MACRO DESIGNING AND COMPARATIVE EVALUATION OF VARIOUS PREDICTIVE MODELING TECHNIQUES OF CREDIT CARD DATA

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Software Engineering

Name: Ravinder Singh

(Roll No. 800931016)

Under the supervision of:

Dr. Rinkle Rani

Assistant professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2011

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Macro Designing and Comparative Evaluation of various Predictive Modeling Techniques"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in ***Software Engineering*** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of ***Dr. Rinkle Rani*** and refers to other researcher's work which are listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Ravinder

Ravinder Singh

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Rinkle

Dr. Rinkle Rani

Assistant Professor, CSED

Thapar university, Patiala

(Dr. Maninder Singh)

Head,

Computer Science and Engineering Department,

Thapar University.

Countersigned by

(Dr. S. K. Mohapatra)

Dean (Academic Affairs),

Thapar University,

Patiala

Acknowdgement

I would like to express my gratitude towards my guide Dr. Rinkle Rani, Department of Computer Science and Engineering, Thapar University Patiala. She has been extremely supportive and cooperative during the entire one year of thesis work. She also motivated me to explore each topic in detail and be persistent in the research work.

I am also thankful to Dr. Maninder Singh, Head of Department, Computer Science and Engineering and Mr. Sumit Miglani, P.G coordinator as they provided encouragement and zeal for pursuing thesis work.

I would like express my gratefulness to the staff members who were always there to provide all help regarding lab facilities.

At the last, but not the least I would like to thank my parents and friends for always supporting me.

Ravinder

Ravinder Singh

(800931016)

Abstract

Credit Scoring studies are very important for any financial house. Both traditional statistical and modern data mining/machine learning tools have been evaluated in the credit scoring problem.

Predictive modeling defaulter risk is one of the important problems in credit risk management. There are quite a few aggregate models and data driven models available in literature

But very few of the studies facilitate the comparison of majority of the commonly employed tools in single comprehensive study. Additionally no study assesses the performance on more then two data sets and reports the results at the same time. So a macro or a simulator is designed which would work on multiple data sets and make the process of credit scoring transparent to the novice user. In initial stage, tools were compared using Dtreg predictive modeling software. Subsequently a SAS macro is developed to evaluate the effectiveness of tools available in SAS enterprise miner.

The results revealed that support vector machine and genetic programming are superior tools for the purpose of classifying the loan applicant as their misclassification rates were least as compared to others. Also cross validation is essential, though some of the tools may not support it directly.

Table of contents

List of Tables

List of Figures

Chapter 1 INTRODUCTION

Financial crimes are increasing at enormous rate every year and financial institutions must adopt to methods to safeguard their reputation and their customers. The use of statistical methods to address these problems faces many challenges [13]. Interestingly financial crimes are uncommon events that lead to extreme class imbalances. Criminals deliberately attempt to hide their usage patterns and quickly change their strategies over time, making the process of fraud detection sophisticated. Also sometimes legal constraints and investigations delays make it impossible to actually verify suspected crimes in a timely manner.

Credit scoring methods are statistical tools employed by various banks and other financial institutions, marketing and advertisement companies to estimate the probability whether the loan applicant could be categorized as potential defaulter or not [21]. Basically, credit scoring aims to classify the dependent variable with respect to the response variables. Banks collect the information about the applicant from various sources such as historical data, questionnaires and interviews. This aims at collecting all demographic details such as income, age, sex, type of loan, nationality, job, and income pertaining to the applicant.

The accurate prediction of consumer credit risk is indispensable for lending organizations. Credit scoring is a common technique that helps financial institutions to evaluate the likelihood for a credit applicant to be a potential defaulter on the financial obligation and decide whether to grant credit or not [13]. The precise judgment of the creditworthiness of applicants allows financial institutions to increase the volume of granted credit while minimizing possible losses. The increasing number of potential applicants has driven the need for the development of sophisticated fraud detection techniques that automate the credit approval procedure.

Earlier credit scoring was restricted to statistical techniques such as discrimination between several groups in a data sample. History reveals that credit scoring came into existence in early thirties of the nineteenth century when some financial institution decided to classify their applicants with respect to default status. In late 1950's the first

automated system was used to develop predictive model of the applicants based on their historical data.

Figure 1.1 depicts the basic credit card scoring framework. After the card holders default, several data collection procedures are undertaken. These procedures are expensive relative to the size of most loans. They are often useless when the card holder has gone bankrupt. Therefore, it is significant for the card issuers to identify card holder type at early stage in order to minimize lending to risky customers before the default occurs. It means that it becomes necessary to maximize the 'True positives' (TP). The true positive rate (TRP) is known as sensitivity and the true negative rate (TNR) is sometimes called specificity.

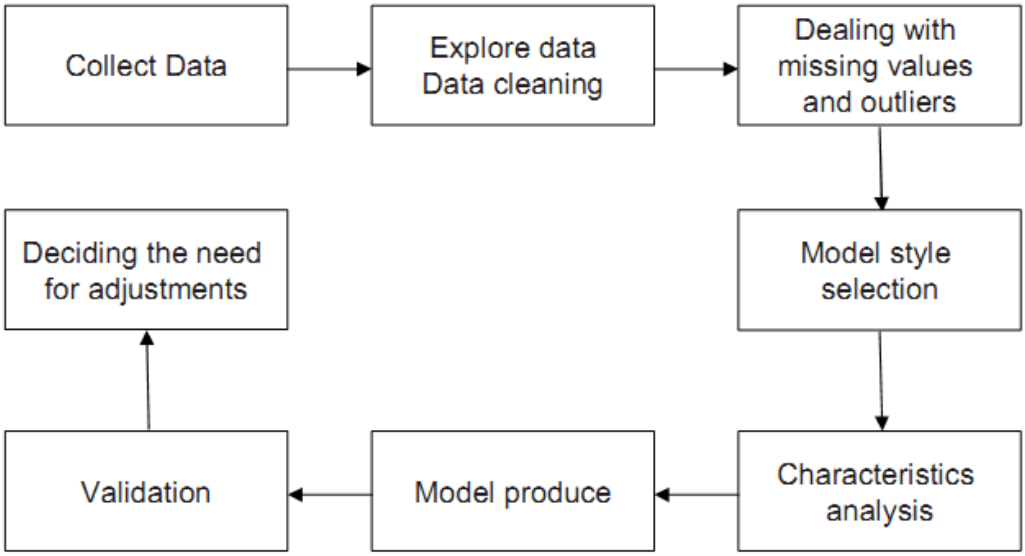

Figure 1.1: Credit model scoring framework

Predictive modeling defaulter risk is one of the important problems in credit risk management. There are quite a few aggregate models and data driven models available in literature. The main contribution of this study is empirical evaluation of the effectiveness of the various components of the predictive modeling process. In particular sampling selection and algorithm used for learning classification are evaluated on a real world dataset.

A credit scoring process is typically a classification problem where objective is to classify the loan applicant into predefined category [34]. The objective of credit scoring is to find a rule that separates goods and bads with the smallest percentage of misclassifications. Note that perfect classification is impossible due to several reasons. For instance, there could be errors in the sample data. Moreover, it is possible that some 'good' applications and 'bad' applications have the exactly the same information in all data fields.

Modeling is the process of creating a scoring rule from a set of examples. In order for modeling to be effective, it has to be integrated into a larger process. Let's look at application scoring. On the input side, before the modeling step, the set of example applications must be prepared. On the output side, after the modeling, the scoring rule has to be executed on a set of new applications so that credit-granting decisions can be made.

Collecting performance data occurs at the beginning and end of the credit-scoring process. Before a set of example applications can be prepared, performance data has to be collected so that applications can be tagged as good or bad. After new applications have been scored and decided upon, the performance of the accepted accounts again must be tracked and reports created. By doing so, the scoring rules can be validated and possibly substituted, the acceptance policy finely tuned and the current risk exposure calculated [21]. Currently credit scoring has become indispensable for any financial house. Both the US federal loan home corporation and the US National Mortagage Corporation have encouraged mortgage lenders to use credit scoring which should provide consistency for under writers. The Basel committee on banking supervision is an international organization which formulates broad supervisory standards and guidelines for banks.

The increasing complexity of credit instruments, the vitality of the economic conditions and the importance of risk management in minimizing credit losses of credit portfolios impose the need for decision support systems with learning capabilities for dynamically analyzing various sources of historical data and capturing complex relations amongst the most important attributes for credit evaluation.

1.1 Challenges faced by credit scoring:

a) Volume and Complexity of Data: Typically a financial house is characterized by millions of customers and high number of customer transactions per second over multiple channels (e.g., internet, telephone, ATM, branch offices). Thus a need for maintaining a huge database is indispensable. So the decision should be made in real time. A credit card fraud detection tool is only useful if it can identify and stop a fraudulent transaction immediately. This requirement establishes severe constraints on the tools used for the purpose of credit risk assessment. Ideally, detection systems should work on data at three levels: transaction, account, and customer-levels. Transaction level data include information about the current transaction such as amount, date, time, or location; account-level data includes information about the account history with the bank, such as average balance or account age; finally, customer level data includes information such as credit risk scores, stated income, or number of accounts with the institution. But usage of all three types of data is not feasible, as it will require quick retrieval of information from very large databases holding all account-holders' records. As a result, algorithms aimed at real-time detection will often need to make a decision solely based on the data present in the current transaction, and at most, a very limited amount of cached customer information.

b) Class Imbalance: Financial crimes are rare events, resulting in severe class imbalance: The number of defaulter cases is very less as compared to non defaulter cases. The classification of rare events is a common problem, that has been studied well in past.. A simple solution often used to balance the highly skewed class distributions is to subsample the majority class.

c) Concept Drift: One of the main objectives of the fraud detection methods is to identify general patterns of suspicious behavior. But even the formulation of this problem presents a challenge as these patterns are very dynamic and continuously evolve over time to bypass existing detection methods. Models must be

continuously validated and adapted to accommodate these changing distributions and patterns.

d) Class Overlap: Another challenge faced during predictive risk assessment that criminals often try to hide their activities by making illegal transactions seem as normal as possible, resulting in a extensive overlap between the defaulter and non defaulter classes [23].

Current credit risk regulations are governed by Basel Committee on Banking Supervision - the New Basel Capital Accord (Basel II, 2004) which is an amendment of Basel I Capital Accord. The main objectives of Basel I were to promote the soundness and stability of the banking system and adopt a standard approach across banks in different countries. Although it was initially intended to be only for the international active banks in the G-10 countries, it was gradually adopted by over 120 countries and treated as a global standard. However, the shortcomings of the Basel I became increasingly apparent over time. After the successive rounds of proposals between 1999 and 2003, the Basel II Capital accord was found in mid 2004. The main objectives of this revised capital adequacy framework are integrating an effective approach to supervision, incentives for banks to improve their risk management and measurement, and risk-based capital requirements.

1.2 Credit Card Scoring model development:

Step1. Definition of Defaults: The first step in a credit scoring model development project is defining the default event. Traditionally, credit rating or scoring models were developed upon the bankruptcy criteria. However, banks also meet with losses before the event of bankruptcy [14]. Therefore, in the Basel II Capital Accord, the Basel Committee on Banking Supervision gave a reference definition of the default event and announced that banks should use this regulatory reference definition to estimate their internal rating-based models. According to this proposed definition, a default is considered to have occurred

with regard to a particular obligator when either or both of the two following events have taken place [11].

