Predicting credit card customer churn in banks using data mining

Dudyala Anil Kumar and V. Ravi*

Institute for Development and Research in Banking Technology
Castle Hills Road #1, Masab Tank
Hyderabad 500 057 (AP), India
Fax: +91–40–2353 5157
E-mail: anilkumard001@gmail.com
E-mail: rav_padma@yahoo.com
*Corresponding author

Abstract: In this paper, we solve the customer credit card churn prediction via data mining. We developed an ensemble system incorporating majority voting and involving Multilayer Perceptron (MLP), Logistic Regression (LR), decision trees (J48), Random Forest (RF), Radial Basis Function (RBF) network and Support Vector Machine (SVM) as the constituents. The dataset was taken from the Business Intelligence Cup organised by the University of Chile in 2004. Since it is a highly unbalanced dataset with 93% loyal and 7% churned customers, we employed (1) undersampling, (2) oversampling, (3) a combination of undersampling and oversampling and (4) the Synthetic Minority Oversampling Technique (SMOTE) for balancing it. Furthermore, tenfold cross-validation was employed. The results indicated that SMOTE achieved good overall accuracy. Also, SMOTE and a combination of undersampling and oversampling improved the sensitivity and overall accuracy in majority voting. In addition, the Classification and Regression Tree (CART) was used for the purpose of feature selection. The reduced feature set was fed to the classifiers mentioned above. Thus, this paper outlines the most important predictor variables in solving the credit card churn prediction problem. Moreover, the rules generated by decision tree J48 act as an early warning expert system.

Keywords: churn prediction; Multilayer Perceptron; MLP; Logistic Regression; LR; decision tree; Random Forest; RF; radial basis function network; Support Vector Machine; SVM; Synthetic Minority Oversampling Technique; SMOTE; undersample; oversampling.

Reference to this paper should be made as follows: Anil Kumar, D. and Ravi, V. (2008) 'Predicting credit card customer churn in banks using data mining', *Int. J. Data Analysis Techniques and Strategies*, Vol. 1, No. 1, pp.4–28.

Biographical notes: Dudyala Anil Kumar is pursuing his MTech in Information Technology with specialisations in Banking Technology and Information Security from the Institute for Development and Research in Banking Technology (IDRBT), Hyderabad, and the University of Hyderabad, India. He holds a BTech in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad. His research interests include data mining, soft computing, operating systems and compiler design.

Vadlamani Ravi is an Assistant Professor at the IDRBT, Hyderabad, India, since April 2005. He holds a PhD in Soft Computing from Osmania University, Hyderabad, and Rheinisch-Westfaelische Technische Hochschule

(RWTH) Aachen Germany. Earlier, he was a Faculty Member at the National University of Singapore (NUS), Singapore, for three years. Prior to that, he was the Assistant Director and a Scientist at the Indian Institute of Chemical Technology (IICT), Hyderabad. He published 53 papers in refereed journals/conferences and invited book chapters. He edited *Advances in Banking Technology and Management: Impacts of ICT and CRM*, which was published by IGI Global, USA. He is a Referee for several international journals and on the editorial board of the *International Journal of Information and Decision Sciences (IJIDS),* the *International Journal of Data Analysis Techniques and Strategies (IJDATS),* the *International Journal of Information Systems in Service Sector (IJISS)* and the *International Journal of Information Technology Project Management (IJITPM).*

1 Introduction

In the current business environment, banks and financial companies have a huge number of customers. Banks provide services through various channels, like ATMs, debit cards, credit cards, internet banking, *etc*. The number of customers has increased enormously and customers have become increasingly conscious of the quality of service. This fuels tremendous competition amongst various banks, resulting in a significant increase in the reliability of and quality of service from banks. Also, customers shift loyalties from one bank to another because of varied reasons, such as the availability of the latest technology, customer-friendly bank staff, low interest rates, the proximity of the geographical location, the various services offered, *etc*. Hence, there is a pressing need to develop a model which can predict which customer is likely to churn out based on the customers' demographic, psychographic and transactional data.

An effective Customer Relationship Management (CRM) decision support system increases the quality of customer relationships, thereby increasing retention in several ways:

- it supports predictive modelling to help banks identify who is likely to leave and why and what to do about it
- it enables a new level of personalisation in service offers and marketing approaches, which fosters loyalty
- it brings greater richness to customer interaction, thereby increasing customer satisfaction, because consistent information is shared across all customer touch points (CRM in Banking, A SAS White Paper, 2001).

The phenomenon of customer churn is not limited to just banking and financial industries. It is very much prevalent in other service industries too, such as mobile telecommunications, television viewership, *etc*.

Bolton (1998) suggested that service organisations should be proactive and learn from customers before they defect by understanding their current satisfaction levels. He also suggested that service encounters act as early indicators of whether an organisation's relationship with a customer is flourishing or not. He concluded that the customers who have longer relationships with the firm have higher prior cumulative satisfaction ratings and smaller subsequent perceived losses that are associated with subsequent service

encounters. Bolton *et al*. (2000) suggested that it is theoretically more profitable to segment and target customers on the basis of their (changing) purchase behaviours and service experiences, rather than on the basis of their (stable) demographics or other variables. Lejeune (2001) presented a CRM framework that is based on the integration of the electronic channel. He concluded that churn management consists of developing techniques that enable firms to keep their profitable customers and aims at increasing customer loyalty.

Au *et al*. (2003) worked on the Credit Card Database and PBX Database with Data Mining by Evolutionary Learning (DMEL) and the experimental results on the telecom subscriber database showed that DMEL is a robust way to predict churn. Lariviere and Van den Poel (2004b) emphasised the natural differences between Savings and Investment (SI) products and explored the most convenient products to cross-sell in terms of both maximising the customers' retention proneness and their preferences with respect to SI products. Burez and Van den Poel (2007) concluded that, using the full potential of their churn prediction model and the available incentives, the pay-TV company's profits from the churn prevention programme would double when compared to its current model.

Lu (2008) (SUGI 28) applied survival analysis to estimate the customer survival curve and in the calculation of the customer lifetime value. He carried out the study on four major data sources of the census block level, which include marketing and financial information, customer-level demographic data, customer internal data and customer contact records. He concluded that the customer lifetime value is a powerful measure that synthesised the customer profitability and churn risk.

Lariviere and Van den Poel (2004a) concluded that in the financial services industry, two 'critical' churn periods can be identified: the early years (after becoming a customer) and a second period (after being a customer for some 20 years). The online bank customers are less price sensitive, which means that once they are captured, the likelihood that they will leave, if, for example, they receive better offers from competitors or their bank raises its fees, is lower than that for traditional bank customers (Mols, 1998). If a customer starts acting in a way that manifests attrition, management should prepare an anti-churn strategy that is usually far less expensive than acquiring new customers (Chu *et al*., 2007). Estevez *et al*. (2006) showed that analysing the customer antecedents at the time of application could prevent subscription fraud in telecommunications. It is argued that a 5% increase in customer retention can result in an 18% reduction in the operating costs (Karakostas *et al*., 2005).

In this paper, we conducted the most comprehensive investigation of the credit card churn prediction problem in banks using data mining. We employed a host of intelligent techniques (both in isolation and combination) to predict customer churn within the framework of the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology.[1] For this purpose, we considered a dataset from the Business Intelligence Cup that was organised by the University of Chile in 2004. Furthermore, we developed an ensemble system based on majority voting that involved the top techniques among the several that are considered in the study.

The rest of the paper is organised as follows. The literature review is discussed in Section 2. Then, Section 3 presents a brief overview of various intelligent techniques. Section 4 describes the dataset analysed in the study and presents the methodology that is followed for carrying out the experiments. Section 5 presents the results and discussion, followed by the conclusions in Section 6.

2 Literature review

In the following paragraphs, we present a brief overview of the various models that were developed for customer churn prediction by researchers in different domains. Bolton *et al*. (2000) used Logistic Regression (LR) and *t*-tests for loyalty programme membership and concluded that loyalty rewards programmes help build stronger relationships with customers. Mozer *et al*. (2000) analysed the subscriber data provided by a major wireless carrier by applying LR, decision trees, neural networks and boosting techniques.

