Paper 2601-2018

A Cautionary Tale of Using Machine Learning within Credit Risk in the Utilities Industry

Paul Malley, Centrica; Spyridon Potamitis, Centrica;

ABSTRACT

Machine learning is a scientific area that is constantly growing and enriched with evermore sophisticated and powerful algorithms. The question that arises is what is actually valuable in the business world and what algorithms can be applied in a business environment, such as Credit Risk, which is heavily regulated and interpretation of results constitutes a top priority? We will address this issues by showcasing a real-life business case and the algorithms that have been used to provide the most accurate results along with their advantages and disadvantages. In addition to providing an overview of the most popular machine learning algorithms that we use in SAS® Enterprise Miner™, we compare the algorithms' results with the incumbent methodology used within Credit Risk, namely Weight of Evidence (WoE) logistic regression. We identify the benefits that complex algorithms could bring to businesses, and in which business cases they could and should be implemented.

INTRODUCTION

Within the UK utilities industry, where energy is typically consumed by customers before payment for consumed energy is received, there is a need for a Credit Risk team assessing customer risk to mitigate against the risk of customers falling into arrears. In our experience with Credit Risk teams, the go-to approach to building predictive models usually relies on tried and tested methodology. One possible approach could involve Weights of Evidence logistic regression. The reasons for this include the direct impact the model may have on the customer and their experience with their energy company. The output of the algorithm may affect which payment method the customer may choose to use, for example by Cash/Cheque or Direct Debit. Or in the most extreme cases for B2B (Business to Business) customers, if a customer's risk of falling into arrears is too high, the business customer's application may be rejected.

Typically any processes or modelling will be subject to regulations from Ofgem (Office of Gas and Electricity markets), therefore interpretation of predictive modelling outputs and their impact on customers remains a key concern. Internal processes typically are in place for advanced analytics solutions to be presented and validated at Governance forums, and signed off by senior leadership and business stakeholders. Therefore interpretation of results is number one priority.

Within a Credit Risk environment, there may be use cases where more adventurous, more advanced learning techniques can be applied which could provide a significant uplift on the final results. By means of an example, we present a Credit Risk model built for B2B customers of an energy provider. Commercial customers have bigger bills than residential customers, so presenting greater risk of loss to the supplier. They also often have more complex reasons for, and consequences of, non-payment.. If a business customer fails to meet their payment due date, then they will typically proceed to a debt treatment path. The course of treatment on this path designed to help customers pay the outstanding amount may consist of a variety of activities, such as outbound dialing the customer, sending a reminder letter, and so on. Some customers often exhibit good behaviours such that no intervention or treatments are required to help them pay their energy bills if they have surpassed the payment due date. In this instance, we could describe it as "self-curing".

A Self-Cure model could be built to predict those customers who are likely to pay their outstanding bill with no treatment. How this model could be used is to exclude specific indebted customers from dialling activities for, say, a week into their treatment path, who are predicted to most likely self-cure without any effort from the point of view of the energy company. This is a great example where the model cannot negatively affect the customer and instead potentially may increase customer satisfaction by enhancing those more conscientious customers and their debt journeys, i.e. avoiding an unnecessary ‘chasing’ call from the energy supplier. Additionally, reducing a company’s number of outbound dialing can save on operational expenditure and contribute to a more efficient process. In this paper we will discuss the different machine learning algorithms that were used for predicting the self-cure outcome and compare the results with the more conventional Credit Risk modelling approach.

DATA AND METHODOLOGY

The model’s purpose is to identify customers that are going to self-cure by the end of the first week that they would typically enter the stage of the debt treatment path involving a debt outbound call. This, for example, may be Days 14-21 past their due payment date. Once we can identify which customers are going to self-cure or not, then we can use this to drive the appropriate method of treatment for each customer. The development data sample used for the modelling process selected all customers who were in debt and not dialled between Days 14-21 past due to their bill date. A self-cure flag was created for customers that have paid their balance in full during this period. This flag would serve as the target variable of interest. Also, internal and external data sources were used to bring together approximately five hundred potential input variables which formed the remainder of the development data sample. We then proceeded to use this data accordingly based on the following modelling methodology.

WOE LOGISTIC REGRESSION

As mentioned above WoE Logistic regression is typically a “go to” predictive modelling technique used in a Credit Risk environment. SAS Enterprise Miner includes a Credit Scoring tab where an accurate and robust model can be developed in some simple steps. One possible approach can be seen in the Enterprise Miner diagram as shown in Figure 1.

