Masters Research Report Title:	Modelling for the optimal product to offer a financial services customer.
Type of research report:	Mathematical Statistics
Candidate:	John Shingirai Mukomberanwa (Part-time student)
Student Number:	0417657j
School:	Statistics and Actuarial Science
Supervisor:	Professor David Lubinsky
Head of School:	Professor Peter Fridjohn
Date:	29 May 2014

Research report submitted to the Higher Degrees Committee, University of the Witwatersrand, in partial fulfilment of the requirement for the acceptance to the degree of Master of Science in Statistics by coursework and research report.

Declaration

I declare that this research report is my own, unaided work. It is being submitted for the degree of Master of Science in the University of Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university. I am fully aware that plagiarism is wrong and failure to adhere to the required conventions in referencing the thoughts and ideas of others may result in disciplinary action being taken by the university.

Signed on 29 May 2014

John Shingirai Mukomberanwa

Abstract

This study, illustrates how various statistical classification models can be compared and utilised to resolve cross-selling problems encountered in a financial services environment. Various statistical classification algorithms were deployed to model for the appropriate product to sell to a financial services customer under a multi-classifier setting. Four models were used, namely: multinomial logistic regression, multinomial bagging with logistic regression, multinomial random forests with decision trees and error correcting output coding. The models were compared in terms of predictive accuracy, generalisation, interpretability, ability to handle rare instances and ease of use. A weighted score for each model was obtained based on the evaluation criteria stated above and an overall model ranking thereof.

In terms of the data, banked customers who only had a transactional account at the start of the observation period were used for the modelling process. Varying samples of the customers were obtained from different time points with the preceding six to twelve months information being used to derive the predictor variables and the following six months used to monitor product take-up.

Error correcting output coding performed the best in terms of predictive accuracy but did not perform as well on other metrics. Overall, multinomial bagging with logistic regression proved to be the best model. All the models struggled with modelling for the rare classes. Weighted classification was deployed to improve the rare-class prediction accuracy. Classification accuracy showed significant limitation under the multi-classifier setting as it tended to be biased towards the majority class. The measure of area under the receiver operating characteristic curve (AUC) as proposed by Hand and Till (2001) proved to be a powerful metric for model evaluation.

Table of Contents

List of Figures

List of Tables

Notation and Terminology

AUC – Area under the receiver operating characteristic curve

MLR - Multinomial logistic regression

MBLR - Multinomial bagging with logistic regression

ECOC - Error correcting output coding

MRFD - Multinomial random forests with decision trees

BIC - Bayesian Information Criterion

AIC - Akaike Information Criterion

PCC – Proportional by chance correction

b.INV - Investment Product

c.SL - Secured Loan

d.UL - Unsecured Loan

e.CARD - Credit Card

NO_TAKE – No product take up

CART – Classification and regression trees

The following notation was used in the investigation and is applied consistently throughout the research report. Let:

- $J \text{ represent the number of categories in the dependent variable}, J > 2$
- $i - \text{represents the observation}$
- $n -$ number of observations
- $T -$ coding matrix
- $L - \text{ binary classification models used in}$ ECOC
- $K \text{ represent the number of independent variables}$

  Where: $X = (x_1, x_2, x_3, \dots \dots \dots .., x_K)^T \text{ is the vector of covariates}$

1 Introduction

1.1 Application of Classification Models in Financial Services Industry

In recent years, statistical modelling methods have found their way into the Direct Marketing and Customer Relationship Management (CRM) framework within large service providers. Applied Statistics have been used extensively in the behavioural and social sciences. With the advent of high powered computers and extensive data across different industries, some of these traditional techniques are now being used to explain behaviours of different groups of people. This is common in the financial services environment where millions of customer-initiated financial transactions are taking place (Prinzie and Van den Poel, 2006). Using advanced statistical modelling techniques, data can be used to understand customer behavioural patterns which thus assist in inferring customer needs and preferences. Empowered by this knowledge, strategic decision makers are able to formulate products and offers to the different sub-groups of customers.

Arguably, the most famous example of application of these techniques is the "customers who bought this product also bought..." section on www.amazon.com. This method is generally known as market basket analysis.

Market basket analysis has extensively been used in the retail grocery setting to optimally arrange products on shelves to maximise customer spend on any given visit. Burez and Van den Poel (2007) and Prinzie and Van den Poel (2006) are examples of studies which analyse customer purchase events to support CRM. Li, Sun, and Wilcox (2005) consider that at different stages of customers' demand, customers present different requirements which are derived in a particular product purchase sequence.

Recently the technique has been extended to financial services and it is premised that customers can be segmented into different behavioural groups based on their previous purchasing patterns and other data such as demographics. This allows the institution to predict the next likely product to be purchased by a customer. Hastie, Tibshirani and Friedman (2009) discuss the difficulty in motivating for such an approach based on previous purchasing patterns and deducing the reasons behind that behaviour without additional external information. They recommend the purchasing associations be used as input variables into a more comprehensive multivariate predictive model which is easier to justify based on results. One such instance is the use of traditional modelling techniques to classify a customer’s affinity to a certain product or need. For example, a customer’s affinity to purchase short term insurance.

In instances where one is modelling for a simple binary choice between accepting a specific service or product, the considerations for model development are much simpler; the most commonly used techniques being; logistic regression, decision trees and discriminant analysis. A complication arises if one has to model for the optimal product to sell a customer from a full complement of products offered by the financial services provider. Advanced statistical methods such as multi-classifier predictive modelling techniques as well as optimisation techniques are employed in order to ascertain optimality (Kamakura, Wedel, Rossa and Mazzon, 2003).

1.2 Aim of the study

The main objective of this study was to create a model which predicts the best product to cross sell to a customer with a reasonable degree of accuracy. In this study, we analysed banked customers’ purchasing patterns of banking products and services including transactional behaviour. This was done so as to identify cross buying patterns used in model development.

Since banks offer more than two products, a multi-classifier model was required. The models built had a nominal classification variable. For the purposes of this study, the classes of the response variable were restricted to five. Many multi-classifier models could have been used, but this study was restricted to selecting the best model from a choice of four statistical models of interest.

The classification methods were limited to the following four:

- multinomial logistic regression (MLR)
- multinomial bagging with logistic regression (MBLR)
- error correcting output coding (ECOC)
- multinomial random forests with decision trees (MRFD)

1.3 Structure of the Research Report

The research report is organised as follows. Chapter 2 gives a brief literature survey; this includes the theory behind the different techniques, varying sampling ratios, evaluation criteria, applications of the different models, and evolution of the techniques over time. The survey extends to review the comparative studies that have been done before on these classification methods. Chapter 3 outlines the methodology followed as well as the different assessment measures deployed. Chapter 4 details the model results and Chapter 5 concludes with a discussion and the model rankings.

2 Literature Review

This section provides a comparison of the different modelling techniques which are available in the literature. It provides an overview of the different modelling techniques, the theory behind the different statistical techniques used in this paper, different sampling issues, evaluation criteria and a marketing perspective to data mining classification.

2.1 Overview of the different classification methods

In traditional applications of statistics, the most commonly used methods for estimation and prediction have been techniques such as discriminant analysis and least squares regression. Classifiers are algorithms that discriminate between classes of patterns. In classification related problems, discriminant analysis and logistic regression have been applied extensively in cases related to binary classification (Rao, Solka and Wegman, 2005). An example would be classifying patients as having a specific disease using varying factors as input measures. An extension to binary classification is multi-classification. In a multi-classification model, the classification variable has more than two categories. The categories can either be classified as ordinal or nominal. A model with an ordinal classification variable implies that there is some form of order attached to the different classes whereas the nominal classification variable would have no quantifiable order. MLR is a popular modelling method for multi-classifier scenarios.

MLR assumes independence of classes within the classification variable. It also assumes non perfect separation of the predicted outcomes otherwise unrealistic parameter coefficients will be estimated (Rao *et al*, 2005). The major limitation of MLR has been the lack of convergence in the model using the maximum likelihood method. The inability to easily find a stable and robust model has been another issue of concern (Schafer, 2001).

Another form of MLR which is extensively used in multi-classifier scenarios is to run multiple binary regression models using one category as a reference category (Schafer, 2001). This approach is generally applied if the classification variable can be arranged into a sequence of binary choices.

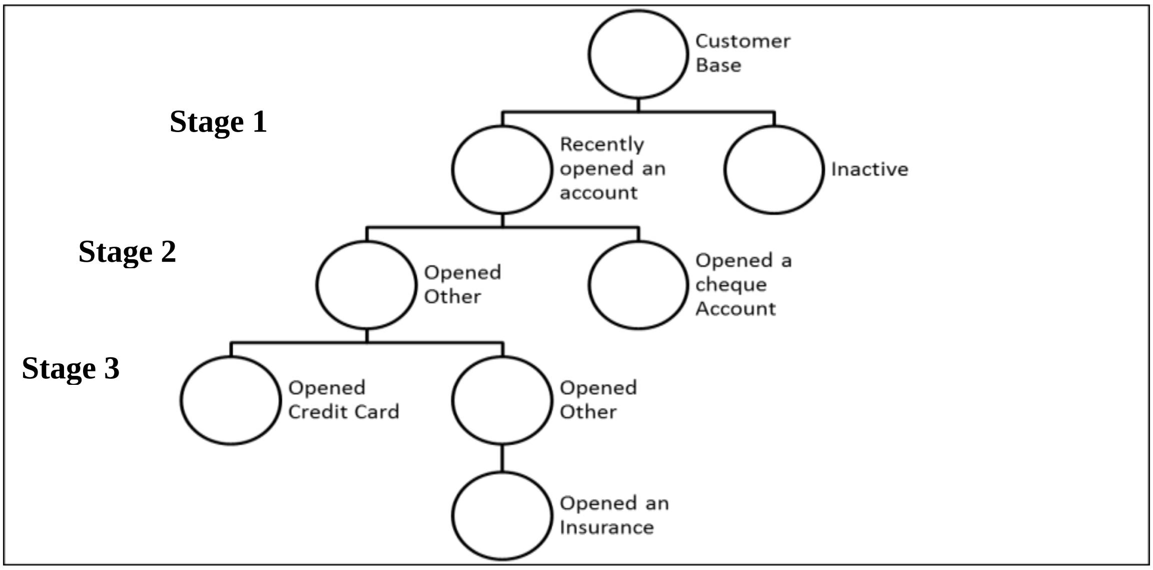

Figure 1: Illustration of a multi-classifier problem using a sequence of binary models

An example of this is shown in Figure 1; Stage 1 is modelling on all customers and computing the log-odds of "**Recently opened an account**". Stage 2 models only for those that recently opened a product where the log odds of opening a cheque account is compared to other products opened. Stage 3 models only for customers that did not open a cheque account but another secondary product. It compares the log odds of customers that did not open a cheque account but opened a credit card compared to those that only opened an insurance product. In this type of model, the overall maximum likelihood cascades into the three individual likelihood functions created by the three stages of binary classification (Schafer, 2001).

Tree-structured classification models are alternative algorithms that are not constrained by the assumptions of normality and homogeneity of variance. "*Unlike other classification methods such as nearest neighbour method and kernel density estimation they produce predictors which are simple functions of the input variables and thus easy to use.*" (Cios, Kurgan, Perdrycz and Swiniarski, 2007). Cios *et al* (2007) further state that this was the main reason the trees became popular. Tree-structured classification models have evolved significantly since the 1960's when they began as automatic interaction detection (AID) through to the eighties where Breiman, Friedman, Olshen and Stone (1984) developed classification and regression trees (CART), which is a complex algorithm of fitting trees to data.

CART models are easy to fit and interpret but are very unstable. A small change in the training sample data can lead to significant deviation in classification results. Breiman *et al* (1984) and Cios *et al* (2007) attribute the instability to the hierarchical nature of the tree derivation process. The effect of a choice in the top split is cascaded to all the splits below it, which intuitively makes sense. They further state that this can be corrected by choosing a more stable split criterion but one cannot remove the inherent instability of the algorithm.

Another issue with CART models is non-optimality. Since they use a *greedy algorithm* (Hastie *et al*, 2009), a tree will split on a variable which reduces impurity the most or that which provides the highest information gain. However, the variable chosen might not necessarily give the optimal model at the end. Cios *et al* (2007) state that the CART modelling approach sacrifices optimality in exchange for computational efficiency.

Bagging which derives its name from a technique called bootstrap aggregation is a technique proposed by Breiman (1996a). It is a model averaging technique which is used to improve model stability and predictive power. It is important to note that this technique is not model specific and can be easily applied to methods such as CART and logistic regression. Bagging is further discussed in section 2.2.3.

Random forests attempts to improve the bagging methodology by building a large complement of de-correlated trees and then averaging them. De-correlated trees imply greater independence between the trees and thus leading to error reduction. Random forests attempt to reduce the variance of the trees through decreasing the correlation of the trees by randomly selecting the input variables (Hastie *et al*, 2009). Random forests obtain a class vote for each tree and classify according to the majority vote. Various authors commend the technique for its superior performance, in addition to not requiring much tuning (Hastie *et al*, 2009).

ECOC is another technique used to model multi-classifier problems. The approach it uses can be classified into two stages. The first involves developing a “coding strategy”, which is to develop a series of de-correlated classifiers commonly referred to as “weak learners”. These weak learners attempt to explain different aspects of the data being modelled. The second stage involves developing a “decoding strategy”, which relates to developing a voting mechanism to assign predicted classes based on the outcomes of the weak-learners. Various coding and decoding strategies exist with common examples being exhaustive techniques and the Hamming's distance (Dietterich and Kong, 1995).

2.2 Theory of the different modelling techniques and parameter estimation

2.2.1 MLR

MLR is an extension of binary logistic regression which allows for more than two categories in the dependent variable. An illustration of the model is shown below:

- Obtain a data sample of size M and divide it into N sub populations each of varying sizes n_i such that
$$\sum_{i=1}^{N} n_i = M \qquad (1)$$
- Let J represent the number of categories in the dependent variable, $J > 2$
- Let K represent the number of independent variables
Where, $X = (x_1, x_2, x_3, \dots \dots \dots .., x_K)^T$ is the vector of covariates
- Let Y be the dependent variable with $y_{ij} = 0$ for all i besides one j with $y_{iJ} = 1$
- Assume the responses are nominal

The response for row *i* is:

$$y_i = (y_{i1}, y_{i2}, y_{i3}, \dots \dots \dots .., y_{ij})^T$$

is assumed to have a multinomial distribution with index $n_i = 1$ and parameter

$$\pi_i = (\pi_{i1}, \pi_{i2}, \pi_{i3}, \dots \dots \dots .., \pi_{ij})^T$$

By choosing the baseline category to be J, the model can then be written as follows:

$$\log\left(\frac{\pi_{ij}}{\pi_{i\mathrm{J}}}\right) = \sum_{k=0}^{K} x_{ik}\beta_{kj} \text{ ,for } j \neq \mathrm{J} \qquad (2)$$

Solving for one π_{ij} obtains:

$$\pi_{ij} = \frac{e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}}{1+\sum_{j=1}^{J-1} e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}} \qquad j < J \qquad (3)$$

$$\pi_{iJ} = \frac{1}{1+\sum_{j=1}^{J-1} e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}} \qquad (4)$$

The equations listed above were obtained from (Schafer, 2001; Böhning, 1992).

Parameter Estimation

a) Binary logistic

$$\hat{\beta} = \begin{pmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_K \end{pmatrix} \qquad (5)$$

The aim of the model is to estimate the $(K + 1)$ β parameters. This is done with maximum likelihood estimation by attempting to find the set of parameters for which the probability of the observed data is greatest. The maximum likelihood equation is derived from the probability distribution of the dependent variable. Since each y_i represents a binomial count in the i^{th} population, the joint probability density function of Y is:

$$f(y|\boldsymbol{\beta}) = \left[\prod_{i=1}^{n} \frac{n_i!}{y_i!(n_i-y_i)!} \pi_i^{y_i}(1-\pi_i)^{n_i-y_i}\right] \qquad (6)$$

And it follows that the likelihood function is represented as follows:

$$L(\boldsymbol{\beta}|y) = \prod_{i=1}^{n} \frac{n_i!}{y_i!(n_i-y_i)!} \pi_i^{y_i}(1-\pi_i)^{n_i-y_i} \qquad (7)$$

This can be written as:

$$L(\boldsymbol{\beta}|y) = \prod_{i=1}^{n} \left(\frac{\pi_i}{(1-\pi_i)}\right)^{y_i} (1-\pi_i)^{n_i} \qquad (8)$$

Since:

$$\pi_{ij} = \frac{e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}}{1+\sum_{j=1}^{J-1} e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}} \qquad j < J \qquad (9)$$

It can be shown that the equation above can be simplified to:

$$L(\boldsymbol{\beta}|y) = \prod_{i=1}^{n} \left(e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}\right)^{y_i} (1 + e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}})^{-n_i} \qquad (10)$$

Equation (10) is the kernel of the likelihood to maximise. The equation can also be simplified by taking its natural logarithm.

$$l(\beta) = \sum_{i=1}^{N} y_i \left(\sum_{k=0}^{K} x_{ik}\beta_k\right) - n_i . \log(1 + e^{\sum_{k=0}^{K} x_{ik}\beta_k}) \qquad (11)$$

To maximise this likelihood function, the derivative equation (11) is set to zero and solve.

b) MLR

For each population, the dependent variable follows a multinomial distribution with J levels. Thus, the joint probability density function is:

$$f(y|\boldsymbol{\beta}) = \prod_{i=1}^{N} \left[\prod_{j=1}^{J} \frac{n_i!}{y_{ij}!} \pi_i^{y_i} (1 - \pi_{ij})^{n_i - y_i}\right] \qquad (12)$$

To maximise (12) with respect to β, the factorial terms that do not contain any of the π_{ij} terms can be treated as constants. Thus, the kernel of the log likelihood function for MLR models is:

$$L(\boldsymbol{\beta}|y) \cong \prod_{i=1}^{N} \prod_{j=1}^{J} \pi_{ij}{}^{y_i} \qquad (13)$$

It can be shown that the likelihood function can be simplified to the equation below (Czepiel, 2002);

$$l(\beta) = \sum_{i=1}^{N} \sum_{j=1}^{J-1} \left(y_{ij} \sum_{k=0}^{K} x_{ik}\beta_{kj}\right) - n_i . \log(1 + \sum_{j=1}^{J-1} e^{\sum_{k=0}^{K} x_{ik}\beta_{kj}}) \qquad (14)$$

The problem then becomes that of finding the values of β which maximise the above likelihood function. This can be achieved by applying the Newton-Raphson algorithm making use of the first and second derivatives of the likelihood equation.

The equations listed above were adapted from Czepiel (2002).

Variable Selection

There are varying ways with which MLR model selects the variables to be used for the model fit process. Backward, forward, stepwise and best subset selections are the most common (Georges, 2004).

2.2.2 Decision trees

Decision trees belong to the general family of CART models first introduced by Breiman *et al* (1984). A decision tree is a non-parametric algorithm which models a dataset by recursively partitioning the dataset using variables which explain the most information. A criterion is specified for selecting and ordering the variables which are deployed into the recursive partitioning algorithm. The order of these variables change at every partitioning node as other variables become less or more relevant. A decision tree can either have a categorical or continuous dependent variable. A decision tree with a categorical dependent variable is called a classification tree, whereas one that has a continuous dependant variable is called a regression tree (Cios *et al*, 2007).

The recursive partitioning technique forms subgroups of data which are generally homogenous in terms of the distribution of the dependent class. Each of these subgroups is allocated a predicted response based on the distribution of the response class within that subgroup. New observation are categorised into one of these subgroups based on their independent variables and are assigned the predicted response of the subgroups they resemble the most. In short, a decision tree is a collection of data partitioning rules. English partitioning rules can easily be formulated from a decision tree (Hastie *et al*, 2009).

Generally, there are three main stages involved in growing a decision tree; growing a large tree, then pruning the tree to the optimal size, the final stage involves evaluating the predictive performance of the tree (Hastie *et al*, 2009).

Decision trees are difficult to handle as their size grows since concerns such as over fitting become more pronounced. Pruning techniques are generally applied in order to manage the size of the tree. Pruning is a good way of solving over fitting without significantly affecting the quality and the accuracy of the model. Hastie *et al* (2009) state that pruning can either be classified as post pruning or pre pruning. They describe pre pruning as stopping the algorithm for further splits if no significant additional information gain is achieved, whereas post pruning is tree size reduction of an already fully established tree.

Variable Reduction

To measure the variable that will give the most discriminatory feature, there are various measures. Two commonly used functions are Entropy and the Gini index.

1. Entropy

The entropy function is defined as:

$$Entropy(S) = \sum_{j=1}^{J} -p_j . \log_2(p_j) \qquad (15)$$

Where p_j is the proportion of the data belonging to the j^{th} class. In order for this measure to be directly comparable with other measures it can be redefined as,

$$Entropy(n) = \sum_{j=1}^{J} -p(j|n) . \log_2 p(j|n) \qquad (16)$$

Where $p(j|n)$is the proportion of data belonging to the j^{th} class at node n.

The variable that reduces the entropy the most at each subsequent node is used as the splitting variable. Information gain which measures expected reduction in the entropy caused by knowing the value of variable F_k is used to obtain this variable:

$$Information\ Gain(S, F_k) = Entropy(S) - \sum_{v_j \in V_{F_k}} \frac{|S_{v_j}|}{|S|} . Entropy(S_{v_j}) \qquad (17)$$

Where V_{F_k} is a set of all possible values of variable F_k and S_{v_j} is a subset of S, for which variable F_k has value v_i .

"*Information Gain tends to show some level of bias for cases with multiple outcomes, Gains ratio is then employed to compensate for this bias*" (Cios *et al*, 2007); this is defined as:

$$Gain\ Ratio(S, F_k) = \frac{Information\ Gain(S,F_k)}{Split\ Information(S,F_k)} \qquad (18)$$

Where:

$$Split\ Information(S, F_k) = \sum_{j=1}^{J} \frac{|S_j|}{|S|} . \log_2 \left(\frac{|S_j|}{|S|}\right) \qquad (19)$$

Split information is the entropy of S with respect to values of variable F_k. In a scenario where two or more variables have the same value of information gain, the variable that has the smaller number of categories is selected (Cios *et al*, 2007).

2. Gini index

Another measure for impurity is through the use of Gini index. Gini index which is also known as population diversity is defined as:

$$Gini(n) = 1 - \sum_{j=1}^{J} p^2(j|n) \qquad (20)$$

The Gini measure of a node is the sum of the squares of the proportions of the classes.

All the measures indicate the same trend, namely: they show decreasing values when data become more homogenous after performing a split on a variable that most reduces the "chaos" in the data (Cios *et al*, 2007).

2.2.3 Bagging

When discussing multivariate models, Hastie *et al* (2009) refer to the need to maximise the performance of the multivariate predictive model by minimising the prediction error. The prediction error can be split into three major components, namely:

- Irreducible error - This is the error relating to the variance of the predicted classifier.

- Statistical Bias- This is the squared difference between the true class and the predicted class across the whole dataset. It measures the level by which the average predicted estimate differs from the true mean.
- Variance – This is the expected squared deviation of the predicted class around its mean. In most instances, variance increases with model complexity (Hastie *et al*, 2009).

Dietterich and Kong (1995) illustrate that model performance is vastly improved by minimising the statistical bias and variance. They show that decision trees have low statistical bias but high variance. MLR has low statistical bias and slightly lower variance but is constrained by the assumption of linearity which brings in further bias into the model.

Hastie *et al* (2009) propose bagging as it can "*dramatically reduce the variance of unstable procedures like trees, leading to improved prediction*".

There are two main ways in which bagging is done (Hastie *et al*, 2009; Dietterich, 2000). The first is based on a "*consensus of independent weak learners*" (Dietterich, 2000). In a classification setting, bootstrap samples are drawn from the training data. For example, if a MLR model is built for each sample, to produce a predicted class for a given input, the final classification is the classification which occurs most frequently in all models. It premises on the opinion that collectively, subjects have greater information than individually and this information can be improved by taking the view of the majority, through some form of democratic consensus such as independent voting. This type of concept is generally referred to as "*Wisdom of Crowds*" (Surowiecki, 2004). "*A key assumption of this approach is of the weak learners being independent and identically distributed and for large enough number of bags, they follow a binomial distribution"* (Hastie *et al*, 2009). The direct implication of such an approach is the non-correction of bias specific to the algorithm selected but an improvement in prediction due to the reduction in the variance of the predicted class.

A second approach is to bag class probability estimates for each class, the predicted class will be that class which has the largest average estimated probability. The key problem with bagging is correlation between the different bootstrap sampled models (Hastie *et al*, 2009). By bagging the class probabilities of individual models, the models become difficult to interpret. Performance of the two different approaches has been shown to differ according to the size of the data and the number of the bootstrap samples (Hastie *et al*, 2009).

A simplified algorithm of how bagging is carried out can also be found from Hastie *et al* (2009); see Figure 2.

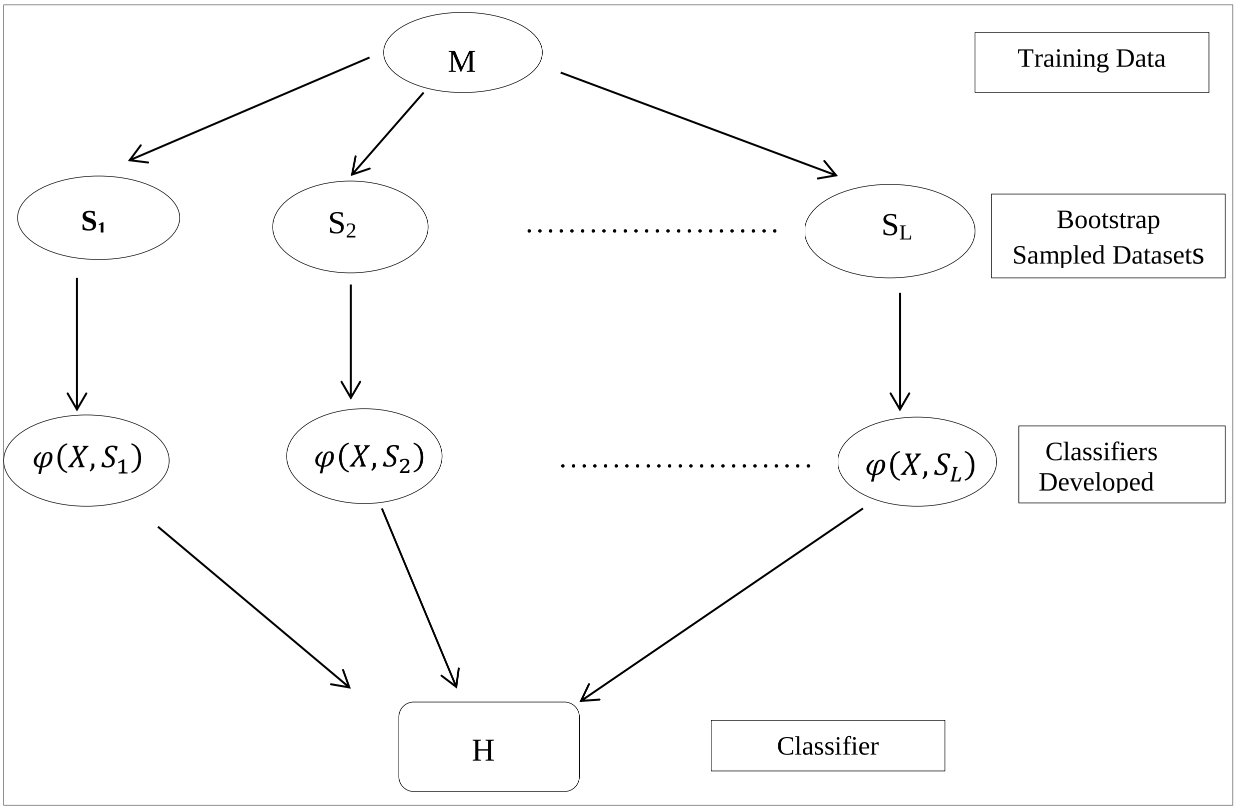

Figure 2: Illustration of bagging.

Given a dataset, $\boldsymbol{S}$, at each iteration $\boldsymbol{l}$, a training set $\boldsymbol{S_l}$ is sampled with replacement from $\boldsymbol{S}$ (i.e. bootstrapping). A classifier $\varphi(X, S_l)$ is learned for each $\boldsymbol{S_l}$. Given a test data sample $\boldsymbol{X}$, each classifier $\varphi(X, S_l)$ returns its class prediction. The bagged classifier $\boldsymbol{H}$ counts the votes and assigns the class with the most votes to $\boldsymbol{X}$ **(**Breiman, 1996a).

In classification: $\boldsymbol{H}$ is equal to the majority class in $\{\varphi(X, S_1), \ldots, \varphi(X, S_L)\}$.

Bagging works well if the classification procedure that is being bagged is not stable. The argument below shows that bagging helps under "squared error loss" evaluation criteria. This is because averaging reduces variance and leaves bias unchanged (Breiman, 1996a).

1. Let Y be the dependent variable.
2. Let X be the vector of the population covariates.
3. $X = (x_1, x_2, x_3, \dots \dots \dots .., x_K)^T$ is the vector of the sample covariates.
4. Let S be the training dataset.
5. Let P be the underlying distribution "true population distribution", of S.
6. Let P_S is the bootstrap approximation to the distribution of P.
7. Let $\{S_l\}$ be a sequence of training datasets containing a subset of S such that:

$$\cup_{l=1}^{L} S_l = S$$

For each bootstrap sample S_l, a classifier is estimated by the function, $\varphi(X, S_l)$

Let $\varphi_A(X, P_S)$ be the aggregated predictor.

Bagging replaces the prediction of the model with the majority of the predictions given by the classifiers.

$$\varphi_A(X, P_S) = E_S(\varphi(X, S_l)) \tag{21}$$

Direct error:

$$e = E_S E_{Y.X}[Y - \varphi(X, S_l)]^2 \tag{22}$$

Bagging error:

$$e_A = E_{Y.X}[Y - \varphi_A(X, P_S)]^2 \tag{23}$$

Jensen's inequality states that given a random variable Z:

$$E[Z]^2 \leq E[Z^2] \tag{24}$$

Therefore it follows that $$e = E[Y^2] - 2YE[\varphi_A] + E_{Y.X}E_S[\varphi^2(X, S_l)] \tag{25}$$

$$\geq E(Y - \varphi_A)^2 = e_A \tag{26}$$

The extra error comes from the variance of $\varphi(X, S_l)$ around its mean $\varphi_A(x, P_S)$. Therefore true population aggregation never increases mean squared error. This suggests that bagging, drawing samples from the training data, will often decrease mean-squared error (Breiman, 1996a). The size of the decrease is dependent on the size of the difference $E_s(\varphi(X, S_l))^2 - E_S[\varphi^2(X, S_l)]$. If the classifier is stable, the difference will be minimal and MBLR will not improve the model.

2.2.4 Random forests

Breiman (2001a) proposes a technique called random forests which uses boosting and bagging as the foundation. He argues that the technique increased noise robustness, which was a major limitation of decision trees and thus reduces the variance leading to lower prediction error (Kim, 2009).

Algorithm

The following is an adaptation of the random forests algorithm as stated by Hastie *et al* (2009).

```
1. For $l = 1 \; to \; L$:
    (a) Draw a bootstrap sample $S_l$ of size $n$ from the training data.
    (b) Grow a random-forest tree $T_l$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
        i. Select $m$ variables at random from the $k$ variables.
        ii. Pick the best variable/split-point among the $m$.
        iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_l\}_1^L$.To make a prediction at a new point $x$:
    Regression: $\hat{f}_{rf}^L(x) = \frac{1}{L}\sum_{l=1}^{L} T_l(x)$    (27)
```

Classification: Let $\hat{C}_l(x)$ be the class prediction of the l^{th} random-forest tree.

Then $\hat{C}^L_{rf}(x) = majority\ vote\ \{\hat{C}_l(x)\}_1^L$ (28)

Feature/Variable Importance

After modelling for the optimal random forests, with say, $\boldsymbol{m}$ unique features; in order to obtain feature importance the following procedure is carried out (Kim, 2009):

- Calculate performance/predictive accuracy on the data left out of the ${T_l}^{th}$ decision tree ("*Out Of Bag"*)
- Randomly permute $\boldsymbol{m^{th}}$ feature in the "*Out of Bag"* data and apply to the respective ${T_l}^{th}$ decision tree. Measure the decrease in accuracy
- Average the decrease in accuracy over all decision trees containing feature $\boldsymbol{m}$ and standardize. This will become the variable importance measure for feature $\boldsymbol{m}$ (Kim, 2009).