Ultsch (2001) worked with data from Swisscom AG by using Emergent Self-Organising Feature Maps (ESOM). Buckinx and Van den Poel (2005) used LR and Random Forests (RF) and concluded that compared to customer demographics, Recency, Frequency and Monetary (RFM) variables are better at separating behaviourally loyal customers than those who have a tendency to defect. Lariviere and Van den Poel (2004b) studied the defection of the SI customers of a large Belgian financial services provider using survival estimates, hazard ratios and multinomial probit analysis. Burez and Van den Poel (2007) worked on pay-TV company data using LR, the Markov Chain (MC) and RF. Hung and Yen (2006) applied the decision tree and back propagation neural network on a wireless telecom company's customer data and concluded that the performance of building a predictive model on individual segments is more accurate than the one built on the entire customer population.

Neural networks, combined with a powerful rule discovery method in the form of a genetic algorithm, provide a customer churn prediction model with very good predictive capabilities (Hadden *et al*., 2005). Hadden *et al*. (2006) compared neural networks and decision trees in predicting customer churn. The decision tree outperformed all of the techniques. Bloemer *et al*. (1998) carried out an empirical study on customer loyalty with the help of the customer data of a major bank in the Netherlands. They used multivariate regression analysis to gain additional insight into the data and proposed a model to test the relationship between image and loyalty. Hu (2005) used lift as a proper measure for attrition analysis and compared the lift of data mining models to decision trees, boosted naïve Bayesian networks, selective Bayesian networks, neural networks and the ensemble of the classifiers of the methods above. He reported a highest hit ratio of 14.8% for an ensemble classifier.

Michael *et al*. (2000) worked with wireless subscribers. They employed Logit Regression (LR), decision trees and neural networks for churn modelling in an online shopping application. They concluded that using sophisticated neural networks, $93 could be saved. Mutanen (2006) presented a customer churn analysis of the personal retail banking sector based on LR. Neslin *et al.* (2004) suggested five approaches to estimating customer churn: logistic, trees, novice, discriminant and explain. Their results suggested that by using a logistic or tree approach, a company could achieve a good level of prediction. Ferreira *et al*. (2004) used Multilayer Perceptron (MLP), C4.5 decision trees, Hierarchical Neuro-Fuzzy Systems and a data mining tool named Rule Evolver, based on Genetic Algorithms (GA). He used a wireless dataset from Brazil. Neural networks with 15 hidden units accomplished the best classification. Euler (2005) worked with telecommunications data. He used decision trees to solve the customer churn problem and achieved an accuracy rate of 82%. Yuan and Chang (2001) presented a mixed-initiative synthesised learning approach for a better understanding of customers and the provision of clues for the improvement of customer relationships based on

different sources of online customer data. The approach was a combination of hierarchical automatic labelling Self-Organising Map (SOM), decision trees, cross-class analysis and human tacit experience. The approach was implemented as a system called CRMiner and applied to the data of the Taiwanese branch of a leading printer company.

Larivie`re *et al*. (2005) demonstrated that RF provided a better fit for the estimation and validation sample, compared to ordinary linear regression and LR models. Richeldi *et al*. (2002) created a model using a decision tree operator, which predicts the likelihood of a customer becoming a churner. Hung and Yen (2006) reported that both the decision tree and neural network techniques can deliver accurate churn prediction models by using customer demographics, billing information, the contract/service status, call detail records and service change logs. Wezel and Potharst (2000) worked with decision trees, ensemble versions of decision trees and the LR model. They concluded that ensemble learning usually improves the prediction quality of flexible models like decision trees and, thus, leads to improved predictions and the ensemble models are found to outperform individual decision trees and LR. Buckinx *et al*. (2005) had worked with LR, Automatic Relevance Determination (ARD), neural networks and RF. Their results show that RF consistently outperformed everything else. Smith and Gupta (2000) employed multilayer feed forward neural networks, Hopfield neural networks and self-organising neural networks to solve churn problems. Chu *et al*. (2007) constructed a churn model using decision trees and achieved an accuracy rate of 85%. Revett *et al*. (2006) used the rough sets algorithm and obtained an overall classification accuracy rate of 90%.

3 Intelligent models employed in the study

3.1 Random forest

RF, invented by Breiman and Cutler, generates many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification and we say that the tree 'votes' for that class. The forest chooses the classification that has the most votes (over all of the trees in the forest). RFs possess the following properties:[2]

- it is extremely good in accuracy among the current algorithms
- it runs efficiently on large databases
- it can handle thousands of input variables without variable deletion
- it gives estimates of what variables are important in the classification
- it generates an internal, unbiased estimate of the generalisation error as the forest building progresses
- it has an effective method for estimating missing data and maintains accuracy when a large proportion of the data is missing
- it has methods for balancing unbalanced datasets
- generated forests can be saved for future use on other data
- prototypes that give information about the relation between the variables and the classification are computed

- it computes the proximities between pairs of cases that can be used in clustering, locates outliers or (by scaling) gives interesting views of the data
- the capabilities of the random forest can be extended to unlabelled data, leading to unsupervised clustering, data views and outlier detection
- it offers an experimental method for detecting variable interactions.

After each tree is built, all of the data are run down the tree and the proximities are computed for each pair of cases. If two cases occupy the same terminal node, their proximity is increased by one. At the end of the run, the proximities are normalised with respect to the number of trees. Proximities are used in replacing missing data, locating outliers and producing low-dimensional views of the data.[2]

3.2 Support vector machine

A Support Vector Machine (SVM) performs classification by constructing an *N*-dimensional hyperplane that optimally separates the data into two categories. SVM models are closely related to neural networks. In fact, an SVM model that uses a sigmoid kernel function is equivalent to a two-layer perceptron neural network. SVM models are a close cousin to classical MLP neural networks. Using a kernel function, SVMs are an alternative training method for polynomial, Radial Basis Function (RBF) networks and MLP classifiers, in which the weights of the network are found by solving a quadratic programming problem with linear constraints, rather than by solving a nonconvex, unconstrained minimisation problem, as in standard neural network training.

The goal of SVM modelling is to find the optimal hyperplane that separates samples, in such a way that the samples with one category of the target variable should be on one side of the plane and the samples with the other category are on the other side of the plane. The samples near the hyperplane are the support vectors. An SVM analysis finds the hyperplane that is oriented so that the margin between the support vectors is maximised. If all analyses consisted of two-category target variables with two predictor variables and a straight line could divide the cluster of points, life would be easy. For classification, SVM operates by finding a hyperplane in the space of possible inputs. This hyperplane attempts to split the positive examples from the negative examples.[3] The 'margin' of a separating hyperplane is the distance from the hyperplane to the closest training case. One idea is that performance on test cases will be good if we choose the separating hyperplane that has the largest margin.[4] There are various ways to train SVM. One particularly simple and fast method is Sequential Minimal Optimisation (SMO). Konstanz Information Miner (KNIME) uses SMO for training SVM.

4 Methodology

4.1 Dataset

The dataset is from a Latin American bank that suffered from an increasing number of churns with respect to their credit card customers and decided to improve its retention system. Two groups of variables are available for each customer: sociodemographic and behavioural data, which are described in Table 1. The dataset comprises 22 variables, with 21 predictor variables and 1 class variable. It consists of 14 814 records, of which

13 812 are nonchurners and 1002 are churners, which means there are 93.24% nonchurners and 6.76% churners. Hence, the dataset is highly unbalanced in terms of the proportion of churners versus nonchurners. Before supplying these data to the classifier, we need to balance the data, so that the classifiers do not tend towards the majority class (nonchurner) while predicting. The balancing is done separately using the Synthetic Minority Oversampling Technique (SMOTE), undersampling and oversampling.