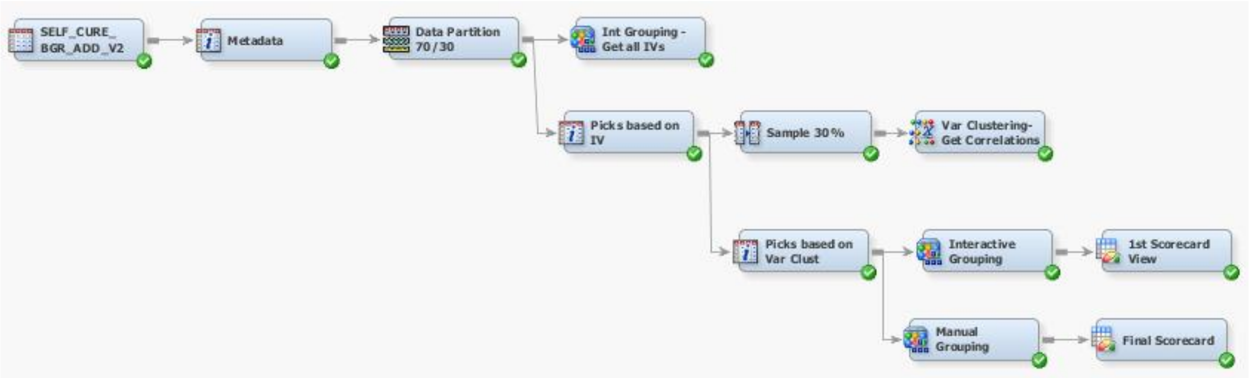

Figure 1: SAS Enterprise Miner diagram for WoE Logistic Regression

Understandably there are numerous ways to achieve an effective WoE logistic regression model within SAS Enterprise Miner, however here is our recommended approach:

- The development data sample is loaded into SAS Enterprise Miner, where we recommend to check that each variable has been loaded in the right format.
- A Metadata node is used to explore the data and remove any useless variables for predictive modelling purposes, for example variables with unary values, too many missing values, and so on.
- The next step is to split the data into conventional training and validation sets (this will help ensure later to establish whether or not we are not overfitting the model). How an analyst chooses to split their data, for example, including a test data set or not, is up to their discretion.

- We typically then use an Interactive Grouping node (available in the Credit Scoring tab within SAS Enterprise Miner) to get a preliminary view of the information value (IV) of each variable. Please refer to SAS guides for more details about information values if necessary. The variables with IVs greater than 0.05 are then selected in another Metadata node.
- To assist in further reducing the dimensionality of the data, we would recommend using a Variable Clustering node. Sampling may be required prior to running the Variable Clustering node as this will not run with more than 50,000 observations. Our methodology, that we have previously found to be practically very useful, is to select the variables from each resulting cluster with the highest IVs. The default chosen cluster representatives by SAS Enterprise Miner are not necessarily the variables with the greatest IV within each respective subset of clustered variables, and we find doing this helps optimise performance of the logistic regression.
- We then recommend running another Metadata node (to act as a decision point to aid future auditing of decisions made by the analyst) and Interactive Grouping node. It is here when we run a Scorecard node (available in the Credit Scoring tab) using default settings, to get an initial flavor, a sense check if you will, of the effectiveness of the chosen input variables.
- The next step involves running a final Interactive Grouping node, whereby we can perform manual edits to the groupings of the input variables, again at the discretion of the analyst. For example, sometimes the automated binnings of the variables may not make “business sense” which would be sensible to tailor accordingly.
- The final stage is to run a logistic regression using another Scorecard node. Model selection is at the discretion of the analyst, for example, stepwise selection or backwards selection and so on.

The final variables that were selected by the model along with their I.V. can be seen in table 1.

Variable Name	Type	Description	IV
Risk Score	Continuous	Refers to the risk score of a customer	0.34
Debt Treatment Stage	Ordinal	Historic maximum debt treatment stage reached	0.23
Payment Method	Ordinal	Customer’s preferred payment method	0.22
Contract Type	Ordinal	The customer’s current contract type	0.14
Contact History	Ordinal	Historic amount of customer contact	0.13
Debt Amount	Continuous	Historic view of debt amount owed by the customer	0.13
Debt Score	Continuous	External debt score	0.10

Table 1: Table showing the final variables in the WoE logistic regression with their data types and brief descriptions.

