THE UNIVERSITY *of* EDINBURGH

Mathematical Programming models for classification problems with applications to credit scoring

Konstantinos Falangis

Thesis Presented for the degree of
Doctor of Philosophy
The University of Edinburgh
2013

Abstract

Mathematical programming (MP) can be used for developing classification models for the two–group classification problem. An MP model can be used to generate a discriminant function that separates the observations in a training sample of known group membership into the specified groups optimally in terms of a group separation criterion. The simplest models for MP discriminant analysis are linear programming models in which the group separation measure is generally based on the deviations of misclassified observations from the discriminant function.

MP discriminant analysis models have been tested extensively over the last 30 years in developing classifiers for the two–group classification problem. However, in the comparative studies that have included MP models for classifier development, the MP discriminant analysis models either lack appropriate normalisation constraints or they do not use the proper data transformation. In addition, these studies have generally been based on relatively small datasets. This thesis investigates the development of MP discriminant analysis models that incorporate appropriate normalisation constraints and data transformations. These MP models are tested on binary classification problems, with an emphasis on credit scoring problems, particularly application scoring, i.e. a two–group classification problem concerned with distinguishing between good and bad applicants for credit based on information from application forms and other relevant data. The performance of these MP models is compared with the performance of statistical techniques and machine learning methods and it is shown that MP discriminant analysis models can be useful tools for developing classifiers.

Another topic covered in this thesis is feature selection. In order to make classification models easier to understand, it is desirable to develop parsimonious classification models with a limited number of features. Features should ideally be selected based on their impact on classification accuracy. Although MP discriminant analysis models can be extended for feature selection based on classification accuracy, there are computational difficulties in applying these models to large datasets. A new MP heuristic for selecting features is suggested based on a feature selection MP discriminant analysis model in which maximisation of classification accuracy is the objective. The results of the heuristic are promising in comparison with other feature selection methods.

Classifiers should ideally be developed from datasets with approximately the same number of observations in each class, but in practice classifiers must often be developed from imbalanced datasets. New MP formulations are proposed to overcome the difficulties associated with generating discriminant functions from imbalanced datasets. These formulations are tested using datasets from financial institutions and the performance of the MP-generated classifiers is compared with classifiers generated by other methods. Finally, the ordinal classification problem is considered. MP methods for the ordinal classification problem are outlined and a new MP formulation is tested on a small dataset.

I declare that this thesis has been composed by myself with the work of my own and that this work has not been submitted for any other degree or professional qualification.

Konstantinos Falangis

Acknowledgments

In my long effort to bring this work to fruition, the contribution of a number of people has proved invaluable. First and foremost, I would like to express my deepest gratitude to Dr John Glen, not only for being an excellent supervisor, but also for providing his full practical and moral support throughout. His precious guidance and deep knowledge in the field of mathematical programming has greatly contributed to the successful completion of this thesis. Many thanks are also to Professor Jonathan Crook for sharing his knowledge and expertise in the field of credit scoring. Also, I would like to thank George A., William G., Ross G., and John O. for providing with me data for chapters 3, 4 and 5 respectively. Finally, I would like to thank my family and my friends for their moral support all these years. This work has been supported by the Credit Research Centre, Edinburgh.

Table of Contents

Chapter 1

1. Introduction

1.1 The classification problem

Patterns are considered to be the means by which the world can be interpreted. Based on this idea, people are able to read a book and recognise every character or image included in the pages. This ability is based on knowledge gained by experience in reading these same characters or seeing similar pictures. Using similar rules (or experience) people are able to discriminate between different colours, sizes, faces, etc. This concept initiated scientists to develop methods to solve other types of problems, such as discrimination between benign and malignant tumors (e.g. Mangasarian, 1965) the detection of fraudulent transactions, (e.g. Brause et al, 1999) and discrimination between bad and good payers, e.g. Thomas et al (2002). All these problems are set under the general label of classification. Specifically, in classification the aim is to assign observations into a number of pre-specified classes so that the objects in the same class are similar to one another (Gordon, 1981). After learning these patterns a model is used to classify new examples.

The process of classification from a model development aspect consists of several steps: data collection, data preprocessing, feature selection, classifier development, and assessment of the results. Data collection is very important because data quality affects the quality of the results. The GIGO (Garbage In Garbage Out) principle characterises classification problems because the final results depend on the data used as inputs to the process. So before using the data it is important to apply some preprocessing actions such as data transformation, sampling or feature selection. The latter action is used in making the classifier more flexible and possibly more accurate when applied to different data than the data used in the development. When assessing the results it becomes important to use the most appropriate criterion depending on the nature of the problem as some measurements are less accurate under some data conditions such as imbalanced class sizes. The whole process is iterative partially or overall, e.g. feature selection can be repeated several times untill the optimal subset is found, and also some of the steps can be missed.

There are many methods from statistics, machine learning and operational research that have been used for developing classifiers. Mathematical programming is one of the areas that have offered many tools in classification; however it has not been investigated properly, e.g. Baesens (2003). The main focus of this thesis is to investigate the use of the mathematical programming methods in constructing classifiers for binary and ordinal classification problems and also to propose solutions to inefficiencies in the mathematical programming approach. The ordinal classification problem is similar to the binary classification problem although the dependent variable is defined in an ordinal scale.

Classification is relevant to a large range of problems such as cell tissue analysis, e.g. Sun and Xiong (2003), heart disease, marketing, and diabetes, e.g. Adams and Hand (1999). An area that has received much attention during the last three decades is credit scoring. In credit scoring, lenders use data from previous borrowers in order to discriminate between customers that might go bad (miss a number of consecutive payments) and good (who will not). This approach is used for a range of different products such as credit cards, auto loans, personal loans, small business loans and mortgages. In this thesis, the performance of mathematical programming methods for classification problems will be investigated through the use of credit data from different sources and products.

1.2 The Credit Risk Assessment Problem

The credit environment has changed radically in recent years. The lender community has changed by the appearance of new players in the market such as super-market chains and peer-to-peer lending websites and the debtor community has changed after the credit crunch, as a result the circumstances of lending in general have changed. The process of granting credit also has changed with the adoption of new techniques that are more sophisticated and less subjective. These changes in combination with the increased competition, the drive for diversification and liquidity, and regulatory changes such as risk-based capital requirements, (Basel, 2006a) have stimulated the development of many innovative ways to manage credit risk in the financial environment (Basel, 2006b). In this category of innovative ways is included the adoption of a score-based approach helping lenders to quantify the risk related to lending to individuals or small-medium sized companies. This score-based approach, known as credit scoring, uses

methods from different fields such as statistics, operational research and machine learning and tries to build models able to predict the future behaviour of applicants. However, some methods have not had as much attention as others. Specifically, mathematical programming based models have not been examined as thoroughly as other methods such as logistic regression or neural networks. This thesis tries to cover this gap by studying the performance of mathematical programming models in credit scoring and related fields.

A first definition given for credit scoring can be found in Lewis (1992): "Credit scoring is a process whereby some information about a future or current customer is converted into numbers that are then combined to form a score". From the definition it can be seen that scoring is related to two types of decisions that firms who lend to consumers have to make. Firstly, the firms should decide whether or not to grant credit to a new applicant. Methods used for decisions of this type are known as application scoring. The second type of decision is how to deal with existing customers; decisions such as whether to increase the credit limit, or to make a new offer to an existing customer, are very common for the credit risk departments of a bank. Methods used for this kind of decisions are known as behavioural scoring (e.g. Thomas, 2000; Thomas et al, 2002). This thesis is concerned with the development of models for application scoring; but it will also look in fields strongly related to credit scoring such as fraud scoring, i.e. models that rank applicants according to the likelihood their application or transactions may be fraudulent.

The idea of credit scoring is to use data on past applicants to rank current applicants in order of likelihood of default. Any information that could improve the prediction of default should be considered such as data from credit bureaus, which were developed to pool data on the performance of individual consumers with different lenders and to check official documents to obtain further information on the applicant. Because past data are used to explain future behaviour, credit scoring is very sensitive to the data used to develop the models. One of the most important issues in developing credit scoring models is the selection of features used in the model.

This thesis is concerned with the development of credit scorecards using mathematical programming methods and addresses other related issues in scorecard development such as feature selection, calibration and the problem

of imbalanced datasets. These issues are important in the credit classification problem because in big portfolios minimising the number of variables used in a scorecard, or improving the predictive ability of the scorecard means important cost reductions in data storage, or fewer losses. Also these issues provide evidence for the usefulness of mathematical programming models in credit scoring. Apart from examining the usefulness of mathematical programming in application scoring, its use in related fields such as fraud scoring is also discussed.

1.3 Contributions

The main contributions and research questions of this thesis are listed below.

1.3.1 Using mathematical programming models in credit scoring

Mathematical programming (MP) has been used in many fields with great success, e.g. planning production, engineering design, portfolio management (Williams, 1999) but has received little attention from credit decision makers. Indeed even when MP-based methods have been tested, this was done either using small datasets, e.g. Ziari et al (1995) or using a simple linear programming classification model without providing specific details about the structure of the model, e.g. Baesens (2003). In Chapter 3 the performance of MP models is examined and a comparison with other commonly used methods, e.g. logistic regression, neural networks, classification trees is made. Six datasets (four publicly available and two datasets from financial institutions) are used to set up these experiments, each representing different sizes.

1.3.2 Mathematical programming-based feature selection heuristic

The fact that many organisations have created databases consisting of millions of gigabytes of data has made essential the need to identify relevant and irrelevant factors. It is essential to have tools that can help decision makers focus on the most relevant features for use in representing the data. In chapter 4 two MP-based heuristics for feature selection are presented and are tested in three credit scoring datasets (two publicly available and one from a financial institution).

1.3.3 The effect of imbalanced datasets on the performance of classifiers

A very common characteristic of datasets in credit scoring is that the bad class represents a small portion of the whole dataset, e.g. 8-10%. As a result the classifiers are heavily affected by the good class that represents the majority class. This is important considering the fact that bad cases cost more to the lenders, i.e. it is more expensive to accept a bad customer than reject a good customer. The use of imbalanced datasets when training a classifier also affects the performance measurements used to assess the classifiers, e.g. error rate is incapable of capturing the different misclassification costs. In Chapter 5 the use of mathematical programming under imbalanced datasets is investigated and MP-based solutions to overcome this problem are suggested. Four datasets (one from fraud scoring and three from credit retail portfolios) are used to compare these methods with other methods from statistics and machine learning.

