CUSTOMER RETENTION

André Fourie (1270563)

MECN7018

Supervisor: Dr. Joke Bührmann

School of Engineering and the Built Environment
University of the Witwatersrand
Johannesburg, South Africa

A research report submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, May 2018

Candidate's declaration

I declare that this research report is my own unaided work. It is being submitted for the Degree of Master of Science to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other university.

……

…….. day of ……., ……

Abstract

The aim of this study is to model the probability of a customer to attrite/defect from a bank where, for example, the bank is not their preferred/primary bank for salary deposits. The termination of deposit inflow serves as the outcome parameter and the random forest modelling technique was used to predict the outcome, in which new data sources (transactional data) were explored to add predictive power. The conventional logistic regression modelling technique was used to benchmark the random forest's results.

It was found that the random forest model slightly overfit during the training process and loses predictive power during validation and out of training period data. The random forest model, however, remains predictive and performs better than logistic regression at a cut-off probability of 20%.

Acknowledgements

First and foremost, I would like to sincerely thank my advisor Dr. Joke Bührmann for the help, guidance, freedom and great support she provided me.

I am also very grateful to my mother, who encouraged me a lot and supported me financially.

I want to extend my thanks to Jacques Venter for his guidance and assistance.

I would also like to take this opportunity to thank my girlfriend, Marisna Herbst, who provided me with tremendous support, love and encouragement.

Contents

List of Figures

List of Tables

Nomenclature/list of acronyms

ACC	Overall accuracy
AUC	Area under the ROC curve
BRS	Behavioural risk scoring
CART	Classification and regression tree
CIS	Customer information system
CLV	Customer lifetime value
CNN	Condensed nearest neighbour
CRM	Customer relationship management
ENN	Edited nearest neighbour
FN	False negative
FP	False positive
IV	Information value
kNN	k-nearest neighbour
M	Mean absolute deviation
NCL	Neighbourhood cleaning
OOB	Out of bag
PSI	Population stability index
RBF	Radial basis function
ROC	Receiver operating characteristic
SMOTE	Synthetic minority oversampling technique
SVM	Support vector machine
TDS1	Test dataset 1
TDS2	Test dataset 2
TN	True negative
TNR	True negative rate
TP	True positive
TPR	True positive rate
WoE	Weights of evidence

1. Introduction

Banking in general has been revolutionised in recent years. Customer retention needs to be one of the top priorities, as customers are the primary revenue source of most banks. We are currently in the information age and customers are becoming ever more skilful in the use of computers and other technologies. This is forcing banks to be more innovative in their operational strategies and to be competitive and relevant in today's market. Part of this competitiveness entails acquiring new customers as well as retaining existing customers by way of providing better services and better quality of service. Thus, customer relationship management (CRM) has become a key focus within banks. Due to increasing competitiveness, banks have realised that they need to be competent in the CRM space and if done correctly, CRM can be a valuable tool to achieve improved results.

1.1. Research Background

In the past, bank marketing was geared towards selling products to new customers. Recently, a paradigm shift led to banks focusing more on retaining their existing customers and selling products to them. It is widely known that it is more expensive to sign up new customers, than to retain existing ones (Pfeifer, 2004). In order to achieve this paradigm shift, conventional advertisement would not suffice and a new approach needs to be formalised by means of statistical modelling.

1.2. Objective

The aim of this study is to model the probability of a customer to defect from a bank, i.e. not using the bank as their primary bank, whereby a customer's primary bank is defined as the bank into which the customer deposits their salary. The termination of deposit inflow served as the outcome parameter. The random forest modelling technique was used to predict whether a customer is likely to defect and alternative data sources such as transactional data were explored to add predictive power. The logistic regression modelling technique, discussed in Chapter 6.1, was set as a benchmark to compare the performance of the random forest. The findings are summarised and discussed in Chapter 11.

1.3. Research questions

Is it possible to use customer data to predict customer attrition?

Is the random forest modelling technique a viable technique to predict customer attrition and how does it compare to the conventional logistic regression modelling technique?

1.4. Chapter prelude

This study focuses on classifying attrition cases and as a result, classification (Chapter 2) is discussed in the literature review. It is important to shed light on single classification (Chapter 3) as a precursor to ensemble classifiers (Chapter 4).

Customer attrition is considered a rare event and consists of only a small portion of the entire customer base. This causes imbalance between the cases in the dataset, motivating the analysis on how to handle such data (Chapter 5).

Sampling was done in the study, due to the imbalance between attrition and non-attrition and the large number of records. This causes the modelling tool to use a lot of computational power as explained in Chapter 5.2.1.

Measuring model performance is explained in Chapter 5.1.1 and Chapter 6. This is necessary to determine whether the model that was built is predictive and informative. Customer lifetime value can be defined as the present value of the customer's future predicted cash flows, resulting from the customer's relationship with the bank (Pfeifer, 2005). This, however, is dependent on the length of the customer's relationship with the bank. The impact and importance of customer lifetime value are discussed in Chapter 7.

It is important to acquire as well as retain customers in order to maximise future profits generated from customers which is discussed in Chapter 8. Chapter 9 details the methodology followed and the results of which are provided in Chapter 10. The research report and findings are concluded and summarised in Chapter 11.

2. Classification

Machine learning is enabled by statistical learning, which originates from the fields of statistical and functional analysis (Mohri *et al.*, 2012). Science and finance focus a great deal of attention on machine learning algorithms, as it plays a key role in their fields (Friedman *et al.*, 2001). There are three main categories of machine learning: unsupervised learning, supervised learning and reinforcement learning. Supervised learning is the focus of this study.

Predicting is one of the main goals of learning. The supervised learning algorithm is guided to predict the target variable. The target variable is then divided into (1) classification for discrete variable and (2) regression for continuous target variable (Mohri *et al.*, 2012). In this study a discrete variable is assumed. The outcome will be predicted as either defection or non-defection.

Classification is used to determine to which set of categories (sub-populations) a new observation belongs. The learning function maps the relationship between the input observations and the corresponding output categories. The predefined goal function of the predicted target value is optimised by the output model. This is compared to the true value of the target variable to find the error rate, by utilising the training set of data that contains the observations with the known category membership. Datasets with nominal or binary categories are most suitable to predict and describe by a classifier. It is less effective for ordinal categories, for example predicting tomorrow's weather to be cloudy, sunny or rainy, because they do not consider the implicit order among the categories (Frank *et al.*, 2001).

Various algorithms and tools can be used for classification. The algorithms can be divided into two sub-groups, depending on whether assumptions are made about the dataset, i.e. parametric and nonparametric classification algorithms. Due to the complexity of the problem, there is no superior algorithm that always performs the best (Frank *et al.*, 2001).

Gaussian and binomial distributions are parametric methods that are assumed in logistic regression and linear discriminant analysis. In modern parametric techniques like the Naïve Bayesian method, conditional independence assumptions are made on the attribute variables. In contrast, no such assumptions are made on nonparametric methods as the decision boundaries could be of any arbitrary geometry (Hubert *et al.*, 2001). Nearest neighbour-based algorithms

belong to this category. Other algorithms that also fall into this group are: decision tree algorithms, neural network algorithms and support vector machines.

Classification algorithms fall into the following groups in terms of their structure: single classifiers and ensemble classifiers. The single classifier is a standalone classification algorithm and an ensemble classifier is a combination of single classifiers (Frank *et al.*, 2001). Ensemble classifiers can be described as a higher-level classifier combination strategy and not a classification algorithm, with the goal of improving the ensemble classification performance by properly combining the single classifiers. It is important to understand single classifiers to be able to understand ensemble classifiers. The main focus of this study is random forests, which is an ensemble classifier. Single classifiers are discussed in Chapter 3 and ensemble classifiers in Chapter 4.

3. Single classifier

The random forest modelling technique is an ensemble of the single classifier; decision trees. The following three single classifiers are discussed in order to provide introductory knowledge to ensemble classifiers. The single classifiers k-nearest neighbour (kNN), decision trees and support vector machines are discussed below.

3.1. K-nearest neighbour classification

The nearest neighbour problem, also known as the *closest pair of points* problem, has been studied extensively in the field of computational geometry (Shamos *et al.*, 1975). The k-nearest neighbour algorithm is a very intuitive method that classifies unlabelled instances based on their similarity to the training set. Simply put, for an unlabelled example $X^* \in \Re^p$, find the k closest labelled examples in the training set and assign X^* to the class that appears most frequently within the k closest neighbours. A Bayesian prior assigns weights to the classification, based on the relative number of samples, for a potentially better classification.

kNN density estimation is closely related to the kNN classifier. Observe a dataset of N samples, of which N_i is from class ω_i. In order to predict the label of an unknown sample X^*, a hyper-sphere of volume V is drawn around X^*. It is assumed that the volume contains a total of k examples, with k_i from class ω_i.

The likelihood functions using kNN probability density (Cheng et al., 2013) could be approximated by

$$p(X|\omega_i) \cong \frac{k_i}{N_i V} \quad , \tag{1}$$

with $p(X|\omega_i)$ the probability density.

Similarly, the unconditional density is estimated by

$$p(X) \cong \frac{k}{\mathrm{N}V} \quad . \tag{2}$$

The priors are approximated by

$$p(\omega_i) \cong \frac{N_i}{\mathrm{N}} \quad . \tag{3}$$

Using Bayes theorem, the posterior probability membership is obtained:

$$p(\omega_i|X) = \frac{p(X|\omega_i)}{p(X)} \tag{4}$$

$$= \frac{\frac{k_i}{N_i V} \cdot \frac{N_i}{\mathrm{N}}}{\frac{k}{\mathrm{N}V}} \tag{5}$$

$$= \frac{k_i}{k}, \tag{6}$$

with $p(\omega_i|X)$ the posterior probability.

A test point X is assigned to the class having the largest posterior probability, corresponding to the largest value of Eq. (6), in order to minimise the probability of misclassification. The majority class in the k nearest points is assigned to the test point. An example of this classification can be seen in Figure 1. The yellow squares and purple circles in Figure 1 in the 2-dimensional space belong to Class A and B. The newly input points label, the red star, is classified based on the nearest 3 or 6 neighbours, depending on the chosen k. The dotted circles (k=3) indicate the nearest 3 neighbours of the new input point, which includes 2 points of Class B and 1 of Class A.

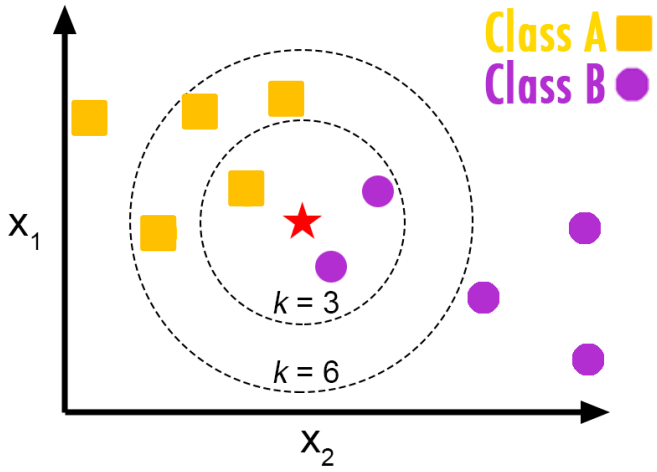


Figure 1: Illustration of the nearest neighbour classification method

As a result, if k=3, this new point is classified as Class B because the majority of the points are from Class B, while if k=6, the point would be classified as Class A (Shamos *et al.*, 1975).

The "nearest" observations can be obtained by using the Euclidean or Mahalanobis distance formula. In contrast to Euclidean distance, the Mahalanobis distance considers the correlations of the dataset and is scale-invariant.

The Euclidean distance between the points x and $y \in \mathbb{R}^p$ is given by (Deza *et al.*, 2009):

$$d(x,y) = d(y,x) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots + (y_n - x_n)^2} \tag{7}$$

$$= \sum_{i=i}^{n} (y_i - x_i)^2 \,. \tag{8}$$

The Mahalanobis distance, $D_m(\vec{x})$, of an observation $\vec{x} = \left(x_{1,} x_{2,} \ldots, x_N\right)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, \mu_2, \ldots, \mu_N)^T$ and covariance matrix S is defined as (De Maesschalck *et al.*, 2000):

$$D_m(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})} \,. \tag{9}$$

The kNN function is only approximated locally and all computations are deferred until classification, thus the kNN is considered a lazy learning algorithm.

Random forest and logistic regression modelling techniques are considered eager learning algorithms, since they compile data into a compressed model and classify incoming patterns by means of the induced model. These techniques require more computational costs in the developmental phase, but less storage capacity for data points. Eager learning algorithms use less computational power to recall in the testing phase than lazy algorithms, implying lower costs associated to them (Wettschereck *et al.*, 1997).

3.2. Decision tree classifier

The decision tree algorithm is a widely-used method for data mining. The aim of the model is to predict the value of a target variable based on a set of input variables. There are two types of decision trees:

1. The classification tree that predicts discrete outcomes; and
2. the regression tree that predicts continuous outcomes.

The term classification and regression tree (CART) analysis is an umbrella term used to refer to both procedures (Breiman, 1996). This study focuses on the classification tree only.

The tree-like structure of a decision tree consists of three parts:

1. Internal (non-leaf) node;
2. branch; and
3. terminal (leaf) node.

Decision tree learning is the process of constructing a decision tree from class-labelled training tuples. Each internal node denotes a test of an attribute or feature. Each branch represents the outcome of the test and each leaf node holds a class label (Breiman, 1996).

It is important to understand the node splitting criteria before analysing the algorithm. By optimising the cost function for each node, the feature and the corresponding threshold of the feature are identified. During the testing phase, the observations are classified as either the right child node or the left child node, depending on the value of the feature. A value larger than the threshold belongs to the right child node and a value smaller than the threshold to the left child node (Breiman, 1996).

Gini impurity, $I_G(f)$ and Entropy,$I_E(f)$ are traditionally employed to select the "best splitting" feature and corresponding threshold. The definitions are as follows (Witten *et al.*, 2011):

$$I_G(f) = -\sum_{i=1}^{m} f_i(1 - f_i) = 1 - \sum_{i=1}^{m} f_i^2, \tag{10}$$

for a set of items with m classes, $i \in \{1,2,\dots,m\}$ and f_i the fraction of items labelled with class i in the set, and

$$I_E(f) = -\sum_{i=1}^{m} f_i \log_2 f_i \quad , and \tag{11}$$

$$\sum_{i=1}^{m} f_i = 1\,. \tag{12}$$

Eq. (12) represents the percentage of each class present in the child node that results from a split in the tree.

Figure 2 illustrates the plots for binary classification. Gini impurity and Entropy provides measures of the homogeneity of the target variable. Both Gini impurity and Entropy will approach 0 where f approaches 0 or 1. This indicates that observations are homogeneous and tend to be from the same class. The feature and threshold that will generate the largest drop of these values between parent and child nodes are selected to split this node. The aim is to maximise the difference and is called the information gain (Breiman, 1996).

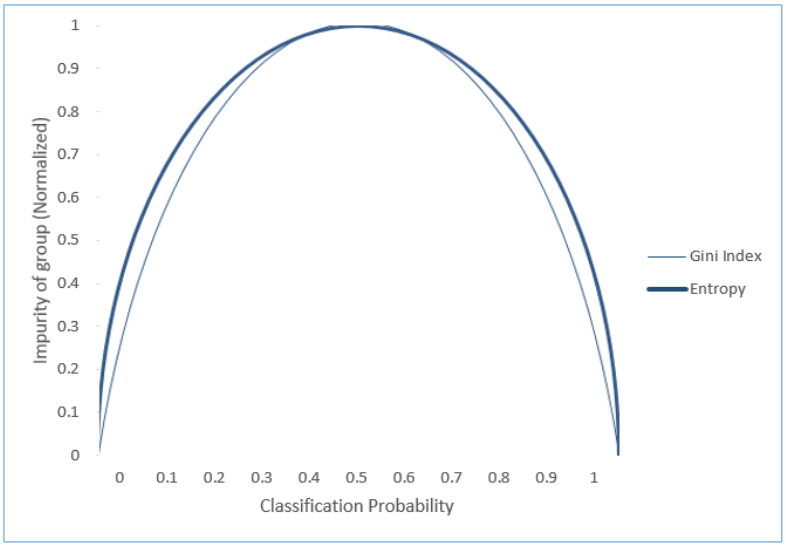


Figure 2: The Gini impurity and Entropy curves illustrate their respective probability distributions (Witten *et al.*, 2011)

Figure 3 below provides the pseudo code for the splitting attribute threshold in Algorithm 1 and for the decision tree in Algorithm 2:

Algorithm 1 splitting_attribute_threshold

Inputs(s): The matrix of training examples $\boldsymbol{X}$ and the corresponding label vector $\boldsymbol{\omega}$
Output(s): Selected attribute $\mathcal{A}$ and its corresponding splitting threshold θ
Require: Homogeneity measure H

set $\theta \leftarrow -\infty$
set $\mathcal{A} = \mathcal{A}_1$
set $max\_gain \leftarrow 0$
for each attribute $\mathcal{A}_i \in X$ **do**
    for each possible threshold $\theta_j^{\mathcal{A}_i} \in \mathcal{A}_i$ **do**
        set $temp\_gain \leftarrow IG(root_{node}, left_{node}, right_{node}, \boldsymbol{\omega})$
        if $temp\_gain >$ max_gain **then**
            set $\theta \leftarrow \theta_j^{\mathcal{A}_i}$
            set $\mathcal{A} \leftarrow \mathcal{A}_i$
            set $max\_gain \leftarrow temp\_gain$
        end if
    end for
end for
return θ and $\mathcal{A}$
end

Algorithm 2 decision_tree

Input(s): The matrix of training examples $\boldsymbol{X}$ and the corresponding label vector $\boldsymbol{\omega}$
Output(s): Decision tree T

if $\boldsymbol{X} == \boldsymbol{\phi}$ **then**
    return a single node with $\boldsymbol{\phi}$
end if
if $\boldsymbol{\omega}$ consists records all with the same value for the class label **then**
    return a single leaf node with that value
end if
set $(\theta, \mathcal{A}) \leftarrow$ splitting_attribute_threshold$(\boldsymbol{X}, \boldsymbol{\omega})$
set $(\boldsymbol{X}_{left}, \boldsymbol{\omega}_{left})$ and $(\boldsymbol{X}_{right}, \boldsymbol{\omega}_{right})$ as the subsets of $(\boldsymbol{X}, \boldsymbol{\omega})$ consisting of observations respectively with value greater than or equal to and less than θ for attribute $\mathcal{A}$
recursively apply decision_tree to subset $(\boldsymbol{X}_{left}, \boldsymbol{\omega}_{left})$ and $(\boldsymbol{X}_{right}, \boldsymbol{\omega}_{right})$ until they are empty or the stopping criteria are met
return a tree with root or node labelled $(\theta, \mathcal{A})$ and child node decision_tree$(\boldsymbol{X}_{left}, \boldsymbol{\omega}_{left})$ and decision_tree$(\boldsymbol{X}_{right}, \boldsymbol{\omega}_{right})$
end

Figure 3: Pseudo code for the splitting attribute threshold and decision tree (Barros *et al.*, 2015)

Algorithms 1 and 2 are the basic components used to build a decision tree. A variety of decision tree structures have been introduced based on these algorithms. Hunt's algorithm (Hunt, 1977) is one of the earliest decision tree algorithms, presented in 1966. Quinlan (1986) created the ID3 (Iterative Dichotomiser 3) algorithm, which is the precursor to the C4.5 algorithm (Quinlan,1993). Several improvements were made by Quinlan (1993) to formulate the C4.5 algorithm, which can handle both discrete and continuous attributes and missing values in the training dataset. The Entropy method serves to calculate the gain for both ID3 and C4.5. The CART method of Breiman (1996) is another popular analysis method that uses Gini impurity to measure homogeneity.

Minimal cost-complexity pruning prevents overfitting and forms part of many methods, including the CART method (Mansour, 1997). The purpose of this step is to build a right-sized tree by estimating the true misclassification cost. The CART method firstly builds a fully-grown tree and then cuts the pair of leaves sequentially. The value of the cost-complexity and misclassification cost are calculated using ten-fold cross-validation for each sub-tree. The final optimal tree is then selected based on the final values produced by the CART algorithm.

The decision tree is a popular classification method due to several advantages that it has over other classification methods, namely (Quinlan, 1987):

1) It is simple to understand and interpret.
2) Data preparation is minimal. Other techniques often require the data to be normalised, the creation of dummy variables and the removal of blank values.
3) It uses a white box model. The classification model is clear and explicit. It can be seen how the variables are associated with the result of the tree structure.
4) It is a robust, non-parametric classifier. The decision tree method is not only a single classifier, but is used as the base for numerous ensemble classifiers, for example, the random forest classification method.

The decision tree also has several disadvantages (Bright Hub Project Management, 2011):

1) The decision tree is not a adequate method for regression and does not predict continuous values.

2) Spurious relationships can occur.
3) Functions such as exponential size or parity are difficult to represent.
4) The same sub-tree on different paths can be duplicated.

3.3. Support vector machine

The support vector machine (SVM) was first introduced by Vladimir Vapnik in 1995 (Cortes *et al.*, 1995), about which Burges (1998) provides a detailed introduction. The following is a brief introduction to the mathematical formulation of linearly separable setting:

Give training dataset $\mathcal{D}$, a set of *n* points of the form

$$\mathcal{D} = \{(x_i, \omega_i) | x_i \in \mathbb{R}^p, \omega_i \in \{-1, 1\}\}, i = 1, \dots, n \ , \tag{13}$$

where ω_i is either 1 or -1, indicating the label of observation x_i. Each x_i is a p-dimensional or multi-dimensional feature vector.

The objective is to find a hyperplane that maximises the margin between the points, having ω_i = 1 and ω_i = −1, where any hyperplane can be written as the set of points $\boldsymbol{x}$ satisfying:

$$\boldsymbol{w} \cdot \boldsymbol{x} - b = 0 \ , \tag{14}$$

with b the distance of one from the hyperplane to the closest points in each class.

The $\cdot$ denotes the dot product and $\boldsymbol{w}$ the normal vector to the hyperplane. The term $\frac{b}{\|w\|}$ calculates the distance from the hyperplane to the origin.

Two support vectors are selected in such a way that the data is separated, with no points falling in between. The distance between the support vectors needs to be maximised, in order to separate the data in a linear fashion. The margin is defined as the region bound by the two support vectors.

The equations below describe the support vectors as follows (Burges, 1998):

$$\boldsymbol{w} \cdot \boldsymbol{x} - b = 1 \text{ and } \boldsymbol{w} \cdot \boldsymbol{x} - b = -1 \ . \tag{15}$$

It can be seen in Figure 4 that the distance between the two support vectors is therefore $\frac{2}{\|w\|}$, thus $\|w\|$ needs to be minimised.