2.2.5 ECOC

This technique is introduced by Sejnowski and Rosenberg (1987). In a multi-classifier setting, each class is assigned a binary string of a specified length depending on the "coding strategy" to form a "codeword". The coding strategy deployed should be able to create a series of decorrelated binary strings across the rows and the columns of the data. Decorrelation is the removal of the covariances between observations. Decorrelation is important as it allows the different classifiers to explain varying aspects of the data and thus creating a more robust classifier with less prediction error. Traditionally, Hamming's distance measure is used to measure and minimise the correlation of the binary strings. Various coding strategies exist such as random hill climbing and BCH codes (Windeatt and Ghaderi, 2003). An example of a coding strategy is the exhaustive technique as described in Dietterich and Bakiri (1995):

- Suppose J represents the number of classes
- If $3 \leq J \leq 7$, construct a code of length $2^{J-1} - 1$ as follows:
    - Row 1 is all ones

- Row 2 consist of 2^{J-2} zeroes and 2^{J-2} – 1 ones
- Row 3 consist of 2^{J-3} zeroes, followed by 2^{J-3} ones, followed by 2^{J-3} zeroes, 2^{J-3} – 1 ones.
- In row j, there alternating runs of 2^{J-j} zeroes and ones.

- An example of the exhaustive code is shown in Table 7 on page 59.

After having developed the binary strings of say, length L (*l5 according to Table 7*), L binary classification models are built using common methods such as logistic regression and decision trees. Each model output unit is viewed as computing the probability that its corresponding bit in the "codeword" is one. If one calls these probability values $B = [b_1, \dots\dots.. b_L]$ and each of the codewords $W_j (j = 1 \dots..J)$, then:

- $\boldsymbol{Class(j)} = \boldsymbol{argmin_j} \sum_{l=1}^{L} |\boldsymbol{b_l} - \boldsymbol{W_{j,l}}|$

Where $i,$ is the class to be modelled. The class with the smallest value is assigned as the predicted class (Dietterich and Bakiri, 1995).

2.3 Sampling

In a dataset with a categorical "class" dependent variable, if the classes are not uniformly distributed, the data is referred to as being class imbalanced. If the classes are severely imbalanced such that one of the classes constitutes a very small amount of data, say 5% or less, it is labelled as a rare class. The other classes which are well represented are regarded as common classes. Data collection in the financial services industry is generally for operational purposes without any specific data mining objective in mind. Due to the sheer size of the data, the main problem is to obtain modelling data which appropriately represents the rare classes in order to improve model performance or shorten the training time without degrading model performance (Hastie *et al*, 2009).

2.3.1 Under-sampling

One popular method is to retain all representatives of the rare classes and under-sample the common class without significantly affecting the variability of the model performance (Georges, 2004).

2.3.2 Bootstrap Sampling

In bootstrap sampling, L sample training datasets of size n each are sampled with replacement from the training dataset. Bootstrap sampling is used for modelling purposes and can also be used for model assessment purposes. There is no restriction on the number of bootstrap samples but due to its sampling with replacement, it contains different samples with overlapping data (Efron and Tibshirani, 1993).

2.3.3 Weighted random forests

Another method of dealing with unbalanced datasets is using weighted random forests. Weighted random forests assign a weight to each class, and the minority class is given a larger weight. One common method of choosing weights is by using a cost sensitive algorithm which assigns a significantly higher cost for misclassifying the rare class. Therefore, they penalise misclassification of the minority class more heavily. Weighted random forests are computationally less efficient with large imbalanced data, since they need to use the entire training dataset. In addition, assigning a weight to the minority class may make the method more vulnerable to noise, "*mislabelled class*" (Chen, Liaw and Breiman, 2004).

2.3.4 Other uses of Sampling

In building classification models, the data is generally divided into 3 partitions (Hastie *et al*, 2009).

Simple random sampling with proportional allocation is commonly used to partition the data. Training data will always contain the highest proportion of the data as this is the dataset used to build the model. Validation datasets are used to fine tune the parameter estimates whilst the test dataset is used to test the model for generalisation and robustness (Hastie *et al*, 2009). The literature commonly refers to the test dataset as the hold out sample as this dataset is held back from training the model (Burez and Van den Poel, 2007).

2.4 Evaluation Criteria

The main aim of the classification models is to fit the data by minimising the prediction error. Therefore, the model needs to be validated in terms of both the goodness of fit and the prediction error whilst avoiding over-fitting or under-fitting the data. Over-fitting the data means an unnecessary increase in model complexity, i.e. increasing the number of parameters and the model degrees of freedom beyond that which is necessary. Under-fitting is the opposite of over-fitting, i.e. too simple a model will not fit the data well. Model assessment techniques are broadly classified into the following three groups (Cios *et al*, 2007).

- Resampling methods are very popular in evaluating supervised learning methods e.g. holdout sampling, cross validation and bootstrap.
- Principle of Parsimony methods are not formal, are very simple, but are probably the most frequently used methods. This is based on how easy the model is to interpret.
- Analytical methods are formal and highly technical but not very practical. E.g. Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC).

2.4.1 Classification Error

It is assumed that some underlying hypothesis exists, this means that in the training data, known inputs correspond to known outputs. It then follows that for the given data, if the total number of rare classes and the total number of common classes are known, we are able to form a misclassification matrix commonly known as confusion matrix. Figure 3 below describes a general form (Cios *et al*, 2007):

Test Result			
Truth(Gold standard Hypothesis)	Positive	Negative	
Positive	True Positive (TP) (no error)	False Negative (FN)(Rejection error, Type 1 error)	Total of true positives
Negative	False Positive (FP)(Acceptance error, Type II error)	True Negative (TN) (no error)	Total of true negatives
	Total Classified Positive	Total Classified Negative	Total Population

Figure 3: The confusion matrix.

Definitions (Bradley, 1997):

Sensitivity=$\frac{TP}{TP+FN}$ (29)

Sensitivity measures how accurate one is predicting class membership. It is also referred to as the hit rate.

Specificity=$\frac{TN}{TN+FP}$ (30)

Specificity measures the number of negative cases correctly classified as negative.

Hand and Till (2001), discuss the above metrics as relatively easy to obtain for a classical two case model. Although the global misclassification rate is easy to calculate, however, it is more complicated to obtain the optimal misclassification costs e.g. see Bradley (1997); Provost, Fawcett and Kohavi (1998); Adams and Hand (1999; 2000). If costs are not easily obtainable, Hand *et al* (2001) state that one is served better by the AUC.

Whilst commonly used to evaluate classifier models, accuracy is considered an inappropriate metric for classification modelling because*:*

- *"It does not take into account predicted class membership probabilities but instead assumes a threshold to obtain classifications from probabilities.*

- *It is unreliable in a situation of class imbalance (Croux and Lemmens, 2006)"* (De Bock and Van den Poel, 2012).

2.4.2 AUC / Gini coefficient

Another evaluation criterion is AUC. Several authors such Provost *et.al* (1998) argue that the AUC is an objective performance criterion, which is suited for the comparison of classifier performance. It evaluates the ability of a classifier to distinguish between the classes based on the predicted class membership probabilities, and is therefore suitable for imbalanced classification problems (Japkowicz, 2000; Demšar, 2006).

The ROC curve is a graphical plot of the true positive rate versus the false positive rate (x-axis). The total area below this curve is known as the AUC. In a two class problem, the AUC measures the level of seperability between the two estimated class distributions. The more separated the two estimated distributions are, the greater is the AUC (Hanley and McNeil, 1982; Hand, 2000). The aim is to maximise the AUC. The major advantage of this evaluation criterion is of not being influenced by the costs of misclassification. It assesses general class seperability and thus allowing the user to select the best classifier as well as the optimal threshold (Cios *et al*, 2007).

The AUC or the *c- statistic* (Hastie *et al*, 2009) can also be calculated and compared for all the different classifiers. A similar measure is the Gini coefficient, which is simply twice the size of the AUC minus 1.

2.4.3 Seperability in a multi–class instance

The definition of the AUC discussed above relates to a two class problem. Hand *et al* (2001) discuss the extension of the AUC to a multi-classifier as not being straightforward as one would have expected. This is due to the fact that the confusion matrix is no longer a $2x2$ matrix as shown but a JxJ matrix, if we assume J classes. "*The default (and, indeed, popular) choice of equal costs for the various different kinds of misclassification, leading to overall misclassification rate, is in fact very rarely really suitable*"(Hand *et al*, 2001). They propose a method which extends from a two class model. This involves aggregating the AUC for all pairs of classes. However, this method is independent of the costs of misclassification and the prior distribution of the classes. This has the implication of measuring some other aspect of model performance. The mathematical derivation as adapted from Hand *et al* (2001) is shown below:

Assume classes $(0, 1, 2, 3 \ldots, J - 1)$

For any pair of classes c and d Let:

$\hat{p}(c|x)\ for\ c = 0,1,2 \ldots., J - 1$ be the estimated probability for class c.

$\hat{p}(d|x)\ for\ d = 0,1,2 \ldots., J - 1$ be the estimated probability for class d.

$\hat{A}(c|d)$ be the probability that a randomly drawn member of class d will have a lower estimated probability of belonging to class c than class d . This measure is computed from either using $\hat{p}(c|x)$ or $\hat{p}(d|x)$.

$\hat{A}(c, d) = \left[\hat{A}(c|d) + \hat{A}(d|c)\right]/2$ be the measure of seperability between classes c and d. (31)

The overall performance of the classification rule, $\boldsymbol{M}$ in separating the c classes is then the average of this over all pairs of classes:

$$\boldsymbol{M} = \frac{2}{j(j-1)} \sum\nolimits_{c<d} \hat{A}(c, d) \qquad (32)$$

2.4.4 Akaike Information Criterion (AIC) / Bayesian Information Criterion (BIC)

AIC and BIC (Cios *et al*, 2007) are statistical measures that are used to choose between models that use different number of parameters, and are closely related to each other. The general idea is motivated by our need to estimate the prediction error, and use it for model selection:

$$\boldsymbol{AIC} = -2logL + 2(\frac{d}{N}) \qquad (33)$$

Where $logL$ is the maximised log-likelihood, defined as:

$$\log L = \sum\nolimits_{i=1}^{N} \log \bar{P}_{\theta}(y_i) \qquad (34)$$

$\bar{P}_{\theta}(y_i)$ is a family of densities containing the “true” density. N is the number of parameters in the model:

$$\boldsymbol{BIC} = -2logL + d\log N \qquad (35)$$

To use AIC and BIC for model selection, we simply choose the model giving smallest AIC or BIC over the set of models considered (Cios *et al*, 2007).

2.4.5 Lift

Lift is an evaluation measurement of a classifier in terms of correctly classifying inputs as opposed to a random classification (Cios *et al*, 2007).

$$\boldsymbol{Lift} = \left(\frac{TP}{TP+FN}\right) / \left(\frac{TP+FN}{TP+TN+FP+FN}\right) \qquad (36)$$

The lift provides a quantitative measure of the gain in performance by using the classification model as opposed to random classification of outcomes. The lift value is provided at any chosen classification threshold. Through the calculation of the lift value across all threshold values, a lift curve is obtainable by ordering these using these values as the y-axis and the proportion of the population classified as the x-axis. A common method is to arrange the population into deciles, with the top decile representing the population with the highest probability estimates and the bottom decile having the lowest estimates. The lift value is then calculated for each decile and a lift curve is constructed by plotting the lift values versus the deciles. The lift curve can either be cumulative or non-cumulative.

A lift benchmark value of 1 is set, with any lift value above this number indicating model improvement. The lift curves are very useful in the selection of the classification thresholds. The lift values have to be analysed in relation to the sampling proportions.

2.5 Comparing techniques

Numerous articles published in the literature discuss the effectiveness of the different models. Although there are relatively few articles which discuss the comparison of the exact techniques to be used in this study, a significant number features a discussion about some of the techniques in combination with others which are not mentioned in this study. An overview of the literature shows key themes in the comparison of the different modelling methods namely:

- Variable importance and interpretability
- Predictive accuracy
- Model generalisation/robustness
- Model efficiency and ease of use

The relative importance of the themes is dependent on the problem at hand. The best model in a specific scenario might not necessarily be good for any other scenario. However, the literature almost always chooses a benchmark model to compare any other model against. Drummond and Holte (2003) state that the decision trees are the default model to compare against. However, a survey of the literature has shown that this is also dependant on the field of study. In the medical field, researchers normally use either logistic regression or survival models as the benchmark models.

2.5.1 Advantages and disadvantages of the techniques

The advantages and disadvantages of the techniques below are adapted from various authors. Other authors/reviewers can also argue these advantages and disadvantages.

Logistic regression

The logistic regression method is mainly used for model building because:

- It is well known, conceptually simple and widely used by marketers.
- It is simple to interpret.
- Generally provides good and robust results in comparative studies.
- In database marketing, it may outperform more sophisticated methods.

However, one of the major drawbacks is of not being able to easily build a multi-classifier model (Levin and Zahavi, 1998).

Decision trees

Decision trees are mainly used because (Breiman *et al*, 1984):

- No tuning parameters are required.
- No need for transformation of the variables.
- Robust to outliers.
- Easily handle missing data.
- Can easily handle multiple classes.
- They do not assume any prior distribution.

Some of the major drawbacks of decision trees include (Murthy, 1998):

- They tend to “overtrain” the data.
- They are highly unstable.
- Generally, as the number of classes increase, so does the number of terminal nodes since rules are constructed to explain each class which may become difficult to handle.

Bagging of MLR

Ensemble method bagging is used for modelling mainly because (Dietterich, 2000); (Hastie *et al*, 2009); (Breiman, 1996a):

- It generalises very well due to the use of random samples.
- It is simple to build a multi-classifier model.
- They are more stable compared to the decision trees.
- Improves accuracy and robustness of regression trees.

However, bagging is not easy to understand and adapt. The combined classifier is also not very transparent. Bagging does not guarantee improved performance as some classifiers are able to extract the maximum attainable accuracy. No amount of bagging would result in improved performance in these instances (Breiman, 1996a).

Random forests

Below are some of the features of random forests (Liaw and Wiener, 2002); (Hastie *et al*, 2009); (Breiman, 2001a):

- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It is robust against over fitting (Liaw and Wiener, 2002).
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced datasets.

ECOC

ECOC is used for modelling mainly because:

- ECOC has a high predictive accuracy due to its ability to reduce the variance in the class predictions (Breiman, 1996b).
- ECOC performs consistently well in small data sets.
- ECOC is not dependent on a specific coding strategy to obtain a high predictive accuracy. Several coding strategies have been shown to provide consistent performance (Windeatt and Ghaderi, 2003).
- Although ECOC does not provide probability estimates, it can provide confidence estimates that are similar in performance as to other approaches to multi-class problems (Dietterich and Bakiri, 1995).
- ECOC is easily scalable to large classification tasks and is inexpensive for these types of tasks compared to other methods such as one per class (binary choice for each class) (Dietterich and Bakiri, 1995).

Listed below are some of the drawbacks of ECOC:

- It is difficult and time consuming to deduce the appropriate codes for the different classes. Dietterich and Bakiri, (1995) also showed that model generalisation is also dependent on the length of the "codeword". The longer the codeword, the better the model performance in terms of generalisation.
- The codewords does not provide specific insights although recent studies have shown that one can apply a coding strategy which explains varying features in the data.
- ECOC gives no insight into the variables affecting different classes as it is purely a classification algorithm.
- Multi-classifier models in general, do not give proper variable insight as much as one per class.
- ECOC tends to produce much larger and more complex decision trees and as well as much more hidden units if Neural Networks are deployed as the modelling strategy (Dietterich and Bakiri, 1995).

2.5.2 Empirical studies

Over the years, techniques have been proposed in terms of identifying models that provide both interpretability and good fit (De Bock and Van den Poel, 2012) such as decision trees, logistic regression and general additive models. However, Neslin, Gupta, Kamakura, Lu and Mason (2006) suggest explanation and prediction as two distinct features of a model that cannot be reconciled.

Strobl *et al* (2007) discuss variable importance in decision trees, bagging and random forests. Generally, in decision trees, variable importance is based on selection frequency and the Gini importance measure. In bagging, variable importance is based on selection frequency and drop in model performance by excluding a specific variable. For random forests, Strobl *et al* (2007) discuss a method called permutation accuracy importance. This method involves randomly permuting a variable and monitoring the drop in model performance for each tree. The result is then averaged across all trees. The variables with the highest relative drop are the most important. They are calculated using out of bag data.

Strobl *et al* (2007) reiterate that bias in the variable importance measures generated by the decision trees due to the use of a "greedy algorithm" results in suboptimal models. They also state that the split criterions used by decision trees tend to favour variables with more categories. This bias also filters through to bagging and random forests which use decision trees. However, they also argue that- in bagging and random forests, variable importance bias is further compounded by bootstrap sampling without replacement. Bickell and Ren (2001) argue that, "*bootstrap hypothesis testing fails whenever the distribution of any statistic in the bootstrap sample, rather than the distribution of the statistic under the null hypothesis, is used for statistical inference*".

De Bock and Van den Poel (2012) propose an ensemble classifier based on bagging and random subspace method (RSM) combined with random forests using the generalised additive model (GAM) as the base classifier. This is done in order to create a model which reconciles high interpretability and superior classification performance, based on the work previously done by De Bock, Coussement, and Van den Poel (2010). They call this ensemble classifier GAMensPlus. They deploy this model on six prediction datasets which were obtained from large European companies. The companies are from varying industries and the data attributes also vary with some of the data having rare instances in the response variable. Six benchmark models were chosen to compare against this model on all datasets namely: bagging, random forests, RSM, logistic regression and GAM. They use four evaluation methods namely accuracy, AUC, top decile lift and lift index. Across all four metrics and datasets, they conclude that GAMensPlus provide the best results followed closely by logistic regression and random forests; bagging does not perform as well. In terms of pure predictive accuracy, random forests prove to be the strongest predictor. Although they discuss the variable importance and interpretability, the authors do not compare against the interpretability of the other techniques such as logistic regression and decision trees. Logistic regression and random forests tend to generalise very well.

In the medical field, machine learning is playing a key role in the medical diagnosis of illness. Hsieh, Lu, Lee, Chiu, Hsu and Li (2011) discuss the medical diagnosis of acute appendicitis in patients using statistical models. They compare the performance of random forests, support vector machines (SVM), artificial neural networks (ANN) and logistic regression as the benchmark. They also have a manual clinical scoring system called

Alvarado scoring system to compare against. They collected patient data between January 2008 and December 2008 by reviewing patient records. They then split the data between training (75%) and testing (25%). Variable selection is done using consistency subset selection and exhaustive search methods. They compare the different models used to fit the data in terms of accuracy, sensitivity, specificity, positive predicted values, negative predicted values and the AUC. Across all the methods, random forests have the highest accuracy (0.98) followed by SVM (0.96), ANN (0.91), logistic regression (0.87) and the Alvarado scoring system (0.77). Using pairwise comparison, they cannot conclude a significant difference in performance between random forests and SVM but random forests is found to be significantly superior to the rest of the models.

Burez and Van den Poel (2007) built a customer churn prediction model for a pay TV channel using the various techniques namely logistic regression, logistic regression with markov chains, random forests and a rules based criterion previously used for the different campaigns. They compare the performance of the different models by assessing the sensitivity, specificity, AUC and the percent correctly classified which is benchmarked against the proportional chance criterion (Morrison, 1969) of each model. The results show that the random forests perform best if a small group of customers is selected whereas the logistic regression worked best for high cut off values. There is no difference in performance between the logistic regression and the logistic regression with markov chains. The random forests model is tested for robustness and generalisation using a hold out sample and an out of time sample. An out of time sample is defined as a data sample extracted from a different time period in comparison to where the training data was obtained. The model maintains the performance on both samples but also shows an even better performance on an out of time sample.

Xie, Li, Ngai and Ying (2009) fit a customer attrition model for customers in a retail banking setting. The aim of the study is to address the imbalance in the data distribution of the response variable by altering the sampling techniques and allowing the algorithms to be more cost sensitive to the misclassification of the minority classes. They adapted the random forests model by combining the balanced random forests with the weighted random forests to obtain the improved balanced random forests model (IBRF). They state that on one hand, the sampling technique which is employed in balanced random forests is computationally more efficient with large imbalanced data, more noise tolerant and on the other, the cost sensitive learning used in weighted random forests has more effect on the classifiers produced by decision tree learning methods. They proceed to apply the IBFR on a dataset about customer attrition from a Chinese bank. The authors use top decile lift and AUC as the evaluation criterion. For model comparison purposes; ANN, weighted random forests, balanced random forests (BRF), decision trees, class weighted score support vector machine (CWC-SVM) and a random model are used.

The authors demonstrate the IBFR model as performing the best, followed very closely by the BFR model. It is interesting to note that amongst all the models compared to the random model, decision trees perform the worst especially in the case where the cut off values are high. The authors do not statistically test for significant difference in performance between the different models and hence it is left to the readers to make visual conclusions. The authors also do not discuss the trade-off between the increased complexities of the IBFR versus the benefits of increased accuracy. Model interpretability and generalisation are also not discussed.

Niculesu-Mizil and Caruana (2006) compare the different modelling techniques across different problems and metrics. For each metric, they rank each model and subsequently calculate the proportion of times each model ranks from top to bottom based on a bootstrap sample. The bootstrap sampling is repeated a total of a thousand times. As can be seen from Table 1, boosting with decision tree weak classifier ranks first- 58% of the times whereas the naïve bayes model ranks the worst- 69% of the time. The difference between boosting with decision trees and boosting with decision stumps is that the latter does not use the full decision trees but rather single level decision trees.

It is clear from the table above that the ensemble methods of classification consistently ranked the highest in terms of all the problem types and metrics.

Model	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
Boosting with decision trees	0.58	0.23	0.16	0.02	0.01	0	0	0	0	0
Random forests	0.39	0.53	0.08	0	0	0	0	0	0	0
Bagging with decision trees	0.03	0.23	0.57	0.15	0.02	0	0	0	0	0
Support vector machine	0	0.01	0.15	0.57	0.24	0.03	0	0	0	0
Neural nets	0	0.01	0.04	0.23	0.61	0.12	0	0	0	0
K nearest neighbour	0	0	0	0.01	0.11	0.59	0.25	0.04	0	0
Boosting with decision stump	0	0	0	0.01	0.01	0.26	0.71	0	0	0
Decision tree	0	0	0	0	0	0	0	0.61	0.29	0.09
Logistic regression	0	0	0	0	0	0	0.04	0.31	0.42	0.23
Naïve bayes	0	0	0	0	0	0	0	0.03	0.28	0.69

Table 1: Classification of the different models

Several authors such as Strobl *et al* (2007) and Niculesu-Mizil and Caruana (2006) compare the techniques mentioned above in different settings and find random forests consistently outperforming other supervised learning techniques. The major drawbacks of this technique are the difficulty in explaining the results as well as providing specific reasons for the classification of a specific individual. This type of information is invaluable to the marketer/modeller and which is why the decision tree is still a very popular predictive modelling technique.

Dietterich and Bakiri, (1995) build ECOC models across various type of data sets and compare model performance with the other traditional modelling techniques. Across varying type of datasets, ECOC consistently outperforms the other models such as decision trees and back propagation. ECOC models are found to be highly robust even in small datasets and also produce class prediction with confidence estimates. ECOC fares favourably against other types of multi-class modelling methods such as one per class and one vs. the rest. The authors also show that model performance is not dependent on the codeword assigned however performance tends to vary with the length of the code-word. Generally, ECOC models with longer codewords outperform models with shorter code words.

Windeatt and Ghaderi (2003), discuss the various coding and decoding strategies that can be used for the ECOC model. They note the importance of Hamming's distance by concluding that a high minimum distance between codewords implies a reduced bound for the generalisation error. They however argue that traditional coding strategies are designed to address model generalisation and, are problem independent. They emphasise the inability of sub-problems to represent the main problem, as well as the errors induced due to this distance based measure. They go on to further propose other various types of coding and decoding strategies including some which contain coding strategies related to the features of the data. Kuncheva (2005), discusses the insufficiency of using the Hamming's distance by stating that, the approach assumes a "worst case scenario" approach. She also states that, since it "guarantees" that a given amount of errors will be corrected; it is thus attractive for deriving bounds on the error. She however argues that this approach of reducing bounds is not practical as one might want to build a classifier which can misclassify small proportions of data but will on average outperform a code with a larger minimum Hamming's distance.

2.6 Gaps in the Literature

Listed below are some of the gaps that have been identified in the literature:

- Although there is a wide array of research comparing various modelling/classification techniques, the majority is limited to only two class modelling objective. Most research appears to assume that the results obtained will easily generalise to the multi-classifier case. Even the most recent literature does not seem to have enough appetite to probe the multi-classifier case.
- Most research limits the comparison to visual comparison of the AUC, lift chart and the top decile lift. Not as much research focuses on whether the differences are statistically significant. The literature discusses model comparison techniques but does not seem to have advanced at the same pace as the creation of the different classification algorithms which are continually being enhanced. The Friedman pairwise comparison test has been used by some authors such as Hsieh *et al* (2011).

- Although a lot of research discusses the use of the AUC as part of evaluation criterion for assessing modelling methods, not much literature has focused on the complexities introduced by a multi-classifier comparison. Hand *et al* (2001) extensively discuss the inadequacies of using misclassification in assessing model performance. They also address the shortcomings of using the standard ROC plot in the multi-classifier setting and they propose another type of AUC. However, not much literature applies this algorithm as there is more focus on the proportion correctly classified as the proxy.

3 Methodology

This section outlines the methodology used. It outlines the objective, analytical approach, data used and evaluation criteria.

3.1 Objective

The key objective to be satisfied is:

- Obtain a statistical predictive model which is able to correctly classify the best product to cross sell a customer. The selection should take into account the best model for implementation using the evaluation criteria defined.

3.2 Analytical Approach

3.2.1 Data Pre-processing

Bank data was obtained for modelling purposes from the bank's data warehouse. A random point, June 2011, was chosen as the reference point. All active customer accounts as at this reference point were obtained, and those accounts with arrears, bad debts or which were frozen were excluded. Since the data was obtained at a customer account level, this meant several rows of data for customers with multiple products. Data was then aggregated to customer level, with indicators created for each product holding.

Customers who did not have a valid identification number or incomplete or redundant product information were excluded. Also excluded, were customer below the age of eighteen, those above the age of sixty, staff members and high net worth customers.

The customers of interest were those that had only a transactional product as at the reference time point, June 2011. These customers constituted seventy percent of the dataset. For the purposes of this study, customers with tenure of less than six months were excluded and those with a banking relationship of greater than two hundred and forty months were right censored. This reduced the dataset to roughly 60-65% of the original dataset depending on the reference point. A sample of four thousand one hundred customers was obtained for each period. The definition period in this study was a month.

Having obtained the base, historical information related to the data was collected and collated. Historical information was limited to a maximum of eight months before the reference time point (represented as $\boldsymbol{T_{-8}}$ in Figure 4 below). For all customers selected as at June 2011, historical information relating to their transactions, demographic profile, geographic profile and account specific profile was collected starting from November 2010 up to June 2011. This information formed the basis of the predictor variables. Variable collation, cleaning and transformation is discussed in more detail later in the section.

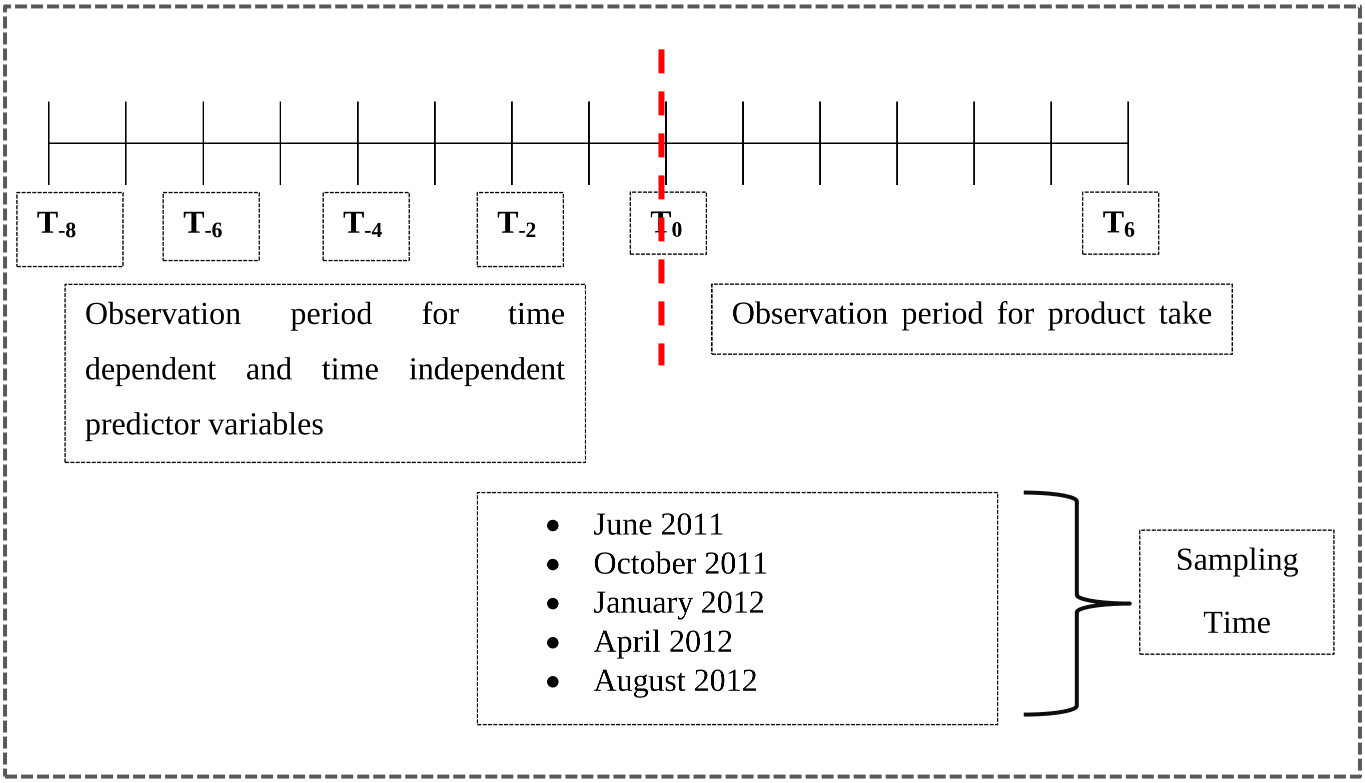
T-8
T-6
T-4
T-2
T0
T6
Observation period for time dependent and time independent predictor variables
Observation period for product take
June 2011
October 2011
January 2012
April 2012
August 2012
Sampling Time

Figure 4: An Illustration of the data collection process. Observations were sampled at T_0 and historical information was obtained from T_{-8}. The observations were then monitored for product take up from T_0 to T_6 (representing 6 months from T_0).

Having obtained the historical information, customer product take up was then observed over the subsequent six months as shown in Figure 4. As an example, customers as at reference time point, June 2011 were observed until December 2011 to take note of the product take up. Any customer who left the bank in the period under observation was only included up to the period of exit.

The product take up of interest were namely; investment, secured loan, unsecured loan, credit card and non-take up of any product. Product take up was classified into states implying that take up of a new product moved a customer from one state to the other. Repeat purchases were however not taken into account. That is, if a customer opened another bank account of a product he/she already had, no movement into a different state occurred.

In order to understand the movement of customers into the different states, the following assumptions were made:

Assumptions

Data handling

- Transitions between states were assumed to occur at discrete times. This implied that customers were not continuously monitored but however, product take up was noted at the end of the observation period.
- The waiting time to taking up a product was not important. Whether a customer took a month into the observation period to take up a product or any other time, say five months, was not considered for these studies.
- For simplicity, in cases where a customer had acquired more than one product - the product which was bought first was considered. This multiple take up occurred in 10% of all instances. This was a significant constraint whose limitation was evident in the classification accuracy as the model could only assign a customer to a single predicted state.
- Repeat purchases of the same product were not taken into consideration.
- The transactional account was assumed to be the entry product into a banking relationship.
- An individual could not move to any other state without a transactional account.
- Only product take up was considered. If a customer opened and closed the same product during the observation period, the product take up is still considered.

Customer Bias

- Customers were exposed to the same conditions and had equal access to information.
- Economic conditions are static.
- No interventions such as direct marketing contributed in customers moving between states.
- Customers have their primary banking relationship with this bank and no products of interest are held with any other financial services company.