1.3.4 Use of MP techniques in credit related fields

MP based methods can be used in binary or multi-class classification problems where the class is usually represented by a nominal variable. However, there is also another type of classification problems in which the target variable is represented on an ordinal scale. In problems, such as the ranking of road projects, e.g. Beuthe and Scannella (2001), or the ranking of venture capital projects, e.g. Siskos and Zopounidis (1985), a model for ranking the alternatives from most to the least preferable is required. This can be achieved using ordinal classification approaches. Ordinal classification can be also useful in credit scoring applications, e.g. scorecard calibration. In Chapter 6 the ordinal classification problem and its use in credit scoring are described. An improvement in an existing MP method for ordinal classification problem is proposed along with its implementation in the ranking of road projects (due to lack of appropriate credit data). Also the performance of an ordinal MP based model that produces nonlinear functions assuming monotonicity for the features included in the model is investigated. This model has been tested mainly with small or simulated datasets, e.g. Zopounidis and Doumpos (1999), so large datasets are used in order to test its performance.

1.4 Thesis Overview

This thesis is structured in the following way.

In Chapter 2 the basic statistical and machine learning methods used in classification problems, and more specifically in application scoring, are analysed along with their main strengths and deficiencies. MP classification methods are also described. The main features of the latter category and the benefits from using MP-based methods are also examined. A discussion of topics strongly related to the classification problem, such as feature selection and imbalanced datasets is also included at the end of this chapter.

In Chapter 3 the development of a credit scorecard is described in more detail, e.g. data transformation, sampling, performance assessment. Other related fields of credit scoring are also described such as profit scoring, attrition scoring, etc. At the end of this chapter the performance of methods considered in Chapter 2 are compared using six credit datasets.

In Chapter 4 the topic of feature selection is outlined and how it is related to classification problems and specifically to credit scoring. Two heuristics based on an MP approach are proposed. In order to test these heuristics three credit datasets are used and comparisons with other methods from the machine learning field are made.

In Chapter 5 the issue of imbalanced datasets is considered in relation to credit and fraud scoring. Difficulties involved in using the most popular techniques for developing classifiers, e.g. logistic regression, neural networks, on imbalanced datasets are discussed and different approaches for dealing with these problems are considered. MP classification methods for imbalanced datasets are also examined and ways to overcome the problems related to imbalanced classes are suggested. Experiments are performed on a fraud scoring dataset and three small-business credit datasets and suggestions based on the results are made.

In Chapter 6 the ordinal classification problem is discussed. The performance of a nonlinear MP based model is investigated using large datasets and an improvement of the model is suggested. Also, an existing MP based method for ordinal classification is discussed and a revised model for scorecard calibration is suggested.

Chapter 7 summarises the conclusions of this study and suggestions for further research are outlined.

Chapter 2

2. Literature Review

2.1 Introduction

Classification models are used to assign observations or objects of unknown group or class membership into one of a predetermined number of groups or classes based on the values of a set of features associated with each observation or object, e.g. Duda et al (2001). The features used in a classification model may be the natural variables associated with each observation, or features may be constructed from the natural variables. A wide range of methods from statistics, machine learning and mathematical programming can be used to develop classification models. Statistical discriminant analysis (Fisher, 1936) was the first formal method proposed for developing classification models, and other statistical techniques were developed later. Advances in computer technology stimulated the development of a number of machine learning methods, which although less formal than statistical techniques are increasingly used for classification model development (e.g. Hand, 1997). MP methods can also be used for classification model development, but MP methods are not as widely used in practice as statistical and machine learning methods. The most commonly used statistical and machine learning methods for classification model development are outlined in sections 2.2 and 2.3, respectively. MP methods for classification model development and the limitations of these methods are discussed in section 2.4. For simplicity, only the two-class, or binary, classification problem will be considered. Finally, areas for research in MP discriminant analysis models will be outlined.

2.2 Statistical methods

Classification models can be developed using a number of statistical techniques, particularly linear discriminant analysis, linear regression, logistic regression, *k*-nearest neighbours, classification trees and naïve Bayes.

2.2.1 Linear Discriminant Analysis

Fisher (1936) proposed linear discriminant analysis (LDA) as a method for classifying observations or objects into one of two mutually exclusive and exhaustive groups based on a linear function of a set of independent variables associated with each observation or object. The linear function of LDA is chosen to maximise a group separation metric. In calculating this linear function, the important variables should be identified and the function can then be used to allow new observations to be classified as belonging to one of the predetermined groups (e.g. Orgler, 1975).

Consider a two-group discriminant problem in which n features are associated with each observation, with $\boldsymbol{x}=(x_1,x_2,\ldots,x_n)$ representing the vector of feature values. The objective of LDA is to estimate $P(y|\boldsymbol{x})$, the probability of membership of group y, y=1,2, given feature vector $\boldsymbol{x}$. It is assumed that the covariance matrices for each group are equal, then if $\boldsymbol{S}$ is the estimated covariance matrix and $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are the estimated mean feature vectors for groups 1 and 2 respectively, the direction which best separates the two groups is given by the vector $\boldsymbol{w}$, where (e.g. Hand, 1997) $\boldsymbol{w} \propto \boldsymbol{S}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$. An observation with vector of feature values $\boldsymbol{x}$ is then classified by considering the function $\boldsymbol{w}'\boldsymbol{x}=c$, where c is a cutoff value, such that if $\boldsymbol{w}'\boldsymbol{x}<c$ the observation is classified as belonging to group 1 and if $\boldsymbol{w}'\boldsymbol{x}>c$ the observation is classified as belonging to group 2. In general, the cutoff value, c, will depend on the prior probabilities of group membership and the costs of misclassifying observations in each group (e.g. Hand, 1997). The assumption of equal covariance matrices can be relaxed and a quadratic discriminant function generated (e.g. Smith, 1947).

Discriminant analysis is an easy-to-use method for developing classification models, but it is sensitive to the data used and as the parameters for each group are unknown, it is necessary to estimate these using a sample of observations, e.g. Eisenbeis and Avery (1972), Eisenbeis (1977, 1978). LDA was used by Altman (1968) to predict bankruptcy using financial ratios and it has also been applied widely in credit scoring, e.g. Lane (1972), Apilado et al (1974), Eisenbeis and Avery (1972), Eisenbeis (1977, 1978), Reichert et al (1983). Hand (1997) notes that even if some assumptions are violated, e.g. categorical data, LDA performs relatively well in comparison with other methods, while Baesens et al (2003) demonstrated that LDA can outperform quadratic discriminant analysis in credit scoring.

2.2.2 Linear Regression

Linear regression is used to express a dependent variable as a linear function of a set of independent variables from a set of observations with known values for the dependent and independent variables, e.g. Neter et al (1996). The coefficients in the linear regression function are chosen to minimise the sum of the squared errors between the actual values of the dependent variable and the values predicted by the linear function. For example, if the estimated regression coefficient for variable j, j=1,2,….,n, is b_j and the estimated constant term in the regression function is b_0, then if X_{ij} represents the value of variable j in observation i, i=1,2,….,m, the predicted value, p_i, of the dependent variable for observation i is:

$$p_i = b_0 + b_1X_{i1} + \ldots. + b_nX_{in} \qquad i=1,2,\ldots,m.$$

Linear regression can be applied to binary classification problems such as credit scoring by defining the actual values of the dependent variables as categorical variables. For example, linear regression might be used in credit scoring to express the probability that an applicant for credit will not default based on a set of variables or features associated with applicants for credit. In this application, p_i may represent the predicted probability that applicant i, i=1,2,….,m, will not default, with the actual value for the probability of non-default being 1 if an applicant has not defaulted (i.e. a "good" applicant) and 0 if an applicant has defaulted (i.e. a "bad" applicant). An obvious weakness in using linear regression in binary classification problems is that it can produce predicted probabilities that are greater than 1 or less than zero. In addition, linear regression is based on the assumption that the dependent variable and the residuals are normally distributed, but the values of this variable cannot be distributed normally in binary classification problems (e.g. Pampel, 2000) as there are only two values for the dependent variable. In binary classification problems, linear regression produces models similar to those produced by discriminant analysis (e.g. Orgler, 1971).

Linear regression has been used in the construction of scorecards, i.e. credit scoring models, mainly because of its simplicity and the widespread availability of appropriate software (Thomas, 2000). For example, Orgler (1970) used linear regression to develop a model for evaluating commercial loans. However, linear regression will not be used in the model comparisons in this thesis because of its underlying assumptions.

2.2.3 Logistic Regression

Logistic regression was developed to address problems in linear regression, particularly the assumption that the dependent variable is continuous and unrestricted in value, i.e. can take any value in the range $-\infty$ to $+\infty$. In logistic regression for the binary classification problem, the independent variable is assumed to be linearly related not to the dependent variable, i.e. membership or non-membership of a specified class, (as in linear regression) but to the natural log of the odds of membership of the specified class. Consider the binary classification problem with two classes, denoted 0 and 1, and assume there are m observations of known class membership, where for observation i, i=1,2,….,m, independent variable j, j=1,2,….,n, has value X_{ij}. For observation i, i=1,2,….,m, let y_i, with value 0 or 1, denote its class membership and let p_i denote its predicted probability of membership of class 1, so that $p_i/(1-p_i)$ represents the predicted odds of membership class of 1. The logistic regression model is then:

$$\ln[p_i/(1-p_i)] = b_0 + b_1X_{i1} + b_2X_{i2} + \ldots. + b_nX_{in} \qquad i=1,2,\ldots,m$$

with p_i, i=1,2,….,m, given by

$$p_i = \exp(b_0 + b_1X_{i1} + \ldots. + b_nX_{in})/[1 + \exp(b_0 + b_1X_{i1} + \ldots. + b_nX_{in})].$$

where coefficients b_j, j=0,1,….,n, are estimated using an iterative procedure to maximise the likelihood function $\prod_{i=1}^{i=n} p_i^{y_i}(1-p_i)^{1-y_i}$ (e.g. Hand, 1997).

Unlike parametric methods, e.g. LDA, logistic regression does not require assumptions about the population. The main drawback of logistic regression is the model parameters must be estimated using an iterative maximum likelihood procedure that requires more computations than, for example, linear regression (Thomas, 2000), although this problem has been reduced by improvements in computing technology. In addition, as with linear regression, logistic regression is sensitive to correlated independent variables (Thomas, 2000). One of the strengths of logistic regression is that, as with discriminant analysis and linear regression, it allows the user to identify the features that are good predictors of the dependent variable. It is therefore possible to produce a parsimonious model with the same (or better) performance as the model containing all the possible features.