Step2. Input Characteristics: The next step is the pre-selection of input characteristics to be included in the sample [13, 14]. Comparing with importing a snapshot of the entire database, pre-selecting input characteristics makes the model development process more efficient and increases the developer's knowledge of internal data.

Step3. Time Horizon: It refers to the period over which the default probability is estimated. The choice of the time horizon is a key decision to build up a credit scoring model [36]. Depending on the objective for which the credit risk model was developed (estimating the short-term or medium-long-term default probability), the time horizon varies. For most financial companies, it is common to select one year as a modelling horizon, as on the one hand one year is long enough to allow banks to take actions to mitigate credit risk, and on the other hand new obligor

Step4. Data Splitting: The statistical assessment of the performance of a predicting scoring model is in general highly sensitive to the data set. So the given data set at hand should be large enough to be randomly split into two data sets: one for development and the other for validation [36]. Normally, 60% to 80% of the total sample is used to estimate the model and the remaining 20% to 40% of the sample is set aside to validate the model.

Step5. Data Exploring: Before initiating the model development, it is very useful to calculate simple statistics for each characteristic, such as mean, median, standard missing Values and Outliers. Most financial industry data contain missing values or outliers that must properly be managed. Several methods with respect to dealing with missing values are available, such as removing all data

with missing values or excluding characteristics or records that have significant missing values from the model, but this may result in too many data being lost.

Step6. Another straightforward way is substituting the missing values with corresponding mean or median values over all observations for the respective time period. While these three methods assume that no further information can be gathered from analyzing the missing data, this is not necessarily true - missing values are usually not random. Missing values may be part of a trend, may be linked to other characteristics, or may indicate bad performance. Therefore, missing values should be analyzed first, and if they are found to be random and performance neutral, they may be excluded or imputed using statistical techniques; otherwise, if missing values are found to be correlated to the performance of the portfolio, it is preferable to include missing values in the analysis

1.3 Assessing model performance

Model performance is one of the most important criteria for choosing a particular model. The true performance of a model is in its ability to classify accurately on "unseen" data. Misclassification matrix is usually used to measure the performance

One common method to represent the discriminatory power of a scoring model is the *receiver operating characteristic* (ROC) curve. The construction of an ROC curve is illustrated in Figure 1.2 which shows the possible distribution of the rating scores for default and non-default counterparties [38]. For a perfect rating model the distributions of defaulters and non-defaulters should be distinguished, but in the real world, perfect discrimination in general is not possible, then both distributions will overlap. V is a cut-of value which provides a simple decision rule to divide counterparties into potential defaulters and non-defaulters. Table 1.1 shows the various performance measures with respect to this cut off value.

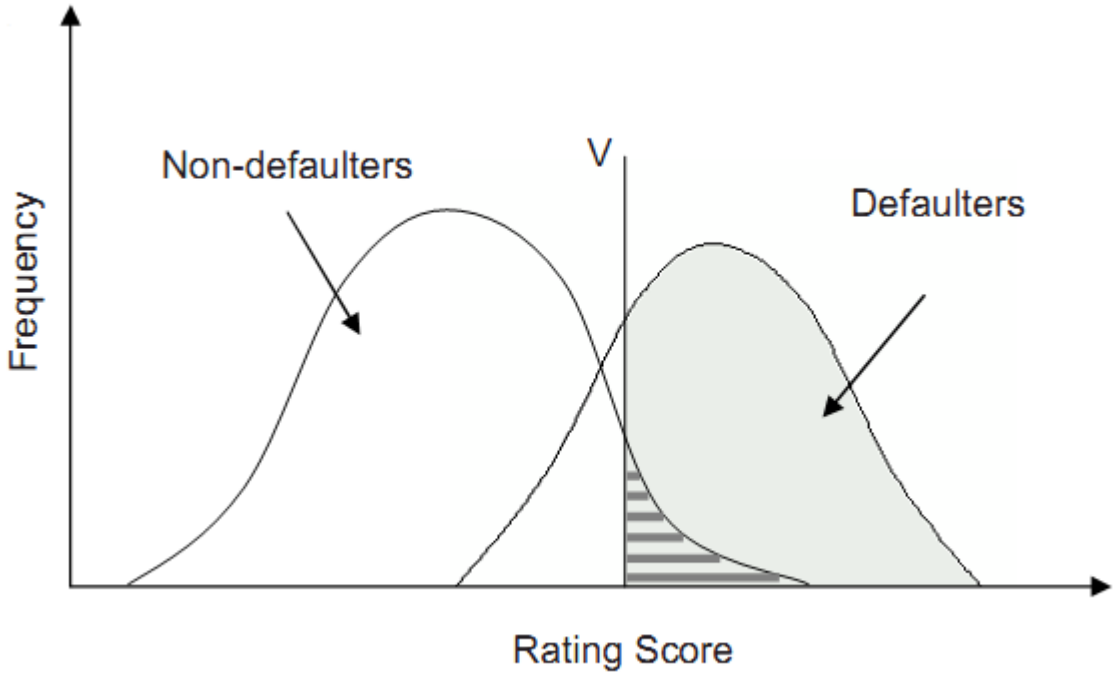

Figure 1.2 Distribution of defaulters and non defaulters

Table 1.1: Sensitivity and specificity

Scores	Defaults	Non-defaults
Above V	True positive prediction (Sensitivity)	False positive prediction (1-Specificity)
Below V	False negative prediction (1-Sensitivity)	True negative prediction (Specifity)

If an applicant with a rating score larger than V defaults or a non-defaulter has a rating score lower than V, then the prediction of the rating system is correct [18]. Otherwise, the rating system makes wrong prediction. The proportion of correctly predicted defaulters is called sensitivity and the proportion of correctly predicted non-defaulters is called specificity. For a given cut-off value V, a rating system should have a high sensitivity and specificity. In statistics, the false positive prediction (specificity) is also called type I

error, which is defined as the error of rejecting a null hypothesis that should have been accepted. The false negative prediction (sensitivity) is called type II error, which means the error of accepting a null hypothesis that should have been rejected. Traditionally, a Receiver Operating Characteristic (ROC) curve shows the false positive prediction rate.

1.4 Organization of thesis

a) The first chapter presents the introductory section regarding the credit card scoring and predictive modeling. A systemic approach is developed explaining the basel regulations and credit card scoring framework.

b) Second chapter is reserved to predictive modeling/data mining techniques, highlighting their main characteristics that support the decision making process. All the eight tools such as LDA (Linear Discriminant Analysis), SVM (support vector machines), MLP (multi layer perceptron), decision trees, Kernel density estimation, LR(logistic regression), GP(genetic programming), K neighborhood were implemented in Dtreg predictive modeling software. After this explanatory data analysis, a SAS macro provided restricted form of analysis, as now six of the above tools were evaluated further. The purpose of using Dtreg was initial statistical analysis and of employing SAS macro was to automate the credit scoring framework. Each tool is explained in detail with its corresponding merits and demerits.

c) The third chapter aims to cover the past comparative studies which have been done in the past in the area of credit scoring. Each of the study is explained in terms of variables used, data set employed, number of techniques and resulting performance compared.

d) The fourth chapter highlights the problem statement, shortcomings of the previous research and modifications required pertaining to credit scoring. A need for an Empirical study, which would assess performance on three data sets is emphasized to provide more general scenario. Additionally a framework to automate the credit card scoring process is recommended.

e) Chapter five contains personal contributions in the field of predictive modeling and automation regarding credit risk assessment. The need for the automation

and data profiling is solved by developing a *SAS macro* which would facilitate the comparison of all the predictive modeling tools in single comprehensive study. In the last chapter some aspects about conclusions and future research directions are highlighted.

Chapter 2 Literature Survey

Most of the attention in the literature focuses on fraud, particularly on credit card fraud. While there is some similarity among the various statistical methods, predictive techniques can be classified into two broad classes as supervised learning and unsupervised learning [37]. The performance of the different techniques is dependent on both quality and quantity of data.

One significant aspect of building credit scoring models is the selection of appropriate classification tool. This arise the need of an empirical comparison of various classification techniques in credit scoring. There are many traditional parametric classification techniques that have been applied to this type of problem, for example, discriminant analysis, linear regression, and logistic regression. Recently soft computing technique such as, decision trees, neural networks, have been applied extensively.

Abdou compared two credit scoring neural architecture, probabilistic NN (PNN) and multilayer perceptron (MLP), with discriminant analysis, probit analysis and logistic regression [11]. Their results demonstrated that PNN and MLP perform better than other models. In the same domain, compared ANN with decision tree analysis and logistic regression for credit risk classification and they concluded that decision tree technique performs better than ANN (with 74.2% of accuracy) and ANN (with 73.4% of accuracy) performs better than logistic regression (with 71.1% of accuracy).

Bahrammirzaee carried out comparative survey of various machine learning tools such as artificial neural network, expert systems and hybrid intelligent systems in finance applications [1]. The comparison was made pertaining to credit evaluation, portfolio management, and financial prediction and planning. The results depicted that modern machine learning tools are superior in terms of their classification rates as compared to traditional statistical tools [21].

In the same year, Angelini developed two NN systems, one with a standard feed-forward network and other one with special purpose architecture [11]. The system was validated with real-world data, obtained from Italian small businesses. They show that

NNs can be strong in learning and estimating the default tendency of a borrower if careful data analysis, data preprocessing and proper training are performed.

West Compared the accuracy of credit scoring of five ANN models: multilayer perceptron, mixture-of-experts, radial basis function, learning vector quantization and fuzzy adaptive resonance [9]. His study was based on two real world data sets :Australian and German. He employed 10 fold cross validation for enhancing his predictive power. He reported both good credit and bad credit rates. He benchmarked the results against five other traditional methods including linear discriminant analysis, logistic regression, k nearest neighbor, kernel density estimation and decision trees. Results demonstrated that the multilayer perception may not be the most accurate ANN model and that both the combination-of-experts and radial basis function NN models should be considered for credit scoring applications. Also, between traditional methods, logistic regression is more accurate method and more accurate than NN models in average case.

Zhang and Huang compared three data mining techniques namely back propagation, genetic programming and SVM for evaluating classification rates pertaining to credit scoring [13]. Initially they calculated the accuracy of each technique on German and Australian credit card data sets, which are easily available from machine repository of university of California. They observed that back propagation and genetic programming are better on average then SVM, but the classification accuracy of the latter was more stable as in each run the result produced was same. The results obtained from German credit card data were not that good as it contained large number of optimistic cases (who paid back their loan). They also proposed a combined model of these three techniques, which yielded better results as compared to individual methods in isolation. This combined model was based on the concept of majority voting. For one applicant, if there are two or three models with same classification rate A, then the customer is classified as case A. Otherwise, the classification result of the customer is the same as that of the model with highest accuracy.

Yeh and Yang proposed an optimal credit scoring model to reassess the default risk of credit card holders for credit card issuing banks in Taiwan. Their research adopted four

credit scoring models namely the linear discriminant analysis, decision tree, back-propagation neural network, and a hybrid method to evaluate the default risk. By comparing the evaluation results of these models, they concluded that the decision tree method has the best classification performance in terms of accuracy and sensitivity. The results were aimed at making the process of loan approval by financial houses more efficient.

