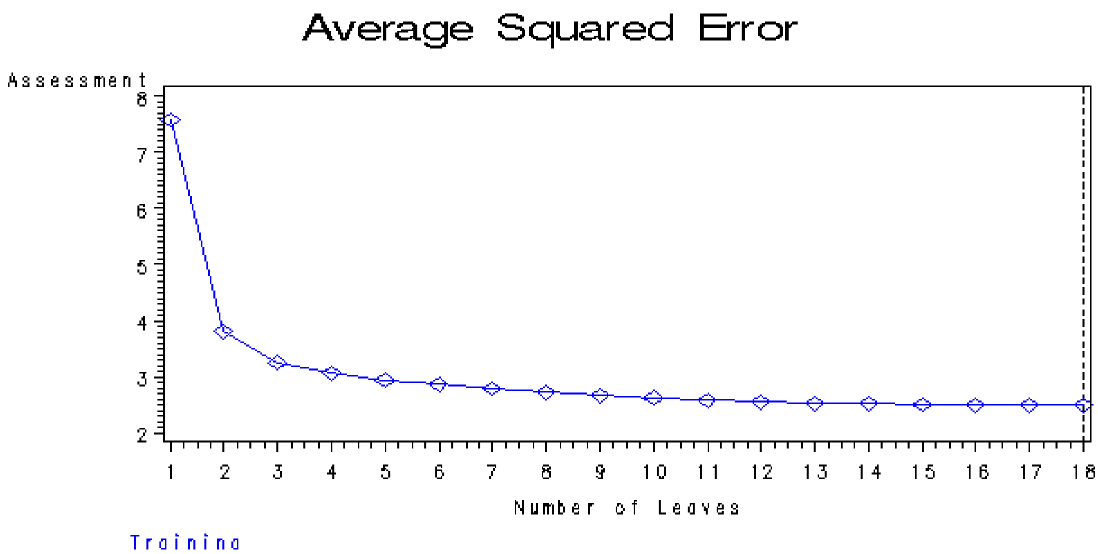


Fig 21: The Tree, number of leaves. The tree diagram is given in Appendix 4 (Fig. bi, Fig. bii, Fig. biii). By the first four splits much of the variation in the data has been described.

Regression

TABLE 13i: Regression output for FCIMPQ, summary of model

Model Summary Regression				
Model	R	R^2	R^2 Adjusted	S.E. of Estimate
1	.762(a)	0.581	0.579	1.7895

Table 13ii: Regression output for FCIMPQ: ANOVA Table

	ANOVA(b)				
Model	Sum of Squares	df	Mean Square	F	Sig.
Regression	21504.696	24	896.029	279.802	.000(a)
Residual	15521.9	4847	3.202		
Total	37026.595	4871			

Tables 13i and 13ii show that 58% of the variation in the data is explained by this regression model. Also the null hypothesis, that all slopes are zero, is very firmly rejected, with an F-value of 279.8. A discussion of table 13iii is given in the next section.

Table 13iii: Regression Coefficients for FCIMPQ

Regression Coefficients(a)					
Model	**B**	**Std. Error**	**t-Value**	**p > |t|**	**VIF**
(Constant)	-0.21800	0.863	0.801	*	*
CLGRAD	-0.08153	0.027	-3.045	0.002	1.070
UMKAETH	0.22400	0.081	2.762	0.006	1.051
FDENT	-0.06560	0.016	-4.069	0.000	1.918
FWHDEN	-0.40900	0.112	-3.660	0.000	1.027
NPCINTB	-0.18200	0.042	-4.309	0.000	1.105

NSERVC	-1.06400	0.065	-16.367	0.000	1.768
BDISBL	-1.37000	0.096	-14.246	0.000	1.152
BHLTHP	0.13300	0.038	3.515	0.000	1.053
BHLTHN	0.21000	0.032	6.660	0.000	1.134
GHEADS	0.83900	0.190	4.415	0.000	1.084
GHMWK	0.40100	0.060	6.667	0.000	1.198
GCENTR	0.64400	0.098	6.572	0.000	1.462
GEVSCH	-0.11600	0.012	-9.553	0.000	1.652
GHCAR	0.45500	0.012	37.053	0.000	3.844
GSCHR	0.15600	0.008	20.165	0.000	4.651
GSELF	-0.45900	0.119	-3.853	0.000	1.101
GWKSC	0.33500	0.090	3.712	0.000	1.447
HPARMAR	-0.03922	0.008	-5.157	0.000	1.064
N4CPROBA	1.71600	0.118	14.522	0.000	3.397
NOACT	-0.00762	0.001	-6.773	0.000	1.147
UENG	-0.17600	0.011	-15.466	0.000	1.894
NWORRYA	0.14800	0.008	17.656	0.000	1.876
NWORRYB	-0.79800	0.165	-4.841	0.000	1.089
NARGUE	-1.26700	0.119	-10.660	0.000	1.110
(a) Dependent Variable: tnwoutp					

