A PREDICTIVE MODELING APPROACH FOR OPTIMAL PREDICTION OF THE PROBABILITY OF CREDIT CARD DEFAULT

SUBMITTED IN FULFILMENT OF THE ILLINOIS STATE UNIVERSITY

ANNUAL STATISTICAL COMPETITION

ON APRIL 1, 2017

ERNEST FORSON ABOAGYE

MS MATHEMATICS (ACTUARIAL SEQUENCE), ILLINOIS STATE UNIVERSITY

Abstract

The growth of credit card debt in the twenty-first century has garnered much attention for researchers to focus attention on various aspects of this unique instrument. This has led to testing the stability of models that forecast personal default and credit risk models more generally. This paper used a real-life data of credit card clients to investigate the predictive accuracy of some data mining techniques on the probability of default, a two-class classification problem. The misclassification rate and area under the receiver operating characteristic curves are the two performance measures used to determine the accuracy of these ten techniques studied. Support vector machines, an advanced predictive modeling technique, and the widely known logistic regression are two main methods that proved to have high objective accuracies for both performance measures. Some variables proved more important than others and using the most significant predictors for the data mining techniques gave comparably close prediction results; a few accuracies improved after variable selection. The logistic regression produced the same error rate and AUC before and after variable selection.

[**Key Words**: Credit Card, Risk, Data Mining, Misclassification Rate, Predictive Modeling, Logistic Regression, AUC, Support Vector Machines, Variable Selection]

1. Introduction

The health of the credit card industry is best measured not by the number of people with credit cards, but rather the number who pay their bills. Credit card debt statistics speak to the financial health of American households and can foretell some serious bubbles that may trigger constriction across lending markets. From that perspective, the fact that the U.S consumers racked up $60.4 billion in credit debt during the fourth quarter of 2016 represents a serious cause for concern.[1] The number of credit card users increase as the population ages into adulthood and the risk of default becomes increasingly worrisome to the banks and financial companies that issue credit cards. Per a 2014 report by the Urban Institute, roughly 1 out of 20 Americans with credit files are at least 30 days late on a credit card or other non-mortgage account (Ratcliff et. al. 2014). Bad payment habits can lead to more fees, lower credit scores and, in some cases, bankruptcy. Personal bankruptcy filings have increased substantially over the last two decades from 0.35% to 1.4% per year, and such high levels in bankruptcy rates, according to Lopes (2008), may put the credit market stability at risk.

Credit card delinquency and charge-off rates have to be assessed in order to get an accurate sense of the consumer debt situation. Dunn and Kim (1999) point out how banks and financial planners have taken the issue of credit card default seriously after default and personal bankruptcy began to increase sharply post 1995 despite the lull in credit activity in the early 1990s. The growth of the credit card debt in the U.S. economy in the twenty-first century has garnered much attention for researchers to focus attention on various aspects of this unique instrument which has led to testing the stability of models that forecast personal default and credit risk models more generally. According to Gross and Souleles (2002) a risk effect and a demand effect account for the explanation of these trends. With the risk effect, they explain that additional credit was advanced to less creditworthy borrowers because based on their observations, the most important factors behind the rise in credit default can be attributed to the growth in the number of credit card offers and the credit limit size. In that paper, they cite the increased willingness over the years of cardholders to default as the demand effect. Unlike the risk effect, the demand effect represents a change in the relationship between default and the variables that lenders typically use to predict default. Lopes (2008) proved that the probability of default is decreasing as the number and level of education increases, ceteris paribus.

Risk management propounds several techniques to manage an efficient system against market risk in today's fast-changing financial markets. Understanding the various ways in which lenders mitigate the default risk is the key to explaining some of the main features of credit markets. Risk prediction is of great importance and in the credit world comes with the prediction of the probability of default which is an essential part of business intelligence in the financial institutions. Recent studies indicate that underestimating this important component might threaten the stability and smooth running of the markets (Titan and Tudor, 2011). To do

this requires analytical processes and prediction models that feed on financial statements, customer transactions and repayment records, among others in order to predict business performance through minimizing credit risk deficiencies to decrease default. In their paper, Titan and Tudor state that the result of predictive accuracy of the estimated probability of default is more valuable than the standard binary classification: good or bad clients. General modeling methods for optimal probability predictions over future observations have been studied and simulation results on both artificial and practical datasets have proved supportive in helping to predict whether or not one would default on the payment of his credit card balance when it is due. Over the years, researchers in the field have supplemented credit scoring algorithms with linear modeling methods to enhance the accuracy of prediction. A relatively newer concept to solve the problem of default prediction with far greater degree of accuracy is data mining and visualization. Data mining can be defined as the process of finding previously unknown patterns and trends in databases and using that information to build predictive models. According to Koh and Tan (2011), data mining is not a new concept because it has hitherto been used by manufacturers, for quality control and maintenance scheduling; retailers, for market segmentation and store layout; and financial institutions, for fraud detection. The most common and important applications in data mining probably involve predictive modeling, a concept which in recent years seem to get more interesting as new and exciting challenges spring up which require efforts to closely equate theoretical applications with real world experiences. Much of these can be seen in healthcare where data mining is becoming increasingly popular, if not increasingly essential, a motivation of this coming from the upsurge in medical insurance fraud and abuse. Healthcare insurers in a bid to cut down their losses resort to data mining tools to enable them find, track, and penalize culprits.

There have been significant applications of supervised learning methods in coming out with very good predictive models. Albashrawi (2016) summarizes the work by researchers over a decade from 2004 to 2015 in which they used various data mining techniques to detect financial fraud. He made mention of methods such as K-Nearest Neighbors, Discriminant Analysis, Naïve Bayes, Neural Networks, Logistic regressions, Decision trees, CART, Support Vector Machines, among a host of others. Of the financial fraud data analyzed, Albashrawi mentions credit card default being analyzed by different researchers using at least one of these methods. It is worthy to note that no two of the researchers had the same data mining technique being the best performing one.

This research, aimed at the case of customers' default payments in Taiwan, also attempts to use multiple predictive modeling techniques to assess the predictive accuracy of the probability of credit card default, and in doing so will seek to answer questions such as:

a. Which, of the ten predictive modeling techniques discussed in this paper, works best to accurately predict the probability of credit card default?

b. What major factors would help detect whether one would default in payment? And how would theses impact the issuance of future credit cards to new users?

The remainder of the paper is structured as follows: Section 2 looks at some related studies on data mining, their applications to financial data and the probability of credit default prediction. A discussion is made on the statistical concepts employed to analyze the data used for this paper. The next section discusses the data set, primary attributes and the methodology for assessing the performance of the data mining techniques discussed. The results of applying the techniques to the data set are presented and analyzed after. These and other empirical findings are explored in section 4. The final section contains a brief conclusion on the findings and provides some remarks for exploring the topic further.

2. Related Studies and Data Mining Techniques

Data mining is defined by Turban and Aronson (2007) as "the process that uses statistical, mathematical, artificial intelligence and machine-learning techniques to extract and identify useful information and subsequently gain knowledge from large databases". So practically, data mining is used to extract information from available data. Despite the host of data mining techniques and applications in our days, studies into credit card default detection looks pristine but for a few reported studies into credit card fraud detection. A possible reason for this is the dearth in data for research. Yeh and Lien (2009) employ six data mining techniques to examine the predictive accuracy of default of credit card clients from 25000 payment observations. Using the Sorting Smoothing technique as a basis to select the best method, Yeh and Lien conclude on the artificial neural network as the best performing technique for predicting what they call the "real" probability of default with reference to performance measures such as the R-squared, regression intercept and coefficient, and based on that they make a bold claim that the artificial neural network should be employed to score clients instead of logistic. Most other papers that have dealt into credit fraud detection have also examined artificial neural networks; which is not surprising given its vast popularity in the 1990s (Bhattacharyya, Jha, Tharakunnel and Westland, 2011; Aleskerov, Freisleben and Rao 1998; Brause and Hepp, 1999). Support vector machines and random forests have been observed in recent years to show superior performance across different applications (Statnikov, Wang, Aliferis, 2008; Whitrow, Hand, Juszczak, Weston, Adams, 2009) and according to Titan and Tudor (2011) artificial neural networks, discriminant analysis, K-nearest neighbors and logistic regression are the most important techniques used for predictive default probability. Kou, Chang-Tien, Sirwongwattana and Huang (2004) survey some analytical techniques used for fraud detection and they go head to review some research done into credit card fraud detection using data

mining techniques. It is worthy to note that no two of the previous research done into credit card default prediction or fraud detection yielded the same best performing data mining technique. Artificial immune systems, random forests, support vector machines, K-means clustering, neural networks, and Bayesian learning were among the few best performing techniques (Albashrawi, 2016; Chen, Chen and Lin, 2006; Gadi, Wand, and do Lago, 2008; Bhattacharyya, Jha, Tharakunnel and Westland, 2011; Wu, Xiong and Cheng 2010; Yeh and Lien, 2009; Panigrahi, Kundu, Sural, Majumdar, 2009). Also, it is surprising none of them consider the importance of variables in the accuracy of their techniques.