Table 1 The description of the data

Variable	*Description*	*Value*
Target	Target variable	0 – Non churner 1 – Churner
CRED_T	Credit in month T	Positive real number
CRED_T-1	Credit in month T-1	Positive real number
CRED_T-2	Credit in month T-2	Positive real number
NCC_T	Number of credit cards in month T	Positive integer value
NCC_T-1	Number of credit cards in month T-1	Positive integer value
NCC_T-2	Number of credit cards in month T-2	Positive integer value
INCOME	Customer's income	Positive real number
N_EDUC	Customer's educational level	1 – University student 2 – Medium degree
		3 – Technical degree 4 – University degree
AGE	Customer's age	Positive integer
SX	Customers sex	1 – Male 0 – Female
E_CIV	Civilian status	1 – Single 2 – Married
		3 – Widow 4 – Divorced
T_WEB_T	Number of web transactions in month T	Positive integer
T_WEB_T-1	Number of web transactions in month T-1	Positive integer
T_WEB_T-2	Number of web transactions in month T-2	Positive integer
MAR_T	Customer's margin for the company in month T	Real number
MAR_T-1	Customer's margin for the company in month T-1	Real number
MAR_T-2	Customer's margin for the company in month T-2	Real number
MAR_T-3	Customer's margin for the company in month T-3	Real number
MAR_T-4	Customer's margin for the company in month T-4	Real number
MAR_T-5	Customer's margin for the company in month T-5	Real number
MAR_T-6	Customer's margin for the company in month T-6	Real number

4.2 SMOTE

SMOTE is an approach in which the minority class is oversampled by creating synthetic (or artificial) samples, rather than by oversampling with replacement. The minority class is oversampled by taking out each sample and introducing synthetic samples along the line segments that join any/all of the k minority class nearest neighbours. SMOTE is used to widen the data region that correponds to minority samples. This approach effectively forces the decision region of the minority class to become more general (Chawla *et al*., 2004).

4.3 Feature selection

Since the dataset has many predictor variables, some of them may be less important in predicting the class of the sample. Therefore, it is important to select only those variables which play a vital role in determining the class of the sample. Feature selection is done

using the Classification and Regression Tree (CART), which gives the importance of each variable in a sorted order. Based on the importance of the variables, we chose the top six variables, whose importance ratings are greater than 50%. Thus, the selected variables are CRED_T, CRED_T-1, CRED_T-2, NCC_T, NCC_T-1 and NCC_T-2. We performed stratified random sampling where the original proportion of the churned versus loyal customers (7%:93%) is preserved and generated a smaller dataset for the purpose of feature selection.

4.4 Selecting top classifiers using the AUC criterion of ROC

A Receiver Operating Characteristics (ROC) graph has long been used in signal detection theory to depict the tradeoff between the hit rates and false alarm rates of classifiers. The ROC graph is a two-dimensional graph which represents the various classifiers based on their output results in point form in a region, which has an False Positive (FP) rate (1-Specificity) on the X-axis and a True Positive (TP) rate (Sensitivity) on the Y-axis. ROC graphs are a very useful tool for visualising and evaluating classifiers. They are more able to provide a richer measure of classification performance than scalar measures such as accuracy, the error rate or error cost (Fawcett, 2006). The Area Under the ROC Curve (AUC) of a Classifier A can be calculated as the sum of the areas of Triangle CEA, Rectangle EAFI and Triangle RAH. The areas are shown in Figure 2. The classifier, which has a bigger area under the ROC curve, is the better classifier (Ravi *et al*., 2007).

After getting the results of the prediction using different classifiers, we computed the sensitivity, specificity and overall accuracy. Based on the results that were obtained by different classifiers, we plotted the ROC. Then, we calculated the AUC for each classifier using a simple geometric formula and ranked the classifiers in the descending order of AUC.

Figure 1 Simple majority voting

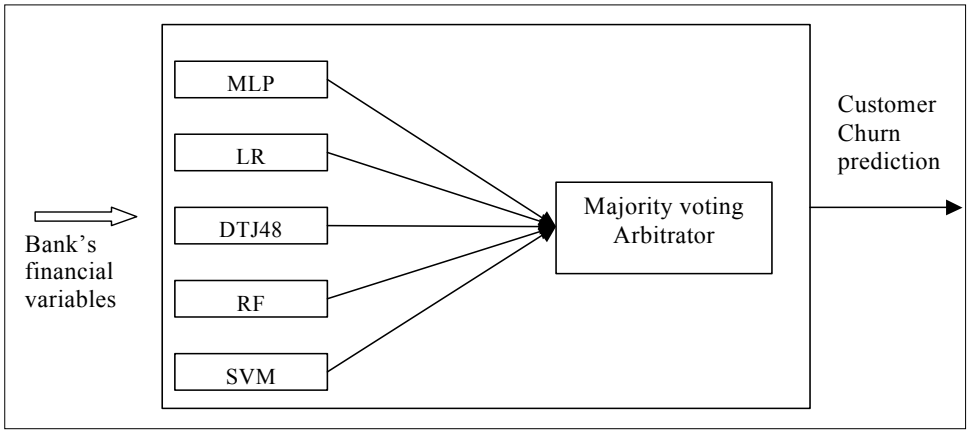
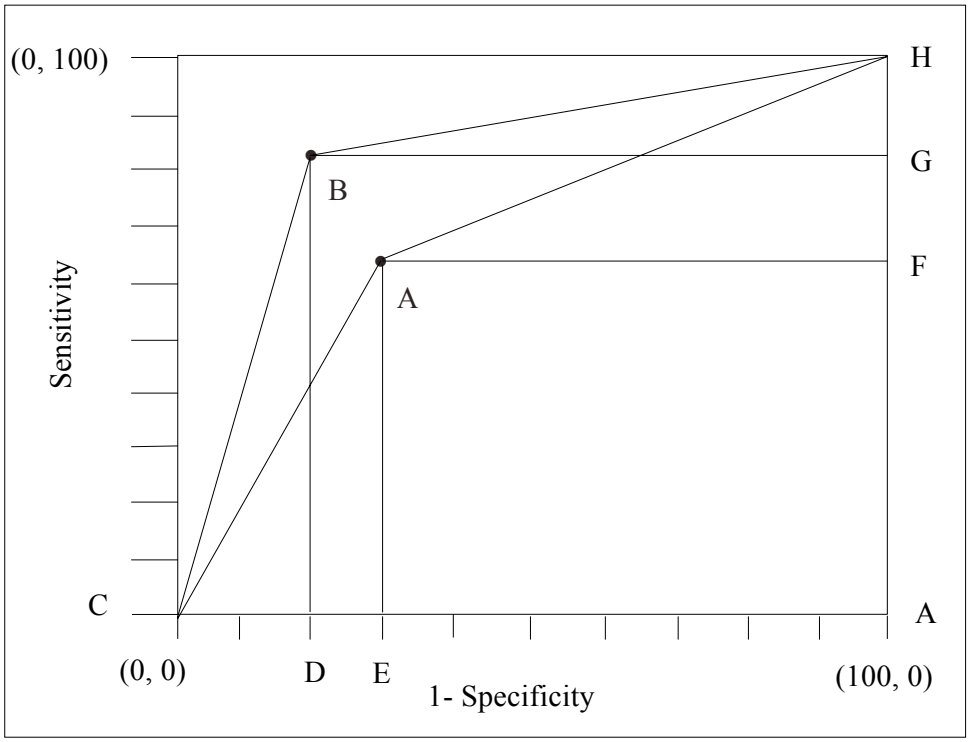


Figure 2 The AUC of classifiers A and B



4.5 Simple majority voting

Simple majority voting is an ensemble method that helps us find out the prediction based on the output of the majority of the classifiers. After getting the majority vote and labelling the samples, we computed the sensitivity, specificity and accuracy of the output. Figure 1 shows the architecture of the simple majority voting system.

5 Results and discussion

The results that were obtained from the experiments conducted in the study are presented in Tables 2 to 19. The organisation of the tables is as follows. Tables 2 to 10 present the results of the experiments that were done using the hold-out method and Tables 11 to 19 present the average results of the experiments that were done using the tenfold cross-validation method. In undersampling, the majority class is separately undersampled in the ratios of 25% and 50%. In oversampling, we replicate the minority class twice for getting 100% oversampled, thrice for getting 200% oversampled and four times for getting 300% oversampled.