Dependant/Target Variable

The products of interest are the following:

1. Investment Account
2. Secured Loan Account
3. Unsecured Loan Account
4. Credit Card Account
5. No take up

The levels of the dependent variable are shown below:

The dependant variable was defined across the four products with “no take up” being the reference category. If a customer took up a specific product during the period of observation, it was classified as below:

$$\mathrm{Y} = \begin{cases} \text{investment account} & (\mathrm{b.INV}) \\ \text{secured loan account} & (\mathrm{c.SL}) \\ \text{unsecured loan account} & (\mathrm{d.UL}) \\ \text{credit card account} & (\mathrm{e.CARD}) \\ \text{no take up} & (\mathrm{NO\_TAKE}) \end{cases}$$

The dependant variable is classified as a nominal variable with the secured loan account being the rare class. At the end of the observation period, each customer was classified into one of the levels above depending on the product taken up.

Other Cohorts

Having created the data as stated above for the reference time point, June 2011, the process was iterated using other reference points. Since product take up could be affected by seasonality with unsecured lending being popular during the start of the year as a case in point, a total of five reference points were selected (see Figure 4). It was therefore important for the reference points not to be close to one another. For each reference point selected, the data selection and sourcing was executed in an identical manner to the creation of the June 2011 cohort.

The five different data sets were later combined to create the model set which contained twenty thousand and five hundred observations. No cohort identifier or marker was retained. If however, a customer was selected at multiple reference points, stratified random sampling with the identification number being the strata. In order to remove duplicates, the strata size was set to one.

Summary of model dataset creation

The following is a summary of the data creation process.

Identify reference points of interest. The reference points should at least span over a year to counter seasonality. This becomes a set of reference points.

For each reference point:

- Obtain a target group of customers from the reference point.
- Identify the time periods to obtain historical data as well as the observation period.
- Obtain historical data as well as monitor product take up in observation period.
- Define the dependant variable.
- Consolidate the datasets to create a model dataset.
- Perform a stratified random sample at customer record level. This ensures that a customer is only represented once in the model set. This is done so as to ensure independents of rows.

Benefits of this approach

Businesses go through various cycles and experience temporal shocks which might distort customer product take-up. Some products are more popular at certain times of the year such as unsecured loans at the start of the year. One thus runs the risk of creating a model which can perform well only at certain times of the year and might not be reusable. Through the selection of various reference points, the model set is less time point dependent. This also ensures that the classification probabilities are more stable and the model is able to capture various nuances in the data which occur during the whole year.

The creation of a six month observation period allows the classification probabilities to be useable over a six month window period. This provides ample time for a marketer to convince a customer as well as provide an opportunity for early prediction. If the observation period was shorter, say three months, a customer might have already made their decision and committed with another provider. Other products such as secured lending do require time for mandatory processes and thus a longer observation period allows the marketer to engage the customer at the optimal time.

Through selecting data at various reference points, customer records may be duplicated as one customer may be represented in multiple datasets. This results in the violation of the assumption of independence of rows. By using stratified random sampling, this ensures that there is no systemic elimination of duplicate records.

Selection of data at various time points also increases the number of observations available for modelling.

Predictor Variable Definition

The predictor variables derived from the data were classified into two categories, namely, snapshot and across history period. Snapshot variables were those that were extracted at the reference time point, whereas, across history period were extracted across the eight months history period prior to the reference point. A total of 117 predictor variable were in the model dataset.

Snapshot variables are those that do not change in the short term and were predominantly demographic variables. Variables age, gender, race, marital status are assumed to be static in the short term. Table 2 lists most of the snapshot variables considered in the model. The rest of the variables are listed in the appendix. Snapshot variables were also derived from other variables- an example being the variable “contacts” listed in Table 2, which is a count of the number of communication channels the bank has with the customer. This variable was derived from counting the listed channels such as email, home phone, cell phone and work phone.

Variables were classified as continuous, nominal, ordinal or binary.

Across history period variables were those that were derived from the historical data across the eight months. Derived predictor variables ranged from simple summations to calculation of averages and rate of change of variables. Available information from the data contained balances, deposits, withdrawals, payments, transfers, enquiries and purchases across the different channels. The banking channels included automated teller machine (atm), bank branch, cell phone, internet, telephone and point of sale services. Point of sale transactions are those that involve a customer swiping a bank card. The transactions were analysed in terms of volume, value and recency. Four examples of derived predictor variables are shown below:

Based on Figure 4, derived variables were calculated using historical information i.e. they are derived from the time period T_{-8} to T_{-1}.

$$Let \ \ i = 1,2, \dots ,8$$

$$Let\ month\ \ T_{-i} = 9 - \ i$$

$$Let\ D_{-i} = total\ deposits\ for\ month\ \ T_{-i}$$

$$1.\text{Total deposits} = \sum_{\mathrm{i}=1}^{8} \mathrm{D}_{-\mathrm{i}}$$

$$2.\text{Average deposit} = \sum_{\mathrm{i}=1}^{8} |\mathrm{D}_{-\mathrm{i}}|/8$$

$$3.\text{Rate of change of deposits} = \frac{\sum_{\mathrm{i}=1}^{8} T_{-i}D_{-i} - \sum_{\mathrm{i}=1}^{8} T_{-i} \sum_{i=1}^{8} D_{-i}}{8\sum_{i=1}^{8} T_{-i}^{2} - \left(\sum_{i=1}^{8} T_{-i}\right)^{2}}$$

Equation (3) above calculates the gradient of the deposits as they change over time (the slope of the regression line). A positive value represents a general increase in monthly deposits over time and a negative value represents a general decline.

4. Significant deposit

$$= \begin{cases} \max_i |D_i| & if \quad \max_i |D_i| > 3\left(\sum_{\mathrm{i}=1}^{8} |\mathrm{D_i}| - \max_i |D_i|\right)\Big/7 \\ & 0 \end{cases}$$

Equation (4) calculates the maximum amount of money deposited over the eight months of interest. Where the maximum monthly deposit is at least 3 times higher than the average monthly deposits of the other 7 months, it is then classified as the significant deposit; otherwise no significant deposit was made.

Listed in Table 2, is the sample of variable used in model training:

Variable	Type	Description	Category
Age	Continuous	Customer age	Snapshot
Race	Nominal	Race group	Snapshot
Gender	Binary	Gender	Snapshot
Income group	Ordinal	Salary	Snapshot
Marital Status	Nominal	Marital Status	Snapshot
Occupation	Nominal	Employment Category e.g. skilled labour	Snapshot
Province /Area	Nominal	Geographical Province	Snapshot
Customer segment	Ordinal	Relationship segment as classified by bank e.g. Private Bank	Snapshot
Tenure	Continuous	Number of months customer has been on book	Snapshot
Preferred Language	Nominal	Preferred Correspondence Language	Snapshot
Credit Rating	Ordinal	Credit Bureau credit rating	Snapshot
Contacts	Ordinal	Number of communication channels customer can be contacted	Snapshot
Banking Channels	Nominal	Type of banking channels used by customer e.g. internet	Across History period
Lazy Balance	Ordinal	Lowest account balance in last 8 months.	Across History period
Transaction Types	Nominal	Popular transaction type performed by customer e.g. deposits, transfers	Across History period
Payment Destination	Nominal	Debit orders linked to the account e.g. motor insurance, gym	Across History period
Customer Revenue	Continuous	Average monthly revenue generated from customer transactions	Across History period
Frequency	Continuous	Volume of monthly transactions	Across History period
Recency	Continuous	Recency in communication with bank measured in months	Across History period
Average monthly deposits	Continuous	Deposits over last 8 months	Across History period
Average monthly residual account balance	Continuous	Average net monthly balance. Could be positive or negative.	Across History period
Significant deposit	Continuous	A deposit at least 3 times the average monthly deposits.	Across History period
Significant withdrawal		A withdrawal at least 3 times the average monthly withdrawal.	Across History period
Gradient of balances	Continuous	Rate of change of balances over last 8 months e.g. deposits, lazy balance	Across History period
Average electronic	Continuous	Average monthly electronic transactions in volume and value	Across History period
Average ATM	Continuous	Average monthly atm transactions in volume and value	Across History period

Table 2: Data Description.

3.2.2 Variable Selection

The data contained a total of one hundred and seventeen predictor variables including derived variables, one customer identification variable and the response variable “N_Product”. A brief description of the data is provided in Appendix A. It should be noted that the more the variables included, the greater is the amount of data points required for the model building purposes due to the curse of dimensionality (Hastie et.al, 2009). All the variables were assessed for missing data and if a variable had more than twenty percent missing data, it was discarded. Missing data was assumed to be missing at random. Missing values were imputed using the mean value where possible and the most popular class was assigned in cases which had missing nominal data.

Variable correlation tests and variable clustering were carried out so as to remove highly correlated data. Pearson correlation tests were done on continuous data with correlation coefficient of +/- 0.45 being used as the significance level. If any two variables were found to be highly correlated (coefficient of +/- 0.7), the variable with the least Gini index in relation to the classification variable was dropped. As an example, total deposits were found to be highly correlated to salary and therefore total deposits was dropped. A listing of retained variables is available in Appendix B. During data creation, there was a risk that some variables might be directly correlated to one of the levels in the dependent variable. This is commonly referred to as circular referencing. All variables were visually analysed for circular referencing to the dependent variable using the chi-squared test and no such references were found.

Chi-squared and Cramer’s V tests for variable importance was deployed to the data for variable selection. Chi-squared tests were carried out for the categorical variables at 5% significance level and those variables which were found not to be significant were dropped as they did not contribute a significant amount of information in explaining the dependant variable. The limitation of this method is the inability to take into account variable interactions since it is a univariate test. The graphical plots of the tests are shown in Appendix B. Variable outliers were right censored.

In the Multinomial logistic regression (MLR) model, the selection criteria specified 0.15 significance level; for variables entering the model and 0.1 for variables staying in the model.

Backward, forward, stepwise and best subset selection criteria were applied and the stepwise selection criterion which provided the best comparative results were used (Georges, 2004).

MBLR selected the appropriate variables in terms of the variable importance criterion. MRFD also determined variable importance according to the “out of bag” procedure previously discussed. ECOC did not have a variable selection procedure. This study did not put much emphasis on variable selection procedures but rather on model fit.

3.2.3 Sampling Methods

For modelling purposes, a random sample of between seventy five to eighty percent of the data was reserved for model training purposes and the rest for the model testing purposes. Simple random sampling was used as the data partitioning method.

Since the data contained rare classes, “under-sampling” as defined in section 2.3.1 was carried out. Before under- sampling was carried out, the distribution of the classes in the target variable was noted, these are generally regarded as “population priors”. The following steps were followed in order to under-sample (Georges, 2004):

- Partition data into training and test datasets using an arbitrary sampling method such as simple random sample.
- Remove the test dataset and remain with the training dataset (S).
- Sample all rare classes from (S).
- Under-sample the common classes from (S) i.e. sample a common class dataset with equal size to the rare classes using stratified random sampling (Weiss and Provost, 2003).
- Combine the rare classes and the sample common classes dataset to form training dataset (S^*).
- Build the classifiers using the training dataset (S^*).
- Adjust the predicted probabilities with the population priors to reflect population estimates.

Under-sampling has a drawback of discarding useful information and thus degrading the classifier performance (Weiss, 2004).

3.2.4 Modelling Methods

As previously stated, the modelling methods compared are the following:

- Multinomial logistic regression (MLR)
- Error correcting output coding (ECOC)
- Multinomial bagging with logistic regression (MBLR)
- Multinomial random forests with decision trees (MRFD)

a) MLR

Category	Setting
Link Function	Logit
Reference Category	NO_TAKE
Data Partition Method	Simple Random Sampling
Variable Selection	Stepwise Selection
Model Selection	Validation Misclassification/AIC/BIC/AUC
Optimisation Algorithm	Newton Raphson
Software	SAS Enterprise Miner
Threshold Setting	Maximise True Positive Rate

Table 3: Table illustrating the modelling procedure undertaken using MLR

MLR was fit to the data using the settings as listed in Table 3 above. The software which was used to fit the data was SAS Enterprise Miner. As stated in the table above, the model selection was based on the misclassification rate of the test data set and the AIC. Both statistics were easily obtainable from each iteration of the model fitting process. For the parameter estimation, MLR used the maximum likelihood method. Since a closed solution cannot be obtained, Newton Raphson optimisation algorithm was applied to obtain a solution.

b) MBLR

Category	Setting
Underlying Modelling Method	Logistic regression
Number of Bagging instances	8
Probability Function	Average
Sampling Type	Bootstrap Sampling with replacement
Sampling Method	Simple Random Sampling
Model Selection	None
Software	SAS Enterprise Miner
Software Packages	Logistic regression, Ensemble

Table 4: Table illustrating the modelling procedure undertaken using MBLR

Modelling method MBLR was fit to the data using the settings as listed in Table 4 above. The software which was used to fit the data was SAS Enterprise Miner. The data was partitioned into training and test data sets as previously stated. The test data set was used for assessing the goodness of fit. The model aggregation method deployed was classification probability across the different models and the class with the highest probability was allocated to that customer as previously stated in the algorithm under the literature review section.

Algorithm Summary: (Hastie et.al, 2009), (Breiman; 1996a)

Following the model dataset creation process as set out under section 3.2.1:

- Data was split into training and test set in the proportions 75% and 25% respectively using simple random sampling
- Using MLR as the modelling algorithm, a prediction function $\phi(\mathbf{x})$ was obtained
- Took bootstrap samples from the training set to construct the predictor. The number of samples was limited to 8 due to size of the data.

$$\phi_{\mathrm{i}}(\mathbf{x})\,, i = 1, \ldots, 8$$

- The class which had the highest average probability of the $\{\phi_{\mathrm{i}}(\mathbf{x})\}$ was the bagged predictor. (Hastie et.al ;2009), (Breiman ;1996a)

c) MRFD

Category	Setting
Underlying Modelling Method	Decision trees
Number of trees	1000
Number of Variables at Each Split	4
Type of Forest	Classification
Variable Selection	Mean decrease in Gini
Model Selection	Out of bag error rate
Optimisation Algorithm	Newton Raphson
Software	R
Software Packages	randomforests,ROCR

Table 5: Table illustrating the modelling procedure undertaken using MRFD.

Applying the settings as listed in Table 5 above, a MRFD model was fit to the training data. Test datasets were used despite the literature insisting on the lack of need for a test data set due to the “out of bag” error estimation. Several authors such as Hastie *et.al* (2009) have shown a correlation in performance of a MRFD with the number of decision trees but only to a certain degree, hence the motivation to use one thousand decision trees. The Gini index was deployed as the variable importance calculation method.

The model was generally easy and relatively quick to fit. Gini was used for node splitting and three variables were attempted at each split. There was no adjustment of class weights at each node split save for under-sampling of the common class which was carried out earlier.

d) ECOC

Category	Setting
Underlying Modelling Method	Logistic regression
Number of logistic regression models	15
Coding Design	Hamming’s Distance
Coding Method	Exhaustive Search
Type of Model	Classification
Class Selection	Classification rate
Software	SAS
Software Packages	Base SAS, SAS Enterprise Miner, Logistic regression

Table 6: Table illustrating the modelling procedure undertaken using ECOC

The approach used in modelling the data via the ECOC was as follows:

- Partitioned the data set into training and test as described above.
- Each response class was assigned a unique binary string P of length 15 based on the exhaustive search method which minimises the Hamming distance as explained below.
- The 15 binary functions were “learned” using logistic regression, one for each bit position in the binary string.
- The 15 binary models were assessed to create a new string P^* of length 15 with each position having the predicted probability for that class.
- P^* was compared to each of the corresponding 15 position in P for each of the classes and the class which was closest based on the absolute error of mean square error was assigned that respective class.

Coding Strategy

Exhaustive Coding Method

As previously discussed in the literature review; in order to create a powerful code one needs to create a code that has maximum separation in the rows and in the columns. Exhaustive coding was applied,Dietterich and Bakiri (1995) showed that it minimised the Hamming's distance. The string of length 15 for each class is shown in Table 7 below. The table has a Hamming distance of eight and contains no identical or complementary columns.

Coding Matrix

Class	Product	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
b.INV	W_i	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
c.SL	W_2	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
d.UL	W_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
e.CARD	W_4	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
NO_TAKE	W_5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Table 7: Exhaustive coding approach for a five class case

In this instance, each product $\mathbf{W_j}$ where $\boldsymbol{j = 1,..,5}$ represent a unique code string $(T1 - T15)$.

Decoding Strategy

For the ECOC approach, each model output unit was viewed as computing the probability that its corresponding bit in the "code-word" is one. The mathematical formulae listed below details the classification procedure for each customer:

Let $i = customers \quad 1,2,\dots\dots..,m$

$$J = \; class \qquad 1,2..,5$$

$$L = binary\ models \qquad 1,2\ \dots.,15$$

$$T = coding\ matrix$$

$\mathrm{T_{jl}}$ is the observation in the coding matrix for the $\mathrm{j^{th}}$class and $\mathrm{l^{th}}$ model

$$e = 0,1$$

$\mathrm{p_{ile}}$ be the predicted probality for the $\mathrm{i^{th}}$customer for level e using the $\mathrm{l^{th}}$ model

$$\mathrm{b_{ij}} = \sum_{\mathrm{i=1}}^{15} \left|\mathrm{T_{jl}} - \mathrm{p_{il1}}\right|$$

- $\mathrm{Class(i)} = \mathrm{argmin_j\ b_{ij}}$

Having ascertained the absolute error, the customers were classified as shown in the decoding strategy.

3.2.5 Evaluation Criteria

Each model was assessed based on its ability to accurately classify the different products to cross sell to different customers, robustness and the ability to easily explain the interaction of the independent variables with the dependent variable on which the data are modelled. These assessment measures were previously discussed.

Area under the ROC curve (AUC)

Recall the AUC as proposed by Hand and Till (2000):

$$\boldsymbol{M} = \frac{2}{\mathrm{j(j-1)}} \sum_{\mathrm{c<d}} \hat{\mathrm{A}}(\mathrm{c,d})$$

Proportional by chance accuracy (Morrison, 1969)

In order to ascertain model as being useful, we compared the overall classification rate of the model to the proportional by chance accuracy. As a rule of thumb, for a model to be considered useful, it must have an accuracy rate which is at least 25% higher than the proportional by chance accuracy rate.

As an example of how to calculate proportion by chance accuracy, we show the steps:

Class Number	Class	Population Percentage
1	b.INV	25%
2	c.SL	6%
3	d.UL	25%
4	e.CARD	18%
5	NO_TAKE	26%

Table 8: Post sampling class distribution of the products taken up by customers during the observation period.

Table 8 above summarises the distribution of the data. It details the percentage of customers in each of the classes in the response variable. Based on the Table 8, we computed the proportional by chance accuracy by squaring and summing the proportion of customers in each group as:

$$\sum_{j=1}^{5} p_j{}^2 \quad \text{where } p_k \text{ is the proportion of customers in class j}$$

In this case, this resolves to

$$(0.25^2 + 0.06^2 + 0.25^2 + 0.18^2 + 0.26^2) = 0.23$$

If the value is then multiplied by a factor 1.25 this results in a benchmark of **0.286**.

Performance league table

The four different models were compared and contrasted using the four techniques listed below:

- Performance Accuracy (Hand, 2000)
- Model Generalisation/Robustness (Breiman, 1996b), (Hand, 2000)
- Variable Selection and Interpretability
- Ease of Use

A league table was constructed which ranked each model based on the metrics above. Each modelling method was ranked from first to last.

4 Results and Analysis

4.1 Introduction

This section initially discusses the data characteristics, sampling procedures and data manipulation. The modelling results for the individual models are then discussed. The model results are listed in the following order.

- MLR
- MBLR
- ECOC
- MRFD

4.2 Data Profiling Results

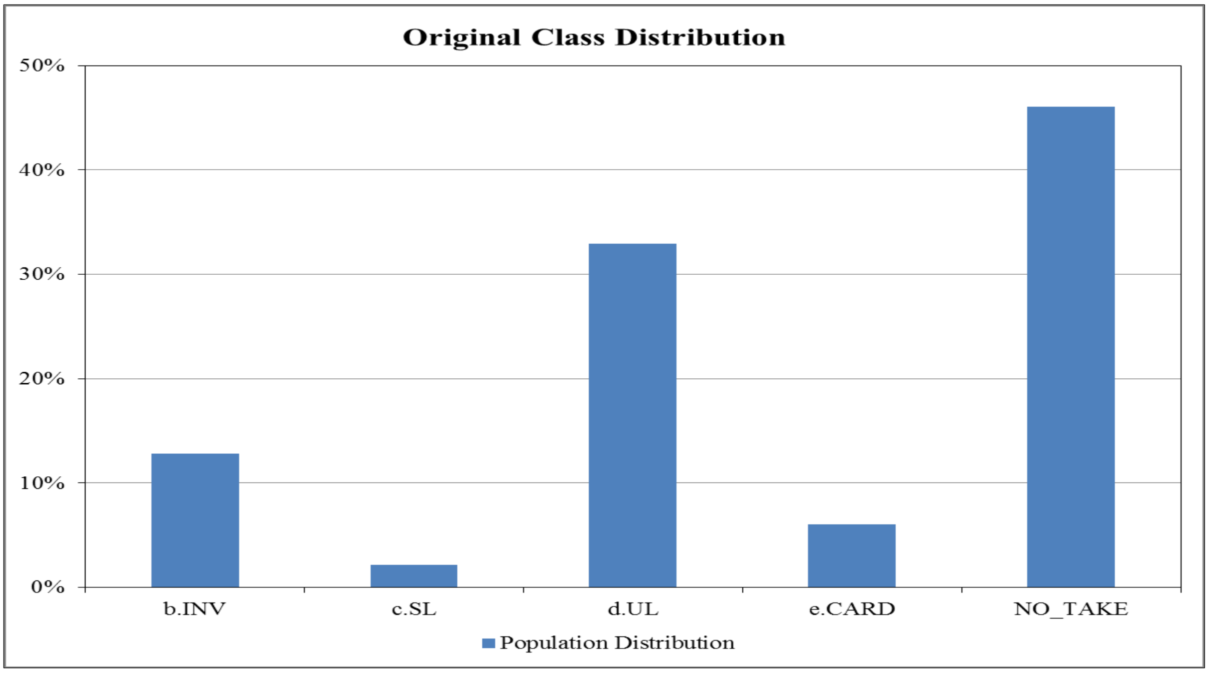

Figure 5: Class distribution of the products taken up by customers during the observation period.

Figure 5 illustrates the movements of the customers in the period of observation. If a customer did not take up any other product, this is classified as “NO_TAKE” and this constituted forty six percent of the twenty thousand and five hundred customers. Twelve percent of the customers took up an investment product “b.INV”, two percent took up a secured loan product “c.SL”, thirty five percent took up an unsecured lending product “d.UL” and five percent of the customers took up a credit card “e.CARD”. If no additional information is known about a subgroup of customers, one would expect them to be in the different states with probabilities illustrated by the Figure 5.

4.3 Sampling and Data Partitioning

Given the class imbalance as can be noted from Figure 5, the common classes were under-sampled especially the “NO_TAKE” class. After stratified sampling with unequal weighting, the maximum representation a class could attain in the data set was capped at a multiple of four times the size of the rare class and for those that could not reach the threshold, all the observations were used. The new dataset had 7400 observations from the original 20500 observations, with the rare class having 440 observations. The updated distribution of the data is represented in Figure 6 below.

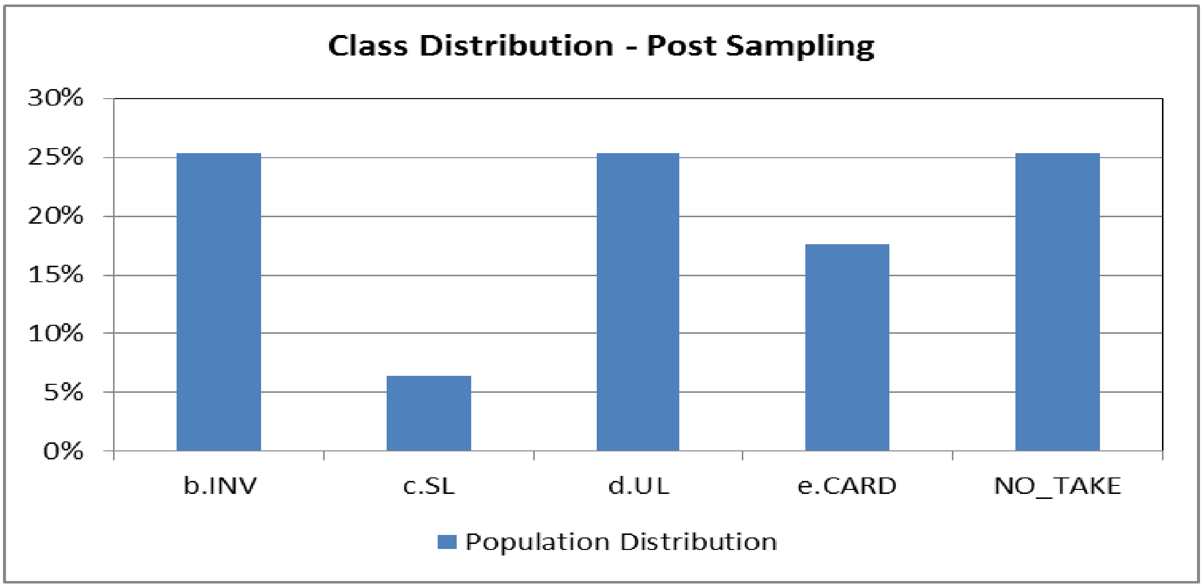

Figure 6: Post sampling class distribution of the products taken up by customers during the observation period.

The data as represented in Figure 6 was used for the modelling purposes. The direct implication of this procedure was the loss of information due to sampling (Georges, 2004).

4.4 Model Analysis

a) Multinomial logistic regression (MLR)

Null Hypothesis: The MLR model fitted performs no better than the null model with no variables.

Table 9 below; details the hypothesis test using the likelihood ratio test at five percent significance level.

Likelihood Ratio Test for Global Null Hypothesis: BETA=0

-2 Log Likelihood (Intercept Only)	-2 Log Likelihood (Covariates)	Likelihood Ratio Chi-Square	DF	Pr > ChiSq
13939	11394	2545	156	< .0001

Table 9: Hypothesis testing for the MLR model fit.

The conclusion is to reject the null hypothesis at five percent significance level and thus the model fitted is superior to the model with intercept only. It has to be noted that hypothesis testing has limitations when handling large data. Since the stepwise method was used for variable selection, the best model was found after eleven steps with the variables listed in Table 10. The entry criterion for variable selection was set at 0.05. The table also details the Wald statistic and the p values of the selected variables. The Wald test is a parametric statistical test for the significance of an independent variable in a statistical model. Using the five percent significance level, selected variables are listed in Table 10 and assists in explaining information about the different classes in the data. Table A1 in the appendix details the maximum likelihood estimates of all the variables selected in the model for each class.

Table A2 in the appendix also provides the odd ratio estimates of each variable relative to the class. Example E1 in the appendix illustrates how one can use the parameter estimates to predict the probability of an observation belong to a specific class.

Effect	DF	Wald Chi-Square	Pr > ChiSq
AGE_GROUP	16	128.8	<.0001
Acc_Age_band	20	108.1	<.0001
Ethnicity_Desc	20	64.3	<.0001
Gender_Type_Desc	4	51.6	<.0001
Marital_Status	24	89.2	<.0001
Preferred_Lang	8	33.9	<.0001
ave_month_end_bal	4	47.7	<.0001
average_atm	4	37.8	<.0001
contacts	12	197.2	<.0001
g_dep_band	16	48.6	<.0001
home_p	4	40.9	<.0001
salary_group	24	310.3	<.0001

Table 10: Variable selected for the MLR model using the Wald's Test.

After model fitting, the overall classification rate ranged on different samples between 43% and 44%. The classification table for the test data is shown in Table 11. The classification rate is consistent across the training, validation and test model sets. The consistency implies model robustness and ability to generalise very well. However, since the study is more concerned with correct prediction of the target class, we assessed the true positive rate of each predicted class.

		PREDICTED					
		b.INV	c.SL	d.UL	e.CARD	NO_TAKE	Row Total
TRUE	b.INV	130	3	70	63	91	357
	c.SL	16	10	16	34	13	89
	d.UL	45	2	170	47	92	356
	e.CARD	56	6	29	140	17	248
	NO_TAKE	55	7	80	53	162	357
	Column Total	302	28	365	337	375	1407

Table 11: Classification table for MLR

	Population Distribution	Precision Rate	True Positive Rate	Model Lift	Maximum Obtainable Lift
b.INV	25%	43%	36%	1.70	4.04
c.SL	6%	36%	11%	5.65	16.19
d.UL	25%	47%	48%	1.84	3.96
e.CARD	18%	42%	56%	2.36	5.63
NO_TAKE	26%	43%	45%	1.70	3.85

Table 12: Statistics table for MLR

It is clear from Table 12 above that the precision rate is consistently above the benchmark rate across all the different classes; population distribution being the benchmark rate. As an example, for the prediction of the class e.CARD, if the model fitted was performing no better than the random classification of customers, one would expect a precision rate of 18%, however using the model we obtained a precision rate of 42%, which is 2.36 times better than the benchmark rate. A perfect model would have obtained a lift of 5.63.

The benchmark proportion by chance accuracy was **0.286**.

Total Correctly Classified	612
Total Base Size	1407
Percent correctly classified	43%
Expected Correctly Classified (Proportional by chance)	28%
Maximum Chance Classification rate	32%
Net Lift	1.53

Table 13: Summary of the classification results with net lift representing the multiple of Proportional by chance relative to the percent correctly classified.

It is clear from Table 13 above that the classification accuracy is at least 50% higher than the proportional by chance accuracy benchmark. This implies that the model has potential.

The measure $\boldsymbol{M}$, measures the overall seperability between the different classes and produces a robust measure for model accuracy. Using the statistical software R and the statistical package named HandTill2001, the $\boldsymbol{M}$ measure for this model was found to be 0.76 for the validation data set and 0.764 for the test data set. This value implies a potentially good model since it is above 0.5 and the consistency between validation and test datasets implies model robustness.

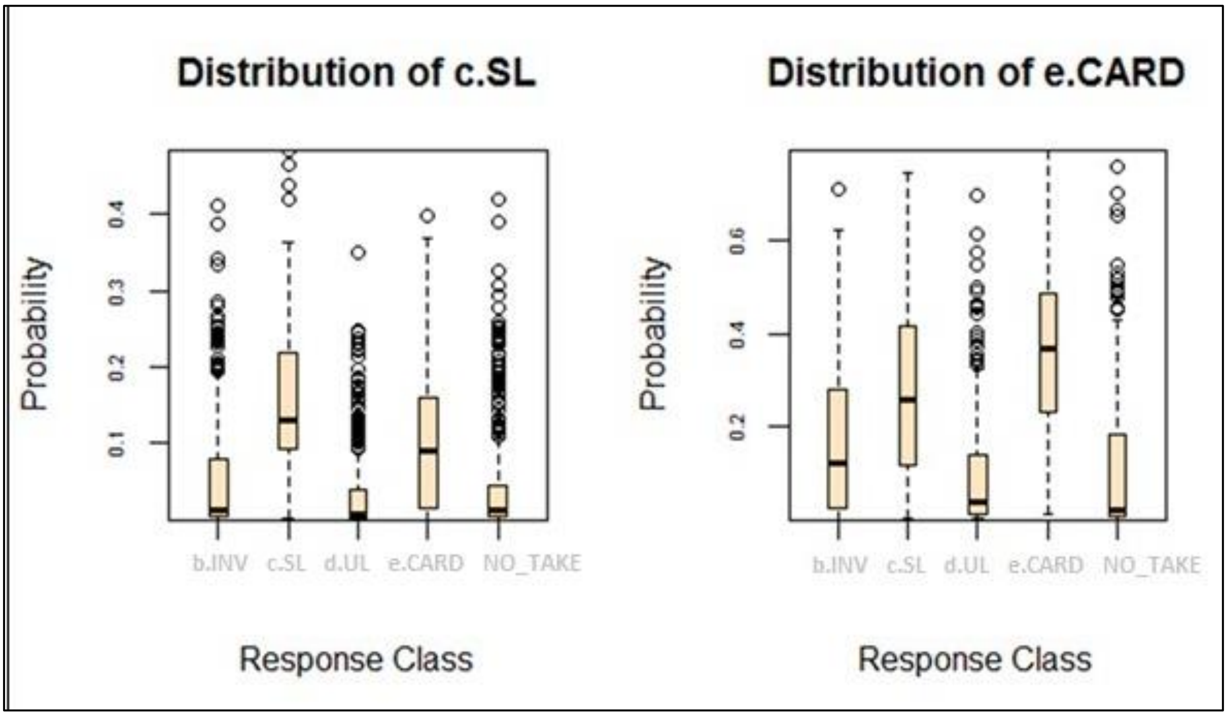

Figure 7: Probability distributions of c.SL and e.CARD across all the classes.