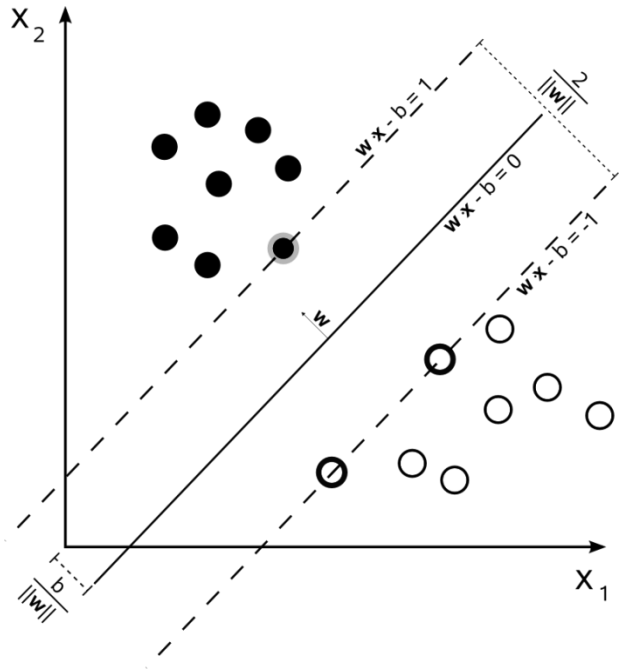


Figure 4: Illustration of SVM classification boundary (Burges, 1998)

In Figure 4 the filled (class label = "1") and unfilled (class label = "-1") circle points represent training points belonging to different classes. The two dashed lines indicate the boundary of maximum margin, while the solid line indicates the classification boundary.

To prevent data points from falling into the margin, the following constraints are added:

$$\boldsymbol{w} \cdot \boldsymbol{x_i} - b \geq 1 \text{ for } \boldsymbol{x_i} \text{ having label "1" and} \quad (16)$$

$$\boldsymbol{w} \cdot \boldsymbol{x_i} - b \leq -1 \text{ for } \boldsymbol{x_i} \text{ having label "-1".} \quad (17)$$

The above constraints could be further reduced to:

$$\omega_i(\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1 \text{ for all } 1 \leq i \leq n, \quad (18)$$

where n is the set of points.

To summarise, the optimisation problem Eq. (18) becomes:

$$\arg \min_{\boldsymbol{w},b} \max_{\alpha \geq 0} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=0}^{n} \alpha_i [\omega_i (\boldsymbol{w}.\boldsymbol{x_i} - b) - 1] \right\}. \quad (19)$$

where $\alpha_i, for\ all\ i = 1, \dots, n$ are positive Lagrange multipliers. (Note that minimising $\|w\|$ mathematically equals minimising $\frac{1}{2}\|w\|^2$.)

By using standard quadratic programming techniques, the above equation can be solved by means of the Karush-Kuhn-Tucker condition (Fletcher, 1987).

Slack variable, ξ_i, is introduced to the objective function for linear inseparable problems and it becomes:

$$\arg \min_{\boldsymbol{w},\xi,b} \max_{\alpha \geq 0, \beta \geq 0} \left\{ \frac{1}{2} \|w\|^2 + \boldsymbol{C} \sum_{i=0}^{n} \xi_i - \sum_{i=0}^{n} \alpha_i [\omega_i (\boldsymbol{w}.\boldsymbol{x_i} - b) - 1 + \xi_i] - \sum_{i=0}^{n} \boldsymbol{\beta_i} \xi_i \right\}, \quad (20)$$

where $\boldsymbol{C}$ is a vector of constraint functions and $\boldsymbol{\beta}_i \in \mathbb{R}^p$. The tolerance of the misclassification of the model is measured by the slack variable ξ_i.

If a problem is not linearly separable, the Kernel trick (Aizerman, 1964) can be applied in order to solve it. Here the input data is mapped to a higher dimension or infinite dimension feature space. The Gaussian radial basis function kernel (RBF), linear kernel and polynomial kernel are among the popular kernel functions used.

SVMs are considered the benchmark in many classification comparison papers, due to their robustness and performance (Cortes *et al.*, 1995). This make them the status quo in classification methods. SVMs are also widely applied in the science field (Cortes *et al.*, 1995).

In the next chapter, ensemble classifiers are reviewed. These are combinations of single classifiers (Friedman *et al.*, 2001).

4. Ensemble classifier

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms (Opitz, 1999; Polikar, 2006). A combination of the output of many weak classifiers produces a powerful predicting committee (Friedman *et al.*, 2001).

An accurate classifier is one that can predict an outcome better than taking a random guess or flipping a coin (Hansen *et al.*, 1990). Classifiers are deemed diverse when two classifiers make different errors on new data points. In order to elaborate on diversity, consider three classifiers $\{h_1, h_2, h_3\}$ and a new case $\mathbf{x}$. The values $h_2(\mathbf{x})$ and $h_3(\mathbf{x})$ will be wrong in the case where the three classifiers are identical and $h_1(\mathbf{x})$ is wrong. In contrast, if the classifiers are uncorrelated and $h_1(\mathbf{x})$ is wrong, $h_2(\mathbf{x})$ and $h_3(\mathbf{x})$ might be correct, thus the majority vote will be correct. The probability of a majority vote being wrong is illustrated in the area under the binomial distribution where more than $L/2$ hypothesis is wrong, i.e. in the case where the error rates of L hypothesis h_l are all equal to $p < 0.5$ and the errors are independent (Dietterich, 2000).

Dietterich (2000) gives an example that shows how ensemble structures work. It features an ensemble of $n = 21$ binary classifiers, each with an error rate of $\epsilon = 0.3$. The ensemble classifier predicts the class label of a test example by taking a majority vote on the predictions made by the base classifier. The resulting ensemble classifier's error rate (P_e), shown in Eq. (21), will be 0.3 if the base classifiers are identical. The ensemble makes a wrong decision if more than half of the base classifiers predict an incorrect value. This can occur when the base classifiers are uncorrelated or independent. The error rate in this case can be calculated by means of the equation below (Dietterich, 2000):

$$P_e = \sum_{i=n/2}^{n} \binom{n}{i} \epsilon^i (1-\epsilon)^{n-i} \tag{21}$$

$$= \sum_{i=11}^{21} \binom{21}{i} \epsilon^i (1-\epsilon)^{21-i} \tag{22}$$

$$= 0.0264 \tag{23}$$

In this example the ensemble error rate is 0.0264. The plot for the error rate vs. ensemble error rate is depicted in Figure 5a and Figure 5b.

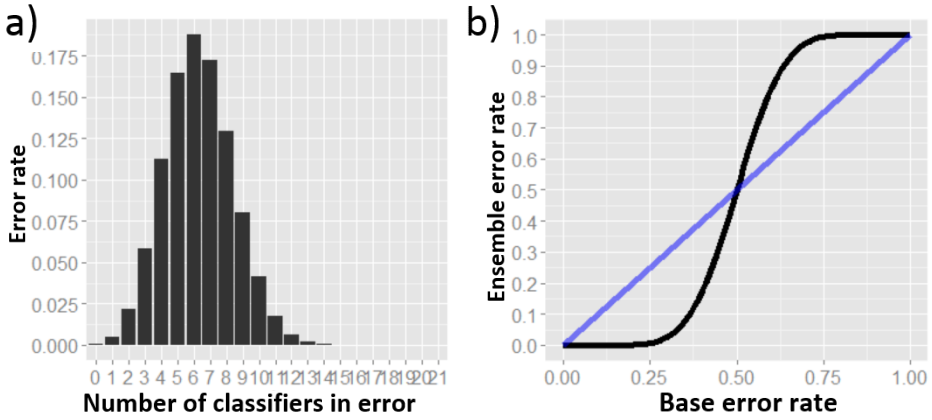


Figure 5: An illustration of the error rate (a) vs. ensemble classifier error rate (b) (Dietterich, 2000)

Figure 5a indicates how the error rate changes with the number of base classifiers, which makes correct predictions when the error rate of base classifier is 0.3. Figure 5b indicates how the ensemble error rate changes with the base classifier's error rate when the majority of the base classifiers make correct predictions (Dietterich, 2000).

The ensemble prediction errors will change, with the base classifier error rate changing. The blue curve (straight line) in Figure 5b shows when all the base classifiers are identical and the black curve (curved line) indicates when all the base classifiers are uncorrelated. The ensemble classifier is weaker than the base classifier if the classifier has an error rate greater than 0.5.

The example above illustrates two conditions that an ensemble classifier needs to meet to perform better than the base classifier:

1. The correlation between the base classifiers has to be low. A thorough explanation of this can be found in papers by Williams (1975), Beriman (2001) and Ahn *et al*. (2007).
2. The base classifier should perform better than a classifier guessing randomly, i.e. the classifier should predict better than flipping a coin and the error rate should be less than 0.5 (Hansen *et al.*, 1990).

Duinand Tax (2000) suggests that combining common classifiers can be divided into the following three groups:

1) Parallel combining of classifiers computed for different feature sets.
2) Stacked combining of different classifiers computed for the same feature space.
3) Combining weak classifiers, in which case large sets of simple classifiers are trained on a modified version of the original dataset.

The third point is the focus in this study.

4.1. Random forests

Bagging, also known as bootstrap aggregating, allows each decision tree in the ensemble to vote with equal weight for the most popular classifier (Breiman, 1996). This results in an improvement in classification accuracy.

Ensemble modelling is the process of combining the predictions of various sub-models into a single prediction output. These ensembles are grown by random vectors that guide the growth of each tree. Bagging is an example of how each tree is grown by a random selection from the examples in the training set (Breiman, 1996). Random split selection is another example of each node being split randomly from the *K* best possible splits (Dietterich, 2000). This falls outside the primary focus of this study.

Breiman (1996) explains tree bagging, the precursor to bagging, as follows:

Consider a training set $(\boldsymbol{x}_i, \boldsymbol{\omega_i})$ for i = 1,2, …, N, where $\boldsymbol{x}_i$ is a p - dimensional vector and $\boldsymbol{\omega}_i$ indicates the target label of $\boldsymbol{x}_i$. Tree bagging repeatedly selects a bootstrap sample of the training set, which consists of N samples, and then fits B number of trees to these samples. A bootstrap sample is a smaller sample that is "bootstrapped" from a larger sample. Bootstrapping is a type of resampling where large numbers of smaller samples of the same size are repeatedly drawn, with replacement, from a single original sample (Statistics How To, 2016). Figure 6 shows the algorithm for tree bagging:

Algorithm 3 tree_bagging

Input(s): Training set (x_i, ω_i) for $i = 1,2, \dots, N$; number of trees B
Output: Tree classifier $T_1, T_2, \dots, T_B$
1. **for** $b = 1$ **to** B **do**
2. Build a dataset S_b, by sampling N items randomly with replacement from the original data pool (x_i, ω_i) for $i = 1,2, \dots, N$.
3. Train decision tree T_b using S_b and save it.
4. **end for**

Figure 6: The tree bagging algorithm (Breiman, 1996)

These B decision trees will vote for the most popular classification, which will be the final classification result for any new test points (Breiman, 1996).

It was Breiman (2001) who first introduced the random forest algorithm. The only difference between the tree bagging and the random forest algorithm is that the random forest algorithm considers a random subset of the features to be selected for splitting the node in the tree building process. Tree bagging searches for the best way of splitting the feature set, considering the whole feature set. Correlation between trees in an ordinary bootstrap sample is reduced through the random forest method. If one or more feature sets have strong predictive power for the target variable using tree bagging, they will be selected in many of the decision trees causing high correlation and reducing the power of the ensemble.

Figure 7 provides the pseudo code of the random forest algorithm.

Algorithm 4 random_forest

Input(s): Training set (x_i, ω_i) for $i = 1,2, \dots, N$; number of trees B; number of features F
Output: Tree classifier $T_1, T_2, \dots, T_B$
1. **for** $b = 1$ **to** B **do**
2. Build a dataset S_b, by sampling N items randomly with replacement from the original data pool (x_i, ω_i) for $i = 1,2, \dots, N$
3. Train decision tree T_b without pruning using S_b. In each node splitting process, randomly select F features without replacement from the whole feature set and search the best splitting threshold inside the selected feature subset
4. **end for**

Figure 7: The random forest algorithm (Breiman, 2001)

New testing point classification is based on the majority vote of the *B* decision trees, similar to tree bagging. The number of trees *B* that need to be built should be specified beforehand, for both tree bagging and random forests. Parameter specification can be done by means of an out of bag (OOB) error. The definition of an OOB error is the mean prediction error on each training sample x_i, using only the trees that did not have x_i in the bootstrap sample.

Random forest is a popular ensemble because of the advantages it offers:

1) The performance of random forest is usually better than tree bagging. It is not as vulnerable to noises or outliers, as is the case with other ensemble classifiers (Skurichina, 2002).
2) It has a strong ability to handle large input data with relatively short running time (Skurichina, 2002).
3) Random forest could also serve to calculate the importance of each variable and proximities between pairs of instances (Breiman,2001; Archer, 2008).

4.2. Random subspace

Random subspace is considered a generalisation of the random forest algorithm (Ho, 1998). While a random forest is made up by a number of decision trees, a random subspace comprises any underlying classifiers, e.g. support vector machines (Tao, 2006), linear classifiers (Skurichina, 2002), nearest neighbours (Tremblay, 2004) and other types of classifiers. Below is the random subspace algorithm:

Algorithm 5 random_subspace

Input(s): Training set (x_i, ω_i) for $i = 1,2, \dots, N$; number of classifiers B; number of features F

Output: Classifiers $C_1, C_2, \dots, C_B$

1. **for** $b = 1$ **to** B **do**
2. Randomly select F features without replacement from the whole feature set to feature subset F_b
3. Train classifier C_b using the original data with feature subset F_b
4. **end for**

Figure 8: The random subspace algorithm, also considered a generalisation of the random forest algorithm (Ho, 1998)

Similar to random forest, the random subspace classification algorithm also allows trees to vote for the most popular classifier from the *B* trained classifiers. The random subspace algorithm is, therefore, a useful method where the number of features greatly exceeds the number of training objectives, such as gene expression data (Bertoni, 2005) or functional magnetic resonance imaging data (Kuncheva, 2010).

Predicting customer attrition using the random forest modelling technique could identify customers likely to attrite which the logistic regression modelling technique miss-classified. Class imbalanced data is a common occurrence when a rare event needs to be predicted with ensemble classifiers and other modelling techniques. Class imbalanced data as well as performance measures for the imbalanced data are discussed in Chapter 5.

5. Class-imbalanced data

The percentage of customers that defect from the bank can be deemed a rare event and will result in imbalanced data. Even though it is a small percentage of the total customer base, the number of customers defecting is far greater than the bank is willing to lose, leading to future profit losses. It give reason to considering techniques of addressing imbalance data, due to this rare event, and accurately identifying these cases.

The recent explosion in data in both quantity and diversity is creating a plethora of opportunities for data-engineering research, knowledge discovery and a wide range of applications (He, 2009). Imbalanced data has elicited great interest in machine learning. Class imbalanced data is when the number of observations is not equal amongst classes. Generally, one of the classes consists of a small number of observations compared to the other classes, creating an imbalance. Medicine (Mac Namee *et al.*, 2002), fraud detection (Fawcett *et al.*, 1997), natural language (Cardie *et al.*, 1997), etc. are examples of fields of study where class imbalanced data are prominent.

The problem with imbalanced data is that it compromises the performance of most learning algorithms. The connection between the error cost and the prior probability of a class is discussed by Breiman (1996). Classes with only a few observations in the training set have a lower prior probability and a lower error cost. It poses a problem when the true error cost of this minority class is significantly higher than the observation distribution of the training set conveys, as is generally the case.

The accurate prediction of these minority events is of great importance for classification, since most classification algorithms require equally distributed data among classes and equal misclassification costs. If these algorithms are applied to imbalanced data, they give preference to the majority class and fail to accurately classify the minority classes. Breiman (1996) suggest that the algorithms compromise performance in these cases.

An example of this can be seen with credit card fraud, where fraud amounts to less than 0.1% of all transactions, but its cost translates into billions of currency lost (Hassibi, 2000).

5.1. Measures of classifier performance

This section explores the measures that determine the accuracy and performance of a classification model on class imbalanced data. It is important to determine whether a model predicts a rare event accurately. To put this into perspective: Say a dataset consists of one hundred observations and only five of the observations are credit card fraud. A model that predicts none of the cases as fraud will have an accuracy of 95% or misclassification of 5%. A model that has a misclassification of 5% is considered good. This model will however not predict the fraud cases, which is the main objective of the model and will be rendered useless. Other performance measures would need to be added, as described below.

5.1.1. Overall accuracy

Overall accuracy (ACC) is a popular measure of performance for classification algorithms, since it is very simple to understand and interpret. It is the fraction of the correctly classified records as the numerator and the number of total records as the denominator (Stehman, 1997):

$$ACC = \frac{N_{correct}}{N_{total}} \ . \tag{24}$$

The overall classification error is defined as 1 −ACC. The error measurement serves as the cost function of the classification algorithm used to be minimised or maximised in the training process. The overall classification error assigns an equal misclassification cost, $\frac{1}{N_{Total}}$, to every data point. This works well in the case of balance data as the population is close to equal amongst classes. A systematic bias occurs as data imbalance increase - which has a more significant effect on the measure - as the population for a specific class increases. The class with the largest population will contribute a greater portion and therefore have a greater effect than the minority class. As a result, more points will be assigned to the majority class. Considering the credit card fraud problem, most of the transactions will be classified as non-fraud with a high classification accuracy of 99.9% and of no use, as transactions that might be fraudulent and of interest are not predicted accurately (Stehman, 1997).

It is therefore clear that although overall accuracy is easy to calculate, it needs to be used with caution. There are however other accuracy measures that will account for the limitations of the overall accuracy measure. These are explained in the next section (Stehman, 1997).

5.1.2. Additional accuracy measures

An improvement to the overall accuracy measure would be to partition the wrong prediction (misses) and the correct prediction (hits) in a confusion matrix, as shown in Table 1. The focus is on binary classification in the research report.

Table 1: A confusion matrix indicating the number of correct and incorrect predictions against the actuals

		Predicted label	
		Positive	**Negative**
Actual label	**Positive**	**TP**	**FN**
	Negative	**FP**	**TN**

All possible binary classification outputs are illustrated above in the confusion matrix. A true positive (TP) is the number of cases correctly classified as positive. A true negative (TN) is the number of cases correctly classified as a negative. False positive (FP) is the number of cases incorrectly classified as negative and false negative (FN) vice versa. These four numbers can be used to calculate the overall accuracy (Fawcett, 2006):

$$ACC = \frac{N_{correct}}{N_{total}} = \frac{TP + TN}{TP + FN + TN + FP} \ . \tag{25}$$

5.1.2.1. Sensitivity and Specificity

In addition to ACC, within class accuracy can also be calculated. The within class accuracy measures are specificity, which is the negative class classification accuracy or true negative rate (TNR) and sensitivity, which is the positive class classification accuracy or true positive rate (TPR) given as (Powers, 2011):

$$TNR = \frac{TN}{TN + FP} \ , and \tag{26}$$

$$TPR = \frac{TP}{TP + FN} \ . \tag{27}$$

The limitations of the overall accuracy can be quantified by means of ACC, TNR and TPR (Song, 2014). Suppose the positive cases number in the training set is $N_+ = TP + FN$, while the number of negative cases is $N_- = TN + FP$. Let $k = \frac{N_+}{N_-}$ define the ratio of the number of the two classes. Then ACC can be calculated as (Song, 2014):

$$ACC = \frac{TP + TN}{TP + FN + TN + FP} \tag{28}$$

$$= \frac{TP}{TP + FN + TN + FP} + \frac{TN}{TP + FN + TN + FP}$$

$$= \frac{TP}{TP + FN + (\frac{(TN+FP)}{(TP+FN)} \times (\mathrm{TP} + \mathrm{FN}))} + \frac{TN}{(\frac{(TP+FN)}{(TN+FP)} \times (\mathrm{TN} + \mathrm{FP})) + TN + FP}$$

$$= \frac{TP}{(TP + FN) + \frac{1}{k}(TP + FN)} + \frac{TN}{k(TN + FP) + (TN + FP)}$$

$$= \frac{k}{1 + k} \times TPR + \frac{1}{1 + k} \times TNR \ , \tag{29}$$

with $\beta = \frac{N_+}{N_+ + N_-} = \frac{k}{1+k}$ as the imbalanced rate of the dataset. Eq. (29) can be reformulated as:

$$ACC = \ \beta \times TPR + (1 - \beta) \times TNR \ , \tag{30}$$

with $0 \leq \ \beta \ \leq 1$.

In the case of balanced data, β is near or equal to 0.5 and maximising the overall accuracy is equivalent to maximising the sensitivity and specificity with the same weight. Maximising the overall accuracy will bias toward specificity and less towards sensitivity. In the case of imbalanced data, β therefore approaches 0, which is called positive class minority (Powers, 2011).

Consider a dataset with $\beta = 0.01$. One unit increase of specificity will contribute a hundred times more than the contribution of one unit in sensitivity. Most cases will be classified as negative in the instance of positive class minority, because the increase contributes more to specificity than overall accuracy. The opposite will occur with negative class minority, as β approaches 1 (Powers, 2011).

5.1.2.2. Recall and Precision

Recall (R) is another term used for sensitivity. Precision (P) is given by the number of correctly classified positive cases as numerator and the total number of positive classified cases as denominator. A low number of FP errors will equate to a high level of precision. Precision and recall are metrics employed in applications where it is more important to successfully classify one class over the other. The definitions of these metrics (Ting, 2011) are as follows:

$$R = \frac{TP}{TP + FN}, and \tag{31}$$

$$P = \frac{TP}{TP + FP}. \tag{32}$$

Baseline models that maximises one metric but not the other are easily built. A model will have a perfect recall if the model predicts that all cases are positive, but will have a poor precision score because of the FPs. If the model predicts all the positive cases as positive, on the other hand, it will have a low recall rate because of the low number of TPs and yet a high precision rate. The best model would be one that maximises both recall and precision simultaneously. The harmonic mean of recall and precision is measured by *F* as follows (Ting, 2011):

$$F = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{33}$$

A high value for *F* indicates that the values for recall and precision are high.