5.1 Hold-out method

5.1.1 Original data

Table 2 presents the results of the original data with full and feature-selected techniques, where the decision tree (J48) ranked at the top for the full dataset with 63.78% sensitivity, 98.31% specificity and 95.97% accuracy, whereas RF ranked at the top for feature-selected dataset with 52.15% sensitivity, 98.74% specificity and 95.59% overall accuracy.

Table 2 The average results of the hold-out method for original data

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	21.92	98.91	93.70	6041	0.66	99.97	93.25	5031
RBF	0	100	93.22	5000	3.98	99.58	93.11	5178
SVM	0	100	93.22	5000	0	100	93.22	5000
LR	5.98	99.44	93.11	5271	0.99	99.95	93.25	5047
Decision tree	*63.78*	*98.31*	*95.97*	*8104*	44.51	99.17	95.47	7184
RF	62.12	99.01	96.51	8056	*52.15*	*98.74*	*95.59*	*7544*

5.1.2 SMOTE data

Table 3 shows the results of the SMOTE data with full and feature-selected techniques, where MLP ranked at the top for both the full and feature-selected data, with 75.41% sensitivity, 92.60% specificity and 91.49% overall accuracy for the full dataset and 81.39% sensitivity, 89.09% specificity and 88.57% overall accuracy for the feature-selected dataset.

Table 3 The average results of the hold-out method for SMOTE data

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	*75.41*	*92.60*	*91.49*	*8400*	*81.39*	*89.09*	*88.57*	*8542*
RBF	90.36	50.82	53.49	7059	80.06	81.03	80.96	8054
SVM	80.06	82.23	82.09	8114	80.39	80.33	80.33	8036
LR	81.39	84.96	84.72	8317	81.06	82.45	82.36	8175
Decision tree	66.44	95.63	93.65	8103	74.41	94.78	93.40	8459
RF	69.76	97.68	95.79	8372	75.08	94.59	93.27	8483

5.1.3 Undersampled data

The results of the undersampling of the majority class data are shown in Tables 4 and 5. Table 4 shows the results of 25% of the undersampled data, in which RF topped with 76.74% sensitivity, 95.84% specificity and 94.60% accuracy for the full dataset and MLP ranked first with 80.06% sensitivity, 90.80% specificity and 90.07% accuracy for the feature-selected dataset. Table 5 presents the results of 50% of the undersampled data, in

which, again, RF topped with 71.09% sensitivity, 97.73% specificity and 95.92% accuracy for the full dataset, whereas MLP topped for the feature-selected dataset, with 80.06% sensitivity, 90.73% specificity and 90.01% accuracy.

Table 4 The average results of the hold-out method for undersampling (25%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	76.74	88.97	88.14	8285	*80.06*	*90.80*	*90.07*	*8543*
RBF	55.48	90.10	87.76	7279	72.42	83.80	83.03	7811
SVM	79.40	84.02	83.71	8171	0.66	99.95	93.22	5030
LR	73.42	92.13	90.86	8277	79.40	90.17	89.44	8478
Decision tree	70.43	94.88	93.22	8265	75.74	94.64	93.36	8519
RF	*76.74*	*95.84*	*94.60*	*8631*	75.74	93.00	91.83	8437

Table 5 The average results of the hold-out method for undersampling (50%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	59.80	95.68	93.25	7774	*80.06*	*90.73*	*90.01*	*8539*
RBF	5.64	98.06	91.81	5185	68.10	94.90	93.09	8150
SVM	0	100	93.22	5000	11.29	98.81	92.89	5505
LR	27.24	97.94	93.16	6259	9.96	98.88	92.95	5442
Decision tree	64.78	96.64	94.48	8071	69.76	96.21	94.42	8298
RF	*71.09*	*97.73*	*95.92*	*8441*	73.75	95.68	94.19	8471

5.1.4 Oversampled data

Tables 6 to 8 show the results of the experiments that were done for the oversampling of the minority class data. Table 6 presents the results of the experiments for 100% of the oversampled data, where RF topped the list with 64.78% sensitivity, 99.01% specificity and 96.69% accuracy for the full dataset and RBF ranked first with 74.41% sensitivity, 94.40% specificity and 93.04% accuracy for the feature-selected dataset. Table 7 presents the results of the experiments for 200% of the oversampled data, where MLP ranked at the top with 77.74% sensitivity, 89.91% specificity and 89.08% accuracy for the full dataset and RBF topped with 78.07% sensitivity, 92.32% specificity and 91.36% accuracy for the feature-selected dataset. Table 8 shows the results of the experiments for 300% of the oversampled data, where MLP topped with 76.74% sensitivity, 91.26% specificity and 90.28% accuracy for the full dataset and LR ranked at the top with 79.73% sensitivity, 90.54% specificity and 89.80% accuracy for the feature-selected dataset.

Table 6 The average results of the hold-out method for oversampling (100%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	44.51	96.83	93.29	7067	10.96	99.30	93.31	5513
RBF	2.32	98.76	92.23	5054	*74.41*	*94.40*	*93.04*	*8440*
SVM	0	100	93.22	5000	0	100	93.22	5000
LR	24.25	98.06	93.07	6115	11.29	98.96	93.02	5512
Decision tree	60.46	96.71	94.26	7858	71.42	95.77	94.12	8359
RF	*64.78*	*99.01*	*96.69*	*8189*	67.77	96.54	94.60	8215

Table 7 The average results of the hold-out method for oversampling (200%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	*77.74*	*89.91*	*89.08*	*8382*	80.39	87.93	87.42	8416
RBF	50.83	92.39	89.58	7161	*78.07*	*92.32*	*91.36*	*8519*
SVM	0.99	99.75	93.07	5037	0	100	93.22	5000
LR	66.11	94.71	92.77	8041	22.25	98.09	92.95	6017
Decision tree	58.13	96.91	94.28	7752	71.76	95.29	93.70	8352
RF	65.78	98.57	96.35	8217	70.09	96.25	94.48	8317

Table 8 The average results of the hold-out method for oversampling (300%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	*76.74*	*91.26*	*90.28*	*8400*	81.72	81.58	81.59	8165
RBF	57.14	89.16	86.99	7315	78.73	89.62	88.88	8417
SVM	79.73	84.79	84.45	8226	80.39	80.96	80.92	8067
LR	75.08	92.08	90.93	8358	*79.73*	*90.54*	*89.80*	*8513*
Decision tree	60.13	97.12	94.62	7862	71.76	94.54	93.00	8315
RF	65.11	98.64	96.37	8187	68.43	96.38	94.48	8240

5.1.5 Combination of undersampling and oversampling

The results of the combination of undersampling and oversampling with various proportions are shown in Tables 9 and 10. Table 9 presents the results of the combination of 25% undersampling and 100% oversampling, where RF topped for both the full and feature-selected datasets with 77.40% sensitivity, 96.47% specificity and 95.18% accuracy for the full dataset and 79.73% sensitivity, 92.25% specificity and 91.40% accuracy for the feature-selected dataset. Table 10 presents the results of the combination of 50% undersampling and 200% oversampling, where LR ranked at the top for the full

dataset with 80.39% sensitivity, 89.47% specificity and 88.86% accuracy and for the feature-selected data, RF topped with 76.07% sensitivity, 95.17% selectivity and 93.88% accuracy.