In addition to combining several of the techniques used in all these papers reviewed in the literature for the analysis, this paper fills an important void of considering the importance of variable selection in influencing the predictive accuracy of probability of default. The paper evaluates ten data mining approaches for assessing the optimal prediction of the credit card default problem. Among the techniques used is the most popularly used in the literature for classification data, logistic regression, together with four advanced approaches: random forests, generalized boosted models, support vector machines and artificial neural networks.

2.1 Logistic Regression

Classification models are used for categorical response variables and since "Default or Not" is a qualitative binary response variable, let us study some of the classification models used to explore accuracy in this paper. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is a cumulative logistic distribution. The conditional distribution of y given x is a Bernoulli distribution because the dependent variable is binary. Logistic regression, a special case of linear regression models, is an alternative to Fisher's 1936 method, linear discriminant analysis, but does not require the multivariate normal assumption of the latter. It is well-understood, easy to use, and remains one of the most commonly used for data-mining in practice and therefore provides a useful baseline for comparing performance of newer methods (Bhattacharyya, Jha, Tharakunnel and Westland, 2011). The major advantage of this approach is that it can produce a simple probabilistic formula of classification.

2.2 Discriminant Analysis

Discriminant Analysis, also known as Fisher's rule is a classification technique which projects onto a line an n-dimensional data by maximizing between-class mean and minimizing within-class variance, and performs classification in this one-dimensional space. We have two common forms of Discriminant Analysis used in data mining: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). According to James, Witten, Hastie and Tibshirani (2013), the Linear Discriminant Analysis (LDA) assumes the predictor $\boldsymbol{X} =$

$(X_1, X_2, \dots, X_p)$ is drawn from a multivariate Gaussian distribution with class-specific mean vector and a common covariance matrix. Formally, the multivariate Gaussian density is defined as

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)\right)$$

We need this model because when the classes are well-separated, the parameter estimates for the logistic regression model, the most used traditional classification technique, are surprisingly unstable. LDA does not suffer this problem and is even more popular with more than two response classes. Like the LDA, the Quadratic Discriminant Analysis (QDA) classifier results from assuming the observations from each class are drawn from a Gaussian distribution. However, unlike LDA, QDA assumes that each class has its own covariance matrix.

2.3 Ridge and Lasso Regression

Shrinkage methods involve fitting a model containing all p predictors but the estimated coefficients are shrunken toward zero relative to their least squares estimates. Hence, shrinkage methods can also perform variable selection. The shrinkage has the effect of reducing variance. The two best-known techniques for shrinking the regression coefficients towards zero are Ridge regression and the Lasso.

Ridge regression is very similar to least squares, except that the coefficients are estimated by minimizing a slightly different quantity. The Ridge coefficient estimates are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

where $\lambda \geq 0$ is a tuning parameter to be determined. As with least squares, Ridge regression seeks coefficient estimates that fit the data well, by making the first term small. However, the second term, $\lambda\sum_{j=1}^{p}\beta_j^2$, called a shrinkage penalty is small when the values of β_j , $j = 1, \dots, p$ are close to zero, and so it has the effect of shrinking the estimates of β_j towards zero. When $\lambda = 0$ the penalty term has no effect, and Ridge regression will produce the least squares estimates.

In some models, only a few important variables help to predict the response. However, Ridge regression will always generate a model involving all predictors, including redundant ones. Increasing the values of λ will tend to reduce the magnitudes of the coefficients, but will not result in exclusion of any of the variables. The Lasso is a relatively recent alternative to Ridge regression that overcomes this disadvantage. It has the same first term as the ridge formula above. But, in the case of the lasso the penalty term is $\lambda\sum_{j=1}^{p}|\beta_j|$ and this

penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large. Hence, much like best subset selection, the Lasso performs variable selection. As a result, models generated from the Lasso are generally much easier to interpret than those produced by Ridge. The Lasso regression yields sparse models – models that involve only a subset of the variables. (James, Witten, Hastie, and Tibshirani, 2013)

2.4 K – Nearest Neighbors

The *k*-Nearest Neighbors (KNN) algorithm is a non-parametric lazy learning method for supervised learning, where an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (Wikipedia, 2017). In learning systems, generalization performance is affected by a trade-off between the number of training examples and the capacity (e.g. the number of parameters) of the learning machine. The major advantage is that it is not required to establish predictive model before classification.

2.5 Random Forests

The popularity of decision tree models in data mining arises from their ease of use, flexibility in terms of handling various data attribute types, and interpretability. Single tree models, however, can be unstable and overly sensitive to specific training data. Ensemble methods seek to address this problem by developing a set of models and aggregating their predictions in determining the class label for a data point. Random decision forests (RF) are an ensemble learning method of classification (or regression) trees operated by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees (Wikipedia, 2017). Random forests correct decision trees' habit of overfitting to their training set. The training algorithm for random forests applies to the general technique of bootstrap aggregating, or bagging to tree learners, by using a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. Random forests combine the concepts of bagging, where individual models in an ensemble are developed through sampling with replacement from the training data, and the random subspace method, where each tree in an ensemble is built from a random subset of attributes (Bhattachcharya, Jha, Tharakunnel and Westland, 2011). Random forests are computationally efficient since each tree is built independently of the others. With large number of trees in the ensemble, they are also noted to be robust to overfitting and noise in the data. The number of attributes, p, used at a node and total number of trees T in the ensemble are user-defined parameters. The error rate for a random forest has been noted to depend on the correlation between trees and the strength of each tree in the ensemble, with lower correlation and higher strength giving lower error. Lower values of p correspond to lower correlation, but also lead to lower strength of individual trees. An optimal value for p can be experimentally determined. The number of random variables

randomly sampled as candidates at each node split for classification is $\sqrt{p}$. Attribute selection at a node is based on the Gini index, though other selection measures may also be used. Breiman (2001) proved random forests to have comparable performance to the other modern sophisticated techniques like support vector machines, boosting and artificial neural networks.

2.6 Generalized Boosted Models (Boosting)

Boosting is a machine learning ensemble meta-algorithm for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees (Breiman, 1996). When first introduced, the hypothesis boosting problem simply referred to the process of tuning a weak learner to a strong learner. While boosting is not algorithmically constrained, most boosting algorithms consist of iteratively learning weak classifiers with respect to a distribution and adding them to a final strong classifier. When they are added, they are typically weighted in some way that is usually related to the weak learners' accuracy. After a weak learner is added, the data are reweighted: examples that are misclassified gain weight and examples that are classified correctly lose weight (some boosting algorithms actually decrease the weight of repeatedly misclassified examples. Thus, future weak learners focus more on the examples that previous weak learners misclassified. In this research, a type of boosting called Adaptive Boosting (Adaboost) is employed as it has been proven to improve performance and is very popular. It is perhaps the most significant historically as it was the first algorithm that could adapt to the weak learners. With Adaboost the output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. Adaboost, with decision trees as weak learners, is often referred to as the best out-of-the-box classifier. It is sensitive to noisy data and outliers.

2.6 Support Vector Machines

Denoted by SVM in the literature, an SVM model is the representation of the points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. It is a non-probabilistic binary linear classifier. Bhattacharyya (2011) states that SVMs are statistical learning techniques that have been found to be very successful in a variety of classification tasks and that several unique features of these algorithms make them especially suitable for binary classification problems like predicting the probability of credit default. SVMs are linear classifiers that work in a high-dimensional feature space that is a non-linear mapping of the input space of the problem at hand. An advantage of working in a high-dimensional feature space is that, in many problems the non-linear classification task in the original input space becomes a linear classification task in the high-dimensional feature space. SVMs work in the high dimensional feature space without incorporating any additional

computational complexity. The simplicity of a linear classifier and the capability to work in a feature-rich space make SVMs attractive for fraud detection tasks where highly unbalanced nature of the data (fraud and non-fraud cases) make extraction of meaningful features critical to the detection of fraudulent transactions is difficult to achieve. The strength of SVMs comes from two important properties they possess — kernel representation and margin optimization. In SVMs, mapping to a high-dimensional feature space and learning the classification task in that space without any additional computational complexity are achieved by the use of a kernel function. A kernel function can represent the dot product of projections of two data points in a high-dimensional feature space. The high-dimensional space used depends on the selection of a specific kernel function. Radial kernel is used for running the analysis to compare the performance of the techniques in this paper as this was the optimal tuning parameter suggested by cross validation in R software. The second property of SVMs is the way the best classification function is arrived at. SVMs minimize the risk of overfitting the training data by determining the classification function (a hyper-plane) with maximal margin of separation between the two classes. This property provides SVMs very powerful generalization capability in classification.