Yen proposed a multistage NN ensemble learning model to evaluate credit risk at the measurement level [44]. The suggested model assessed the credit risk in various stages: In the very first stage different training data subsets were generated by employing a data sampling technique bagging pertaining to data shortage. In the subsequent stage, these training subsets were used to create the different NN models. In the third stage, the generated neural network models were again trained with different training datasets and accordingly the classification score and reliability value of neural classifier were estimated. In the fourth stage, feature selection was performed using appropriate technique. In the fifth stage, the reliability values of the selected NN models will be scaled into a unit interval by logistic transformation. Finally the selected NN ensemble members are joined to obtain final classification result by means of reliability measurement. The authors also used two credit datasets to verify the effectiveness of their proposed model. : One from was Japanese credit card screening and other was pertaining to UK corporation database. They compared their ensemble method with two other categories of classification tools: single methods such as logistic regression, support vector machines and hybrid methods such as neuro-fuzzy systems and neuro-support vector machine systems.

Thomas explored the techniques for predicting the financial risk involved in lending loans to customers [32]. Hardgrave presented a comprehensive and systematic approach to developing an optimal architecture of a neural network model for evaluating the creditworthiness of loan applicants [12]. Zurada investigated data mining techniques for loan granting decisions and predicted defaulters [20].

Here is a short-summary and review of the writings related with credit scoring algorithms and corresponding classification success. Ong applied Genetic Programming to classify good and bad customers [6]. Lee employed Classification and Regression Tree (CART) and Multivariate Adaptive Regression Splines (MARS) and identified better performance in comparison with the Discriminant Analysis, Logistic Regression, Neural Network and Support Vector Machine. Here, credit card dataset was used [39]. Sexton used GA-based algorithm, called Neural Network Simultaneous Optimization Algorithm on a credit dataset [33]. The performance was satisfactory and the model was able to identify significant variables among the other existing variables.

Fahrmeir once conducted a study with 1000 customers of a German bank. Here, the total number of cases is divided equally, but randomly [24]. As a result, 500 customers are taken for model construction and the rest 500 cases are kept alone for model validation. The dependent variable was default, that was coded as 0 (creditworthy) and 1 (non-creditworthy). The total number of predictor variables was 8. Some of the variables pertained to demographic characteristics of the customers, like sex(male/female), marriage (marital status). The other variables used to rationalize the behaviors, like bad (bad account), good (good account), duration (duration of credit in months), pay (payment of previous credits), private (professional/private use) and credit (line of credit). Multilayer Perceptron Architecture was used and a single hidden layer was adopted with two neurons. Backpropagation was employed using SAS Enterprise Miner.

In another study by the authors Xu, Chan, King, and Fu, Neural Network model is constructed with MLP and BP is adopted. The total number of samples is divided into Training Sample (40%), Testing Sample (30%) and Validation Sample (30%). The output variable is a good Indicator (for example, 0.9 if it is a good customer) or bad indicator (for example, 0.1 if it is a bad customer). Hence, the value of all dependent and independent variables fall into the category of 0 and 1. Activation Function and Combination Function is used in the model building. To test the predictive ability of the model, 100 cases (50 good and 50 bad) is used to generate forecasting. The performance of the model is very brilliant. The authors of this study expect that Artificial Neural

Network Models can be a good substitute for the traditional statistical techniques when the predictor variables require non-linear transformation.

A study carried out in the year of 1998 revealed that more than 60% of the largest banks of the USA are using credit scoring models to provide loans to small businesses and among them, only 12% have developed their own (Proprietary) models and 42% of the companies using These models to make automatic credit approval or rejection decisions. [43]. Credit scoring models require huge amount of past data of customers to make a scoring system and many of the financial institutions do not have that large database to make a proprietary model. Credit scoring models are becoming popular day by day. Moreover, although there is a rising trend of using different types of credit scoring models to classify good and bad customers, these models are not without limitations. One of the most important limitations is that credit scoring models take decision based on the data those are used for extended Loan, so it suffers from the *Selection Bias*. According to the same study, to avoid the selection bias, the best way is to include those samples that are accepted rather than only including the rejected samples. Another drawback of credit scoring models is that they do not take into account the changing behavior of borrowers. The borrower's behavior is changing from time to time. Moreover, the nature of the relationship among the variables might change with the progression of the time and new types of variables can come into existence that may be proved useful for better prediction accuracy.

Chapter 3 Problem statement

The above literature review reveals that most of the studies have evaluated the classification rates of various data mining/predictive modeling tools. But very few studies have evaluated most commonly employed data mining tools simultaneously for their comparison on multiple performance parameters such as misclassification rate, positive predictive power, negative predictive power, sensitivity, f-measure etc. In order to model the real word scenario in credit scoring problem, a replicated study that would evaluate the efficacy of most commonly used tools needs to be performed.

Limitations of past study/scope of current research

a) The current work facilitates the comparison of commonly used data mining/predictive modeling tools in a single comprehensive study. All these would be evaluated on multiple parameters at the same time such as specifity, sensitivity, PPU, NPU, f measure.

b) Also the purpose is to grant the freedom to the user such that all the tools can be *executed and compared at the same time*. Additionally, *the results can be stored in any output location such as any excel or document file.*

c) Table 3.1 shows the review of some of the major comparative studies carried in the past. Xu, Chan, King based their comparative study pertaining only to neural networks without incorporating any traditional statistical tool or machine learning tool. Kim and Neslin also assessed the performance of two types of neural networks using a data pertaining to German bank. Their research was more technical as they adhered to using training, validation and test data set. Wan performed comparison on three data mining tools named support vector machines, logistic regression and neural networks in a single study [8] as depicted in table 3.2. They also used cross validation to enhance the predictive power.

Bahrammirzaee evaluated five predictive modeling tools on single data set [1]. Angelina performed comparative assessment of three tools.

Table 3.1 Comparative review in terms of various parameters

Authors	**No of Tools**	**Data sets**	**Name of tools**
Angelini	2	Australian	Two types of ANN
Abdou	5	Australian	MLP,PNN,LDA,LR,PL
Bahrammirzaee	3	Australian, German	MLP, ANN
Yen, Yang	3	Japanese, UK	Ensemble, LR, SVM
Zang	3	German	Four types of MLP

Table 3.2 Comparative review in terms of study and features/parameters

Study	**Parameters compared**
Abdou	Misclassification rate, fratio
Bahrammirzaee	Missclassification rate
Yen, Yang	Accuracy, fratio
Angelini	Misclassification rate

In all the studies highlighted above each tool was evaluated individually. There was no mechanism involved which could produce the desired results of all the tools simultaneously. Also initially statistical analysis was performed as a separate task. Then identification of character and numeric variables was done manually. There was no framework present which could do this task implicitly.

Table 3.3 Potential features of a credit scoring problem

Features	Existing study
More then 2 datasets	No study
Automatic tracking of dataset variables	No study
Simultaneous output of all the tools	No study
Automated mechanism-use of macro	No study
Production of ad-hoc reports	No study

Objectives

Based on the highlighted shortcomings, this study accomplishes the following objectives:

a) **Automated mechanism**: Also none of the studies incorporate the automated mechanism of tracking character and numeric variables as highlighted in table 3.3. So numeric and character variables have to be identified. Then basic *descriptive and quantitative statistics* were produced.

b) **Macro for production of ad-hoc reports**: So a macro is designed to predictive model the credit scoring problem. The macro would simply take the input values and would produce desired results. So the task is to automate the entire process of credit scoring. It is sometimes quiet necessary to produce *ad-hoc reports. So in such cases macros are quiet useful.* The requirement was to just input the values and get the desired results.

c) **Performance assessment on multiple data sets**: Also majority of the studies assess the performance on one or two data sets Australian or German. But I would assess the performance on three data sets, with the Brazilian being the third one. This would provide more generic scenario as compared to the studies which have been already carried out. So all the techniques would be compared with respect to three data sets which is very rare till now.

Chapter 4 Data mining/predictive modeling tools

The classification techniques can be classified into *parametric and non-parametric problems.* Traditional statistical methods are parametric in nature. Parametric methods are based upon the assumptions of normally distributed population and estimate the parameters of the distributions to solve the problem [5]. Non-parametric methods, on the other hand, make no assumptions about the specific distributions involved, and are therefore distribution-free [15]. Figure 4.1 shows the typical flow chart of a parametric technique. Parametric techniques encompass linear regression, generalized linear regression, logistic regression and discriminant analysis. On the other hand neural networks, decision tree, genetic algorithms and k-nearest neighbor techniques fall in to the category of non-parametric methods.

4.1 Linear Discriminant Analysis (LDA)

It is a popular classification technique, which was developed by Fisher. It emphasizes on optimally separating two groups by on the basis of a function, which provides the maximum distance between their respective means [8]. The linear discriminant function used to map input values into output or target variable is:

$$Y_i = a_1 X_1 + a_2 X_2 + \ldots a_n X_n \quad (1)$$

where *Y* represents target variable and *a* , *a* ,..., *a* 1 2 indicates the attributes or the input values. In more technical terms , the discriminant analysis problem can be stated as : let Pl (good) and P2 (bad) denote the two mutually exclusive strata of the population and let a customer coming either from P1 or from P2 be represented by the random vector X=(X1, X2,...,Xp) of explanatory variables. It is assumed here that the rule of discrimination is based on a sample containing both response and explanatory variables randomly chosen from the entire population [38].

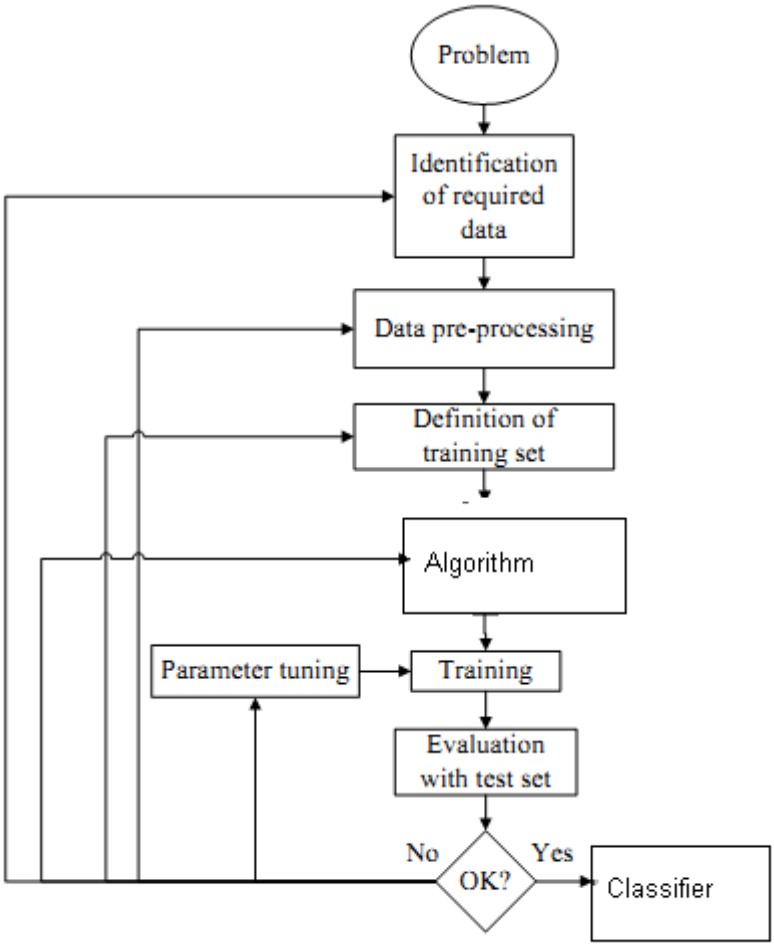

Figure 4.1 General Description of Training Algorithm

The advantages of LDA are its simplicity and that it can be easily estimated. Due to parametric nature, it requires the assumption of normality to be maintained.