Our experience with WoE logistic regression is that it provides a relatively straightforward but powerful approach for modelling. This is because grouping the variables' values based on the weights of evidence we prevent extreme values and outliers having an impact on model performance and the predictors are all using the same scale. The final point to note is the process above may involve the analyst to run multiple iterations of the final model, to try different variable selection techniques (forward, backward, stepwise) and cut-offs in order to obtain optimal results.

ALTERNATIVE MACHINE LEARNING METHODS

Our recommendation to modelling a scenario, such as a self-cure model, with a binary target variable is to not limit the technique to the familiar logistic regression but to explore alternative machine learning techniques. Figure 2 shows a simplified SAS Enterprise Miner diagram summarizing our recommended approach to achieving this.

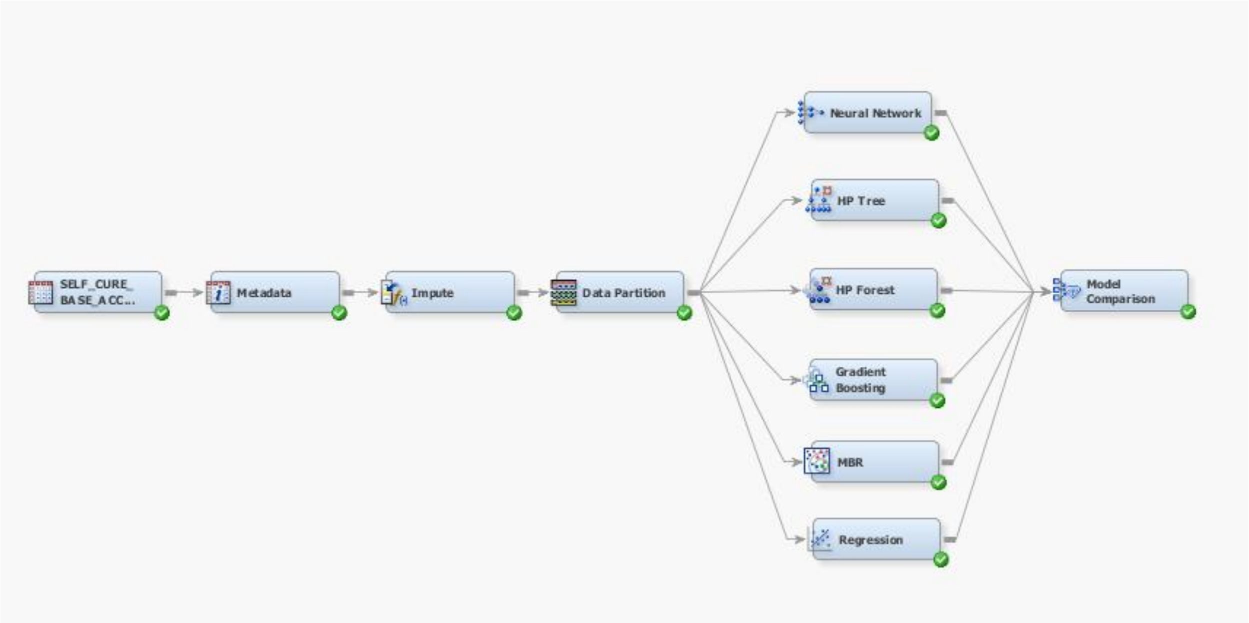

Figure 2: SAS Enterprise Miner diagram for alternative Machine Learning algorithms

A few comments to note:

- This time we have selected only the variables with the highest R-squared values from each cluster that were selected in the previous diagram.
- We then use an Impute node to impute missing values as some of the predictive models used are heavily affected by missing values in the data.
- The next step is to split the data into conventional training and validation sets.
- The sample data also included the WoE variables created in the previous diagram which were only used as inputs in the regression node so we could compare the results of the different methodologies using the model comparison node. The SAS Enterprise Miner nodes that were used, along with their tuning details, are described below.

MBR

The MBR node uses the KNN (k nearest neighbors) algorithm which measures the distances between observation X and the K observations around it and assigns X the most common label. In SAS Enterprise Miner Euclidian distances are used for measuring the distances and KNN turns out to be a very powerful classification algorithm despite its simplicity. However, KNN is very computationally intensive if the analyst selects to use its simplest form (implemented by the "scan" method in the node properties). Finally, small K tend to overfit and large K tend to underfit. The value of k was set to 150 (after experimentation) to avoid overfitting issues that we were facing.