5.1.2.3. Receiver operating characteristic (ROC) and area under the ROC (AUC) curves

The TPR (sensitivity) is plotted as a function of the FPR (100-specificity) in the receiver operating characteristic (ROC) for different cut-off points of a parameter. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. The area under the ROC (AUC) curve is a measure of how well a parameter can distinguish between binary outcomes (MathWorks, 2016). The ROC curve is represented by the solid blue line in Figure 9 and the area between the ROC curve and the diagonal line is the AUC (MathWorks, 2016)

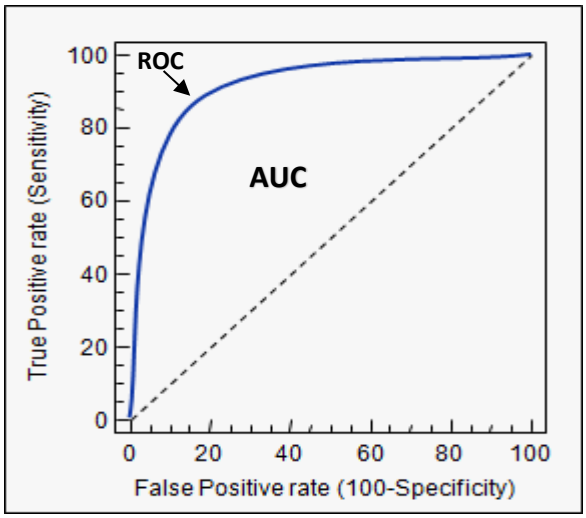


Figure 9: Illustration of the ROC curve and the AUC (MathWorks, 2016)

5.2. Solutions for imbalanced learning

Provost (2000) discusses possible strategies when encountering imbalanced data:

1) High emphasis should be placed on maximising accuracy.
2) The distribution of the training data and the data used to draw the classifier must be the same.
3) The output threshold of the standard machine leaning algorithm should be adjusted when the data is imbalanced.

A number of notations need to be discussed at this point. The imbalanced dataset $\mathcal{D}$ has binary labels {-1;+1}. $\mathcal{D}_{min}$ represents the cases with the minority class, with -1 labels and $\mathcal{D}_{maj}$ represents the set of cases with the majority class with +1 labels. $|\mathcal{D}_{min}|$ and $|\mathcal{D}_{maj}|$ denote the number of observation within each class.

5.2.1. Sampling method for imbalanced data

Provost (2001) poses the question whether a natural class distribution for the training data class distribution would be best. Several popular sampling techniques that effectively deal with imbalanced data are set out in this section. The sampling methods generally modify the degree of imbalance in order to present a balanced distribution (He, 2009). It was found that classification methods are more accurate when they are trained by a balanced training set (Weiss *et al.*, 2001; Estabrooks *et al.*, 2004; Song *et al.*, 2014). Based on this, using sampling methods to obtain a balanced dataset is advisable.

5.2.1.1. Random oversampling and undersampling

Random oversampling is the process of adding a randomly sampled dataset $\mathcal{G}$ from the minority class set $\mathcal{D}_{min}$ when seeking to predict a minority class. The total number of observations in $\mathcal{D}_{min}$ is increased by $|\mathcal{G}|$ to adjust the distribution. Random undersampling requires the removal of random observations from the dataset. During undersampling, a randomly selected set $\mathcal{G}$ of the majority class $|\mathcal{D}_{maj}|$ is removed to provide a dataset of $|\mathcal{D}_{maj}| - |\mathcal{G}| = \ |\mathcal{D}_{min}|$. This is a simple method to adjust the balance of the original dataset $\mathcal{D}$.

It may appear that undersampling and oversampling have the same effect on the dataset and provide the same proportion of balance. This is however not the case, as each method has its own limitations that can potentially hinder the learning process (Holte *et al.*, 1989; Drummond *et al.*, 2003; Mease *et al.*, 2007). In the case of undersampling, valuable information or concepts could be omitted from the majority class necessary for the learning process. In contrast, overfitting can be a consequence of oversampling, because randomly selected examples (duplicates) are appended to the dataset. This gives the impression that more of the original events occurred (Mease *et al.*, 2007). An increase in specificity can result from oversampling. The classifiers produce multiple cases for copies of the same observation. The classification performance in the testing dataset would therefore be worse off in this scenario, but the training accuracy would be high.

Random sampling was used in this study, as there was a sufficient number of observations available for modelling.

5.2.1.2. Tomek links

The definition of Tomek links (Tomek, 1976) is very similar to that of the single linkage definition used in clustering algorithms. Considering two examples $\mathcal{G}_i \in \mathcal{D}_{maj}$ and $\mathcal{G}_j \in \mathcal{D}_{min}$, if $\mathcal{G}_l \neq 0$ and $\mathcal{G}_l \in \mathcal{D}_{min}$ or $\mathcal{G}_l \in \mathcal{D}_{maj}$ for the Euclidean distance $d(\mathcal{G}_i, \mathcal{G}_l) < d(\mathcal{G}_i, \mathcal{G}_j)$ or $\mathcal{G}_l \in \mathcal{D}_{maj}$ it can be said that $\mathcal{G}_i$ and $\mathcal{G}_j$ form a Tomek link. One or both examples will be either noise or borderline when two examples form a Tomek link. Examples from the majority class with Tomek links can be removed as an undersampling technique. Tomek links can also serve as a data cleaning tool by deleting the examples from both classes.

5.2.1.3. Edited nearest neighbour rule (ENN)

The edited nearest neighbour rule employs the kNN methodology as discussed in Chapter 3.1 (Wilson, 1972). The value of *k* is usually set to 3. If the minority class data dominates the *k* nearest neighbour for each point $\mathcal{G}_{\mathrm{i}} \in \mathcal{D}_{\mathrm{maj}}$, remove the point from $\mathcal{D}_{\mathrm{maj}}$ and from the dataset.

5.2.1.4. Condensed nearest neighbour rule (CNN)

The CNN methodology introduced by Hart (1968) is used to find a consistent subset of examples. A subset $\widehat{\mathcal{D}} \in \mathcal{D}$ is chosen from $\mathcal{D}$ if and only if $\widehat{\mathcal{D}}$ could correctly classify the example in $\mathcal{D}$ when using a 1-nearest neighbour. By following these steps, $\widehat{\mathcal{D}}$ can be created:

1) Each point is removed from the training set and determined whether it is classified correctly. Outliers are removed.
2) A new dataset $\widehat{\mathcal{D}}$ is created by drawing a random sample of the majority class and all the examples from the minority class.
3) Random points are selected from the original dataset $\mathcal{D}$ and determined if they are correctly classified based on the points in the new database $\widehat{\mathcal{D}}$ using kNN with k = 1.
4) Correctly classified data points will be left out of the new database $\widehat{\mathcal{D}}$ and the ones incorrectly classified will be removed from the original dataset $\mathcal{D}$ and added to the new dataset $\widehat{\mathcal{D}}$.

The procedure will not find the smallest constant subset from $\mathcal{D}$. The procedure aims to remove the example from the majority class that is considered far enough from the decision border in order to create a consistent subset.

5.2.1.5. Neighbourhood cleaning rule (NCL)

The neighbourhood cleaning rule applies Wilson's ENN to remove majority class examples (Laurikkala, 2001). ENN removes examples with a different class than its nearest two neighbour. NCL is a modified version of ENN that increases the data cleaning process for binary class problems as follows: For every example $\mathcal{G}_i$ find its nearest three neighbours in the training set. $\mathcal{G}_i$ will be removed if it contradicts the classification of the three nearest neighbours from the majority class $\mathcal{G}_{\mathrm{i}} \in \mathcal{D}_{\mathrm{maj}}$. The nearest neighbour of the majority class will be removed if $\mathcal{G}_i$ belongs to the minority class $\mathcal{G}_j \in \mathcal{D}_{min}$ and the three nearest neighbours misclassify $\mathcal{G}_i$.

5.2.1.6. One side selection

One side selection is an undersampling method resulting from the application of Tomek links followed by the application of CNN (Kubat *et al.*, 1997). Only borderline majority class and noise examples are removed using Tomek's undersampling method. This is because borderline examples can negatively affect the classifier, as a small amount of noise could cause them to fall on the wrong side of the decision border.

5.2.1.7. Balance Cascade

The balance cascade algorithm takes a supervised learning approach that develops an ensemble of classifiers to systematically select which majority cases to be included in the undersampled set $\mathcal{G}$ (Liu *et al.*, 2006). $|\mathcal{G}| = |\mathcal{D}_{min}|$ for a sampled set $\mathcal{G}_1$ from majority class $\mathcal{D}_{\mathrm{maj}}$. Subject to $\mathcal{D}_1 = \{\mathcal{G}_1 \cup \mathcal{D}_{min}\}$, an ensemble k_1 is introduced. The classification of k_1 is based on $\mathcal{D}_1$. All the examples correctly classified to $\mathcal{D}_{\mathrm{maj}}$ are called N^1_{maj}. k_1 is known and N^1_{maj} is considered redundant in $\mathcal{D}_{\mathrm{maj}}$ and removed from $\mathcal{D}_{\mathrm{maj}}$, as k_1 is already trained. A new sampled set is then generated from the resulting majority class samples, $\mathcal{G}_1$, with $|\mathcal{G}_1| = |\mathcal{D}_{min}|$. Ensemble k_2 is then obtained, subject to $\mathcal{D}_2 = \{|\mathcal{G}_1| \cup \mathcal{D}_{min}\}$. The procedure is repeated until it reaches the stopping criteria when a cascading combination scheme is used to form a final classifier.

5.2.1.8. kNN undersampling

The kNN classifier structure is employed to guide the kNN undersampling process (Zhang *et al.*, 2003). Four kNN sampling methods were introduced based on the characteristics of the given data distribution:

- NearMiss-1 - Majority examples with the closest average distance to the three minority classes are selected.
- NearMiss-2 - Majority examples with the farthest average distance to the three minority classes are selected.
- NearMiss-3 - A given number of majority examples closest to each minority example is selected in order to guarantee that every minority example is surrounded by the majority examples.
- The "most distant" method - The majority examples with the largest distance to the closest three minority examples are selected.

An experiment by Zhang *et al.* (2003) reflects that NearMiss-2 provides good results with regards to imbalanced learning.

5.2.1.9. Synthetic minority oversampling technique (SMOTE)

Chawla *et al.* (2002) show that a combination of undersampling and oversampling can be used to improve classification performance and argues that it is better than just to undersample a majority class. The synthetic minority oversampling technique creates synthetic minority class examples to boost the minority class, rather than replicating the minority class. Consider the *k*-nearest neighbours in a subset $\mathcal{D}_{min} \in \mathcal{D}$, for each point $x_i \in \mathcal{D}_{min}$ for a specified *k*. In order to create a synthetic sample, one of the randomly selected *k*-nearest neighbours is multiplied by the corresponding feature vector difference with a random number in [0, 1]. This is added to the vector x_i and leads to $x_{new} = x_i + (\hat{x}_i - x_i) \times \delta$, where x_i is one of the *k*-nearest neighbours and $\delta \in [0, 1]$ is a random number. The definition describes a point along the line segment joining x_i and one of its k-nearest neighbours $\hat{x}_i$, which is the resulting synthetic instance. The ties created from random oversampling are broken by these synthetic samples. Chawla (2002) discussed the advantages of SMOTE over random oversampling and random undersampling in further detail.

5.2.1.10. Combination of oversampling with undersampling

The problems pertaining to skewed class distribution are not solved with oversampling the minority class examples to balance the class distribution. Majority class examples frequently

invade the minority class space and as a result class clusters may not be well defined. Artificial minority class examples can expand deeply into the majority class space due to the expanded interpolating minority class clusters. This can be dealt with by applying a classifier, but possibly lead to overfitting. By applying Tomek links to the oversampled SMOTE, better class clusters are defined (Batista *et al.*, 2004). Examples of both classes are removed in the data cleaning process, as opposed to only removing from the majority class. A combination of the SMOTE and ENN methods could also be employed, as the ENN method tends to remove more examples than the Tomek links.

Random undersampling as discussed in chapter 5.2.1.1, was used as the chosen sampling methodology for this study in order to address the class imbalanced data.

5.2.2. Cost-sensitive learning for imbalanced data

While sampling methods are concerned about the balance of the class distribution, cost-sensitive learning is concerned about the costs resulting from misclassification (Elkan, 2001; Ting, 2002). The problems associated with cost-sensitive learning are targeted by using different cost matrices that describe the cost of misclassification of any data example. The cost matrix used by the cost-sensitive learning methodology is seen as a numerical representation of the penalty associated for misclassifying examples. As per Table 2, C(+, -) indicates the cost associated when a positive case is classified as negative. There is no cost associated in the case of correct classification and the cost associated to the misclassification of the minority class is higher than the positive cases, i.e. C(-, +) > C(+, -).

Table 2: Cost matrix

		Predicted label	
		Positive	**Negative**
Actual label	**Positive**	**C(+, +)**	**C(+, -)**
	Negative	**C(-, +)**	**C(-, -)**

The hypothesis developed by cost-sensitive learning, which is the objective to minimise the overall cost of the training set. There are generally three ways of implementing cost-sensitive learning (He *et al.*, 2009):

- Misclassification costs are applied to the dataset in the form of data space weighting. This technique is in essence, cost-sensitive bootstrapping, where the best training distribution is selected based on the misclassification costs.
- Cost-minimising techniques are applied to the combination schemes of ensemble methods. With this approach, ensemble methods are integrated with a standard learning algorithm in order to develop cost-sensitive classifiers.
- The cost-sensitive framework is fitted into classification paradigms resulting from the incorporation of cost-sensitive functions of features. There is no uniform framework for cost-sensitive learning methods, as different classification algorithms have different structures.

Techniques of evaluating the performance of a statistical model are set out in the next chapter. Logistic regression is also discussed, because the performance of the random forest technique was benchmarked against logistic regression. This aided to determine the viability of the model and to compare the results to an alternative modelling technique.

6. Evaluation criteria

The same data and set of variables were modelled through the logistic-regression technique in order to benchmark results. The population stability index and Gini coefficient form part of this evaluation. These were employed to determine the predictability and stability of each model on the data. The results of these tests are discussed in Chapter 10.

6.1. Logistic regression

In model development, the predicted variable is categorical and as such logistic regression is a common and widely used technique. This section focuses on the use of multiple logistic regression to predict a binary outcome, namely attrition (event) or non-attrition (non-event) (Menard, 2018).

Logistic regression uses a set of predictive input variables to predict the likelihood or probability of a specific event occurring (i.e. a customer turning out to defect). Logit transformation is the log of the odds, $L(p_i) = \log\left(\frac{p(event)}{p(non-event)}\right)$ and, as explained in Menard (2018), is used to:

- linearise the available information relating to a probability of an event occurring; and
- limit the outcome of estimated probabilities in the model to between 0 and 1.

The equation for the Logit transformation of a probability of an event occurring is as follows (Menard, 2018):

$$L(p_i) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \ , \tag{34}$$

where $L(p_i)$ is the Logit transformation of the posterior probability of the "event" (p_i) and x_j are input variables for all, $j = 1, \dots, k$. β_0 is the intercept of the regression line and β_j are the parameters for all $j = 1, \dots, k$.

The Maximum-likelihood is measured by the parameters (β_j) and these parameters estimate the rate of change for the Logit transformation for one unit of change in an input variable. This means that the parameter estimates are the slopes of the regression line between the target and the input variables. Since these parameters are dependent on the unit of the input, they need to be

standardised to simplify the analysis. One variable can be a percentage and the other an actual number, for example, like percentage deposit and age. One option when dealing with the unit of input is to perform the regression against the weights of evidence (WoE) of each grouping created in the previous step (Menard, 2018):

$$WoE_i = \left[\left(\ln\left(\frac{Relative\ number\ of\ Non{-}Event_i}{Relative\ number\ of\ Event_i}\right)\right)\right]\times 100\ . \quad (35)$$

Applying the WoE not only solves the problem of having different input units, but also considers the trend and scale of the relationship between groups. It furthermore aids model development by making sure that each characteristic stays intact throughout the regression. Given that the grouping of variables has been done accurately, WoE assists with assigning points that are in line with the trends observed between input variable groups (Menard, 2018).

The WoE needs to be calculated first for each group, within the variables considered for the model, in order to calculate the information value (IV) for each variable. Groups are created within each variable by splitting the values into groups. An example would be to create groups for a variable age as follows: {[0-18] ; [19-25] ; [26-35] ; [36-50] ; [50-120]}. Eq. (36) denotes the formula for calculating the IV for each group. The IVs for all the groups are then added up to provide the IV for the variable (Upadhyay, 2014):

$$IV = \sum_{i=1}^{k}\left\{\left(\frac{Event_i}{(Non - Event_i) + Event_i} - \frac{(Non - Event_i)}{(Non - Event_i) + Event_i}\right) \times WOE_i\right\}, \quad (36)$$

where $Event_i$ refer to the number of positive cases in group i and $Non - Event_i$ refer to the number of false cases in the i^{th} group. The thresholds to determine predictability by means of IVs are depicted in Table 3.

Table 3: IV thresholds to determine predictability of a variable (Upadhyay, 2014)

Information Value	Predictive Power
< 0.02	Useless for prediction
0.02 to 0.1	Weak predictor
0.1 to 0.3	Medium predictor
0.3 to 0.5	Strong predictor
>0.5	Suspicious or too good to be true

6.2. Population Stability Index

The population stability index (PSI) is used to quantify the population shift (over time) from the training dataset to the test datasets, in other words, from the historical data to present day data. It is important to determine whether the data on which the model was built are sufficiently similar to present day data. Generally, a PSI of more than 0.25 is considered a significant shift (Murdoch *et al.*, 1975).

The formula of PSI is shown in Eq. (37) below (Murdoch *et al.*, 1975):

$$PSI = \sum \left(\left(\frac{n_{di}}{N_d} \right) - \left(\frac{n_{vi}}{N_v} \right) \right) * \ln \left(\left(\frac{n_{di}}{N_d} \right) \Big/ \left(\frac{n_{vi}}{N_v} \right) \right), \quad (37)$$

where n_{di} is the number of observations in the i^{th} group of the training dataset, n_{vi} is the total number of observations in the i^{th} group of the test dataset and N_d and N_v are the total number of observations in the training and test datasets respectively.

6.3. Gini coefficient

The Gini coefficient (G) is a measure that is traditionally encountered when exploring income inequality, which is a plot of wealth concentration introduced by Max Lorenz (Lambert *et al.*, 1993) and developed by the Italian statistician Corrado Gini (Lambert *et al.*, 1993).

The Gini coefficient is a measure that can also be used to indicate the predictive power of a model and indicate how well a model can differentiate between good (non-attrition) and bad (attrition)

cases (Siddiqi, 2005). A predictive model will assign high scores to cases that are less likely to attrite (low probability) and low scores to cases that are more likely to attrite (high probability).

A Gini coefficient calculates a scale of predictive power from 0 to 1, whereby a Gini coefficient of 0 means that the cases are randomly classified and a Gini coefficient of one means the model classifies the cases 100% correct or separates event vs. non-event 100%. The diagonal line in the graph (Figure 10) represents a Gini coefficient of 0. The Gini coefficient can be graphically illustrated as the area between the diagonal line and the Lorenz curve, in which the greater the area the higher the Gini coefficient (EFL Global, 2015).

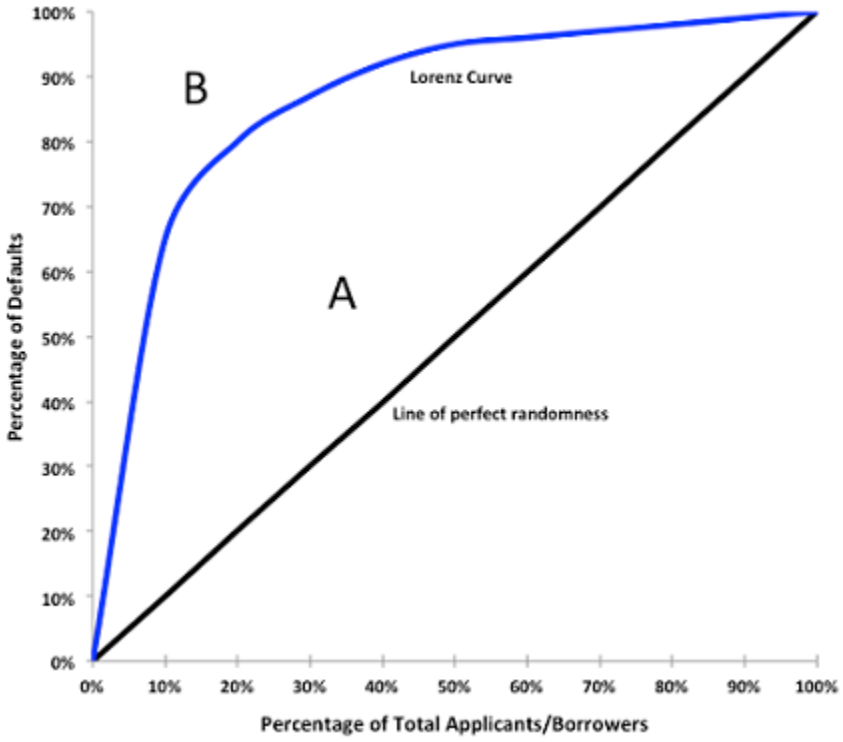


Figure 10: Example of a Gini coefficient graph (Lending times, 2016)

A higher Gini coefficient means more predictive power and a lower Gini coefficient means less predictive power. The formula to calculate the Gini coefficient is as follows (EFL Global, 2015):

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|x_i - x_j|}{2n^2\bar{x}}, \tag{38}$$

where x is an observed value, n denotes the number of observations and $\bar{x}$ is the mean value.

The Gini coefficient should not be confused with Gini impurity reviewed in Chapter 3.2.

Customer lifetime value is addressed in the next chapter in order to highlight the importance of retaining a customer base and the value it adds to prevent customers from defecting.

7. Customer lifetime value

Customer lifetime value (CLV) is highly dependent on how long a customer remains with a bank, emphasising the importance of a bank's retention strategies. The longer the customer remains with the bank, the greater the profits gained as a result of the relationship between the bank and the customer.

CLV can be defined as the present value of the customer's future predicted cash flows resulting from the customer's relationship with the bank (Pfeifer, 2005). A customer is deemed profitable if the revenue generated from the customer exceeds the cost of acquisition (Kotler & Armstrong, 1996).

Profits generated from customers tend to accelerate from one cycle to the next, which is one of the reasons banks aim to retain their existing customers. The profits generated do not stay constant over time. A study done on a credit card company found that the revenue generated does indeed increase year on year (Reichheld & Sasser, 1990).

There are four reasons why this acceleration is observed (Reichheld, 1996):

- Customers will use their products with more ease over time. In the first year of having a credit card customers tend to be cautious in using it, but as they become more comfortable with it, they start to utilise it more.
- Existing customers are more familiar with the bank's offerings and require less time from staff to assist the customer with the available services. This saves employee costs.
- Existing satisfied customers recommend the bank to their peers, which creates new business. They are also more likely to buy additional products from the bank (cross-selling).
- Sometimes long-term customers pay more for services, when the pricing structure in place at time of take-up no longer applies and since they do not qualify for the promotions a bank offers to attract new customers.

Customer lifetime value is affected by the customer acquisition and retention, as discussed in the next chapter.