- Many times, due to the qualitative nature of the data normal distribution is not followed, so the parametric nature of discriminant analysis is violated.
- The linear discriminant model assumes equality between the variance covariance matrices across all groups. Failure to adjust for inequality in the variance covariance matrices makes inefficient use of the available credit information and can severely distort the classification results.

- Prior probabilities should be incorporated to adjust for classification function adopted by the model [18]. Failure to incorporate the prior probabilities in the classification process will not minimize prediction errors and frequently has severe disruptive effect between group classification results.

4.2 Logistic Regression (LR)

Logistic regression model is a commonly employed statistical tool that predicts the membership of an object among two or more groups. It is constrained by the fact that the target or the *response variable should be binary in nature* [8, 21]. It provides a powerful technique analogous to multiple regression and anova for continuous responses. Reference 8 shows that the likelihood function of mutually independent variables y1, y2, y3,…..yn with binary responses is a member of exponential family with ($\log(\prod_1/(1-\prod_1)$,……$\log(\prod_n/(1-\prod_n))$ as a canonical parameter [33]. In technical terms the relationship between a canonical parameter and the vector of explanatory variables x is stated as:

$$\log(\prod_i(1-\prod i)=x\beta_i \quad (2)$$

This linear relationship between the logarithm of odds and the vector of explanatory variables results in a nonlinear relationship between the probability of y equals 1 and the vector of explanatory variables [19].

$$\prod_i \exp(x\beta_i/(1+\exp(x\beta i)) \quad (3)$$

Logistic regression is best suited for dealing with classification problems since the computed outputs can be expressed in terms of probabilities. Though logistic regression is used for categorical variables, the output is a continuous function, S-shaped curve that represents the probability associated with being in a particular category of the dependent variable. It can be an increasing or a decreasing curve [13]. Logistic regression has several similarities with linear regression. Logit coefficients are analogous to beta coefficients in the linear regression equation, the standardized logit coefficients are analogous to beta weights, and a pseudo R2 statistic is available to summarize the

strength of the relationship. Logistic regression does not assume linearity of relationship between the independent variables and the dependent, does not require normally distributed variables, and in general has less stringent requirements than linear regression.

4 .3 Decision Trees

They are also one of the commonly used tools for performing classification tasks. Decision tree learning is a method for approximating discrete valued target function, in which the learned function is represented by a decision tree [15]. The classification process begins by sorting down the tree from the root node, which provides the classification of the instance as shown in figure 4.3. Using appropriate search strategies, decision trees explore the attributes or the input values to generate understandable rules-like relationships with respect to the target variable. The objective is to find the optimum number of splits and determine the node to maximize the information gain. The most predictive variable is placed at the top node of the tree. The operation of decision trees is based on the *ID3 or C4.5 algorithms*. The algorithms make the clusters at the node gradually purer by progressively reducing disorder (impurity) in the original data set.

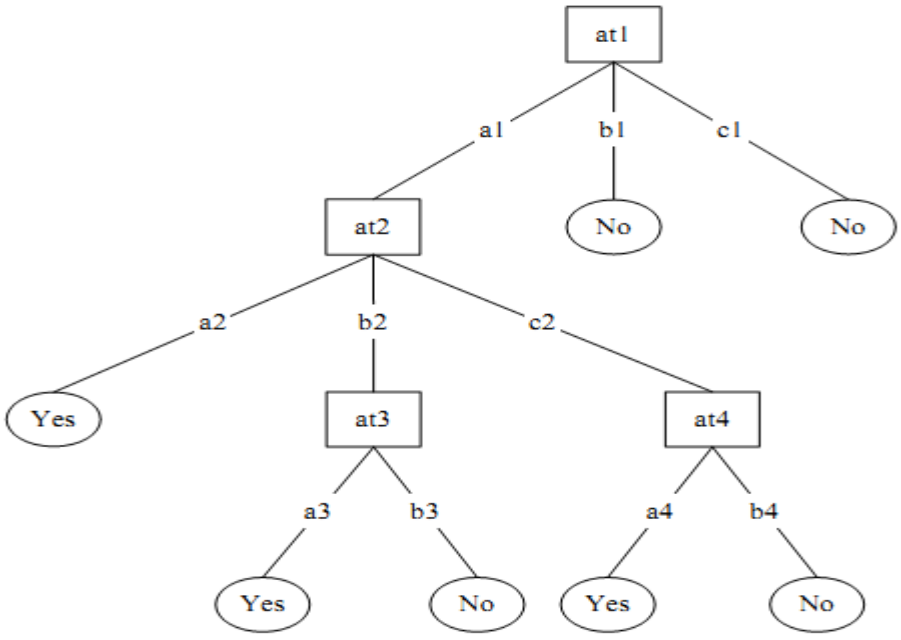

Figure 4.2 Decision Tree Example

Features of Decision Tree:

a) The central choice pertaining to tree splitting algorithm is selecting which attribute to test at each node of the tree [29, 38]. A *quantative measure called information gain* measures the extent how well a given attribute separates the training examples according to their target classification.

b) One of the most significant advantages of decision trees is the fact that knowledge can be extracted and represented in the form of classification (if-then) rules [21]. Each rule represents a unique path from the root to each leaf. Over fitting the training data is an important issue in decision tree learning. Because the training examples are only a sample space of all possible instances it is possible to add branches to the tree that improve performance on the training examples while decreasing performance instances outside this set.

c) One of the major advantages of decision trees is their comprehensibility [28, 36]. A novice learner can easily understand why a decision tree classifies an instance as belonging to a specific class. The output is a combination of the different class distributions that sum to 1. The assumption made in the decision trees is that instances belonging to different classes have different values in at least one of their features. Decision trees tend to perform better when dealing with discrete/categorical features in general.

d) The major problem associated with decision tree is of *overfitting [36]*. In general a hypothesis h is said to overfit training data if another hypothesis h′ exists that has a larger error than h when tested on the training data, but a smaller error than h when tested on the entire dataset. There are two common approaches that decision tree induction algorithms can use to avoid over fitting training data: i) Stop the training algorithm before it reaches a point at which it perfectly fits the training data, ii) Prune the induced decision tree. If the two trees employ the same kind of tests and have the same prediction accuracy, the one with fewer leaves is usually preferred.

4.4 Support Vector Machines(SVM)

Support vector machines implement the principle of structure risk minimization by constructing an optimal hyper plane [19]. In case of two group classification, the equation of hyper plane for a given input vector w is given by:

$$\mathbf{wx+b=0} \quad (4)$$

At the same time the input vector w should satisfy

$$\mathbf{Y_i[(wx + b)]-1 + €_I} \quad (5)$$

are ideally suited for data sets characterized by large feature space.

SVM's do well in classifying nonlinear separable groups. They do not require large training datasets and training converges to a unique global solution. These features make SVM's suitable for applications such as credit card fraud detection [17]. At the same time they suffer from many disadvantages. They are relatively more expensive to implement. The results `are not easily interpretable, and many parameters of the algorithm must be specified, e.g., the type of kernel function used to transform the feature space and all its parameters.

The selection of an appropriate kernel function is important, since the kernel function defines the transformed feature space in which the training set instances will be classified. Genton described several classes of kernels, However, he did not address the question of which class is best suited to a given problem. It is common practice to estimate a range of potential settings and use cross-validation over the training set to find the best one. For this reason a limitation of SVMs is the low speed of the training. Selecting kernel settings can be regarded in a similar way of choosing the number of hidden nodes in a neural network. As long as the kernel function is legitimate, a SVM will operate correctly even if the designer does not know exactly what features of the training data are being used in the kernel-induced transformed feature space.

4.5 Kernel Discrimination

A common approach to non parametric discriminant analysis with continous or discrete nature of predictor variables is to substitute non parametric estimates of group conditional densities in the definition of bayes rule [18, 42]. For a continous p-dimensional feature

vector x, a nonparametric estimate of the ith group conditional density $f_i(x)$ provided by kernel method is:

$$\mathbf{F_i(x)=n_i^{-1}h_i^{-p}\Sigma k_p((x-x_{ij})/h_i)} \quad (6)$$

where kp is a kernel function and hi is a smoothing parameter .The simplest class of kernels consists of probability density function that satisfies [18, 36]:

$$\mathbf{K(x) > 0}$$

$$\mathbf{\int R_p k(x)dx=1} \quad (7)$$

Table 4.1 highlights the various types of kernel functions employed for the purpose of classification.

Table 4.1 Several Kernel Functions

Kernel	$K(u)$
Uniform	$\frac{1}{2}I(\lvert u\rvert \leq 1)$
Triangle	$(1-\lvert u\rvert)I(\lvert u\rvert \leq 1)$
Epanechnikov	$\frac{3}{4}(1-u^2)I(\lvert u\rvert \leq 1)$
Quartic Biweight	$\frac{15}{16}(1-u^2)^2I(\lvert u\rvert \leq 1)$
Triweight	$\frac{35}{32}(1-u^2)^3I(\lvert u\rvert \leq 1)$
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}u^2)$
Cosine	$\frac{\pi}{4}\cos(\frac{\pi}{2}u)I(\lvert u\rvert \leq 1)$

4.6 Artificial Neural Network(ANN)

Neural network learning methods provide a robust approach to approximating real valued, discrete-valued and vector-valued target functions [15]. The motivation behind the neural networks comes from the fact that biological learning systems are built of very complex networks of interconnected neurons [3]. Typically, feed-forward networks with only three layers (input, hidden, and output layers) are used in fraud detection. The input to the neural network is the vector of features. The output layer gives the probability of the activity being criminal, which is used as a suspicion score. Back propagation is commonly used algorithm for training neural networks. The weights are initialized with random values, which are then changed in the direction that minimizes training error [12].

Neural networks are attractive tools in financial crime detection for a few reasons [26, 35].

- First, three-layer nets were shown to be capable of dealing with the highly skewed data that arise in this application.
- Second, once they are trained, they can analyze new data very quickly, an attribute that is necessary when trying to catch fraudulent transactions in real time.

However, neural networks also suffer from drawbacks:

- One major demerit is the need to select and adjust the structure of the network.
- Also the choice of the number of hidden states must be specified be made to optimize learning and generalization.
- Further, the performance of the classifier is very sensitive to the input vectors chosen, so significant attribute selection and preprocessing are necessary.

A particular type of ANN called multilayer perceptron (MLP) is especially suitable for classification [26]. It consists of one input layer, one or more hidden layers and one output layer, each consisting of several neurons. Each neuron processes its inputs and generates one output value that is transmitted to the neurons in the subsequent layer [14]. Each neuron in the input layer (indexed i = 1; 2; ; n) delivers the value of one predictor (or the characteristics) from vector x. When considering default/non-default behavior, one output neuron is satisfactory. Generally, it is cumbersome to properly determining the size of the hidden layer, because an underestimate of the number of neurons can lead

to poor approximation and generalization capabilities, while excessive nodes can result in over fitting and eventually make the search for the global optimum more difficult.

ANN depends upon three fundamental aspects, *input and activation functions of the unit, network architecture and the weight* of each input connection. Given that the first two aspects are fixed, the behavior of the ANN is defined by the current values of the weights. The weights of the net to be trained are initially set to random values, and then instances of the training set are repeatedly exposed to the net. The values for the input of an instance are placed on the input units and the output of the net is compared with the desired output for this instance. Then, all the weights in the net are adjusted slightly in the direction that would bring the output values of the net closer to the values for the desired output. There are several algorithms with which a network can be trained.