2.7 Artificial Neural Networks

The goal of the neural network is to solve problems in the same way that the human brain would, although several neural networks are more abstract. Yang and Zheng (2009) explain artificial neural network (commonly called just neural network) as “an interconnected assemblage of artificial neurons that uses a mathematical or computational model of theorized mind and brain activity, attempting to parallel and simulate the powerful capabilities for knowledge acquisition, recall, synthesis, and problem solving”. Theoretically, artificial neural networks are highly robust in data distribution, and can handle incomplete, noisy and ambiguous data. They are well suited for modeling complex, nonlinear phenomena ranging from financial management, hydrological modeling to natural hazard prediction. For any neural computing, training time is always the biggest bottleneck and thus, every effort is needed to make training effective and affordable. Training time is a function of the complexity of the network topology which is ultimately determined by the combination of hidden layers and neurons. A trade-off is needed to balance the processing purpose of the hidden layers and the training time needed. One of the major developments in neural networks over the last decade is the model combining or ensemble modelling. A network without a hidden layer is only able to solve a linear problem. To tackle a nonlinear problem, a reasonable number of hidden layers is needed. A network with one hidden layer has the power to approximate any function provided that the number of neurons and the training time are not constrained (Hornik, 1993). But in practice, many functions are difficult to approximate with one hidden layer and thus, Flood and Kartam (1994) suggested using two hidden layers as a starting point. As a standard, two hidden layers were adopted in the analysis of this data. The artificial neural network plot for the complete data with two hidden layers is found in the Appendix. The plot shows the network interconnectedness and how the model activity functions to predict the probability of default in this case.

3. Data Set and Methodology

There is no standard definition of what 'default' means. By the terms of the credit card contract, a card user is technically in default if a minimum required payment is missed. To forecast probability of default is a major challenge and it needs intense study. To do this, this section describes the data used for training and testing the models and performance measures used. Data is collected from the UCI Machine Learning Repository website to carry out this research. The dataset contains 24 variables including a binary response variable, *default payment next month*, having a 0 for 'No Default' and a 1 for 'Yes Default'. Table 1 below has a summary of all the variables and their mathematical representation used for the analysis in Microsoft Excel and R software. A random sample of the 30,000 observations (half) is used for training the data and the remaining for testing it to detect the misclassification (test error) rates.

Table 1. Description of Variables used in Data Set

VARIABLE	DESCRIPTION
Default payment next month (Y)	0 for "No", 1 for "Yes"
Limit_bal (X1)	Credit Limit accessible – Amount of the given credit
Gender (X2)	1 for male, 2 for female
Education (X3)	Education level (1=graduate school, 2=bachelors, 3=high school, 4=others)
Marital status (X4)	1 for married, 2 for single, 3 for others
Age (X5)	Ranges from 21 years to 79 years
Pay_0 to Pay_6 (X6, X7, …, X11)	History of past monthly payment from September 2005 (X6) down to April 2005 (X11) where the measurement scale for the repayment status is: -1=pay duly; 1=payment delay for one month,…, 9=payment delay for 9 months and above
Bill_amt1 to Bill_amt6 (X12, X13, …, X17)	Amount of bill statement from September 2005 (X12) down to April 2005 (X17)
Pay_amt1 to pay_amt6 (X18, X19, …, X23)	Amount paid in September 2005 (X18) down to April 2005 (X23)
Input variables are denoted by X and Y for the response	

Note. *For variables X6 to X11 -1 denotes that payment was made duly prior to the month of interest, 0 denotes a payment in that same month bill was due, 1 denotes payment was made one month after it was due, up to 9 which denotes 9 months payment delay and above. Default of not results in the data were recorded for the month of October 2005; where a 0 implies client paid and a 1 implies client defaulted in payment for the amount due in October.*

Classification Performance Measures

The two main measures of classification performance commonly noted in data mining literature are considered as performance measures in this research: the misclassification rate and the area under the receiver operating characteristic curve (AUC). The misclassification (error) rate is the primary classification performance measure. Using a format known as the "confusion matrix" in machine learning, Figure 1 summarizes the four possible outcomes for the two-class classification problem in this study. Positive connotes a Default because the defaulting class is the object of interest in this research. Classifiers that correctly predict the actual Defaults and No Defaults are labeled true positives (TP) and true negatives (TN), respectively; those that incorrectly predict the actual Defaults and No Defaults are denoted false negatives (FN) and false positives (FP), respectively. The sensitivity is also known as the true positive rate (TPR) and the specificity is equally true negative rate (1 –FPR) (Lucas et al, 2013). A cut-off threshold value of probability above 50% is used to denote 'Default'. Accuracy, using the confusion matrix, is calculated as [(TP+TN) / (TP+FN+FP+TN)] and from that we get the misclassification rate by subtracting calculated Accuracy from one. Overall accuracy (minimum error rate) is not a sufficient performance indicator where there is significant class imbalance in the data since a default prediction of all cases into the majority class will show a high performance value (Bhattacharyya, Jha, Tharakunnel and Westland, 2011). To supplement this, the area under the receiver operating characteristic (ROC) curve (AUC) is also used as classification performance metric. Ling, Huang and Zhang (2003) argue that the AUC is a better measure than accuracy in comparing learning algorithms.

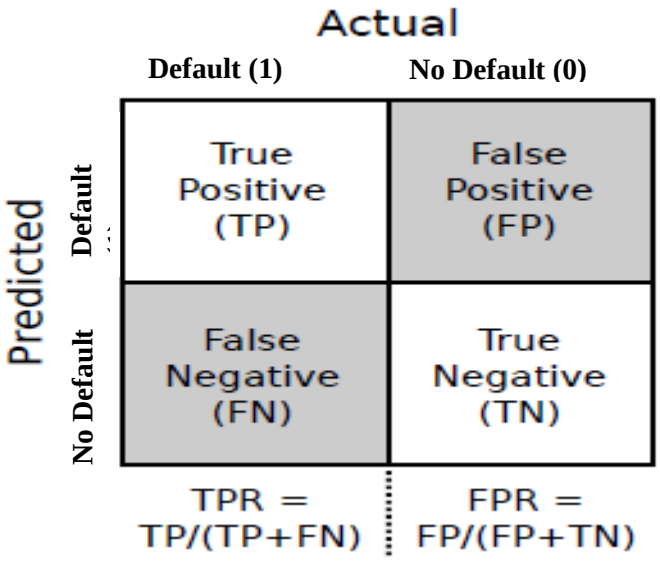


Figure. 1. The confusion matrix showing the four possible outcomes for a two-class classification problem

4. Results and Discussion

Of the 30,000 client observations, 6,636 missed the payment due on their credit card the current month under consideration. This represents a high default rate of 22.12%, as can be seen in Table 2. A card issuing organization may want to find means to efficiently minimize the losses from issuing to less creditworthy customers and the techniques discussed in this section may be a starting point for detecting the models to use and most important factors to consider in their decision making. A preliminary analysis of the whole data set begins the discussion.

Table 2. Distribution of Data

	Percent of Training	**Percent of Testing**	**Percent of Total**
Defaulters (1)	3,343 (22.29%)	3,293 (21.95%)	6,636 (**22.12%)**
Non-defaulters (0)	11,657 (77.71%)	11,707 (78.05%)	23,364 (**77.88%)**
TOTAL	**15,000** (50.00%)	**15,000** (50.00%)	**30,000 (100.00%)**

Regressing the response variable on all the predictors, gave a significant overall model based on the p-value of approximately zero but with low a residual standard error of 0.3886 and a low R-squared of 12.4% (mainly because of the large observation size). Since the data has a categorical response the R-squared value cannot be used as a measure of goodness of fit. From the full model in Figure 2 only 10 (excluding the intercept) of the coefficients are significant at even significantly low alpha levels. The p-values for these selected variables are approximately zero showing they are significant at all levels of alpha. This is supported by the reduced model regression containing only the ten selected predictors. Deleting the non-significant variables, the R-squared and residual standard error become 12.32% and 0.3887 respectively, almost the same as that of the full model. The adjusted R-squared adjusts for the number of variables used in a model, unlike the usual R-squared which increases as the number of variables increases. The model with the largest adjusted R-squared is preferred. Removing the redundant variables and rerunning the code gave an adjusted R-squared of 12.29% for the reduced model comparable to 12.33% suggested by the full model with all 23 variables. This means that taking out the non-significant variables does not have much impact on explaining the variation in the response, even though in the case of categorical variables the R-squared is not a useful determinant of goodness of fit. The Analysis of Variance table (Figure 2) suggests we fail to reject the reduced model at significant levels below the p-value of 10.29%. The hypothesis for the reduced model selection is as below:

Null: $H_0: \beta_1 = \beta_2 = \cdots = \beta_{p-1} = 0$

Alternate: $H_1: \beta_i \neq 0$, for all values of *i* =1, 2… p – 1, where p = 23 predictors + intercept

Further tests are conducted to determine if truly we can predict the accuracy of the data mining techniques discussed based on only the ten important variables and get equally good results.