8. Acquisition and retention

Customer retention is equally, if not more, important than acquisition for generating as much profit from customers as possible. Sound retention strategies are thus essential to the organisation to generate the necessary profits to cover the initial acquisitions costs and turn it into a profitable relationship.

According to Pfeifer (2004), the cost of acquiring a new customer is five times more expensive than retaining an existing one. In some instances, for example in the cellular industry, it is believed that the cost can be up to ten times higher (Pfeifer, 2004). Although some disagreement exists, there is general consensus that it is far more expensive to acquire a new customer than to retain an existing one (Sterne, 2003).

Churning costs consist of CLV and the loss of shareholder value. It was found that the churning costs in the cellular market in the USA totals more than four billion dollars each year (Sterne, 2003).

Reichheld and Sasser (1990) conducted a study on a credit card company. They found that if the defection rate could decrease from 20% to 10%, the average lifespan of the customers doubles from 5 years to 10 years, which effectively more than doubled the profits. If the defection rate was to drop by another 5%, the average lifespan of the customer doubles again and the profits rise by another 75%.

It is a riskier strategy to focus predominantly on acquiring new customers, as new customers are more likely to continue their churning behaviour when they are new to a company (Lewis & Bingham, 1991; McNeal, 1999).

Switching to a new bank has become increasingly easy. South African banks created teams of *switching experts* to assist new customers to sign up with them and the South African government is also pressuring banks to facilitate the process of changing to another bank. A "switching code of conduct" was introduced to assist in this regard, which lists certain

requirements that need to be fulfilled by the existing bank when a customer seeks to switch to a new bank. The code of conduct forces the existing or previous bank to provide all debit order information to the new bank within a predetermined period and it also becomes liable for any costs should the deadline not be met, due to errors and delays (Wasserman, 2010).

8.1. Customer satisfaction and customer retention

Since the mid-1970s, conferences have been held on customer satisfaction and the proceedings published in journals such as the *Journal of Customer Satisfaction, Dissatisfaction and Complaining Behaviour* (Hunt, 1977). Numerous publications have been released stating that customer satisfaction leads to customer retention, which in turn encourages banks to strengthen their relationship marketing strategies. Kotler (1994) describes it as follows: "The key to customer retention is customer satisfaction."

Three known groups of studies have been conducted in this regard (Hennig-Thurau *et al.*, 1997). The first and most researched is the use of monetary data, such as profit and revenue as the dependent variables (Reichheld & Sasser, 1990; Anderson *et al.*, 1994). Among the limitations of this group of study is that the data is aggregated in such a way that it is virtually impossible to do the analysis on an individual customer level. The other limitation is that profits are determined by a range of variables that are highly correlated, thus inhibiting the validity of the statement on the customer's relationship.

The second group of studies analyses the *repurchase intentions* of customers on an individual level to investigate the relationship between satisfaction and retention (Oliver & Swan, 1989; Bitner, 1990; Oliver & Bearden, 1985; Oliver, 1980). A limitation of this group of studies is that the data gathering process is done through questionnaires. Because satisfaction values and intention measures are included on the same questionnaire, the data observed is highly correlated. This correlation leads to overestimating the strength of the relationship. It is also observed that the predictive validity of intention measures varies, depending on the

measurement scale, the product, the nature of the respondent as well as the timeframe when determining customer loyalty (Morwitz & Schmittlein, 1992).

The last group of studies focuses on real purchasing data, on an individual customer level, to analyse the relationship between satisfaction and retention (Hennig-Thurau *et al.*, 1997). These studies show virtually no correlation between the variables considered for the respective models. This group provides better results than the first and second group of studies, since none of the problems encountered in the latter have arisen.

Hennig-Thurau and Klee (1997) indicate that a conceptual model showed that the relationship between satisfaction and customer retention is moderated by relationship quality. They aimed to predict the retention rate (dependent variable) and customer satisfaction as the independent variable in order to estimate the impact of customer satisfaction on customer retention, with the different aspects of the perception of a customer's quality as a mediating variable.

Customer satisfaction, as discussed above, is relevant to attrition, but falls outside the scope of this research as it would involve the collection of survey data and thus the effect of customer satisfaction on retention rate is not considered further in this work.

8.2. Customer Relationship Management

Customers are one of the most important assets of any bank in any part of the world. Courteousness, efficiency and correctness are important characteristics of any leading bank (Gayathry, 2016). A satisfied customer will market the bank by word of mouth, which in turn will increase the bank's customer base. A survey done by KPMG (2015) showed that a large portion of the next generation banking clients would not hesitate to switch banks or any other organisation.

In order to prevent customers from leaving to join a competitor, state-of-the-art policies and strategies should be in place to predict customers' demands and how to resolve any issues a customer may have. In addition, the bank must know how to promote and approach the customer with the right product. This will drive sales as well as retain customers that are prone

to leaving the bank to seek alternative providers that will address their individual and unique demands (Gayathry, 2016).

Putting the right strategy in place and offering the right products to the right customers will enable banks to approach customers in the most cost-effective way. An effective strategy needs to be scalable to target and affect a range of customer behaviour. There are initial set-up costs involved, but the strategy persuades customers to remain with the bank and thus the bank does not lose acquisition costs when customers churn. A good strategy will be a hugely cost-saving and revenue driving tool that will result in increased sales and customer retention (Gayathry, 2016).

CRM entails maintaining customer satisfaction as well as persuading customers considering leaving to stay with the bank. The proper implementation of a strong customer retention model is of paramount importance, since it could aid in retaining customers that are on the verge of leaving the bank. Effective communication is equally important to sustain a healthy relationship with customers (Gayathry, 2016).

A sound strategy for customer retention needs to be in place for a significant amount of time to ensure its success, given that customer retention is an evolving process in which the organisation needs to learn from past experience and be well informed of customers' wants and needs. A customer information system (CIS) is a system that manages and stores incoming communication from customers through channels such as online platforms, telephonic communication, etc. The database is available for cross-referencing customers' information and is important in the distribution of relevant customer data. As Newby (2007) indicates, there needs to be a reliable CIS in place to efficiently convey communication between the customer and the business.

Due to the volatility of customers, banks are not ensured of the continuous business of a customer once they have joined. Communication from the business has a psychological effect on a customer and it could be used to the business' advantage. A personalised relationship between customer and business is becoming increasingly important, which entails catering to a customer's individual needs and reacting swiftly to any concerns the customer may have (Redstarsim, 2015). The complexity of communication with customers increases with the increase of channels

available to the customer to access banking. CRM can be a reliable tool in monitoring customer behaviour, interaction and communication. Comprehensive reports can also be generated of customer interaction with the bank (Chakrabarty, 2004).

An effective CRM model will enhance customer satisfaction, motivating customers to continue transacting with the bank and using its services. The collection of data will enable the bank to more effectively model customer behaviour and to determine when what action is needed (Chakrabarty, 2004).

The high-level aim of such a model would be to attain or generate customer loyalty, which would require that the total customer experience be a key focus point for any bank doing business in any economy across the world (Gayathry, 2016). Service needs to be of a high quality in order for a bank to remain competitive in today's market, whether it is to attract new customers or keep existing customers from leaving the bank. Innovation in the technological arena is important in dealing with these issues, but difficult challenges are inevitably encountered.

A recent study by Gayathry (2016) attempted to develop an empirically tested CRM model for banks to enrich their CRM strategies. The objective of this study was to examine the effectiveness of CRM in banks and to identify the effectiveness of the CRM strategy, whereby shortcomings in their processes were determined by means of an empirically tested CRM model.

The study was, however, only conducted in sample areas and was based on the perceptions of customers and the views of the banks, in reference to the current economic climate, which will inevitably change over time. Customer behaviour is also different in different cultures/countries and could be considered another limitation, since the study targeted a specific group of customers.

The study of CRM at banks by Gayathry (2016) included descriptive as well as analytical research methods. Data was collected by means of a questionnaire completed by banking executives and their customers of old and new private sector banks as well as multinational banks. The secondary data was gathered from a range of print and online sources.

The author attempted to match the opinions of customers and bank employees across six elements to determine whether the CRM that was implemented was perceived in the same manner and then modelled based on a couple of demographic variables. The CRM elements (dependent variables) were modelled by means of multiple regression analysis, using demographic variables as the independent variable.

The six CRM elements in question were:

1. customer acquisition;
2. customer satisfaction;
3. customer loyalty;
4. maintaining CRM through general policies;
5. implementing CRM; and
6. maintaining CRM through specific strategies.

The study did not aim to predict the probability of a customer leaving the bank, but it highlighted many elements to consider. This however resulted in the successful implementation of a high-ranking predictive CRM model (Gayathry, 2016).

8.3. Predicting customer retention

Larivière and Van den Poel (2005) set out to predict customer retention and profitability using random forest and regression forest techniques. This research is similar to their study, except it does not review regression forests and profitability factors as included in the cited study. Reinartz and Kumar (2000), however, argue that customers that are less prone to attrite are not necessarily the most profitable customers.

Larivière and Van den Poel (2005) explicitly tested the differences with regards to the impact of the same variable set in random forest and logistic regression. Their research investigated repeat purchases as well as attrition outcome. Three models were built that predicted next buy (repeat purchase), active partial-defection and customer profitability. The probability of a customer purchasing another product was set as the dependent variable for the first retention model (next buy), given a set of independent variables.

There are two types of financial products available to a customer. The first product has an expiry date, for example a personal loan. If a customer applies for the personal loan, they receive a payment plan and once the loan has been repaid, the product ends. If a customer applies for a cheque account with an overdraft, the product does not end until the customer closes it, in other words a non-ending product. A non-ending product was analysed for the active partial-defection variable, which is defined as a customer cancelling a non-ending product. It is called "partial" since a customer can close one product, but still have another (Larivière *et al.*, 2005).

Two measures were combined for customer profitability. Profit evolution served as first profit measure, which is the evolution of the profits generated over the observed period and profit drop the second. Profit drop is a binary variable indicating whether the customer was less profitable by the end of the observation period (Larivière *et al.*, 2005).

Larivière and Van den Poel (2005) analysed two outcomes: customer retention and customer profitability. Retention was investigated by measuring the next buy as well as the closure of an open product. Customer profitability was analysed by means of a linear (profit evolution) and a binary (profit drop) dependent variable.

This methodology modelled customers' next buy, partial-defection and profit drop by means of random forests, as discussed in Chapter 4.1, which were binary measures. Regression forests were used to predict the linear measure, profit evolution.

Decision trees have become a common binary classification tool due to the ease of use and interpretation (Duda, Hart & Stork, 2001). Decision trees are very good for dealing with covariates measured at different measurement levels, including nominal variables. Some of the disadvantages, as mentioned by Dudoit, Fridlyand and Speed (2002), are the lack of robustness and suboptimal performance. Many of the disadvantages are being addressed by using an ensemble of decision trees, followed by a vote for the most popular class, called forests (Breiman, 2001) and the result of a decision tree optimisation, as discussed in Chapter 4.1.

The study by Larivière and Van den Poel (2005) used the random forests introduced by Breiman (2001), which is the same method used in this study. A random forest randomly selects a set of *m* predictors to grow each tree, where each tree is grown on a bootstrap sample of the training set, and the number (*m*) of split nodes is smaller than the number of available input variables for the analysis.

The application of random forest models is growing in popularity, especially in the bioinformatics field (Deng *et al.*, 2004), but rarely in the economic and marketing fields (Buckinx & Van den Poel, 2005). Random forests are among the best prediction techniques available (Luo *et al.*, 2004) and possesses the interesting feature of showing which independent variables have the strongest impact on the dependent variable that is being modelled (Ishwaran *et al.*, 2004). Random forests are robust and can often solve the problem that a single decision tree cannot. They are also fast to compute and easy to use, since the user only needs to specify how many trees need to be created and the number of variables (*m*) that need to be randomly selected from the available subset of variables.

Regression forests was another modelling technique used in the study by Larivière and Van den Poel (2005). The principles of random forests can be applied to regression cases. Regression can be used to grow trees that depend on a random vector in such a manner that the tree predictor takes on a numerical value and not a class label, as in the case of a random forest (decision tree). The predictor is formed by taking the average number of trees.

The four dependent variables investigated in the Larivière and Van den Poel (2005) study were: active partial-defection, next buy, profit evolution and profit drop. Active partial-defection, next buy and profit drop involve binary classification, which are predicted with random forests. Larivière and Van den Poel (2005) opted for normal logistic regression (see Chapter 6.1) to benchmark their results, using the same set of customers, dependent and independent variables. Profit evolution represents the change in the customer's profitability during the period of analysing and can have a wide range of positive and negative values. The mean absolute deviation (M) is used to evaluate the predicted values.

$$M = \frac{1}{n}\sum_{i=1}^{n}|P_i - R_i| \ , \tag{39}$$

where n is the sample size, P_i the predicted profit evolution for customer i and R_i the real profit evolution for customer i.

A non-parametric test, introduced by De Long *et al.* (1988), was used to compare the results of random forests vs. logistic regression. The training set of 50 000 was made up of a random selection customers and another set of 50 000 was selected for validation and data extraction and observation occurred within a set period. Banking service data and insurance data was provided by a Belgium Bank. The explanatory data comprised of:

- past customer behaviour data;
- specific product ownership;
- internet vs. branch banking usage;
- total number of products as well as monetary value;
- cross-buying indicators;
- customer demographics;
- age;
- lifecycle stage;
- gender;
- geo-demographic data (geographical area of residence);
- geographical region; and
- intermediaries.

The AUC served to benchmark the predictive accuracy of the random forests against the linear regression. ROC curves are used to judge the discrimination ability of various statistical methods that combine variables, test results, etc. for predictive purposes and the most common quantitative index describing a ROC curve is the AUC, as discussed in Chapter 5.1.2.3 (Hanley & McNeil, 1982). It was found that random forests and regression forests predicted the four dependent variables better than the logistic regression model.

The predictive accuracy was quite low for the next-buy variable in the study by Larivière and Van den Poel (2005), but the difference in predictive accuracy was significant, with an AUC improvement (DeLong *et al.*, 1988) of 0.006 for the validation sample and 0.005 for the estimation sample. The predictive power of the profit drop variables was also significant with an AUC difference of 0.019 for the validation sample and 0.019 for the estimation sample. It was observed that the greatest difference in AUC (validation - 0.106; estimation - 0.094) was the partial-defection prediction.

The findings are insightful when considering a predictive model for use as a retention model, since random forests clearly outperformed logistic regression. Logistic regression is a widely used technique in the banking industry, especially when predicting a customer's probability of defaulting on a credit product (Sohn & Kim, 2007).

Chu *et al.* (2007) aimed to predict customer churn in the cellular market and also modelled the policies that need to be implemented to prevent the customer from churning. It was observed that 53% of the time, customers churned for reasons other than pricing. After calculating the probability of churn, customers were clustered by means of a policy model and each group was labelled with the most significant attribute. Appropriate policies were then created for each cluster. The policy model used classification to predict the conditions under which a subscriber may defect. Clustering was used to create the policies for each group.

The churn model used historical data including defection history, deactivation behaviour, usage patterns, payment history, spending trends and transaction changes. A decision tree was used to model the probability of churning. The model was tested with sample data and found to predict churn with 85% accuracy. The effectiveness of the policy model remains unknown due to a lack of real data.

The study done by Chu, *et al.* (2007) highlights the importance of taking appropriate action when high churn probability customers are identified, otherwise prediction is done in vain. There is, however, more work that needs to be done on the policy model and real-life testing is required. A significant limitation of this study is that the modelling technique was not highly effective, since

decision trees are highly biased to the training set and the model was not validated on real-life data.

In another study by Lariviere and Van den Poel (2004) they attempted to understand why customers would abandon a particular company producing a product or service for a competitor as well as they can be prevented from defecting. They studied the defection of savings and investment (SI) customers for a large financial service provider, by examining the duration of the products (fixed term vs. non-ending products) and the capital and revenue risk related to the higher-risk products, such as car and fire insurance. The study set out to investigate the impact of cross-selling on the vulnerability of the customer to churn.

They firstly looked for explanatory insights into the timing of the churn event by means of Kaplan-Meier estimations. The Kaplan-Meier estimator is a non-parametric statistic using lifetime data to estimate a survival function (Kaplan *et al.*, 1958.).

Emphasis was then placed on the customers that were most likely to defect, identified in the first explanatory analysis. Lariviere and Van den Poel (2004) selected two customer types in the analysis, of which the first customer type has one fixed-term product with a certain expiration date. They compared this customer to a customer with an opened product that subsequently opened two or more products. The impact of opening a second or third product was examined to determine the difference in likelihood to churn.

The churn rates of seven SI products were analysed. It was interesting to find that customers with different SI products had different churn rates. The most popular SI product had the highest churn rate, indicating the need for a marketing strategy to persuade customers not to churn. The higher-risk products and insurance products had the lowest churn rates, highlighting the benefit of having insurance products as part of a bank's offerings. It was also observed that there is a high tendency to churn when products expire, indicating the need for a sound strategy to approach these customers before their products expire.

The question was then asked: What products can be cross-sold to retain these customers that are on the verge of leaving the financial institution? A multinomial probit was built to help select

the products to cross-sell in order to reduce customer churn by estimating the customer's preference with regard to products that can be cross-sold. The multinomial probit model was also used to test the findings of the survival analysis. Based on the results of the analysis, the type of SI products to market to the customer could be identified.

This study highlights the importance of a customer having more than one product at the financial institution and the direct correlation with a low probability of attrition. This was considered when developing a retention model in this study.

9. Methodology

This section describes the methodology followed and a description of the data gathered for the research.

9.1. Enabling software and hardware

SAS Enterprise Guide 7.1 software was used for data manipulation and SAS Enterprise Miner 13.2 software for model building purposes. Further analysis was done in SAS Enterprise Guide 7.1 software (SAS Institute, 2016).

The study was conducted on a personal computer, using a 64-bit Microsoft Windows 365 software operating system with 16GB ram and a 2.60 GHz processor.

9.2. Data description

This section details the data and time periods pertaining to the model building process as well as the ethical considerations related to the personal data of customers and its protection.

9.2.1. Ethical clearance

No personal data formed part of the training, validation and test datasets, i.e. identification numbers, names and addresses. Personal data was only used in the beginning stages to obtain customer numbers, internally generated and therefore unique to the bank. The customer numbers were used to merge all the relevant data, after which the personal data and customer numbers were removed from the dataset prior to modelling. The modelling datasets are stored on secure servers and there are no means to trace the data back to any individual.

The data cannot be obtained without access to the library where the data is stored (this includes other banking staff). Only the person involved in this research report had permission to access the data, which was deleted as soon as it was no longer necessary for storage.

Permission was provided by the bank to use the data and the University of Witwatersrand obtained the necessary ethical clearance (ethical clearance number: MIAEC 004/18).

9.2.2. Training and testing datasets

This study focuses on predicting, by means of various data sources, the probability that customers will attrite. A model would need to predict at least three months or more in advance

whether the customer is planning on leaving. One month would be too short, because the customer would by then already have made up their mind and any attempt from the bank to persuade the customer otherwise may be too late (Ascarza, 2016).

The scope of the study was limited to customers with a Gold status, although the model could also have incorporated the easy, platinum, private client and private wealth segments. Gold segment customers have an annual gross income of between R84 000 and R300 000, which makes up +/-20% of the bank's customer base and is a profitable segment. The lower income segment consists of approximately 65% of the entire consumer base, but is not as profitable and thus the Gold segment was the preferred segment for this research.

The bank's data warehouse stores extensive data on its customers and served as the model's information source. The data attributes explored included transactional data, customer behavioural data and demographic data on the customers. The entire Gold customers base for a specific time period of February 2016 to April 2017 was extracted and then sampled for the training set. The total population can be seen in Table 4, indicating the sample sizes in brackets.

Attrition and non-attrition cases were both sampled to 100 002 records in the training dataset, of which 80% of the training dataset was used to built the models and the remaining 20% was used for validation. The defection status, known on the training set, served as the dependent variable. Testing was done on test dataset 1 (TDS1) in Table 4, in which the defection status is known. TDS1 consisted of data from August 2016 to July 2017 (refer to Figure 11 for data timelines). Further model testing was conducted on test dataset 2 (TDS2) from November 2016 to August 2017. The observation data of the three datasets overlap, but this is not a concern as the outcome periods differ.

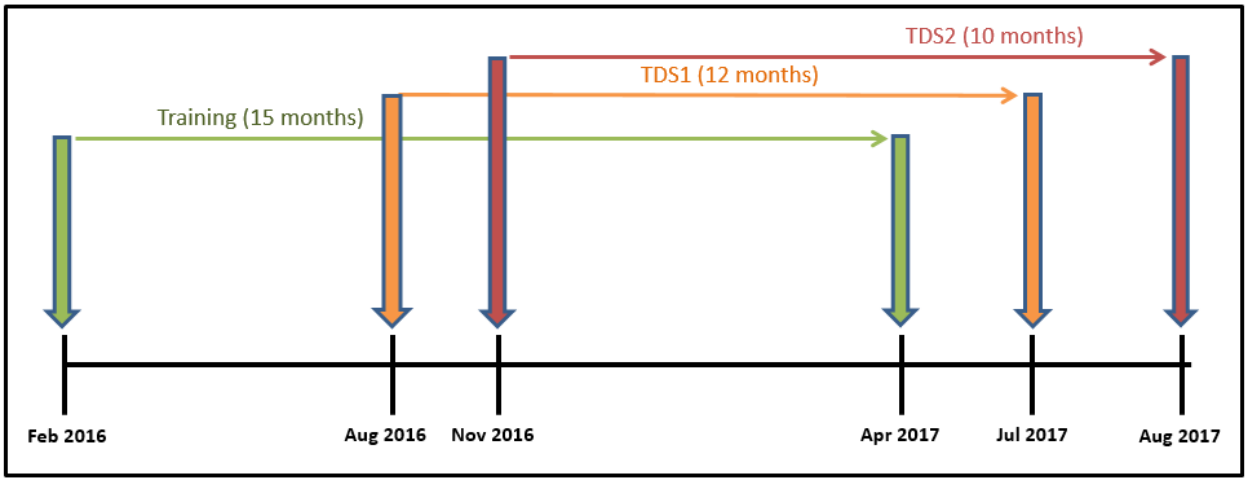


Figure 11: Data extraction timelines

Table 4: Population sizes and attrition rates for the training, validation and test datasets

	Attrition		
	No	**Yes**	**Rate**
Total	3 453 932 (100 002)	229 850 (100 002)	6.7%
Training	2 763 120 (80 002)	183 917 (80 002)	6.7%
Validation	690 812 (20 000)	45 933 (20 000)	6.6%
TDS1	1 725 386	114 761	6.7%
TDS2	573 457	36 833	6.4%

Data was sampled during the training process in order to reduce computational power and storage space. The training population consisted of 3 683 782 observations. The population that did not defect during the observation period was undersampled by means of the simple random sampling technique (discussed in Chapter 5.2.1.1) to 100 002 observations. The population that did defect during the observation period was oversampled in relation to the observations that did not defect to 100 002 using the same method. Weights were applied during the model training process.