However, the most well-known and widely used learning algorithm to estimate the values of the weights is the *Back Propagation (BP) algorithm*. Generally, BP algorithm includes the following six steps [35]:

a) Present a training sample to the neural network.
b) Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
c) For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
d) Adjust the weights of each neuron to lower the local error.
e) Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to
f) Neurons connected by stronger weights.
g) Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.
h) With more details, the general rule for updating

Over fitting is one of the major problems encountered while training neural networks. A neural network can perfectly fit the training data by adequately increasing the

dimensionality of the neural network. However, such a model will be poor in predicting future cases, for which results are unknown. This problem of is solved by using special data set known as validation data set not seen by the model to validate and restrict the dimensionality of the model. Testing data is a set of observations unused during model building process. Testing data can measure the predictive power of the neural network. Finally, run data refers to data for which output variable is unknown, the neural network model can be used on this data for decision-making.

Neural networks allow nonlinear relations and complex interactions among predictor variables and thus are preferred over parametric methods. When such relations are not present, in theory, neural network models should at least perform as well as the parametric statistical models.

4.7 Instance-Based Learning Algorithms

They are commonly known as *lazy-learning algorithms*. Learning in these algorithms consists of simply storing the presented training data. When a new query instance is encountered a set of similar related instances is retrieved from memory and used to classify the new query instance [15]. Such predictive tools can construct different approximation to the target function for each distinct query instance that must be classified.

Lazy-learning algorithms require less computation time during the training phase as compared to their counterparts such as decision trees, neural and Bayesian network but more computation time during the classification process [21, 43]. One of the most straightforward instance-based learning algorithms is the nearest neighbor algorithm. The k-nearest neighbor classifier serves as an example of a non-parametric statistical approach. When given an unknown case, a K-NN classifier searches the pattern space for the k training cases that are similar to unknown cases. These k training cases are the K-nearest neighbors" of the unknown cases.

K-NN can also be useful when the dependent variable takes more than two values: high risk, medium risk and low risk. Generating the nearest-neighbor rule is very computationally intensive ($O(n^2)$ process) and can take considerable computational time

for large datasets [8]. K-NN also requires an equal number of good and bad sample cases for better performance.

4.7.1 Features of Instance Based Algorithm

a) The choice of k also affects the performance of the k-NN algorithm [15]. This can be determined experimentally. Starting with k=1, we use a test case to estimate the error rate of the classifier. This process is repeated each time by incrementing k to allow for one more neighbors. The K-value that gives the minimum error rate may be selected. In general, larger the number of training samples is, the larger the value of k will be.

b) The choice of k affects the performance of the kNN algorithm. A typical k-nearest neighbour classifier might incorrectly classify a query instance due to some of the reasons [24, 16]. When noise is present in the locality of the query instance, the noisy instance(s) win the majority vote. A larger k could solve this problem. When the region defining the class, or fragment of the class, is so small that instances belonging to the class that surrounds the fragment win the majority vote. A smaller k could solve this problem.

c) One major weakness associated with instance based algorithms is that cost of classifying the new instances can be high. The reason for this is that all the computation is performed at classification time rather than when the training samples are first encountered. Second weakness pertaining to instance based techniques is that they typically consider all the attributes of the instances when attempting to retrieve similar training examples from memory.

4.8 Genetic Programming

Genetic programming(GP) is a machine learning tool motivated by analogy to biological evolution [15]. Basically genetic programming searches for best candidate hypothesis. The best hypothesis refers to the one which optimizes a predefined numerical measure for the problem at hand, called the hypothesis fitness. On each iteration, all members of the population are evaluated according to fitness function. A new set is generated by probabilistically selecting the fit individuals from the current population.

A GP algorithm works on a population of individuals, each of which represents a potential solution to a problem. In order to solve a problem using GP it is necessary to specify the following [35]:

• The terminal set: A set of input variables or constants.

• The function set: A set of domain specific functions used in conjunction with the terminal set to construct potential solutions to a given problem. For symbolic regression this could consist of a set of basic mathematical functions, while Boolean and conditional operators could be included for classification problems.

• The fitness function: Fitness is a numeric value assigned to each member of a population to provide a measure of the appropriateness of a solution to the problem in question.

• The termination criterion: This is generally a predefined number of generations or an error tolerance on the fitness.

Genetic algorithms are different from other search algorithms in the way they take the initial solutions. Other algorithms browse through the surface of the solution starting from a single point. Thus the starting point of the solution can significantly affect optimization in case of complex surfaces. GAs start with a set of solutions all across the surface of the solution, and take the solutions that do particularly well, and then mix and match (mutate and blend) the attributes of these solutions and then reevaluate them. The original set of solutions is discarded. This is done till the improvement in the solution seems to be negligible as shown in figure 4.3.

In general GA model building comprises number of iterations of the following steps [21]:

- An initial population of solutions is taken and encoded as binary strings
- The strings are evaluated for fitness and a set of solutions is taken
- These strings are copied in proportion of their fitness
- Pairs of strings are randomly matched and string segments are swapped (*Crossover*)
- A small proportion of bits on each string are randomly mutated (*Mutation*)

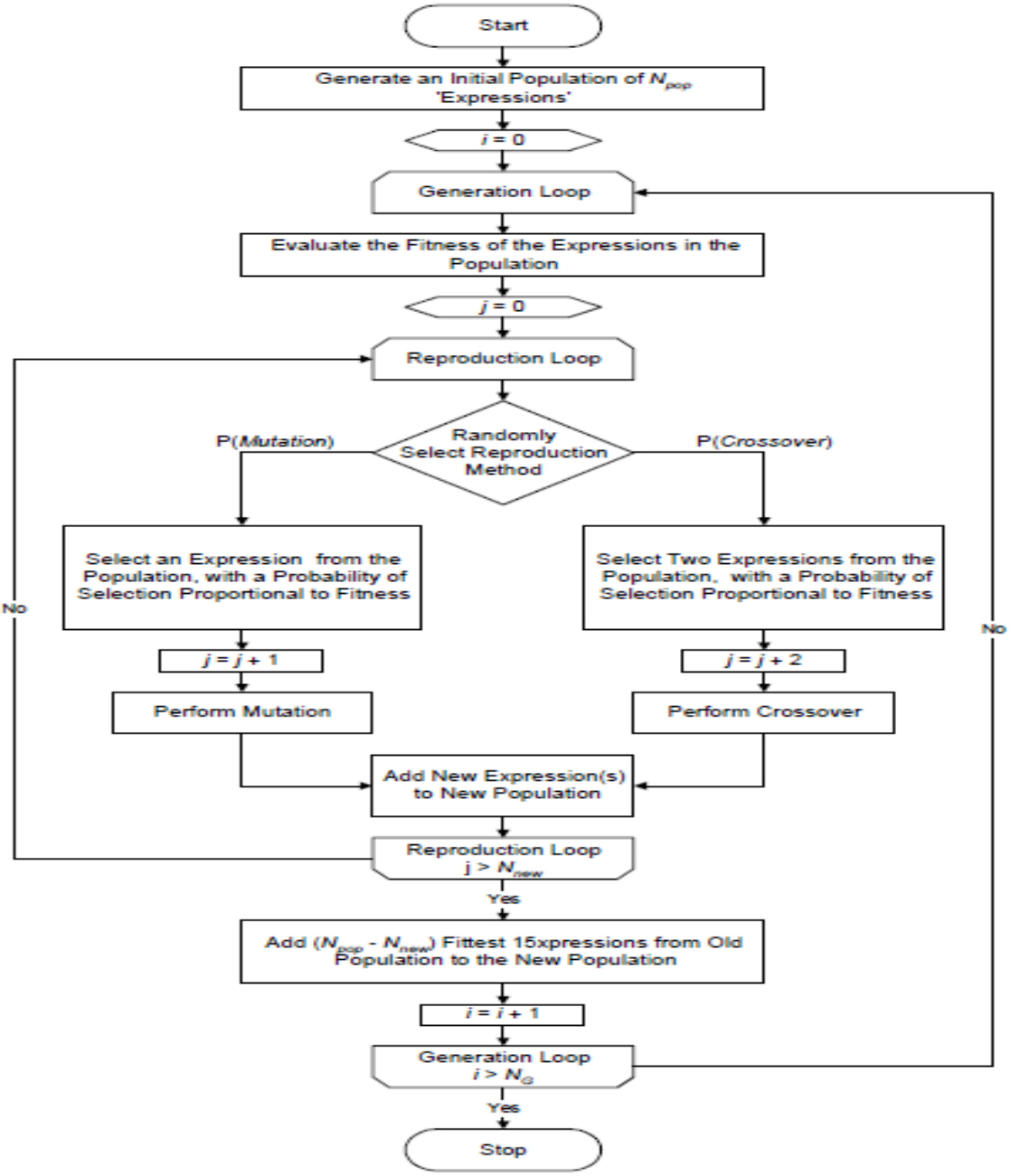

Figure 4.3 Genetic Programming Flow chart

Chapter 5 Macro Designing in SAS

Macros

A macro is a general purpose program which is required for the purpose of automation. It lends flexibility to your code as you have to just input the parameters such as the name of the data set, target or the response variable. The entire process of execution of macro should be transparent to the novice user. Macro programming enhances reusability and efficiency. As the growth of the industry is expanding, the need of automated macros is indispensable. In credit scoring, very few macros exist which automate the predictive modeling process.

This chapter represents a macro which would automate the comparison of various predictive modeling/data mining techniques in a single comprehensive study. The macro would work on two labeled data sets Australian and German credit card data, where the defaulter status is known. The need of such macro arises as the various techniques have to be assessed in terms of various statistical parameters such as *PPU, NPU, sensitivity and specifity.*

The execution of the macro can be divided into stages

5.1 Profiling of the data/ Assessing data quality

Firstly the macro would identify the numeric and character variables. *The need for this arises as both of the character and numeric variables should be processed independently.* If the variable is numeric, then various descriptive statistics such as mean, maximum value, minimum value, mode, coefficient of skewness, percentiles (P1, P10, P25, P50, P75, P90, P100). If the variable is character, then to identify the spread of data various statistics such as N, missing values, top 5 most frequent values. bottom 5 frequent values, shortest and largest length variables. Data quality can be assessed by identifying the various statistics pertaining to both numeric and quality variables. It would automatically

generate an excel file which would contain all the numeric variables.

Once the data type has been identified it would help to identify the various transformations required. For example character variables need to be converted into numeric and vice versa. In particular, kernel density estimation may require all the variables converted in to numeric.

The structure of the macro would be

%comparative(parameter1, parameter2,..paramertern)

To be specific, macro is executed as %comparative(dsn=, target=)

Where dsn is the name of the data set, target represents the response variable and options represent the various options that can be requested such as the location of the output, number of tools to be compared. By default the macro would do comparative analysis of all the six tools.

So the specific calls to the macro are stated as

%comparative(dsn=Australian, target=dep)

%comparative(dsn=german,target=dep, output=temp, method1=LR, method2=LDA)

So in this macro call the output would be stored in file C:/temp and only two tools LR and LDA would be assessed.

5.2 Comparative assessment of all tools:

Second step in the macro is to assess the comparative performance of all the tools such as LR, LDA, MLP, DT, kernel density estimation and k-neigbhourhood. But the macro would facilitate comparison of only those tools which are available in SAS enterprise miner. Then the macro would output the various results associated with the technique.