In a comparative study of logistic regression and discriminant analysis, Press and Wilson (1978) found that logistic regression outperformed linear discriminant analysis, although not by a large amount. In general, logistic

regression is a practical and easy-to-use method that can produce good results in building classification models. Even though the results of the first study which used logistic regression in credit scoring (Wiginton, 1980) were poor, the adoption of this technique by the credit industry has been extensive and it is now the method most widely used in credit scoring (e.g. Mays, 1998) with many published applications, e.g. Joanes (1993), Laitinen (1999), and Westgaard and Van der Wijst (2001).

2.2.4 Nearest Neighbour Methods

Nearest neighbour methods, such as the *k*-nearest neighbour (*k*-nn) method, are nonparametric methods of estimating the probability of class membership from a set of values of features associated with an observation or object. For example, in the *k*-nn method, the probability, $p(y|\boldsymbol{x})$, of membership of class y for an observation or object of unknown class with vector of feature values $\boldsymbol{x}$ may be given by the proportion of its k nearest neighbouring observations of known class membership that belong to class y. In the *k*-nn method, the parameter k, which defines the size, but not the shape, of a neighbourhood, and a separation metric for assessing proximity must be specified. Euclidean distance or more complex metrics in which different weights are attached to each dimension (e.g. Hand and Henley, 1997) may be used as the separation metric in *k*-nn methods.

The *k*-nn method is suitable for credit scoring (e.g. Hand and Henley, 1997) and is easy to apply. For example, using the credit scoring notation of Hand and Henley (1997), let $p(g|\boldsymbol{x})$ and $p(b|\boldsymbol{x})$ be the probability of good or bad risk respectively for an applicant with characteristic vector $\boldsymbol{x}$. For a new applicant for credit, let k_g and k_b, where $k_b=k-k_g$, denote the number of good and bad cases respectively in the k design-set cases of known good/bad status nearest to the new case, as determined by the separation metric. The estimates of $p(g|\boldsymbol{x})$ and $p(b|\boldsymbol{x})$ are then given by k_g/k and k_b/k respectively and the new cases is classified to class c where $k_c=\max[k_g,k_b]$. The *k*-nn can also be updated as the population of applicants changes and it is fairly easy to incorporate misclassification costs (e.g. Hand and Vinciotti, 2003). In addition, it is easy to provide reasons for refusing credit, which may be a legal requirement, as the neighbours can provide a case-based explanation (e.g. Hand and Henley, 1997). However, an appropriate separation metric must be specified and, in order to classify a new case, this metric must be

used to calculate the separation between the new case and all the cases in the design sample. The computational requirements, which will depend on the number of cases in the design sample, the number of features associated with each case and the form of the separation metric must therefore be taken into account in using *k*-nn methods (e.g. Guyon and Elisseeff, 2003).

The simplest version of *k*-nn method, in which each point is assigned to the class of its nearest neighbour of known class, i.e. k=1, was used by Fogarty and Ireson (1994) in a credit scoring application. Hand and Henley (1997) found that *k*-nn classifiers compared favourably with linear regression, logistic regression, and classification trees in credit scoring,

2.2.5 Naïve Bayes

The naïve Bayes classifier technique is based on Bayes' theorem and is particularly appropriate when the dimensionality of the feature space is high, e.g. Guo et al (2009). For a problem in which a vector $\boldsymbol{x}=(x_1,x_2,\ldots,x_n)$ of n features is associated with each observation, naïve Bayes learns the class-conditional probabilities $p(x_i|y_i)$ of each categorical variable i, $i=1,2,\ldots,n$, given the class label y_i. A new observation with feature vector $\boldsymbol{x}$ is classified by using the Bayes' rule to compute the posterior probability of each class y_i given the vector of attributes:

$$p(y \mid \mathbf{x}) = \frac{p(y_i)\, p(\mathbf{x} \mid y_i)}{p(\mathbf{x})}$$

The basic assumption of naïve Bayes' classifier is that the variables are conditionally independent given the class label, so that:

$$p(\mathbf{x} \mid y) = \prod_{i=1}^{n} p(x_i \mid y_i)$$

This assumption is considered to be an unrealistic assumption as features are generally related in practice. For example, in credit scoring characteristics such as income and age are often strongly related. However, it is easy to construct a classifier using naïve Bayes because there is no need to learn as the model is given *a priori*.

Baesens (2003) tested the performance of naïve Bayes using eight credit datasets and found that the performance of this method was very poor and that it did not compete with any of the other methods in any of the eight datasets used in a comparative study.

2.2.6 Classification Trees

A classification tree, or recursive partitioning, is a nonparametric classification approach in which observations are split into sets of similar class membership using appropriate tests or splitting rules. Classification trees can be represented by a tree diagram, such as the binary tree, i.e. a tree in which there are two branches at each node other than the terminal nodes, as appears in Figure 2.1. The non-terminal nodes, represented by circles, in a classification tree specify a test to split observations into different subsets and the branches at non-terminal nodes represent the outcomes associated with the test. The top node is the root of the tree and a class label is associated with each leaf or terminal node (denoted by a square). The splitting rules in a classification tree can be based on simple comparisons or metrics such as the Kolmogorov-Smirnov statistic (e.g. Thomas et al, 2002). The classification and regression tree (CART) proposed by Breiman (1984) is an example of a classification tree.

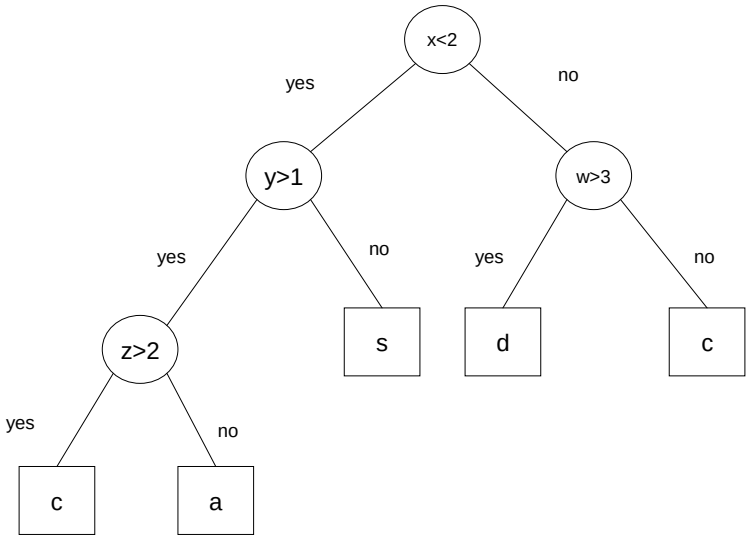

Figure 2.1: Example of Classification Tree (Squares represent possible outcomes and circles represent decision nodes)

In using a binary classification tree for credit scoring, a design sample of applicants of known default risk is first split into two subsets, where each subset is composed of applicants with more similar default risk than the

complete set of applicants. Each of these two subsets is then split into two using a different splitting rule to generate two more similar subsets in terms of default risk. This process of repeatedly splitting subsets of applicants into two is repeated until further subdivision does not yield more homogeneous subsets, i.e. a terminal node is generated. The tree can then be used to classify a new applicant, where for a new applicant with a specified vector of feature values, the predicted probability of low risk is given by the proportion of good applicants in the subset of the design sample at the terminal node associated with this vector of feature values.

Classification trees are very suitable for use in credit scoring because (i) the underlying decision process can be represented in a sequential way rather than simultaneously as is the case with other methods, e.g. linear discriminant analysis; (ii) it is easy to construct non-linear classifiers; and (iii) it is able to handle both categorical and nominal variables. However, classification trees can become very large and since most approaches use a fixed design or training set, tree redesign may be required as additional data become available (Safavian and Landgrebe, 1991). An additional disadvantage of classification trees is that continuous variables are implicitly discretized by the splitting process, with information lost in this process (Dreiseitl and Ohno-Machado, 2002). Classification trees have been found to perform reasonably well in a number of comparative credit scoring studies, e.g. Srinivasan and Kim (1987), Boyle et al (1992), and Baesens (2003).

2.3 Machine Learning

Machine learning methods have been used in many classification tasks, e.g. Piramuthu (1999b), Shaw and Gentry (1988), Wang et al (2005). Methods such as neural networks, support vector machines and expert systems are less restrictive than many statistical methods as they do not require assumptions about the data used to build a model. However, these methods use a “black-box” approach for classifier construction and since information on the steps followed in deriving the weights for each feature is not produced, it is generally not possible to provide an interpretation of the results.

2.3.1 Neural Networks

A neural network (NN) can be defined as a model of reasoning based on the human brain (Negnevitsky, 2002). A NN consists of a number of interconnected processors called neurons. A neuron receives input signals from its input links, computes an output signal and transmits this signal through its output links. An input signal can be raw data or the outputs from other neurons. The output signal can be either a final solution to the problem or an input to other neurons, Figure 2.2 represents a typical NN where the neurons, represented by circles, are connected by links, with each link having an associated weight that represents the importance of that link. A NN is set through repeated adjustments of these weights.

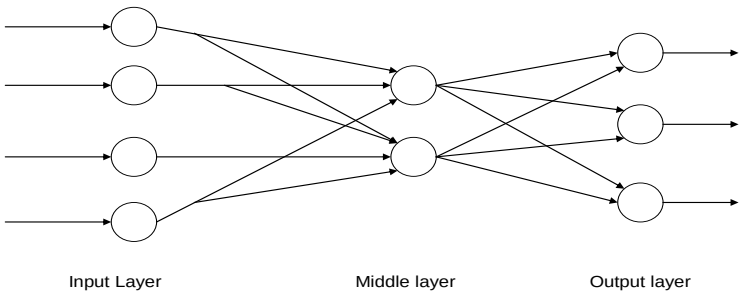

Figure 2.2: Architecture of a Typical Artificial Neural Network.

In order to build a NN the number of neurons, the method for connecting neurons and the learning algorithm must be specified. The weights of the network links must then be initialised using a training sample. The output at each neuron is determined by computing the output signal from the input signals to this neuron. For example, a sign function output is determined by calculating the weighted sum of the input signals and comparing the result with a threshold value, with output –1 if the weighted input is less than the threshold value and output +1 otherwise. Other types of functions can also be used, e.g. Negnevitsky, (2002).

Neural networks have good generalisation capabilities and it is possible to learn many different types of function in the middle layers of the network

(Piramuthu, 1999a), as a result it is not necessary to specify the relationships in the model. It is also easier for this kind of model to identify bad cases, e.g. West, 2000. On the other hand, NNs are considered to have poor performance when there are irrelevant features or applied to small datasets (John et al, 1994). The latter problem is very common when developing a scorecard for new products that do not have any previous applicants, so that initially it is necessary to work with a small number of observations.