The proportion of attrition events was greatly inflated in relation to the non-attrition events, due to the sampling that was done. Sampling occurred in such a way that 50% of the dataset were attrition cases and the other 50% non-attrition cases. Attrition cases however represent 6.7% of

the population. Weighting was applied during the data importation process and did reflect the true population and the model was adjusted to accommodate for the sampling. The datasets, TDS1 and TDS2, were not sampled.

9.2.3. Data sources

The data sources that were used are as follows:

Demographic data: This database consisted of the data supplied by the customers upon applying for products at the bank. Data is continually updated according to the latest provided data when customers make changes to their profiles. This data consists of variables like age, gender, income etc. Personal data, such as identification number, names or addresses were excluded.

Behavioural risk scoring (BRS) variables: This is an extensive data source of internal customer behavioural variables, ranging from product ownership, customer behaviour on owned products, monitory variables on products owned and cross-buying.

Transactional data: The transactional database is an aggregated view of customer transactions over different time periods and channels, whereby these aggregations are done on both debit and credit card transactions. The channels included for these aggregations are point-of-sale transactions, ATM transactions, cash transactions, eCommerce transactions and all the above aggregated. Aggregated variables are created in this database and then presented in ratio form, to counter the effect of inflation on monitory variables.

The BRS, demographic and transactional data are readily available to quantitative analysts employed by the bank.

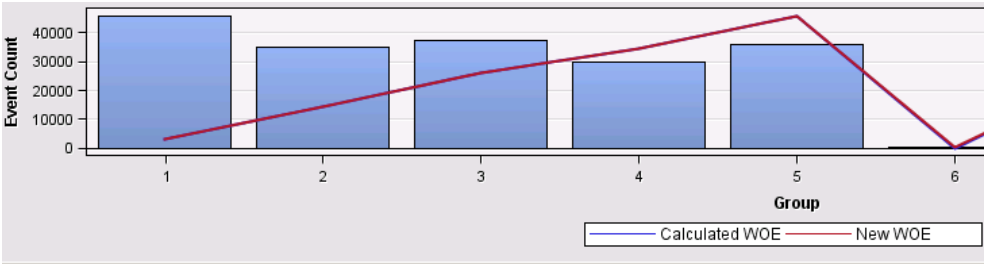
9.3. Model building process

The model building process started off with data collection and analysis. Suitable data sources were obtained and merged to compile the training, validation and test datasets that would provide data with the potential to predict attrition. SAS Enterprise Guide was used for the data collection.

The training data was imported into SAS Enterprise Miner once the training set was compiled. The target variable, weight variable and input variables were specified as part of the import

process, upon which the programme calculated the variable importance and only variables above the importance cut-off were considered for the model. The cut-off was set at a Gini coefficient of 10, meaning that all variables with a Gini coefficient of lower than ten were omitted. This cut-off left only 30 variables for building the models. The random forest, as discussed in Chapter 4.1, randomly selects variables to build the trees, i.e. variables from the available thirty were randomly selected with each iteration.

All the variables considered for the model are manual grouped. The grouping process entails manually creating groups within each variable, whereby groups are created by grouping values of a variable together that have the same attrition or event rate. The attrition rate trend of the variable needs to increase or decrease in such a way that it makes logical sense. Figure 12 illustrates the desired outcome after grouping, indicating that pockets of data with the same event rate are grouped together in such a manner that the groups combined WoE_i, see Eq. (35), trends in an upwards or downwards fashion.



Value	Group	Event Rate	WOE
MISSING	6	0.116	-0.67711
-91919	7	0.0	0.0
-91919.00	7	0.0	0.0
-92929	8	0.0	0.0
-92929.00	8	0.0	0.0
-93939	9	0.0	0.0
-93939.00	9	0.0	0.0
var30 < 368	1	0.123	-0.74114
368 <= var30 < 375	1	0.109	-0.60925
375 <= var30 < 379	1	0.092	-0.41819
379 <= var30 < 382	2	0.092	-0.4194
382 <= var30 < 385	2	0.082	-0.3006
385 <= var30 < 388	2	0.076	-0.21477
388 <= var30 < 390	3	0.069	-0.10357
390 <= var30 < 393	3	0.065	-0.03801
393 <= var30 < 395	3	0.062	0.008
395 <= var30 < 397	3	0.057	0.09045
397 <= var30 < 400	4	0.055	0.1334
400 <= var30 < 401	4	0.047	0.30637
401 <= var30 < 404	4	0.05	0.22782
404 <= var30 < 406	4	0.05	0.23664
406 <= var30 < 409	5	0.045	0.351
409 <= var30 < 412	5	0.044	0.37478
412 <= var30 < 415	5	0.04	0.4721

Figure 12: A screenshot of a trending variable after grouping, using SAS Enterprise Miner (SAS Institute, 2016)

SAS Enterprise Miner selects the default maximum number of trees as fifty and maximum depth also as fifty. These settings caused the random forest to predicts very well on the training set, but the prediction accuracy falls drastically with the validation dataset, indicating that the model overfits. A number of different settings were explored to determine a satisfactory combination between the maximum number of trees and the maximum depth; considering predictiveness, robustness and overfitting. The model proved to predict lower, but closer to the validation set when the maximum number of trees was changed to twenty and the maximum depth was changed to ten, which resulted in the optimum combination from all the different settings explored.

A random forest model was used to test its ability to predict attrition. A logistic regression model was then built to predict attrition and served as benchmark for the random forest model. The SAS code for the final models of the random forest and logistic regression was then exported to SAS Enterprise Guide in order to run the training, validation and test datasets through the models. Further analysis was performed on the output of the models in SAS Enterprise Guide, which is described in Chapter 10.

10. Results

This chapter reports the results of the PSI, Gini coefficient, misclassification and variable stability in relation to the methodology described in Chapter 9.

10.1. PSI on datasets

TDS1 and TDS2 were used to determine whether the model remains robust and stable over time, since a population shift would shift the model and ultimately cause the model to lose predictive power. Population stability tests became necessary from training to TDS2 to ensure that there is no volatile variable selected in the model that might compromise predictability in the short term. These tests would highlight such occurrences. In order to perform a PSI, the training outcomes of the models were compared to the outcomes of TDS1, after TDS1 was scored through the same models to give a value between zero and one. The training outcomes of each model were also compared to the outcome of TDS2 as an additional stability test.

The loss of predictability and a shifting population are inevitable over time. Recalibration can be done on the model if a shift is observed, but the rule of thumb is to rebuild a model, four years after implementation, depending on the shift in population. The cumulative probability distribution for the training dataset, TDS1 and TDS2 are depicted in Figure 13 and Figure 14. These illustrate the population shifts across the three datasets for both the random forest and logistic regression model.

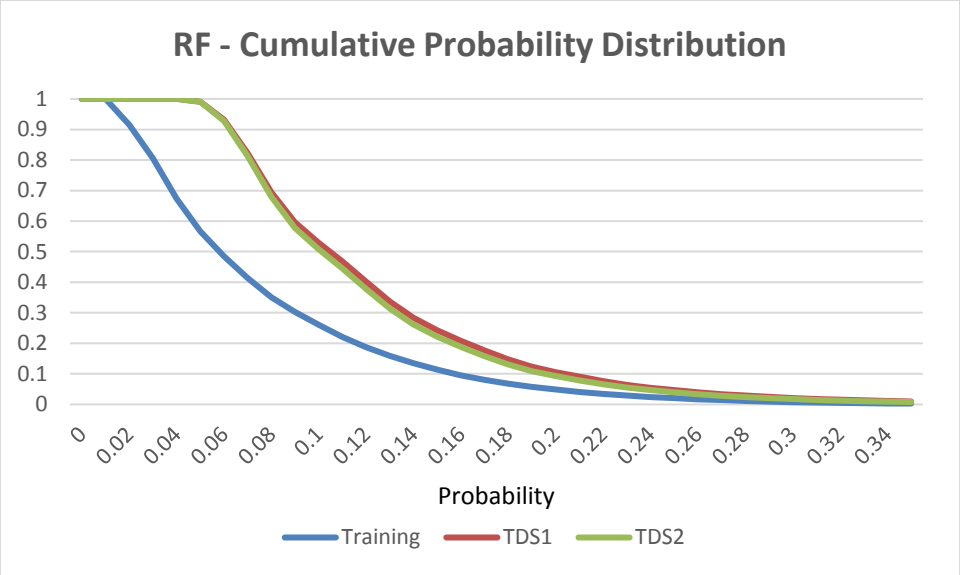


Figure 13: The random forest cumulative probability distribution for the training and test datasets

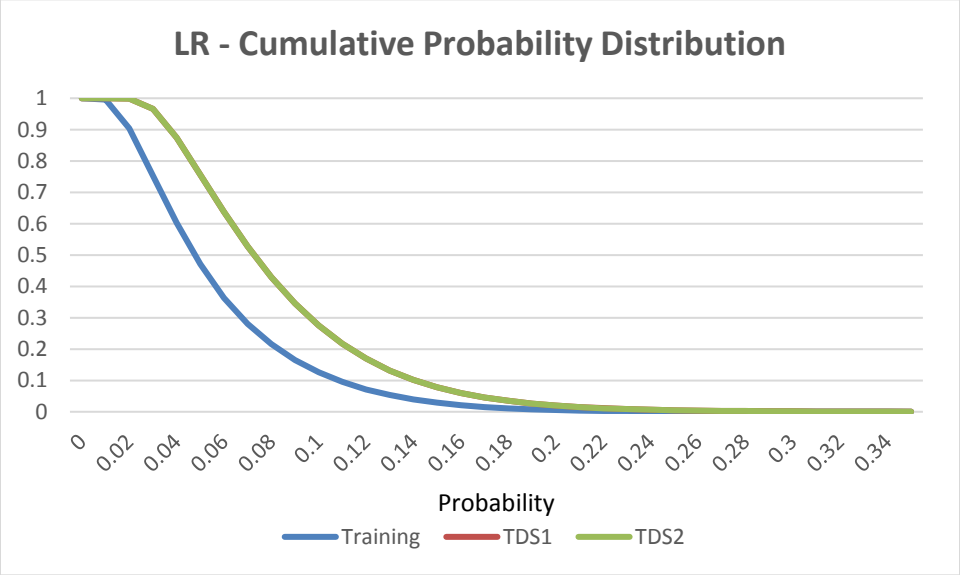


Figure 14: The logistic regression cumulative probability distribution for the training and test datasets

Figure 13 and Figure 14 reflect a shift in both the random forest and the logistic regression model, with the shift being more severe in the random forest model. This shift can be quantified by performing a PSI to determine whether the shift is within reasonable bounds, as explained in Chapter 6.2. The PSI for the random forest for training vs. TDS1 is 0.22 and 0.24 for the training dataset vs. TDS2, which is very close to the limit of acceptable shifts. This translates to the fact

that if the shift was above 0.25, the model would have had to be rebuilt with different variables. The logistic regression shifts were not as severe, at 0.17 for training vs. TDS1 and 0.18 for training vs. TDS2. This shift indicated that the logistic regression model is less prone to overfit than the random forest model, given that both modelling techniques was used on the same data. The shifts in both cases are below the threshold of 0.25. The calculations can be seen in Table 5 to Table 8.

In order to calculate a PSI for a model, the probability outcome of the training dataset should be compared against the probability outcome of the test data. The outcome of the training dataset must then be split in ten equal population sizes and the probability noted at each 10%. After the test dataset is run through the model, the population sizes need to be measured at the same probabilities as the training dataset. The population percentages are then populated into the formula of Eq. (37).

In order to understand the calculations in Table 5 to Table 8, Eq. (37) is broken down into portions:

$$PSI = \sum \left(\left(\frac{n_{di}}{N_d} \right) - \left(\frac{n_{vi}}{N_v} \right) \right) * \ln \left(\left(\frac{n_{di}}{N_d} \right) \Big/ \left(\frac{n_{vi}}{N_v} \right) \right) , \tag{40}$$

$$PSI = \sum (D_i - V_i) * \ln(D_i / V_i) \ . \tag{41}$$

Table 5: PSI – Random forest - Training vs. TDS1

Population Stability Index - Training vs TDS1					
	Training	TDS1	$D_i - V_i$	D_i/V_i	PSI
Band 1	10.30%	1.50%	0.088	6.867	0.16955
Band 2	10.00%	7.00%	0.030	1.429	0.01070
Band 3	9.90%	8.40%	0.015	1.179	0.00246
Band 4	9.60%	9.20%	0.004	1.043	0.00017
Band 5	9.30%	11.20%	-0.019	0.830	0.00353
Band 6	10.10%	11.90%	-0.018	0.849	0.00295
Band 7	10.80%	11.40%	-0.006	0.947	0.00032
Band 8	9.60%	12.70%	-0.031	0.756	0.00868
Band 9	10.40%	12.80%	-0.024	0.813	0.00498
Band 10	10.00%	14.10%	-0.041	0.709	0.01409
					0.22

Table 6: PSI – Random forest - Training vs. TDS2

Population Stability Index - Training vs TDS2					
	Training	TDS2	$D_i - V_i$	D_i/V_i	PSI
Band 1	10.30%	1.10%	0.092	9.364	0.20579
Band 2	10.00%	7.90%	0.021	1.266	0.00495
Band 3	9.90%	9.00%	0.009	1.100	0.00086
Band 4	9.60%	9.80%	-0.002	0.980	0.00004
Band 5	9.30%	10.30%	-0.010	0.903	0.00102
Band 6	10.10%	12.30%	-0.022	0.821	0.00434
Band 7	10.80%	11.50%	-0.007	0.939	0.00044
Band 8	9.60%	11.10%	-0.015	0.865	0.00218
Band 9	10.40%	13.20%	-0.028	0.788	0.00668
Band 10	10.00%	14.00%	-0.040	0.714	0.01346
Total					0.24

Table 7: PSI – Logistic regression - Training vs. TDS1

Population Stability Index - Training vs TDS1					
	Training	TDS1	$D_i - V_i$	D_i/V_i	PSI
Band 1	10.30%	2.00%	0.083	5.150	0.13604
Band 2	10.00%	7.60%	0.024	1.316	0.00659
Band 3	9.90%	8.90%	0.010	1.112	0.00106
Band 4	9.60%	9.40%	0.002	1.021	0.00004
Band 5	9.30%	10.70%	-0.014	0.869	0.00196
Band 6	10.10%	11.30%	-0.012	0.894	0.00135
Band 7	10.80%	12.00%	-0.012	0.900	0.00126
Band 8	9.60%	12.20%	-0.026	0.787	0.00623
Band 9	10.40%	13.50%	-0.031	0.770	0.00809
Band 10	10.00%	12.40%	-0.024	0.806	0.00516
Total					0.17

Table 8: PSI - Logistic regression - Training vs. TDS2

Population Stability Index - Training vs TDS2					
	Training	TDS2	$D_i - V_i$	D_i/V_i	PSI
Band 1	10.30%	2.80%	0.075	3.679	0.09769
Band 2	10.00%	5.50%	0.045	1.818	0.02690
Band 3	9.90%	7.60%	0.023	1.303	0.00608
Band 4	9.60%	9.60%	0.000	1.000	0.00000
Band 5	9.30%	9.70%	-0.004	0.959	0.00017
Band 6	10.10%	11.10%	-0.010	0.910	0.00094
Band 7	10.80%	13.10%	-0.023	0.824	0.00444
Band 8	9.60%	13.60%	-0.040	0.706	0.01393
Band 9	10.40%	16.00%	-0.056	0.650	0.02412
Band 10	10.00%	11.00%	-0.010	0.909	0.00095
Total					0.18

10.2. Gini coefficient comparison

In order to assess the predictive performance of the random forests technique, the Gini coefficient criterion was used (see Chapter 6.3). Furthermore, the performance of the random forest model was benchmarked against the Gini coefficient resulting from the conventional logistic regression model using the same set of customers, independent and dependent variables.

Figure 15a illustrates the Gini coefficient of both the random forest model and the logistic regression model on the training dataset. Figure 15b illustrates the Gini coefficient of the validation data for both the random forest and logistic regression models. The random forest model predicted attrition very well on the training dataset, but the Gini coefficient dropped quite significantly during validation, indicating overfitting. A model overfits when it predicts well on the training dataset, but loses predictability with the validation dataset. The validation dataset is a subset of the training dataset, eliminating the possibility that a population shift caused the loss in predictability. It can be observed in Figure 15a and Figure 15b that the area between the diagonal line and the Lorenz curve, (see Figure 10 in Chapter 6.3) shrunk from training to validation. This indicates a drop in Gini coefficient for the random forest model. The logistic regression model had a lower Gini coefficient in the training set, but did not lose predictive power during validation, indicated by only 0.3% drop in Gini coefficient value.

A further drop in the Gini coefficient was observed after running TDS1 and TDS2 through the random forest model, once again indicating that the random forest model overfitted during training. The logistic regression model did not lose as much predictive power, indicating a more robust modelling technique. Figure 15 and Figure 16 illustrate the change in Gini coefficient for both modelling techniques from training to TDS2. The Gini coefficient values are provided in Table 9.

Table 9: Gini coefficient values for both models after all datasets was scored through them

Model	Gini coefficient			
	Training	Validate	TDS1	TDS2
Random forest	41.4%	32.2%	23.0%	22.6%
Logistic regression	33.6%	33.3%	29.0%	28.0%

The Gini coefficient dropped by 18.1% (41.4% - 22.6%) from the training dataset to TDS2 when the data was modelled with the random forest model. In contrast, the Gini coefficient only dropped by 5.6% (33.6% - 28.0%) when using the logistic regression model. This demonstrates that the logistic regression model showcase more robustness and stability over time.

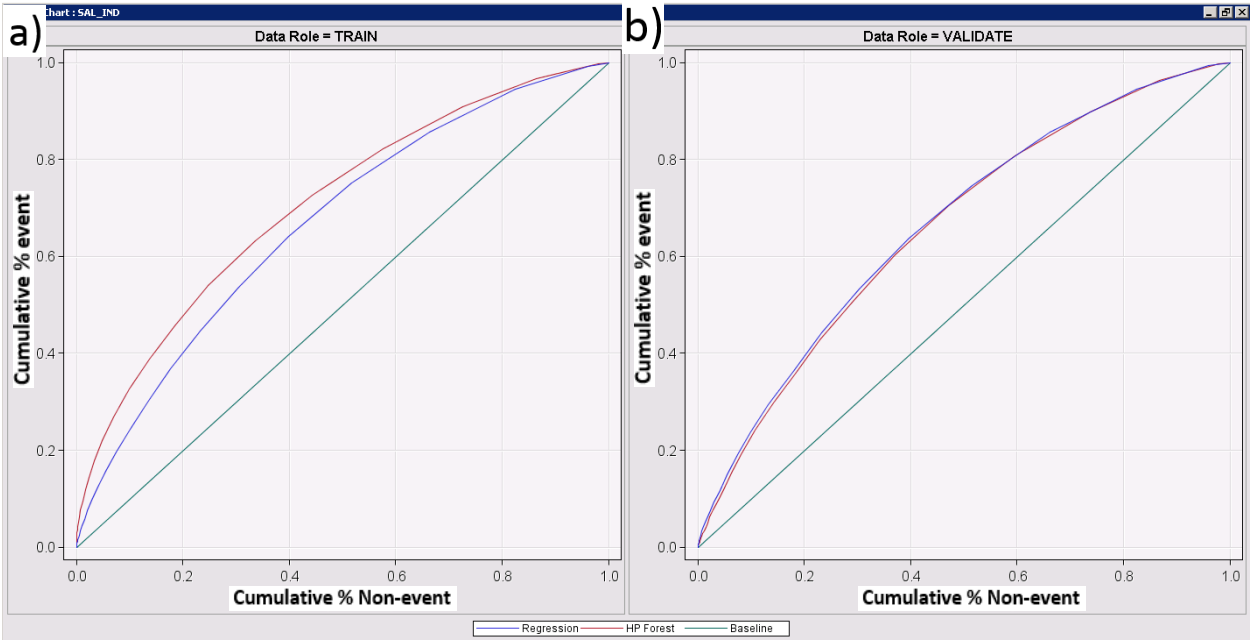


Figure 15: Screenshot of the random forest and logistic regression Gini coefficient graphs for the a) training dataset and b) validation dataset

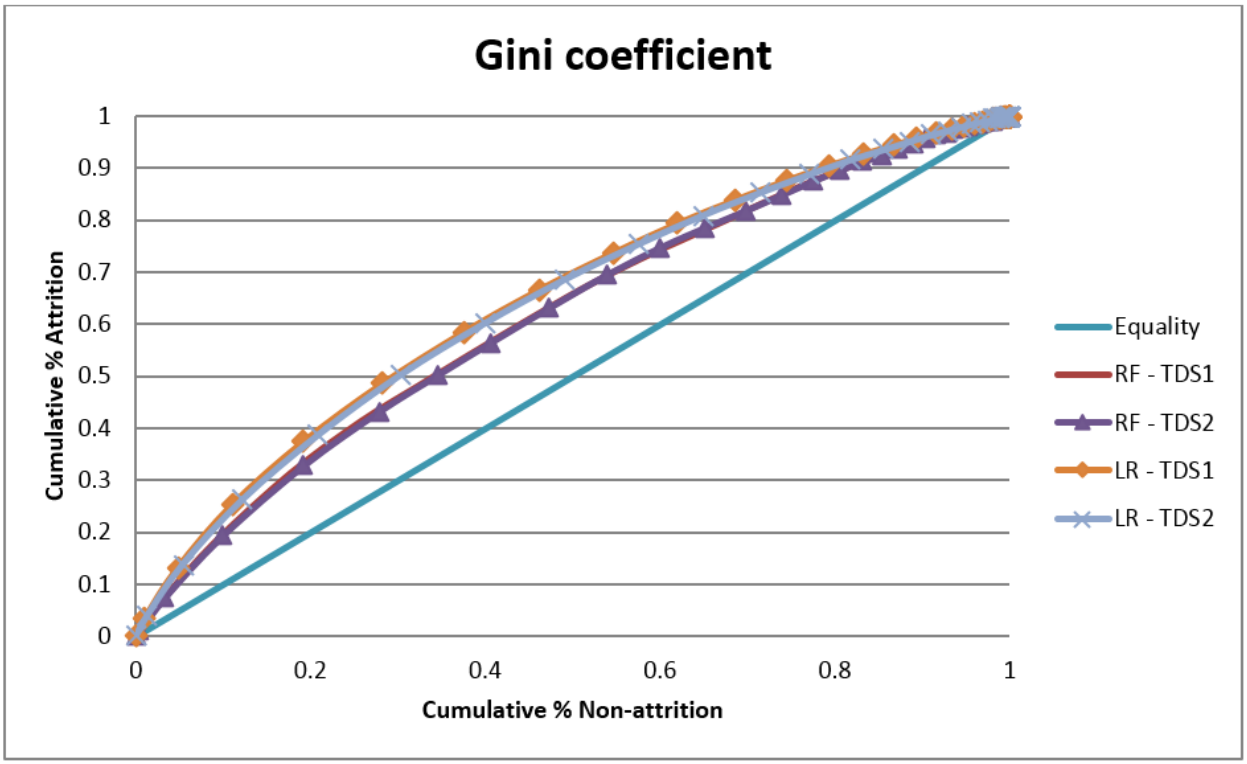


Figure 16: Random forest and logistic regression - TDS1 and TDS2 Gini coefficient graphs

10.3. Misclassification

The accuracy measures (ACC, TNR, TPR, Recall and Precision) derived from a confusion matrix (see Chapter 5.1.2) describe the performance of a classification model. It shows how the model misclassifies and is "confused" when making the predictions (Machine Learning Mastery, 2016).