Figure 2. Preliminary Analysis for Full Model and Corresponding Reduced Model

FULL MODEL

```
lm(formula = Y ~ ., data = credit)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.142e-01  1.791e-02  17.541  < 2e-16 ***
X1          -9.053e-08  2.159e-08  -4.193 2.76e-05 ***
X2          -1.453e-02  4.642e-03  -3.130  0.00175 **
X3          -1.513e-02  3.012e-03  -5.022 5.15e-07 ***
X4          -2.382e-02  4.768e-03  -4.996 5.88e-07 ***
X5           1.409e-03  2.749e-04   5.128 2.95e-07 ***
X6           9.571e-02  2.766e-03  34.596  < 2e-16 ***
X7           1.946e-02  3.339e-03   5.828 5.68e-09 ***
X8           1.167e-02  3.585e-03   3.256  0.00113 **
X9           3.362e-03  3.974e-03   0.846  0.39755
X10          5.699e-03  4.304e-03   1.324  0.18545
X11          7.920e-04  3.521e-03   0.225  0.82201
X12         -6.225e-07  1.141e-07  -5.453 4.98e-08 ***
X13          1.587e-07  1.603e-07   0.990  0.32225
X14          3.005e-08  1.510e-07   0.199  0.84222
X15         -6.793e-08  1.573e-07  -0.432  0.66587
X16         -2.049e-08  1.845e-07  -0.111  0.91159
X17          1.153e-07  1.460e-07   0.789  0.42998
X18         -7.437e-07  1.770e-07  -4.201 2.67e-05 ***
X19         -2.092e-07  1.457e-07  -1.436  0.15095
X20         -2.874e-08  1.689e-07  -0.170  0.86492
X21         -2.521e-07  1.839e-07  -1.371  0.17047
X22         -3.410e-07  1.908e-07  -1.787  0.07393 .
X23         -9.770e-08  1.365e-07  -0.716  0.47422
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3886 on 29976 degrees of freedom
Multiple R-squared:  0.124,     Adjusted R-squared:  0.1233
F-statistic: 184.5 on 23 and 29976 DF,  p-value: < 2.2e-16
```

REDUCED MODEL

```
Call:
lm(formula = Y ~ . - X9 - X10 - X11 - X13 - X14 - X15 - X16 -
    X17 - X19 - X20 - X21 - X22 - X23, data = credit)

Residuals:
     Min       1Q   Median       3Q      Max
-1.30307 -0.23949 -0.16138  0.03274  1.26179

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.133e-01  1.791e-02  17.495  < 2e-16 ***
X1          -1.105e-07  2.082e-08  -5.309 1.11e-07 ***
X2          -1.429e-02  4.640e-03  -3.080  0.00207 **
X3          -1.528e-02  3.011e-03  -5.075 3.91e-07 ***
X4          -2.403e-02  4.767e-03  -5.041 4.67e-07 ***
X5           1.412e-03  2.749e-04   5.138 2.79e-07 ***
X6           9.737e-02  2.740e-03  35.537  < 2e-16 ***
X7           2.007e-02  3.306e-03   6.071 1.28e-09 ***
X8           1.785e-02  2.983e-03   5.983 2.22e-09 ***
X12         -4.587e-07  3.445e-08 -13.315  < 2e-16 ***
X18         -7.858e-07  1.399e-07  -5.616 1.97e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3887 on 29989 degrees of freedom
Multiple R-squared:  0.1232,    Adjusted R-squared:  0.1229
F-statistic: 421.4 on 10 and 29989 DF,  p-value: < 2.2e-16

************************************************************
                        ANOVA TABLE

1. Null Hypothesis:        Reduced Model
2. Alternative Hypothesis: Full Model

  Res.Df   RSS     Df  Sum of Sq     F      Pr(>F)
1  29989   4531.4
2  29976   4527.2   13   4.1695     2.1237  0.01029 *

#Fail to reject the null at sig. level below 10.29%
Conclusion: Reduced Model is significant
```

Note. **** denotes significance at all levels of alpha. ** denotes significance at 0.001 level of alpha. Variables with three stars (***) are important at all levels of alpha. From the full model figure, these are X1, X2, X3, X4, X5, X6, X7, X12, X18, and the intercept. Variable X8 is significant at alpha levels above 0.001.*

4.1 Measuring the accuracy of the Full Model

The Logistic model produced a prediction accuracy of 81.16% corresponding to a misclassification rate of 18.84%. It has a sensitivity of 71.88% and a specificity of 81.87%. This means it can correctly classify the default class 71.88% of the time and the probability of correctly classifying the negative (no default) class is 81.87%. Given the fact that only a small percentage of clients default in payment compared to the non-default class, the specificity values will almost always be higher than the sensitivity values. The LDA technique gave an

accuracy of 81.3%, a misclassification rate of 18.7%, and sensitivity and specificity values of 70.89% and 82.18% respectively. QDA results have a discouragingly high error rate of 54.9%. Based on this data set this method can correctly classify clients with a low accuracy below probability of chance, 50%. Even with this bad performance it produces a better specificity rate of 89.03% as compared to LDA, and not surprisingly, 26.58% rate for specificity worse than LDA. These values can be inferred from Table 3 below.

K=100 nearest neighbors are suggested from cross validation. With this parameter, KNN gives 78.13% predictive accuracy, 21.83% misclassification rate, a low sensitivity of 53.63% and the least specificity of 78.58%. Since the data set contains just 22.12% positive class (default) values, sampling 100 nearest neighbors will yield a higher probability of getting non-default values than default values in a sample, thereby giving a low sensitivity as compared to specificity for the KNN algorithm. As discussed in literature review under section 2, tuning the parameter required for computing the Ridge regression yields an error-minimizing log λ value of –4.204615 implying $\lambda \cong 0.01493$ for Ridge. This value is not big enough for the penalty term to have a significant impact of shrinking some of the coefficients towards zero. Using this best lambda value for the analysis gives an accuracy of 79.83%, a misclassification rate of 20.17%, sensitivity 72.19% and specificity 80.15%. Similarly, the Lasso also produced an error-minimizing log λ of –6.739785 implying $\lambda \cong 0.00118$, also too small for lasso to have an impactful penalty. The Lasso regression, with this λ value shrinks variables X13, X13, X14, X15, X16, and X17 to zero and produces a prediction accuracy of 79.93%, a lower misclassification rate of 20.07% compared to that for ridge, with a correct positive (default) prediction of 72.6% and correct negative (no default) prediction of 80.24%.