PSEUDOCODE

//Run proc contents to identify numeric and character variables

//Now create macro variable lists with Proc compare for each of the numeric and character variables.

//If type=1 place it into numeric variable lists

//Else

//If type=2 place it into character variable list

/ /For numeric profiling

//Initialize the various variables in sas such as pertaining to descriptive and quantile statistics

//Count the number of words and options specified in the numeric lists

//***Store them into macro variables with the help of call symput routines***

//Perform match merging of the variable keywords,

//%DO ii = 1 %TO &Num_ST_Keywords ;

//%IF %upcase(&&ST_KEYWORD&ii) = N %THEN %DO ;
//output n=&VAR_LIST. out=n;
// %END ;

//%IF %upcase(&&ST_KEYWORD&ii) = MIN %THEN %DO ;
// output min=&VAR_LIST. out=min;
%END ;
/ / %IF %upcase(&&ST_KEYWORD&ii) = MEAN %THEN %DO ;
output mean=&VAR_LIST. out=mean;
//%END ;
%IF %upcase(&&ST_KEYWORD&ii) = STD %THEN %DO ;
output std=&VAR_LIST. out=std;
%END ;
// %IF %upcase(&&ST_KEYWORD&ii) = CSS %THEN %DO ;

```
            output css=&VAR_LIST.       out=css;
        %END ;
 // Continue till desired
```

Repeat the entire procedure for quantile statistics keywords;

```
%IF &QS_STAT_LIST = 1 %THEN %DO
   output pctlpts=99 pctlpre= %DO counter=1 %TO &no_of_words; P&counter %END
     out=p99 (rename=(%DO mm=1 %TO &no_of_words; P&mm.99 =
&&VAR_WORD&mm %END
     output pctlpts=95 pctlpre= %DO counter=1 %TO &no_of_words; P&counter %END
out=p95 (rename=(%DO mm=1 %TO &no_of_words; P&mm.95 =
&&VAR_WORD&mm %END
 output pctlpts=90 pctlpre= %DO counter=1 %TO &no_of_words; P&counter %END ;
     out=p90 (rename=(%DO mm=1 %TO &no_of_words; P&mm.90 =
&&VAR_WORD&mm %END
```

//Apply proc transpose to get the variables in vertical order merge all the temporary datasets

Sample SAS code:

```
options replace;
%macro profiling();
options mlogic symbolgen mprint;
*name of the input data set;
%let dsn = sasuser.hospital;
*use of proc contents to identify numeric and character variables ;
proc contents data=&dsn out=temp;
run;
*creation of numeric macro variable list;
Proc sql noprint;
  select name
  into :num_vars separated by  " "
```

```
  from temp
  where type=1;

  select name
  into :char_vars separated by " "
  from temp
  where type=2;

quit;
```

character profiling

```
//read the input data file and import it into sas;
 //perform the basic text conversion;
//then initially implement linear discriminant analysis and logistic regression;
//specify the prior probability for the both the techniques;
  //implement the neural networks
//Implement the decision trees
Check for base cases
  {For each attribute a find the feature that best
divides the training data such as information  gain, according to entropy
Let a best be the attribute with the highest normalized information gain
 Create a decision node that splits with respect to minimum pruning
  Recursively traverse sub-lists obtained by splitting on a best and add those
nodes as children of node
```

}

procedure decision trees(Testing Instances)

for each testing instance

{

find the k most nearest instances of the training set according to a

distance metric

// Resulting Class= most frequent class label of the k nearest instances

}

Sample code and invocation

```
%MACRO Decision tree(DATA=,TARGET=,TTYPE=,INP=,INPTYPE=);
PROC DMDB BATCH DATA=&DATA OUT=DMTRAIN DMDBCAT=CATTRAIN ;
 CLASS  &TARGET.(DESC)
       &INP ;
 TARGET &TARGET ;
RUN ;
 PROC SPLIT DATA=DMTRAIN DMDBCAT=CATTRAIN
    CRITERION=PROBCHISQ
    SPLITSIZE=2
  PADJUST=NONE
    MAXBRANCH=2
  SUBTREE=LARGEST
  ASSESS=PROFIT
  VALIDATA=VALID
    OUTTREE=TREE ;
DECISION DECDATA=T3.DECDATA DECVAR=RESP NORESP  ;
CODE FILE="C:\SUGI30\&INP..SAS" DUMMY  ;
 INPUT &INP /LEVEL=&INPTYPE ;
```

```
 TARGET &TARGET /LEVEL=&TTYPE ;
RUN ;
%MEND SPLITV;
Procedure neural networks(no of hidden layers)
{
 //Define the response and independent variable
   //define the training and test data sets
     //prevent over pruning by using validation data set
// store the results in an external file
// format the data set if required
}
```

SAMPLE CODE

```
proc dmdb data=ravi out=dmddata dmdbcat=dmcdata;
class dep; * class statement is required for a ordinal-valued or nominal-valued
           target variable;
var    inde2 inde5 inde8;
run;
proc discrim method=npar kernel=normal r=0.07 crossvalidate;
class dep;
var inde1-inde20;
run;
* Neural network SAS procedure with ordinal or nominal categorical target
```

```
   variable based on the MLP algorithm ;
proc neural data=ravi dmdbcat=dmcdata ranscale=.1 graph random=99999;
                        * read data mining data set ;
input  inde2  inde5 inde8          / level=int;  * standardized input interval variable; target
dep          / level=nom;  * nominal target variable;
   archi mlp hidden=2;              * MLP architecture with two hidden units;
   prelim 5;                        * 5 preliminary runs;
   train;                           * this statement must be specified to
                                    train the network model and must be
                                    placed after all neural network
                                    modeling configuration statements;
   score data=ravi out=pred outfit=fit; * Create a score output data set;
   code file='c:\temp\code.sas';    * Create training code file that can be used
                                   to fit new data;
run;
```

Chapter 6 RESULTS

Comparative study is carried out on Australian data set. The eight techniques are assessed using the k-fold cross validation. But unfortunately since kernel density estimation and k-neighbourhood does not directly facilitate 10-fold cross validation, the parameters used to judge the efficacy of these techniques could not be calculated. So leave one cross validation approach is adopted in order to allow the comparison of more tools. Since leave one out cross validation is a special case of k-fold crosses validation when in the latter one the value of k is set to number of original cases. So the techniques are also assessed using 688 fold cross validation. Kernel density estimation and k-neighbourhood does not support k-fold cross validation. So the corresponding results can not be evaluated for them. All these parametric and non-parametric techniques are evaluated on parameters such as misclassification rate, positive predictive value(ppu), negative predictive value(npu), specificity and sensitivity. In case of LDA and logistic regression equal prior probability and misclassification costs were assumed. Support vector machine employed RBF kernel for non-linear mapping and produced 517 support vectors.

Table 6.1 Comparative evaluation using 10 fold cross validation

	Misclassification rate	Positive predictive value(ppu)	Negative Predictive value(npu)	sensitivity	Specifity
LDA	15.239	79.82	89.49	87.91	87.25
Logistic regression	13.7	78.8	76.8	*	*
MLP	14.078	83.6	87.83	84.97	86.68
Decision tree	14.369	78.67	93.29	92.80	79.90
GP	15.239	79.47	89.94	88.56	81.72
SVM	13.208	83.49	89.67	87.58	86.16
Kernel density estimation	*	*	*	*	
K-neighbourhood	*	*	*	*	

***K fold crossvalidation option not supported in these cases**

Table 6.2 Comparative evaluation using 688 fold cross validation or leave one out crossvalidation

	Misclassification rate	PPU	NPU	Sensitivity	Specifity
LDA	15.965	84.78	95.11	94.44	86.42
Logistic regression	13.7	82.8	89.7	*	*
MLP	12.917	85.11	88.68%	85.95	87.99
Decision tree	14.369	78.67	79.90	92.81	79.90
GP	16.546	77.27	89.91	88.89	79.11
SVM	13.498	82.97	89.23	87.58	85.64
Kernel density estimation	19.7	80.08	87.89	82.8	87.8
K-neighbourhood	18.2	79.08	85.02	84.50	82.20

***The parameters cannot be calculated**

Table 6.3 Neural network procedure output

Preliminary Training Run	Starting Random Seed	Objective Function Value	Number of Iterations	Terminating Criteria
1	99999	0.654442904273	17	FCONV
2	1134464880	0.654504774971	20	
3	588078564	0.654908713447	20	
4	2120446311	0.653824977988	20	
5	1482940400	0.643282967269	20	

Table 6.4 Parameter estimates of neural network

N	Parameter	Estimate	Gradient Objective Function
1	inde2_H11	0.787125	-0.000032783
2	inde3_H11	-0.527192	-0.000061510
3	inde2_H12	-0.891382	0.000443
4	inde3_H12	2.839908	0.000115
5	BIAS_H11	0.451846	0.000033370
6	BIAS_H12	0.033569	0.000092746
7	H11_dep0	-1.138974	-0.000015730
8	H12_dep0	-1.057200	0.000013456
9	BIAS_dep0	0.287843	-0.000013259

Table 6.5 Levenberg-Marquardt Optimization

Minimum Iterations	0
Maximum Iterations	100
Maximum Function Calls	2147483647
Maximum CPU Time	604800
ABSGCONV Gradient Criterion	0.00001
GCONV Gradient Criterion	1E-8
GCONV2 Gradient Criterion	0
ABSFCONV Function Criterion	0
FCONV Function Criterion	0.0001
FCONV2 Function Criterion	0
FSIZE Parameter	0
ABSXCONV Parameter Change Criterion	0
XCONV Parameter Change Criterion	0
XSIZE Parameter	0
ABSCONV Function Criterion	0.00139886
Trust Region Initial Radius Factor	1
Singularity Tolerance (SINGULAR)	1E-8

Table 6.6 Output of Proc Neural Network Procedure

Iteration	Function Calls	Active Constraints	Objective Function	Objective Function Change	Max Abs Gradient Element	Lambda	Ratio between actual and predicted change
1	3	0	0.64328	1.125E-6	0.000446	0.0521	0.146
2	4	0	0.64328	1.092E-6	0.000544	0.150	0.102
3	5	0	0.64328	8.262E-7	0.000491	0.115	0.0733
4	6	0	0.64328	4.906E-6	0.000237	0.742	0.510
5	7	0	0.64327	1.766E-6	0.000205	0.125	0.806
6	8	0	0.64327	2.393E-6	0.000297	0.0516	0.768
7	9	0	0.64327	3.347E-6	0.000419	0	0.787
8	10	0	0.64327	3.548E-7	0.000598	0	0.0400
9	12	0	0.64326	2.963E-6	0.000613	0.132	0.154
10	13	0	0.64326	2.321E-6	0.000759	0.163	0.106

Table 6.7: Frequency Output

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	508.4086	19	<.0001
Score	408.2660	19	<.0001
Wald	207.4696	19	<.0001

Table 6.8: Output of logistic regression

Effect	DF	Wald Chi-Square	Pr > ChiSq
inde1	1	0.0121	0.9126
inde2	1	0.0099	0.9206
inde3	1	0.0078	0.9296
inde4	2	5.0676	0.0794
inde6	7	14.2837	0.0464
inde7	1	1.8661	0.1719
inde8	1	134.4224	<.0001
inde9	1	4.3978	0.0360
inde10	1	5.9445	0.0148
inde11	1	1.8104	0.1785
inde12	2	16.2496	0.0003