Although the model performs reasonably well, it appears not to fit rare classes very well as it can only account for 11% of the rare class c.SL. This implies that the model is misclassifying 89% of customers belonging to this class despite under-sampling the common classes. From Table 11, some of the individuals belonging to this class are being misclassified as belonging to the class e.CARD. As can also be seen from Figure 7 above, the probability distribution of the response class c.SL is distinctly higher than all classes but the same customers do have a high probability distribution for e.CARD as well. Due to the fact that e.CARD is well represented in the data; most customers belonging to class c.SL end up being misclassified as e.CARD. Intuitively, given the sampling time frames, it makes sense that the model might struggle to model this class as this relates to Home Loan and Vehicle Loan application which take longer to process and in some instances, the six months window period is not sufficient to accurately predict this class. Based on the data analysis done before, it was also observed that a significant proportion of Home Loan and Vehicle Loan applicants do acquire credit cards so as to pay for the deposit of the properties and related costs such as bond registration and transfer duties. This is indicative of the limitations of classification error as opposed to prediction error. Classification limits the model in the sense that it assumes that an observation can only be assigned to one class but this does not imply that the observation does not exhibit similar characteristics of other classes.

Several researchers such as Georges, (2004) note that most multi-classifier models tend to be biased towards the common class as they focus on overall predictive accuracy and this appears to hold true in this case. Chen, Tsai, Young and Kodell, (2005) even go further to state that over or under-sampling will not completely address this problem as can also be seen in this example. They propose cost sensitive algorithms which will impose heavy penalties on misclassifying the rare classes as opposed to the common classes whilst others such as Burez and Van den Poel (2009) propose methods such as weighting the class distributions to favour the rare class.

In this study, for simplicity, an ad-hoc solution was proposed which involved biased weighting of the rare class distribution so as to improve the precision of the rare classes without necessarily compromising the accuracy rates of the other classes. This method is discussed and applied in greater detail in the following paragraph. Greater emphasis was placed on the rare class since it was noted to be a low volume but a highly profitable product.

Biased weighting of the rare class distribution

Biased weighting of the rare class was carried out through multiplying the probability distribution of the response class c.SL by varying weighting factors. Figure 8 below; details the net effect of boosting the rare class and its effect on the true positive rates and overall classification rates. The biased weighting method improved the precision rate of c.SL although the precision rate of e. Card dropped. In this instance, the losses incurred through reduced accuracy for e.Card will have to be compared to the improved profits from the rare class. Recall that the proportion by chance benchmark is 28% and by using a weighting factor of two, one would still obtain a classification rate of around 43% whilst the true positive rate of rare class c.SL is increased to 34%, which is significantly better than the previous 11%. These results did hold for both the validation and test data sets.

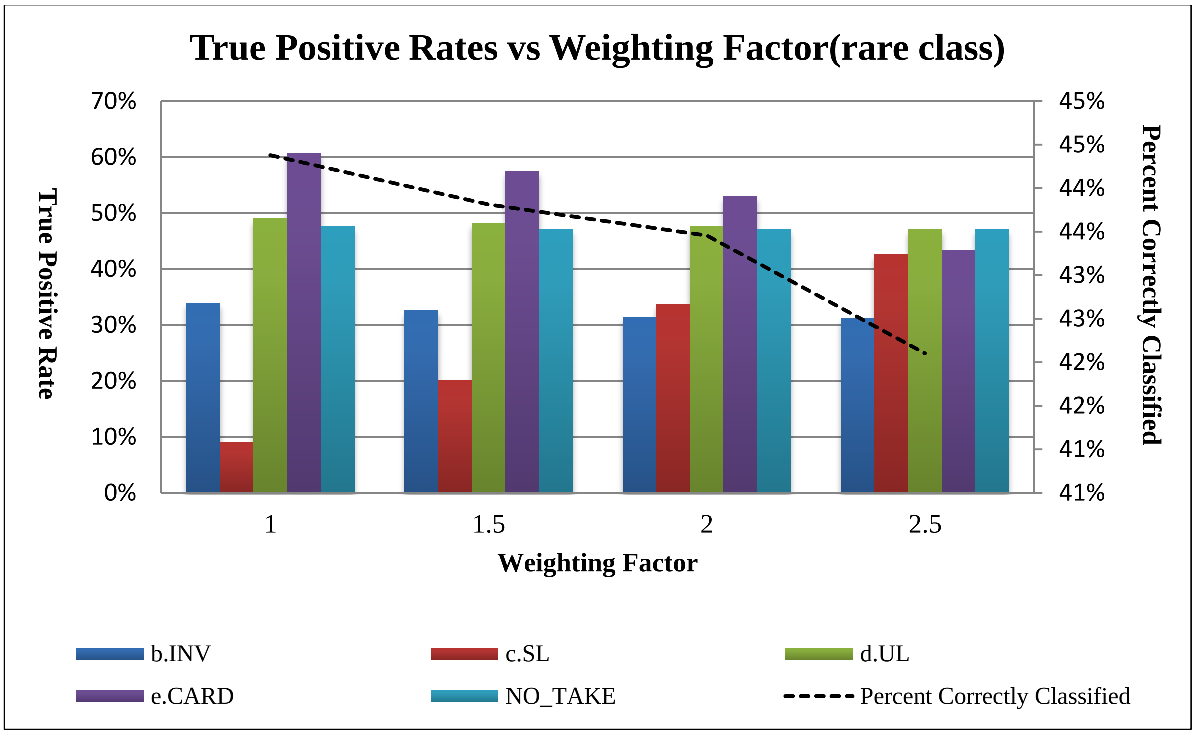

Figure 8: Post modelling classification rates across varying weighting factors. The results shown above were from a different test sample to that which produced results for Table 12.In the graph above, Percent Correctly Classified (PCC) on the right axis is the overall proportion of observations correctly classified. The True Positive Rate (TPR) on the left axis is a product specific proportion of observations correctly classified.

b) Multinomial bagging with logistic regression (MBLR)

Based on the null hypothesis in Section 4.4 (a) above, all the MLR models which were deployed rejected the null hypothesis at five percent significance level.

After the model fitting, the overall classification rate was found to be between 46% and 47%. The classification table for the test data is shown in Table 14 below. The classification rate was also found to be consistent across the training and test model sets. The consistency implies model robustness and ability to generalise well as above.

Strikingly, the results appear to be consistently similar to those obtained by the MLR although the accuracy has improved by three percentage points. This implies that MLR is a stable classifier and thus MBLR would exhibit a similar performance.

Since MBLR is a model averaging technique, it does not provide any additional insight into the variable selection procedure. Its main aim is to improve accuracy by reducing the variance in the error rate.

	PREDICTED						
		b.INV	c.SL	d.UL	e.CARD	NO_TAKE	Row Total
TRUE	b.INV	171	3	94	89	89	446
	c.SL	11	13	20	52	15	111
	d.UL	65	4	224	46	107	446
	e.CARD	60	2	44	191	12	309
	NO_TAKE	69	4	100	65	208	446
	Column Total	376	26	482	443	431	1758

Table 14: Classification table for MBLR

	Population Distribution	**Precision Rate**	**True Positive Rate**	**Model Lift**	**Max Obtainable Lift**
b.INV	25%	45%	38%	1.79	3.94
c.SL	6%	50%	12%	7.92	15.84
d.UL	25%	46%	50%	1.83	3.94
e.CARD	18%	43%	62%	2.45	5.69
NO_TAKE	25%	48%	47%	1.90	3.94

Table 15: Statistics table for MBLR

From Table 16, the percent correctly classified has improved to 46% compared to 43% from the previous model. This implies that the classification rate is now at least 60% above the benchmark proportional by chance classification. This classification rate is consistent across all the data sets. The precision and true positive rates are slightly higher than those obtained in the previous model.

Measuring the AUC as proposed by Hand and Till (2001), the ***M*** measure is found to be 0.79 for the test set and 0.78 for the validation data set. This result does imply that the models are generally useful in terms of class seperability. The previous model, **MLR**, had a ***M*** measure of 0.76, which implies that the current model is better.

Total Correctly Classified	807
Total Base Size	1758
Percent Correctly Classified	46%
Expected Correctly Classified (Proportional by chance)	28%
Maximum Chance Classification rate	32%
Net Lift	1.61

Table 16: Summary of the classification results with net lift representing the multiple of Proportional by chance relative to the percent correctly classified.

Regarding the rare class c.SL, MBLR encounters a similar problem as that of individual MLR models. This is not surprising as the problem emanates from the algorithm thus highlighting one of the major weaknesses of MBLR. From the assessment of the results, MBLR assists in model generalisation and improved overall accuracy but does not address the deficiencies which are algorithm specific. In fact, it might actually magnify the deficiency especially if the same algorithm is being applied leading to the same deficiency being replicated. Authors such as De Bock *et.al* (2010) argue that in instances of MBLR, during the creation of the bootstrap samples, some classes can be rare such that they become insignificantly represented in the sample and thus the modelling algorithms cannot account for them appropriately. This issue may apply in this instance, since the rare class c.SL only constitutes six percent of the population and eight bootstrap samples are derived from this data.

Biased weighting the model scores for the rare classes achieved similar results as in the previous section but due to the increased accuracy of MBLR, a higher weighting factor was adopted but still maintains at roughly 44% overall accuracy rate whilst the true positive rate of the rare class improved to 34%. Due to the higher initial accuracy rate, this allowed one to be more aggressive with the weightings.

c) Multinomial random forests with decision trees (MRFD)

For the model fitted by MRFD, the classification rate obtained based on the out of bag sample was 47%, which is four percentage points higher than the MLR model. The error rate was consistent with the test datasets, which is a good indicator of model generalisation. Variable selection was carried out using the out of bag sample as discussed in the literature review section. The variable selection procedure was based on the mean decrease in accuracy of the model as each variable is removed. The same procedure was applied using the mean decrease in Gini of the model. Figure 9 below, details the variables used in the model, in descending order.

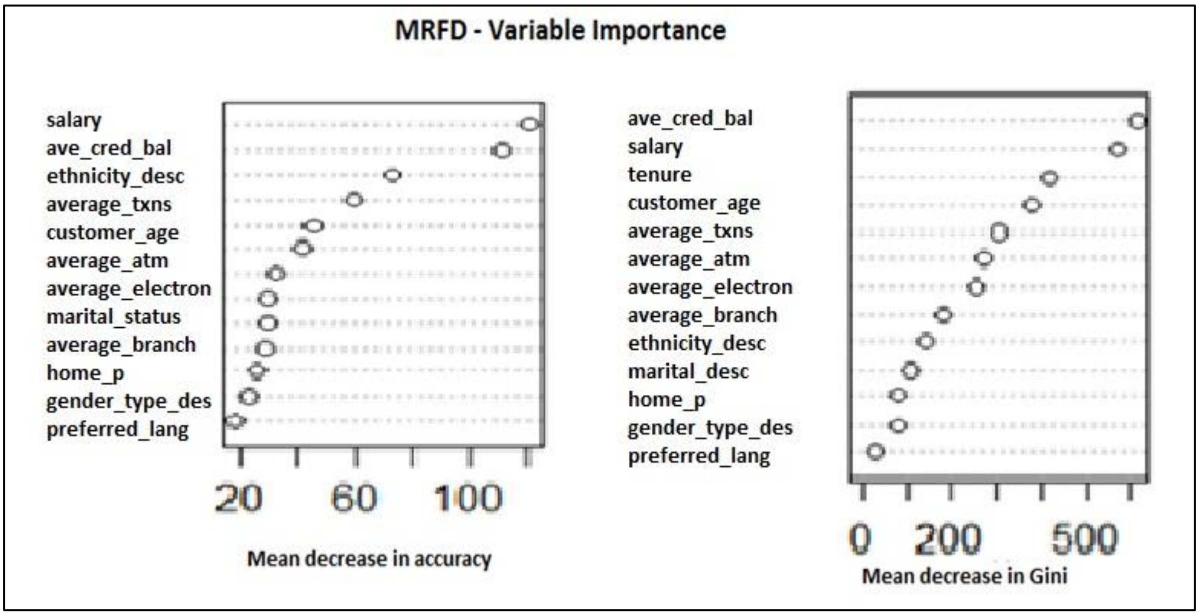

Figure 9: Variables selected from the MRFD model and their relative importance.

It is clear from the plot in Figure 9 that the salary earned by a customer, the relationship length with the bank (tenure) as well as his average credit balance over a six month period have the biggest influence in the model. Customer ethnicity and how frequently the customer transacts also play a big role. Similar to the MLR model above, the MRFD model also details how each of the variables relate to the different response classes. A detailed listing is provided in the Appendix. Similar to MBLR, the MRFD model is also dependent on the algorithm implied and the inherent shortcomings of the algorithm will be reflected in the final result albeit to a lesser extent.

Based on Table 17 and Table 18, it is clear that MRFD classifies the response class d.UL better than the previous models but is similar to other models in prediction of the other classes. The percent correctly classified is at roughly sixty percent (60%) higher than the proportional by chance accuracy.

		PREDICTED					
		b.INV	c.SL	d.UL	e.CARD	NO_TAKE	Row Total
TRUE	b.INV	191	6	100	95	85	477
	c.SL	28	3	9	59	21	120
	d.UL	56	0	255	60	79	450
	e.CARD	59	5	26	191	19	300
	NO_TAKE	72	4	103	61	172	412
	Column Total	406	18	493	466	376	1759

Table 17: Classification table for MRFD

	Population Distribution	Precision Rate	True Positive Rate	Model Lift	Max Obtainable Lift
b.INV	27%	47%	40%	1.73	3.69
c.SL	7%	17%	3%	2.44	14.66
d.UL	26%	52%	57%	2.02	3.91
e.CARD	17%	41%	64%	2.40	5.86
NO_TAKE	23%	46%	42%	1.95	4.27

Table 18: Statistics table for MRFD

Total Correctly Classified	802
Total Base Size	1759
Percent Correctly Classified	46%
Expected Correctly Classified (Proportional by chance)	28%
Maximum Chance Classification rate	32%
Net Lift	1.60

Table 19: Summary of the classification results with net lift representing the multiple of Proportional by chance relative to the percent correctly classified.

Measuring the AUC as proposed by Hand and Till (2001), the ***M*** measure for the MRFD is 0.74 which is four percentage points lower than the one obtained by the MBLR methodology. The drop in the AUC is likely due to the lower than expected true positive rate of c.SL compared to that obtained using MBLR. Although the measure is lower, it still exhibits a good level of seperability. It should also be noted that MRFD are excellent in reducing the overall error rate by reducing the overall "noise" in the model estimates.

Due to its emphasis on overall accuracy rate, classification of the rare classes is sacrificed in pursuit of greater accuracy in common classes. It can also be noted that having decision trees as the base algorithm which apply the "greedy algorithm" could be a limitation. The literature review mentioned the sub-optimality of such an approach and De Bock *et al* (2010) note that this deficiency cascades down to the overall MRFD. This approach might lead to the factors relating to the classification of the rare classes being suppressed in relation to the other classes.

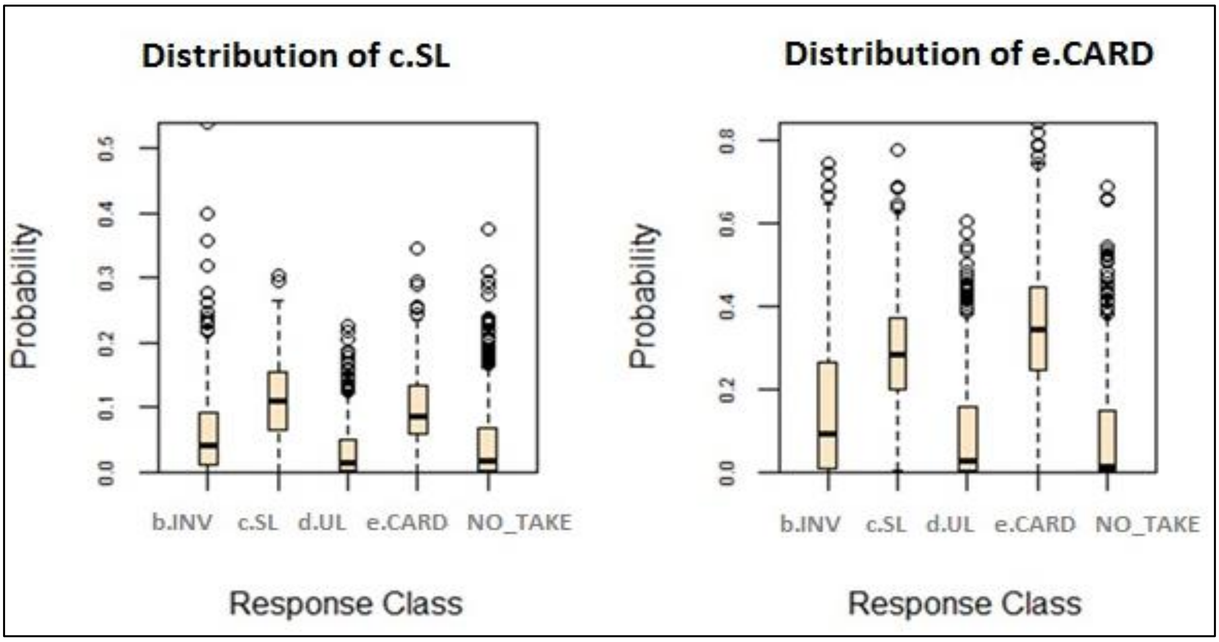

Figure 10: Predicted probability distributions of the response classes c.SL and e.CARD obtained from the MRFD model.

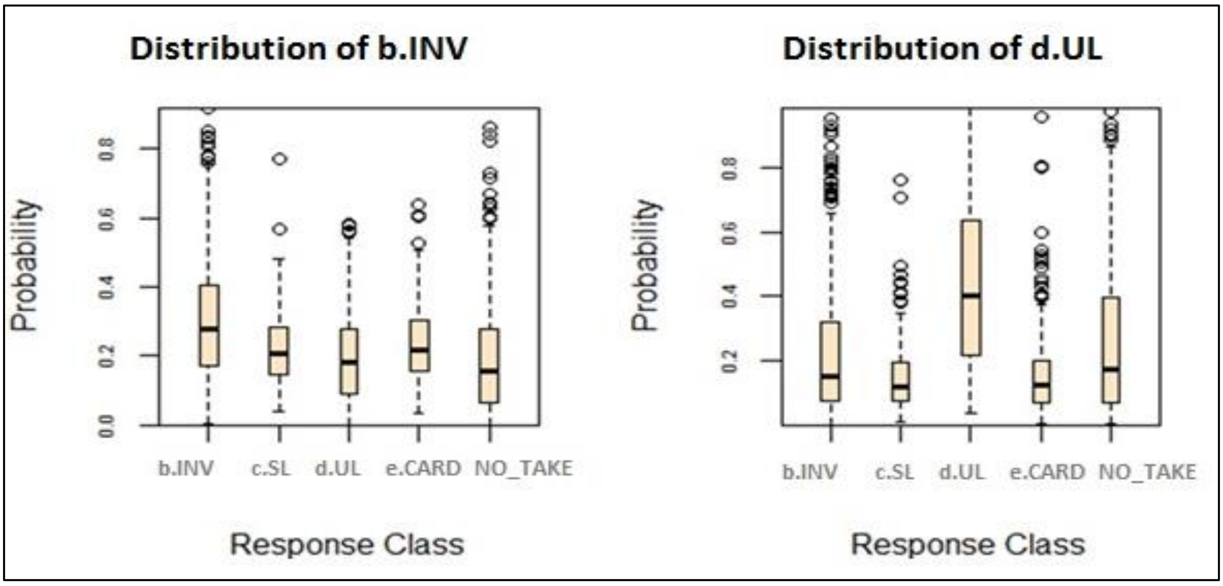

Figure11: Predicted probability distributions of the response classes d.UL and b.INV obtained from the MRFD model.

Figure 10 and Figure 11 above; detail the predicted probability distributions of the different response classes obtained from the MRFD model. It is clear that the class d.UL achieve the highest level of seperability between class and hence the high true positive rates. The response class b.INV appears to show relative degree of accuracy as well but of concern is the distrubutions of both e.CARD and c.SL. Figure 10 above clearly shows the correlation between the estimated distribution of these two classes which suggest that the factors used in their predictions are similar. Due to the response class e.CARD being the common class, its distribution tends to dominate that of the rare class c.SL leading to a significant amount of customers in this response class being misclassified as e.CARD. This is also further compounded by the fact that it is less expensive to misclassify a rare class than a comon class leading to the model being biased towards the common class.The model however is able to distinctly separate the rare class c.SL from the other response classes namely b.INV and d.UL.

In order to address the issue of seperability, it was futher proposed to build a series of binary classification models in order to increase the degree of seperability between the c.SL and e.CARD. In that instance,one could artificially over-sample the rare class in order for the model to distinquish the classes better. Burez and Van den Poel (2009) propose artificially altering the class weights at the terminal node so as to boost the rare class distributions.

d) Error correcting output coding (ECOC)

The classification table for the ECOC model is shown in Table 20 below. The precision rates are high for all response classes, with almost all precision rates being at least twice the benchmark rate as shown by the column, **Model Lift**. The true positive rate is high for all classes except the rare class (c.SL).

The true positive rate of the response class (c.SL) is a concern as the model is failing to account for the rare class. As evidenced by the precision rate of the common classes, the ECOC is “greedy” in terms of pursuing overall classification rate whilst sacrificing the rare classes regardless of the under-sampling of the common classes previously described.

		PREDICTED					
		b.INV	c.SL	d.UL	e.CARD	NO_TAKE	Row Total
TRUE	b.INV	192	58	78	73	46	446
	c.SL	11	44	8	39	10	112
	d.UL	53	45	251	53	45	446
	e.CARD	35	68	24	181	3	310
	NO_TAKE	77	38	107	53	172	446
	Column Total	367	252	467	399	275	1759

Table 20: Classification rates

	Population Distribution	**Precision Rate**	**True Positive Rate**	**Model Lift**	**Max Obtainable Lift**
b.INV	25%	43%	52%	1.72	3.94
c.SL	6%	40%	18%	6.7	15.77
d.UL	25%	56%	54%	2.24	3.94
e.CARD	18%	58%	45%	3.2	5.69
NO_TAKE	25%	39%	62%	1.56	3.94

Table 21: Classification table for the ECOC model

From Table 22 below, one can observe the percent correctly classified is 48% which is the highest across all the models fitted. Using the proportional by chance classification as the benchmark, the error correcting output coding model is over 70% above this benchmark rate.

This model performance holds for both the validation and the test datasets implying high model generalisation. Dietterich and Bakiri (1995) argue that, by having the independent binary functions modelling on the data; they are modelling for different aspects in the data and thus would generally outperform other multi-class models as in this instance.

Total Correctly Classified	867
Total Base Size	1759
Percent Correctly Classified	48%
Expected Correctly Classified	28%
Maximum Chance Classification	32%
Net Lift	1.71

Table 22: Summary of the classification results with net lift representing the multiple of Proportional by chance relative to the percent correctly classified

Although the model has a high accuracy rate, the true positive rate for c.SL is a major concern. Artificial reduction of the error for the class c.SL was undertaken by multiplying b_{ij} for class c.SL by a weighting factor of 0.75 so as to separate its predictions from those of e.CARD. By using this approach, the true positive rate for c.SL increased to 33% from 18% but the overall accuracy dropped to 44% which is still significantly higher than the benchmark proportional by chance accuracy of 28%.

Another major drawback of this method is that the algorithm does not produce a probability estimate and the question is whether the absolute error can be used as a proxy for probability. Having probability estimates, one can easily calculate the expected misclassification costs as well as deriving other statistics such as the estimated class distributions and the confidence limits of the estimates. Dietterich and Bakiri (1995) state that if the difference between the second lowest error and lowest error is huge for those classes correctly classified, it follows that the algorithm has high confidence in its classification. The calculation of the cumulative distributions of the difference in distance is shown below:

Procedure:

- The ECOC was fitted to the test data and the absolute error was calculated for each observation as previously shown
- The difference between the two lowest error classes was calculated as shown below:

Recall that:

$$b_{ij} = \sum_{i=1}^{15} \left|T_{jl} - p_{il1}\right|$$

Let:

- $L_i = \mathrm{argmin}_j\, b_{ij}$
- $S_i = \mathrm{second\ argmin}_j\, b_{ij}$
- $\delta = \begin{cases} 1 & \text{correct classification} \\ 0 & \text{incorrect classification} \end{cases}$
- Difference $D_i = L_i - S_i$
- $D_i(\delta) = D_i$ for classification δ

Where $\sum_i D_i(1)$ is the cumulative distribution of correctly classified (labelled "cumulative % (correctly classified)" in the graph)

$\sum_i D_i(0)$ is the cumulative distribution of incorrectly classified (labelled "cumulative % (incorrectly classified)" in the graph)

- The test dataset was then sorted based on the difference D_i from the largest value down to the lowest regardless of classification.
- The test data set was then ranked into pentiles, with pentile 1 representing the highest values and pentile 20 the lowest values.
- The cumulative distribution of each class $\sum_i D_i(\delta)$ was then derived and plotted as shown in Fig 13 below:

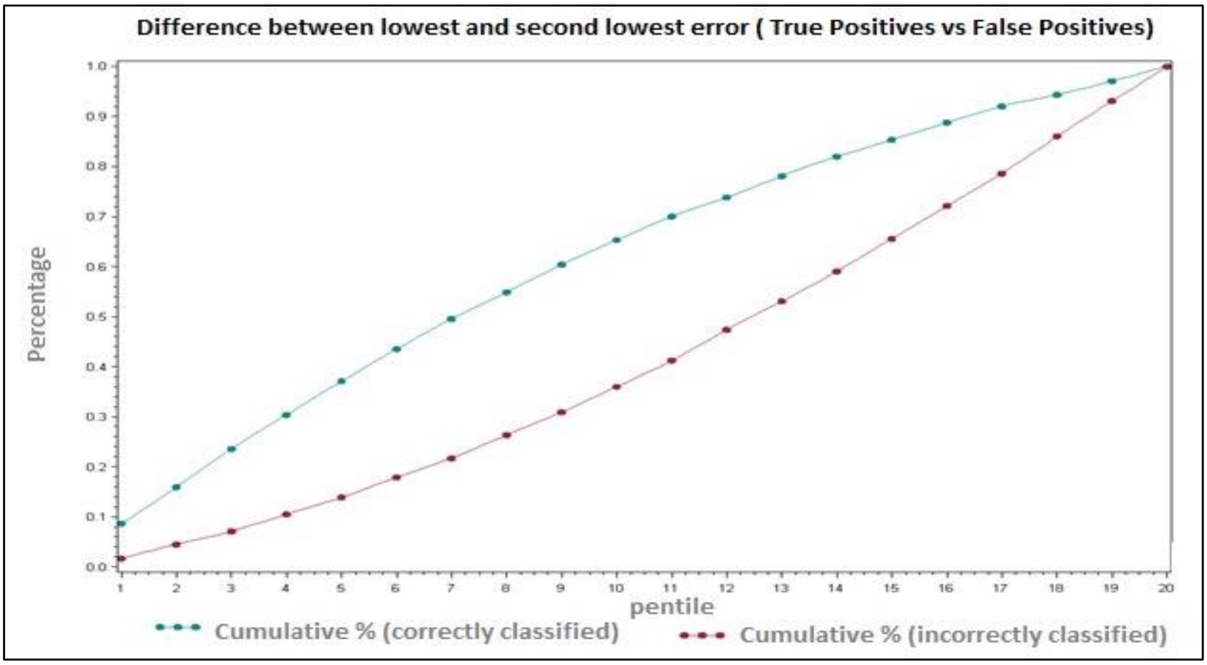

Figure 12: Plot of the difference between the lowest distance measure and the second lowest.

If the ECOC model did not have confidence in its classifications one would expect the cumulative distribution of the correctly classified class to be either, lower or at most very close to the distribution of the incorrectly classified. As can be seen from Figure 12, the graph for the correctly classified is distinctly above that of the incorrectly classified across the whole pentile range which indicates high "absolute deviation values" for the correct classifications. As an example, if one would take pentile 10, for the correctly classified the roughly 65% of these are already accounted for whereas for the incorrectly classified, roughly 36% are accounted for. This statistic is a clear indicator that the model has confidence in its correct classifications.

5 Discussion

5.1 Model Comparison

This section details the comparison of model performance across the various benchmarks mentioned in the methodology section. Each model is allocated a ranking relative to the metric being assessed with one being the highest and four being the lowest. Some of the metrics were given a higher weighting than others due to the objectives being fulfilled. The weights are subjective and dependent on the modeller's objective. The rankings were aggregated and a performance league table constructed in order to rank the models. The model with the highest ranking in the league table was classified as the best model to classify customers.

a) Performance Accuracy

	PCC	M measure	Rare Class (PPV)- before	Rare Class (PPV)-using biased weights
MLR	0.43	0.76	0.11	0.34
MBLR	0.46	0.79	0.12	0.34
MRFD	0.47	0.74	0.03	0.30
ECOC	0.48	0.77	0.18	0.33

Table 23: Summary of the classification results

As can be noted from the Table 23, ECOC has the highest overall classification rate. This is consistent with the findings from other researchers such as Dietterich and Bakiri (1995), whereas the MLR has the lowest overall classification rate. Although the ECOC has the highest classification rate, it is the within-class classification rate which poses concern. It has been noted that whilst achieving the highest overall classification rate, the ECOC tended to perform comparatively well in classifying the common class but however relatively poorly in the rare classes. This issue can generally be called "taking the safe bet", whereby, the model is not be penalised heavily for misclassifying the few records in the rare class as opposed to misclassifying the larger common classes. As previously noted, classification also limits the model by assuming that an observation can only be assigned to one class but this does not imply the observation does not exhibit characteristics similar to other classes.

MBLR had the highest $\boldsymbol{M}$ measure, which implies that it is the best model in achieving class seperability across the whole data set. It was difficult to obtain the $\boldsymbol{M}$ measure for the ECOC since it does not give probability outcomes but rather a distance measure between classes. Probability was estimated by standardising the distance between classes. MLR had the lowest overall classification rates but however, it attempts to balance the overall classification rate with the within-class classification rates. It can also be noted that the adjusted overall classification rates are relatively similar after taking into account the weighted classification rates of the rare class.

Overall, in terms of the overall classification rate, $\boldsymbol{M}$ measure, within class true positive rates and the ability to handle rare classes, the models were assigned an overall performance ranking. The MLR was assigned a ranking of 4, MBLR a ranking of 2, MRFD a ranking of 3, and ECOC a ranking of 1.

b) Model Generalisation

In terms of model generalisation, all the models appeared to generalise very well and the issues of over or under-fitting were not encountered. Across the training, validation and test data sets, the model results were obtained with a fair degree of similarity in accuracy. The MRFD has no need for the test data sets since it obtains its goodness of fit statistics as well as variable importance from the out of bag sample.

The $\boldsymbol{M}$ measures calculated from the training datasets were replicated in the test datasets and were found to be statistically similar. It is also clear from the improved model performance obtained by the MBLR method, that it is improving the model generalisation of the MLR by reducing the variance of the posterior probability estimates. If this did not suffice, the MBLR methodology would have obtained identically similar $\boldsymbol{M}$ measures and model accuracy rates.

It is also clear from Figure 12 that the ECOC achieves a great degree of generalisation across the data as similar plots were obtained from the test data sets.

A high $\boldsymbol{M}$ measure indicates the ability of the models to generalise very well across the whole dataset in terms of class seperability. MBLR had the highest $\boldsymbol{M}$ measure followed by ECOC. ECOC also produced a highly confident classifier with good levels of seperability as illustrated by Figure 12 above.