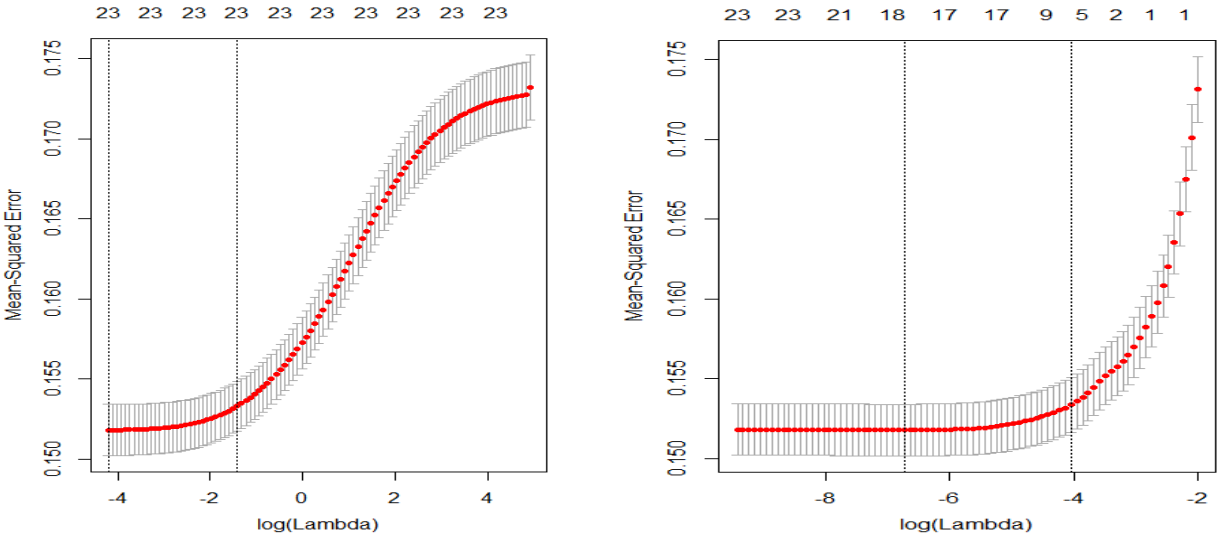
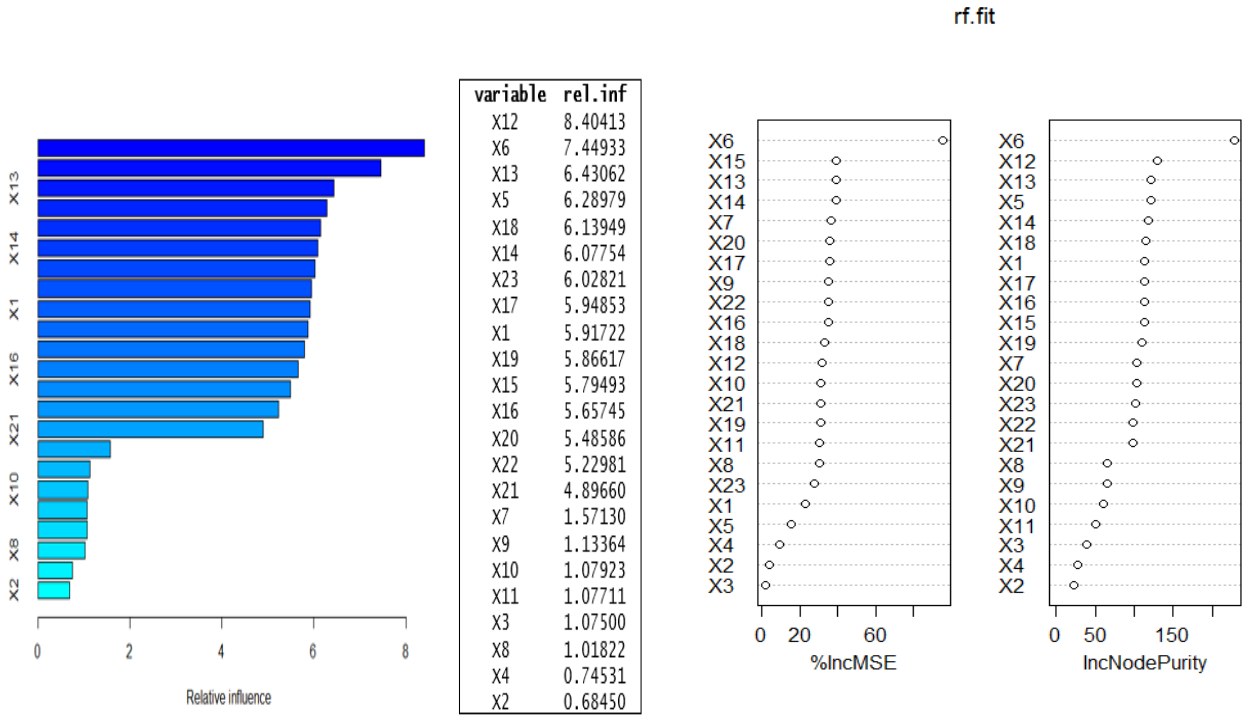


Figure 3. The plots for optimal log lambda for both Ridge and Lasso Regression

1. *Ridge Regression* 2. *Lasso Regression*

Note. *The best log(Lambda) value on the x value can be found from the left of the two vertically-dotted lines in each plot. These values are chosen automatically based on the least mean squared error criteria in R software*

A cross-validation of the parameters for random forest yields T=7000 trees for use in training the model and 3 best variables randomly sampled as candidates at each split instead of approximately 5 ($= \sqrt{23}$) variables used as a standard for classification problem. The 7000 trees and 3 each-node-split variables gave better overall accuracy for the random forest algorithm of 81.85%, an error rate of 18.15% with respective sensitivity and specificity of 66.05% and 83.98%. Node purity of the variable importance plot from random forest (Figure 4) suggests *X6 as the most important variable*, followed by X12 with X2 and X4 being the least important variables. Using the Adaptive Boosting (Adaboost) algorithm with fine-tuned parameters of 7500 trees and interaction depth of 3 from cross validation, we get an accuracy of 79.47% implying a misclassification rate of 20.53%. The sensitivity for Adaboost is also not encouraging (54.82%) but with a better specificity value of 83.75%. The relative influence plot, also in Figure 4, shows that variable X12 and X6 are the most important variables the boosting algorithm recognizes with X2 and X4 being the least important just like the random forest predicted. Unsurprisingly, from the variable description one can find that X6 (the most recent amount paid from the immediate past month under consideration) and X12 (bill statement for the immediate past month) are great influences, in real life, of what we would expect to pay in the current month and whether or not we would make that payment. As to whether X2 (one's gender) and X4 (marital status) do not have any influence on predicting default is another topic for discussion.



variable	rel.inf
X12	8.40413
X6	7.44933
X13	6.43062
X5	6.28979
X18	6.13949
X14	6.07754
X23	6.02821
X17	5.94853
X1	5.91722
X19	5.86617
X15	5.79493
X16	5.65745
X20	5.48586
X22	5.22981
X21	4.89660
X7	1.57130
X9	1.13364
X10	1.07923
X11	1.07711
X3	1.07500
X8	1.01822
X4	0.74531
X2	0.68450

Figure 4. Relative Influence Plot from Adaboost and Variable Importance Plot of Random Forest

Note. The plot on the left is the relative influence plot for boosting and that on the right is the VarImp plot of RF

For Support Vector Machines (SVMs), a general-purpose kernel with good performance results, Gaussian radial basis function, is used. After experimenting with different combinations of the cost parameter C and the gamma, these were set to be C=1, gamma=0.0435. The results from the SVM technique showed that the algorithm used a maximum of 6675 support vectors to produce an accuracy of 81.99%, a misclassification rate of 18.01%, with 70.22% sensitivity and 83.27% specificity. Based on the commonly used hidden layer of 2 in data mining literature, the artificial neural network produced low accuracy and sensitivity of 68.43% and 30.73% respectively with a high specificity of 80.96%. The neural interconnectedness of the artificial neural network is shown in Figure 7 found in the Appendix.

Table 3. Summary of the Performance Accuracy of the Full model

	FULL MODEL		
TECHNIQUE	MISCLASSIFICATION RATE	SENSITIVTY	SPECIFICITY
Logistic	18.84%	71.88%	81.87%
LDA	18.70%	70.89%	82.18%
QDA	54.90%	26.58%	89.03%
Ridge	20.17%	72.19%	80.15%
Lasso	20.07%	72.60%	80.24%
KNN	21.83%	53.63%	78.58%
Random Forest	18.15%	66.05%	83.98%
Boosting	20.53%	54.82%	83.75%
SVM	18.01%	70.22%	83.27%
Neural network	31.57%	30.73%	80.96%

From Table 3 and the discussion so far, *SVM had the least misclassification rate* of 18.01% (a predictive accuracy of 81.99%), followed closely by Random Forest, LDA, and Logistic with error rates of 18.15%, 18.70%, and 18.84% respectively. The error rate of 31.57% for neural network is one of the worst but QDA had an unrealistically high misclassification rate which may possibly suggest the predictors in this data set have nothing to do with the "quadratic" nature of a QDA. Even though it had the highest specificity of 89.03%, it also had the worst sensitivity of 26.58% meaning it can only predict with about 27% accuracy the positive class (default). Random Forest, Boosting and SVM have the highest specificities after QDA; these three techniques can correctly classify the negative class (non-default) with at least 83% certainty. For correctly classifying the default class, SVM, LDA and Logistic have comparable results and follow Ridge and Lasso regression values of 72.19% and 72.60% respectively. Lasso and Ridge have approximately same values for misclassification rates, sensitivities and specificities because the Lasso, by definition, is a recent alternative to Ridge and the only

difference between these two techniques is seen most when the lambda is high enough for the penalty term to have a considerable impact on the coefficients: in this analysis, it has already been established that this is not so because of very small values of lambda. The Lasso has slightly better figures because the algorithm is able to use the variables that it needs for prediction. Next, let us consider the subset of variables that may help give similar classification performance.

4.2 Subset selection algorithm

Based on the preliminary analysis, we notice that certain variables are not needed in order to make decisions regarding predictive accuracy. This was confirmed by the analysis of variance table in Figure 2. Lasso regression shrunk variables X13 to X17 to zero, random forest and boosting suggest X6 and X12 as the two most important variables, and accordingly X4 and X2 are the worst. This section employs three variable reduction techniques to detect the subset of variables that best describe the data and to confirm these conclusions reached so far.

Forward stepwise selection is a subset selection method that starts with a model containing only the intercept (and no predictors), and then adds predictors to the model, one-at-a-time, until all the predictors are in the model. At each step the variable that gives the greatest additional improvement to the fit is added to the model. The iteration stops when it has sufficient predictors that give the lowest AIC, BIC or RSS, or the highest adjusted R-squared. The backward stepwise selection algorithm begins with the full least squares model containing all p=23 predictors, and then iteratively removes the least useful predictor one at a time until the optimal AIC is reached for the sufficient variables. Finally, under best subset selection algorithm, we fit a separate least squares regression for each possible combination of 23 predictors and select the best model from among all the 2^p combinations, in our case $2^{23} = 8388608$ possibilities.