The measures derived from the confusion matrices rendered thought-provoking results for the training data, TDS1 and TDS2. The accuracy measures can be seen in Table 10, Table 11 and Table 12. The accuracy measures were calculated at different probability cut-offs (0.1; 0.2; 0.3; 0.4; 0.5). If the assigned probabilities were above the cut-off rate, the outcome would be predicted as an attrition case or if the probability was lower than the cut-off rate, it would be assigned as a non-attrition case. The following is an example of how attrition and non-attrition cases were assigned based on the cut-off rate:

> If a record had a probability of 0.25, and the cut-off was set at 0.2, the predicted outcome was assigned as an attrition case, i.e. all cases with probability of higher than 0.2 would be designated as an attrition case. If the model assigned a probability of 0.18 and the cut-off was set to 0.2, the predicted outcome was assigned a non-attrition case, i.e. all cases with probability of lower than 0.2 would be assigned as a non-attrition case. The confusion matrices were drawn with the predicted outcome vs. the actual outcome and the measures calculated from the confusion matrices.

It can be seen that the ACC (see Chapter 5.1.1) dropped significantly for the random forest model for the training to TDS1 and TDS2 from 70%, 33% and 34% respectively at a probability cut-off of 0.1. The accuracy of the logistic regression model on a 0.1 probability cut-off also dropped, but not as severely as the random forest model: 81%, 69% and 71% respectively.

The random forest model, however, performed better than the logistic regression model for a probability cut-off of 0.2 in TDS1 and TDS2. The TPR, see Chapter 5.1.2.1, came to 30% for the random forest model and 6% for the logistic regression model on TDS1 and 28% and 5% respectively for TDS2. This indicates that at a cut-off rate of 20%, the random forest has the ability to identify the positive (attrition) cases accurately.

The sampling method applied caused the random forest and logistic regression models to adjust the predicted probabilities downwards in a linear fashion with a fixed factor. The factor of adjustment increases, as the attrition cases proportion of the population decrease, i.e. the smaller the true attrition proportion, the greater the factor and the smaller the predicted probabilities are adjusted.

The downwards adjustment was more severe for the logistic regression model, adjusting the probabilities very low, regardless if the actual outcome was attrition or none-attrition. The adjustment was not as severe for the random forest model, indicating that it separated the two outcomes better. This also indicates the superiority of the random forest model over the logistic regression model in classifying rare events.

Training dataset

Table 10: Accuracy measures on the training dataset. RF – random forest; LR – logistic regression

Probability cut-off	0.1		0.2		0.3		0.4		0.5	
	RF	LR	RF	LR	RF	LR	RF	LR	RF	LR
Accuracy (ACC)	70%	81%	90%	93%	93%	94%	94%	94%	94%	94%
Specificity (TNR)	71%	85%	95%	99%	99%	100%	100%	100%	100%	100%
Sensitivity (TPR)	50%	33%	13%	3%	3%	0%	0%	0%	0%	0%
Recall	50%	33%	13%	3%	3%	0%	0%	0%	0%	0%
Precision	10%	13%	15%	23%	20%	26%	27%	50%	51%	52%

TDS1

Table 11: Accuracy measures on TDS1. RF – random forest; LR – logistic regression

Probability cut-off	0.1		0.2		0.3		0.4		0.5	
	RF	LR	RF	LR	RF	LR	RF	LR	RF	LR
Accuracy (ACC)	33%	69%	76%	92%	90%	94%	93%	94%	94%	94%
Specificity (TNR)	30%	70%	79%	98%	95%	100%	99%	100%	100%	100%
Sensitivity (TPR)	80%	49%	30%	6%	8%	1%	2%	0%	0%	0%
Recall	80%	49%	30%	6%	8%	1%	2%	0%	0%	0%
Precision	7%	10%	9%	16%	10%	23%	11%	50%	11%	71%

TDS2

Table 12: Accuracy measures on TDS2. RF – random forest; LR – logistic regression

Probability cut-off	0.1		0.2		0.3		0.4		0.5	
	RF	LR	RF	LR	RF	LR	RF	LR	RF	LR
Accuracy (ACC)	34%	71%	78%	93%	91%	94%	93%	94%	94%	94%
Specificity (TNR)	31%	72%	81%	98%	96%	100%	99%	100%	100%	100%
Sensitivity (TPR)	78%	47%	28%	5%	7%	0%	1%	0%	0%	0%
Recall	78%	47%	28%	5%	7%	0%	1%	0%	0%	0%
Precision	7%	10%	9%	15%	10%	23%	10%	47%	9%	94%

10.4. Variable discussion

The most influential and predictive variables are discussed in this section. It was interesting to note how past customer behaviour has a significant impact on the likelihood that a customer will churn or not. The most predictive variable was the customer's propensity to borrow. Customers who had a high likelihood to borrow were more prone to attrite. In a credit hungry economic environment where the general population and especially the income segment upon which the model was built, it made sense that customers churn more if their propensity to borrow is high. In the banking industry, it is referred to as shopping for credit (Clements, 2015), where people are willing to defect to another financial services provider when they are granted credit elsewhere.

The number of years that the bank was the customer's primary bank of choice using a direct deposit account was a very predictive variable. It shows that the longer a customer was a client of the bank, the less likely they were to defect. This indicates that it is of utmost importance to nurture relationships with customers, especially in the early stages of joining the bank. Customers who are pleased with the services rendered to them by the bank are not likely to seek services elsewhere, thus resulting in a longer relationship between the customer and the bank. The longer a customer had and serviced an ending product, like a personal loan, at the bank showed the same trend. The same could be said of the variable that portrays the customer's total relationship age.

The credit score of the customer was also found to have great influence on customer retention. Customers with better credit records are less likely to shop around for loans and obtain loans with very high interest rates. Credit shoppers tend to default often on the high-interest loans. The three-month rolling income into the primary bank account attribute proved to be a very good indication of attrition, showing that once it starts to decay, the client is in the process of moving their funds elsewhere.

Thirty variables were selected in the model building process of the random forest model by SAS Enterprise Miner, of which twenty-eight variables were included in the logistic regression model. The logistic regression model only chose twenty-eight variables as the remaining two variables did not add any predictive power to the model.

Table 13 below sets out the Gini coefficient and IV (see Chapter 6.1, Eq. (36) and Eq. (38)) for each variable.

Table 13: Gini coefficient and IV of variables

Gini Coefficient	IV	Variable description
21.869	0.154	Propensity to borrow
17.911	0.104	Customer relationship age
17.908	0.104	Age of primary direct deposit account
17.868	0.104	Maximum age of direct deposit account
17.817	0.103	Age of oldest loan relationship
17.814	0.105	Credit score
17.754	0.108	Three months' rolling income into primary account
16.804	0.093	Propensity to pay
13.828	0.064	Bank's credit score
13.802	0.063	Early default indicator
13.433	0.062	Expected default frequency percentage
13.114	0.06	Average credit turnover primary direct deposit account, final quarter relative to fifth quarter
12.499	0.05	Age of credit card account
12.49	0.053	The sum of the last three-monthly credit turnovers
12.436	0.058	The average credit turnover of the primary DDA account over the quarter as a percentage of the minimum balance of the primary DDA account over the quarter

11.62	0.046	The sum of the month-end balances for savings accounts in the first month relative to the average total credit turnover over the first quarter for all direct deposit accounts
11.282	0.046	Sum of monthly positive credit cashflows
11.232	0.051	Recommended overdraft limit
11.232	0.051	Credit turnover index
11.035	0.042	Sum of monthly positive credit cashflows ratio
11.034	0.049	The average minimum monthly balance for all DDA accounts over the first quarter relative to the average credit turnover for all DDA accounts over the first quarter
10.872	0.048	Tolerance limit
10.828	0.041	Number of monthly positive cash flow ratio
10.737	0.047	Count of all monthly positive and negative cashflow
10.671	0.039	Sum of quarterly positive cashflows
10.666	0.042	The total unsecured lending relative to the total unsecured limit expressed as a percentage
10.55	0.039	Relative change in utilisation of limits for all direct deposit accounts
10.518	0.037	Direct deposit account - Number of returned items last 6 months
10.252	0.035	Volatility of primary direct deposit account in the last 12 months
10.01	0.039	The maximum balance of the primary DDA account over the quarter

Twenty-three of the variables were weak predictors and seven were medium predictors, according to the IV thresholds discussed in Chapter 6.1, Table 3. It is advised to use predominantly medium strength predictors when building a model, but given the data sources available, weak predictors had to be included as well to ensure a wide variety of variables. The twenty-three weak predictors were also included as they still hold predictive power and a Gini of greater than 10 as discussed in Chapter 9.3.

10.5. Variable stability

In order to illustrate variable stability, the population distribution among the groups within each variable were plotted over the observation period. The sum of the distribution across all the groups in the variable added up to 100% for each observation month.

The variable stability for the eight (seven medium predictors and the strongest weak predictor) most influential variables are depicted in Figure 17 to Figure 24. Variable stability is graphically illustrated by showing the population distribution across the groups within the variables for the observation months of the training and test datasets (February 2016 to August 2017).

Some instability can be observed in Figure 19 and Figure 23 for the age of primary direct deposit account and three months' rolling income variables, but is not significant and stable over time. The two variables added predictive power, which gave reason to keep them in the models

The cause of the shift in the population distribution of the models can be seen in Figure 18 and Figure 21 on a variable level. The reason for this drastic shift in population between the two groups ([.,-91919] and [0-1097]) in each of the variables is due to a change in the code that generates the data for the behavioural data warehouse. The special code "-91919" was erroneously assigned to customers under certain conditions. This was amended by assigning a "0" rather than a "-91919", which reduced the number of cases that fell in the [.,-91919] group and shifted the volumes to the group that contains zeroes. The data prior to this point remained the same and only new data from June 2017 going forward will be affected. As it was unclear which data was erroneously given a value of -91919, the data was left as-is. The majority of the variables has a group that contains special values and are illustrated since these special values will also be present after implementation, i.e., the special values within the variables are recognised as valid values and could not be omitted. The special values are: -997, -91919 and -92929.

The legend on the right-hand side of Figure 17 to Figure 24 shows the groups for each variable and the population size of each group per month is depicted on the vertical axis. The observation months are displayed on the horizontal axis.