Neural networks have been used in many business applications (e.g. Vellido et al, 1999). The main characteristics of NNs that may make them inappropriate for credit scoring are: (i) the inability to identify the relative importance of potential input variables; (ii) the long training process; (iii) difficulties in interpreting the results produced; and (iv) difficulties in deciding the number of hidden units and learning parameters, e.g. learning rate (Piramuthu, 1999a, 1999b), although efforts have been made to address these issues. For example, Chen and Huang (2003) examined a genetic algorithm-based approach to overcome the problem of interpretation, while Baesens (2003) investigated methods that can be used to extract rules in order to interpret the results from NNs. Nevertheless, NNs have been applied in credit scoring, e.g. Arminger et al (1997), Baesens et al (2003), Desai et al (1997), Glorfeld and Hardgrave (1996), Limsombunchai et al (2005), Piramuthu (1999a), Piramuthu et al (2004), and West (2000) and in most of these applications, NNs were reported to perform better than standard statistical techniques.

2.3.2 Support Vector Machines

Support vector machines (SVMs) were first proposed by Vapnik (1995) as learning systems for binary classification. SVMs are trained using an algorithm from optimisation theory and statistical learning theory to derive a separating hyperplane in a high dimensional feature space (Cristianini and Shawe-Taylor, 2000). Figure 2.3 represents a simple linearly separable two-dimensional example in which H is the separating hyperplane and H_1 and H_2 are the support vectors, or separating hyperplanes, parallel to H that are as far apart as possible, i.e. with maximal margin. SVMs are based on a non-linear mapping of the problem data into a higher dimension feature space (Cristianini and Shawe-Taylor, 2000). However, the learning algorithm may be inefficient and SVMs may be difficult to implement as a large number of

parameters is required. In addition, small training samples will result in overfitting, with poor generalisation ability (Cristianini and Shawe-Taylor, 2000). The original model proposed by Vapnik was a linear classifier, but other types were later proposed in order to improve the accuracy of the original model. The main difference of the new models compared to the initial model is the function used to map the data into a higher dimensional space. New functions were proposed, namely: polynomial, radial basis function (RBF) and sigmoid. All these functions transform the original data into a higher dimensional space and then the linear classifier is used subsequently.

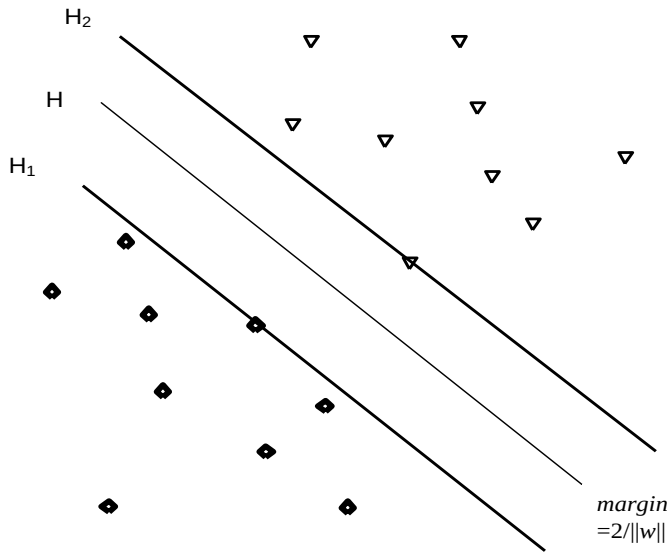

Figure 2.3: Maximum-margin Hyperplanes for a Two-class SVM (Solid points and open points represent observations of the different classes)

SVMs have been used in a number of credit scoring studies, e.g. Tian and Deng (2004), Li et al (2006). Huang et al (2007) reported that SVMs performed well in comparison with neural networks, genetic algorithms and classification trees in credit scoring..

2.3.3 Expert Systems

An expert system is a computer-based collection of processes, i.e. software, which mimics the decision making behavior of a human expert

(Thomas et al, 2002). The basic components of an expert system are (i) a knowledge base containing domain knowledge as a set of IF-THEN rules; (ii) a database for matching against the IF conditions in the knowledge base; (iii) an inference engine that connects the rules in the knowledge base with the database; and (iv) a user interface. In addition, an expert system may include an explanation facility to explain a conclusion and/or the need for specific information. Expert systems provide a clear separation of knowledge from the processing of this knowledge and as a result can deal with incomplete and uncertain data.

Expert system have been used in credit scoring, e.g. Shaw and Gentry (1988), Nikbakht and Tafti (1989), Srinivasan and Ruparel (1990), Pinson (1992), Fogarty and Ireson (1993), Michalopoulos et al (2001), Bryant (2001) and Metaxiotis and Psarras (2003). The main disadvantages of expert systems are (1) substantial computational effort may be required as all rules must be considered by the inference engine; (2) rule-based expert systems cannot learn from experience; (3) expert systems can be expensive to develop and maintain (Metaxiotis and Psarras, 2003); and (4) feature selection is based solely on expert knowledge.

2.3.4 Hybrid Methods

Hybrid methods that combine different techniques have also been proposed for classifier development. For example, De Andrés et al (2011) proposed a hybrid approach based on fuzzy clustering (Dunn, 1973) and multivariate adaptive regression splines (Friedman, 1991). This hybrid approach was found to perform better than linear discriminant analysis and neural networks on a Spanish bankruptcy dataset, although a limitation of this study is that only the five financial ratios proposed by Altman (1968) were used in the analysis. Finlay (2011) proposed a boosting method in which weak classifiers are combined with increased weight applied iteratively to borderline observations, and found that this hybrid approach outperformed linear discriminant analysis, logistic regression, a classification tree, a neural network and a *k*-nearest neighbour method on a credit scoring dataset. A hybrid method based on a neural network and fuzzy logic (Zadeh, 1965) was proposed by Akkoç (2012). This hybrid method was found to outperform linear discriminant analysis, logistic regression and a neural network on a balanced credit dataset for 2000 customers of a

Turkish bank. Because this hybrid method uses fuzzy logic, explanations of credit decisions can be obtained.

2.4 Mathematical Programming Methods

Statistical and machine learning methods have been applied to many classification problems, but they may not be appropriate or may not perform well in all applications. For example, some statistical methods involve restrictive assumptions, while machine learning methods may not be appropriate when interpretable results are required. Mathematical programming (MP) techniques provide an alternative to statistical and machine learning methods for developing classification models. Although applications of MP in classification first appeared in the late 1960s and early 1970s, e.g. Rosen (1965), Mangasarian (1965), Smith (1968) and Grinold (1972), wider interest in MP methods for classification model development was stimulated by the MP discriminant analysis models proposed by Freed and Glover (1981a, 1981b) and Hand (1981). These MP methods can be used to develop classification models for multi-group problems (e.g. Freed and Glover, 1981b; Gehrlein, 1986), but since most research in this area has focused on models for the two-group discriminant problem, the two-group problem will be used to illustrate these MP methods. For the two-group discriminant problem it will be assumed that the training sample contains m observations known to belong to either group 1 (G_1) or group 2 (G_2), with $G_1 \cap G_2 = \emptyset$, and that each observation consists of the values of n features with X_{ij} denoting the value of feature j, j=1,2,….,n, in observation i, i=1,2,….,m. MP methods for the two-group discriminant problem are concerned with generating a discriminant function that separates the observations in a training sample into the two groups so that, as far as possible, observations in group 1 and group 2 lie respectively below and above the discriminant function, which is defined by a constant term, a_0, and the coefficient, a_j, of feature j, j=1,2,....,n.

As the initial MP methods for classification model development were based on linear programming (LP) discriminant analysis models, these methods will be considered first. The use of integer programming (IP) and nonlinear programming methods in this area will then be discussed.

2.4.1 Linear Programming Based Methods

The simplest models for MP discriminant analysis are LP models in which the group separation measure is based on the deviations of misclassified observations from the discriminant function, with minimisation of the sum of deviations (MSD) being the most widely used objective, e.g. Freed and Glover (1981a). Let d_i, $d_i \geq 0$, represent the deviation from the discriminant function of observation i, i=1,2,....,m, where d_i=0 if the observation is correctly classified and d_i>0 if the observation is misclassified. Note that if the two groups are linearly separable, the minimum sum of deviations is zero and the discriminant function will be a separating hyperplane. The basic LP model for generating the MSD discriminant function is:

$$\text{Minimise} \qquad \sum_{i=1}^{m} d_i \tag{2.1a}$$

$$\text{subject to} \qquad \sum_{j=1}^{n} X_{ij} a_j - a_0 - d_i \leq 0 \qquad i \in G_1 \tag{2.1b}$$

$$\sum_{j=1}^{n} X_{ij} a_j - a_0 + d_i \geq 0 \qquad i \in G_2 \tag{2.1c}$$

a_j unrestricted, j=0,1,2,....,n; $d_i \geq 0$, i=1,2,....,m.

A major weakness of the basic MSD formulation (2.1) is that an obvious solution is the trivial solution with a_j=0, j=0,1,....,n. LP model (2.1) must therefore be normalised to prevent such trivial solutions. A simple and widely used normalisation method is to set the constant term, a_0, in the discriminant function to a non-zero value (e.g. Freed and Glover, 1981a). However, this standard normalisation does not permit the generation of discriminant function that pass through the origin and the coefficients a_j, j=1,2,....,n, are not invariant under origin shift in the problem data (Markowski and Markowski, 1985). Freed and Glover (1986b) demonstrated that by constraining coefficients a_j, j=1,2,....,n, to sum to a constant, only the constant term, a_0, in the discriminant function is dependent on the choice of origin. However, the MSD model with this normalisation constraint should be solved with positive and negative normalisation constants (Glover et al, 1988) and this model does not permit solutions in which the coefficients a_j, j=1,2,....,n, sum to zero (Koehler, 1991). A non-linear normalisation constraint was proposed by Cavalier et al (1989), but the heuristic method used to solve this extended model may not yield the optimal solution. Glen (1999) demonstrated that the weaknesses in these normalisation methods can be addressed by representing the free variables a_j, j=1,2,....,n, by a pair of

non-negative variables, a_j^+ and a_j^-, with

$$a_j = a_j^+ - a_j^- \tag{2.2}$$

and constraining the absolute values of the a_j, j=1,2,....,n, to sum to a constant, e.g.

$$\sum_{j=1}^{n}(a_j^+ + a_j^-) = 1. \tag{2.3}$$

where it is required that for each variable j, j=1,2,....,n, at most one of a_j^+ and a_j^- may be non-zero. By defining two non-negative variables, a_j^+ and a_j^-, for each variable a_j, j=1,2,....,n, Glen (1999) represented this requirement by a set of four constraints for each variable a_j, resulting in a mixed integer programming (MIP) model for the MSD problem. As each variable pair, a_j^+ and a_j^-, j=1,2,....,n, forms a special ordered set of type 1 (SOS1), there may be computational benefits (e.g. Williams, 1993) from using the SOS1 representation if an SOS1 resource is provided in the MP solver software (Glen, 2006).