Table 9 The average results of the hold-out method for combination (25% + 100%)

Classifier	*Full data*				*Reduced feature data*			
	Sensitivity	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	100	0.07	6.81	5003	81.39	85.03	84.79	8321
RBF	82.05	69.90	70.73	7579	79.06	80.57	80.47	7981
SVM	80.06	81.51	81.41	8078	80.39	80.06	80.08	8022
LR	81.06	87.64	87.19	8435	80.73	84.33	84.09	8253
Decision tree	75.74	90.17	89.20	8295	77.40	92.85	91.81	8512
RF	*77.40*	*96.47*	*95.18*	*8693*	*79.73*	*92.25*	*91.40*	*8599*

Table 10 The average results of the hold-out method for combination (50% + 200%)

Classifier	*Full data*				*Reduced feature data*			
	Sensitivity	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	79.06	89.40	88.70	8423	80.39	81.66	81.57	8102
RBF	69.77	85.76	84.61	7726	79.40	88.03	87.44	8371
SVM	80.06	82.74	82.56	8140	80.39	80.35	80.35	8037
LR	*80.39*	*89.47*	*88.86*	*8493*	80.39	86.31	85.91	8335
Decision tree	70.43	94.98	93.31	8270	75.74	94.44	93.18	8509
RF	72.75	97.12	95.47	8493	*76.07*	*95.17*	*93.88*	*8562*

5.2 Tenfold cross-validation method

5.2.1 Original data

Table 11 presents the results of the original data with full and feature-selected techniques, where the decision tree (J48) produced the highest sensitivity of 62.07%, 98.51% specificity and 96.05% overall accuracy for the full dataset. However, RF ranked at the top for the feature-selected dataset with 50.29% sensitivity, 99.20% specificity and 95.89% overall accuracy.

Table 11 The average results of the tenfold cross-validation method for original data

Classifier	*Full data*				*Reduced feature data*			
	Sensitivity	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	6.28	99.47	93.17	5302	0.19	100	93.24	5009
RBF	1.49	99.25	92.64	5051	0	100	93.23	5000
SVM	0	100	93.23	5000	0	100	93.23	5000
LR	5.48	99.51	93.15	5264	1.69	99.89	93.24	5079
Decision tree	*62.07*	*98.51*	*96.05*	*8029*	45.90	99.33	95.72	7261
RF	58.78	99.39	96.64	7908	*50.29*	*99.20*	*95.89*	*7474*

5.2.2 SMOTE data

Table 12 shows the results of SMOTE data with full and feature-selected techniques, where RF ranked at the top for both the full and feature-selected datasets with 79.14% sensitivity, 98.55% specificity and 97.84% overall accuracy for the full dataset and 92.80% sensitivity, 95.40% specificity and 94.10% overall accuracy for the feature- selected dataset.

Table 12 The average results of the tenfold cross-validation method for SMOTE data

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	94.08	91.02	92.55	9255	84.91	86.80	85.85	8585
RBF	93.00	49.59	71.29	7129	74.35	84.51	79.43	7943
SVM	82.43	82.47	82.45	8245	80.50	80.33	80.41	8041
LR	87.05	86.05	86.55	8655	81.72	83.03	82.37	8237
Decision tree	96.31	97.11	96.71	9671	91.81	94.19	93.00	9300
RF	*97.14*	*98.55*	*97.84*	*9784*	*92.80*	*95.40*	*94.10*	*9410*

5.2.3 Undersampled data

Tables 13 and 14 present the results of the undersampling of the majority class data, where Table 13 shows the results of 25% of the undersampled data, in which RF topped for both the full and feature-selected datasets with 73.45% sensitivity, 97.71% specificity and 92.25% accuracy for the full dataset and 75.44% sensitivity, 95.62% specificity and 91.08% accuracy for the feature-selected dataset. Table 14 presents the results of 50% of the undersampled data, in which the decision tree (J48) has topped with 68.26% sensitivity, 96.98% specificity and 93.34% accuracy for the full dataset, whereas RF has topped for the feature-selected dataset with 72.65% sensitivity, 96.53% specificity and 93.51% accuracy.

Table 13 The average results of the tenfold cross-validation method for undersampling (25%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	65.66	92.87	86.75	7926	70.85	91.28	86.68	8106
RBF	55.28	89.71	81.97	7249	70.85	86.35	82.87	7860
SVM	77.64	83.95	82.53	8079	72.05	84.70	81.86	7837
LR	75.44	92.73	88.84	8408	78.54	91.1	88.28	8482
Decision tree	70.65	94.32	89.0	8248	72.45	95.82	90.57	8413
RF	*73.45*	*97.71*	*92.25*	*8558*	*75.44*	*95.62*	*91.08*	*8553*

Table 14 The average results of the tenfold cross-validation method for undersampling (50%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	55.78	95.1	90.12	7544	60.87	94.12	89.90	7749
RBF	6.58	98.2	86.59	5239	21.25	98.11	88.37	5968
SVM	0	100	87.34	5000	0	100	87.32	5000
LR	37.22	69.06	89.99	5314	10.37	99.13	87.88	5475
Decision tree	*68.26*	*96.98*	*93.34*	*8262*	70.05	96.61	93.24	8333
RF	64.17	98.78	94.39	8147	*72.65*	*96.53*	*93.51*	*8459*

5.2.4 Oversampled data

Tables 15 to 17 shows the results of the experiments that were done for the oversampling of the minority class data. Table 15 presents the results of the experiments for 100% of the oversampled data, where RF topped the list for both the full and feature-selected datasets with 94.11% sensitivity, 99.31% specificity and 98.65% accuracy for the full dataset and 90.71% sensitivity, 96.76% specificity and 95.99% accuracy for the feature-selected dataset. Table 16 presents the results of the experiments for 200% of the oversampled data, where RF has ranked at the top again for both the full and feature-selected datasets, with 99.2% sensitivity, 99.27% specificity and 99.26% accuracy for the full dataset and 95.10% sensitivity, 96.60% specificity and 96.33% accuracy for the feature-selected dataset. Table 17 shows the results of the experiments for 300% of the oversampled data, where RF topped for both the full and feature-selected datasets with 100% sensitivity, 99.13% specificity and 99.32% accuracy for the full dataset and 95.75% sensitivity, 96.66% specificity and 96.46% accuracy for the feature-selected dataset.

Table 15 The average results of the tenfold cross-validation method for oversampling (100%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	60.67	94.34	90.07	7750	41.76	95.38	88.58	6857
RBF	6.33	97.97	86.36	5215	20	98.5	88.55	5925
SVM	0	100	87.32	5000	0	100	87.32	5000
LR	32.43	98.03	89.71	6523	10.52	99.1	87.87	5481
Decision tree	81.43	96.74	94.8	8908	74.55	96.18	93.44	8536
RF	*94.11*	*99.31*	*98.65*	*9671*	*90.71*	*96.76*	*95.99*	*9373*

Table 16 The average results of the tenfold cross-validation method for oversampling (200%)

Classifier	*Full data*				*Reduced feature data*			
	Sensitivity	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	72.62	91.97	88.51	8229	70.29	92.33	88.39	8131
RBF	19.69	96.65	82.89	5817	67.76	91.5	87.25	7963
SVM	12.54	98.08	82.79	5531	0	100	82.12	5000
LR	70.85	94.22	90.04	8253	75.08	93.88	90.52	8448
Decision tree	95.74	96.58	96.43	9660	81.03	95.72	93.09	8837
RF	*99.2*	*99.27*	*99.26*	*9923*	*95.10*	*96.6*	*96.33*	*9585*

Table 17 The average results of the tenfold cross-validation method for oversampling (300%)

Classifier	*Full data*				*Reduced feature data*			
	Sensitivity	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	75.29	91.46	87.82	8337	78.36	89.48	86.89	8392
RBF	49.57	90.11	80.99	6984	70.63	85.41	82.08	7802
SVM	78.04	84.37	82.95	8120	79.54	80.95	80.63	8024
LR	75.89	92.38	88.67	8413	78.54	90.77	88.02	8465
Decision tree	99.37	96.6	97.22	9798	86.62	95.19	93.26	9090
RF	*100*	*99.13*	*99.32*	*9956*	*95.75*	*96.66*	*96.46*	*9620*

5.2.5 Combination of undersampling and oversampling

Tables 18 and 19 show the results of the combination of undersampling and oversampling with various proportions. Table 18 presents the results of the combination of 25% undersampling and 100% oversampling, where RF topped for both the full and feature-selected datasets with 98.0% sensitivity, 94.43% specificity and 97.01% accuracy for the full dataset and 93.61% sensitivity, 94.78% specificity and 94.35% accuracy for the feature-selected dataset. Table 19 presents the results of the combination of 50% undersampling and 200% oversampling, where RF ranked at the top for the full and feature-selected datasets with 99.66% sensitivity, 98.45% specificity and 98.81% accuracy for the full dataset and for the feature-selected dataset, 99.34% sensitivity, 95.87% selectivity and 95.71% accuracy.