Neural Networks

Neural Network can be a very powerful and "fast" model especially when we have to deal with large amount of data. It consists of the input layer, hidden layers and output layer, where each layer is made of interconnected nodes. Finally they use a system of weighted connections for the processing of the data [1].

It is important to mention here that tuning is a procedure of great significance in Neural Networks, to make accurate predictions. There are two ways to tune Neural Networks in SAS Enterprise Miner:

1. Trying different values manually and compare the results; or
2. Use the AutoNeural node in SAS Enterprise Miner.

In this project, we have not used the AutoNeural node in order to define the hidden layers that are required. Instead we experimented with different numbers of hidden layers and identified that six hidden layers provide the optimal results in this case. In addition, we have tried different architectures for Neural Networks and identified that the multilayer perceptron outperforms the others.

Neural networks are strongly affected by missing values and outliers. However, the problem of outliers is partly controlled by the hyperbolic tangent activation functions in the hidden units, which compress the extreme values between -1 and +1 [2]. Another way to increase the accuracy of Neural Networks is to reduce the number of inputs, which reduces the number of weights, thus possibly improving the performance of the model [2].

Decision Tree

Decision trees partition large amounts of data into smaller segments by applying a series of rules, which split the data into pieces until no further splits can occur [2]. By default, the variables that are used in each splitting process are the ones that provide the most information gain. Usually, for the splitting process to terminate, the user should define the maximum depth of the tree or the minimum number of observations that the tree could split on.

Another point we would like to make is that decision trees perform very well even on "dirty" data, as they can efficiently ignore irrelevant inputs and missing values. The only weakness that we have to take into consideration is that they do not have a built-in ability to treat redundant inputs. For us, this issue was tackled as we fed uncorrelated variables as inputs that were selected from the Variable Clustering node in the first diagram.

Finally, the model complexity is resolved by the "pruning" procedure, while we should adjust the maximum depth to avoid over-fitting. In order to fine-tune the model, we ran the model using different values and the optimal depth was found to be eight.

HP Forest

HP Forest uses the Random Forest algorithm which has many applications in the real world (for example, Microsoft Kinect), as it can produce very accurate predictions. The main idea behind the algorithm is to use multiple fully grown decision trees that do not produce similar results and, for the final prediction, take a majority vote.

The tuning of Random Forests is based on the number of trees used, the maximum depth that each tree could reach and the number of variables that each tree can consider in each split. The default options were used as they are tuned in a way that generally provide good results although there are macros that the analyst can use for different parameters to be tested and optimal tuning of the random forest. This methodology is discussed in more detail in “https://communities.sas.com/t5/SAS-Communities-Library/Tip-Getting-the-Most-from-your-Random-Forest/ta-p/223949 “ [3].

Gradient Boosting

Gradient Boosting machine is based on an algorithm developed by Jerome Friedman. Although it is an ensemble model which uses decision trees to make predictions, just like Random Forests, it’s main difference with Random Forests is that it is based on weak learners with high bias and low variance. Gradient boosting resamples the data several times and a tree is developed on the residuals of the previous tree, while in Random Forest decision trees are built in parallel. Finally the algorithm is less prone to overfitting but a lot of tuning is required to achieve the best results. The maximum depth of trees was set to four as there is no need for fully grown trees in gradient boosting. The idea is that each simple tree created aims to enhance the overall performance of the model by correcting the mistakes made by the previous trees.

RESULTS

In order to compare the results of the models a Model Comparison node was used in SAS Enterprise Miner. We recommend using the Gini coefficients in the validation sets as a key selection criterion. The ROC curves are automatically produced by the node for both the training and validation sets and are presented in Figure 3.

.