A further weakness of the basic LP model (2.1) is that observations lying on the discriminant function are regarded as correctly classified, irrespective of their group membership. This problem can be overcome by introducing a small rejection interval, with all observations lying in this rejection interval regarded as misclassified (e.g. Glen 2001).

Freed and Glover (1981a) also proposed an alternative LP model for generating discriminant functions in which the objective is maximisation of the minimum deviation (MMD). Let d denote the minimum deviation of observations from the discriminant function, where $d \geq 0$ if all observations are correctly classified and $d<0$ if at least one observation is misclassified. The basic LP model for the two-group MMD problem is then:

$$\text{Maximise} \qquad d \tag{2.4a}$$

$$\text{subject to} \qquad \sum_{j=1}^{n} X_{ij}a_j - a_0 + d \leq 0 \qquad i \in G_1 \tag{2.4b}$$

$$\sum_{j=1}^{n} X_{ij}a_j - a_0 - d \geq 0 \qquad i \in G_2 \tag{2.4c}$$

$$d,\ a_j \text{ unrestricted},\ j=0,1,2,....,n$$

As with the MSD model (2.1), the MMD model (2.4) must be normalised to prevent trivial solutions, e.g. by normalising for invariance under origin shift by representing the free variables a_j, j=1,2,....,n, by a pair of non-negative variables, a_j^+ and a_j^-, defined by (2.2) and adding constraint (2.3), with additional constraints to ensure that at most one of a_j^+

and $a_{\bar{j}}$ may be non-zero. In addition, it may be necessary to define a_j, j=1,2,....,n, as bounded variables, $-U \leq a_j \leq U$, $U>0$, j=1,2,....,n, to ensure bounded solutions (e.g. Freed and Glover, 1986a). An undesirable feature of the MMD model is that it is particularly sensitive to outliers in the training sample (e.g. Bajgier and Hill, 1982). In an experimental study, Freed and Glover (1986a) found that the MMD model did not perform as well as the MSD model and statistical discriminant analysis, due mainly to the impact of outliers.

Goal programming (e.g. Charnes and Cooper, 1977) extensions of the MSD and MMD models have also been proposed (e.g. Freed and Glover, 1981b) to allow multiple goals to be considered in LP models for generating linear discriminant functions. The different goals are included in the objective function with weights assigned to each goal, e.g. to reflect costs associated with each goal, but as the results depend on the weight of each goal, care is required in assigning weights to individual goals. In some goal programming applications, it may be possible to assign weights with the help of expert knowledge but, in general, it may be necessary to try different sets of weights to find the most appropriate weightings. The difficulty in assigning weights is the main drawback of goal programming methods. Freed and Glover (1981b) presented goal programming based LP discriminant analysis models with two goals, one related to the minimisation of deviations of misclassified observations, i.e. a measure of group overlap, and the other related to maximisation of deviations of correctly classified observations, i.e. a measure of group separation. Similar goal programming models were also proposed by Glover et al (1988), Glover (1990) and Lam et al (1993). For example, using symbols defined for MSD model (2.1) and for observation i, i= 1,2,….,m, let e_i, denote the deviation of correctly classified observations from the discriminant function and let H_i and K_i denote the weight in the objective function associated with correct and incorrect classification respectively of observation i, one of the goal programming formulations proposed by Freed and Glover (1981b) is:

Maximise $\sum H_i e_i - \sum K_i d_i$ (2.5a)

subject to $\sum_{j=1}^{n} X_{ij} a_j - a_0 - d_i + e_i \leq 0 \qquad i \in G_1$ (2.5b)

$\sum_{j=1}^{n} X_{ij} a_j + a_0 + d_i - e_i \geq 0 \qquad i \in G_2$ (2.5c)

e_i , $d_i \geq 0$, i=1,2,....,m; a_j unrestricted, j=0,1,2,....,n

The model above uses in its objective functions two different goals: the minimisation of group overlap and the maximisation of the interior deviations, i.e. the deviations of correctly classified observations, with different weights for each of these goals. These weights are varied according to which of these goals is emphasized. In a similar way are defined the other models defined by Freed and Glover (1981b). Freed and Glover (1986a) examined the performance of goal programming discriminant analysis models, but although the results obtained were promising, these goal programming models did not outperform the simple MSD model.

Retzlaff-Roberts (1996) proposed a model based on the goal programming model of Glover et al (1988), but with the ratio of the weighted sum of internal deviations, i.e. deviations of correctly classified observation, to the weighted sum of external deviations, i.e. deviations of misclassified observations, or vice versa, as the objective. As this objective is non-linear, the problem is linearised in a manner similar to data envelopment analysis (e.g. Charnes et al, 1977) by introducing a constraint in which the numerator of this ratio is set to a constant and using minimisation of the denominator as the objective function. This constraint also has a normalisation role, but as with similar normalisation constraints proposed by Glover et al (1988), this normalisation may generate discriminant functions solutions that are clearly non-optimal.

In practice, it is often desirable to develop a classification model with a relatively small number of features, e.g. to reduce the cost of data collection and storage and to make the model easier to understand. Parsimonious models, i.e. models with a limited number of features, may also have better classification performance than models that include all the original variables. A number of variable selection techniques, such as stepwise methods (e.g. Huberty1994), can be used with statistical discriminant analysis methods, and one of the main criticisms of early LP discriminant

analysis models concerned the lack of a methodology for feature, or variable, selection (e.g. Glorfeld and Gaither, 1982). Nath and Jones (1988) proposed a jackknife procedure for feature selection in LP discriminant analysis models, but as this approach involves running the LP model with one observation from the training sample excluded in turn, the computational effort may be unacceptable in practice. Glen (1999) demonstrated that MSD model (2.1) normalised for invariance under origin shift, i.e. by representing each free variable a_j, j=1,2,....,n, by non-negative variables, a_j^+ and a_j^-, as in (2.2), and adding constraint (2.3), can be extended to select a specified number, p, $1 \le p \le n$, of features and generate the MSD discriminant function in these p features. For feature selection, define a binary variable, γ_j, for each feature j, j=1,2,….,n, where γ_j=1 if and only if feature j is selected. The conditions associated with this definition of γ_j can be represented by constraints:

$$a_j^+ + a_j^- - \varepsilon\gamma_j \ge 0 \quad j=1,2,\ldots,n \tag{2.6a}$$

$$a_j^+ + a_j^- - \gamma_j \le 0 \quad j=1,2,\ldots,n \tag{2.6b}$$

where ε is small and positive. The requirement to select p features can then be modelled by the constraint:

$$\sum_{j=1}^{n} \gamma_j = p\,. \tag{2.7}$$

The MSD feature selection is then:

$$\text{Minimise} \quad \sum_{i=1}^{m} d_i \tag{2.8a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} X_{ij}(a_j^+ - a_j^-) - a_0 - d_i \le 0 \quad i \in G_1 \tag{2.8b}$$

$$\sum_{j=1}^{n} X_{ij}(a_j^+ - a_j^-) - a_0 + d_i \ge 0 \quad i \in G_2 \tag{2.8c}$$

$$\sum_{j=1}^{n} (a_j^+ + a_j^-) = 1 \tag{2.8d}$$

$$a_j^+ + a_j^- - \varepsilon\gamma_j \ge 0 \quad j=1,2,\ldots,n \tag{2.8e}$$

$$a_j^+ + a_j^- - \gamma_j \le 0 \quad j=1,2,\ldots,n \tag{2.8f}$$

$$\sum_{j=1}^{n} \gamma_j = p \tag{2.8g}$$

α_0 unrestricted; $a_j^+, a_j^- \ge 0$, j=1,2,....,n; $d_i \ge 0$, i=1,2,....,m; γ_j=0,1, j=1,2,….,n, where a_j^+, a_j^- form an SOS1, j=1,2,....,n. MSD feature selection model (2.8) can be used to develop parsimonious classification models.

2.4.2 Integer Programming

The MSD and MMD models, or any other LP model that tries to optimise a deviation-based metric, attempt indirectly to minimise the number of misclassified cases or the total misclassification cost. Instead of trying to minimise a metric of this type it is possible to use a MIP formulation to directly minimise the total number of misclassified observations or maximise classification accuracy, i.e. the number of correctly classified observations. Stam (1990) considers this criterion very important, as the ultimate goal of classification is to minimise the total predicted number of misclassified cases. For the problem of maximising classification accuracy in a two-group discriminant problem with m observations, define a binary variable β_i, i=1,2,….,m, for each observation such that β_i=1 if observation i is correctly classified and β_i=0 otherwise. Then defining other symbols as before, the basic form of the MIP model for determining the classification accuracy maximizing discriminant function is:

$$\text{Maximise} \qquad \sum_{i=1}^{m} \beta_i \tag{2.9a}$$

$$\text{subject to} \qquad \sum_{j=1}^{n} X_{ij} a_j - a_0 + M\beta_i \le M \quad i \in G_1 \tag{2.9b}$$

$$\sum_{j=1}^{n} X_{ij} a_j - a_0 - M\beta_i \ge -M \quad i \in G_2 \tag{2.9c}$$

a_j unrestricted, j=0,1,2,....,n; β_i=0,1, i=1,2,....,m.

where M, M>0, is large.

As with the MSD model (2.1), the MIP model (2.9) for maximising classification accuracy (MCA) must be normalised to prevent trivial solutions, e.g. by normalising for invariance under origin shift by representing the free variables a_j, j=1,2,....,n, by a pair of non-negative variables, a_j^+ and a_j^-, defined by (2.2) and adding constraint (2.3), with additional constraints to ensure that at most one of a_j^+ and a_j^- may be non-zero. As observations lying on the discriminant function generated by MIP model (2.9) are regarded as correctly classified irrespective of their group membership, a small rejection interval can be introduced so that all observations lying in this rejection interval are regarded as misclassified (e.g. Glen, 2001). By normalising for invariance under origin shift, the MIP model (2.9) can be extended for feature selection in a manner similar to the MSD feature selection model (2.8). The advantage of the MIP feature

selection model is that features are selected on the basis of their contribution to classification accuracy.