Therefore, in terms of model generalisation, MLR was assigned a ranking of 4, MBLR a ranking of 1, MRFD a ranking of 3, and ECOC a ranking of 2.

c) Variable Selection and Interpretability

Variable selection was met with varying challenges across the different models. For logistic regression, the stepwise methodology was used and variable selection was very transparent. The parameter estimates were also easy to interpret and the Wald's statistic was used to ascertain variable importance. The interaction of the variables with the different classes was easily obtained using the odds ratio estimates and thus easy to make deductions. Variable interaction was limited to a level of two but if one wanted to increase the levels, it was easy to do so in logistic regression. Due to the simplistic structure of the MLR and its assumption of linearity, one can easily calculate a customer's probability of falling into a specific class given all the variables used by the model. This is very important as it allows the modeller to ascertain the reason of achieving a specific probability score on an individual customer level.

MBLR is a model aggregation method which puts greater emphasis on model accuracy and model generalisation by decreasing the variance of the probability scores. It places greater emphasis on minimising the variance of the posterior probabilities. It is not easy to obtain variable importance under MBLR as one has to assess the individual input models to understand the important variables. MBLR models were found not be easy in interpretation as they appear to be a "black box" and it is difficult to ascertain as to the reason for varying probabilities within a specified group of customers without reverting to the underlying models. Hastie *et al* (2009) have shown that one can create partial dependence plots in order to ascertain variable importance in the MBLR procedure. These plots however do not provide much insight into variable interactions.

Although MRFD is also a model aggregating method, its great advantage is its ability to provide model parameter estimates as well as the variable importance measures. Variable importance measures were easily obtainable for each class as well as for the overall model. A variable importance plot was also obtainable as shown in Figure 9 above. One was also able to ascertain the number of variables used at each splitting node for the model building process. The variable importance measures were obtained from the mean decrease in Gini in the out of bag sample data.

ECOC provided little or no insight regarding variable importance or selection. The ECOC method is a strict “black box” approach which is aimed at improving overall classification accuracy. Model interpretability is also very difficult under the ECOC approach.

Therefore, in terms of model variable selection, interaction of variables and model interpretability, MLR was assigned a ranking of 1, MBLR a ranking of 3, MRFD a ranking of 2, and ECOC a ranking of 4.

d) Ease of Use

The principle of parsimony is very important regarding model deployment. The selected model should be easy to understand as well was simple to deploy without incurring large costs in terms of resources.

The MLR model is widely used and thus a vast amount of research around this subject is available. Due to its simplistic structure such as the assumption of linearity, the MLR was easy to understand and very computationally efficient. It is widely available in various software packages and thus easy to deploy. It also provided the user, the ability to vary a lot of settings such variable selection methods, variable entry threshold setting as well as the ability to easily adjust for sampling bias by taking into account the prior distribution of the marginal response class distributions. It also provided class probabilities and thus providing ability for the modeller to choose a probability threshold for classification.

The MBLR methodology was also easy to understand although it required more effort to construct as opposed to a single MLR model. The methodology requires one to fully assess the model goodness of fit for all the models being aggregated. This tended to be time consuming. Once developed, they are relatively easy to deploy and are similar to the MLR since they would be using the same data. MBLR also provided probability estimates and thus allowing one to adjust the classification thresholds. However, the model did not provide as many options as the MLR since it is a model aggregation method. One was allowed an option to choose the class voting method as well as varying the sample sizes during bootstrap sampling. MBLR is also widely available in the literature and several articles detailed various software packages available for use.

MRFD were found to be more difficult to understand and are not catered for by various software packages. This does not however affect deployment but is a limitation for the model consumers. The package used to develop the model provided various options for the modelling process but still the options were not as robust and efficient as the MLR. The MRFD provides probability estimates thus providing the user the ability to set the thresholds.

ECOC was found to be difficult and time consuming to code and decode for the different classes. A total of 15 models had to be built for each of the binary classes in the code. ECOC proved the hardest to implement as it required a lot of resources to deploy. It is not easy to deduce the reason for a customer specific classification as it gave little insight into the important variables. The model does not provide probability estimates but however the distance measure provided a good proxy with a great degree of confidence. Of all models, ECOC would prove the most difficult to implement. Literature on ECOC is not as widely available as the other models previously mentioned above.

Therefore, in terms of model efficiency, ease of use, literature availability and threshold setting, MLR was assigned a ranking of 1, MBLR a ranking of 2, MRFD a ranking of 3, and ECOC a ranking of 4.

5.2 Model Choice – Performance Table

Having discussed model performance relative to the varying metrics listed in the previous section, a quantifiable aggregate was required to choose the best model to address the objectives of the modelling exercise. A performance league table was thus constructed which would classify and rank the model based on the metrics.

	Performance accuracy	**Model generalisation**	**Variable selection and interpretability**	**Ease of use**
MLR	4	4	1*	1*
MBLR	2	1*	3	2
MRFD	3	3	2	3
ECOC	1*	2	4	4

Table 24: Performance ranking of the models. The ranking order is from 1 to 4, with 1 being the highest ranking. The asterisk indicated the best model for the respective metric.

From the performance Table 24, MBLR provided the best model fit across the varying metrics. MBLR is therefore the model of choice.

5.3 Conclusion

It is clear from the study that classification algorithms have a potential to improve product classification within the financial services industry. On average, all the models fitted are performing at least two and half times better than random classification.

Overall, MBLR was chosen as the model of choice based on the performance ranking as set out. However, performance ranking is subjective and dependent on objectives of the modelling exercise. Varying the score allocations can significantly vary the results. Model performance is highly dependent on area of study and as such, different results could have been obtained in other areas.

Based on methodology of the dataset creation, the models are immune to seasonality as the data was sampled from various time frames. One could also monitor model performance over time and statistically test for seasonality in the model performance. Although some variables are affected by seasonality, the variable averaging techniques deployed in the data as well as the correlation tests carried out assisted in countering seasonality. Although correction for sampling bias impacted model performance, it is key to note that all models had performed distinctly better than chance.

Prediction error, proved to be a good statistic in assessing model performance and so did the AUC ($\boldsymbol{M}$ measure) as proposed by Hand and Till (2001).

All the models fitted did not handle the rare classes very well despite under-sampling the common classes. The conditional distribution of the predicted response class for the rare class was very good but however, the same customers had a similar distribution for one of the common classes and this resulted in high misclassification. This implies that, either one can induce biased weighting in favour of the rare classes or assign the predicted class of the rare class in such a way that they do not compete with other classes except the non-take-up class.

5.4 Further Work

In this study, the model assessment was subjective and visual, thus another researcher might find slightly different conclusions. Scientific benchmarks could be developed for model assessment. A framework could be developed which sets the generally accepted data partitioning ratios, prediction error rates, classification rates and AUC. Research on benchmarks has tended to be skewed towards the binary classifiers and more work is required for multi-class instances.

An assumption was made which stated that during the observation period, the order of product take up is not important. It was further assumed that the time elapsed between the start of the observation period and the product take up is not important. The time elapsed till product take up could provide useful information to the modeller or the marketer. By ignoring the waiting time, it is difficult to assess the optimal time to engage a customer. Further work could be done to reweight the probability estimates to take into account the proximity of the event. This is similar to modelling for the hazard rate in survival modelling.

This study ignored multiple product take up. If a customer took up multiple products during the observation period, only the first was considered. This could have resulted in a correct classification being unwittingly mislabelled as a misclassification. An interested researcher could consider setting up a two stage classifier. The first stage could consist of classifying a customer’s ability to take up multiple products. An example is shown below:

$$\mathrm{Y} = \begin{cases} 2 & \text{customer opens multiple products} \\ 1 & \text{customer opens only 1 products} \\ 0 & \text{customer does not take up any product} \end{cases}$$

In the second stage, for those who have a high propensity to take up only one product, build a multi classifier model as before. For those customers with high propensity to take up more than one product, construct different combinations of baskets of products and formulate a multi-class classifier to model for these baskets. The basket sizes could be limited to a maximum of two products in order to minimise the different combinations of products.

6 References

Adams, N. M. and Hand, D.J. (1999): Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition, 32,* pp. 1139–1147.

Adams, N. M. and Hand, D.J. (2000): Improving the practice of classifier performance assessment. *Neural Computation,12*, pp. 305–311.

Bickell P.J, Ren J.J (2001): The bootstrap in hypothesis testing. In *State of the Art in Probability and Statistics, Fetschrift for William R. Van Zwet*, IMS Lecture Notes Monograph Series, Beachwood, OH, USA 36, pp. 91-112

Böhning, D (1992): MLR Algorithm***. *Annals of the Institute of Statistical Mathematics 44,No.1*, pp. 197–200.

Bradley, A.P., (1997): The Use of the Area under the ROC Curve in the Evaluation of Machine Learning Algorithms, *Pattern Recognition*, 7, pp. 1145-1159.

Breiman, L. (1996a): MBLR predictors. *Machine Learning* 24, pp. 123–140.

Breiman, L. (1996b): Heuristics of instability and stabilization in model selection. *Annals of Statistics 24*, pp. 2350–2383.

Breiman, L. (2001a): MRFD s. *Machine Learning,* 45, pp 5-32.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984): *Classification and Regression Trees*. Boca Raton, FL. Chapman and Hall/CRC.

Burez, J. and Van den Poel, D. (2007): CRM at a pay-tv company: Using analytical models to reduce customer attrition by targeted marketing for subscription services. *Expert Systems with Applications, 32(2),* pp. 277–288

Burez, J., and Van den Poel, D. (2009): Handling class imbalance in customer churn prediction. *Expert Systems with Applications, 36,* pp. 4626–4636.

Chen, C., Liaw, A., and Breiman, L. (2004): Using MRFD s to learn imbalanced data. *Technical Report 666, Statistics Department, University of California at Berkeley*.

Chen, J.J., Tsai, C.A., Young, J.F., Kodell, R.L. (2005): Classification ensembles for unbalanced class sizes in predictive toxicology. *SAR and QSAR in Environmental Research*, Vol 16, pp. 517—29.

Cios, K.J., Kurgan, L. A., Pedrycz, W. and Swiniarski, R. W. (2007): *Data Mining: A knowledge discovery approach.* Springer.

Croux, C. and Lemmens, A., (2006): MBLR and boosting classification trees to predict churn. *Journal of Marketing Research.*

Czepiel, S.A., (2002): *Maximum Likelihood Estimation of MLR Models*: Theory and Implementation, [online], available: *http://czep.net/contact.html*

De Bock, K. W., Coussement, K., and Van den Poel, D. (2010): Ensemble classification based on generalized additive models. *Computational Statistics and Data Analysis*, 54(6), pp. 1535–1546.

De Bock, K.W. and Van den Poel, D. (2012): Reconciling performance and interpretability in customer churn prediction using ensemble learning methods. *Expert Systems with Applications*, *39 (8),* pp. *6816-26.*

Demšar, J. (2006): Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research*, *7*),pp. *1-30*.

Dietterich, T. (2000): An experimental comparison of three methods for constructing ensembles of decision trees: MBLR , boosting and randomisation. *Machine Learning, 40(2),* pp. 139-157.

Dietterich, T.G. and Bakiri, G. (1995): Solving multiclass learning problems via error correcting output codes. *Journal of Artificial Intelligence Research*, Vol 2, pp. 263–286.

Dietterich, T. G. and Kong, E. B. (1995): Machine Learning Bias, Statistical Bias, and Statistical Variance of decision Tree Algorithms. *Technical report, Department of Computer Science, Oregon State University*.

Drummond, C., Holte, R.C. (2003): C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Workshop on learning from imbalanced datasets II, international conference on machine learning.

Efron, B., and Tibshirani, R. (1993): *An introduction to the bootstrap*. New York: Chapman and Hall.

Georges, J. (2004): Data Preparation for Data Mining Using SAS® Software Course Notes. SAS Institute Inc., Cary, NC, USA.

Hand, D. J. (2000): Measuring diagnostic accuracy of statistical prediction rules. *Statistica Neerlandica, 53,* pp. 1–14.

Hand, D. and Till, R. (2001): A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning 45,* pp. 171–186.

Hanley, J.A. and McNeil, B.J. (1982): The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve, Radiology, 143, pp. 29-36

Hastie, T., Tibshirani, R. and Friedman, J. (2009): *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, New York, 464 pages.

Hsieh, C.H., Lu, R.H., Lee, N.H, Chiu, W.T., Hsu, M.H. and Li, Y.C. (2011): Novel solutions for an old disease: Diagnosis of acute appendicitis with MRFD , support vector machines, and artificial neural networks.*Surgery*;149(1): pp. 87-93.

Japkowicz, N. (2000): The class imbalance problem: Significance and strategies. In Proceedings of the 2000 international conference on artificial intelligence(IC-AI'2000): Special track on inductive learning, Las Vegas, Nevada.

Kamakura, W.A., Wedel, M., De Rossa, F., Mazzon, J.A. (2003): Cross-selling through database marketing: A mixed data factor analyzer for data augmentation and prediction. *International Journal of Research in Marketing 20 (1),*pp 45–65.

Kim, T.K. (2009): Boosting and Tree-structured Classifier. *ICCV09 Tutorial University of Cambridge.*

Kuncheva, L.I. (2005): Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters 26,* pp. 83–90.

Levin, N. and Zahavi, J. (1998): Continuous predictive modelling: A comparative analysis. *Journal of Interactive Marketing. 12(2),* pp 5 -22.

Li, S., Sun, B., and Wilcox, R. (2005): Cross-selling sequentially ordered products: An application to consumer banking services. *Journal of Marketing Research, 42(2),* pp 233–239.

Liaw, A., and Wiener, M. (2002): Classification and regression by MRFD , *R News*, 2, pp. 18–22.

Morrison, D. G. (1969): On the interpretation of discriminant analysis. *Journal of Marketing Research*, 6, pp. 156–163.

Murthy, S.K. (1998): Automatic construction of decision trees from data: a multi-disciplinary survey. Data Mining Knowledge Discovery 2(4), pp. 345–389

Neslin, S., Gupta, S., Kamakura, W., Lu, J., and Mason, C. (2006): Detection defection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2), pp. 204–211.

Niculesu-Mizil, A., and Caruana, R. (2006): An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning* .pp 161-168.

Prinzie, A. and Van den Poel, D. (2006): Investigating purchasing-sequence patterns for financial services using Markov, MTD and MTDg models. *European Journal of Operational Research*, 170(3), pp. 710–734.

Provost, F., Fawcett, T. and Kohavi, R. (1998): The Case Against Accuracy Estimation for Comparing Induction Algorithms. In: *Proc. Fifteenth Intl. Conf. Machine Learning*, Madison,WI, pp. 445-453.

Rao, C.R., Solka, J.L., Wegman, E.J. (2005): Handbook of Statistics, Vol. 24: Data Mining and Data Visualisation. Elsevier Books.

Schafer, J.L. (2001): Lecture Notes for Statistics 544: *Categorical Data Analysis I, Fall 2001. Penn State Univ.* http://www.stat.psu.edu/_jls/

Sejnowski, T.J. and Rosenberg, C. R. (1987): Parallel networks that learn to pronounce English text. *Journal of Complex Systems*, vol 1(1), pp. 145 -168.

Strobl, C., Boulesteix, A.L., Zeileis, A., and Hothorn, T. (2007): Bias in MRFD Variable Importance Measures: Illustrations, Sources and a Solution, *BMC Bioinformatics*, 8, 25.

Surowiecki, J. (2004): The Wisdom of Crowds: Why the Many are Smarter than the Few and how Collective Wisdom Shapes Business, Economics, Societies and Nations, Little, Brown.

Weiss, G.M. (2004): Mining with rarity: A unifying framework. *SIGKDD Explorations,* 6(1), pp. 7–19.

Weiss, G.M, and Provost, F. (2003): Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research,* 19, pp. 315–354.

Windeatt, T. and Ghaderi, R. (2003): Coding and decoding strategies for multi-class learning problems. *Information Fusion,* Vol 4, pp. 11–21.

Xie, Y., Li, X., Ngai, E., and Ying, W. (2009): Customer churn prediction using improved balanced MRFD s. *Expert Systems with Applications, 36(3, Part 1),* pp 5445–5449.

7 Appendix

7.1 Appendix A: Bank data set

Variable Name	Type	Level	Description
Customer_Num	ID	Nominal	Customer Identifier
target	Dependant	Binary	Product take up indicator
tenure	Independent	Interval	Relationship length
Product_S	Segment	Ordinal	Transactional Product type
ID_Regis_Num	ID	Nominal	Customer Identifier
ID10	ID	Nominal	Customer Identifier
Customer_Num_1	ID	Nominal	Customer Identifier
pers_entps_i	Class	Binary	Individual/Business customer indicator
new_cust_n	ID	Nominal	Customer Identifier
cust_segmt	Segment	Ordinal	Customer Segment as defined by Bank
Preferred_Lang	Input Variable	Nominal	Language
Ethnicity_Desc	Input Variable	Nominal	Race group
Marital_Status	Input Variable	Nominal	Marital status
Gender_Type_Desc	Input Variable	Nominal	Gender
Segment_Desc	Input Variable	Ordinal	Customer Financial Segment
salary	Input Variable	Interval	Income
home_p	Input Variable	Binary	Home phone
mobile_p	Input Variable	Binary	Mobile phone
bus_p	Input Variable	Binary	Business Phone
contacts	Input Variable	Ordinal	Number of contact channels
customer_age	Input Variable	Interval	Age
banker	Input Variable	Binary	Banker assigned
average_liabilities	Input Variable	Interval	Average value of customer liabilities over last 8 months
average_assets	Input Variable	Interval	Average value of customer assets over last 8 months
average_st_assets	Input Variable	Interval	Average value of short term assets over last 8 months
average_st_liab	Input Variable	Interval	Average value of short term liabilities over last 8 months
ave_month_end_bal	Input Variable	Interval	Average month end balance in transactional accounts over last 8 months
ave_cred_bal	Input Variable	Interval	Average credit balance in transactional accounts over last 8 months
ave_debit_bal	Input Variable	Interval	Average debit balance in transactional accounts over last 8 months
average_nii	Input Variable	Interval	Average net interest income over last 8 months
average_nir	Input	Interval	Average non interest revenue over last 8

	Variable		months
average_op_income	Input Variable	Interval	Average operating income (NIR +NIR) over last 8 months
ave_od	Input Variable	Interval	Average overdraft value over last 8 months
average_txnal_balance	Input Variable	Interval	Average transactional balance over last 8 months
lazy_balance	Input Variable	Interval	Average transactional account residual balance over last 8 months
min_balance	Input Variable	Interval	Lowest transactional account residual balance over last 8 months
max_balance	Input Variable	Interval	Highest transactional account residual balance over last 8 months
OD	Input Variable	Binary	Overdraft indicator
transactional_acc	Input Variable	Unary	Transactional account verification
N_PRODUCT	Dependent	Nominal	Customer product taken up
average_fees	Input Variable	Interval	Average fees paid over last 8 months
average_wdrw	Input Variable	Interval	Average value of withdrawals over last 8 months
average_digital	Input Variable	Interval	Average value of digital banking transactions over last 8 months
average_branch	Input Variable	Interval	Average value of branch banking transactions over last 8 months
average_atm	Input Variable	Interval	Average value of atm banking transactions over last 8 months
average_electronic	Input Variable	Interval	Average value of electronic transactions over last 8 months
average_pos	Input Variable	Interval	Average value of card swipe transactions over last 8 months
average_enq	Input Variable	Interval	Average volume of enquires over last 8 months
average_do	Input Variable	Interval	Average value of debit order transactions over last 8 months
digital	Input Variable	Binary	Digital banking indicator
tot_txn_7	Input Variable	Interval	Total transactions 7 months before reference point
tot_txn_6	Input Variable	Interval	Total transactions 6 months before reference point
tot_txn_5	Input Variable	Interval	Total transactions 5 months before reference point
tot_txn_4	Input Variable	Interval	Total transactions 4 months before reference point
tot_txn_3	Input Variable	Interval	Total transactions 3 months before reference point
tot_txn_2	Input Variable	Interval	Total transactions 2 months before reference point
tot_dep_7	Input Variable	Interval	Total deposits 7 months before reference point
tot_dep_6	Input Variable	Interval	Total deposits 6 months before reference point
tot_dep_5	Input Variable	Interval	Total deposits 5 months before reference point
tot_dep_4	Input	Interval	Total deposits 4 months before reference

	Variable		point
tot_dep_3	Input Variable	Interval	Total deposits 3 months before reference point
tot_dep_2	Input Variable	Interval	Total deposits 2 months before reference point
tot_spend_7	Input Variable	Interval	Total spend 7 months before reference point
tot_spend_6	Input Variable	Interval	Total spend 6 months before reference point
tot_spend_5	Input Variable	Interval	Total spend 5 months before reference point
tot_spend_4	Input Variable	Interval	Total spend 4 months before reference point
tot_spend_3	Input Variable	Interval	Total spend 3 months before reference point
tot_spend_2	Input Variable	Interval	Total spend 2 months before reference point
bal_diff_7	Input Variable	Interval	Net spend 7 months before reference point
bal_diff_6	Input Variable	Interval	Net spend 6 months before reference point
bal_diff_5	Input Variable	Interval	Net spend 5 months before reference point
bal_diff_4	Input Variable	Interval	Net spend 4 months before reference point
bal_diff_3	Input Variable	Interval	Net spend 3 months before reference point
bal_diff_2	Input Variable	Interval	Net spend 2 months before reference point
average_credit	Input Variable	Interval	Average value of credit transactions over last 8 months
total_credit	Input Variable	Interval	Total value of credit transactions over last 8 months
average_debit	Input Variable	Interval	Average value of debit transactions over last 8 months
total_debit	Input Variable	Interval	Total value of debit transactions over last 8 months
last_credit	Input Variable	Interval	Total value of credit transactions over last month
last_debit	Input Variable	Interval	Total value of debit transactions over last month
average_txns	Input Variable	Interval	Average volume of transactions over last 8 months
total_txns	Input Variable	Interval	Total volume of transactions over last 8 months
max_dep	Input Variable	Interval	Maximum deposit over last 8 months
significant_deposit	Input Variable	Interval	Significant deposit
increase_save_trend	Input Variable	Interval	Month on month change in money saved
min_bal_d	Input Variable	Interval	Minimum Net spend over last 8 months
min_e_bal	Input Variable	Interval	Minimum month end balance
g_tot_txn_	Input Variable	Interval	Rate of change of monthly total transactions over last 8 months

g_tot_dep_	Input Variable	Interval	Rate of change of monthly total deposits over last 8 months
g_tot_spend_	Input Variable	Interval	Rate of change of monthly total spend over last 8 months
g_bal_diff_	Input Variable	Interval	Rate of change of monthly net spend over last 8 months
g_txn_band	Input Variable	Interval	Rate of change of monthly total transactions over last 8 months (grouped)
g_spend_band	Input Variable	Interval	Rate of change of monthly total deposits over last 8 months (grouped)
g_dep_band	Input Variable	Interval	Rate of change of monthly total spend over last 8 months (grouped)
g_bal_diff_band	Input Variable	Interval	Rate of change of monthly net spend over last 8 months (grouped)
AGE_GROUP	Input Variable	Nominal	Age group
salary_group	Input Variable	Ordinal	Income group
Acc_Age_band	Input Variable	Nominal	Account tenure group
Target_I	Classification	Binary	Indicator for Investment take up
Target_S	Classification	Binary	Indicator for Secured Lending take up
Target_U	Classification	Binary	Indicator for Unsecured Lending take up
Target_C	Classification	Binary	Indicator for Credit card take up
Product	Classification	Nominal	Numerical indicator of product taken up
T1	Classification	Binary	Binary code for ECOC model
T2	Classification	Binary	Binary code for ECOC model
T3	Classification	Binary	Binary code for ECOC model
T4	Classification	Binary	Binary code for ECOC model
T5	Classification	Binary	Binary code for ECOC model
T6	Classification	Binary	Binary code for ECOC model
T7	Classification	Binary	Binary code for ECOC model
T8	Classification	Binary	Binary code for ECOC model
T9	Classification	Binary	Binary code for ECOC model
T10	Classification	Binary	Binary code for ECOC model
T11	Classification	Binary	Binary code for ECOC model
T12	Classification	Binary	Binary code for ECOC model
T13	Classification	Binary	Binary code for ECOC model
T14	Classification	Binary	Binary code for ECOC model
T15	Classification	Binary	Binary code for ECOC model
Code	Classification	Nominal	Code string for ECOC model

Table 25 : A list of all the variables created in the dataset

Variable	Mean	Standard Deviation	NonMissing Obs	Minimum	Median	Maximum	Skewness	Kurtosis
OD	0.1	0.3	20 540	-	-	1.0	2.6	5.0
Product	3.7	1.4	20 540	1.0	4.0	5.0	-0.6	-0.8
ave_cred_bal	28 794.7	212 620.8	20 540	-	1 589.0	11 441 200.0	26.3	1 001.9
ave_debit_bal	4 043.7	44 751.1	20 540	-	0.9	4 494 255.0	56.1	5 067.6
ave_month_end_bal	30 604.1	228 944.9	20 540	-4 421 847.0	1 820.2	12 444 887.0	24.9	985.8
ave_od	13.8	44.1	20 540	-	-	350.0	3.4	11.8
average_assets	28 794.7	212 620.8	20 540	-	1 589.0	11 441 200.0	26.3	1 001.9
average_atm	2.8	4.8	20 540	-	-	63.0	2.7	11.8
average_branch	0.4	1.0	20 540	-	-	41.2	11.6	329.1
average_credit	5 922.4	30 042.8	20 540	-	-	2 286 273.0	42.1	2 690.4
average_debit	5 705.4	28 169.4	20 540	-	-	2 283 459.0	43.0	2 878.7
average_digital	1.3	4.1	20 540	-	-	102.0	6.4	67.5
average_do	1 070.2	4 401.7	20 540	-	-	222 359.7	15.7	499.7
average_electronic	3.9	9.1	20 540	-	-	152.5	4.0	24.6
average_enq	0.4	1.8	20 540	-	-	58.7	8.5	123.1
average_fees	62.2	192.8	20 540	-	-	14 293.6	29.3	1 665.2
average_liabilities	4 043.7	44 751.1	20 540	-	0.9	4 494 255.0	56.1	5 067.6
average_nii	167.4	1 064.1	20 540	-161.7	10.0	58 654.8	24.0	871.8
average_nir	558.5	1 200.6	20 540	-56 812.8	315.8	75 385.0	15.2	1 200.6
average_op_income	725.9	1 710.8	20 540	-56 737.6	348.9	77 235.5	12.9	475.4
average_pos	538.9	1 444.1	20 540	-	-	39 725.1	6.0	73.3
average_st_assets	28 794.7	212 620.8	20 540	-	1 589.0	11 441 200.0	26.3	1 001.9
average_st_liab	4 043.7	44 751.1	20 540	-	0.9	4 494 255.0	56.1	5 067.6
average_txnal_balanc	28 794.7	212 620.8	20 540	-	1 589.0	11 441 200.0	26.3	1 001.9
average_txns	7.8	13.0	20 540	-	-	165.2	2.4	8.4
average_wdrw	2.4	3.9	20 540	-	-	50.0	2.3	7.5
customer_age	33.3	10.5	20 540	22.0	30.0	60.0	0.8	-0.4
g_bal_diff_	0.0	0.5	20 540	-3.0	-	3.1	0.7	7.6
g_tot_dep_	0.0	0.5	20 540	-3.8	-	3.3	0.9	12.1
g_tot_spend_	0.0	0.5	20 540	-3.4	-	3.4	1.1	14.4
g_tot_txn_	0.2	1.9	20 540	-21.6	-	30.9	1.4	22.0
increase_save_trend	0.8	0.4	20 540	-	1.0	1.0	-1.3	-0.3
last_credit	6 179.0	33 386.3	20 540	-	-	1 687 531.0	30.4	1 218.0
last_debit	5 947.1	31 176.8	20 540	-	-	1 750 000.0	32.9	1 476.8
lazy_balance	30 604.1	228 944.9	20 540	-4 421 847.0	1 820.2	12 444 887.0	24.9	985.8
max_balance	28 794.7	212 620.8	20 540	-	1 589.0	11 441 200.0	26.3	1 001.9
max_dep	35 534.4	180 257.0	20 540	-	-	13 717 635.0	42.1	2 690.4
min_bal_d	3 241.1	31 364.4	20 540	-	-	2 545 782.0	48.8	3 104.9
min_balance	4 043.7	44 751.1	20 540	-	0.9	4 494 255.0	56.1	5 067.6
min_e_bal	36 801.6	228 030.7	20 540	-	2 580.7	12 444 887.0	25.9	998.5
salary	5 125 919.0	697 920 000.0	20 540	-	3 886.0	100 000 000 000.0	143.2	20 519.5
significant_deposit	229 600 000 000 000.0	13 130 000 000 000 000.0	20 540	-54 000 000 000 000 000.0	-	54 040 000 000 000 000.0	0.2	7.8
tenure	39.5	61.6	20 540	5.0	11.0	454.0	3.0	9.4
tot_dep_2	6 179.0	33 386.3	20 540	-	-	1 687 531.0	30.4	1 218.0
tot_dep_3	6 380.7	93 661.4	20 540	-	-	12 524 883.0	118.5	15 593.3
tot_dep_4	6 445.1	53 980.7	20 540	-	-	4 288 151.0	53.5	3 618.0
tot_dep_5	5 571.5	38 148.3	20 540	-	-	3 043 254.0	53.7	3 858.6
tot_dep_6	5 577.0	31 286.2	20 540	-16 250.0	-	3 050 000.0	56.2	4 817.0
tot_dep_7	5 381.2	32 084.7	20 540	-	-	2 698 000.0	41.4	2 751.1
tot_spend_2	5 947.1	31 176.8	20 540	-	-	1 750 000.0	32.9	1 476.8
tot_spend_3	6 266.4	93 361.2	20 540	-	-	12 466 517.0	119.0	15 564.7
tot_spend_4	6 034.2	56 155.4	20 540	-	-	6 250 424.0	78.7	7 959.2
tot_spend_5	5 258.9	25 066.8	20 540	-	-	1 500 000.0	28.9	1 260.0
tot_spend_6	5 495.5	30 378.4	20 540	-	-	3 039 347.0	56.8	5 064.5
tot_spend_7	5 230.0	30 337.8	20 540	-	-	2 579 712.0	41.6	2 846.8
tot_txn_2	8.4	14.7	20 540	-	-	192.0	2.6	9.9
tot_txn_3	8.1	14.2	20 540	-	-	189.0	2.6	10.9
tot_txn_4	7.9	14.0	20 540	-	-	174.0	2.6	10.3
tot_txn_5	7.5	13.2	20 540	-	-	145.0	2.5	9.0
tot_txn_6	7.8	13.9	20 540	-	-	182.0	2.6	10.3
tot_txn_7	6.9	12.6	20 540	-	-	188.0	2.8	12.0
total_credit	35 534.4	180 257.0	20 540	-	-	13 717 635.0	42.1	2 690.4
total_debit	34 232.2	169 016.7	20 540	-	-	13 700 751.0	43.0	2 878.7
total_txns	46.7	77.9	20 540	-	-	991.0	2.4	8.4

Table 26 : Data description of the interval variables

7.2 Appendix B: Variable Selection

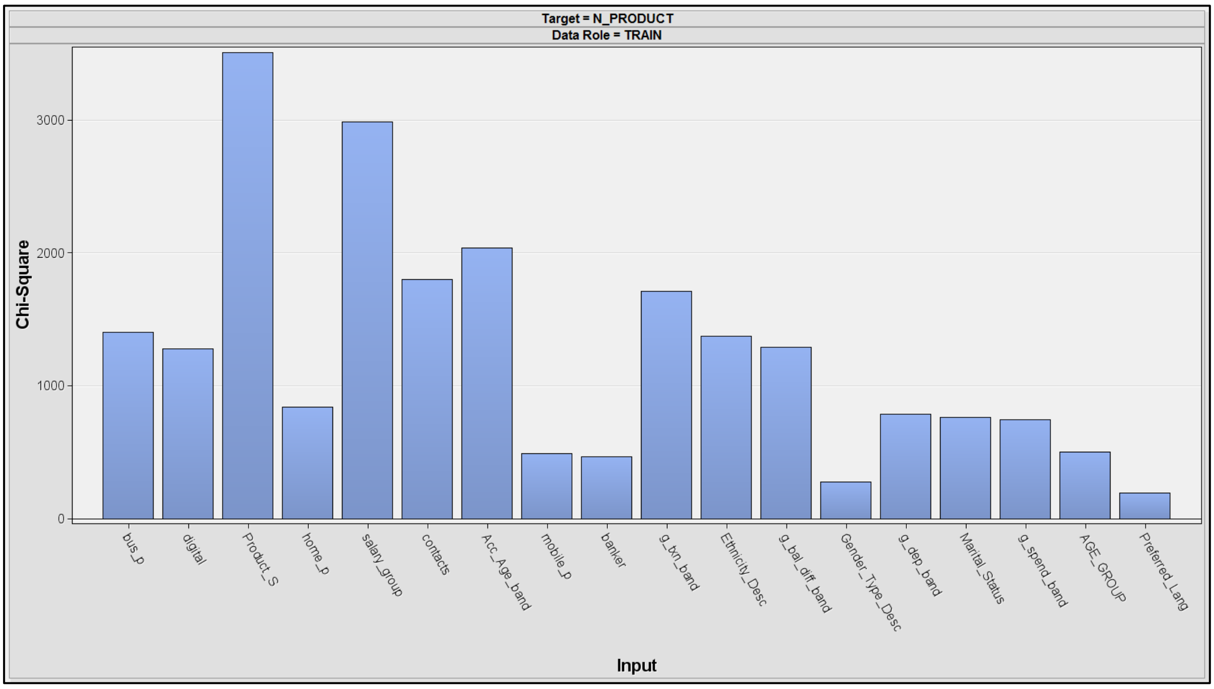

Figure 13 : Chi-Square Test for variable importance

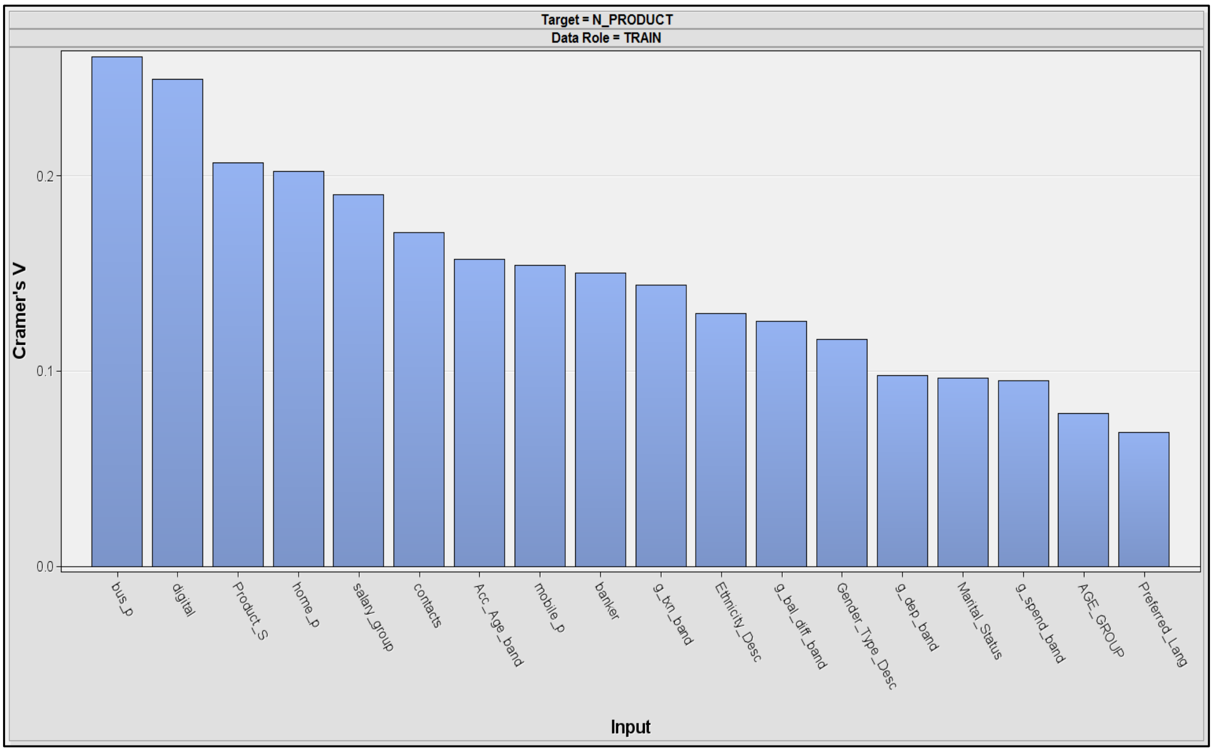

Figure 14 : Cramer's V test for variable importance

The Figures above depict the relative strength of each variable after the respective variable selection method was carried out. Only the top eighteen variables are shown.