Table 6.9 Association of Predicted Probabilities and Observed Responses

Dep	Variable Name	Frequency	Weight	Proportion	Prior Probability
0	0	383	383.0000	0.555878	0.500000
1	1	306	306.0000	0.444122	0.500000

Table 6.10 Classification summary of Logistic regression

Number of Observations and Percent Classified into dep			
From dep	0	1	Total
0	316 82.51	67 17.49	383 100.00
1	67 21.90	239 78.10	306 100.00
Total	383 55.59	306 44.41	689 100.00
Priors	0.5	0.5	

Table 6.11: Classification Summary of Kernel Discrimination

Number of Observations and Percent Classified into dep			
From dep	**0**	**1**	**Total**
0	383 100.00	0 0.00	383 100.00
1	0 0.00	306 100.00	306 100.00
Total	383 55.59	306 44.41	689 100.00
Priors	0.5	0.5	

Table 6.12 Classification Summary of K-neighbhourhood

Number of Observations and Percent Classified into dep			
From dep	0	1	Total
0	383 100.00	0 0.00	383 100.00
1	0 0.00	306 100.00	306 100.00
Total	383 55.59	306 44.41	689 100.00
Priors	0.5	0.5	

Table 6.13 Error Count Estimates for Logistic Regression

Error Count Estimates for dep			
	0	1	Total
Rate	0.1749	0.2190	0.1969
Priors	0.5000	0.5000	

Table 6.14: Classification Summary of Decision tree

Number of Observations and Percent Classified into dep			
From dep	0	1	Total
0	316 82.51	67 17.49	383 100.00
1	67 21.90	239 78.10	306 100.00
Total	383 55.59	306 44.41	689 100.00
Priors	0.5	0.5	

Table 15: Discrim procedure

Class Level Information					
dep	Variable Name	Frequency	Weight	Proportion	Prior Probability
0	0	383	383.0000	0.555878	0.500000
1	1	306	306.0000	0.444122	0.500000

Chapter 7 Conclusion

7.1 Conclusions

a) The research depicts that in case of both 10 fold and 688 fold cross validation, genetic programming yielded highest misclassification rates for classifying good cases in to bad and bad cases in to good. In case of leave-one–out cross validation, MLP model emerged to be clear winner as it yielded the lower classification rate.

b) Statistical techniques such as LDA, LR and Kernel density estimation, due to their quiet sensitive nature of data, were not able to classify bad cases into good and good cases in to bad irrespective of cross validation approaches incorporated.

c) In general Soft computing techniques such as decision tree and neural network were found to be more superior as compared to statistical tools such as LDA and LR. They yielded less misclassification rate and comparatively better PPU, NPU as compared to other tools.

d) Non parametric techniques did not performed well. The increase in value of k led to decrease in misclassification rate. The value of k was adjusted by hit and trail in case of k-neighbhourhood.

7.2 Future research

a) If the comparative study is carried out on most recent and real data set, then it could yield more meaningful results. The Australian and German credit card data sets have been used for quiet long. Since the defaulter pattern tends to change invariably, the results may not truly depict the actual scenario.

b) The current work does not facilitate the use of ensemble method. As the nature of data is dependent upon the characteristics the data set variables, so if ensemble or combined method is used, then the predictive model of the model can be judged more effectively. The results obtained from ensemble method can provide more technical insight.

c) Also the Bayesian networks can also be incorporated in to the study. They have gained much popularity in the past. So the current study can be expanded by also including Bayesian networks, which are based on the concept of posterior probability. Also the macro developed does not consider the SVM and GP, as they were not available in the SAS software. Since the purpose of the macro was to automate the entire credit scoring framework, so it is recommended to encompass as many tools as it can. So in future, efforts can be invested in exploring the possibility to include these two tools also.

REFERENCES

[1]. A. Bahrammirzaee, "Comparative survey of artificial Intelligence applications: artificial neural networks, expert systems, hybrid intelligence systems.", *Neural Computing and Applications*, vol. 19, no. 8, pp. 1165-1195, 2010.

[2]. Bolton, "Logistic regression and it's application in credit Scoring" Msc. Thesis, University of Pretoria, 2009.

[3]. C.J Speck, "Abstracts of Significant Cases Bearing on the Regulation of Insurance", *Journal of Insurance Regulation* vol. 23, no. 4, pp. 81-84, 2005.

[4]. C.L. Blake and C.J. Merz. 1998. UCI repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], University of California, Department of Information and Computer Science.

[5]. C.M. Bishop, "Pattern Recognition and Machine Learning", Springer, Ist edition, 2006

[6]. C.S. Ong, J.J Huang and G.H 2005, "Building Credit Scoring models Using Genetic Programming" *Expert Systems with Applications*, vol 29, no. 1, pp. 41-47, 2005.

[7]. D. Dutton and G. Conroy "A review of machine learning", *Knowledge Engineering Review* vol. 12, pp. 341-367, 1996.

[8]. D.M. Chickering, "Optimal Structure Identification with Greedy Search" *Journal of Machine Learning Research*, Vol. 3, pp. 507-554, 2002.

[9]. D. Martens, B. Baesens, T.V. Gesteland and J.Vanthienen, "Comprehensive credit Scoring Models using Rule Extraction from Support Vector Machines", *European Journal of Operational Research*, vol. 183, pp. 1466-1476, 2007

[10]. D. West, ”Neural Network Credit Scoring Model” *Computational Operation Research* vol. 27, no. 11/12, pp. 1131–1152, 2000.

[11]. E. Angelini, G.D Tollo and A. Roli , “A neural network approach for credit risk evaluation”, *The Quarterly Review of Economics and Finance*, vol. 48, no. 4. pp. 733-755, 2008.

[12]. G. Castellano, A. Fanelli, and M. Pelillo, “An Iterative Pruning Algorithm for Feedforward Neural Networks” *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 519–531, 1997.

[13]. H. Abdou, J Pointon and E-Masry. “Neural Nets versus Conventional Techniques in Credit Scoring in Egyptian Banking” *Expert System Application*, vol. 35, no. 3, pp. 1275- 1292, 2008.

[14]. H. Brighton and C. Mellish. “Advances in Instance Selection For Instance-Based Learning Algorithms.” *Data Mining and Knowledge Discovery*, vol. 6, pp. 153–172, 2002.

[15]. H. Zhang, Q Huang, Y. Chen. “Comparison of Credit Scoring Models” Third international conference of Natural Computation” 2007 .

[16]. I. Bruha, “From Machine Learning to Knowledge Discovery: Survey of Preprocessing and Postprocessing.” *Intelligent Data Analysis*, vol. 4, no. 4, pp. 363-374, 2000.

[17]. I. Guyon and A. Elissee, “An Introduction to Variable and Feature Selection.” *Journal of Machine Learning Research*, vol. 3, pp.1157-1182, 2003

[18]. J. Banasik, J. Crook, and L. Thomas, “Sample selection bias in credit scoring models” *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 822–832, 2003.

[19]. J. Cohen and P. Cohen, “Applied multiple regression / correlation analyses for the behavioral sciences”, Hillsdale, NJ: Erlbaum, 1983.

[20]. J. Han and M. Kamber, “Data Mining: Concepts and Techniques”, Morgan Kaufmann Publishers, USA, 2001.

[21]. J. Hand and W. Henley, “Statistical classification methods in consumer credit scoring,” *Computer Journal of the Royal Statistical Society” Series A* vol. 160, no. 3, pp. 523-541, 1997.

[22]. J. Zurada. and M. Zurada, “How Secure are Good Loans: Validating Loan-Granting Decisions and Predicting Default Rates on Consumer Loans” *The Review of Business Information Systems*, vol. 6, no 3, pp. 65-83, 2002 .

[23]. J. Galindo and P. Tamayo, “Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications” *Computational Economics*, 15:107–143, 2000.

[24]. K. Cuh, W.C. Tan and C.P. Goh, ”Credit scoring using data mining techniques”, Singapore Management Review, vol. 26, no. 2, pp. 25–47, 2004.

[25]. L. Fahrmeir, G. Tutz, “Multivariate Statistical Modelling Based on Linear Models” Springer, New York, 1994.

[26]. L.W. Glorfeld and B.C. Hardgrave, "An Improved Method for Developing Neural Networks: the Case of Evaluating Commercial Loan Credit Worthiness", *Computer & Operations Research*, vol. 23, no. 10, pp. 933-944, 1996.

[27]. M.J.A. Berry, and G.S. Linoff, ”Data Mining Techniques for Marketing, Sales, and Customer Support”, John Wiley & Sons, Inc, 1997 .

[28]. Murthy, “Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey”, *Data Mining and Knowledge Discovery*, vol. 2, no. 4, pp. 345–389, 1998.

[29]. N.C. Hsieh, “Hybrid mining approach in the design of credit scoring models,” *Expert Systems with Applications*, vol. 28, no. 4, pp. 655-665, 2005

[30]. P. Brazdil, C. Soares and J. Da. Costa, "Ranking Learning Algorithms: Using IBL and Meta-Learning on Accuracy and Time Results", *Machine Learning*, vol. 50, no. 3, pp. 251-277, 2003.

[31]. R. Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings. of the Second International Conference on Knowledge Discovery and Data Mining, 1996.

[32]. R.S. Sexton, S. Mcmurtrey and D.J. Cleavenger, "Knowledge Discovery using a Neural Network Simultaneous Optimization Algorithm on a Real World Classification Problem", *European Journal of Operational Research*, vol. 168, pp. 1009-1018, 2006.

[33]. S. Baik and J. Bala, "A Decision Tree Algorithm for Distributed Data Mining: Towards Network Intrusion Detection", *Lecture Notes in Computer Science*, vol. 3046, pp. 206 – 212, 2004.

[34]. "Data Mining Using Enterprise Miner Software: A Case Study Approach.", 1st edition, http://www.sas.com, SAS Institute.

[35]. Thomas. "A Survey of Credit and Behavioral scoring: Forecasting financial risks of lending to customers", *International Computer Journal of Forecasting*, vol. 16, no. 2, pp. 149-172, 2000.

[36]. T.H Lee, SC Jung, "Forecasting Credit Worthiness: Logistic vs. Artificial neural net." *Journal of Business Forecasting*, vol. 18, no 4, pp. 28–30, 1999/2000.

[37]. T. Diana, "Credit Risk Analysis and Credit Scoring – Now and in the Future", *Business Credit*, vol. 107, no. 3, pp. 12-16, 2006.

[38]. T.L. Bharatheesh and S.S Iyengar, "Predictive Data Mining for Delinquency Modeling", in Proceedings of *ESA/VLSI* pp. 99-105, 2004.

[39]. T.M. Mitchell "Machine learning" International edition Mcgraw Hill, 1997.

[40]. T. S. Lee and I.F. Chen, "A Two Stage Hybrid Credit Scoring Model Using Artificial Neural Networks and Multivariate Adapative

Regression Splines.", *Expert Systems With Applications*, vol. 28, no. 4, pp. 743-752, 2005.

[41]. W. Witten and E. Frank, "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005

[42]. W.E. Henley "Statistical aspects of credit scoring" Phd. Dissertation, The Open University, Milton Keynes, UK.1995.

[43]. W.S. Frame, A. Srinivasan and L. Woosley, "The Effect of Credit Scoring on Small-Business Lending", *JSTOR: Journal of Money, Credit and Banking*, vol. 33, no. 3, pp. 813-825, 2001.