The need for a binary variable for each observation creates computational problems when trying to apply IP in large datasets because of the problems of handling large number of binary variables in the branch and bound algorithm of integer programming, e.g. Williams (1999). As a result, integer programming models have not been examined extensively in classification problems in which large datasets are used, as in credit scoring. Stam (1997) argues that it is almost impossible to use standard MP software to use MIP models for maximising classification accuracy or minimising misclassifications in problems with training samples of more than 100 observations. For this reason, these MIP models have been tested on small real and simulated datasets (e.g. Koehler and Erenguc, 1990; Stam and Joachimsthaler, 1990). Although improvements in computing technology and algorithmic developments mean that commercial software can be applied to larger problems, these MIP models can, in practice, still only be applied to relatively small discriminant problems. The MIP approach, however, provides a benchmark for evaluating the training sample performance of other linear classifiers (Stam and Joachimsthaler, 1990), although MIP models may not, depending on the nature of the datasets, perform as well as other methods on holdout samples (e.g. Koehler and Erenguc, 1990; Stam and Joachimsthaler, 1990).

Liittschwager and Wang (1978) were among the first to suggest an MIP formulation for the binary classification problem. In this formulation, a binary variable is defined for each observation and the costs of misclassifying an observation in each class must be specified. The MIP model for determining the discriminant function that minimises the expected total misclassification cost is normalised by introducing two binary variables for each feature and adding constraints to ensure that at least one feature has a discriminant function coefficient of ±1. An algorithm for solving this MIP model was suggested, but this algorithm was only applied to small simulated discriminant problems. In practice, it can also be difficult to assess the misclassification costs for both classes (e.g. Adams and Hand, 1999). Extensions of this model, with different heuristic solution procedures, have been proposed (e.g. Banks and Abad, 1994), but there are similar difficulties with these models.

Bajgier and Hill (1982) proposed a mixed integer goal programming formulation for the two-group discriminant problem in which the primary goal was concerned with minimising the number of misclassifications and the secondary goal was concerned with minimising the sum of exterior deviations and maximisation of interior deviations, with different weights assigned to each component of the objective function. Choo and Wedley (1985) suggested an MIP model for the multi-group discriminant problem, but although an application to a two-group discriminant problem is described, it appears that only an LP simplification of the MIP model, i.e. essentially an MSD model, was used in this application. In discussing their results, Choo and Wedley (1985) note that one of the advantages of MP discriminant analysis methods over statistical discriminant analysis is that constraints can be imposed on the discriminant function coefficients.

Gehrlein (1986) proposed an MIP model to maximise classification accuracy in multi-group discriminant problems by generating either a single linear discriminant function with group dependent cutoffs or a separate function for each group boundary. The multi-function model requires many more constraints than the single function model, but both these approaches can only be applied to relatively small problems as a binary variable is required for each observation. An alternative multi-group MIP model proposed by Wilson (1996) requires more variables and constraints than the multi-group MIP model of Gehrlein (1986), but although a hierarchical approach can be used to determine each function separately, the model can only be applied to relatively small problems.

A general multicriteria MIP formulation of the two-group discriminant problem given by Stam (1990) uses the number of misclassifications and the sum of deviations of incorrectly classified observations as criteria. Solutions to this multi-objective model are derived for a number of iteratively generated criteria weights and the decision maker then chooses a preferred solution from a set of selected solutions, but the process involved in selecting these different solutions is subjective. The use of secondary goals in MIP discriminant analysis models creates problems and it is essential to design the models with great care (Stam, 1997). The classification performance of four MIP models with secondary goals was examined by Pavur et al (1997) using simulated datasets, with the results indicating that the choice of secondary goal can have a significant impact on holdout sample classification performance

Koehler (1991) suggested a genetic algorithm for generating discriminant functions that minimise both the number of misclassifications and the number of features with non-zero coefficients in the discriminant function. This approach produced good results on simulated datasets, but there is no guarantee that genetic algorithms, or other heuristic methods, will yield the optimal solution. In addition, as simulated datasets cannot generally represent the distinctive problem characteristics that are normally associated with problems in which there are benefits from developing parsimonious classification models, it would be better to test the approach on real datasets.

The main difficulty limiting the application of MIP discriminant analysis models in practice is that these models can be only be applied to relatively small training samples because a binary variable must be associated with each training sample observation. Because of the computational difficulties in solving large MIP models, heuristic procedures have been suggested for solving variants of the MIP discriminant analysis model (e.g. Koehler and Erenguc, 1990; Abad and Banks, 1993; Banks and Abad, 1994; Rubin, 1997), but most of these heuristics (e.g. Koehler and Erenguc, 1990; Abad and Banks, 1993; Banks and Abad, 1994) have been tested on relatively small problems with at most 100 observations in the training samples. Although the decomposition based heuristic proposed by Rubin (1997) was applied to a training sample with up to 683 observations, the classification performance of the classifiers generated by this heuristic was not examined.

Stam and Ragsdale (1992) proposed a two-stage method for minimising the number of misclassified observations. In the first stage a simple LP model, similar to the MSD model but with a unit classification gap, is used to generate a discriminant function. The function generated in the first stage is then used to identify observations that are correctly classified and observations that are misclassified or lie in the classification gap. In the second stage, an MIP model is used to minimise misclassifications in observations that the first-stage function identified as misclassified or in the classification gap, subject to additional constraints that ensure that observations that were correctly classified by the first-stage function remain correctly classified. The second-stage MIP model involves fewer binary variables than the standard, i.e. single-stage, MIP model as it is not necessary to define binary variables for observations that were correctly

classified by the first-stage function. Consequently, this two-stage approach can be applied to larger training samples than the standard MIP model. However, Stam and Ragsdale (1992) note that the training sample classification performance, i.e. the apparent hit rate, of the function generated by the second-stage MIP model may be worse than the apparent hit rate produced by the standard MIP model due to the additional constraints in the second-stage model. Indeed, in a comparison of standard and two-stage methods on real and simulated datasets, Glen (2006) found that the standard MIP model outperformed the two-stage method of Stam and Ragsdale (1992).

The two-stage method proposed by Stam and Ragsdale (1992) can use an MSD model in the second stage to generate a classifier. Sueyoshi (1999) proposed a similar two-stage approach in which MSD-type models were used in the first and second stages. In the first stage, two linear functions with non-negative coefficients that sum to one are generated such that only correctly classified observations lie above/below these functions. A single function with non-negative variable coefficients that sum to one is generated in the second stage of this approach. Sueyoshi (1999) argued that this approach had similarities with data envelopment analysis (DEA), which was first proposed by Charnes et al (1977), and named this two-stage approach "DEA-discriminant analysis", but this terminology is inappropriate as this discriminant analysis technique is not based on DEA. A modified version of this two-stage approach was later proposed (Sueyoshi, 2001), in which only one function is generated at each stage and the requirement that these functions have non-negative variable coefficients was removed. Sueyoshi (2006) later proposed a similar two-stage approach in which a MIP model was used in the second stage to determine functions that minimise misclassifications. The first stage of this two-stage approach is used to identify an overlap between the two groups of observations and to generate a first-stage separation function. A function which minimises the number of misclassified observations in the overlap is then generated in the second stage. The major weakness of this approach is that observations that are correctly classified at the first stage are not considered in the second stage. As a consequence, some observations that were correctly classified in the first stage may not be correctly classified by the second-stage function. A two stage approach is also involved when the results from this two-stage approach are used to classify observations. Observations are first classified

into group 1, group 2 or the overlap, and then observations in the overlap are classified into group 1 or 2. Two attempts are therefore made at classifying some observations, biasing the apparent hit rates produced by this two-stage approach. These limitations were not recognised by Tsai et al (2009) who reported that this two-stage approach outperformed linear discriminant analysis, logistic regression and neural networks in predicting default and non-default using account and survey data for 281 customers of a Taiwanese bank.

Both internal and external deviations were considered in two-stage MP approaches proposed by Lam et al (1996). In the first stage, an LP model is used to determine the feature weights that minimise the sum of the interior and exterior deviations from the group means of each feature. These weights are then used to calculate a score for each observation. In the second stage, either an LP model is used to determine the cut-off value that minimises the sum of deviations of the scores of misclassified observations from this cut-off value, or an MIP model is used to determine the cut-off value that minimises the number of misclassified observations. However, as a binary variable is required for each observation, the second-stage MIP model can only be applied to relatively small problems. These two-stage approaches were applied to a small credit scoring data set, but the scoring functions generated were unstable. In practice, there would be difficulties in using a classification model based on either of these approaches. In order to classify a new observation, it would be necessary to re-calculate the deviations of its feature values from the group means in order to calculate its score. In addition, for binary features, i.e. features that can take only two values such as 0 and 1, the practical significance of the deviation from the mean value is unclear.

Gehrlein and Wagner (1997) proposed a two-stage cost-based MIP approach for credit scoring, with minimisation of misclassification costs used as the objective function in the MIP model for the first stage. The second-stage MIP model then takes account of the cost of obtaining additional information on applicants who are classified as not worthy of credit at the first stage. The use of this approach was demonstrated on a small problem using different sets of costs. It was argued that this two-stage approach can lead to a significant reduction in total costs, but it is clear that the approach cannot be applied to large datasets.

An iterative method for generating classification accuracy maximising discriminant functions in problems with many more observations than can be handled by the standard MIP model was developed by Glen (2003). In this iterative procedure, an MSD model for the complete set of observations is first used to generate a discriminant function. A neighbourhood of correctly and incorrectly classified observations, defined by interior and external deviations respectively, is then constructed about this discriminant function. An MIP model is then used to generate a discriminant function that maximises classification accuracy in this neighbourhood of observations, subject to constraints that ensure that correctly classified observations outside this neighbourhood remain correctly classified. A new neighbourhood of correctly and incorrectly classified observations is then constructed about this new discriminant function and the MIP model is again used to generate a discriminant function that maximises classification accuracy in this new neighbourhood, subject to constraints that ensure that correctly classified observations outside this new neighbourhood remain correctly classified. This process is repeated until there is no improvement in the total classification accuracy between successive iterations. This iterative procedure, which can be extended for feature selection, was applied to a credit scoring dataset with 690 observations with promising results. However, this method is computationally intensive as it is necessary to set up and run the MIP model several times.