It is observed from our experiments that RF and the decision tree have given the best performance, irrespective of the method used. But other classifiers performed very badly, in the cases, of the full dataset and the dataset with reduced features. However, these classifiers yielded significantly improved results when they were supplied by the undersampling and oversampling combinations and SMOTE. For the hold-out method, the data is first split into 70:30 ratios for training and testing. Only the training data are SMOTED using the SMOTE function that is available in KNIME and is supplied for training. The test data is then supplied in the original form to the predictor for predicting the output of the given data. A similar procedure is carried out for undersampling, oversampling and the combination of undersampling and oversampling methods, irrespective of the full dataset or the feature-selected dataset. For the tenfold

cross-validation method, the original data is SMOTED first and then split into ten folds. A similar method is applied for undersampling, oversampling and the combination of both and also for the feature-selected data. The average values of sensitivity and specificity for each classifier over all of the folds are calculated.

Table 18 The average results of the tenfold cross-validation method for combination (25% + 100%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	78.64	89.02	85.21	8383	78.99	88.99	85.32	8399
RBF	75.14	79.23	77.73	7718	75.94	82.30	79.97	7912
SVM	80.43	75.55	77.36	7799	79.54	80.71	80.28	8012
LR	78.89	88.32	84.86	8360	79.24	85.95	83.43	8259
Decision tree	90.16	90.93	90.65	9054	80.68	94.06	89.15	8737
RF	*98.0*	*94.43*	*97.01*	*9621*	*93.61*	*94.78*	*94.35*	*9419*

Table 19 The average results of the tenfold cross-validation method for combination (50% + 200%)

	Full data				*Reduced feature data*			
Classifier	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*	*Sensitivity*	*Specificity*	*Accuracy*	*AUC*
MLP	75.31	91.23	86.41	8327	78.14	90.47	86.73	8430
RBF	70.65	84.23	80.11	7744	73.81	84.6	81.24	7920
SVM	78.7	82.89	81.62	8079	81.03	78.74	79.43	7988
LR	78.27	89.79	86.29	8403	79.04	87.44	84.89	8324
Decision tree	97.53	94.2	95.21	9586	82.50	94.90	91.14	8870
RF	*99.66*	*98.45*	*98.81*	*9905*	*99.34*	*95.87*	*95.71*	*9560*

All of the experiments have been done using the KNIME (1.2.1) tool and the Waikato Environment for Knowledge Analysis (WEKA) (3.5) tool. KNIME is an open source tool that can be downloaded from their website.[5] The hold-out methods are performed in KNIME, while the experiments with tenfold cross-validation method are done in WEKA. WEKA is also a free open source tool obtained from their website.[6] The SMOTING of the data is done with the SMOTE function available in KNIME. The resulting data is exported into Comma Separated Values (CSV) file format, which is used for SMOTE experiments that are carried out in WEKA.

With the help of AUC, we selected only those classifiers which gave AUC value higher than 5000 and used them for building the ensemble. Table 20 presents the majority voting of the various techniques and methods that were used and their performance comparisons. In majority voting, we observed that the original full dataset fared poorly. However, there is a significant improvement in sensitivity when the data are SMOTED and when supplied with a combination of undersampling and oversampling. It is evident from Table 20 that the combination of 25% undersampling and 100% oversampling produced a good prediction rate with 80.73% sensitivity, 89.26% specificity and 88.68%

accuracy for the full dataset, whereas for the feature-selected dataset, the combination of 50% undersampling and 200% oversampling produced a good prediction rate with 80.39% sensitivity, 87.98% specificity and 87.46% accuracy. For the tenfold cross-validation method, the SMOTE method produced good results for the full and feature-selected datasets. The 92.37% sensitivity, 91.40% specificity and 91.90% accuracy was achieved for the full dataset and for the feature-selected dataset, 81.68% sensitivity, 89.79% specificity and 85.74% accuracy has been achieved. The hold-out method has given very low sensitivity for the original full dataset. It is observed that the SMOTING of data indeed helped in obtaining more accurate results. The AUC for the different classifiers in various cases is presented in Tables 2 to 19.

Table 20 Majority voting

Method	*Data*		*Sensitivity*	*Specificity*	*Accuracy*
Hold-out (70:30)	Original	Full	0.33	100	93.25
		Feature-selected	0.66	99.97	93.25
	SMOTE	Full	78.07	92.68	91.69
		Feature-selected	80.06	89.40	88.77
	Undersampling	Full	75.74	93.67	92.46
	(25%)	Feature-selected	74.41	94.81	93.43
	Undersampling	Full	20.93	99.54	94.21
	(50%)	Feature-selected	68.43	96.42	94.53
	Oversampling	Full	13.95	99.78	93.97
	(100%)	Feature-selected	6.97	99.66	93.38
	Oversampling	Full	57.14	99.27	96.42
	(200%)	Feature Selected	69.43	96.93	95.07
	Oversampling	Full	72.09	94.20	92.71
	(300%)	Feature-selected	79.73	90.73	89.98
	Combination	*Full*	*80.73*	*89.26*	*88.68*
	(25% + 100%)	Feature-selected	80.39	85.52	85.17
	Combination	Full	77.74	91.33	90.41
	(50% + 200%)	*Feature-selected*	*80.39*	*87.98*	*87.46*
Tenfold Cross-validation	Original	Full	0.79	99.98	93.27
		Feature Selected	0.09	100	93.24
	SMOTE	*Full*	*92.37*	*91.40*	*91.90*
		Feature-selected	*81.68*	*89.79*	*85.74*
	Undersampling	Full	73.45	94.72	89.94
	(25%)	Feature-selected	76.94	92.99	89.38
	Undersampling	Full	26.34	99.59	90.31
	(50%)	Feature-selected	23.45	98.71	89.17
	Oversampling	Full	27.19	99.81	90.61
	(100%)	Feature-selected	18.81	99.42	89.21
	Oversampling	Full	69.32	99.08	93.76
	(200%)	Feature-selected	74.45	96.72	92.74
	Oversampling	Full	77.51	94.84	90.94
	(300%)	Feature-selected	78.79	91.64	88.75
	Combination	Full	79.29	90.76	86.54
	(25% + 100%)	Feature-selected	79.09	89.28	85.54
	Combination	Full	78.54	92.57	88.31
	(50% + 200%)	Feature-selected	79.04	90.67	87.14

5.3 Rules extraction from the decision tree

The rules for predicting the customers' loyalties are extracted using the decision tree (J48). Initially, the dataset has been split into a training set with 80% data and a validation set with 20% data. Within the training data, tenfold cross-validation was performed. For the SMOTED data, first, the training data was SMOTED and supplied for training. From the predictions obtained on the training data in each fold, the fold that yielded the highest accuracy and sensitivity was noted and the rules that were generated by that fold were extracted and coded for testing on the validation set, which comprises 20% of the original data. This procedure was adopted to give recommendations in the form of 'if-then' rules to the management. The rules that were obtained for various methods are presented in Tables 21 and 22. Table 21 presents the rules that were obtained by applying the original data. The rules used here are from Fold 7, which gave 96.79% accuracy and 58.95% sensitivity for the validation set. Table 22 presents the rules that were obtained by applying reduced feature-selected data. The rules used here are from Fold 8, which gave 96.42% accuracy and 47.67% sensitivity for the validation set.