[44]. Y. Lu, S. Wang, "Credit Risk Assessment with a Multistage Neural Network Ensemble Learning Approach", *Expert System Application*, vol. 34, no. 2, pp. 1434–1444, 2008.

[45]. Z. Huang, H. Chu, C.J. Hsu, W.H Chen and S. Wu " Credit rating analysis with support vector machines and neural networks: A market comparative study", *Decision Support System*" vol. 37, no. 2, pp. 543–558, 2004.

Paper communicated

Research paper named "Comparative Evaluation of various Predictive Modelling Techniques on Credit card Data" submitted to journal of ICJTE(International Journal of computer theory and technology) in April 2011.

Appendix

German credit card data: Two datasets pertaining to credit card applications is available in two formats [4]. The original data set which was contributed by provided by Prof. Hofmann, contains categorical/symbolic attributes. Strathclyde University produced another data set which consisted of all numeric attributes. This data set was edited and several indicator variables were added to make it suitable for certain algorithms which cannot deal with categorical variables. The data set contains 1000 records with 31 dependent variables and one response variable.

1 6 4 12 5 5 3 4 1 67 3 2 1 2 1 0 0 1 0 0 1 0 0 1 1
2 48 2 60 1 3 2 2 1 22 3 1 1 1 1 0 0 1 0 0 1 0 0 1 2
4 12 4 21 1 4 3 3 1 49 3 1 2 1 1 0 0 1 0 0 1 0 1 0 1
1 42 2 79 1 4 3 4 2 45 3 1 2 1 1 0 0 0 0 0 0 0 0 1 1
1 24 3 49 1 3 3 4 4 53 3 2 2 1 1 1 0 1 0 0 0 0 0 1 2
4 36 2 91 5 3 3 4 4 35 3 1 2 2 1 0 0 1 0 0 0 0 1 0 1
4 24 2 28 3 5 3 4 2 53 3 1 1 1 1 0 0 1 0 0 1 0 0 1 1
2 36 2 69 1 3 3 2 3 35 3 1 1 2 1 0 1 1 0 1 0 0 0 0 1
4 12 2 31 4 4 1 4 1 61 3 1 1 1 1 0 0 1 0 0 1 0 1 0 1
2 30 4 52 1 1 4 2 3 28 3 2 1 1 1 1 0 1 0 0 1 0 0 0 2
2 12 2 13 1 2 2 1 3 25 3 1 1 1 1 1 0 1 0 1 0 0 0 1 2
1 48 2 43 1 2 2 4 2 24 3 1 1 1 1 0 0 1 0 1 0 0 0 1 2
2 12 2 16 1 3 2 1 3 22 3 1 1 2 1 0 0 1 0 0 1 0 0 1 1
1 24 4 12 1 5 3 4 3 60 3 2 1 1 1 1 0 1 0 0 1 0 1 0 2
1 15 2 14 1 3 2 4 3 28 3 1 1 1 1 1 0 1 0 1 0 0 0 1 1
1 24 2 13 2 3 2 2 3 32 3 1 1 1 1 0 0 1 0 0 1 0 1 0 2
4 24 4 24 5 5 3 4 2 53 3 2 1 1 1 0 0 1 0 0 1 0 0 1 1
1 30 0 81 5 2 3 3 3 25 1 3 1 1 1 0 0 1 0 0 1 0 0 1 1
2 24 2 126 1 5 2 2 4 44 3 1 1 2 1 0 1 1 0 0 0 0 0 0 2
4 24 2 34 3 5 3 2 3 31 3 1 2 2 1 0 0 1 0 0 1 0 0 1 1
4 9 4 21 1 3 3 4 3 48 3 3 1 2 1 1 0 1 0 0 1 0 0 1 1
1 6 2 26 3 3 3 3 1 44 3 1 2 1 1 0 0 1 0 1 0 0 0 1 1
1 10 4 22 1 2 3 3 1 48 3 2 2 1 2 1 0 1 0 1 0 0 1 0 1
2 12 4 18 2 2 3 4 2 44 3 1 1 1 1 0 1 1 0 0 1 0 0 1 1

4 10 4 21 5 3 4 1 3 26 3 2 1 1 2 0 0 1 0 0 1 0 0 1 1

1 6 2 14 1 3 3 2 1 36 1 1 1 2 1 0 0 1 0 0 1 0 1 0 1

4 6 0 4 1 5 4 4 3 39 3 1 1 1 1 0 0 1 0 0 1 0 1 0 1

3 12 1 4 4 3 2 3 1 42 3 2 1 1 1 0 0 1 0 1 0 0 0 1 1

2 7 2 24 1 3 3 2 1 34 3 1 1 1 1 0 0 0 0 0 1 0 0 1 1

1 60 3 68 1 5 3 4 4 63 3 2 1 2 1 0 0 1 0 0 1 0 0 1 2

2 18 2 19 4 2 4 3 1 36 1 1 1 2 1 0 0 1 0 0 1 0 0 1 1

1 24 2 40 1 3 3 2 3 27 2 1 1 1 1 0 0 1 0 0 1 0 0 1 1

2 18 2 59 2 3 3 2 3 30 3 2 1 2 1 1 0 1 0 0 1 0 0 1 1

4 12 4 13 5 5 3 4 4 57 3 1 1 1 1 0 0 1 0 1 0 0 1 0 1

3 12 2 15 1 2 2 1 2 33 1 1 1 2 1 0 0 1 0 0 1 0 0 0 1

2 45 4 47 1 2 3 2 2 25 3 2 1 1 1 0 0 1 0 0 1 0 1 0 2

4 48 4 61 1 3 3 3 4 31 1 1 1 2 1 0 0 1 0 0 0 0 0 1 1

3 18 2 21 1 3 3 2 1 37 2 1 1 1 1 0 0 0 1 0 1 0 0 1 2

3 10 2 12 1 3 3 2 3 37 3 1 1 2 1 0 0 1 0 0 1 0 0 1 1

2 9 2 5 1 3 3 3 1 24 3 1 1 1 1 0 0 1 0 0 1 0 0 1 1

4 30 2 23 3 5 3 2 3 30 1 1 1 1 1 0 0 1 0 0 1 0 0 0 1

2 18 3 62 1 3 3 4 1 44 3 1 2 2 1 0 0 1 0 0 1 0 1 0 1

1 30 4 62 2 4 4 4 3 24 3 2 1 1 1 0 1 1 0 1 0 0 0 1 1

1 48 4 61 1 5 2 4 4 58 2 2 1 1 1 0 1 1 0 0 0 0 1 0 2

4 11 4 14 1 2 2 4 3 35 3 2 1 1 1 1 0 1 0 0 1 0 0 0 1

4 36 2 23 3 5 3 4 3 39 3 1 1 1 1 0 0 1 0 0 1 0 0 1 1

1 6 2 14 3 1 2 2 2 23 3 1 1 2 1 0 1 1 0 1 0 1 0 0 1

4 11 4 72 1 3 3 4 2 39 3 2 1 1 1 1 0 1 0 0 1 0 1 0 1

4 12 2 21 2 3 2 2 1 28 3 1 1 1 1 0 0 0 1 0 1 0 0 1 1

2 24 3 23 5 2 3 2 2 29 1 1 1 1 1 0 0 1 0 0 1 0 1 0 1

2 27 3 60 1 5 3 2 3 30 3 2 1 2 1 0 1 1 0 0 1 0 0 0 1

4 12 2 13 1 3 3 2 3 25 3 1 1 1 1 0 0 1 0 0 1 0 0 1 1

4 18 2 34 5 3 3 1 2 31 3 1 1 2 1 0 1 1 0 0 1 0 0 1 1

2 36 3 22 1 5 3 4 4 57 1 2 1 2 1 1 0 1 0 0 0 0 0 1 2

4 6 1 8 5 3 3 2 1 26 2 1 2 1 1 1 0 0 0 0 1 0 1 0 1

b

Australian credit card data: The data set is available from machine learning repository of University of California [4]. It contains 690 observations with 14 predictor variables and one response variable. These 14 predictor variables are divided into 8 categorical and six continuous variables. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. This dataset contains good mixture of both continuous and nominal attributes.

a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 a11 a12 a13 a14 a15
1 22.08 11.46 2 4 4 1.585 0 0 0 1 2 100 1213 0
0 22.67 7 2 8 4 0.165 0 0 0 0 2 160 1 0
0 29.58 1.75 1 4 4 1.25 0 0 0 1 2 280 1 0
0 21.67 11.5 1 5 3 0 1 1 11 1 2 0 1 1
1 20.17 8.17 2 6 4 1.96 1 1 14 0 2 60 159 1
0 15.83 0.585 2 8 8 1.5 1 1 2 0 2 100 1 1
1 17.42 6.5 2 3 4 0.125 0 0 0 0 2 60 101 0
0 58.67 4.46 2 11 8 3.04 1 1 6 0 2 43 561 1
1 27.83 1 1 2 8 3 0 0 0 0 2 176 538 0
0 55.75 7.08 2 4 8 6.75 1 1 3 1 2 100 51 0
1 33.5 1.75 2 14 8 4.5 1 1 4 1 2 253 858 1
1 41.42 5 2 11 8 5 1 1 6 1 2 470 1 1
1 20.67 1.25 1 8 8 1.375 1 1 3 1 2 140 211 0
1 34.92 5 2 14 8 7.5 1 1 6 1 2 0 1001 1
1 58.58 2.71 2 8 4 2.415 0 0 0 1 2 320 1 0
1 48.08 6.04 2 4 4 0.04 0 0 0 0 2 0 2691 1
1 29.58 4.5 2 9 4 7.5 1 1 2 1 2 330 1 1
0 18.92 9 2 6 4 0.75 1 1 2 0 2 88 592 1
1 20 1.25 1 4 4 0.125 0 0 0 0 2 140 5 0
0 22.42 5.665 2 11 4 2.585 1 1 7 0 2 129 3258 1
0 28.17 0.585 2 6 4 0.04 0 0 0 0 2 260 1005 0
0 19.17 0.585 1 6 4 0.585 1 0 0 1 2 160 1 0
1 41.17 1.335 2 2 4 0.165 0 0 0 0 2 168 1 0
1 41.58 1.75 2 4 4 0.21 1 0 0 0 2 160 1 0
1 19.5 9.585 2 6 4 0.79 0 0 0 0 2 80 351 0
1 32.75 1.5 2 13 8 5.5 1 1 3 1 2 0 1 1
1 22.5 0.125 1 4 4 0.125 0 0 0 0 2 200 71 0

1 33.17 3.04 1 8 8 2.04 1 1 1 1 2 180 18028 1

0 30.67 12 2 8 4 2 1 1 1 0 2 220 20 1

1 23.08 2.5 2 8 4 1.085 1 1 11 1 2 60 2185 1

1 27 0.75 2 8 8 4.25 1 1 3 1 2 312 151 1

0 20.42 10.5 1 14 8 0 0 0 0 1 2 154 33 0

1 52.33 1.375 1 8 8 9.460 1 0 0 1 2 200 101 0

1 23.08 11.5 2 9 8 2.125 1 1 11 1 2 290 285 1

1 42.83 1.25 2 7 4 13.875 0 1 1 1 2 352 113 0

1 74.83 19 1 1 1 0.04 0 1 2 0 2 0 352 0

1 25 12.5 2 6 4 3 1 0 0 1 1 20 1 1

1 39.58 13.915 2 9 4 8.625 1 1 6 1 2 70 1 1

0 47.75 8 2 8 4 7.875 1 1 6 1 2 0 1261 1