Conclusively, all three methods chose the 15 predictors as the number of variable maximum with the least AIC. Interestingly, for the first best 10 performing variables, all three algorithms selected the same predictors as the reduced model: X1 to X8, X12 and X18. X6 was always chosen to enter the model first by all three selection methods, followed by X12, implying those two are the most important variables for this credit default data set. In order not to overfit the model any further and for the sake of parsimony, we test the ten data mining techniques on the reduced model with the ten most important variables (circled in Figure 5). The plots for subset selection based on adjusted R-squared for all the three methods discussed here is located in the Appendix under Figure 8.

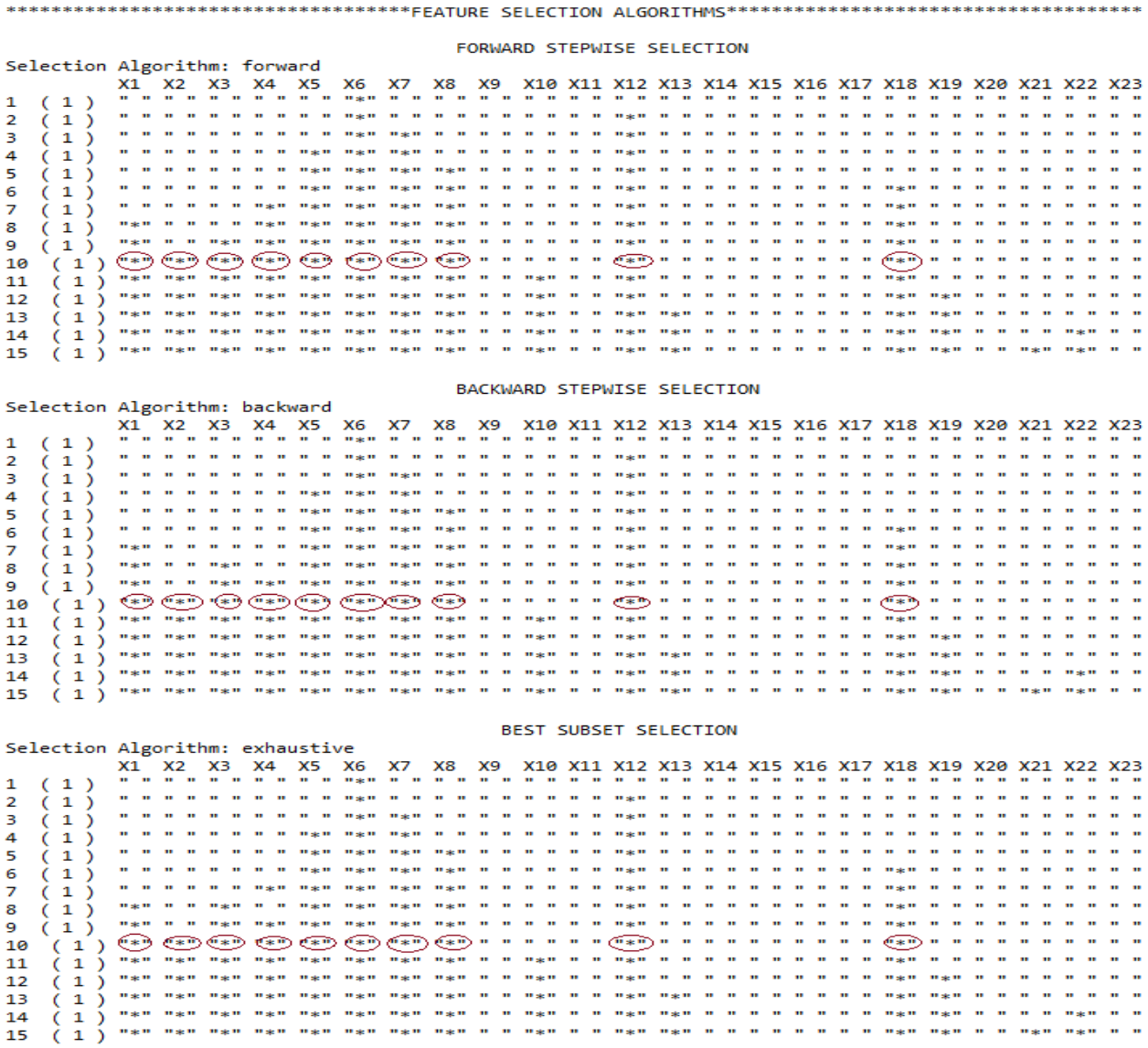


Figure 5. Three Subset Selection Algorithms

***Note**. For all three procedures, the best 1 variable model is the model that contains only X6 and the intercept. The best two variable technique contains, with the intercept, variables X6 and X12. The best three variable model adds X7 to the previous two. The best ten variable model is the same as the reduced model in Figure 1. The subset selections chose the variables to enter the model based on the least RSS, AIC, or BIC, or the Mallow Cp and the highest Adjusted R-squared. Figure 8 in the Appendix displays the adjusted R-squared plots associated with this design for selecting variables to enter the model.*

4.3 Measuring the accuracy of the Reduced Model

Table 4 provides a summary of the classification performance for the reduced model with the ten important predictors. *SVM had the least misclassification rate* of 17.88% (a predictive accuracy of 82.12%), followed closely by LDA, and Logistic with error rates of 18.75%, and 18.84% respectively. The error rates for all the classifiers this time around were well below 30% with random forest having the highest misclassification rate

of 27.24%. Neural network and QDA performed far better with the omission of redundant variables. Even though it still had the highest specificity of 87.36%, QDA also produced a not-so-good sensitivity of 48.35% which was a big improvement in the previous 26.58%. Boosting and SVM have the highest specificities after QDA; these three techniques can correctly classify the negative class (non-default) with at least 83% certainty.

Comparing Tables 3 and 4, *our best performing technique for both full and reduced, SVM showed an improvement in misclassification rate as it reduced from 18.01% to 17.88%.* Its values for sensitivity and specificity also increased after variable reduction. SVM was the only technique to have seen an increase in specificity after the change; but for logistic regression which showed no change, the rest decreased in specificity. Logistic regression values for all three performance measures in the table remain unchanged suggesting the easiest and most widely known and used technique for categorical variable is indifferent to the independent variables that do not aid prediction. The biggest improvements in misclassification rate can be seen with QDA error rate declining from 54.9% to 22.81% and artificial neural network error rate also decreasing from 31.57% to 22.27%. Their sensitivities also show a similar improvement pattern.

Table 4. Summary of the performance accuracy of the reduced model

	REDUCED MODEL		
TECHNIQUE	MISCLASSIFICATION RATE	SENSITIVTY	SPECIFICITY
Logistic	18.84%	71.88%	81.87%
LDA	18.75%	70.80%	82.12%
QDA	22.81%	48.35%	87.36%
Ridge	20.28%	71.45%	80.06%
Lasso	20.13%	71.70%	80.22%
KNN	22.01%	27.78%	78.05%
Random Forest	27.24%	22.94%	78.15%
Boosting	20.85%	53.79%	83.48%
SVM	17.88%	71.08%	83.30%
Neural network	22.27%	44.81%	78.77%

4.4 The Area under the ROC Curve (AUC)

The Receiver Operating Characteristic (ROC) curve is a popular graphic for simultaneously displaying the two types of errors (FPR – False Positive rate and FNR – False Negative Rate) for all possible thresholds. The overall performance of a classifier summarized over all possible thresholds, is given by the area under the (ROC) curve

(AUC). An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier. An AUC value of 1 represents a perfect classifier and we expect a classifier that performs no better than chance to have an AUC of 0.5 (when evaluated on an independent test set not used in model training). ROC curves are useful for comparing different classifiers, since they take into account all possible thresholds. According to Ling, Huang and Zhang (2003) the AUC is a better measure than accuracy in comparing learning algorithms based on formal definitions of discriminancy and consistency.