Table 21 The rules for the tenfold cross-validation method using original full data

Rule no.	*Rule antecedents*	*Consequent*
1	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 <= 93.95 then	Churner
2	If CRED_T<= 593.19 and NCC_T <= 0 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T <= –3.54 then	Nonchurner
3	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= -0.01 and MAR_T-1 <= –1.31 and T_WEB T <= 3 then	Nonchurner
4	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= -0.01 and MAR_T-1 <= –1.31 and T_WEB T > 3 then	Churner
5	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= –0.01 and MAR_T-1 > –1.31 then	Churner
6	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= 0.07 and MAR_T-6 > –0.01 and CRED_T-1 <= 93.96 and AGE <= 51 then	Nonchurner
7	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= 0.07 and MAR_T-6 > –0.01 and CRED_T-1 <= 93.96 and AGE > 51 then	Churner
8	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= 0.07 and MAR_T-6 > –0.01 and CRED_T-1 > 93.96 and MAR_T-2 <= 0.05 then	Churner
9	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 <= 0.07 and MAR_T-6 > –0.01 and CRED_T-1 > 93.96 and MAR_T-2 > 0.05 then	Nonchurner
10	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 > 0.07 and MAR_T-1 <= 0.01 then	Churner

Table 21 The rules for tenfold cross-validation using original full data (continued)

Rule no.	*Rule antecedents*	*Consequent*
11	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 > 0.07 and MAR_T-1 > 0.01 and AGE <= 32 and N_EDUC <= 2 then	Nonchurner
12	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 > 0.07 and MAR_T-1 > 0.01 and AGE <= 32 and N_EDUC > 2 then	Churner
13	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 <= 0 and MAR_T > –3.54 and MAR_T-6 > 0.07 and MAR_T-1 > 0.01 and AGE > 32 then	Nonchurner
14	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 > 0 and MAR_T-3 <= –2.55 then	Nonchurner
15	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T <= 0.04 and CRED_T-2 > 93.95 and NCC_T-2 > 0 and MAR_T-3 > –2.55 then	Churner
16	If NCC_T <= 0 and MAR_T > 0.04 and CRED_T<= 592.75 and MAR_T <= 3.61 then	Churner
17	If NCC_T <= 0 and MAR_T > 3.61 and CRED_T-2 <= 94.04 and T_WEB T <= 0 and CRED_T-1 <= 93.48 and CRED_T<= 590.23 then	Churner
18	If NCC_T <= 0 and CRED_T<= 592.75 and MAR_T > 3.61 and CRED_T-2 <= 94.04 and T_WEB T <= 0 and CRED_T-1 <= 93.48 and CRED_T> 590.23 then	Nonchurner
19	If NCC_T <= 0 and CRED_T<= 592.75 and MAR_T > 3.61 and CRED_T-2 <= 94.04 and T_WEB T <= 0 and CRED_T-1 <= 93.95 and CRED_T-1 > 93.48 then	Churner
20	If NCC_T <= 0 and CRED_T<= 592.75 and MAR_T > 3.61 and CRED_T-2 <= 94.04 and T_WEB T <= 0 and CRED_T-1 > 93.95 then	Nonchurner
21	If NCC_T <= 0 and CRED_T<= 592.75 and MAR_T > 3.61 and CRED_T-2 <= 94.04 and T_WEB T > 0 then	Nonchurner
22	If NCC_T <= 0 and CRED_T<= 592.75 and MAR_T > 3.61 and CRED_T-2 > 94.04 then	Nonchurner
23	If NCC_T <= 0 and MAR_T > 0.04 and CRED_T> 592.75 and MAR_T-2 <= –1.47 and CRED_T<= 592.94 then	Churner
24	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T > 0.04 and MAR_T-2 <= –1.47 and CRED_T> 592.94 then	Nonchurner
25	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T > 0.04 and CRED_T> 592.75 and MAR_T-2 > –1.47 and MAR_T-4 <= –2.7 and CRED_T-2 <= 94.01 then	Churner
26	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T > 0.04 and CRED_T> 592.75 and MAR_T–2 > –1.47 and MAR_T–4 <= –2.7 and CRED_T–2 > 94.01	Nonchurner
27	If CRED_T<= 593.19 and NCC_T <= 0 and MAR_T > 0.04 and CRED_T> 592.75 and MAR_T–2 > –1.47 and MAR_T–4 > –2.7 then	Nonchurner
28	If CRED_T<= 593.19 and NCC_T > 0 then	Nonchurner
29	If CRED_T> 593.19 then	Nonchurner

Table 22 The rules for the tenfold cross-validation using reduced feature data

Rule no.	*Rule antecedents*	*Consequent*
1	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T<= 591.75 and CRED_T–1 <= 92.89 then	Churner
2	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 <= 93.42 and CRED_T<= 591.75 and CRED_T–1 > 92.89 then	Nonchurner
3	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 <= 93.42 and CRED_T<= 592.34 and CRED_T> 591.75 then	Churner
4	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 <= 93.42 and CRED_T> 592.34 then	Nonchurner
5	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.42 and CRED_T–1 <= 93.95 and CRED_T<= 592.18 then	Nonchurner
6	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.42 and CRED_T–1 <= 93.95 and CRED_T<= 592.36 and CRED_T> 592.18 then	Churner
7	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.42 and CRED_T<= 592.67 and CRED_T> 592.36 and CRED_T–1 <= 93.84 then	Churner
8	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 <= 93.95 and CRED_T> 592.36 and CRED_T–1 > 93.84 and CRED_T<= 592.6 then	Nonchurner
9	If NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 <= 93.95 and CRED_T<= 592.67 and CRED_T–1 > 93.84 and CRED_T> 592.6 then	Churner
10	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T–1 > 93.42 and CRED_T–1 <= 93.95 and CRED_T> 592.67 and CRED_T–2 <= 93.85 then	Nonchurner
11	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.42 and CRED_T–1 <= 93.95 and CRED_T> 592.67 and CRED_T–2 > 93.85 then	Churner
12	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.95 and NCC_T–2 <= 0 then	Nonchurner
13	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T–2 <= 93.95 and CRED_T–1 > 93.95 and NCC_T–2 > 0 then	Churner
14	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T<= 592.36 then	Nonchurner
15	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T> 592.36 and NCC_T–2 <= 0 and CRED_T<= 592.74 and CRED_T–2 <= 100.4 then	Nonchurner
16	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T> 592.36 and NCC_T–2 <= 0 and CRED_T<= 592.74 then	Churner
17	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T<= 592.75 and NCC_T–2 <= 0 and CRED_T> 592.74 then	Churner
18	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T> 592.36 and CRED_T<= 592.75 and NCC_T–2 > 0 then	Churner
19	If NCC_T <= 0 and CRED_T–2 > 93.95 and CRED_T<= 592.76 and CRED_T> 592.75 and CRED_T–2 <= 93.96 then	Nonchurner

Table 22 The rules for the tenfold cross-validation using reduced feature data (continued)

Rule no.	*Rule antecedents*	*Consequent*
20	If NCC_T <= 0 and CRED_T<= 592.76 and CRED_T> 592.75 and CRED_T−2 > 93.96 and CRED_T−1 <= 93.96 then	Nonchurner
21	If NCC_T <= 0 and CRED_T<= 592.76 and CRED_T> 592.75 and CRED_T−2 > 93.96 and CRED_T−1 <= 93.98 and CRED_T−1 > 93.96 then	Churner
22	If NCC_T <= 0 and CRED_T<= 592.76 and CRED_T> 592.75 and CRED_T−2 > 93.96 and CRED_T−1 > 93.98 then	Nonchurner
23	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−2 > 93.95 and CRED_T> 592.76 and CRED_T−1 <= 93.96 and NCC_T−1 <= 0 then	Nonchurner
24	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−2 > 93.95 and CRED_T> 592.76 and CRED_T−1 <= 93.96 and NCC_T−1 > 0 then	Churner
25	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−2 > 93.95 and CRED_T> 592.76 and CRED_T−1 <= 94.21 and CRED_T−1 > 93.96 then	Churner
26	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−2 > 93.95 and CRED_T> 592.76 and CRED_T−1 > 94.21	Nonchurner
	And NCC_T−2 <= 0 and CRED_T−2 <= 105.16 then	
27	If NCC_T <= 0 and CRED_T> 592.76 and CRED_T−1 > 94.21 and NCC_T−2 <= 0 and CRED_T−2 > 105.16 and CRED_T<= 592.89 then	Churner
28	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−1 > 94.21 and NCC_T−2 <= 0 and CRED_T−2 > 105.16 and CRED_T> 592.89 then	Nonchurner
29	If CRED_T<= 593.02 and NCC_T <= 0 and CRED_T−2 > 93.95 and CRED_T> 592.76 and CRED_T−1 > 94.21 and NCC_T−2 > 0 then	Churner
30	If CRED_T<= 593.02 and NCC_T > 0 then	Nonchurner
31	if CRED_T> 593.02 then	Nonchurner

6 Conclusions

In this paper, we conducted the most comprehensive investigation by far into the credit card churn prediction problem in banks by resorting to data mining. We proposed an ensemble system with majority voting that involved MLP, LR, decision tree (J48), RF, RBF network and SVM. This system gives the best result when the unbalanced original data is SMOTED, as well as for the combination of undersampling and oversampling. Among the various methods tested, the results show that the tenfold cross-validation method on SMOTED data has produced excellent results with 92.37% sensitivity, 91.40% specificity and 91.90% overall accuracy. It is observed that the RF yielded good results for the full feature-selected datasets. We also generated a set of 'if-then' rules using a decision tree J48. This set of rules could act as an 'early warning' system for churn modelling, prediction and management.