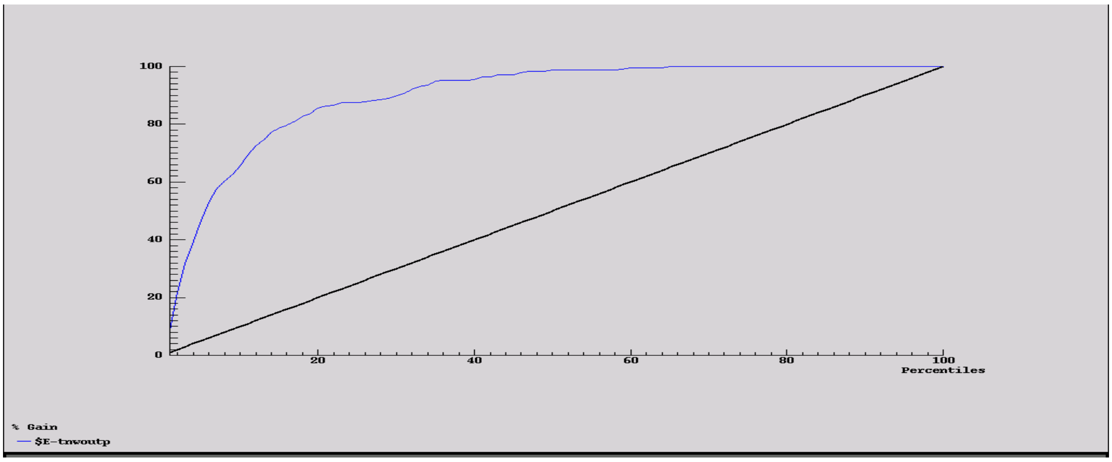


Fig 23 Gain Chart for Regression model FCIMPQ. Here 95% of the variation is described after 40 %of the data is trained

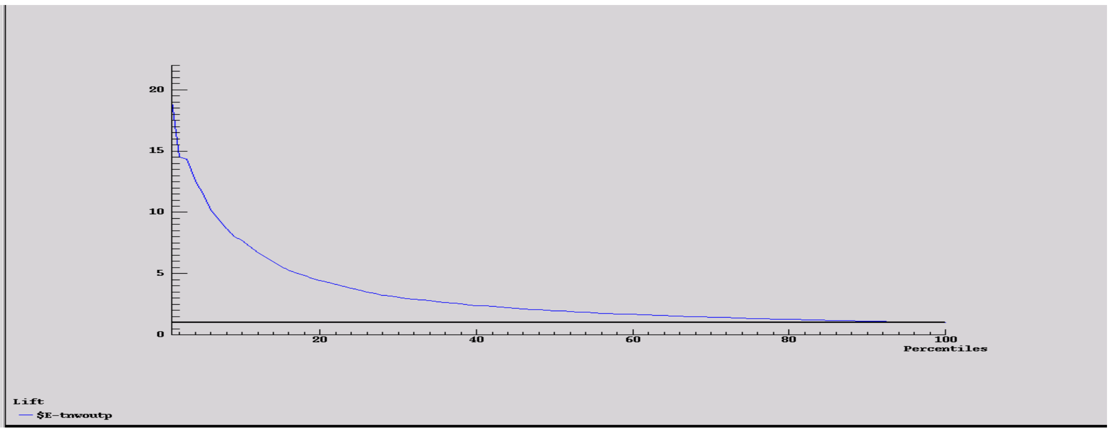


Fig 24: Lift Chart for Regression for FCIMPQ. Here the error is down to around 2.5 by the 40th percentile.

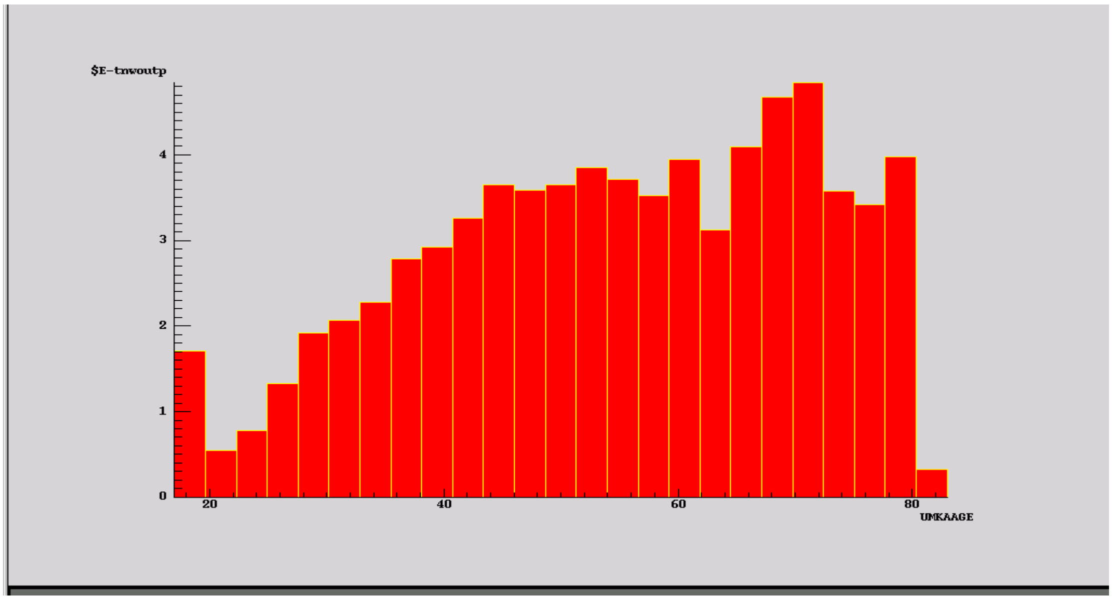


Fig 25: Bar Chart of fitted values (Regression) vs. Age of Most Knowledgeable Adult. MKA's in their 60's, 70's will have the most difficulty. This almost certainly represents children being brought up by grandparents. The exception to this trend, is if the MKA is less than 20, either a very young parent, or a sibling being the MKA.