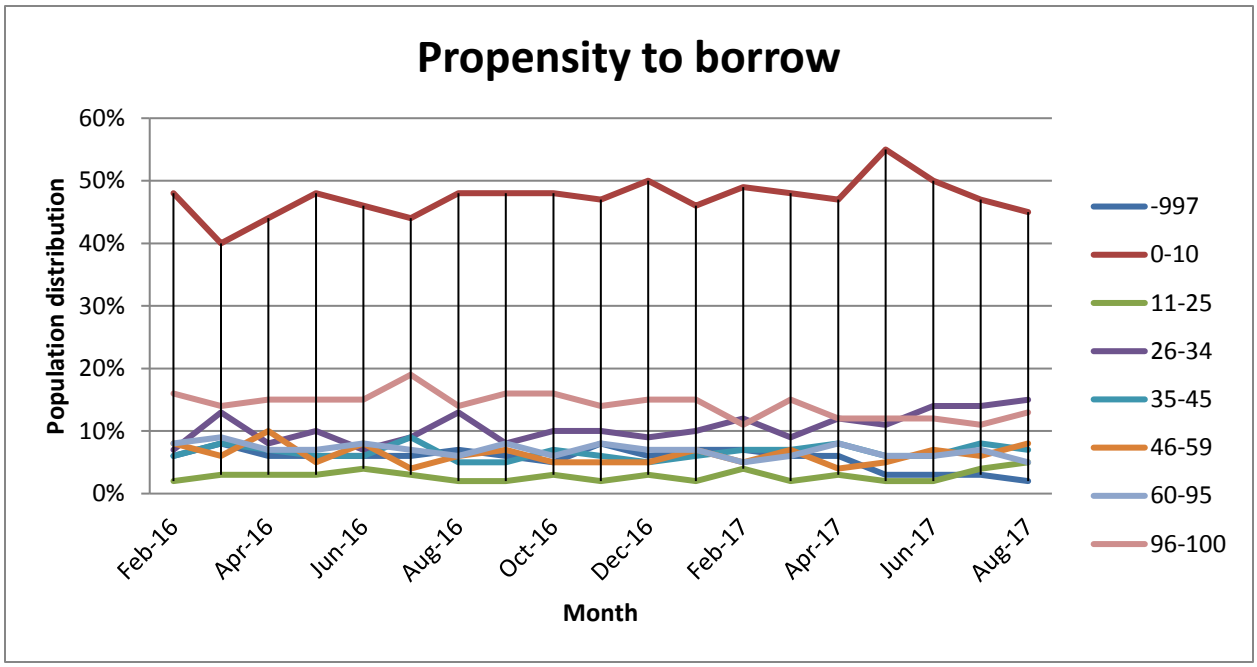


Figure 17: Variable stability - Propensity to borrow

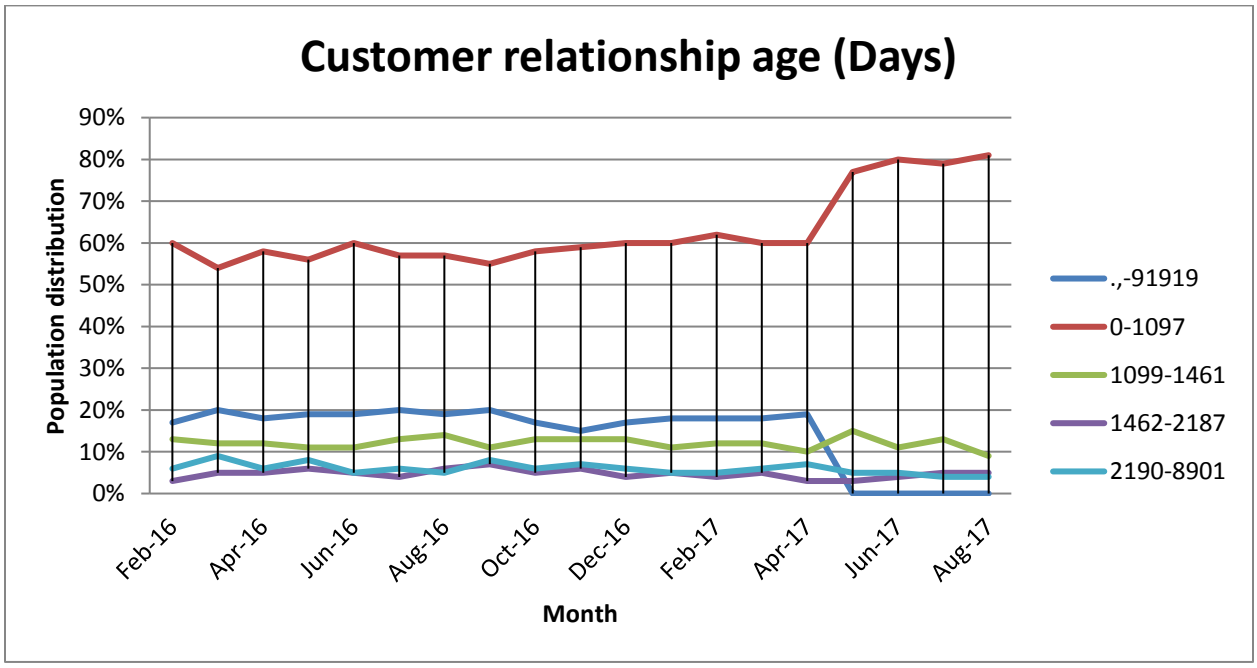


Figure 18: Variable stability - Customer relationship age

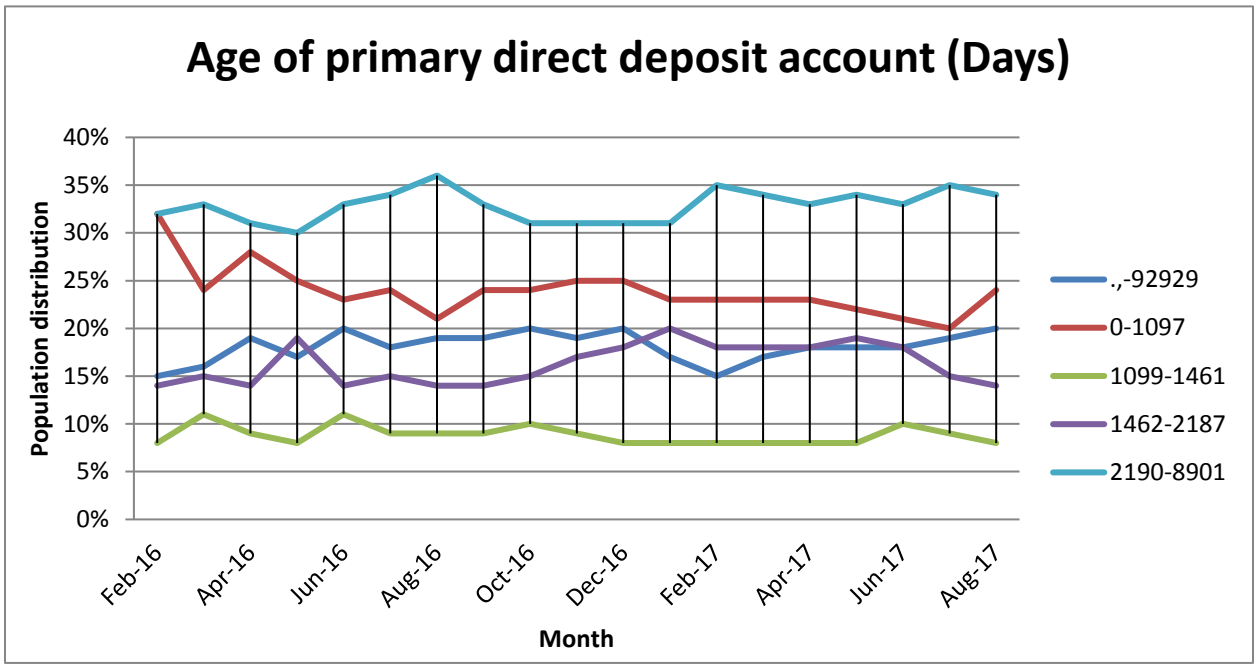


Figure 19: Variable stability - Age of primary direct deposit account

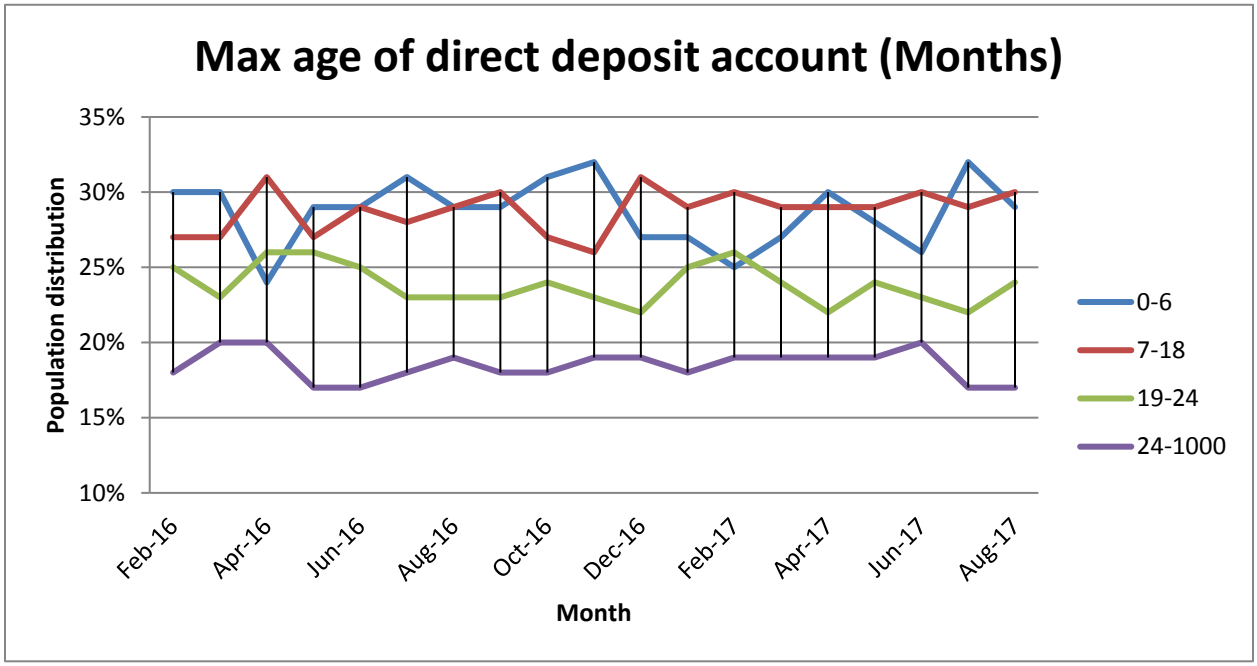


Figure 20: Variable stability - Max age of direct deposit account

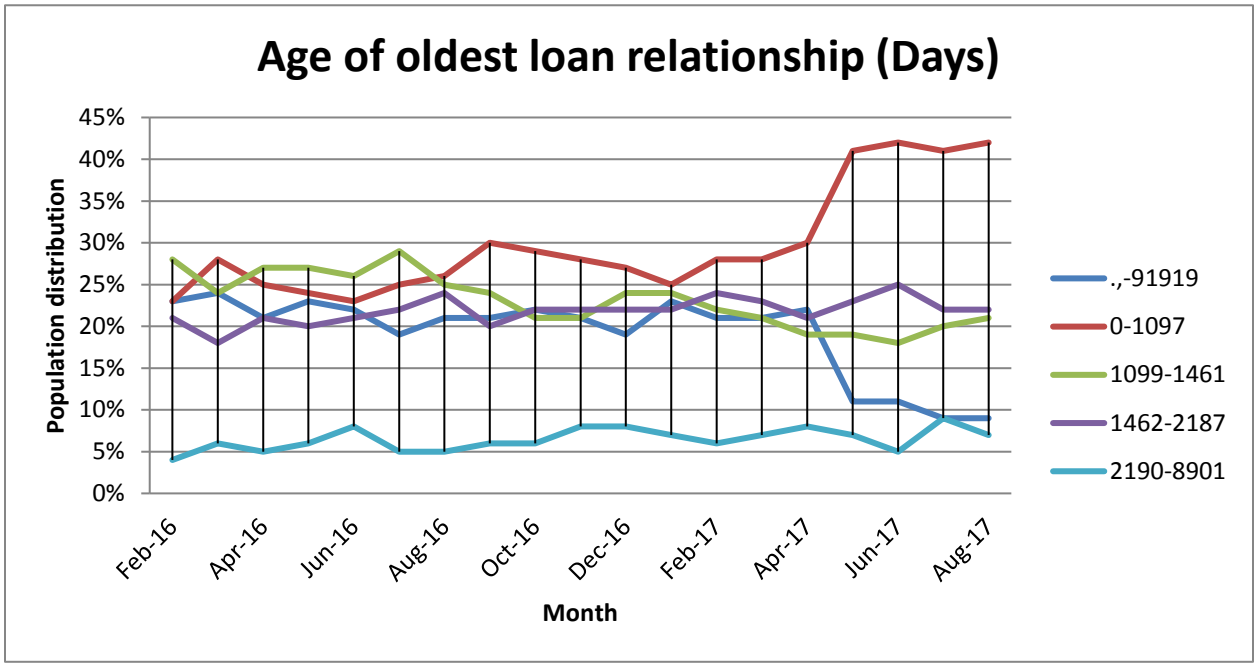


Figure 21: Variable stability - Age of oldest loan relationship

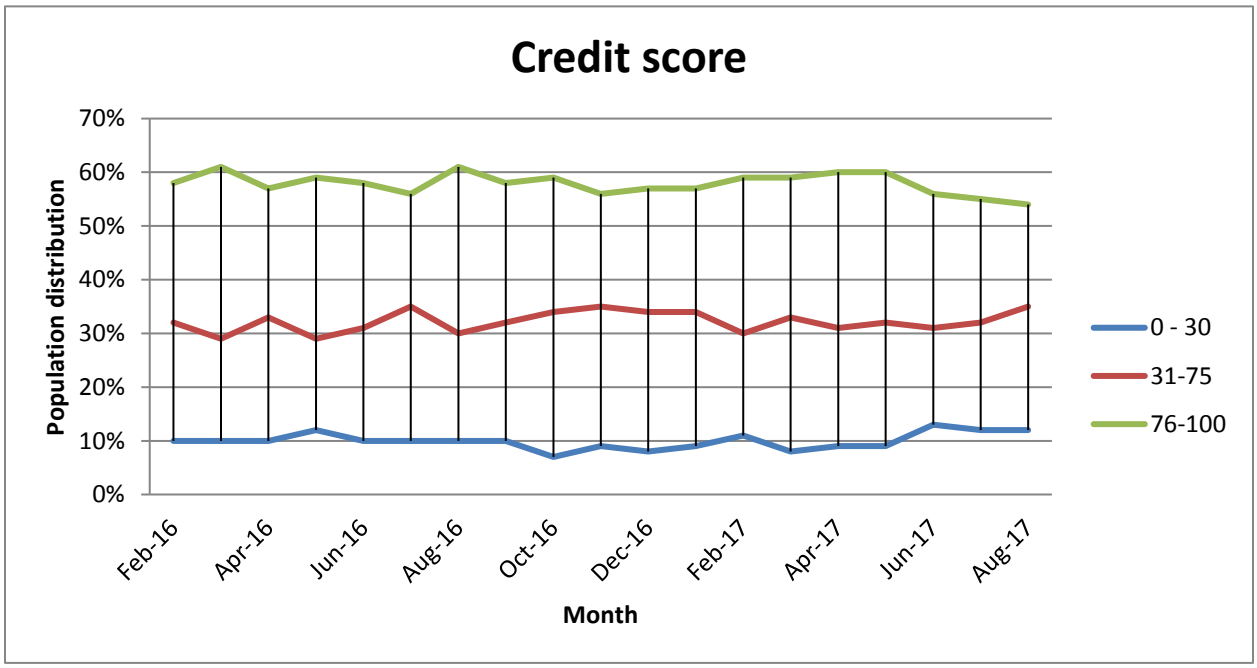


Figure 22: Variable stability - Credit score

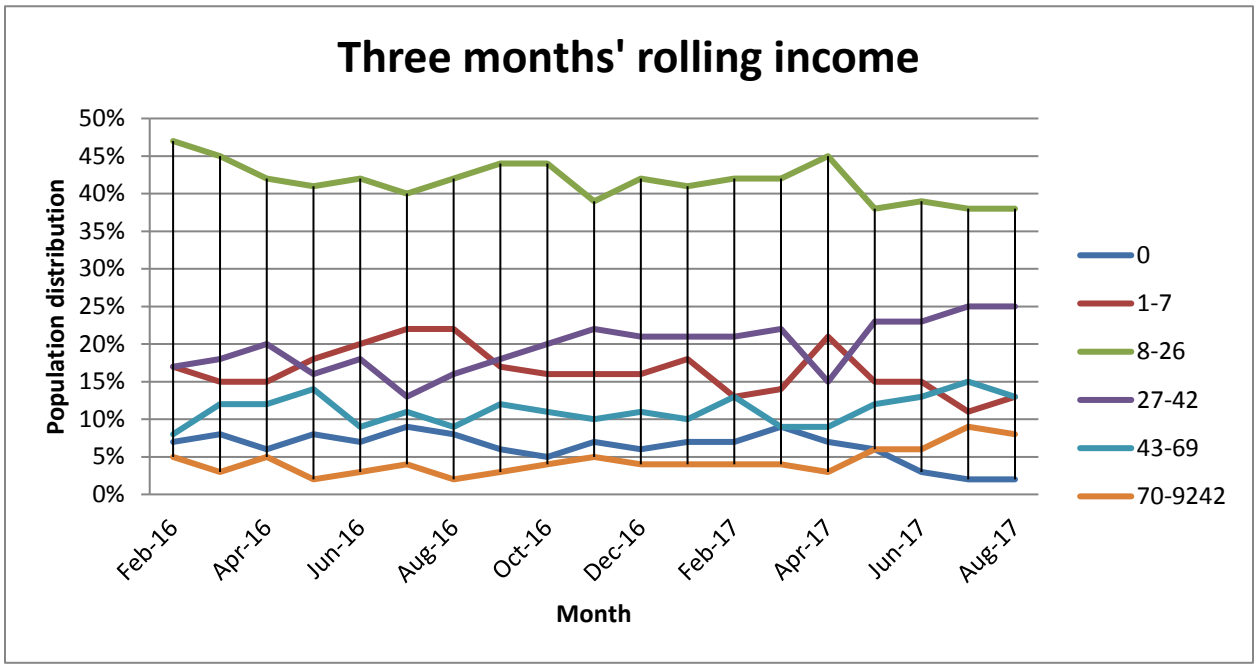


Figure 23: Variable stability - Three months' rolling income

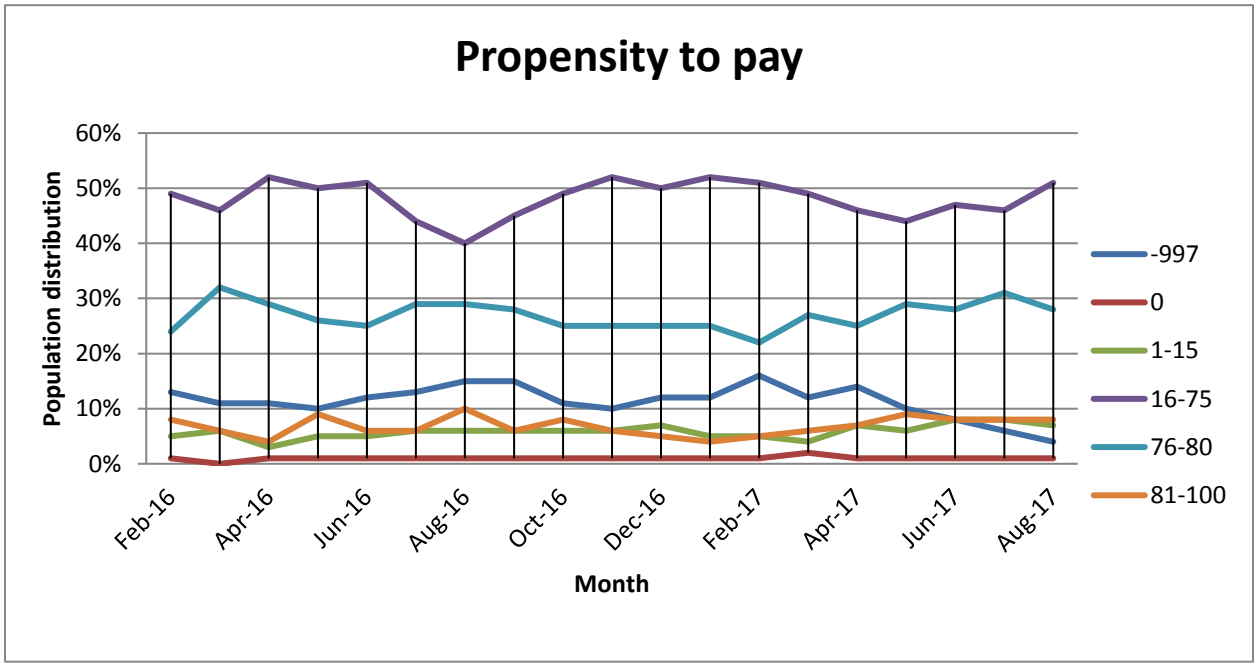


Figure 24: Variable stability - Propensity to pay

10.6. Histogram

The probability distributions of both models can be seen as histograms illustrated in Figure 25 below. The probability distribution functions of the random forest model did not shift from the training dataset to TDS1 and TDS2, which are depicted in Figure 25a, Figure 25c and Figure 25e respectively. The probability distribution functions of the logistic regression model also did not shift from the training dataset to TDS1 and TDS2, as indicated in Figure 25b, Figure 25d and Figure 25f respectively. These histograms indicate stability and predictiveness over time as both models' probability distributions did not shift from the training dataset to the test datasets.

There was a population shift in two of the variables, as discussed in Chapter 10.5. This caused the dip in the probability distribution function of the random forest model in TDS1 and TDS2 at a probability of 0.1, evident in Figure 25c and Figure 25e. The variables customer relationship age and age of oldest loan relationship were however retained as they were predictive, despite the shift.

The probability distribution functions of the random forest model were more widely spread between 0 and 0.4 than the values of the logistic regression model, indicating that the random forest model is superior in separating rare events.

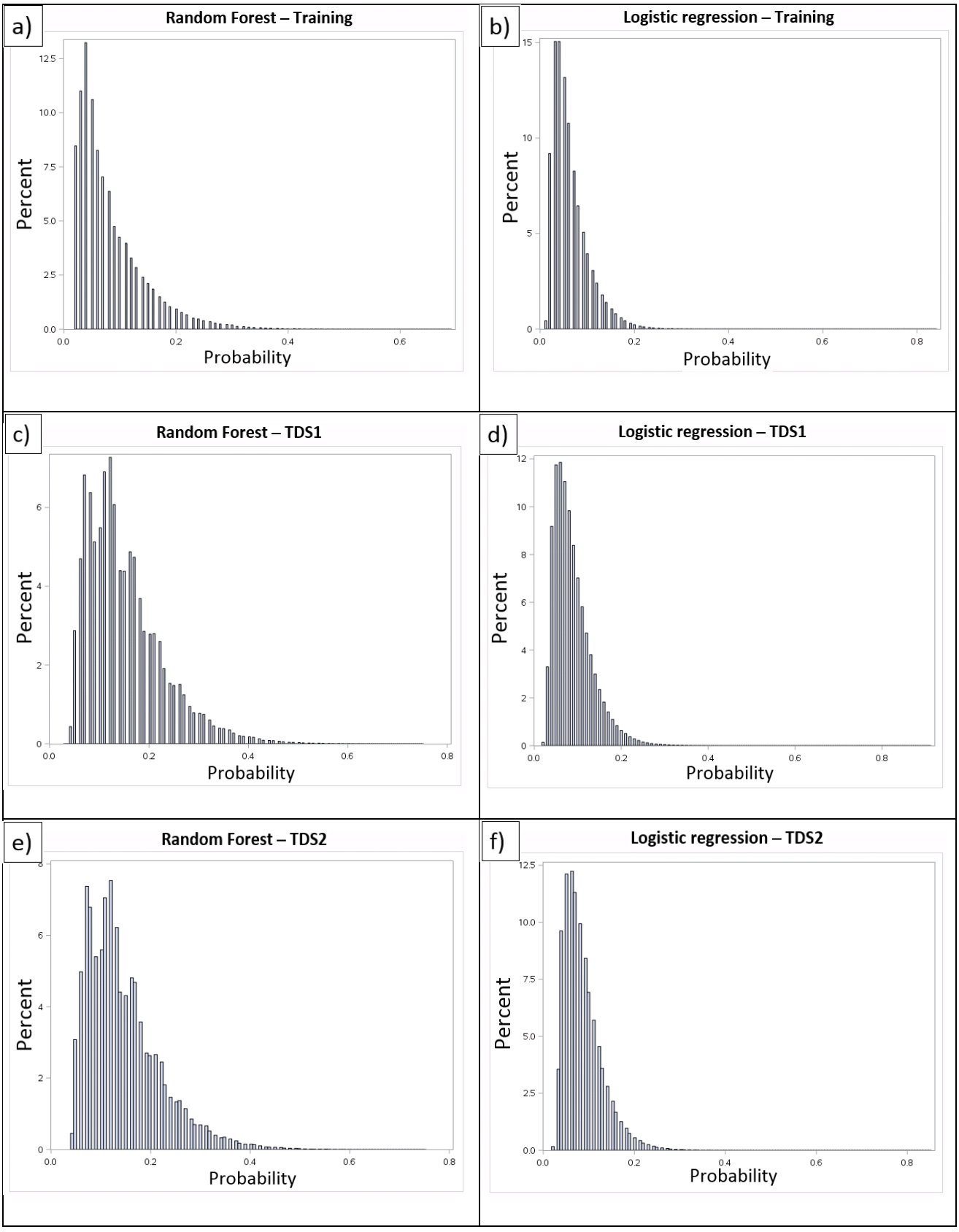


Figure 25: Probability distributions for the random forest model for the a) training, c) TDS1 and e) TDS2 dataset and the probability distributions for the logistic regression model for the b) training, d) TDS1 and f) TDS2 dataset

11. Conclusions and recommendations

The findings with respect to the TPR on TDS1 and TDS2 strongly indicates that the random forest model is a viable modelling technique to predict customer attrition, even though it overfitted on the training set. The logistic regression and random forest models will perform very similar to TDS1 and TDS2 in a production environment, after implementation, as the observation periods are more recent. The model will act as a ranking tool (rank cases from highest attrition probability to lowest attrition probability) and the model has not lost ranking ability, which makes it a viable option.

The logistic regression model however showed better robustness and stability during testing, which is important. The logistic regression model has higher Gini coefficients than the random forest model in TDS1 and TDS2, but Gini coefficient is not the only measure that should be considered when evaluating model performance.

The random forest model illustrated that it predicts rare events well at a probability cut-off rate of 0.2. The random forest model will therefore be the preferred model to predict attrition, despite the shift in stability for the two test datasets as shown in Figure 13.

Future analyses should test additional data sources and test the viability of non-linear techniques, such as rough data modelling and genetic programming, which Kowalczyk *et al.* (1999) found to perform better than linear models with respect to predicting customer retention or customer churn.

12. References

Ahn, H., Moon, H., Fazzari, M.J., Lim, N., Chen, J.J. & Kodell, R.L. 2007. Classification by ensembles from random partitions of high-dimensional data. *Computational statistics and data analysis,* 51(12):6166-6179.

Amit, Y. & Geman, D. 1997. Shape quantization and recognition with randomized trees. *Neural computation,* 9(7):1545-1588.

Anderson, E.W., Fornell, C. & Lehmann, D.R. 1994. Customer satisfaction, market share, and profitability: findings from Sweden. *The journal of marketing,* pp.53-66.

Barros, R.C., de Carvalho, A.C. & Freitas, A.A. 2015. Automatic design of decision-tree induction algorithms. Springer International Publishing.

Batista, G.E., Prati, R.C. & Monard, M.C. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM Sigkdd explorations newsletter,* 6(1):20-29.

Bitner, MJ, 1990. Evaluating Service Encounters: The Effects of Physical Surroundings and Employee Responses. *Journal of Marketing*, [Online]. 54, 69-82. Available at: http://www.jstor.org/stable/1251871 [Accessed 1 October 2007].

Breiman, L. 2001. Random forests. *Machine learning*, 45(1):5-32.

Breiman, L. 1996. Bagging predictors. *Machine learning*, 24(2):123-140.

Bright Hub Project Management. 2011. Disadvantages to Using Decision Trees. [ONLINE] Available at: https://www.brighthubpm.com/project-planning/106005-disadvantages-to-using-decision-trees/. [Accessed 23 May 2018].

Buckinx, W. & Van den Poel, D. 2005. Customer base analysis: partial defection of behaviourally loyal customers in a non-contractual FMCG retail setting. *European journal of operational research,* 164(1), pp.252-268.

Burges, C.J. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery,* 2(2):121-167.

Chakrabarty, A. 2004. Barking up the wrong tree-factors influencing customer satisfaction in retail banking in the UK. *International journal of applied marketing,* 3:39-57.

Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research,* 16, :321-357.

Cheng, M.Y., Hoang, N.D. and Chang, N.W., 2013. Bayesian classifier with K-Nearest Neighbor density estimation for slope collapse prediction. In ISARC 2013-30th International Symposium on Automation and Robotics in Construction and Mining, Held in Conjunction with the 23rd World Mining Congress.

Chu, B.H., Tsai, M.S. & Ho, C.S. 2007. Toward a hybrid data mining model for customer retention. *Knowledge-based systems*, 20(8):703-718.

Clements, N. 2015. How to shop for credit without hurting your credit score. *Forbes,* 1. 28 April 2015.

Cortes, C. & Vapnik, V. 1995. Support-vector networks. *Machine learning,* 20(3):273-297.

De Maesschalck, R., Jouan-Rimbaud, D. & Massart, D.L. 2000. The mahalanobis distance. *Chemometrics and intelligent laboratory systems,* 50(1):1-18.

DeLong, E.R., DeLong, D.M. & Clarke-Pearson, D.L. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pp.837-845.

Deza M.M., Deza, E., 2009. Encyclopedia of distances. In Encyclopedia of Distances (pp. 1-583). 1st ed. Berlin, Heidelberg: Springer.

Dietterich, T.G. 2000. Ensemble methods in machine learning. *Multiple classifier systems,* 1857:1-15.

Drummond, C. & Holte, R.C. 2003, August. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Workshop on learning from imbalanced datasets II (Vol. 11). Washington DC: Citeseer.

Duda, R.O., Hart, P.E. & Stork, D.G., 2012. Pattern classification. 2nd ed. New York, United States of America: John Wiley and Sons.

Dudoit, S., Fridlyand, J. & Speed, T.P. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association,* 97(457):77-87.

Duin, R.P. & Tax, D.M. 2000 June. Experiments with classifier combining rules. In International Workshop on Multiple Classifier Systems. Springer, Berlin, Heidelberg. pp. 16-29.

EFL Global. 2015. Every lender has a Gini Coefficient, so what exactly is it?. [ONLINE] Available at: https://www.eflglobal.com/every-lender-has-a-gini-coefficient-so-what-exactly-is-it/#_ftn1. [Accessed: 28 Apr. 2018]

Elkan, C. 2001, August. The foundations of cost-sensitive learning. In International joint conference on artificial intelligence, 17(1):973-978). Lawrence Erlbaum Associates Ltd.

Estabrooks, A., Jo, T. & Japkowicz, N. 2004. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence,* 20(1):18-36.

Eva Ascarza (2018) Retention Futility: Targeting High-Risk Customers Might Be Ineffective. *Journal of Marketing Research*: February 2018, 55(1):80-98

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters,* 27(8):861-874.

Fletcher, R., 1987. Practical methods of optimization. 1st ed. New York: John Wiley and Sons.

Frank, E. & Hall, M. 2001. A simple approach to ordinal classification. *Machine Learning*: ECML 2001, pp.145-156.

Frederick, F.R. & Sasser, W.E. 1990. Zero defections: quality comes to services. *Harvard business review,* 68(5):105.

Friedman, J., Hastie, T. and Tibshirani, R., 2001. The elements of statistical learning (Vol. 1, pp. 337-387). New York: Springer series in statistics.

Gayathry, S. 2016. Customer relationship management model for banks. *Journal of Internet banking and commerce,* 21(S5):1.

Hart, P., 1968. The condensed nearest neighbor rule (Corresp.). IEEE transactions on information theory, 14(3), pp.515-516.

Hanley, J.A. & McNeil, B.J. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology,* 143(1):29-36.

Hansen, L.K. & Salamon, P. 1990. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence,* 12(10):993-1001.

Hassibi, K. 2000. Business applications of neural networks, Singapore-New Jersey-London-Hon Kong. *World scientific,* (9):141-158.

He, H. & Garcia, E.A. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering,* 21(9):1263-1284.

Hennig-Thurau, T. & Klee, A. 1997. The impact of customer satisfaction and relationship quality on customer retention: a critical reassessment and model development. *Psychology and marketing,* 14(8):737-764.

Holte, R.C., Acker, L. and Porter, B.W., 1989, August. Concept Learning and the Problem of Small Disjuncts. In IJCAI 89, :813-818).

Hubert-Moy, L., Cotonnec, A., Le Du, L., Chardin, A. & Pérez, P. 2001. A comparison of parametric classification procedures of remotely sensed data applied on different landscape units. *Remote sensing of environment,* 75(2):174-187.

Hunt, H.K., 1977. Conceptualization and measurement of consumer satisfaction and dissatisfaction. 1st ed. Massachusetts, United States: Marketing Science Institute.

Ishwaran, H., Blackstone, E.H., Pothier, C.E. & Lauer, M.S. 2004. Relative risk forests for exercise heart rate recovery as a predictor of mortality. *Journal of the American statistical association,* 99(467):591-600.

Jiang, H., Deng, Y., Chen, H.S., Tao, L., Sha, Q., Chen, J., Tsai, C.J. & Zhang, S. 2004. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC bioinformatics,* 5(1):81.

Kaplan, E.L. & Meier, P. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association,* 53(282):457-481.

Kotler, P. & Armstrong, G., 1996. *Principles of marketing*. 7th ed. Prentice Hall. Englewood Cliffs NJ: Simon and Schuster.