An iterative dual-based heuristic for minimising misclassifications was proposed by Sarkar (2005). In the first stage the dual problem of an IP model is solved and outliers identified. Outliers are then deleted in subsequent iterations. This approach was applied to credit scoring problems with large numbers of observations and the classification performance compared with logistic regression, with good results. Sarkar (2005) suggests that an advantage of this approach is that the non-zero weights that are produced for some features can be used to indicate the most important features to include in the classifier. Sundbom (2007) examined the performance of this heuristic and concluded that even though the approach produced competitive results compared with logistic regression, it is time consuming and rather inflexible. In particular, a solution has to be found before terminating and only then is it possible to make any adjustment to the model. As a number of iterations must be performed, this heuristic can be very time consuming.

2.4.3 Nonlinear Programming Methods

The objective functions of the MSD, MMD and MIP discriminant analysis models are linear. A more general l_p-norm can be adopted as the criterion in discriminant analysis, where the l_p-*norm* generally gives rise to a nonlinear optimisation problem. Consider a discriminant problem with m observations. Defining d_i, $d_i>0$, $i=1,2,\ldots,m$, as before, i.e. the deviation of misclassified observations from the discriminant function then, with other symbols defined as before, a general form of the l_p-norm discriminant problem is:

$$\text{Minimise} \quad \left(\sum_{i=1}^{m} d_i^{\,p} \right)^{1/p} \tag{2.10a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} X_{ij} a_j - a_0 - d_i \leq 0 \qquad i \in G_1 \tag{2.10b}$$

$$\sum_{j=1}^{n} X_{ij} a_j - a_0 + d_i \geq 0 \qquad i \in G_2 \tag{2.10c}$$

a_j unrestricted, $j=0,1,2,....,n$; $d_i \geq 0$, $i=1,2,....,m$.

where parameter p takes a value in the range zero to infinity. As with other MP discriminant analysis models, the l_p-norm model (2.10) must be normalised to prevent trivial solutions.

Clearly, the l_p-norm model (2.10) produces a linear discriminant function. Models with $p=1$, i.e. MSD, and $p=\infty$, i.e. MMD, can be solved using linear programming, whereas models with $p=0$ models can be solved using integer programming. For all other values of p, nonlinear programming techniques must be used for problem solution. Although only external deviations, i.e. the deviations of misclassified observations, are considered in l_p-norm models such as (2.10), l_p-norm methods can be extended to include internal deviations. Stam (1997) notes that more weight is given to outlying observations as the value of p is increased. An advantage of l_p-norm methods with $p<2$, is that robust classifiers can be generated even if there are outliers in the training sample (Stam, 1997).

Stam and Joachimsthaler (1989) examined the classification performance of the l_1, $l_{1.5}$, l_2, l_5 and l_∞ models, where the l_p-norm model was solved using nonlinear programming software, and found that the $l_{1.5}$ and l_2 objectives produced small improvements in holdout sample classification performance in comparison with the l_1 and l_∞ objectives. However, these results were obtained using small simulated datasets. Stam and Joachimsthaler (1989) suggest that the best l_p-norm should be determined by

evaluating different l_p-norms, with analysis restricted to $1 \leq p \leq 3$ and $p=\infty$, but in practice it would be computationally expensive to determine the most effective l_p-norm on large datasets.

Gallagher et al (1997) developed an integer programming model for maximizing the number of correctly classified observations subject to nonlinear constraints that approximate restrictions on the misclassification probabilities proposed by Anderson (1969). This model incorporates a classification gap for observations that are difficult to classify. However, some of the constraints associated with the binary variables were not included in order to reduce the size of the model, and as a result the classification accuracy may be miscalculated. Two linearisations of this nonlinear integer programming model were considered and tested on small datasets, but although the results seem promising there are computational difficulties in the solution procedures and the approach can only be applied to small datasets.

2.4.4 Nonlinear Functions

Nonlinear classifiers may produce better classification performance than linear classifiers (e.g. Glen, 2005). Stam and Ragsdale (1990) proposed a two-phase procedure for obtaining nonlinear classifiers for binary classification problems. In the first phase, an l_p-norm model is used to generate a linear discriminant function for different values of p. In the second phase, the parameters of a nonlinear transformation, such as a modified hyperbolic tangent transformation, of each of these linear functions are estimated from the linear function's fitted values by using a maximum likelihood method. This approach was tested on two small datasets and found to produce results similar to logistic regression. However, this two-phase procedure involves considerable computational effort as different l_p-norms must be evaluated.

Duarte Silva and Stam (1994) proposed an MP approach for generating nonlinear discriminant functions by introducing quadratic and cross-products of the original variables as features in MP discriminant analysis models. In practice, however, only a limited number of transformations of the original variables can be included in the analysis. Banks and Abad (1994) proposed a similar method for generating nonlinear discriminant functions, but as with the approach proposed by Duarte Silva and Stam (1994), the increased number of features results in increased computational

time and overfitting the data (Rubin, 1994). In an experimental study of a number of MP discriminant analysis models that incorporated quadratic and cross-product transformations of the original variables, Wanarat and Pavur (1996) concluded that models with second-order terms tend to overfit the data and can produce worse results than the simple LP models. Moreover, although models that incorporate quadratic and cross-product transformations of the original variables offer great flexibility, these models do not always produce robust results. Loucopoulos and Pavur (1997) compared the performance of two three-group MIP discriminant analysis models with second-order terms and concluded there is a need for more research to evaluate the performance of these models.

Classifiers that are nonlinear functions of the original variables can also be produced by creating categorical features from the original variables. For example, binary features can be generated by defining a threshold level, with the binary feature assigned value 1 if the value of the original variable exceeds the threshold level and 0 otherwise. Glen (2003) proposed an MP method for forming binary features and generating a linear discriminant function in these features, but information is lost in forming these binary features and the method requires additional computational effort.

Piecewise-linear functions can approximate nonlinear functions, and MP models can be used to generate piecewise-linear discriminant functions, resulting in nonlinear classifiers. Glen (2005) developed two MP methods for generating piecewise-linear discriminant functions. The first method uses MCA as the objective, while the second uses an approach based on MSD. The latter is more difficult to formulate because of the difficulty in calculating the deviation of some observations from the piecewise-linear function. The main disadvantage of these formulations is that more constraints and more binary variables and special ordered sets are required compared to the standard MCA and MSD models.

A modified version of the multicriteria additive utility ranking method, UTA (utilité additive), of Jacquet-Lagrèze and Siskos (1982) can also be used to generate nonlinear discriminant functions composed of a piecewise-linearisation of each feature's marginal utility function. In the UTA method, an LP model is used to generate an additive piecewise-linear utility function from a weak-order preference ranking of a training sample of observations. By modifying this LP model to deal with each observation's group membership, rather than its ranking, an additive piecewise-linear

discriminant function can be generated. A weakness of UTA-based methods for generating discriminant functions (e.g. Jacquet-Lagrèze and Siskos, 1982; Jacquet-Lagrèze, 1995; Doumpos et al, 2001) is that, as in the UTA method, each feature's marginal utility is assumed to be monotone non-decreasing, where features with monotone non-increasing marginal utility functions must first be transformed to monotone non-decreasing form. In practice, as noted by Glen (2008), for some features it may not be clear if the marginal utility function is monotone non-decreasing or non-increasing. Glen (2008) proposed an MIP model for generating additive piecewise-linear utility functions where it is not necessary to specify in advance the form of the marginal utility function of each feature and demonstrated that this MIP model could be extended for feature selection. Although test results indicated that this additive utility MIP model may be useful for developing nonlinear discriminant functions, these results were obtained on small datasets.

2.4.5 Discussion

MP discriminant analysis methods have advantages over other methods for developing classification models. For example, Glover et al (1988) argue that the main advantages of MP methods over traditional statistical techniques are:

1. MP methods are free from parametric assumptions of some statistical methods, e.g. normal populations, equal covariances matrices.
2. MP methods can consider classification accuracy directly and can be extended to deal with more complex problem formulations, e.g. different misclassification costs for each group, and to incorporate constraints, e.g. non-negative coefficients.
3. LP methods, such as the MSD model, are less sensitive to outliers because they are based on linear metrics rather than squared metrics.
4. Different weights can be assigned to different observations or groups of observations.

In spite of these advantages, there have also been criticisms of MP discriminant analysis methods, particularly following the paper by Freed and Glover (1981a) that stimulated much of the recent research in this area. For example, Glorfeld and Gaither (1982) considered the LP formulation as simple, unrealistic and lacking facilities, such as variable selection, available

with statistical discriminant analysis. Markowski and Markowski (1985) noted that neither the LP model of Freed and Glover (1981a) nor the MIP model of Bajgier and Hill (1982) was invariant under origin shift, although it was later shown that this problem can be addressed by use of an appropriate normalisation method (e.g. Freed and Glover, 1986b; Glen, 1999).

Several studies have compared the performance of MP discriminant analysis models with other techniques. For example, Bajgier and Hill (1982) compared statistical and MP approaches to the discriminant problem. Bajgier and Hill (1982) found that the MP approaches were more effective than LDA under certain conditions, such as when there is high overlap between groups and the variance-covariance matrices are very unequal, but these conclusions are based on analysis of small datasets. Mahmood and Lawrence (1987) compared the performance of quadratic and linear statistical discriminant analysis, rank discriminant analysis and logistic regression to MMD using data for 190 bankrupt and 42 non-bankrupt financial institutions. Although Mahmood and Lawrence (1987) found that the performance of the non-parametric methods other than MMD was similar to the performance of parametric methods, the only MP model considered, i.e. the MMD model, has been found to have poor performance in other studies e.g. Freed and Glover (1986a), Erenguc and Koehler (1990), partly because its results are outlier dependent. Markowski (1990) compared the performance of LDA and the MSD model on small two-group discriminant problems and reported that not only did LDA produce better holdout sample classification performance than the MSD model, but LDA achieved better balance in the classification performance in each group.

The performance of the linear and quadratic statistical discriminant analysis and MP discriminant analysis methods was compared by Stam and Jones (1990) on two-group problems under different experimental conditions. They found that quadratic discriminant analysis tended to give best results on both training and holdout samples when the group variances are different, while the MIP formulation performed best on both training and holdout samples when the group variances are equal, although the results were dependent on the size of the small training samples used in the study. Lam and Moy (1997) examined the classification performance of five LP discriminant analysis models and LDA under two simulated data conditions with (i) different number of observations in each group and (ii) outliers. The results indicated that the LP methods and LDA tend to be biased towards the

larger group, but the LP models outperformed LDA on the outlier contaminated datasets. However, the training samples in these studies were small, consisting of only 100 observations.