References

A SAS White Paper (2001) 'Customer relationship management in banking', www.sas.com.

Au, W-H., Chan, K.C.C. and Yao, S. (2003) 'A novel evolutionary data mining algorithm with applications to churn prediction', *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 6.

Bloemer, J., Ruyter, K.D. and Peeters, P. (1998) 'Investigating drivers of bank loyalty: the complex relationship between image, service quality and satisfaction', *International Journal of Bank Marketing*, Vol. 16, No. 7, pp.276–286.

Bolton, R.N. (1998) 'A dynamic model of the duration of the customer's relationship with a continuous service provider: the role of satisfaction', *Marketing Science*, p.45.

Bolton, R.N., Kannan, P.K. and Bramlett, M.D. (2000) 'Implications of loyalty program membership and service experiences for customer retention and value', *Journal of the Academy of Marketing Science*, Vol. 28, pp.95–108.

Buckinx, W. and Van den Poel, D. (2005) 'Customer base analysis: partial defection of behaviorally loyal clients in a non-contractual FMCG retail setting', *European Journal of Operational Research*, Vol. 164, pp.252–268.

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

Business Intelligence Cup that was organized by the University of Chile (2004) http://www.tis.cl/bicup_04/text_bicup/BICUP/202004/20public/20data.zip.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2004) 'SMOTE: synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research*, Vol. 16, pp.324–357.

Chu, B-H., Tsai, M-S. and Ho, C-S. (2007) 'Toward a hybrid data mining model for customer retention', *Knowledge-Based Systems*, Vol. 20, pp.703–718.

Estevez, P.A., Held, C.M. and Perez, C.A. (2006) 'Subscription fraud prevention in telecommunications using fuzzy rules and neural networks', *Expert Systems with Applications*, Vol. 31, pp.337–344.

Euler, T. (2005) *Churn Prediction in Telecommunications Using MiningMart*, http://www-ai.cs.uni-dortmund.de.

Fawcett, T. (2006) 'An introduction to ROC analysis', *Pattern Recognition Letters*, Vol. 27, pp.861–874.

Ferreira, J.B., Vellasco, M., Pacheco, M.A. and Barbosa, C.H. (2004) 'Data mining techniques on the evaluation of wireless churn', *ESANN'2004 Proceedings – European Symposium on Artificial Neural Networks*, Bruges, Belgium, 28–30 April, d-side publi., pp.483–488, ISBN: 2-930307-04-8

Hadden, J., Tiwari, A., Roy, R. and Ruta, D. (2005) 'Computer assisted customer churn management: state-of-the-art and future trends', *Computers & Operations Research*, Vol. 34, pp.2902–2917.

Hadden, J., Tiwari, A., Roy, R. and Ruta, D. (2006) 'Churn prediction: does technology matter?', *International Journal of Intelligent Technology*, Vol. 1, No. 1.

Hu, X. (2005) 'A data mining approach for retailing bank customer attrition analysis', *Applied Intelligence*, Vol. 22, pp.47–60.

Hung, S-Y. and Yen, D.C. (2006) 'Applying data mining to telecom churn management', *Expert Systems with Applications*, Vol. 31, pp.515–524.

Karakostas, B., Kardaras, D. and Papathanassiou, E. (2005) 'The state of CRM adoption by the financial services in the UK: an empirical investigation', *Information & Management*, Vol. 42, pp.853–863.

Lariviere, B. and Van den Poel, D. (2004a) 'Customer attrition analysis for financial services using proportional hazard models', *European Journal of Operational Research*, Vol. 157, pp.196–217.

Lariviere, B. and Van den Poel, D. (2004b) '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*, Vol. 27, pp.277–285.

Larivie`re, B. and Van den Poel, D. (2005) 'Predicting customer retention and profitability by using random forests and regression forests techniques', *Expert Systems with Applications*, Vol. 29, pp.472–484.

Lejeune, M.A.P.M. (2001) 'Measuring the impact of data mining on churn management', *Electronic Networking Applications and Policy*, Vol. 11, No. 5, pp.375–387.

Lu, J. (2008) 'Modeling customer lifetime value using survival analysis – an application in the telecommunications industry', *Data Mining Techniques*, SUGI 28, pp.120–128.

Michael, A.J., Mothersbaugh, D.L. and Beatty, S.E. (2000) 'Switching barriers and repurchase intentions in services', *Journal of Retailing*, Summer, Vol. 76, No. 2.

Mols, N.P. (1998) 'The behavioral consequences of PC banking', *International Journal of Bank Marketing*, Vol. 16, No. 5, pp.195–201.

Mozer, M.C., Wolniewicz, R., Grimes, D.B., Johnson, E. and Kaushansky, H. (2000) 'Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry', *IEEE Transactions of Neural Networks*, May, Vol. 11, No. 3.

Mutanen, T. (2006) 'Customer churn analysis – a case study', Research Report No. VTTR0118406, http://www.vtt.fi/inf/julkaisut/muut/2006/customer_churn_case_study.pdf (retrieved on 10 July 2008).

Neslin, S., Gupta, S., Kamakura, W., Lu, J. and Mason, C. (2004) 'Defection detection: improving predictive accuracy of customer churn models', Working paper, Teradata Center at Duke University.

Ravi, V., Kumar, P.R., Srinivas, E.R. and Kasabov, N.K. (2007) 'A semi-online training algorithm for the radial basis function neural networks: applications to bankruptcy prediction in banks', in V. Ravi (Ed.) *Advances in Banking Technology and Management: Impact of ICT and CRM*, IGI Global Inc., USA.

Revett, K., Gorunescu, F. and Gorunescu, M. (2006) 'An investigation into a beta-carotene/retinol dataset using rough sets', *Proceedings of ECML/PKDD Workshop on Practical Data Mining: Applications, Experiences and Challenges*, Berlin, 22 September.

Richeldi, M. and Perrucci, A. (2002) *Churn Analysis Case Study Enabling End-User Datawarehouse Mining*, December.

Smith, K.A. and Gupta, J.N.D. (2000) 'Neural networks in business: techniques and applications for the operations researcher', *Computers & Operations Research*, Vol. 27, pp.1023–1044.

Ultsch, A. (2001) 'Emergent self-organising feature maps used for prediction and prevention of churn in mobile phone markets', *Journal of Targeting, Measurement and Analysis for Marketing*, Vol. 10, No. 4, pp.314 – 324.

Wezel, M.V. and Potharst, R. (2000) 'Improved customer choice predictions using ensemble methods', *European Journal of Operational Research*, Vol. 181, pp.436–452.

Yuan, S-T. and Chang, W-L. (2001) 'Mixed-initiative synthesized learning approach for web-based CRM', *Expert Systems with Applications*, Vol. 20, pp.187–200.

Notes

1 www.crisp-dm.org/index (2007).

2 http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm#remarks (retrieved 22 July 2008).

3 http://www.dtreg.com/svm.htm

4 www.utstat.utoronto.ca/~radford/sta414.S06/slides7b.ps (retrieved on 10 July 2008).

5 http://www.KNIME.org

6 www.cs.waikato.ac.nz