Comparison of Clementine Models:

Table 14: Comparison of output from Clementine for NN and Regression

Model	Architecture	Occurrences	Predicted accuracy %
Regression	Maximum Likelihood	18000	58.1
Neural Network	37 input 21 HL1 14 HL2 8 HL3 1 output	17973	93.00

Table 15 shows the Neural Network to have the greatest predictive accuracy.

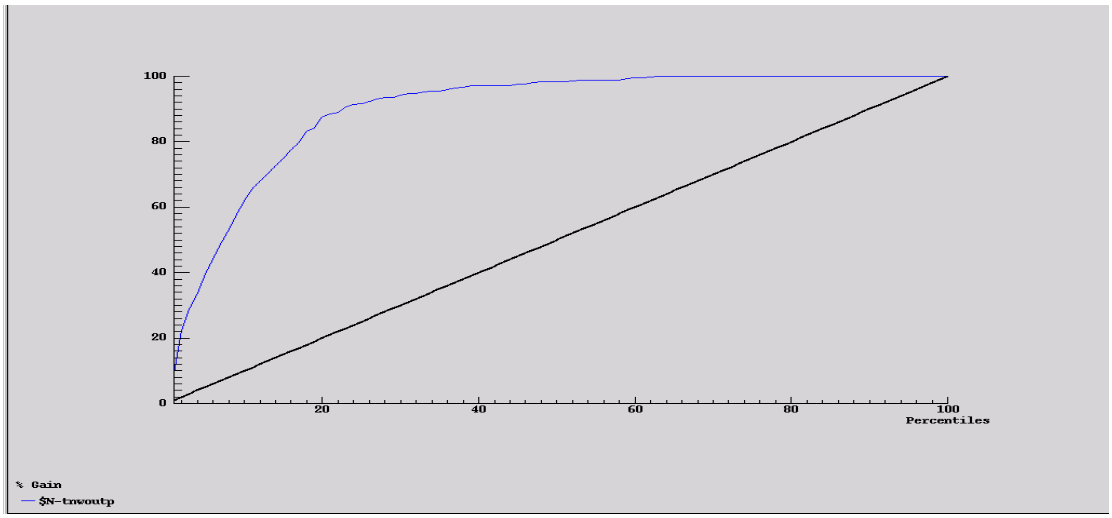


Fig 26: Gain chart for NN model, FCIMPQ. Here the 40th percentile achieves 90% predictive accuracy.

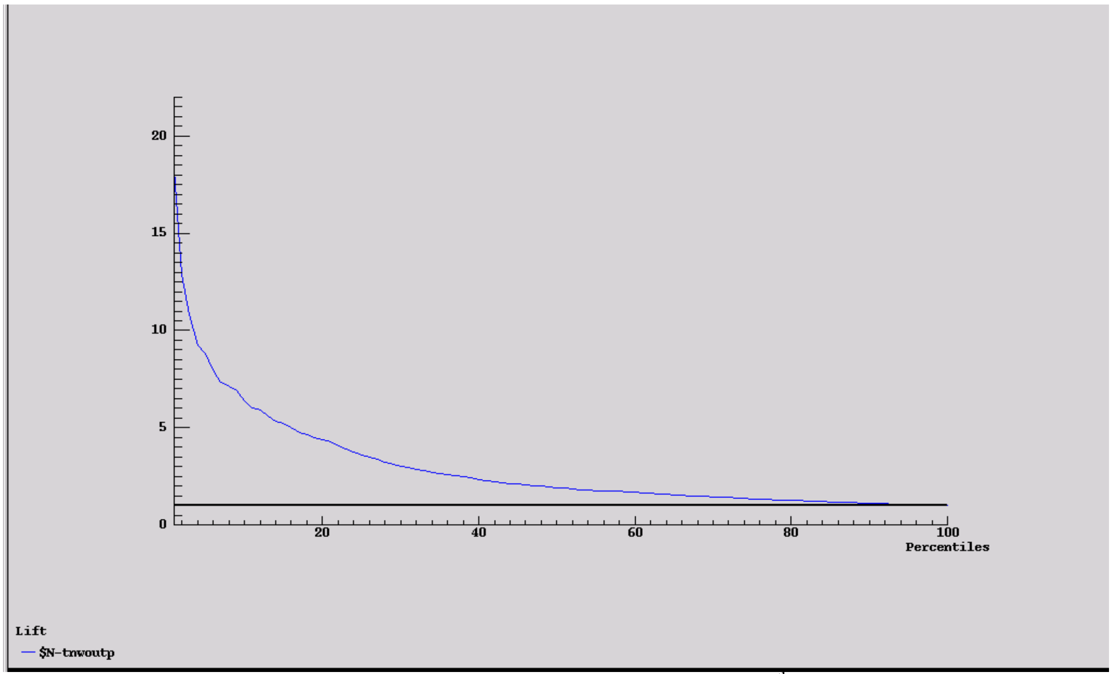


Fig 27: NN Lift Chart for FCIMPQ. Here the error rate is 2.5% by the 40th percentile.

Table 15 : Relative importance of different Inputs for the NN model for FCIMPQ.