Table 5. Comparing the AUC for Full and Reduced Models

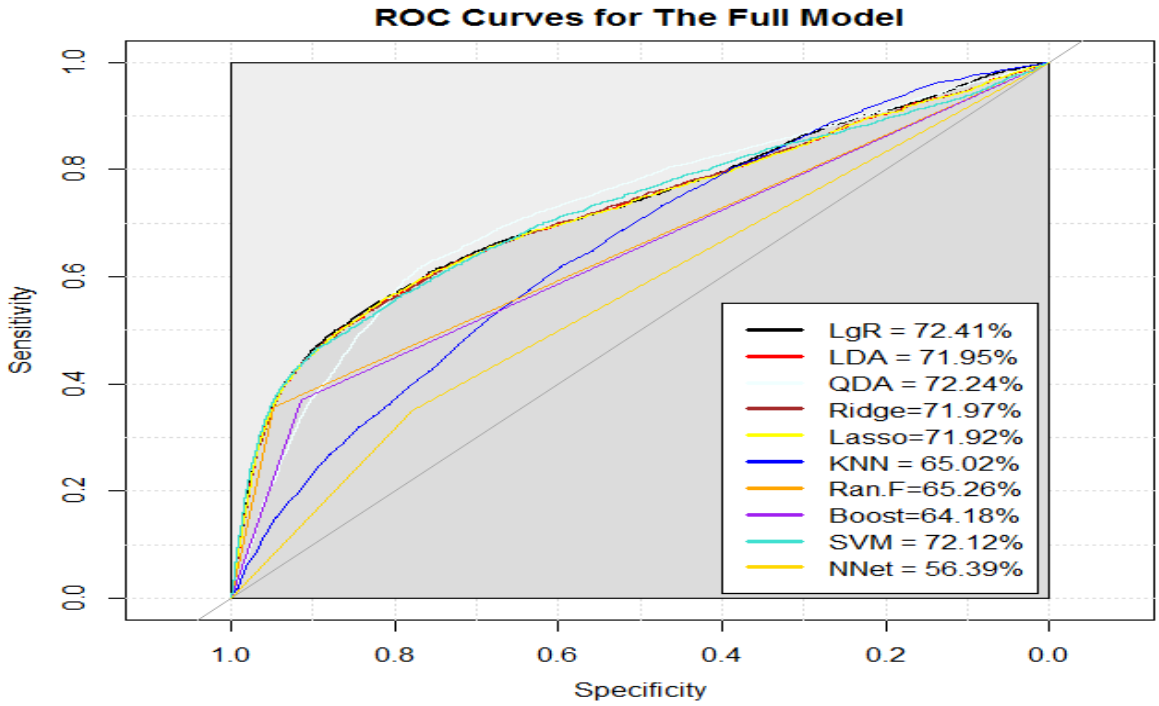
Area Under the ROC Curve for Both Models										
MODEL	Lg.R	LDA	QDA	Ridge	Lasso	KNN	Ran.F	Boost	SVM	NNet
Full Model (%)	72.41	71.95	72.24	71.97	71.92	65.02	65.26	64.18	72.12	56.39
Reduced Model (%)	72.41	72.11	73.23	72.19	72.11	62.99	50.28	63.55	71.78	52.05

Note. *Lg.R = Logistic, Ran.F = Random Forest, NNet = Artificial Neural Network, the others have previously been defined. The values displayed in the table are all in percentages.*

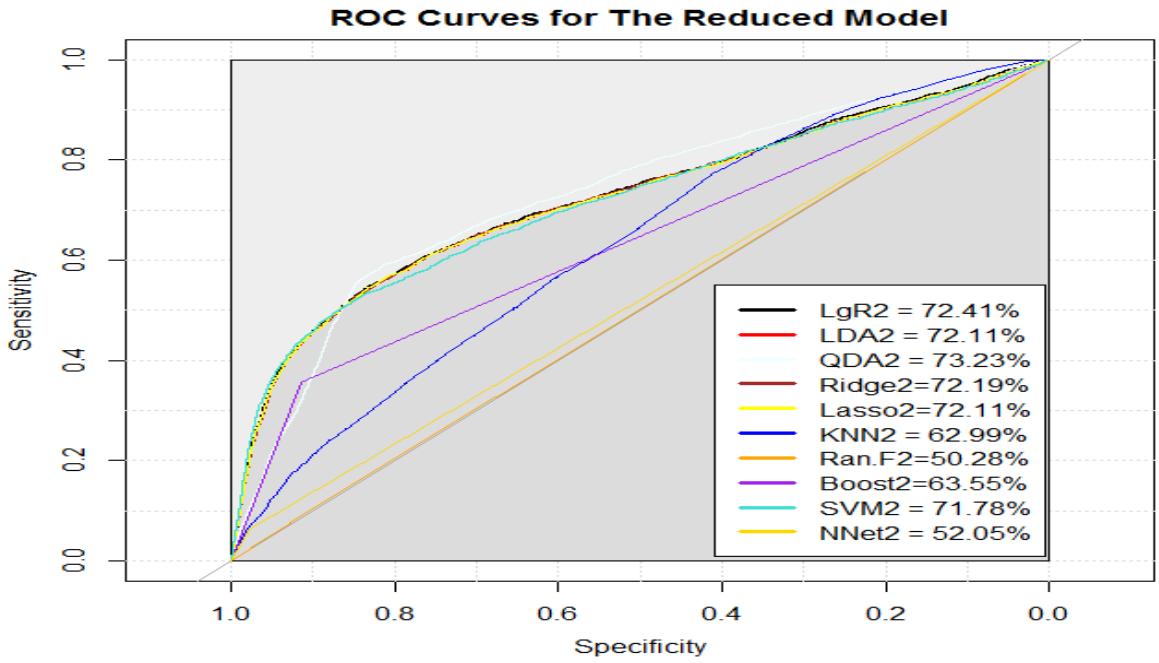
For the full model, Logistic regression has the highest AUC value of 72.41%, followed by QDA and SVM with 72.24% and 72.12% respectively; not much difference between these three. Ridge and Lasso regressions have comparable areas of around 72% for the full model as can be seen in Table56. Neural network had the least area: its AUC value of 56.39% is almost a "classification by chance" – it is indifferent between true positives and true negatives. Referring to Figure 6, the ROC curve for neural network for the full model (marked with Gold) is closer to the 45-degree line which represents the "no information" classifier; this is what we would expect if predictors and default status are not associated with probability of default. Since most of the classifiers have AUC values around 72%, this makes it difficult to see the beauty of the ROC curve but in generally a preferred ROC curve will hug the top left of the curve and have the highest AUC.

The Reduced model, produced similar results. The Logistic AUC did not change, just like the values for its performance measures discussed previously. The AUC for SVM decreased slightly to 71.78%, but the AUCs for QDA, Ridge and Lasso improved. Neural network was still not good.

Figure 6. Receiver Operating Characteristic Curves



Note. *The true positive rate is the sensitivity: the fraction of defaulters that are correctly identified, using a given threshold value. The false positive rate is 1-specificity: the fraction of non-defaulters that we classify incorrectly as defaulters, using that same threshold value. The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The* 45^o *line represents the "no information" classifier; this is what we would expect if predictors and default status are not associated with probability of default. The legend displays the AUCs for the various data mining techniques considered.*



5. Conclusion

This paper used a real-life data of credit card clients to investigate the predictive accuracy of some data mining techniques on the probability of default, a two-class classification problem. The data contains some basic underwriting information required for credit card approval to a new applicant aside some payment history for monthly balances due. The misclassification rate and the area under the curve were the two main performance metrics used to measure the precision of ten data mining techniques to predicting default. These techniques include: Logistic regression, Ridge and Lasso regression, K-nearest neighbors, Linear and Quadratic discriminant analysis, and these four advanced predictive modeling approaches – support vector machines, random forest, generalized boosted models and artificial neural networks.

Preliminary analysis supported by subset selection methods reveal ten important variables may give the same information and comparably competitive accuracy to the twenty-three predictors. Random forest and boosting algorithms reveal variable X6, the immediate past monthly payment, and variable X12, the immediate past bill amount, are the two most important predictors. This conclusion is further supported by subset selection and reality as what amount we pay this month depends more or less on these two recent amounts. Random forest and boosting also suggested the least important variable was X2, the gender of a client. Support vector machines had the least error rate for both the full model with twenty-three predictors and the reduced model with ten predictors. Linear discriminant analysis and logistic regression follow this value closely for both full and reduced models. Dealing with the ROC curves, the area for logistic was the highest followed by support vector machines and quadratic discriminant analysis which had the worst misclassification rate for the full model. It was found that the performance measures for logistic does not change with the reduction in variables, making logistic a very good robust model for default prediction. To conclude, both support vector machines, an advanced predictive model, and logistic regression, the easiest and most used model for classification, should be good predictive models for optimal prediction of the probability of credit card default.

The credit market has been saddled with increasing levels of credit default in spite of the strength of the U.S. economy. Clearly, credit card default is a complex phenomenon involving many factors beyond the scope of the present research. The variables together with techniques and performance measures which have been examined here capture some key behaviors which have not been studied previously and hopefully shed new light on this default problem. Such new conclusions will be very useful for machine learning and its applications. Yeh and Lien (2009) employ six data mining techniques to investigate credit card default prediction using the novel Sorting Smoothing technique. Further research into other data mining techniques not considered in this paper can be done on the performance measures to improve prediction. As discussed, artificial neural networks have been used widely in the health industry but may have spilling applications for

credit markets in the finance world. A critical look at the reason for the underperformance of the neural network in predicting probability of credit default is an interesting topic for further study. Likewise, using very advanced statistical knowledge in machine learning like the popularly known artificial intelligence (AI) available only on commercial basis to large organizations can be utilized to improve accuracy considerably. AI, based on heuristics as opposed to statistics, is used to apply human-thought like processing to statistical problems like the one encountered in this paper.