7.3 Appendix C: Transition Matrix

Notations

T = Transactional Account

I = Investment Account

S = Secured Lending

U = Unsecured Lending or Credit Card

Assumptions

1. Transitions between states are to be assumed to occur in discrete time
2. In case of ties, where a customer has acquired more than one product, the product which was bought first within the selected time point will be considered.
3. An individual cannot move to any other state without product (**T**), a transactional account. Therefore, product (T) is regarded as the entry product into a banking relationship.

The matrices in Figure 15 below illustrate the transition probabilities of the customers as they move from one state to the next. At any given state, a customer has 2 options, namely; either to move forward to the next state or remain in the same state. The transition matrices are then constructed.

If no additional information is known about a subgroup of customers, one would expect then to move to the different states with probabilities illustrated by the different matrices.

	T	TI	TS	TU	TIS	TIU	TSU	TISU
T	0.90	0.02	0.01	0.06	0.00	0.01	0.00	0.00
TI	0	0.91	0	0	0.01	0.07	0	0.00
TS	0	0	0.91	0	0.02	0	0.07	0.01
TU	0	0	0	0.96	0	0.03	0.02	0.00
TIS	0	0	0	0	0.93	0	0	0.07
TIU	0	0	0	0	0	0.98	0	0.02
TSU	0	0	0	0	0	0	0.97	0.03
TISU	0	0	0	0	0	0	0	1.00

Figure 15 : Transition Matrices after 6 months

7.4 Appendix D : Program Codes

7.4.1 Data Creation

```
/*Initial data extraction for using the reference time points
*/

%macro assign_dates;

     %global date mnth  end_date ;

     data _null_;
       call symput('date',intnx('month',&month.d,-&i));
       call         symput('mnth',put(intnx('month',&month.d,-
&i),MONYY7.));
       call     symput('end_date',put(intnx('month',&month.d,-
&i,'end'),DATE9.));
       call    sympt('end_datea1',put(intnx('month',&month.d,-
%eval(&i+0),'beginning'),yymmn6.));
       call    sympt('end_datea2',put(intnx('month',&month.d,-
%eval(&i+4),'beginning'),yymmn6.));
       call    sympt('end_datea3',put(intnx('month',&month.d,-
%eval(&i+8),'beginning'),yymmn6.));
       call    sympt('end_datea4',put(intnx('month',&month.d,-
%eval(&i+12),'beginning'),yymmn6.));
       call        symput('start_date',intnx('month',&month.d,-
%eval(&i+0),'beginning'));
       call       symput('start_date1',intnx('month',&month.d,-
%eval(&i+4),'beginning'));
       call       symput('start_date2',intnx('month',&month.d,-
%eval(&i+8),'beginning'));
       call       symput('start_date3',intnx('month',&month.d,-
%eval(&i+12),'beginning'));
       call         symput('end_date1',intnx('month',&month.d,-
%eval(&i-3),'end'));
       call         symput('end_date2',intnx('month',&month.d,-
%eval(&i+1),'end'));
       call         symput('end_date3',intnx('month',&month.d,-
%eval(&i+5),'end'));
       call         symput('end_date4',intnx('month',&month.d,-
%eval(&i+9),'end'));
     run;

     data _null_;
       call symput('date',intnx('month',&month.d,-&i));
       call     symput('end_date',put(intnx('month',&month.d,-
&i,'end'),DATE9.));
```

```
       call    symput('end_datea1',put(intnx('month',&month.d,-
%eval(&i+0),'beginning'),yymmn6.));
       call    symput('end_datea2',put(intnx('month',&month.d,-
%eval(&i+4),'beginning'),yymmn6.));
       call    symput('end_datea3',put(intnx('month',&month.d,-
%eval(&i+8),'beginning'),yymmn6.));
       call    symput('end_datea4',put(intnx('month',&month.d,-
%eval(&i+12),'beginning'),yymmn6.));
        call  symput('tran_datea1',put(intnx('month',&month.d,-
%eval(&i+7),'beginning'),yymmn6.));
       call   symput('tran_datea2',put(intnx('month',&month.d,-
%eval(&i+11),'beginning'),yymmn6.));
       call   symput('tran_datea3',put(intnx('month',&month.d,-
%eval(&i+15),'beginning'),yymmn6.));
       call   symput('tran_datea4',put(intnx('month',&month.d,-
%eval(&i+19),'beginning'),yymmn6.));
         call
symput('tran2_datea1',put(intnx('month',&month.d,-
%eval(&i+1),'beginning'),yymmn6.));
       call  symput('tran2_datea2',put(intnx('month',&month.d,-
%eval(&i+5),'beginning'),yymmn6.));
       call  symput('tran2_datea3',put(intnx('month',&month.d,-
%eval(&i+9),'beginning'),yymmn6.));
       call  symput('tran2_datea4',put(intnx('month',&month.d,-
%eval(&i+13),'beginning'),yymmn6.));
       call        symput('tran_date1',intnx('month',&month.d,-
%eval(&i+7),'beginning'));
       call        symput('tran_date2',intnx('month',&month.d,-
%eval(&i+11),'beginning'));
       call        symput('tran_date3',intnx('month',&month.d,-
%eval(&i+15),'beginning'));
       call        symput('tran_date4',intnx('month',&month.d,-
%eval(&i+19),'beginning'));
        call      symput('tran2_date1',intnx('month',&month.d,-
%eval(&i+1),'beginning'));
       call       symput('tran2_date2',intnx('month',&month.d,-
%eval(&i+5),'beginning'));
       call       symput('tran2_date3',intnx('month',&month.d,-
%eval(&i+9),'beginning'));
       call       symput('tran2_date4',intnx('month',&month.d,-
%eval(&i+13),'beginning'));
     run;

%put &end_date &mnth ;
%mend;
%macro global_base;

data global_base1 ;
set bi_account1 ;
```

```
format status_date1 date9. status_date2 date9. status_date3
date9. status_date4 date9. ;

if Account_Open_Dt le &start_date and Account_Open_Dt ne . and
( Account_Close_Dt eq . or
   Account_Close_Dt gt &end_date1)and trans=1 then
status_date1=&start_date;
if Account_Open_Dt lt &start_date1 and Account_Open_Dt ne .
and ( Account_Close_Dt eq . or
   Account_Close_Dt gt &end_date2) and trans=1 then
status_date2=&start_date1;
if Account_Open_Dt le &start_date2 and Account_Open_Dt ne .
and ( Account_Close_Dt eq . or
   Account_Close_Dt gt &end_date3) and trans=1 then
status_date3=&start_date2;
if Account_Open_Dt le &start_date3 and Account_Open_Dt ne .
and ( Account_Close_Dt eq . or
   Account_Close_Dt gt &end_date4) and trans=1 then
status_date4=&start_date3;

if &start_date =< Account_Open_Dt =< &end_date1 and trans=0
then new_acc_1 = 1;
    else new_acc_1 = 0;
if &start_date1 =< Account_Open_Dt =< &end_date2 and trans=0
then new_acc_2 = 1;
    else new_acc_2 = 0;
if &start_date2 =< Account_Open_Dt =< &end_date3 and trans=0
then new_acc_3 = 1;
    else new_acc_3 = 0;
if &start_date3 =< Account_Open_Dt =< &end_date4 and trans=0
then new_acc_4 = 1;
    else new_acc_4 = 0;

if Account_Open_Dt le &start_date and trans=0 and (
Account_Close_Dt eq . or
   Account_Close_Dt gt &start_date ) then prod_1=1 ;
        else prod_1=0 ;
if Account_Open_Dt le &start_date1 and trans=0 and (
Account_Close_Dt eq . or
   Account_Close_Dt gt &start_date1 ) then prod_2=1 ;
        else prod_2=0 ;
if Account_Open_Dt le &start_date2 and trans=0 and (
Account_Close_Dt eq . or
   Account_Close_Dt gt &start_date2 ) then prod_3=1 ;
        else prod_3=0 ;
if Account_Open_Dt le &start_date3 and trans=0 and (
Account_Close_Dt eq . or
   Account_Close_Dt gt &start_date3 ) then prod_4=1 ;
        else prod_4=0 ;
```

```
 if prod_1=1 or new_acc_1 = 1 then status_date1=&start_date;
  if prod_2=1 or new_acc_2 = 1 then status_date2=&start_date1;
   if      prod_3=1      or      new_acc_3      =      1      then
status_date3=&start_date2;
 if prod_4=1 or new_acc_4 = 1 then status_date4=&start_date3;

   if status_date1 ne . then tg_1=1;
     else tg_1=0;
   if status_date2 ne . then tg_2=1;
     else tg_2=0;
   if status_date3 ne . then tg_3=1;
     else tg_3=0;
   if status_date4 ne . then tg_4=1;
     else tg_4=0;

run;

%mend;
%macro sample_base;


data base;
set global_base1;
format information_date date9. ;
if tg_&i = 1 ;
information_date=status_date&i ;
drop status_date1 status_date2 status_date3 status_date4 ;
 account_age=intck('month',Account_Open_Dt,information_date);
run;

proc summary data=base sum nway missing;
   class customer_num ;
    var trans prod_&i new_acc_&i;
 output out=prod_&i (drop=_type_ _freq_) sum()= ;
run;

data base_&i;
set prod_&i;
if trans >= 1 and prod_&i =0 and new_acc_&i > 0 then target=1;
 else  if  trans  >=  1  and  prod_&i  =  0  and  new_acc_&i  =  0  then
target=0;
else target=99;
if target=99 then delete;
run;


data base1_&i;
set base_&i;
```

```
run;


proc sql;
      create table base1_&i
      as select a.*,
                 b.account_num,
                    b.trans as base_acc_ind,
                      b.account_age as tenure,
                    b.Product_S,
                    b.pd_cat_id_aam,
                    b.pd_grp_id_aam

      from base1_&i as a left join
            base as b

      on a.customer_num = b.customer_num ;
quit;


     proc sort data=base1_&i;
          by account_num;
     run;

     proc sql;
          create table accounts_&i as
          select distinct
                    account_num
          from base1_&i
       where base_acc_ind=1;
     quit;

     proc sort data=accounts_&i;
          by account_num;
     run;

     proc sql;
          create table customers_&i as
          select distinct
                          customer_num
          from base1_&i;
     quit;

     proc sort data=customers_&i;
          by customer_num ;
     run;
```

```
%mend;

%macro demographics;

proc sql;
      create table jon.customers_&i
      as select a.*,
                 b.ID_Regis_Num,
                    b.ID_Type

      from jon.customers_&i as a left join
            *****  as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;

DATA jon.customers_&i;
  SET jon.customers_&i;
%VALIDATEIDNO(ID_Regis_Num);
IF NOT VALID THEN DELETE;
DROP VALID;
RUN;

data jon.customers_&i;
set jon.customers_&i;
 ID10 = SUBSTR(LEFT(ID_Regis_Num),1,10);
run;

proc sql;
      create table jon.age_&i
      as select a.*,
                 b.Birth_Date,
                    b.Age

      from jon.customers_&i as a left join
            ******** as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;


proc sql;
      create table segment_lookup
      as select *
      from ********* ;
quit;


proc sql;
      create table jon.lang_&i
      as select a.Customer_Num,
                a.ID10,
                 b.Preferred_Lang

      from jon.customers_&i as a left join
            ********   as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;


proc sql;
      create table jon.race_&i
      as select a.Customer_Num,
                a.ID10,
                 b.Ethnicity_Cd,
                    b.Ethnicity_Desc

      from jon.customers_&i as a left join
            *********  as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;

proc sql;
      create table jon.marital_status_&i
      as select a.Customer_Num,
                a.ID10,
                 b.Marital_Status

      from jon.customers_&i as a left join
            *********  as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;

proc sql;
      create table jon.GENDER_&i
      as select a.Customer_Num,
                a.ID10,
                 b.Gender_Type_Cd,
                b.Gender_Type_Desc


      from jon.customers_&i as a left join
            *********  as b
      on compress(a.Customer_Num) = compress(b.Customer_Num);
quit;

data jon.customers_&i;
  set jon.customers_&i;
 Customer_Num_1= input(Customer_Num,10.);
run;
```

```
proc sql ;
      create table jon.income_&i as
      select customer_num,
              customer_name,
              derived_income_amt,
                segment_cd,
                estimated_income_amt

      from ********
      where   Customer_Num   in   (select   Customer_Num_1   from
jon.customers_&i)
       and Current_Month eq &&end_datea&i..
;
quit;


data jon.income_&i ;
set jon.income_&i;
salary=max(derived_income_amt,estimated_income_amt);
run;

proc sort data=jon.income_&i ;
 by customer_num descending salary ;
run ;

proc sort data=jon.income_&i nodupkey out=jon.income1_&i ;
 by customer_num ;
run ;


proc sql;
      create table jon.contact_&i
      as select      a.*,
                      b.pers_entps_i,
                     b.cust_segmt_n,
                          b.home_phone,
                          b.cell_phone,
                          b.busns_phone,
                          b.mgmt_rep_n,
                          b.MARTL_STTUS_X,
                          b.OCPTN_CAT_C


      from jon.customers_&i as  a left join
            *********  b
      on compress(a.id10) = compress(b.id10);
quit;
```

```
proc sql;
      create table jon.segment_&i
      as select a.*,
                 b.Segment_Desc,
                    b.Main_Segment_Desc,
                    b.Financial_Segment_Desc

      from jon.contact_&i as a left join
            segment_lookup as b
      on a.cust_segmt_n = b.segment_cd;
quit;


%mend;
%macro transactional;

proc sql ;
      create table jon.transactional_&i as
      select *
      from ******
      where    account_num    in    (select    account_num    from
jon.accounts_&i)
      and Processing_Dt >= &&tran_date&i..
      and Processing_Dt <= &&tran2_date&i..
;
quit;


proc sql ;
      create table jon.acc_value_&i as
      select *
      from **********
      where    account_num    in    (select    account_num    from
jon.accounts_&i)
      and profit_cycle_ccyymm >= &&tran_datea&i..
      and profit_cycle_ccyymm <= &&tran2_datea&i..
;
quit;


proc sql;
create table jon.transactional_&i as

select a.*,
       b.*

from jon.transactional_&i as a left join
    TRANSACTION_TYPE as b
```

```
on a.Transaction_Type_Cd = b.Trans_Type ;
quit;



proc sql;
create table jon.transactional_&i as

select a.*,
       b.txn_catg,
        b.Channel,
        b.type

from jon.transactional_&i as a left join
    Bi_posted_trans_lookup_txn_cd as b
on a.Transaction_Type_Cd = b.Transaction_Type_Cd ;
quit;

data jon.transactional_&i;
set jon.transactional_&i;
if type ne '' then post=1;
 else post = 0 ;
run;


data jon.transactional_&i;
set jon.transactional_&i;
TXN_COUNT=1;
if  post=1 then output;
run;


%mend;
%macro sampling;
%mend;
%macro main;

     %global month o_loop i_loop i;
     %let month = '01OCT2012';
     %let o_loop = 5;
     %let i_loop = 1;

%do i = 1 %to 5;
          %assign_dates;

          %global_base;

 %end;
```

```
%do i = 1 %to &o_loop;

          %assign_dates;

            %sample_base;

            %demographics;

            %transactional;

%end;

%mend;
/***********************************************    OPEN    CODE
***********************************************/
%main;


/*code to create derived transactional variables */

%macro assign_dates;

     %global date mnth  end_date ;

     data _null_;
       call symput('date',intnx('month',&month.d,-&i));
       call          symput('mnth',put(intnx('month',&month.d,-
&i),MONYY7.));
       call        symput('start_date',intnx('month',&month.d,-
%eval(&i+1),'beginning'));
       call          symput('end_date',intnx('month',&month.d,-
%eval(&i+1),'end'));
     run;

%put &end_date &mnth ;
%mend;
%macro accs_base;



     proc sql;
          create table acc_trans2_&i as
          select customer_num,
                          sum(wdrw_count) as WDRW_COUNT_&i,
                          sum(dep_count) as DEP_COUNT_&i,
                          sum(pay_count) as ENQ_COUNT_&i,
                          sum(pur_count) as POS_COUNT_&i,
                          sum(di_count) as DI_COUNT_&i,
                          sum(trfi_count) as TI_COUNT_&i,
                          sum(trfo_count) as TO_COUNT_&i,
                          sum(do_count) as DO_COUNT_&i,
                          sum(wdrw_amt) as WDRW_AMT_&i,
                          sum(dep_amt) as DEP_AMT_&i,
                          sum(pay_amt) as PAY_AMT_&i,
                          sum(pur_amt) as POS_AMT_&i,
                          sum(fees_amt) as FEES_AMT_&i,
                          sum(di_amt) as DI_AMT_&i,
                          sum(do_amt) as DO_AMT_&i,
                          sum(trfi_amt) as TI_AMT_&i,
                          sum(trfo_amt) as TO_AMT_&i,
                          sum(DIGITAL_COUNT)                 as
DIGITAL_COUNT_&i,
                          sum(BRANCH_COUNT)                  as
BRANCH_COUNT_&i,
                          sum(ATM_COUNT) as ATM_COUNT_&i,
                          sum(ELECTRONIC_COUNT)              as
ELEC_COUNT_&i
          from acc_trans_&i
          where Processing_Dt >= &start_date
         and Processing_Dt <= &end_date
          group by customer_num ;
     quit;

    %if &i = 2 %then %do;
          data jon.account_txns;
                         set  acc_trans2_&i(in=b);
          run;

          proc sort data=jon.account_txns;
             by customer_num;
         run;
       %end;

     %else %do;
          data jon.account_txns;
               merge jon.account_txns(in=a)
                              acc_trans2_&i(in=b);
               by customer_num;
               if a or b ;
          run;
     %end;

%mend;
%macro main;

     %global month o_loop i_loop i;
     %let month = '01OCT2012';
     %let o_loop = 7;
     %let i_loop = 6;

%do i = 2 %to &o_loop;
          %assign_dates;

/*            %accs_base;*/

     %end;

%mend;
/***********************************************   OPEN   CODE
***********************************************/
%main;
endrsubmit;




/*code to calculate rate of change of balances */


%MACRO TRANSFORM (ind,VARIABLE,ln,y);


data T_&VARIABLE.;
    set test;
z=&y.;
log = &ln.;

if  log = 1 then do ;

   if z=0 then do;

    ln_&VARIABLE.2=log(abs(&VARIABLE.2+1));
    ln_&VARIABLE.3=log(abs(&VARIABLE.3+1));
    ln_&VARIABLE.4=log(abs(&VARIABLE.4+1));
     ln_&VARIABLE.5=log(abs(&VARIABLE.5+1));
     ln_&VARIABLE.6=log(abs(&VARIABLE.6+1));



    x2=5; x3=4; x4=3; x5=2; x6=1;
    sum_xi=sum(x2,x3,x4,x5,x6);

sum_yi=sum(ln_&VARIABLE.2,ln_&VARIABLE.3,ln_&VARIABLE.4,ln_&VA
RIABLE.5,ln_&VARIABLE.6);

sum_xiyi=x2*ln_&VARIABLE.2+x3*ln_&VARIABLE.3+x4*ln_&VARIABLE.4
+x5*ln_&VARIABLE.5+x6*ln_&VARIABLE.6;
```

```
    sum_xi2=(x2**2)+(x3**2)+(x4**2)+(x5**2)+(x6**2);
    n=5;
    g_&VARIABLE.=(n*sum_xiyi    -    sum_xi*sum_yi)/(n*sum_xi2-
sum_xi**2);
    keep customer_num g_&VARIABLE.;
  end;

   else if z=1 then do;

    ln_&VARIABLE.2=log(abs(&VARIABLE.2+1+&ind.));
    ln_&VARIABLE.3=log(abs(&VARIABLE.3+1+&ind.));
    ln_&VARIABLE.4=log(abs(&VARIABLE.4+1+&ind.));
     ln_&VARIABLE.5=log(abs(&VARIABLE.5+1+&ind.));
     ln_&VARIABLE.6=log(abs(&VARIABLE.6+1+&ind.));



    x2=5; x3=4; x4=3; x5=2; x6=1;
    sum_xi=sum(x2,x3,x4,x5,x6);

sum_yi=sum(ln_&VARIABLE.2,ln_&VARIABLE.3,ln_&VARIABLE.4,ln_&VA
RIABLE.5,ln_&VARIABLE.6);

sum_xiyi=x2*ln_&VARIABLE.2+x3*ln_&VARIABLE.3+x4*ln_&VARIABLE.4
+x5*ln_&VARIABLE.5+x6*ln_&VARIABLE.6;
    sum_xi2=(x2**2)+(x3**2)+(x4**2)+(x5**2)+(x6**2);
    n=5;
    g_&VARIABLE.=(n*sum_xiyi    -    sum_xi*sum_yi)/(n*sum_xi2-
sum_xi**2);
    keep customer_num g_&VARIABLE.;
  end;

end;

else if  log = 0 then do;

  x2=5; x3=4; x4=3; x5=2; x6=1;
    sum_xi=sum(x2,x3,x4,x5,x6);

sum_yi=sum(&VARIABLE.2,&VARIABLE.3,&VARIABLE.4,&VARIABLE.5,&VA
RIABLE.6);

sum_xiyi=x2*&VARIABLE.2+x3*&VARIABLE.3+x4*&VARIABLE.4+x5*&VARI
ABLE.5+x6*&VARIABLE.6;
    sum_xi2=(x2**2)+(x3**2)+(x4**2)+(x5**2)+(x6**2);
    n=5;
    g_&VARIABLE.=(n*sum_xiyi    -    sum_xi*sum_yi)/(n*sum_xi2-
sum_xi**2);
 keep customer_num g_&VARIABLE. ;
```

```
end;
    run;
proc sort data=T_&VARIABLE.;
  by customer_num;
run;


%MEND;


%MACRO loop;
  **************  note  that  august2012  is  first  date  data
extracted therefore adjust ind;
    %TRANSFORM                                       (ind=0,VARIABLE
=average_credit_balance_,ln=1,y=1);
     %TRANSFORM                (ind=0,VARIABLE                 =
average_debit_balance_,ln=1,y=1);
     %TRANSFORM                         (ind=min_e_bal,VARIABLE
=month_end_balance_,ln=0,y=0);
     %TRANSFORM (ind=0,VARIABLE =tot_txn_,ln=0,y=1);
     %TRANSFORM (ind=0,VARIABLE =tot_dep_,ln=1,y=1);
     %TRANSFORM (ind=0,VARIABLE =tot_spend_,ln=1,y=1);
     %TRANSFORM (ind=min_bal_d,VARIABLE = bal_diff_,ln=1,y=0);

%MEND;
%loop;
proc sort data=test;
  by customer_num;
run;

data model;
  merge test(in=a)

     T_tot_txn_  (in=e)
     T_tot_dep_  (in=f)
     T_tot_spend_  (in=g)
     T_bal_diff_  (in=g)      ;

   by customer_num;
if a;
run;

data model_1  ;
  set model  ;
   if     g_tot_txn_  =<  -1  then  g_txn_band='a.STEEP  DECLINE
';
  else if  -1 < g_tot_txn_ =< -0.1 then g_txn_band='b.DECLINE
';
```

```
  else if  -0.1 < g_tot_txn_ =< 0.1 then g_txn_band='c.DORMANT
';
  else if 0.1 < g_tot_txn_ =< 1 then g_txn_band='d.SLOW
INCREASE        ';
  else if 1 < g_tot_txn_   then g_txn_band='e.STEEP INCREASE
';

     if     g_tot_spend_ =< -1 then g_spend_band='a.STEEP
DECLINE           ';
  else if     -1 < g_tot_spend_ =< -0.1 then
g_spend_band='b.DECLINE                ';
  else if     -0.1 < g_tot_spend_ =< 0.1 then
g_spend_band='c.DORMANT                ';
  else if 0.1 < g_tot_spend_ =< 1 then g_spend_band='d.SLOW
INCREASE        ';
  else if 1 < g_tot_spend_   then g_spend_band='e.STEEP
INCREASE        ';

     if    g_tot_dep_ =< -1 then g_dep_band='a.STEEP DECLINE
';
  else if  -1 < g_tot_dep_ =< -0.1 then g_dep_band='b.DECLINE
';
  else if  -0.1 < g_tot_dep_ =< 0.1 then g_dep_band='c.DORMANT
';
  else if 0.1 < g_tot_dep_ =< 1 then g_dep_band='d.SLOW
INCREASE        ';
else if 1 < g_tot_dep_ then g_dep_band='e.STEEP INCREASE
';

     if     g_bal_diff_ =< -1 then g_bal_diff_band='a.STEEP
DECLINE           ';
  else if     -1 < g_bal_diff_ =< -0.1 then
g_bal_diff_band='b.DECLINE                ';
  else if     -0.1 < g_bal_diff_ =< 0.1 then
g_bal_diff_band='c.DORMANT                ';
  else if 0.1 < g_bal_diff_ =< 1 then g_bal_diff_band='d.SLOW
INCREASE        ';
  else if 1 < g_bal_diff_   then g_bal_diff_band='e.STEEP
INCREASE        ';
run;
```