Field no	**Ranking**	**Relative Importance of variable**	**Real Name**
N4CPROBA	1	0.21992	Feels worthless or Inferior
GSCHR	2	0.15746	Hours per week child in School
GHCAR	3	0.15526	Have child care in MKA's home
GCENTR	4	0.10160	Attended group care Centre
FDENT	5	0.09540	Dental visits last year
UENG	6	0.09393	Child's engagement in School scale
GHEADS	7	0.07301	Attended Head Start
BDISBL	8	0.06335	Has a health condition that limits activity
NWORRYA	9	0.05977	Worry about keeping out of trouble
NARGUE	10	0.05081	MKA and Children argue a lot
GSELF	11	0.04486	Child cared for self some time
BHLTHN	12	0.04181	Current Health Status
GEVSCH	13	0.03838	In School last four weeks
NWORRYB	14	0.03622	Tried to get help to keep out of trouble
BHLTHP	15	0.03320	Current health Status compared to twelve months ago
NOACT	16	0.03305	FC2 in organised activities in the past year
UFAMSTR	17	0.03204	Living arrangement of Children
UMKAETH	18	0.02775	Hispanic
HMBIO	19	0.02475	Child's mother lives elsewhere
NSERVC	20	0.02458	Knows a place family can go if fighting
FWHDEN	21	0.02188	Postponed dental care last year
UMKAAGE	22	0.01775	MKA's Age
GSUMWK	23	0.01530	MKA worked during summer program hours
FWHDRG	24	0.01329	Postponed drugs last year
GSCAR	25	0.01301	Hours per week Child in school

NPCINTB	26	0.01259	Child really bothers MKA a lot
UPRIMARY	27	0.01000	Primary CPS family indicator
UMEDULEV	28	0.00824	MKA's highest level of education
GHMWK	29	0.00670	MKA worked during care in MKA's home
CATTSC	30	0.00477	Attending Summer School
GSTOTH	31	0.00472	Child other place when away from home
FWHMED	32	0.00470	Postponed medical care last year
FWELL	33	0.00445	Well Child care last year
CLGRAD	34	0.00403	Current grade
USOURCE	35	0.00380	Usual source of care
GWKSC	36	0.00230	Weeks child in School while at home
HPARMAR	37	0.00176	Child's parents married when born

This data set is interesting if only because it relies on the original variables.
Analyses done included Neural Networks, Regression, Kohonen SOM, Decision Tree. The same target variable as used in FCIMPC was used.
Neural Networks: This again gave the same optimal network architecture of 37 input variables and three hidden layers consisting of 21, 14 and 8 neurons, and 1 output layer. This time Clementine gave a predicted accuracy of 93%, and SAS EM gave an error rate of 2.74, much higher than the data set selected by principal components. The AIC was 23779.13, significantly higher than that of regression. (See Table 11). The sensitivity analysis (See Table 15) showed no one variable describing much of the variation as in the principal components data set, however many of the same variables showed up here as did in the regression analysis.
SOM's This showed three variables (See Table 9) the age of MKA, The MKA's highest level of education, and the Child's engagement in school as having a strong influence.
Decision Trees were a little more illuminating for this data set. Here the tree had 18 leaves (SAS EM), and Appendix 4, (Figures bi, bii, biii) show the tree. Here the tree is very interpretable, with several of the variables being used to create the binary splits in the tree. Attending Head Start is the first split in the tree, next is the child's engagement in school, at the second level. The left hand side of the tree is split next by worry about keeping out of trouble, and the right hand side by health condition limiting activity. The parent aggravation scale score provides the next level in all parts of the tree, with 100-point mental health score, feeling worthless or inferior, and getting help because argue a lot, the next level. The final splits, all terminal nodes, show a variety of variables - child difficult to care for, child's engagement in school, health condition limiting activity, MKA and children argue a lot, and parent aggravation scale score. This is quite easily interpretable as before, but not so accurate in prediction.
Regression output is interesting because although multicolinearity is not a problem with this data set, still the two different software packages selected predictor variables. This was something of a mystery, and with unlimited time this would prove an interesting study. A possible reason is that Clementine samples the dataset (SRS), i.e. only 50% of the data points are included in the analysis. However this alone should not provide discrepancies of this order. SAS EM DM Reg. is not the usual least squares, but uses an iterative method, as described in the section on the FCIMPC dataset. Another reason for the discrepancy could be that SAS appeared to make factors out of what should have been ordinal variables, and more correctly treated as covariates, some interval variables were treated in this way (conversion to indicator variables, as would be correct for nominal variables). Another possibility was that on repeating this analysis with exactly the same inputs, and target variable, different results were obtained, this is indicative of using different seeds (starting values) and different sampling. All of these together suggest a 'bootstrap' approach might be best- Sample many times and aggregate the results. This should point more clearly to a global minimum error rate, as opposed to a finding many local minimum error rates for the error (as the surface being estimated is in fact very flat) depending on sampling and the seed chosen. Breiman (2001) describes this phenomenon, and suggests that if a large dataset is sampled, and a regression analysis performed, on subsequent sampling only 60% of the time will the same subset of predictors be selected.
Clementine gives an R^2 of 0.58, which means the model is describing 58% of the variation in the data.
Common to both Analyses are the variables (contributing to the child's problem score): 1: Current health