Kotler, P., 1994. *Market segmentation analysis, planning, implementation and control.* 8th ed. Prentice Hall. New Jersey. US: Macmillan Publishers.

KPMG. 2015. Next generation banking survey. [ONLINE] Available at: https://assets.kpmg.com/content/dam/kpmg/pdf/2015/10/Next-Generation-Banking-Survey.pdf [Accessed: 1 Oct. 2018].

Kowalczyk, A.E.T.E.W. & Slisser, F. 1999. Modelling customer retention with statistical techniques, rough data models, and genetic programming. Rough fuzzy hybridization: a new trend in decision-making. p. 330.

Kubat, M. & Matwin, S. 1997, July. Addressing the curse of imbalanced training sets: one-sided selection. In ICML 97, :179-186.

Lambert, P.J. & Aronson, J.R. 1993. Inequality decomposition analysis and the Gini coefficient revisited. *The economic journal,* 103(420):1221-1227.

Larivière, B. and Van den Poel, D. 2005. Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert systems with applications,* 29(2):472-484.

Larivière, B. & Van den Poel, D. 2004. Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: the case of financial services. *Expert systems with applications,* 27(2):277-285.

Laurikkala, J. 2001. Improving identification of difficult small classes by balancing class distribution. *Artificial intelligence in medicine,* 2101(6):63-66.

Lending times. 2016. How to measure quality of underwriting: the Gini coefficient. [ONLINE] Available at: https://lending-times.com/2016/03/23/how-to-measure-quality-of-underwriting-the-gini-coefficient/ [Accessed: 23 Mar. 2018].

Lewis, B.R. & Bingham, G.H. 1991. The youth market for financial services. *International journal of bank marketing,* 9(2):3-11.

Liu, X. Y., Wu, J. & Zhou, Z. H, 2009. Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39(2):539--550.

Luo, T., Kramer, K., Goldgof, D.B., Hall, L.O., Samson, S., Remsen, A. & Hopkins, T. 2004. Recognizing plankton images from the shadow image particle profiling evaluation recorder. *IEEE transactions on systems, man, and cybernetics, part B (cybernetics),* 34(4):1753-1762.

Mac Namee, B., Cunningham, P., Byrne, S. and Corrigan, O.I., 2002. The problem of bias in training data in regression problems in medical decision support. Artificial intelligence in medicine, 24(1):51-70.

Machine Learning Mastery. 2016. What is a confusion matrix in machine learning. [ONLINE] Available at: https://machinelearningmastery.com/confusion-matrix-machine-learning/ [Accessed: 18 Nov. 2018]

Mani, I. & Zhang, I. 2003, August. kNN approach to unbalanced data distributions: a case study involving information extraction. In Proceedings of workshop on learning from imbalanced datasets (Vol. 126).

Mansour, Y. 1997, July. Pessimistic decision tree pruning based on tree size. In machine learning-international workshop then conference- (pp. 195-201). Morgan Kaufmann publishers, inc.

MathWorks. 2016. Detector performance analysis using ROC curves - MATLAB and Simulink example. [ONLINE] Available at: https://nl.mathworks.com/help/phased/examples/detector-performance-analysis-using-roc-curves.html [Accessed: 12 Jul. 2018].

McCullough, W., Maguire, B., Maguire, K., Goldberg, R. & Goldberg, M. OC Concepts, Inc., 2010. Customer information system. U.S. Patent 7,684,550.

McNeal, J., 1999. *The kids market: Myths and realities.* 1st ed. USA: Paramount Market Publishing.

Mease, D., Wyner, A.J. & Buja, A. 2007. Boosted classification trees and class probability/quantile estimation. *Journal of machine learning research,* 8(3):409-439.

Menard, S. 2018. Applied logistic regression analysis (Vol. 106). SAGE publications.

Mohri, M., Rostamizadeh, A. and Talwalkar, A., 2012. *Foundations of machine learning*. 1st ed. Cambridge, Massachusetts: MIT press.

Morwitz, V.G. and Schmittlein, D., 1992. Using segmentation to improve sales forecasts based on purchase intent: Which" intenders" actually buy?. *Journal of marketing research*, pp.391-405.

Murdoch, W.W. and Oaten, A., 1975. Predation and population stability. *In Advances in ecological research* (9), pp. 1-131. Academic Press.

Newman, J.W. and Werbel, R.A., 1973. Multivariate analysis of brand loyalty for major household appliances. *Journal of marketing research*, 10(4):404-409.

Nguyen, T.H., Sherif, J.S. & Newby, M. 2007. Strategies for successful CRM implementation. *Information management and computer security,* 15(2):102-115.

Oliver, R.L. & Bearden, W.O. 1985. Disconfirmation processes and consumer evaluations in product usage. *Journal of business research,* 13(3):235-246.

Oliver, R.L. and Swan, J.E., 1989. Consumer perceptions of interpersonal equity and satisfaction in transactions: a field survey approach. *The Journal of Marketing*, 54(2):21-35.

Oliver, R.L., 1980. A cognitive model of the antecedents and consequences of satisfaction decisions. *Journal of marketing research*, 17(4):460-469.

Opitz, D.W. & Maclin, R. 1999. Popular ensemble methods: an empirical study. *J. Artif. Intell. Res.(JAIR)*, 11:169-198.

Pfeifer, P.E. 2005. The optimal ratio of acquisition and retention costs. *Journal of targeting, measurement and analysis for marketing,* 13(2):179-188.

Pfeifer, P.E., Haskins, M.E. and Conroy, R.M., 2005. Customer lifetime value, customer profitability, and the treatment of acquisition spending. *Journal of managerial issues*, 17(1):11-25.

Polikar, R. 2006. Ensemble based systems in decision making. *IEEE circuits and systems magazine,* 6(3):21-45.

Powers, D.M. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.

Provost, F. & Domingos, P., 2001. *Well-Trained PETs: Improving Probability Estimation Trees*. 1st ed. Stern School of Business: NY, NY, 10012.

Provost, F., 2000, July. Machine learning from imbalanced data sets 101. In Proceedings of the AAAI'2000 workshop on imbalanced data sets (pp. 1-3).

Quinlan, J.R. 1987. Simplifying decision trees. *International journal of man-machine studies,* 27(3):221-234.

Quinlan, J.R., 1993. C4. 5: *Programming for machine learning*. Morgan Kauffmann, 38, :48.

RedStarKim. 2015. Psychology and business communication: an introduction to transactional analysis (TA). [ONLINE] Available at: http://www.kimtasso.com/psychology-business-communication-introduction-transactional-analysis-ta/ [Access: 9 Oct. 2018].

Reichheld, FF and Sasser, WE, 1990. Zero defections: quality comes to services. *Harvard business review*, [Online]. 68(5), 105-111. Available at: http://europepmc.org/abstract/MED/10107082 [Accessed 17 January 2009].

Reichheld, F., 1996. The quest for loyalty: creating value through partnership. 1st ed. Harvard Business Press: Massachusetts, United States.

Reinartz, W.J. & Kumar, V. 2000. On the profitability of long-life customers in a noncontractual setting: an empirical investigation and implications for marketing. *Journal of marketing,* 64(4):17-35.

SAS Institute Inc. 2016. *SAS 9.1.3 help and documentation*. [ONLINE] Available at: http://support.sas.com/documentation/onlinedoc/91pdf/index_913.html. [Accessed 19 July 2017].

Siddiqi, N., 2005. *Credit Risk Scorecards Developing and Implementing Intelligent Credit Scoring*. 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc..

Shamos, M.I. and Hoey, D., 1975, October. Closest-point problems. In Foundations of Computer Science, 1975., 16th Annual Symposium on (pp. 151-162). IEEE.

Skurichina, M. & Duin, R.P. 2002. Bagging, boosting and the random subspace method for linear classifiers. *Pattern analysis and applications,* 5(2):121-135

Statistics How To. 2016. What is a Bootstrap Sample?. [ONLINE] Available at: http://www.statisticshowto.com/bootstrap-sample/. [Accessed 23 May 2018].

Sohn, S.Y. & Kim, H.S. 2007. Random effects logistic regression model for default prediction of technology credit guarantee fund. *European journal of operational research,* 183(1):472-478.

Song, B., Zhang, G., Zhu, W. & Liang, Z. 2014. ROC operating point selection for classification of imbalanced data with application to computer-aided polyp detection in CT colonography. *International journal of computer assisted radiology and surgery,* 9(1):79-89.

Stehman, S.V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote sensing of environment,* 62(1):77-89.

Sterne, J., 2003. Web metrics: Proven methods for measuring web site success. 1st ed. John Wiley and Sons.

Tao, D., Tang, X., Li, X. & Wu, X. 2006. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE transactions on pattern analysis and machine intelligence,* 28(7):1088-1099.

Ting, K.M., 2011. Precision and recall. In Encyclopedia of machine learning. 1st ed. (pp. 781-781). Springer US.

Tomek, I. 1976. Two modifications of CNN. *IEEE Transaction on systems, man and cybernetics,* 6, :769-772.

Tremblay, G., Sabourin, R. and Maupin, P., 2004, August. Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on (Vol. 1, pp. 208-211). IEEE.

Upadhyay, R. 2014. Information value (IV) and weight of evidence (WOE) – a case study from banking (part 4). [ONLINE] Available at: http://ucanalytics.com/blogs/information-value-and-weight-of-evidencebanking-case/ [Accessed: 11 Apr. 2018].

Wasserman, H. 2010. The pitfalls of switching banks. [ONLINE] Available at: https://www.fin24.com/Money/Money-Clinic/The-pitfalls-of-switching-banks-20101110 [Access: 14 Nov. 2018].

Weiss, G.M. and Provost, F., 2001. The effect of class distribution on classifier learning: an empirical study. 1st ed. Rutgers Univ.

Wettschereck, D., Aha, D.W. and Mohri, T., 1997. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273-314.

Williams, D.A. 1975. The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics,* 31(4):949-952.

Wilson, D.L. 1972. Asymptotic properties of nearest neighbour rules using edited data. *IEEE transactions on systems, man, and cybernetics,* 2(3):408-421.

Witten, I.H., Frank, E. & Hall, M.A. 2011. Data mining. 1st ed. Burlington, MA. USA: Elsevier.

Appendix A

Random forest code

```
data Random1;
setres.attrition_final2;
*-------------------------------------------------------------*;
* EM SCORE CODE;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Partition Class;
* TYPE: SAMPLE;
* NODE: Part;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Extension Class;
* TYPE: CREDSCORE;
* NODE: IGN;
*-------------------------------------------------------------*;
length _UFormat $200;
drop _UFormat;
_UFormat='';

*-------------------------------------------------------------*;
* Variable: var126;
*-------------------------------------------------------------*;
LABEL GRP_var126 =
"Grouped: BLNC_M1_CRTO_Q1_PCT";
LABEL WOE_var126 =
"Weight of Evidence: BLNC_M1_CRTO_Q1_PCT";

if MISSING(var126) thendo;
GRP_var126 = 9;
WOE_var126 = -0.677115808;
end;
elseifNOT MISSING(var126) thendo;
if var126 <0.1thendo;
GRP_var126 = 1;
WOE_var126 = -0.342439176;
end;
else
if0.1<= var126 AND var126 <0.3thendo;
GRP_var126 = 2;
```

```
WOE_var126 = -0.212334475;
end;
else
if0.3<= var126 AND var126 <0.8thendo;
GRP_var126 = 3;
WOE_var126 = -0.134608132;
end;
else
if0.8<= var126 AND var126 <1.7thendo;
GRP_var126 = 4;
WOE_var126 = -0.072853407;
end;
else
if1.7<= var126 AND var126 <6.3thendo;
GRP_var126 = 5;
WOE_var126 =  0.036019241;
end;
else
if6.3<= var126 AND var126 <15.5thendo;
GRP_var126 = 6;
WOE_var126 = 0.1410114748;
end;
else
if15.5<= var126 AND var126 <126.6thendo;
GRP_var126 = 7;
WOE_var126 = 0.3204403497;
end;
else
if126.6<= var126 thendo;
GRP_var126 = 8;
WOE_var126 = 0.4407076933;
end;
end;

.

.

.

.

ifNOT MISSING(var88) AND var88 eq -92929
thendo;
GRP_var88 = 11;
WOE_var88 = 0.1953145889;
end;

ifNOT MISSING(var88) AND var88 eq -92929.00
thendo;
GRP_var88 = 11;
WOE_var88 = 0.1953145889;
```

```
end;

ifNOT MISSING(var88) AND var88 eq -93939
thendo;
GRP_var88 = 11;
WOE_var88 = 0.1953145889;
end;

ifNOT MISSING(var88) AND var88 eq -93939.00
thendo;
GRP_var88 = 11;
WOE_var88 = 0.1953145889;
end;
run;
*-------------------------------------------------------------*;
* TOOL: Extension Class;
* TYPE: MODEL;
* NODE: HPDMForest;
*-------------------------------------------------------------*;
%letem_score_output = Random1;
data Random1;
SET Random1;
%macroem_hpfst_score;

%if%symexist(hpfst_score_input)=0%then%lethpfst_score_input=&em_score_
output;
%if%symexist(hpfst_score_output)=0%then%lethpfst_score_output=&em_scor
e_output;
%if%symexist(hpfst_id_vars)=0%then%lethpfst_id_vars = _ALL_;

%lethpvvn= %sysfunc(getoption(VALIDVARNAME));
  options validvarname=V7;
  proc hp4score data=&hpfst_score_input;
  id &hpfst_id_vars;
%if%symexist(EM_USER_OUTMDLFILE)=0%then%do;
    score file="/grid/isilon/sharedatafs/EMiner_test/Andre
attrition/Gold attrition/Workspaces/EMWS1/HPDMForest/OUTMDLFILE.bin"
out=&hpfst_score_output;
%end;
%else%do;
    score file="&EM_USER_OUTMDLFILE" out=&hpfst_score_output;
%end;
PERFORMANCE  DETAILS;
  run;

  options validvarname=&hpvvn;

  data &hpfst_score_output;
    set &hpfst_score_output;
%mend;

%em_hpfst_score;
```

```
*-------------------------------------------------------------*;
*Computing Classification Vars: SAL_IND;
*-------------------------------------------------------------*;
length _format200 $200;
drop _format200;
_format200= ' ' ;
length _p_ 8;
_p_= 0 ;
drop _p_ ;
if P_SAL_IND1 - _p_ >1e-8thendo ;
   _p_= P_SAL_IND1 ;
   _format200='1';
end;
if P_SAL_IND0 - _p_ >1e-8thendo ;
   _p_= P_SAL_IND0 ;
   _format200='0';
end;
I_SAL_IND=dmnorm(_format200,32); ;
length U_SAL_IND 8;
label U_SAL_IND = 'Unnormalized Into: SAL_IND';
if I_SAL_IND='1'then
U_SAL_IND=1;
if I_SAL_IND='0'then
U_SAL_IND=0;
data&em_score_output;
set&em_score_output;
*-------------------------------------------------------------*;
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score2;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score2: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_EVENTPROBABILITY = 'Probability for level 1 of SAL_IND';
EM_EVENTPROBABILITY = P_SAL_IND1;
LABEL EM_PROBABILITY = 'Probability of Classification';
EM_PROBABILITY =
max(
P_SAL_IND1
,
P_SAL_IND0
);
LENGTH EM_CLASSIFICATION $%dmnorlen;
LABEL EM_CLASSIFICATION = "Prediction for SAL_IND";
EM_CLASSIFICATION = I_SAL_IND;
Run;
```

Logistic regression code

```
datares_reg;
setres.attrition_final2;

*-------------------------------------------------------------*;
* EM SCORE CODE;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Input Data Source;
* TYPE: SAMPLE;
* NODE: Ids;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Partition Class;
* TYPE: SAMPLE;
* NODE: Part;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* TOOL: Extension Class;
* TYPE: CREDSCORE;
* NODE: IGN;
*-------------------------------------------------------------*;
length _UFormat $200;
drop _UFormat;
_UFormat='';

*-------------------------------------------------------------*;
* Variable: var126;
*-------------------------------------------------------------*;
LABEL GRP_var126 =
"Grouped: BLNC_M1_CRTO_Q1_PCT";

if MISSING(var126) thendo;
GRP_var126 = 4;
end;
elseifNOT MISSING(var126) thendo;
if var126 <1.2thendo;
GRP_var126 = 1;
end;
else
if1.2<= var126 AND var126 <6.3thendo;
GRP_var126 = 2;
end;
else
if6.3<= var126 thendo;
GRP_var126 = 3;
end;
end;
ifNOT MISSING(var126) AND var126 eq -91919
thendo;
```

```
GRP_var126 = 5;
end;

.
.
.
.
ifNOT MISSING(var144) AND var144 eq -92929.00
thendo;
WOE_var144 =            0;
end;

ifNOT MISSING(var144) AND var144 eq -93939
thendo;
WOE_var144 =            0;
end;

ifNOT MISSING(var144) AND var144 eq -93939.00
thendo;
WOE_var144 =            0;
end;

*-------------------------------------------------------------*;
* TOOL: Regression;
* TYPE: MODEL;
* NODE: Reg;
*-------------------------------------------------------------*;
************************************;
*** begin scoring code for regression;
************************************;

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

length I_SAL_IND $ 12;
label I_SAL_IND = 'Into: SAL_IND' ;
*** Target Values;
array REGDRF [2] $12_temporary_ ('1''0' );
label U_SAL_IND = 'Unnormalized Into: SAL_IND' ;
*** Unnormalized target values;
ARRAYREGDRU[2]  _TEMPORARY_ (10);

drop _DM_BAD;
_DM_BAD=0;

*** Check WOE_var144 for missing values ;
ifmissing( WOE_var144 ) thendo;
substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

*** Generate dummy variables for GRP_var126 ;
```

```
drop _1_0 _1_1 _1_2 _1_3 _1_4 ;
*** encoding is sparse, initialize to zero;
_1_0 = 0;
_1_1 = 0;
_1_2 = 0;
_1_3 = 0;
_1_4 = 0;
ifmissing( GRP_var126 ) thendo;
   _1_0 = .;
   _1_1 = .;
   _1_2 = .;
   _1_3 = .;
   _1_4 = .;
substr(_warn_,1,1) = 'M';
   _DM_BAD = 1;
end;

.
.
.

%DMNORMIP( _dm12 )
   _dm_find = 0; drop _dm_find;
if _dm12 <= '4'thendo;
if _dm12 <= '2'thendo;
if _dm12 = '1'thendo;
            _30_0 = 1;
            _dm_find = 1;
end;
elsedo;
if _dm12 = '2'thendo;
               _30_1 = 1;
               _dm_find = 1;
end;
end;
end;
elsedo;
if _dm12 = '3'thendo;
            _30_2 = 1;
            _dm_find = 1;
end;
elsedo;
if _dm12 = '4'thendo;
               _30_3 = 1;
               _dm_find = 1;
end;
end;
end;
end;
elsedo;
if _dm12 <= '6'thendo;
if _dm12 = '5'thendo;
```

```
            _30_4 = 1;
            _dm_find = 1;
end;
elsedo;
if _dm12 = '6'thendo;
               _30_5 = 1;
               _dm_find = 1;
end;
end;
end;
elsedo;
if _dm12 = '8'thendo;
            _30_0 = -1;
            _30_1 = -1;
            _30_2 = -1;
            _30_3 = -1;
            _30_4 = -1;
            _30_5 = -1;
            _dm_find = 1;
end;
end;
end;
ifnot _dm_findthendo;
      _30_0 = .;
      _30_1 = .;
      _30_2 = .;
      _30_3 = .;
      _30_4 = .;
      _30_5 = .;
substr(_warn_,2,1) = 'U';
      _DM_BAD = 1;
end;
end;

*** If missing inputs, use averages;
if _DM_BAD >0thendo;
   _P0 = 0.062407564;
   _P1 = 0.937592436;
goto REGDR1;
end;

*** Compute Linear Predictor;
drop _TEMP;
drop _LP0;
_LP0 = 0;

***  Effect: GRP_var126 ;
_TEMP = 1;
_LP0 = _LP0 + (  1.65127941521712) * _TEMP * _1_0;
_LP0 = _LP0 + (  1.59214803379679) * _TEMP * _1_1;
_LP0 = _LP0 + (  1.40914406174984) * _TEMP * _1_2;
_LP0 = _LP0 + (   -8.34350429008713) * _TEMP * _1_3;
```

```
_LP0 = _LP0 + (  1.59352023347184) * _TEMP * _1_4;

.

.

.

***  Effect: GRP_var88 ;
_TEMP = 1;
_LP0 = _LP0 + (   -0.11783941173409) * _TEMP * _30_0;
_LP0 = _LP0 + (  0.07789164386003) * _TEMP * _30_1;
_LP0 = _LP0 + (   -0.03763633521992) * _TEMP * _30_2;
_LP0 = _LP0 + (     -0.109693686399) * _TEMP * _30_3;
_LP0 = _LP0 + (  0.35716607752005) * _TEMP * _30_4;
_LP0 = _LP0 + (  0) * _TEMP * _30_5;

***  Effect: WOE_var144 ;
_TEMP = WOE_var144 ;
_LP0 = _LP0 + (  0.1274852934716 * _TEMP);

*** Naive Posterior Probabilities;
drop _MAXP _IY _P0 _P1;
_TEMP =    -2.75361943728833 + _LP0;
if (_TEMP <0) thendo;
_TEMP = exp(_TEMP);
   _P0 = _TEMP / (1 + _TEMP);
end;
else _P0 = 1 / (1 + exp(-_TEMP));
_P1 = 1.0 - _P0;

REGDR1:


*** Posterior Probabilities and Predicted Level;
label P_SAL_IND1 = 'Predicted: SAL_IND=1' ;
label P_SAL_IND0 = 'Predicted: SAL_IND=0' ;
P_SAL_IND1 = _P0;
_MAXP = _P0;
_IY = 1;
P_SAL_IND0 = _P1;
if (_P1 >  _MAXP + 1E-8) thendo;
   _MAXP = _P1;
   _IY = 2;
end;
I_SAL_IND = REGDRF[_IY];
U_SAL_IND = REGDRU[_IY];

************************************;
***** end scoring code for regression;
************************************;
*-------------------------------------------------------------*;
```

```
* TOOL: Score Node;
* TYPE: ASSESS;
* NODE: Score4;
*-------------------------------------------------------------*;
*-------------------------------------------------------------*;
* Score4: Creating Fixed Names;
*-------------------------------------------------------------*;
LABEL EM_EVENTPROBABILITY = 'Probability for level 1 of SAL_IND';
EM_EVENTPROBABILITY = P_SAL_IND1;
LABEL EM_PROBABILITY = 'Probability of Classification';
EM_PROBABILITY =
max(
P_SAL_IND1
,
P_SAL_IND0
);
LENGTH EM_CLASSIFICATION $%dmnorlen;
LABEL EM_CLASSIFICATION = "Prediction for SAL_IND";
EM_CLASSIFICATION = I_SAL_IND;
run;
```