References

[1] A WalletHub 2017 study. https://wallethub.com/edu/credit-card-debt-study/24400/

Albashrawi, M. (2016). Detecting Financial Fraud Using Data Mining Techniques: A Decade Review from 2004 to 2015. *Journal of Data Science*, *14*(3), 553-569.

Aleskerov, E., Freisleben, B., & Rao, B. (1997, March). Cardwatch: A neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997* (pp. 220-226). IEEE.

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. *Decision Support Systems*, *50*(3), 602-613.

Brause, R., Langsdorf, T., & Hepp, M. (1999). Neural data mining for credit card fraud detection. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on* (pp. 103-106). IEEE.

Breiman, L. (1996). Bias, variance, and arcing classifiers.

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

Chen, R. C., Chen, T. S., & Lin, C. C. (2006). A new binary support vector system for increasing detection rate of credit card fraud. *International Journal of Pattern Recognition and Artificial Intelligence*, *20*(02), 227-239.

Data. https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients . From the UCI Machine Learning Repository website.

Domowitz, Ian, and Thomas L. Eovaldi, "The Impact of the Bankruptcy Reform Act of 1978 on Consumer Bankruptcy," *Journal of Law and Economics*, 36 (2), October 1993, 803-835.

Dunn, L. F., & Kim, T. (1999). An empirical investigation of credit card default. *Ohio State University, Department of Economics Working Papers*, (99-13).

Flood, I., & Kartam, N. (1994). Neural networks in civil engineering. II: Systems and applications. *Journal of Computing in Civil Engineering*, *8*(2), 149-162.

Gadi, M. F. A., Wang, X., & do Lago, A. P. (2008, August). Credit card fraud detection with artificial immune system. In *International Conference on Artificial Immune Systems* (pp. 119-131). Springer Berlin Heidelberg.

Gross, D. B., & Souleles, N. S. (2002). An empirical analysis of personal bankruptcy and delinquency. *Review of financial studies*, *15*(1), 319-347.

Hornik, K. (1993). Some new results on neural network approximation. *Neural networks*, *6*(8), 1069-1072.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 6). New York: springer.

Koh, H. C., & Tan, G. (2011). Data mining applications in healthcare. *Journal of healthcare information management*, *19*(2), 65.

Kou, Y., Lu, C. T., Sirwongwattana, S., & Huang, Y. P. (2004). Survey of fraud detection techniques. In *Networking, sensing and control, 2004 IEEE international conference on* (Vol. 2, pp. 749-754). IEEE.

Ling, C. X., Huang, J., & Zhang, H. (2003, June). AUC: a better measure than accuracy in comparing learning algorithms. In *Conference of the Canadian Society for Computational Studies of Intelligence* (pp. 329-341). Springer Berlin Heidelberg.

Lopes, P. (2008). Credit card debt and default over the life cycle. *Journal of Money, Credit and Banking*, *40*(4), 769-790.

Lucas, D. D., Klein, R., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D., & Zhang, Y. (2013). Failure analysis of parameter-induced simulation crashes in climate models. *Geoscientific Model Development*, *6*(4), 1157-1171.

Panigrahi, S., Kundu, A., Sural, S., & Majumdar, A. K. (2009). Credit card fraud detection: A fusion approach using Dempster–Shafer theory and Bayesian learning. *Information Fusion*, *10*(4), 354-363.

Ratcliffe, C., McKernan, S. M., Theodos, B., Kalish, E., Chalekian, J., Guo, P., ... & BRIEF, O. I. (2014). Delinquent debt in America. *Urban Institute*, *29*.

Statnikov, A., Wang, L., & Aliferis, C. F. (2008). A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. *BMC bioinformatics*, *9*(1), 319.

Titan, E., & Tudor, A. I. (2011). Conceptual and Statistical Issues Regarding the Probability of Default and Modeling Default Risk. *Database Systems Journal*, *2*(1), 13-22.

Turban, E., Aronson, J. E., Liang, T. P., & Sharda, R. (2007). Decision support and business intelligence systems (Eighth ed.). Pearson Education.

Whitrow, C., Hand, D. J., Juszczak, P., Weston, D., & Adams, N. M. (2009). Transaction aggregation as a strategy for credit card fraud detection. *Data Mining and Knowledge Discovery*, *18*(1), 30-55.

Wikipedia (2017). https://en.wikipedia.org/wiki/Random_forest.

Wikipedia (2017). https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

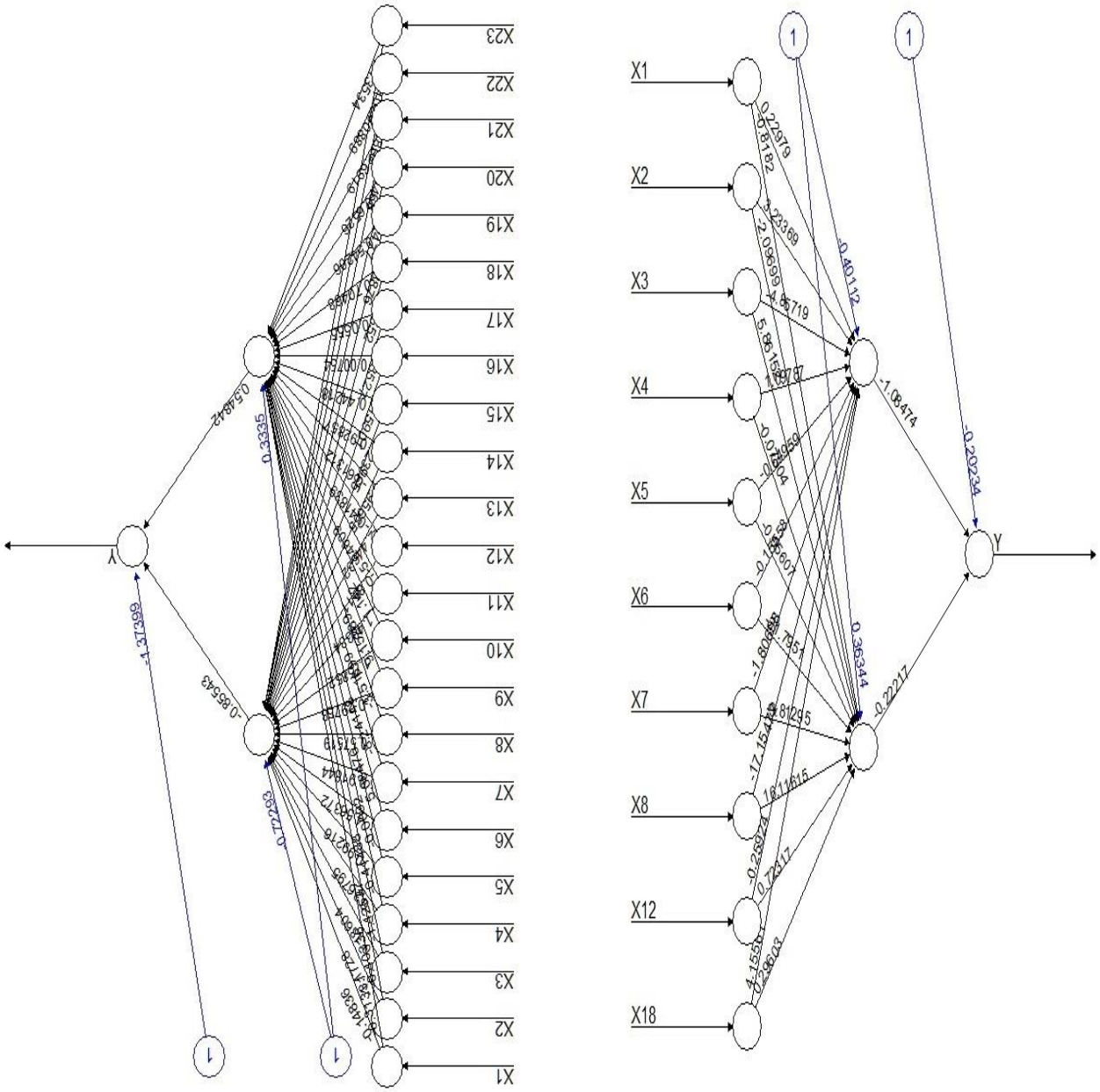
Wu, J., Xiong, H., & Chen, J. (2010). COG: local decomposition for rare class analysis. *Data Mining and Knowledge Discovery*, *20*(2), 191-220.

Yang, X., & Zheng, J. (2009). Artificial neural networks. *Handbook of Research on Geoinformatics*, *122*.

Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, *36*(2), 2473-2480.

APPENDIX

Figure 7. Artificial neural network plot used in training the models



FULL MODEL REDUCED MODEL

Note. With two hidden layers, the neural interconnectedness of the full model and reduced model are displayed above. As discussed, artificial neural networks have been used widely in the health industry but may have spilling applications for credit markets in the finance world. The algorithm, based on this 'brain network' procedure produced competing results when applied to this data set. More research and better parameter tuning might yield better results.

Figure 8. Subset Selection plots based on the Adjusted R-squared