7.4.2 MLR model

```
*------------------------------------------------------------
*;
* EM SCORE CODE;
* VERSION: 6.12;
* GENERATED BY: Insight;
* CREATED: 01DEC2013:16:22:44;
*------------------------------------------------------------
*;
*------------------------------------------------------------
*;
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*------------------------------------------------------------
*;
*------------------------------------------------------------
*;
* TOOL: Partition Class;
* TYPE: SAMPLE;
* NODE: Part;
*------------------------------------------------------------
*;
*------------------------------------------------------------
*;
* TOOL: Regression;
* TYPE: MODEL;
* NODE: Reg;
*------------------------------------------------------------
*;
*************************************;
*** begin scoring code for regression;
*************************************;

length _WARN_ $4;
label _WARN_ = 'Warnings' ;

length I_N_PRODUCT $ 8;
label I_N_PRODUCT = 'Into: N_PRODUCT' ;
*** Target Values;
array  REGDRF  [5]  $8  _temporary_  ('E.CARD'  'D.UL'  'C.SL'
'B.INV' 'NO_TAKE' );
label U_N_PRODUCT = 'Unnormalized Into: N_PRODUCT' ;
length U_N_PRODUCT $ 8;
*** Unnormalized target values;
array REGDRU[5] $ 8 _temporary_ ('e.CARD  '  'd.UL     '  'c.SL
'
'b.INV   '  'NO_TAKE ' );

drop _DM_BAD;
_DM_BAD=0;

*** Check OD for missing values ;
if missing( OD ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check ave_month_end_bal for missing values ;
if missing( ave_month_end_bal ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check average_pos for missing values ;
if missing( average_pos ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check average_wdrw for missing values ;
if missing( average_wdrw ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check bal_diff_4 for missing values ;
if missing( bal_diff_4 ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check g_tot_spend_ for missing values ;
if missing( g_tot_spend_ ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Check tot_spend_4 for missing values ;
if missing( tot_spend_4 ) then do;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Generate dummy variables for AGE_GROUP ;
drop _1_0 _1_1 _1_2 _1_3 ;
```

```
*** encoding is sparse, initialize to zero;
_1_0 = 0;
_1_1 = 0;
_1_2 = 0;
_1_3 = 0;
if missing( AGE_GROUP ) then do;
   _1_0 = .;
   _1_1 = .;
   _1_2 = .;
   _1_3 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm20 $ 20; drop _dm20 ;
   %DMNORMCP( AGE_GROUP , _dm20 )
   if _dm20 = '[21 - 26)'  then do;
      _1_0 = 1;
   end;
   else if _dm20 = '[26 - 36)'  then do;
      _1_1 = 1;
   end;
   else if _dm20 = '[36 - 46)'  then do;
      _1_2 = 1;
   end;
   else if _dm20 = '[46 - 55)'  then do;
      _1_3 = 1;
   end;
   else if _dm20 = '[55 - 66)'  then do;
      _1_0 = -1;
      _1_1 = -1;
      _1_2 = -1;
      _1_3 = -1;
   end;
   else do;
      _1_0 = .;
      _1_1 = .;
      _1_2 = .;
      _1_3 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for Ethnicity_Desc ;
drop _3_0 _3_1 _3_2 _3_3 _3_4 ;
*** encoding is sparse, initialize to zero;
_3_0 = 0;
_3_1 = 0;
```

```
_3_2 = 0;
_3_3 = 0;
_3_4 = 0;
if missing( Ethnicity_Desc ) then do;
   _3_0 = .;
   _3_1 = .;
   _3_2 = .;
   _3_3 = .;
   _3_4 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm32 $ 32; drop _dm32 ;
   length _dm250 $ 250; drop _dm250;
   _dm250 = put( Ethnicity_Desc , $250. );
   %DMNORMCP( _dm250, _dm32 )
   if _dm32 = 'AFRICAN'  then do;
      _3_0 = 1;
   end;
   else if _dm32 = 'WHITE'  then do;
      _3_0 = -1;
      _3_1 = -1;
      _3_2 = -1;
      _3_3 = -1;
      _3_4 = -1;
   end;
   else if _dm32 = 'COLOURED'  then do;
      _3_2 = 1;
   end;
   else if _dm32 = 'ASIAN'  then do;
      _3_1 = 1;
   end;
   else if _dm32 = 'UNKNOWN'  then do;
      _3_4 = 1;
   end;
   else if _dm32 = 'NOT APPLICABLE'  then do;
      _3_3 = 1;
   end;
   else do;
      _3_0 = .;
      _3_1 = .;
      _3_2 = .;
      _3_3 = .;
      _3_4 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;
```

```
*** Generate dummy variables for Gender_Type_Desc ;
drop _4_0 ;
if missing( Gender_Type_Desc ) then do;
   _4_0 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm32 $ 32; drop _dm32 ;
   length _dm50 $ 50; drop _dm50;
   _dm50 = put( Gender_Type_Desc , $50. );
   %DMNORMCP( _dm50, _dm32 )
   if _dm32 = 'MALE'  then do;
      _4_0 = -1;
   end;
   else if _dm32 = 'FEMALE'  then do;
      _4_0 = 1;
   end;
   else do;
      _4_0 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for Marital_Status ;
drop _5_0 _5_1 _5_2 _5_3 _5_4 _5_5 ;
*** encoding is sparse, initialize to zero;
_5_0 = 0;
_5_1 = 0;
_5_2 = 0;
_5_3 = 0;
_5_4 = 0;
_5_5 = 0;
if missing( Marital_Status ) then do;
   _5_0 = .;
   _5_1 = .;
   _5_2 = .;
   _5_3 = .;
   _5_4 = .;
   _5_5 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm32 $ 32; drop _dm32 ;
   length _dm50 $ 50; drop _dm50;
   _dm50 = put( Marital_Status , $50. );
   %DMNORMCP( _dm50, _dm32 )
   if _dm32 = 'SINGLE'  then do;
      _5_4 = 1;
   end;
   else if _dm32 = 'MARRIED'  then do;
      _5_1 = 1;
   end;
   else if _dm32 = 'DIVORCED'  then do;
      _5_0 = 1;
   end;
   else if _dm32 = 'UNCLASSIFIED'  then do;
      _5_5 = 1;
   end;
   else if _dm32 = 'WIDOWED'  then do;
      _5_0 = -1;
      _5_1 = -1;
      _5_2 = -1;
      _5_3 = -1;
      _5_4 = -1;
      _5_5 = -1;
   end;
   else if _dm32 = 'MISSING'  then do;
      _5_2 = 1;
   end;
   else if _dm32 = 'SEPARATED'  then do;
      _5_3 = 1;
   end;
   else do;
      _5_0 = .;
      _5_1 = .;
      _5_2 = .;
      _5_3 = .;
      _5_4 = .;
      _5_5 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for Preferred_Lang ;
drop _6_0 _6_1 ;
if missing( Preferred_Lang ) then do;
   _6_0 = .;
   _6_1 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm32 $ 32; drop _dm32 ;
   length _dm50 $ 50; drop _dm50;
   _dm50 = put( Preferred_Lang , $50. );
   %DMNORMCP( _dm50, _dm32 )
   if _dm32 = 'E'  then do;
      _6_0 = 0;
      _6_1 = 1;
   end;
   else if _dm32 = 'A'  then do;
      _6_0 = 1;
      _6_1 = 0;
   end;
   else if _dm32 = 'Z'  then do;
      _6_0 = -1;
      _6_1 = -1;
   end;
   else do;
      _6_0 = .;
      _6_1 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for Product_S ;
drop _7_0 _7_1 _7_2 _7_3 _7_4 _7_5 _7_6 _7_7 _7_8 _7_9 _7_10
_7_11 ;
*** encoding is sparse, initialize to zero;
_7_0 = 0;
_7_1 = 0;
_7_2 = 0;
_7_3 = 0;
_7_4 = 0;
_7_5 = 0;
_7_6 = 0;
_7_7 = 0;
_7_8 = 0;
_7_9 = 0;
_7_10 = 0;
_7_11 = 0;
if missing( Product_S ) then do;
   _7_0 = .;
   _7_1 = .;
   _7_2 = .;
   _7_3 = .;
   _7_4 = .;
   _7_5 = .;
   _7_6 = .;
   _7_7 = .;
   _7_8 = .;
   _7_9 = .;
   _7_10 = .;
   _7_11 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm20 $ 20; drop _dm20 ;
   %DMNORMCP( Product_S , _dm20 )
   if _dm20 = 'OTHER'  then do;
      _7_6 = 1;
   end;
   else if _dm20 = 'ELITE'  then do;
      _7_5 = 1;
   end;
   else if _dm20 = 'STUDENTACHIEVER'  then do;
      _7_11 = 1;
   end;
   else if _dm20 = 'PRESTIGE'  then do;
      _7_7 = 1;
   end;
   else if _dm20 = 'CTA'  then do;
      _7_2 = 1;
   end;
   else if _dm20 = 'ACHIEVERGO'  then do;
      _7_1 = 1;
   end;
   else if _dm20 = 'PRIVATE (140)'  then do;
      _7_8 = 1;
   end;
   else if _dm20 = 'CONSOLIDATOR'  then do;
      _7_4 = 1;
   end;
   else if _dm20 = 'VALUEACCOUNT'  then do;
      _7_0 = -1;
      _7_1 = -1;
      _7_2 = -1;
      _7_3 = -1;
      _7_4 = -1;
      _7_5 = -1;
      _7_6 = -1;
      _7_7 = -1;
      _7_8 = -1;
      _7_9 = -1;
      _7_10 = -1;
      _7_11 = -1;
   end;
   else if _dm20 = 'CLASSIC'  then do;
      _7_3 = 1;
   end;
   else if _dm20 = 'PRIVATE (800)'  then do;
      _7_9 = 1;
   end;
   else if _dm20 = 'STAFFCURRENTACCOUNT'  then do;
      _7_10 = 1;
   end;
   else if _dm20 = 'ACHIEVER'  then do;
      _7_0 = 1;
   end;
   else do;
      _7_0 = .;
      _7_1 = .;
      _7_2 = .;
      _7_3 = .;
      _7_4 = .;
      _7_5 = .;
      _7_6 = .;
      _7_7 = .;
      _7_8 = .;
      _7_9 = .;
      _7_10 = .;
      _7_11 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for bus_p ;
drop _9_0 ;
if missing( bus_p ) then do;
   _9_0 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( bus_p , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '0'  then do;
      _9_0 = 1;
   end;
   else if _dm12 = '1'  then do;
      _9_0 = -1;
   end;
   else do;
      _9_0 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for contacts ;
drop _10_0 _10_1 _10_2 ;
if missing( contacts ) then do;
   _10_0 = .;
   _10_1 = .;
   _10_2 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( contacts , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '1'  then do;
      _10_0 = 0;
      _10_1 = 1;
      _10_2 = 0;
   end;
   else if _dm12 = '2'  then do;
      _10_0 = 0;
      _10_1 = 0;
      _10_2 = 1;
   end;
   else if _dm12 = '3'  then do;
      _10_0 = -1;
      _10_1 = -1;
      _10_2 = -1;
   end;
   else if _dm12 = '0'  then do;
      _10_0 = 1;
      _10_1 = 0;
      _10_2 = 0;
   end;
   else do;
      _10_0 = .;
      _10_1 = .;
      _10_2 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for g_dep_band ;
drop _12_0 _12_1 _12_2 _12_3 ;
*** encoding is sparse, initialize to zero;
_12_0 = 0;
```

```
_12_1 = 0;
_12_2 = 0;
_12_3 = 0;
if missing( g_dep_band ) then do;
   _12_0 = .;
   _12_1 = .;
   _12_2 = .;
   _12_3 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm28 $ 28; drop _dm28 ;
   %DMNORMCP( g_dep_band , _dm28 )
   if _dm28 = 'C.DORMANT'  then do;
      _12_2 = 1;
   end;
   else if _dm28 = 'D.SLOW INCREASE'  then do;
      _12_3 = 1;
   end;
   else if _dm28 = 'B.DECLINE'  then do;
      _12_1 = 1;
   end;
   else if _dm28 = 'E.STEEP INCREASE'  then do;
      _12_0 = -1;
      _12_1 = -1;
      _12_2 = -1;
      _12_3 = -1;
   end;
   else if _dm28 = 'A.STEEP DECLINE'  then do;
      _12_0 = 1;
   end;
   else do;
      _12_0 = .;
      _12_1 = .;
      _12_2 = .;
      _12_3 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for home_p ;
drop _15_0 ;
if missing( home_p ) then do;
   _15_0 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
```

```
else do;
   length _dm12 $ 12; drop _dm12 ;
   _dm12 = put( home_p , BEST12. );
   %DMNORMIP( _dm12 )
   if _dm12 = '0'  then do;
      _15_0 = 1;
   end;
   else if _dm12 = '1'  then do;
      _15_0 = -1;
   end;
   else do;
      _15_0 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** Generate dummy variables for salary_group ;
drop _18_0 _18_1 _18_2 _18_3 _18_4 _18_5 ;
*** encoding is sparse, initialize to zero;
_18_0 = 0;
_18_1 = 0;
_18_2 = 0;
_18_3 = 0;
_18_4 = 0;
_18_5 = 0;
if missing( salary_group ) then do;
   _18_0 = .;
   _18_1 = .;
   _18_2 = .;
   _18_3 = .;
   _18_4 = .;
   _18_5 = .;
   substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;
else do;
   length _dm20 $ 20; drop _dm20 ;
   %DMNORMCP( salary_group , _dm20 )
   if _dm20 = 'A.[0 - 3K)'  then do;
      _18_0 = 1;
   end;
   else if _dm20 = 'B.[3K - 8K)'  then do;
      _18_1 = 1;
   end;
   else if _dm20 = 'C.[8K - 15K)'  then do;
      _18_2 = 1;
   end;
   else if _dm20 = 'D.[15K - 25K)'  then do;
```

```
      _18_3 = 1;
   end;
   else if _dm20 = 'E.[25K - 40K)'  then do;
      _18_4 = 1;
   end;
   else if _dm20 = 'G.[60K - @@)'  then do;
      _18_0 = -1;
      _18_1 = -1;
      _18_2 = -1;
      _18_3 = -1;
      _18_4 = -1;
      _18_5 = -1;
   end;
   else if _dm20 = 'F.[40K - 60K)'  then do;
      _18_5 = 1;
   end;
   else do;
      _18_0 = .;
      _18_1 = .;
      _18_2 = .;
      _18_3 = .;
      _18_4 = .;
      _18_5 = .;
      substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
   end;
end;

*** If missing inputs, use averages;
if _DM_BAD > 0 then do;
   _P0 = 0.175884759;
   _P1 = 0.2536012805;
   _P2 = 0.0633113996;
   _P3 = 0.2536012805;
   _P4 = 0.2536012805;
   goto REGDR1;
end;

*** Compute Linear Predictor;
drop _TEMP;
drop _LP0  _LP1 _LP2 _LP3;
_LP0 = 0;
_LP1 = 0;
_LP2 = 0;
_LP3 = 0;

***  Effect: AGE_GROUP ;
_TEMP = 1;
_LP0 = _LP0 + (    0.87785483278105) * _TEMP * _1_0;
```

```
_LP1 = _LP1 + (     0.02758358091987) * _TEMP * _1_0;
_LP2 = _LP2 + (     0.46007497819997) * _TEMP * _1_0;
_LP3 = _LP3 + (    -0.04214254533298) * _TEMP * _1_0;
_LP0 = _LP0 + (     0.35225165801596) * _TEMP * _1_1;
_LP1 = _LP1 + (     0.08116402127161) * _TEMP * _1_1;
_LP2 = _LP2 + (     0.26588338137001) * _TEMP * _1_1;
_LP3 = _LP3 + (    -0.27674770865706) * _TEMP * _1_1;
_LP0 = _LP0 + (    -0.14522483520453) * _TEMP * _1_2;
_LP1 = _LP1 + (    -0.00708640560967) * _TEMP * _1_2;
_LP2 = _LP2 + (    -0.04058682941402) * _TEMP * _1_2;
_LP3 = _LP3 + (    -0.21301251310724) * _TEMP * _1_2;
_LP0 = _LP0 + (    -0.45192302991722) * _TEMP * _1_3;
_LP1 = _LP1 + (     0.14786737297211) * _TEMP * _1_3;
_LP2 = _LP2 + (    -0.21644523080116) * _TEMP * _1_3;
_LP3 = _LP3 + (     0.16876073343022) * _TEMP * _1_3;

***  Effect: Ethnicity_Desc ;
_TEMP = 1;
_LP0 = _LP0 + (     0.51114958222504) * _TEMP * _3_0;
_LP1 = _LP1 + (     1.86141336760198) * _TEMP * _3_0;
_LP2 = _LP2 + (     0.68014721240495) * _TEMP * _3_0;
_LP3 = _LP3 + (     1.44545757007455) * _TEMP * _3_0;
_LP0 = _LP0 + (     1.00796634367875) * _TEMP * _3_1;
_LP1 = _LP1 + (     1.43520957162287) * _TEMP * _3_1;
_LP2 = _LP2 + (     0.50972195180205) * _TEMP * _3_1;
_LP3 = _LP3 + (     1.16668990596439) * _TEMP * _3_1;
_LP0 = _LP0 + (     0.52149411175979) * _TEMP * _3_2;
_LP1 = _LP1 + (     1.67716792513442) * _TEMP * _3_2;
_LP2 = _LP2 + (     0.51616293126679) * _TEMP * _3_2;
_LP3 = _LP3 + (     1.05703306779482) * _TEMP * _3_2;
_LP0 = _LP0 + (    -3.52498019373668) * _TEMP * _3_3;
_LP1 = _LP1 + (    -5.90170630730973) * _TEMP * _3_3;
_LP2 = _LP2 + (    -3.88490767338708) * _TEMP * _3_3;
_LP3 = _LP3 + (    -6.18583020260086) * _TEMP * _3_3;
_LP0 = _LP0 + (     0.57541682493071) * _TEMP * _3_4;
_LP1 = _LP1 + (    -0.28789197903883) * _TEMP * _3_4;
_LP2 = _LP2 + (     0.95503332598123) * _TEMP * _3_4;
_LP3 = _LP3 + (     1.52945733594434) * _TEMP * _3_4;

***  Effect: Gender_Type_Desc ;
_TEMP = 1;
_LP0 = _LP0 + (      -0.0056088687164) * _TEMP * _4_0;
_LP1 = _LP1 + (    -0.15759250546776) * _TEMP * _4_0;
_LP2 = _LP2 + (     0.02528879253795) * _TEMP * _4_0;
_LP3 = _LP3 + (     0.13684943334771) * _TEMP * _4_0;

***  Effect: Marital_Status ;
_TEMP = 1;
_LP0 = _LP0 + (    -1.11604105901877) * _TEMP * _5_0;
```

```
_LP1 = _LP1 + (     1.06897894531762) * _TEMP * _5_0;
_LP2 = _LP2 + (     0.21685544664234) * _TEMP * _5_0;
_LP3 = _LP3 + (     0.09069285712701) * _TEMP * _5_0;
_LP0 = _LP0 + (    -0.65745008150838) * _TEMP * _5_1;
_LP1 = _LP1 + (     0.85773440239199) * _TEMP * _5_1;
_LP2 = _LP2 + (    -0.50857170116797) * _TEMP * _5_1;
_LP3 = _LP3 + (    -0.15297977648945) * _TEMP * _5_1;
_LP0 = _LP0 + (    -1.91691792941586) * _TEMP * _5_2;
_LP1 = _LP1 + (    -5.25772858294855) * _TEMP * _5_2;
_LP2 = _LP2 + (    -6.59383063073055) * _TEMP * _5_2;
_LP3 = _LP3 + (    -0.93298209632964) * _TEMP * _5_2;
_LP0 = _LP0 + (     6.96803272715947) * _TEMP * _5_3;
_LP1 = _LP1 + (     0.47876848782757) * _TEMP * _5_3;
_LP2 = _LP2 + (     10.6909867358537) * _TEMP * _5_3;
_LP3 = _LP3 + (     0.44981371779966) * _TEMP * _5_3;
_LP0 = _LP0 + (    -1.27084910089527) * _TEMP * _5_4;
_LP1 = _LP1 + (     1.01503873152244) * _TEMP * _5_4;
_LP2 = _LP2 + (    -0.79412739412886) * _TEMP * _5_4;
_LP3 = _LP3 + (    -0.13830464718482) * _TEMP * _5_4;
_LP0 = _LP0 + (    -2.06518235153294) * _TEMP * _5_5;
_LP1 = _LP1 + (      -0.369746626222) * _TEMP * _5_5;
_LP2 = _LP2 + (    -2.35266674926515) * _TEMP * _5_5;
_LP3 = _LP3 + (    -0.77117660525014) * _TEMP * _5_5;

***  Effect: OD ;
_TEMP = OD ;
_LP0 = _LP0 + (     -0.4555333687032 * _TEMP);
_LP1 = _LP1 + (     0.12233628079603 * _TEMP);
_LP2 = _LP2 + (    -0.04670298817189 * _TEMP);
_LP3 = _LP3 + (    -0.00330210888825 * _TEMP);

***  Effect: Preferred_Lang ;
_TEMP = 1;
_LP0 = _LP0 + (     0.31077799559055) * _TEMP * _6_0;
_LP1 = _LP1 + (     2.26972584150899) * _TEMP * _6_0;
_LP2 = _LP2 + (     1.38541565314526) * _TEMP * _6_0;
_LP3 = _LP3 + (     0.65477715706154) * _TEMP * _6_0;
_LP0 = _LP0 + (     0.20342928415994) * _TEMP * _6_1;
_LP1 = _LP1 + (     2.22367533838391) * _TEMP * _6_1;
_LP2 = _LP2 + (     2.58593281442323) * _TEMP * _6_1;
_LP3 = _LP3 + (     0.84697631113895) * _TEMP * _6_1;

***  Effect: Product_S ;
_TEMP = 1;
_LP0 = _LP0 + (    -0.47561897275838) * _TEMP * _7_0;
_LP1 = _LP1 + (    -0.68110543770952) * _TEMP * _7_0;
_LP2 = _LP2 + (     2.83390295524645) * _TEMP * _7_0;
_LP3 = _LP3 + (       6.163603229836) * _TEMP * _7_0;
_LP0 = _LP0 + (    -1.50636857373753) * _TEMP * _7_1;
```

```
_LP1 = _LP1 + (    -1.01530699036594) * _TEMP * _7_1;
_LP2 = _LP2 + (     -1.0087996071919) * _TEMP * _7_1;
_LP3 = _LP3 + (    -1.08810317542961) * _TEMP * _7_1;
_LP0 = _LP0 + (    -1.41352653069106) * _TEMP * _7_2;
_LP1 = _LP1 + (     0.73796829456964) * _TEMP * _7_2;
_LP2 = _LP2 + (    -2.10785779681061) * _TEMP * _7_2;
_LP3 = _LP3 + (    -1.33700449879339) * _TEMP * _7_2;
_LP0 = _LP0 + (    -1.88403154322633) * _TEMP * _7_3;
_LP1 = _LP1 + (     5.62446779712567) * _TEMP * _7_3;
_LP2 = _LP2 + (     4.77197741716435) * _TEMP * _7_3;
_LP3 = _LP3 + (     4.03962644273863) * _TEMP * _7_3;
_LP0 = _LP0 + (    -1.58995748701088) * _TEMP * _7_4;
_LP1 = _LP1 + (    -0.50379113723591) * _TEMP * _7_4;
_LP2 = _LP2 + (    -7.82552799711789) * _TEMP * _7_4;
_LP3 = _LP3 + (    -2.13587464343475) * _TEMP * _7_4;
_LP0 = _LP0 + (    -0.27974232770704) * _TEMP * _7_5;
_LP1 = _LP1 + (    -0.02382329399084) * _TEMP * _7_5;
_LP2 = _LP2 + (    -1.29509997175635) * _TEMP * _7_5;
_LP3 = _LP3 + (    -1.33363624295683) * _TEMP * _7_5;
_LP0 = _LP0 + (     -0.6852221500191) * _TEMP * _7_6;
_LP1 = _LP1 + (     0.77279045201344) * _TEMP * _7_6;
_LP2 = _LP2 + (    -1.76012420467729) * _TEMP * _7_6;
_LP3 = _LP3 + (    -1.03500691456799) * _TEMP * _7_6;
_LP0 = _LP0 + (     0.38387616269333) * _TEMP * _7_7;
_LP1 = _LP1 + (     -0.0525134972082) * _TEMP * _7_7;
_LP2 = _LP2 + (    -0.91396113087927) * _TEMP * _7_7;
_LP3 = _LP3 + (    -0.59292257109897) * _TEMP * _7_7;
_LP0 = _LP0 + (     1.61014106238248) * _TEMP * _7_8;
_LP1 = _LP1 + (     0.75650316456172) * _TEMP * _7_8;
_LP2 = _LP2 + (     0.06345423998319) * _TEMP * _7_8;
_LP3 = _LP3 + (    -0.37161312150058) * _TEMP * _7_8;
_LP0 = _LP0 + (     8.76078491783214) * _TEMP * _7_9;
_LP1 = _LP1 + (      1.1199459807578) * _TEMP * _7_9;
_LP2 = _LP2 + (    -2.40312911576993) * _TEMP * _7_9;
_LP3 = _LP3 + (     -0.6054627139215) * _TEMP * _7_9;
_LP0 = _LP0 + (    -1.00973434625343) * _TEMP * _7_10;
_LP1 = _LP1 + (    -1.80151426581029) * _TEMP * _7_10;
_LP2 = _LP2 + (     16.6973380449376) * _TEMP * _7_10;
_LP3 = _LP3 + (    -1.47605601848781) * _TEMP * _7_10;
_LP0 = _LP0 + (    -1.44299247948416) * _TEMP * _7_11;
_LP1 = _LP1 + (    -0.78169692522397) * _TEMP * _7_11;
_LP2 = _LP2 + (    -0.24680211569636) * _TEMP * _7_11;
_LP3 = _LP3 + (    -0.89095014611876) * _TEMP * _7_11;

***  Effect: ave_month_end_bal ;
_TEMP = ave_month_end_bal ;
_LP0 = _LP0 + (  1.796113908669E-6 * _TEMP);
_LP1 = _LP1 + ( -1.768764645940E-6 * _TEMP);
_LP2 = _LP2 + (  2.283665122338E-6 * _TEMP);
```

```
_LP3 = _LP3 + (  2.5540546131084E-6 * _TEMP);

***  Effect: average_pos ;
_TEMP = average_pos ;
_LP0 = _LP0 + (    0.00007670740917 * _TEMP);
_LP1 = _LP1 + (   -0.00001630417215 * _TEMP);
_LP2 = _LP2 + (   -0.00004938006489 * _TEMP);
_LP3 = _LP3 + (    -0.0000412954758 * _TEMP);

***  Effect: average_wdrw ;
_TEMP = average_wdrw ;
_LP0 = _LP0 + (    0.08052070849603 * _TEMP);
_LP1 = _LP1 + (    0.12611014328797 * _TEMP);
_LP2 = _LP2 + (    0.06204520293007 * _TEMP);
_LP3 = _LP3 + (    0.06711381868175 * _TEMP);

***  Effect: bal_diff_4 ;
_TEMP = bal_diff_4 ;
_LP0 = _LP0 + ( -5.7445353881921E-6 * _TEMP);
_LP1 = _LP1 + ( -5.7134988818908E-6 * _TEMP);
_LP2 = _LP2 + ( -6.3117275860336E-7 * _TEMP);
_LP3 = _LP3 + (  4.0575461377161E-6 * _TEMP);

***  Effect: bus_p ;
_TEMP = 1;
_LP0 = _LP0 + (    0.10872013288827) * _TEMP * _9_0;
_LP1 = _LP1 + (   -0.02527646132367) * _TEMP * _9_0;
_LP2 = _LP2 + (   -1.86970161865719) * _TEMP * _9_0;
_LP3 = _LP3 + (    0.36182992566982) * _TEMP * _9_0;

***  Effect: contacts ;
_TEMP = 1;
_LP0 = _LP0 + (   -0.81283221251323) * _TEMP * _10_0;
_LP1 = _LP1 + (   -0.66059615589708) * _TEMP * _10_0;
_LP2 = _LP2 + (    1.19435946135309) * _TEMP * _10_0;
_LP3 = _LP3 + (   -0.90496305412338) * _TEMP * _10_0;
_LP0 = _LP0 + (   -0.41075307297889) * _TEMP * _10_1;
_LP1 = _LP1 + (     0.0192432624001) * _TEMP * _10_1;
_LP2 = _LP2 + (   -0.71250994899883) * _TEMP * _10_1;
_LP3 = _LP3 + (   -0.37593582829826) * _TEMP * _10_1;
_LP0 = _LP0 + (    0.36167431609904) * _TEMP * _10_2;
_LP1 = _LP1 + (    0.13637385325897) * _TEMP * _10_2;
_LP2 = _LP2 + (    0.20698030095822) * _TEMP * _10_2;
_LP3 = _LP3 + (    0.32951185045155) * _TEMP * _10_2;

***  Effect: g_dep_band ;
_TEMP = 1;
_LP0 = _LP0 + (   -0.08846934397583) * _TEMP * _12_0;
_LP1 = _LP1 + (   -0.81242677935395) * _TEMP * _12_0;
```

```
_LP2 = _LP2 + (   -0.34488196190592) * _TEMP * _12_0;
_LP3 = _LP3 + (   -0.24002124177848) * _TEMP * _12_0;
_LP0 = _LP0 + (   -0.14031661231783) * _TEMP * _12_1;
_LP1 = _LP1 + (    0.14841485126363) * _TEMP * _12_1;
_LP2 = _LP2 + (    0.10566955761772) * _TEMP * _12_1;
_LP3 = _LP3 + (    0.04643963744948) * _TEMP * _12_1;
_LP0 = _LP0 + (    0.03814295749619) * _TEMP * _12_2;
_LP1 = _LP1 + (    0.41285759514945) * _TEMP * _12_2;
_LP2 = _LP2 + (    0.14310558231191) * _TEMP * _12_2;
_LP3 = _LP3 + (   -0.11847141669544) * _TEMP * _12_2;
_LP0 = _LP0 + (    0.11434720001953) * _TEMP * _12_3;
_LP1 = _LP1 + (    0.27103609446418) * _TEMP * _12_3;
_LP2 = _LP2 + (    0.32291587595805) * _TEMP * _12_3;
_LP3 = _LP3 + (    0.19176352489662) * _TEMP * _12_3;

***  Effect: g_tot_spend_ ;
_TEMP = g_tot_spend_ ;
_LP0 = _LP0 + (    0.19264701957124 * _TEMP);
_LP1 = _LP1 + (    0.40983896635293 * _TEMP);
_LP2 = _LP2 + (    0.06259409170674 * _TEMP);
_LP3 = _LP3 + (    0.00644347233053 * _TEMP);

***  Effect: home_p ;
_TEMP = 1;
_LP0 = _LP0 + (    0.10104764710672) * _TEMP * _15_0;
_LP1 = _LP1 + (    0.17378526110927) * _TEMP * _15_0;
_LP2 = _LP2 + (   -0.63340950990617) * _TEMP * _15_0;
_LP3 = _LP3 + (     0.3304396752026) * _TEMP * _15_0;

***  Effect: salary_group ;
_TEMP = 1;
_LP0 = _LP0 + (   -2.34536831642152) * _TEMP * _18_0;
_LP1 = _LP1 + (   -0.08872063726272) * _TEMP * _18_0;
_LP2 = _LP2 + (    0.15246274994396) * _TEMP * _18_0;
_LP3 = _LP3 + (   -0.53000019302075) * _TEMP * _18_0;
_LP0 = _LP0 + (    0.07843908285776) * _TEMP * _18_1;
_LP1 = _LP1 + (    0.93252233107954) * _TEMP * _18_1;
_LP2 = _LP2 + (   -0.56982866502539) * _TEMP * _18_1;
_LP3 = _LP3 + (    0.22667829338864) * _TEMP * _18_1;
_LP0 = _LP0 + (     1.0182327739118) * _TEMP * _18_2;
_LP1 = _LP1 + (    0.36852602014212) * _TEMP * _18_2;
_LP2 = _LP2 + (    0.09186629380121) * _TEMP * _18_2;
_LP3 = _LP3 + (     0.4443089681112) * _TEMP * _18_2;
_LP0 = _LP0 + (    1.10729338527489) * _TEMP * _18_3;
_LP1 = _LP1 + (     0.3753374759241) * _TEMP * _18_3;
_LP2 = _LP2 + (    0.47899034549857) * _TEMP * _18_3;
_LP3 = _LP3 + (     0.5058610102247) * _TEMP * _18_3;
_LP0 = _LP0 + (    0.52660910790276) * _TEMP * _18_4;
_LP1 = _LP1 + (   -0.15311296218599) * _TEMP * _18_4;
```

```
_LP2 = _LP2 + (    -0.1675238795499) * _TEMP * _18_4;
_LP3 = _LP3 + (   -0.11488495511202) * _TEMP * _18_4;
_LP0 = _LP0 + (    0.17834886495998) * _TEMP * _18_5;
_LP1 = _LP1 + (   -0.65485892436405) * _TEMP * _18_5;
_LP2 = _LP2 + (    0.17276326096394) * _TEMP * _18_5;
_LP3 = _LP3 + (   -0.33974074038693) * _TEMP * _18_5;

***  Effect: tot_spend_4 ;
_TEMP = tot_spend_4 ;
_LP0 = _LP0 + ( -3.0886852882416E-6 * _TEMP);
_LP1 = _LP1 + ( -6.1495013807506E-7 * _TEMP);
_LP2 = _LP2 + ( -2.2923645163206E-6 * _TEMP);
_LP3 = _LP3 + ( -1.5291327882472E-6 * _TEMP);

*** Naive Posterior Probabilities;
drop _MAXP _IY _P0 _P1 _P2 _P3 _P4;
drop _LPMAX;
_LPMAX= 0;
_LP0 =    -0.43185065243911 + _LP0;
if _LPMAX < _LP0 then _LPMAX = _LP0;
_LP1 =    -6.25552467458312 + _LP1;
if _LPMAX < _LP1 then _LPMAX = _LP1;
_LP2 =    -3.74366155302926 + _LP2;
if _LPMAX < _LP2 then _LPMAX = _LP2;
_LP3 =    -1.1487270752530 9 + _LP3;
if _LPMAX < _LP3 then _LPMAX = _LP3;
_LP0 = exp(_LP0 - _LPMAX);
_LP1 = exp(_LP1 - _LPMAX);
_LP2 = exp(_LP2 - _LPMAX);
_LP3 = exp(_LP3 - _LPMAX);
_LPMAX = exp(-_LPMAX);
_P4 = 1 / (_LPMAX + _LP0 + _LP1 + _LP2 + _LP3);
_P0 = _LP0 * _P4;
_P1 = _LP1 * _P4;
_P2 = _LP2 * _P4;
_P3 = _LP3 * _P4;
_P4 = _LPMAX * _P4;

REGDR1:


*** Posterior Probabilities and Predicted Level;
label P_N_PRODUCTe_CARD = 'Predicted: N_PRODUCT=e.CARD' ;
label P_N_PRODUCTd_UL = 'Predicted: N_PRODUCT=d.UL' ;
label P_N_PRODUCTc_SL = 'Predicted: N_PRODUCT=c.SL' ;
label P_N_PRODUCTb_INV = 'Predicted: N_PRODUCT=b.INV' ;
label P_N_PRODUCTNO_TAKE = 'Predicted: N_PRODUCT=NO_TAKE' ;
P_N_PRODUCTe_CARD = _P0;
_MAXP = _P0;
```

```
_IY = 1;
P_N_PRODUCTd_UL = _P1;
if (_P1 - _MAXP > 1e-8) then do;
   _MAXP = _P1;
   _IY = 2;
end;
P_N_PRODUCTc_SL = _P2;
if (_P2 - _MAXP > 1e-8) then do;
   _MAXP = _P2;
   _IY = 3;
end;
P_N_PRODUCTb_INV = _P3;
if (_P3 - _MAXP > 1e-8) then do;
   _MAXP = _P3;
   _IY = 4;
end;
P_N_PRODUCTNO_TAKE = _P4;
if (_P4 - _MAXP > 1e-8) then do;
   _MAXP = _P4;
   _IY = 5;
end;
I_N_PRODUCT = REGDRF[_IY];
U_N_PRODUCT = REGDRU[_IY];

*************************************;
***** end scoring code for regression;
*************************************;




*------------------------------------------------------------
*;
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score;
*------------------------------------------------------------
*;
*------------------------------------------------------------
*;
* Score: Creating Fixed Names;
*------------------------------------------------------------
*;
LABEL EM_EVENTPROBABILITY = 'Probability for level E.CARD of
N_PRODUCT';
EM_EVENTPROBABILITY = P_N_PRODUCTe_CARD;
LABEL EM_PROBABILITY = 'Probability of Classification';
EM_PROBABILITY = max(
P_N_PRODUCTe_CARD
,
P_N_PRODUCTd_UL
,
P_N_PRODUCTc_SL
,
P_N_PRODUCTb_INV
,
P_N_PRODUCTNO_TAKE
);
LENGTH EM_CLASSIFICATION $%dmnorlen;
LABEL EM_CLASSIFICATION = "Prediction for N_PRODUCT";
EM_CLASSIFICATION = I_N_PRODUCT;
```