status compared to twelve months ago, 2: Attended Head Start, 3: Have child care in MKA's home, 4: Feels worthless or inferior, 5: Worry about keeping out of trouble; Variables that are preventing the child's problems (negative coefficients) are 1: Current Grade, 2: Postponed Dental care last year, 2: Child really bothers MKA a lot, 3: Has a health condition that limits activity, 4: FC2 in organised activities past year, 5: Child's engagement in school scale, 6: tried to get help to keep out of trouble, 7: MKA and children argue a lot. Also in the SAS analysis are 1: Well Child care last year, 2: Postponed medical care last year, 3: Living arrangement of children, 4: Usual source of care.
Additional to the Clementine analysis are 1: Dental visits last year, 2: MKA worked during summer program hours, 2: attended group care centre, 3: In school last four weeks, 4: Child attended before / after school care, 5: Child cared for self some time, 6: Weeks child in school while at home, 7: Child's parents married when born.
Fig. 27 shows a bar chart of the predicted score vs. age of the MKA, (Regression).

Conclusion
Data Mining is characterised by dividing the dataset into training, validation and test data sets. If the test data (and validation) data show an error rate similar to that of the training, the model is not overtrained, and can be considered a valid model. In each case after discounting surrogates for the child's age, (this was used t construct the target variable) generally what was left was variables relating to an unhappy child who felt worthless or inferior, whose MKA resented the child, and felt they gave up a lot for the child. however a model has been trained which can be used to classify new data, with 99.9 % accuracy for the FCIMPC data, and 93% accuracy for the original variables data. NN offers a high degree of prediction accuracy for new data that the older methods do not. The recurring theme that a child who is ignored by its parents is more likely to have problems is repeated throughout. Regression is still the best tool for describing the relationship between inputs and output, and a combination of these methods will produce the most interpretable and predictive model. It is however of concern to see different results from different packages, but the fact that Clementine 6.0 uses Least Squares Regression, whereas SAS EM DM Reg. is a Generalised Linear Model, (GLM's are found in Clementine 6.0, using the logistic node, but only for categorical outputs) goes some way to explaining this. It would be interesting to apply these trained models to New Zealand data.

Acknowledgement
I would like to thank Dr Barry McDonald for his helpful comments to improve this paper.

Glossary of Terms

AIC	Akiake Information Criterion (Akiake 1977)
Av Err	Average Error
Clementine 6.0	Clementine Data Mining Software (SPSS)
DM	Data Mining
DM Reg.	Data Mining Regression Procedure
FC	Focal Child
FC 2	Sibling of Focal Child
FCIMPC	Focal Child imputed dataset constructed by Principal Components.
FCIMPQ	Focal Child dataset imputed using PROC PRINQUAL
GLM	Generalised Linear Model
Input Variable	This is known as the independent variable in Classical Statistical language
MKA	Most Knowledgeable Adult
MSE	Mean Squared Error
NN	Neural Network
SAS EMSAS	Enterprise Miner Software (SAS Institute)
SBC	Schwartz Bayesian Criterion (Schwartz, 1978)
SOM	Self-Organising Map, or Self-Organising Feature Map (SOFM)
SRS	Simple Random Sampling
Target Variable	Also known as dependent variable in Classical Statistics
Tree	Decision tree, also known as Classification and Regression Tree
VIF's	Variance Inflation Factors (Montgomery, D.C, Peck, E.A., ;1982)

Bibliography

Akaike, H., (1974) A New Look at the Statistical Model Identification *IEEE Transactions on Automatic Control* **AC-19**, 716-723
Aleksander, I., Morton, H., (1990) *An Introduction to Neural Computing*. Chapman and Hall
Breiman, L. (2001) Statistical Modelling: The Two Cultures. *Statistical Science* **16** (3) 199-231
Berry, M.J.A., Linolf, G.S., (1997) *Data Mining Techniques*. Wiley
Berry, M.J.A., Linolf, G.S., (2000) *Mastering Data Mining.* Wiley
Groth, R., (1998) *Data Mining: A hands on Approach for Business Professionals* Prentice-Hall
Kaufman, L.., Rousseeuw, P.J., (1990) *Finding groups in data*. Wiley
Montgomery, D.C, Peck, E.A., (1982). *Introduction to Linear Regression Analysis*. Wiley
Schwartz, G.(1978) Estimating the Dimensions of a Model. *The Annals of Statistics* 6 461-464
Smith, K.A., (1999) *Introduction to Neural Networks and Data Mining for Business Applications.* Eruditions Publishing
SPSS (2001) *The C&RT Component*, SPSS Technical Report.
SAS (1994) PRINQUAL Procedure *SAS/STAT VOL II* SAS Institute
Westphal, C., Blaxton, T,.(1998) Data Mining Solutions. *Wiley*

Appendix 1: FCIMPC

Here the 37 variables selected by principal components are:

Field No.	**Variable**	**Real Name**
Field 3	RACE	Race (Ethnic origin)
Field 4	LGC	Last grade completed
Field 5	PMHELP	Child knows a place they can get help
Field 6	MKARES	MKA resents child, feel they give up a lot for the child, Angry with child and that the child is difficult
Field 7	MENDHTH	Child had mental or dental health visits last year
Field 8	NWELAT	Negative attitude to welfare
Field 9	ARGTRUB	Argue a lot and need help to get out of trouble
Field 10	WELCVD	Had well child visits to Doctor and Nurse
Field 11	ATTSS	Attended Summer School
Field 12	CCARR	Child Cared for in Relatives Home
Field 13	ASCC	Child Cared for in Own Home
Field 14	HWKGCC	Hours per week in group Child Care
Field 15	CNOTSCH	Child Elsewhere, not at School
Field 16	HOMALON	Child Home Alone whilst Parent Works
Field 17	SUMSCH	Child attended Summer program
Field 18	PMPSPLIT	Childs Father is Elsewhere
Field 19	MGONE	Mother lives Elsewhere
Field 20	CESTPAT	Court Established Father Supports Child
Field 21	DEFGBY	Does enough Homework to get by when Forced
Field 22	FAMATT	Amount child is taken out or read to by family members
Field 24	SUSPWK	Suspended or expelled in the past 12 months or works
Field 25	ATTDSCH	Attended School in the past 12 months
Field 26	UNHAPPY	Unhappy Child, doesn't socialise well, feels sad depressed, Worthless and inferior, acts young for his/her age
Field 27	CDEPWLS	Contrast between feeling sad and inferior, and not getting along well with others, has no concentration
Field 29	DEVIACH	Does poorly at School, lies, cheats and doesn't sleep well
Field 30	SIBSACT	Activities of siblings, sports lessons after school

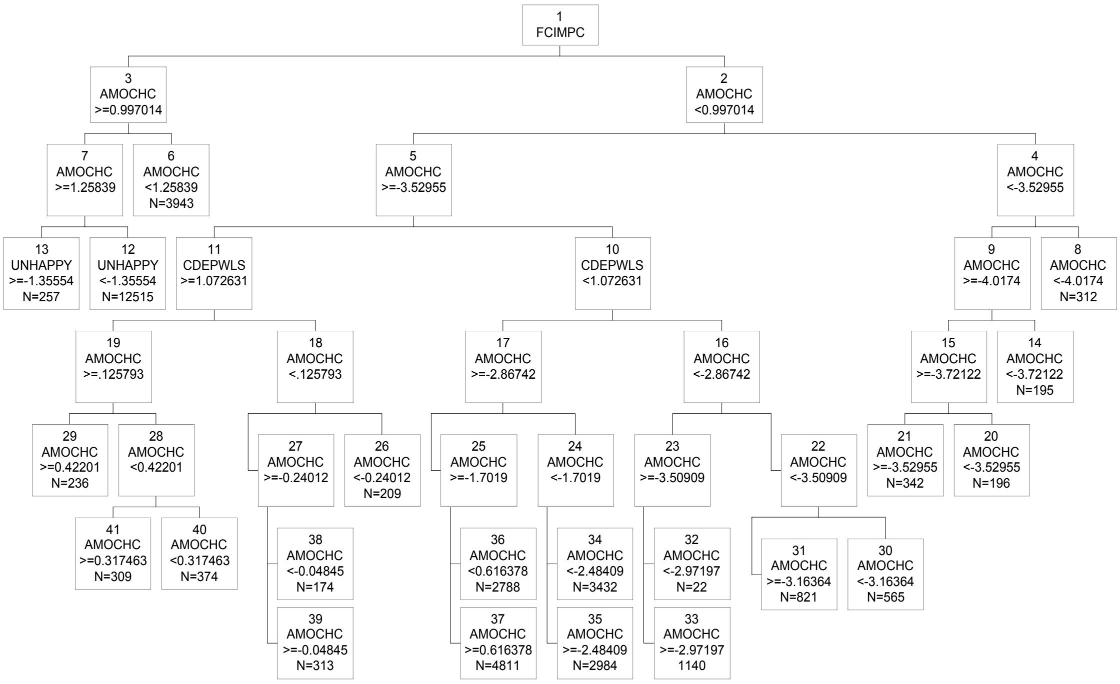
Field 31	CPSFAM	CPS family
Field 32	ACTPSEN	Child in other activity outside school
Field 33	NEGPAGG	Negative parent aggravation
Field 34	PPARAGG	Positive parent aggravation
Field 35	POVERTY	Social family income % poverty, CPS family income %poverty
Field 36	CMHPAGG	Child mental health score, parent Aggravation
Field 37	MKAED	MKA educational level
Field 38	MKAHEDG	Highest educational level and Age
Field 39	USOCNO	Usual source of care of child
Field 40	CHLIVAR	Childs Living Arrangement
Field 42	AMOCHC	Amount of child care

Appendix 2: FCIMPQ The 37 Variables Selected for Use by an Initial Regression Analysis

CLGRAD:	Current grade
UMKAETH:	Hispanic
FDENT:	Dental visits last year
FWHMED:	Postponed medical care last year
FWHDEN:	Postponed dental care last year
FWHDRG:	Postponed drugs last year
NPCINTB:	Child really bothers MKA a lot
NSERVC:	Know a place where family can go if fighting
BDISBL:	Has a health condition that limits activity
BHLTHP:	Current health compared to twelve months ago
BHLTHN:	Current Health Status
FWELL:	Well child care last year
GHEADS:	Attended head start
GHMWK:	MKA worked during care in MKA's home
GSCAR:	Child attended before / after school care
GCENTR:	Attended group care centre
GEVSCH:	In school last four weeks
GHCAR:	Have child care in MKA's home
GSCHR:	Hours per week child in school
GSELF:	Child cared for self some time
GSTOTH:	Child other place when away from home
GSUMWK:	MKA worked during summer program hours
GWKSC:	Weeks child in school while at home
HMBIO:	Childs mother lives elsewhere
HPARMAR:	Childs parents married when born
CATTSC:	Attending summer school
N4CPROBA:	Feels worthless or inferior
NOACT:	FC 2 in organised activities past year
UPRIMARY:	Primary CPS family Indicator
UENG:	Child's engagement in school scale
UMKAAGE:	MKA's age
UMEDULEV:	MKA's highest level of education
UFAMSTR:	Living arrangement of children
USOURCE:	Usual source of care
NWORRYA:	Worry about keeping out of trouble
NWORRYB:	Tried to get help to keep out of trouble
NARGUE:	MKA and children argue a lot