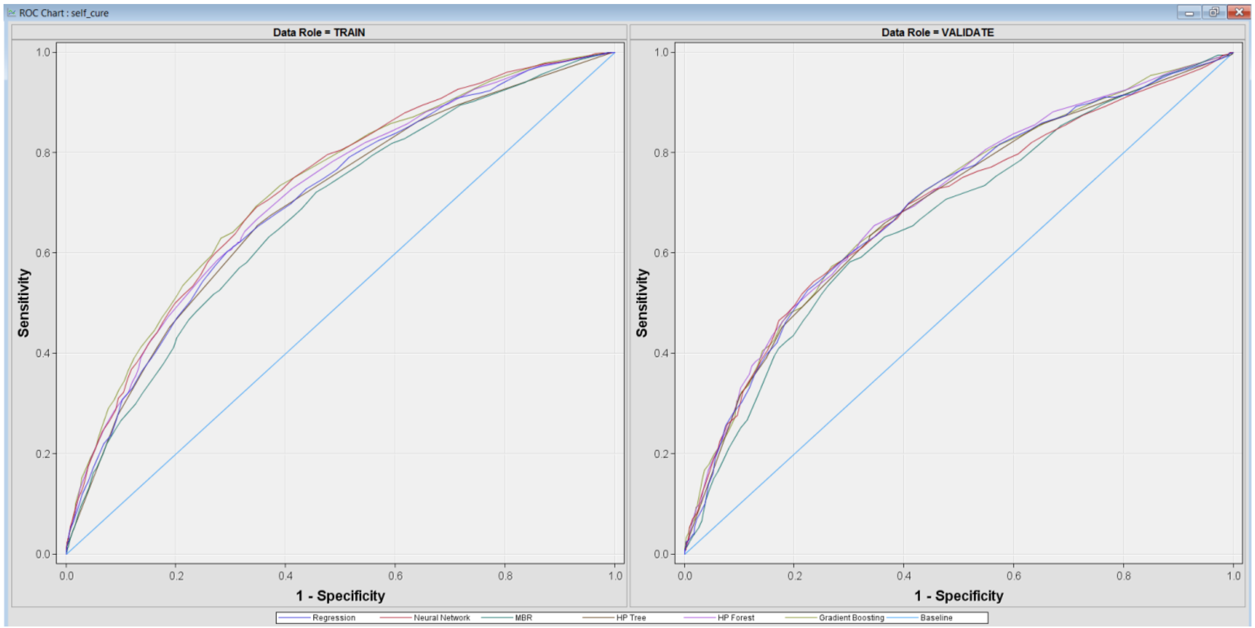

Figure 3: ROC charts taken from the Model Comparison node output

Table 2 shows the Gini coefficients that are achieved by the various models. Recall that Gini coefficient is a metric widely used in classification problems and is calculated using the Area Under the Curve (AUC) relating to ROC, specifically Gini = (2 * AUC) -1.

ML Node	Gini Coefficient
HP Forest	39.9%
Gradient Boosting	39.5%
WoE Logistic Regression	38.0%
HP Tree	37.9%
Neural Network	37.4%
MBR	33.3%

Table 1: Comparison table for the different ML algorithms

As observed in Table 2, Random Forest has achieved the best classification performance based on the Gini metric with an uplift of 5% against WoE logistic regression. It is important to point out that we could have further enhanced the predictability of the Random Forest by fine-tuning its parameters (using the methodologies that were described in the previous section) but we achieved our goal of proving that advanced ML algorithms can achieve better accuracy of results than WoE logistic regression without spending extensive time with such fine-tuning.

CONCLUSION

In this paper a real business problem regarding “self-cure” customers was discussed and a comparison has been made between different predictive modeling methods; the WoE logistic regression, which is widely used in Credit Risk area, and more advanced Machine Learning methods that are implemented via SAS Enterprise Miner. It was observed that although WoE logistic regression performs well and offers great interpretability, Gradient Boosting and Random Forests outperformed the conventional methods showing significant uplift which translates into tangible business benefit. On the other hand, more advanced Machine Learning algorithms risk issues of interpretability, they can make the implementation of these techniques challenging in heavily regulated environments like Credit Risk. Our recommendation is it would be preferable to take advantage of alternative Machine Learning techniques in cases where the most accurate classification is needed and there could be no negative impact in a direct decision made for the customer.

REFERENCES

[1] Wisconsin, U. of, 2015. No Title. Available at: http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html [Accessed April 12, 2015].

[2] Sas.mytraining, 2018. No Title. , p.https://support.sas.com/edu/viewmyelearn.html. Available at: https://support.sas.com/edu/viewmyelearn.html [Accessed March 24, 2015].

[3] Tip: Getting the Most from your Random Forest. SAS Institute, Brett Wujek

https://communities.sas.com/t5/SAS-Communities-Library/Tip-Getting-the-Most-from-your-Random-Forest/ta-p/223949

CONTACT INFORMATION

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

Paul Malley
paulmalley@hotmail.com

Spyridon Potamitis
sp.potamitis@gmail.com