7.4.3 MBLR model

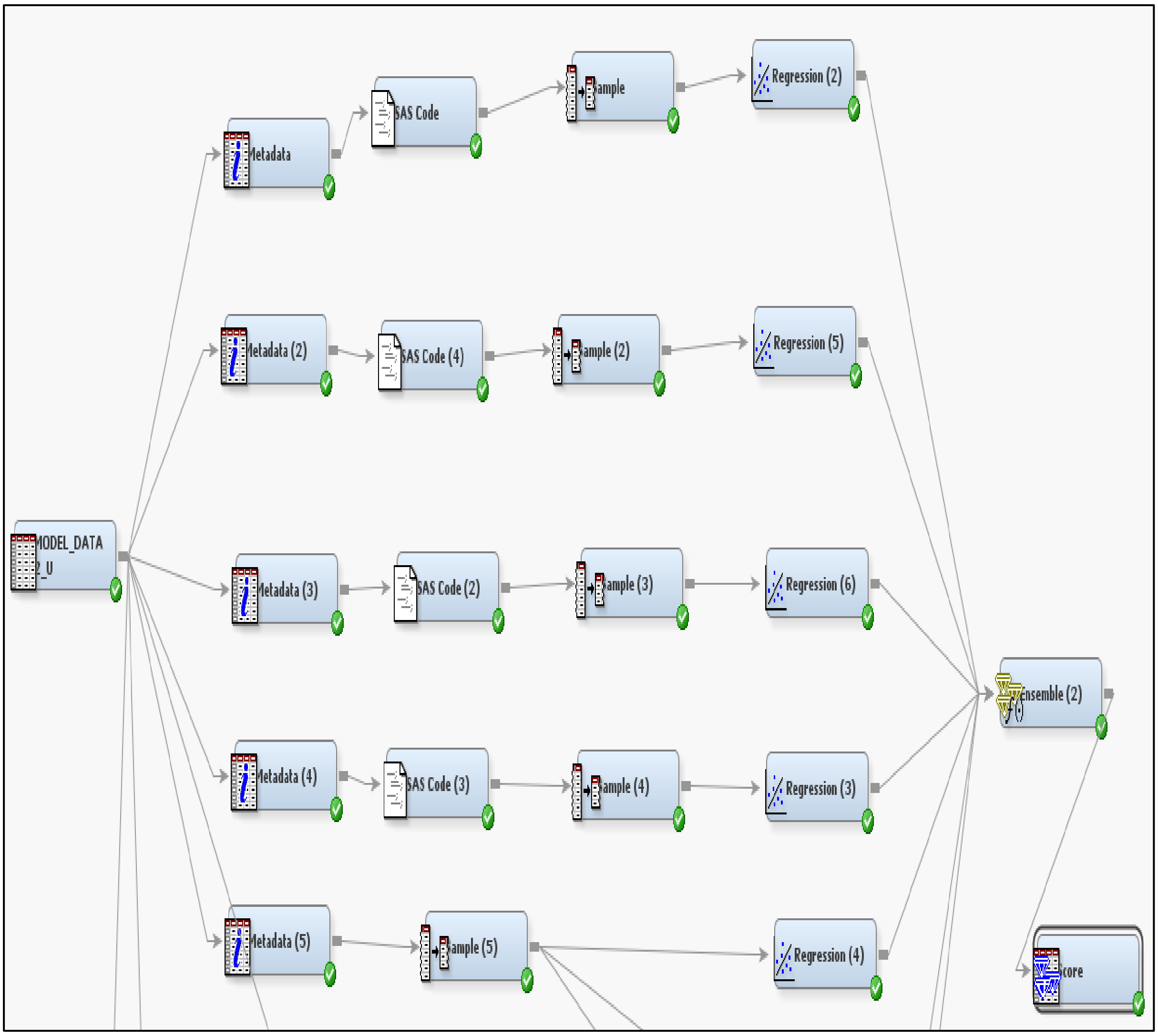

FIGURE 16 : BAGGING USING ENTERPRISE MINER

Figure 16 above illustrates the Bagging process as carried out in Enterprise Miner. A total of eight logistic regression models were developed.

7.4.4 MRFD

Listed below is the script for the Random forests model as deployed in R.

```
# Creating the data partitions

index= sample(1:nrow(model_base_U),as.integer(nrow(model_base_U)*.75),replace=FALSE)
train = model_base_U[index,]
test = model_base_U[-index,]
> View(train)

# New data sets after correlation tests

>new_train=subset(train,select=-c(average_op_income,average_nir ,average_nii,average_liabilities,average_assets,average_st_liab,average_st_assets,total_txns,average_txnal_balance,average_fees,Product_S,max_balance,average_credit,average_debit,average_wdrw,average_pos,average_digital,average_enq,ave_debit_bal,average_do))
>new_test=subset(test,select=-c(average_op_income,average_nir ,average_nii,average_liabilities,average_assets,average_st_liab,average_st_assets,total_txns,average_txnal_balance,average_fees,Product_S,max_balance,average_credit,average_debit,average_wdrw,average_pos,average_digital,average_enq,ave_debit_bal,average_do))


#Model Fitting

> MultiClass_RF <- randomForest(N_PRODUCT ~ ., data= new_train, ntree=1000, keep.forest=FALSE,importance=TRUE)
> MultiClass_RF
> varImpPlot(MultiClass_RF)

#Probabilities

new_probs =
predict(MultiClass_RF,newdata=new_test,type="class",predict.all=FALSE,cutoff=c(0
.40,0.60))
result3a=as.data.frame(new_probs)
new_data3=cbind(new_test,result3a)

#Calculating AUC

>MultiClass_RF.pred2=
predict(MultiClass_RF,newdata=new_test,type="prob",predict.all=FALSE)
> summary(MultiClass_RF.pred2)
> result_all2=as.data.frame(MultiClass_RF.pred2)
>> N_PRODUCT=subset(new_test, select=c(N_PRODUCT))
> N_PRODUCT$N_PRODUCT <- as.factor( N_PRODUCT$N_PRODUCT)
  x_all2=cbind(N_PRODUCT,result_all2)
> auc(multcap( response = x_all2$N_PRODUCT, predicted = as.matrix(x_all2[,
levels(x_all2$N_PRODUCT)])))
[1] 0.7430778

#Box plots

> par(mfrow=c(1,2))
rb1 <- boxplot(c.SL ~ N_PRODUCT, data = reg_validate2,main = "Distribution of
c.SL",boxwex = 0.25,at= 1:5, col = "bisque", xlab = "Response Class", ylab =
"Probability",yaxs = "i",outwex = 0.4, cex.axis=0.4)  #boxplot
```

```
> rb1 <- boxplot(e.CARD ~ N_PRODUCT, data = reg_validate2,main = "Distribution of e.CARD",boxwex = 0.25,at= 1:5, col = "bisque", xlab = "Response Class", ylab = "Probability",yaxs = "i",outwex = 0.4, cex.axis=0.4)  #boxplot
```

7.4.5 ECOC

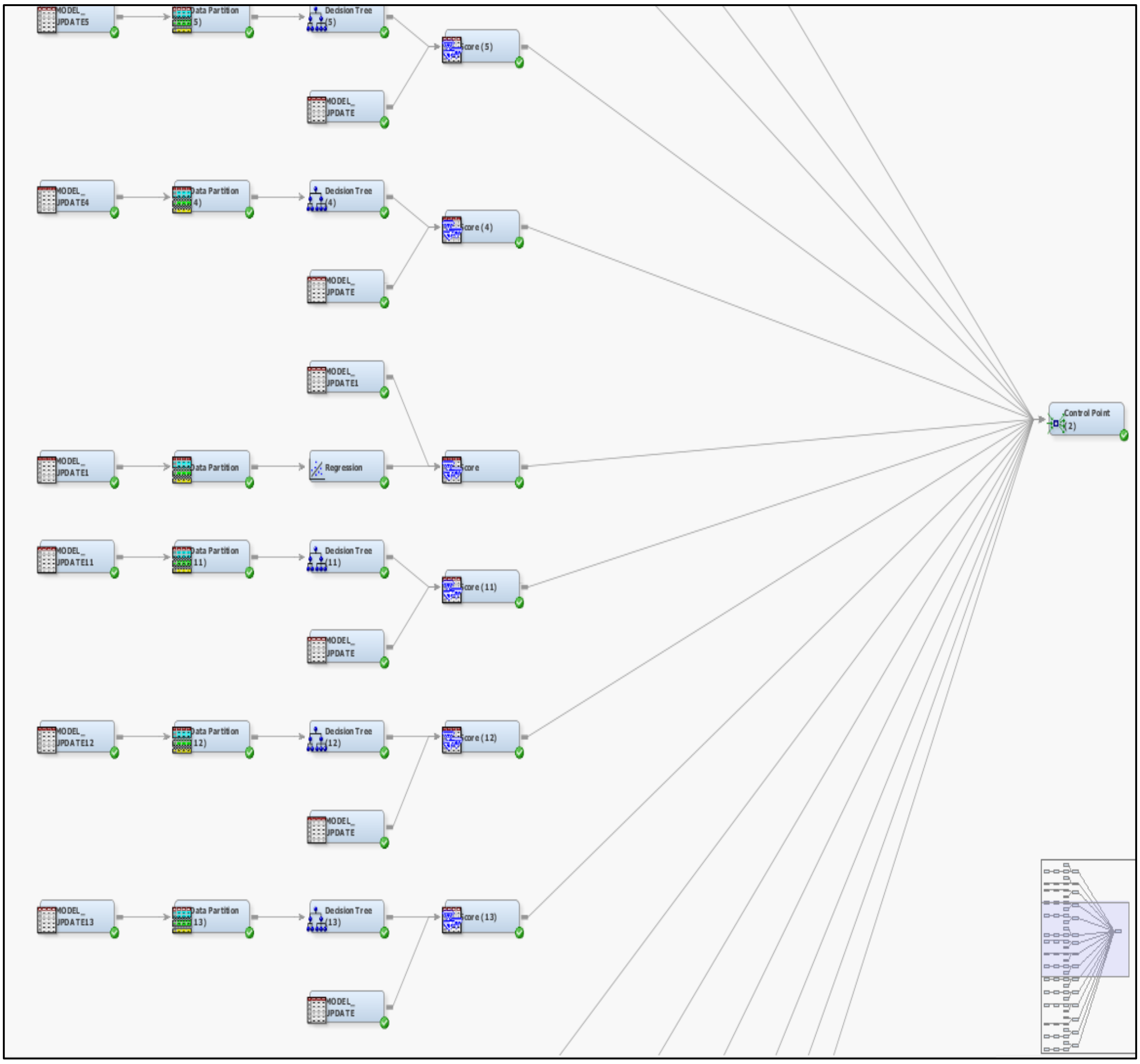

Figure 17

Figure 17 above illustrates the models process for ECOC as carried out in Enterprise Miner. A total of fifteen models were developed as there were 15 binary strings in the code word.

After the 15 models were fit , the code below details how the code strings for the ECOC model were computed and the plots thereof.

```
/***** create a unique coding string in SAS
*******************/

data jon.Coding_matrix1;
set Coding_matrix;
length Code $15;
length N_PRODUCT $8;
Code=compress(T1||T2||T3||T4||T5||T6||T7||T8||T9||T10||T11||T1
2||T13||T14||T15);
if product =1 then N_PRODUCT='b.INV';
  else if product =2 then N_PRODUCT='c.SL';
  else if product =3 then N_PRODUCT='d.UL';
  else if product =4 then N_PRODUCT='e.CARD';
  else if product =5 then N_PRODUCT='NO_TAKE';
run;

/***** combine the string with the original modelling data set
*******************/
proc sql;
create table jon.model_data2_u as select
   a.*,
   b.*

from jon.model_data2_u  a ,
     jon.Coding_matrix1 b

  where a.N_PRODUCT=b.N_PRODUCT ;
quit;

/***** Sampling process for each product *******************/

%macro sample;

proc sql;
          create table counts as
          select Target_S,
                          count(1) as COUNT
          from jon.model_data2_u
          group by Target_S;
     run;

     data _null_;
```

```
          set counts(where=(Target_S=1));
          call symput('count1',count);
     run;


     data _null_;
          call symput('rate',&count1*4);
     run;

     proc                                                  surveyselect
data=jon.model_data2_u(where=(Target_I=1)) out=base1_sample
                                             method=srs
n=&rate seed=3287;
     run;
     proc                                                  surveyselect
data=jon.model_data2_u(where=(Target_U=1)) out=base2_sample
                                             method=srs
n=&rate seed=8471;
     run;
     proc                                                  surveyselect
data=jon.model_data2_u(where=(Target_C=1)) out=base3_sample
                                             method=srs  n=1238
seed=6736;
     run;
     proc                                                  surveyselect
data=jon.model_data2_u(where=(Target_I=0)) out=base4_sample
                                             method=srs
n=&rate seed=4355;
     run;

     data base ;
          set base1_sample
              base2_sample
               base3_sample
               base4_sample
                    jon.model_data2_u(where=(Target_S=1));
     run;

%mend;
%sample

data jon.model_update;
set base;
run;


/*****   obtaim   scores   for   each   of   the   15   modelled
strings******/
```

```
%macro freq(ind,target);
proc freq data=jon.model_update;
 tables &target.&ind. ;
 run;
%mend ;
%macro loop;
%do i=1 %to 15;
%freq(&i,T);
%end;
%mend;
%loop ;

data score_3;
set em.score3_score;
keep id10 new_cust_n P_T31;
run;



%macro look;
data score_&i ;
  set em.score&i._score ;
  keep id10 new_cust_n P_T&i.1;
%mend ;
%macro loop;
%do i=2 %to 15;
%look;
%end;
%mend;
%loop ;


%macro lt;

%do i=1 %to 15;

%if &i = 1 %then %do;
               data base_fin;
                    set score_&i;
               run;
          %end;

          %else %do;
            proc sort data=score_&i;
               by id10;
           run;
            proc sort data=base_fin;
               by id10;
           run;
               data base_fin;
                    merge base_fin  score_&i;
                    by id10 ;
               run;
          %end;
%end;
%mend;
%lt ;


data jon.base_fin;
set base_fin;
run;


/*** Compute Linear Predictors*****/

data jon.BASE_fin2 BASE_fin2;
  set  jon.base_fin;
  _LP1 = 0;
  _LP2 = 0;
  _LP3 = 0;
  _LP4 = 0;
  _LP5 = 0;

  length pred_class $8;
  ***  Effect: 'b.INV' ;

_LP1 = _LP1 + abs(1-P_T11 );
_LP1 = _LP1 + abs(1-P_T21 );
_LP1 = _LP1 + abs(1-P_T31 );
_LP1 = _LP1 + abs(1-P_T41);
_LP1 = _LP1 + abs(1-P_T51 );
_LP1 = _LP1 + abs(1-P_T61 );
_LP1 = _LP1 + abs(1-P_T71 );
_LP1 = _LP1 + abs(1-P_T81 );
_LP1 = _LP1 + abs(1-P_T91 );
_LP1 = _LP1 + abs(1-P_T101 );
_LP1 = _LP1 + abs(1-P_T111 );
_LP1 = _LP1 + abs(1-P_T121 );
_LP1 = _LP1 + abs(1-P_T131 );
_LP1 = _LP1 + abs(1-P_T141 );
_LP1 = _LP1 + abs(1-P_T151 );

  ***  Effect: 'c.SL' ;

_LP2 = _LP2 + abs(0-P_T11 );
_LP2 = _LP2 + abs(0-P_T21 );
```

```
_LP2 = _LP2 + abs(0-P_T31 );
_LP2 = _LP2 + abs(0-P_T41);
_LP2 = _LP2 + abs(0-P_T51 );
_LP2 = _LP2 + abs(0-P_T61 );
_LP2 = _LP2 + abs(0-P_T71 );
_LP2 = _LP2 + abs(0-P_T81 );
_LP2 = _LP2 + abs(1-P_T91 );
_LP2 = _LP2 + abs(1-P_T101 );
_LP2 = _LP2 + abs(1-P_T111 );
_LP2 = _LP2 + abs(1-P_T121 );
_LP2 = _LP2 + abs(1-P_T131 );
_LP2 = _LP2 + abs(1-P_T141 );
_LP2 = _LP2 + abs(1-P_T151 );

  ***  Effect: 'd.UL' ;

_LP3 = _LP3 + abs(0-P_T11 );
_LP3 = _LP3 + abs(0-P_T21 );
_LP3 = _LP3 + abs(0-P_T31 );
_LP3 = _LP3 + abs(0-P_T41);
_LP3 = _LP3 + abs(1-P_T51 );
_LP3 = _LP3 + abs(1-P_T61 );
_LP3 = _LP3 + abs(1-P_T71 );
_LP3 = _LP3 + abs(1-P_T81 );
_LP3 = _LP3 + abs(0-P_T91 );
_LP3 = _LP3 + abs(0-P_T101 );
_LP3 = _LP3 + abs(0-P_T111 );
_LP3 = _LP3 + abs(0-P_T121 );
_LP3 = _LP3 + abs(1-P_T131 );
_LP3 = _LP3 + abs(1-P_T141 );
_LP3 = _LP3 + abs(1-P_T151 );

  ***  Effect: 'e.CARD' ;

_LP4 = _LP4 + abs(0-P_T11 );
_LP4 = _LP4 + abs(0-P_T21 );
_LP4 = _LP4 + abs(1-P_T31 );
_LP4 = _LP4 + abs(1-P_T41);
_LP4 = _LP4 + abs(0-P_T51 );
_LP4 = _LP4 + abs(0-P_T61 );
_LP4 = _LP4 + abs(1-P_T71 );
_LP4 = _LP4 + abs(1-P_T81 );
_LP4 = _LP4 + abs(0-P_T91 );
_LP4 = _LP4 + abs(0-P_T101 );
_LP4 = _LP4 + abs(1-P_T111 );
_LP4 = _LP4 + abs(1-P_T121 );
_LP4 = _LP4 + abs(0-P_T131 );
_LP4 = _LP4 + abs(0-P_T141 );
_LP4 = _LP4 + abs(1-P_T151 );
```

```
  ***  Effect: 'NO_TAKE' ;

_LP5 = _LP5 + abs(0-P_T11 );
_LP5 = _LP5 + abs(1-P_T21 );
_LP5 = _LP5 + abs(0-P_T31 );
_LP5 = _LP5 + abs(1-P_T41);
_LP5 = _LP5 + abs(0-P_T51 );
_LP5 = _LP5 + abs(1-P_T61 );
_LP5 = _LP5 + abs(0-P_T71 );
_LP5 = _LP5 + abs(1-P_T81 );
_LP5 = _LP5 + abs(0-P_T91 );
_LP5 = _LP5 + abs(1-P_T101 );
_LP5 = _LP5 + abs(0-P_T111 );
_LP5 = _LP5 + abs(1-P_T121 );
_LP5 = _LP5 + abs(0-P_T131 );
_LP5 = _LP5 + abs(1-P_T141 );
_LP5 = _LP5 + abs(0-P_T151 );

_LP2=0.75*_LP2;

propensity=min(_LP1,_LP2,_LP3,_LP4,_LP5);

if propensity=_LP1 then pred_class='b.INV';
 else if propensity=_LP2 then pred_class='c.SL';
   else if propensity=_LP3 then pred_class='d.UL';
      else if propensity=_LP4 then pred_class='e.CARD';
 else if propensity=_LP5 then pred_class='NO_TAKE';

run;


/****  Calculating  the  difference  between  the  correctly
classified vs the incorrectly classified*************/


data smallest;
  set jon.BASE_fin2 ;
  firsts=smallest(1,_LP1,_LP2,_LP3,_LP4,_LP5);
  seconds=smallest(2,_LP1,_LP2,_LP3,_LP4,_LP5);
  thirds=smallest(3,_LP1,_LP2,_LP3,_LP4,_LP5);
  fourths=smallest(4,_LP1,_LP2,_LP3,_LP4,_LP5);

  length  pred_class2  $8  pred_class3  $8  pred_class4  $8
pred_class5 $8;

if seconds=_LP1 then pred_class2='b.INV';
 else if seconds=_LP2 then pred_class2='c.SL';
   else if seconds=_LP3 then pred_class2='d.UL';
```

```
      else if seconds=_LP4 then pred_class2='e.CARD';
 else if seconds=_LP5 then pred_class2='NO_TAKE';

if N_Product=pred_class or N_Product=pred_class2 then prd=1;
   else prd=0 ;
if N_Product=pred_class  then prd_0=1;
   else prd_0=0 ;
diff  =seconds-firsts;

 if diff < 0.25 then  pred_class3='';
   else pred_class3=pred_class;

 if diff < 0.5 then  pred_class4='';
   else pred_class4=pred_class;

 if diff < 0.75 then  pred_class5='';
   else pred_class5=pred_class;

 if N_Product=pred_class3  then prd3_0=1;
/*  else if pred_class3 = ' ' then prd2_0=*/
   else prd3_0=0 ;

 if N_Product=pred_class4  then prd4_0=1;
   else prd4_0=0 ;

 if N_Product=pred_class5  then prd5_0=1;
   else prd5_0=0 ;

  if pred_class3=''  then rej3_0=1;
   else rej3_0=0 ;

  if pred_class4=''  then rej4_0=1;
   else rej4_0=0 ;

  if pred_class5=''  then rej5_0=1;
   else rej5_0=0 ;



run;
data small;
set smallest;
if prd=0 then N_DIFF=diff;
if prd=1 then P_DIFF=diff;
if prd_0=0 then N_DIFF0=diff;
if prd_0=1 then P_DIFF1=diff;
run;
```

```
/****   Creating   the   class   separation   plots   in
pentiles*************/

%macro prof(dsn,field);
   /* sort dataset*/
      proc sort data=&dsn;
      by &field;
      run;
 proc univariate noprint data=&dsn;
        var &field;
        output out=tmp pctlpts = 0 5 10 15 20 25 30 35 40 45
50 55 60 65 70 75 80 85 90 95 100
        pctlpre =pct_;
 run;

      proc transpose data=tmp out=tmpb;
      run;



      proc sql;
      select col1
      into :pct_0-:pct_20
      from tmpb;
      quit;



  data &dsn._2;
  set &dsn;

   if &field le &pct_1 then decile=20;
        else if &field le &pct_2 then decile=19;
        else if &field le &pct_3 then decile=18;
        else if &field le &pct_4 then decile=17;
        else if &field le &pct_5 then decile=16;
        else if &field le &pct_6 then decile=15;
        else if &field le &pct_7 then decile=14;
        else if &field le &pct_8 then decile=13;
        else if &field le &pct_9 then decile=12;
        else if &field le &pct_10 then decile=11;
          else if &field le &pct_11 then decile=10;
          else if &field le &pct_12 then decile=9;
        else if &field le &pct_13 then decile=8;
        else if &field le &pct_14 then decile=7;
        else if &field le &pct_15 then decile=6;
        else if &field le &pct_16 then decile=5;
        else if &field le &pct_17 then decile=4;
        else if &field le &pct_18 then decile=3;
        else if &field le &pct_19 then decile=2;
        else if &field le &pct_20 then decile=1;
  run;

  data &dsn._2;
  set &dsn._2;
  where decile ne .;

  run;

%mend prof;
%prof(small,diff);



/************ using mean square error approach **********/

data base_test;
  set  jon.base_fin;

*** Compute Linear Predictor;
  _LP1 = 0;
  _LP2 = 0;
  _LP3 = 0;
  _LP4 = 0;
  _LP5 = 0;

  length pred_class $8;
  ***  Effect: 'b.INV' ;
_LP1 = _LP1 + (abs(1-P_T11 ))**2;
_LP1 = _LP1 + (abs(1-P_T21 ))**2;
_LP1 = _LP1 + (abs(1-P_T31 ))**2;
_LP1 = _LP1 + (abs(1-P_T41))**2;
_LP1 = _LP1 + (abs(1-P_T51 ))**2;
_LP1 = _LP1 + (abs(1-P_T61 ))**2;
_LP1 = _LP1 + (abs(1-P_T71 ))**2;
_LP1 = _LP1 + (abs(1-P_T81 ))**2;
_LP1 = _LP1 + (abs(1-P_T91 ))**2;
_LP1 = _LP1 + (abs(1-P_T101 ))**2;
_LP1 = _LP1 + (abs(1-P_T111 ))**2;
_LP1 = _LP1 + (abs(1-P_T121 ))**2;
_LP1 = _LP1 + (abs(1-P_T131 ))**2;
_LP1 = _LP1 + (abs(1-P_T141 ))**2;
_LP1 = _LP1 + (abs(1-P_T151 ))**2;

  ***  Effect: 'c.SL' ;

_LP2 = _LP2 + (abs(0-P_T11 ))**2;
_LP2 = _LP2 + (abs(0-P_T21 ))**2;
```

```
_LP2 = _LP2 + (abs(0-P_T31 ))**2;
_LP2 = _LP2 + (abs(0-P_T41))**2;
_LP2 = _LP2 + (abs(0-P_T51 ))**2;
_LP2 = _LP2 + (abs(0-P_T61 ))**2;
_LP2 = _LP2 + (abs(0-P_T71 ))**2;
_LP2 = _LP2 + (abs(0-P_T81 ))**2;
_LP2 = _LP2 + (abs(1-P_T91 ))**2;
_LP2 = _LP2 + (abs(1-P_T101 ))**2;
_LP2 = _LP2 + (abs(1-P_T111 ))**2;
_LP2 = _LP2 + (abs(1-P_T121 ))**2;
_LP2 = _LP2 + (abs(1-P_T131 ))**2;
_LP2 = _LP2 + (abs(1-P_T141 ))**2;
_LP2 = _LP2 + (abs(1-P_T151 ))**2;

  ***  Effect: 'd.UL' ;

_LP3 = _LP3 + (abs(0-P_T11 ))**2;
_LP3 = _LP3 + (abs(0-P_T21 ))**2;
_LP3 = _LP3 + (abs(0-P_T31 ))**2;
_LP3 = _LP3 + (abs(0-P_T41))**2;
_LP3 = _LP3 + (abs(1-P_T51 ))**2;
_LP3 = _LP3 + (abs(1-P_T61 ))**2;
_LP3 = _LP3 + (abs(1-P_T71 ))**2;
_LP3 = _LP3 + (abs(1-P_T81 ))**2;
_LP3 = _LP3 + (abs(0-P_T91 ))**2;
_LP3 = _LP3 + (abs(0-P_T101 ))**2;
_LP3 = _LP3 + (abs(0-P_T111 ))**2;
_LP3 = _LP3 + (abs(0-P_T121 ))**2;
_LP3 = _LP3 + (abs(1-P_T131 ))**2;
_LP3 = _LP3 + (abs(1-P_T141 ))**2;
_LP3 = _LP3 + (abs(1-P_T151 ))**2;

  ***  Effect: 'e.CARD' ;

_LP4 = _LP4 + (abs(0-P_T11 ))**2;
_LP4 = _LP4 + (abs(0-P_T21 ))**2;
_LP4 = _LP4 + (abs(1-P_T31 ))**2;
_LP4 = _LP4 + (abs(1-P_T41))**2;
_LP4 = _LP4 + (abs(0-P_T51 ))**2;
_LP4 = _LP4 + (abs(0-P_T61 ))**2;
_LP4 = _LP4 + (abs(1-P_T71 ))**2;
_LP4 = _LP4 + (abs(1-P_T81 ))**2;
_LP4 = _LP4 + (abs(0-P_T91 ))**2;
_LP4 = _LP4 + (abs(0-P_T101 ))**2;
_LP4 = _LP4 + (abs(1-P_T111 ))**2;
_LP4 = _LP4 + (abs(1-P_T121 ))**2;
_LP4 = _LP4 + (abs(0-P_T131 ))**2;
_LP4 = _LP4 + (abs(0-P_T141 ))**2;
_LP4 = _LP4 + (abs(1-P_T151 ))**2;
```

```
  ***  Effect: 'NO_TAKE' ;

_LP5 = _LP5 + (abs(0-P_T11 ))**2;
_LP5 = _LP5 + (abs(1-P_T21 ))**2;
_LP5 = _LP5 + (abs(0-P_T31 ))**2;
_LP5 = _LP5 + (abs(1-P_T41))**2;
_LP5 = _LP5 + (abs(0-P_T51 ))**2;
_LP5 = _LP5 + (abs(1-P_T61 ))**2;
_LP5 = _LP5 + (abs(0-P_T71 ))**2;
_LP5 = _LP5 + (abs(1-P_T81 ))**2;
_LP5 = _LP5 + (abs(0-P_T91 ))**2;
_LP5 = _LP5 + (abs(1-P_T101 ))**2;
_LP5 = _LP5 + (abs(0-P_T111 ))**2;
_LP5 = _LP5 + (abs(1-P_T121 ))**2;
_LP5 = _LP5 + (abs(0-P_T131 ))**2;
_LP5 = _LP5 + (abs(1-P_T141 ))**2;
_LP5 = _LP5 + (abs(0-P_T151 ))**2;

propensity=min(_LP1,_LP2,_LP3,_LP4,_LP5);

if propensity=_LP1 then pred_class='b.INV';
 else if propensity=_LP2 then pred_class='c.SL';
   else if propensity=_LP3 then pred_class='d.UL';
      else if propensity=_LP4 then pred_class='e.CARD';
 else if propensity=_LP5 then pred_class='NO_TAKE';

run;

data smallest;
  set base_test ;
  firsts=smallest(1,_LP1,_LP2,_LP3,_LP4,_LP5);
  seconds=smallest(2,_LP1,_LP2,_LP3,_LP4,_LP5);
  thirds=smallest(3,_LP1,_LP2,_LP3,_LP4,_LP5);
  fourths=smallest(4,_LP1,_LP2,_LP3,_LP4,_LP5);

  length pred_class2 $8;

if seconds=_LP1 then pred_class2='b.INV';
 else if seconds=_LP2 then pred_class2='c.SL';
   else if seconds=_LP3 then pred_class2='d.UL';
      else if seconds=_LP4 then pred_class2='e.CARD';
 else if seconds=_LP5 then pred_class2='NO_TAKE';

if N_Product=pred_class or N_Product=pred_class2 then prd=1;
   else prd=0 ;
if N_Product=pred_class  then prd_0=1;
   else prd_0=0 ;
diff  =seconds-firsts;
```

```
run;
endrsubmit;


data small;
set smallest;
if prd=0 then N_DIFF=diff;
if prd=1 then P_DIFF=diff;
if prd_0=0 then N_DIFF0=diff;
if prd_0=1 then P_DIFF1=diff;
run;


/****Kernel density plots *********/

ODS GRAPHICS ON ;
proc kde data=sasswork.small;
univar P_DIFF N_DIFF / plots=DensityOverlay out=KerOut GridL=0
GridU=5;
run;
proc kde data=sasswork.small;
univar  P_DIFF1  N_DIFF0  /  plots=DensityOverlay  out=KerOut1
GridL=0 GridU=5;
run;
```