Appendix 3

Figure a**: Tree diagram for FCIMPC**



Appendix 4: Tree diagram for FCIMPQ

Fig bi: Diagram to show the Tree 'Rules'.
bi: Left hand side of tree

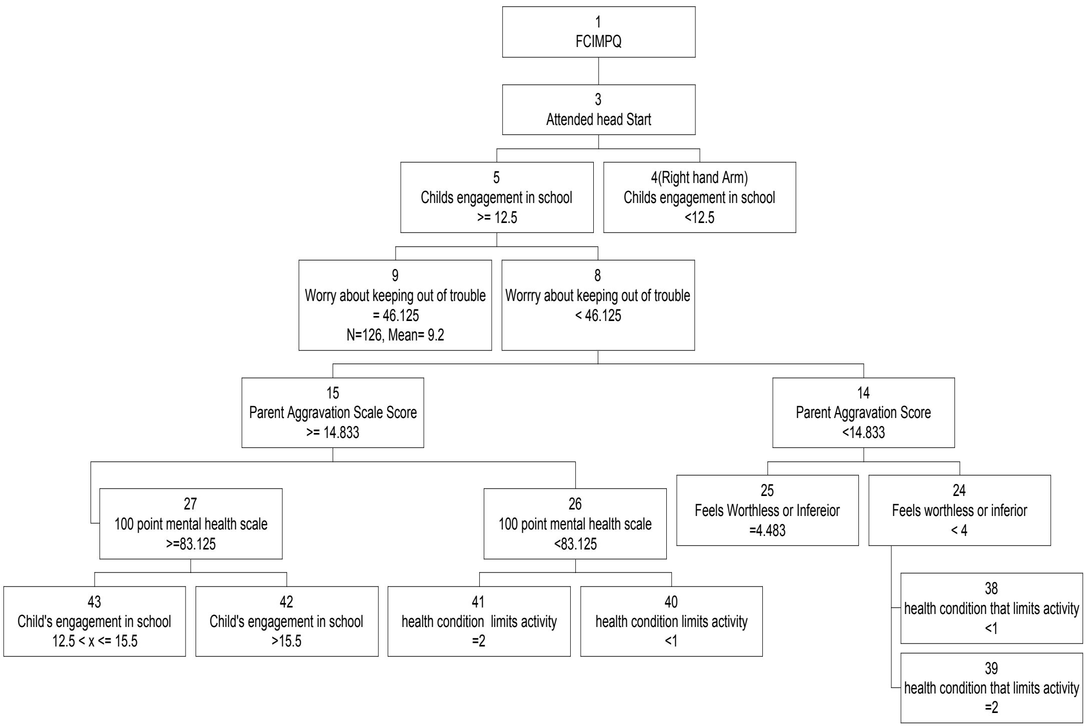
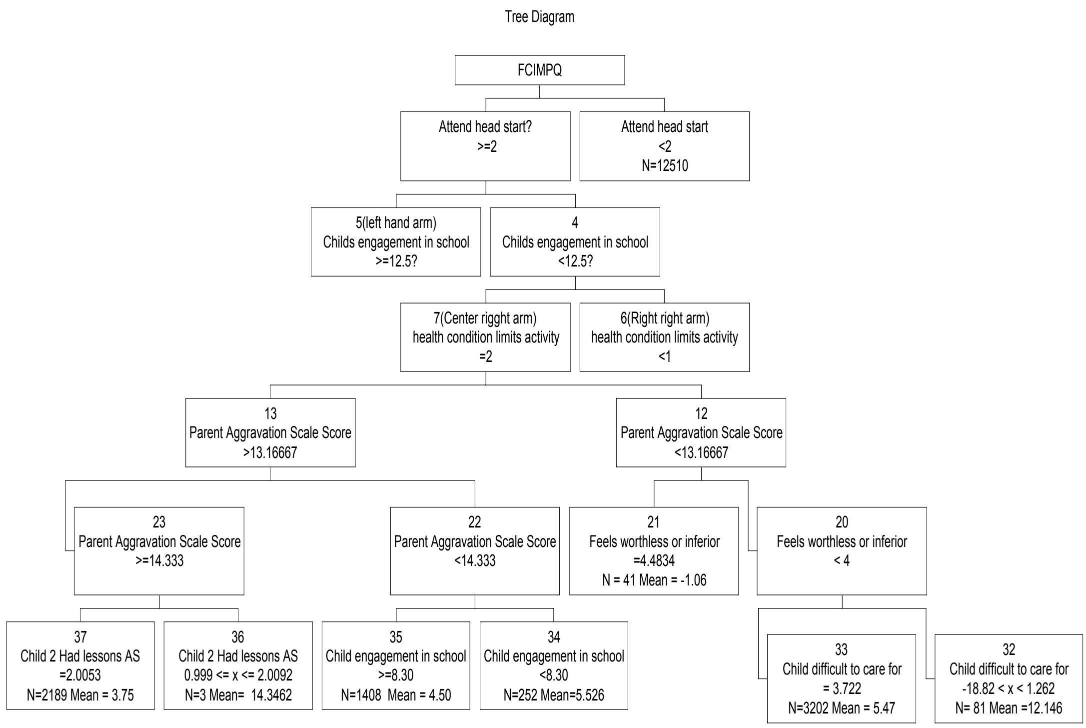
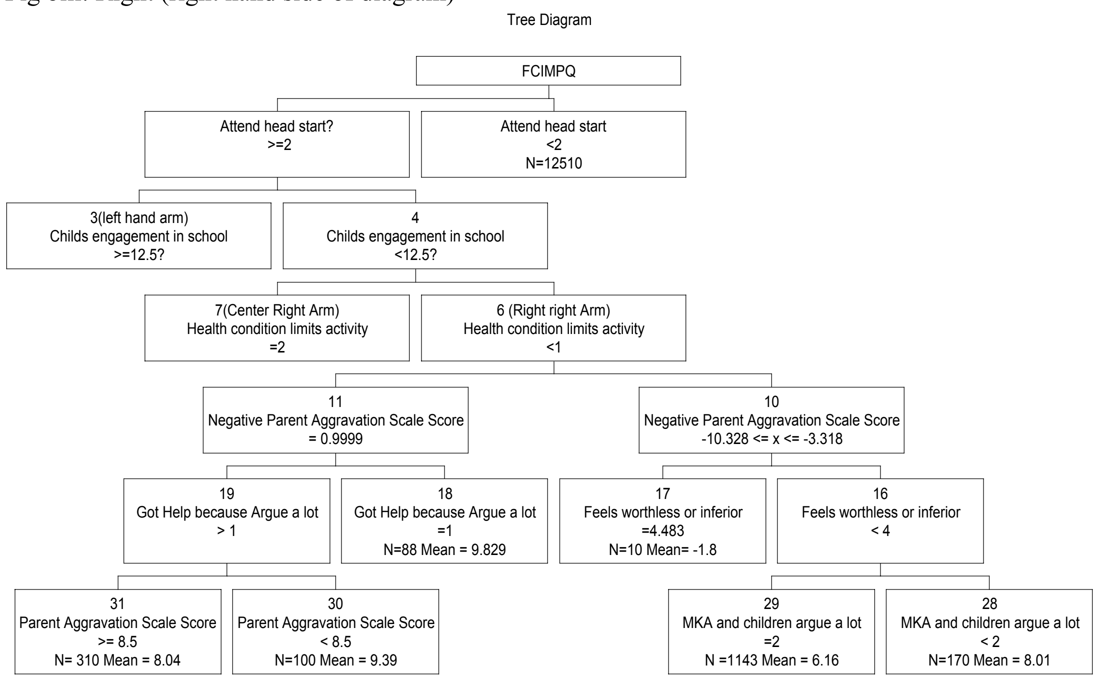


Fig bii: Centre right of tree diagram



This diagram joins on the left hand side to the right of the preceding diagram. Nodes follow the same numbering as the rules throughout. The following diagram joins on to the current diagram.

Fig biii: Right (right hand side of diagram)



This shows how the twenty-two leaves (terminal nodes) of the tree fit together, as per appendix 4. Also shown are the mean of each node, and the number of observations in each node.

CART output from Clementine:

GHEADS < 2.0 [Ave: -0.0, Effect: -2.681] (1658, 1.0) -> -0.0
GHEADS >= 2.0 [Ave: 4.064, Effect: +1.383] (3214)
  UENG < 11.5 [Ave: 5.516, Effect: +1.452] (913)
    NARGUE < 1.999 [Ave: 7.606, Effect: +2.09] (140, 1.0) -> 7.606
    NARGUE >= 1.999 [Ave: 5.137, Effect: -0.379] (773, 1.0) -> 5.137
  UENG >= 11.5 [Ave: 3.488, Effect: -0.576] (2301, 1.0) -> 3.48

Appendix 5: Factor Analysis FCIMPC

Equation for Factor-1:

0.000956 * field3 +
-0.000772 * field4 +
0.00332 * field5 +
-0.000451 * field6 +
0.05067 * field7 +
-0.003625 * field8 +
0.0149 * field9 +
0.004041 * field10 +
-0.009456 * field11 +
0.022798 * field12 +
-0.003236 * field13 +
0.035176 * field14 +
0.000452 * field15 +
0.011608 * field16 +
-0.001166 * field17 +
0.008358 * field18 +
0.003856 * field19 +
0.02589 * field20 +

0.056828 * field21 +
0.000242 * field22 +
-0.05641 * field24 +
0.020097 * field25 +
-0.036523 * field26 +
0.062892 * field27 +
0.050987 * field29 +
-0.06621 * field30 +
-0.071966 * field31 +
0.004913 * field32 +
-0.064595 * field33 +
0.003638 * field34 +
-0.014931 * field35 +
0.002677 * field36 +
-0.005233 * field37 +
-0.021583 * field38 +
0.004267 * field39 +
0.009627 * field40 +
0.058443 * field42 +
0.005231

Statistics for field : $F-Factor-1
Occurrences = 17998
Mean = -0.00051181

Correlation (Pearson Product-Moment) for field :
field41 = -0.682
(Strong negative correlation)