Baesens et al (2003) compared LDA, logistic regression, *k*-nearest neighbours, neural networks, classification trees, support vector machines and the MSD model on eight credit scoring datasets, the largest of which contained 11,700 observations, with performance measured by classification accuracy and area under the receiver operating characteristic (ROC) curve. It was found that a single method did not outperform the other methods on all datasets. Although the MSD model performed relatively well in these comparative studies, it should be noted that a version of the MSD model with a simple normalisation was used. Glen (2006) compared the performance of the MSD, MCA and two two-stage MP discriminant analysis methods (Stam and Ragsdale, 1992; Sueyoshi, 2001) on one real dataset and simulated datasets under six simulated data conditions. The two-stage methods generally did not perform as well as the other methods, and, as with Baesens et al (2005), it was found that one method did not outperform the others under all data conditions. These results suggest that in practice a number of methods should be considered in developing classification models, with the most suitable approach adopted for a specific problem (Glen, 2006).

2.5 Research Issues in MP Discriminant Analysis Methods

There has been considerable research in MP discriminant analysis models, but these models are not as widely used as other methods, particularly statistical methods, for developing classification models in spite of the benefits of MP-based approaches for classification model development. The relatively limited use of MP-based methods is partly due to poor communication with developers of classification models, with Stam (1997) arguing that there is a particular need for improved interaction with researchers in related statistical methods. This problem can be addressed by demonstrating the use of MP discriminant analysis methods in developing classification models in specific problem domains using relevant datasets. Credit scoring is an area of significant practical and theoretical interest (e.g. Thomas et al, 2002), which is generally characterised by large datasets. This

thesis will focus on the use of MP methods in developing credit scoring models.

Stam (1997) also argued that there was a need for further research to address some of the problems associated with MP methods, with variable selection identified as an area in which further research is required. There has been progress in developing a methodology for variable selection through extensions of the MSD and MCA models (Glen, 1999). Ideally features should be selected based on their impact on classification accuracy, i.e. by extending the MCA model, but since the MIP MCA model requires a binary variable for each training sample observation, the variable selection MCA model can only be applied to relatively small problems. This thesis will develop heuristic variable selection methods based on the MCA model to allow MCA-based variable selection methods to be applied to problems with a large number of observations. The use of these heuristic methods will be demonstrated on credit scoring datasets.

MP discriminant analysis methods have generally been demonstrated on relatively small real or simulated datasets. One of the features of many classification problems is that the datasets are imbalanced, with one group providing most of the observations. For example, in credit scoring (e.g. Thomas et al, 2002), where applicants are classified as good (i.e. unlikely to default) or bad (i.e. likely to default) using data from application forms, less than 10% of cases are typically classified as bad (e.g. Vinciotti and Hand, 2003). The degree of class imbalance in the data can be even greater in applications such as the identification of fraudulent credit-card transactions, where fraudulent cases typically comprise less than 0.2% of total transactions (e.g. Brause et al, 1999). If an imbalanced training sample is used to develop a classification model, it is likely that the analysis will be strongly influenced by the class with most observations. For example, in using a discriminant analysis technique to develop a credit scoring model, if the training sample has 1% bad cases, then any discriminant function that classifies all the cases as good will have 99% classification accuracy on the training sample. Ideally the training sample of observations of known class membership should contain approximately the same number of observations in each class, but in practice it may be difficult to obtain a balanced training sample of observations because of the imbalanced nature of the available data. Different approaches have been proposed for dealing with imbalanced datasets, e.g. over-sampling from the minority class, under-sampling from

the majority class or use of different misclassification costs for each class (e.g. Japkowicz and Stephen (2002). However, there are difficulties associated with pre-processing the data, e.g. over-sampling the minority class may lead to overfitting while under-sampling the majority class may discard important data. In addition, it can be difficult to determine misclassification costs in practice (e.g. Adams and Hand, 1999). The class imbalance must also be taken into account in evaluating the performance of a classifier. As additional constraints can easily be incorporated in MP models, MP discriminant analysis methods offer alternative approaches for dealing with imbalanced datasets. This thesis will examine MP methods for addressing the problems associated with imbalanced datasets and demonstrate the use of these methods in credit scoring applications.

Most MP discriminant analysis methods generate linear discriminant functions. Although there is no guarantee that nonlinear functions will necessarily form the basis of better classifiers (Stam, 1997), nonlinear classifiers may outperform linear classifiers (e.g. Glen, 2005). Discriminant functions that are nonlinear functions of the original variables can be developed by first transforming these variables, e.g. to form quadratic and cross-product features (Duarte Silva and Stam, 1994), but only a limited number of transformations of the original variables can be considered in practice. Additive utility based methods (Jacquet-Lagrèze and Siskos, 1982) can be used to generate nonlinear discriminant functions formed from piecewise-linear approximations of each feature's marginal utility functions, where each marginal utility function is assumed to be monotone non-decreasing. This weakness of additive utility based discriminant analysis methods can be overcome by using an MIP model, rather than an LP model, to generate these nonlinear discriminant functions (Glen, 2008), with the additional benefit that this MIP additive utility method can be extended for feature selection. As with the original LP-based methods (e.g. Jacquet-Lagrèze, 1995; Zopounidis and Doumpos, 1999), the MIP additive utility method has, however, only been applied to relatively small datasets. This thesis will examine the performance of the MIP additive utility model on large credit scoring datasets. The wider potential for ranking based methods in developing classification models will also be investigated.

Chapter 3

3. Credit Scoring

3.1 Introduction

Credit is the promise to pay within some limited time in the future a sum of money after services or products have been provided. All financial intermediaries who provide credit in any form face the risk of losing the capital they lent and the interest they were expecting. In order to quantify this risk, i.e. credit risk, credit providers have developed systems known as credit scoring, or scorecards, which are used to predict a borrower's future repayment performance using all the available data.

It is essential for companies that operate as lenders to have methods that can help predict if an applicant for credit will return the money in full or not, i.e. a "good" or "bad" applicant respectively. Specifically, "good" is usually defined as a borrower that keeps making payments to the lender and repays the loan, whereas "bad" is defined as a customer that misses a number of consecutive payments, e.g. misses three consecutive monthly payments, i.e. is more than 90 days past due. There are other reasons for adopting a good credit scoring system, including fierce competition and regulatory changes (e.g. Basel, 2006b). These factors work as an incentive for making banks adopt new techniques that are more sophisticated, more efficient and have better predictive accuracy.

Credit scoring appeared in the late 1960s, largely through the efforts of a small company, named Fair and Isaac, (e.g. Poon, 2007) through the introduction of application scorecards, i.e. models that combine information provided in the application form and credit bureaus, i.e. organisations that collect credit related information on individual consumers from a number of sources, into a single score. These methods have been used mostly for assessing the credit risk in portfolios of products for individuals such as mortgages, credit cards and auto loans. However, similar scoring techniques can also be used for assessing the credit risk in portfolios of small business loans, e.g. Bensic et al (2005). The development of scorecards for SME lending has not attracted as much attention as the development of scorecards for consumer lending, but under the new Basel II regulations (e.g. Basel,

2006b), portfolios of small business loans should be treated in the same way as portfolios of products to individuals. Small business credit scoring is similar to application scoring, with the score based on information on the business, its owner and other relevant data, e.g. Bensic et al (2005). The adoption of new methods in assessing the risks in this type of lending has been also stimulated by the importance of SME loans for economic growth, and the competition among banks and other financial institutions in traditional markets.

Scoring techniques that make use of other types of information have also been developed. For example, behavioural scoring (e.g. Thomas, 2000) is used to make decisions related to offers, e.g. to increase or decrease the credit limit, based on information about the behaviour of existing customers, e.g. repayment history, minimum balance, maximum balance, utilisation of overdraft. The methods used in behavioural scoring are similar to those used in application scoring, but with data updated on a continuing basis in order to keep track of the applicant's status. Attrition scoring, is used to predict the probability a customer will start a new relationship with a competitor, and follows the same methodology as behavioural scoring because the same types of variables are used, with customer profiles updated dynamically. Response scoring is used to predict the likelihood of response of a customer to a new offer, like a new credit card or a personal loan. Collection scoring provides tools in choosing the appropriate strategy for accounts that have been bad, e.g. to determine which accounts should be kept, which should be written off and which should be allocated to a collection agency (e.g. Thomas et al, 2002). These models are used in a later part of the credit cycle compared to the application and behavioural scorecards. Another category of tools is concerned with the detection of fraudulent or illegal transactions of an account based on past information for the account (e.g. Brause et al, 1999; Viaene et al, 2002; Bolton and Hand, 2002). This category of tools is known as fraud scoring.

This chapter will focus on credit scoring, but although this terminology has been used in the context of bankruptcy prediction (e.g. Caouette et al, 1998), only consumer and small business credit scoring will be considered. Procedures for developing scorecards will be outlined in section 3.2. Studies that have examined the use of different methods for developing scorecards are reviewed in section 3.3. An experimental study that compares different methods from statistics, machine learning and mathematical programming

on four credit card datasets and one small business loan dataset is described in section 3.4. The results from this benchmarking study are reported in section 3.5 and conclusions are summarised in section 3.6.

3.2 Constructing a Scorecard

There are many published studies which describe the use of credit scoring, e.g. Hand and Henley (1997), Hand and Jacka (1998), Thomas (2000), Thomas et al (2002) and theses e.g. Henley (1996), Andreeva (2004), Baesens (2003). In consumer credit scoring, the main sources of information are the application form and credit bureaus, which were developed to overcome the problem of asymmetrical information, i.e. borrowers know much more than lenders about their own ability and willingness to repay (Stiglitz and Weiss, 1981).

Traditionally credit decisions were judgmental and based on the 5 Cs (e.g. Lewis, 1992), i.e. character (the willingness to repay debts), capacity (the financial ability to repay debts), capital (funds from which payment can be made), collateral (assets from which payments might be made) and conditions (including the general economic environment and special conditions related to the borrower or the type of credit). This traditional approach had major problems and deficiencies, e.g. errors by staff administering the system, inconsistency in application of credit policies, the cost of training and employing staff and the cost of purchasing credit reports, that led to the development of automated scorecards (e.g. Capon, 1982).

Scorecards are generally built using statistical methods such as discriminant analysis (e.g. Lane, 1972) and logistic regression (e.g. Wiginton, 1980), although machine learning techniques such as neural networks (e.g. West, 2000) and mathematical programming (e.g. Srinivasan, 1976) have also been used. In developing a scorecard, it is essential to have a dataset with sufficient numbers of goods and bads (e.g. Lewis, 1992). In order to avoid overfitting and to allow the development of a parsimonious classifier, it is generally desirable to use only a subset of the initial set of features (e.g. Guyon and Elisseeff, 2003). Before developing the scorecard, it is also necessary to transform the data as it is essential to keep consistency and avoid outliers. After the scorecard has been developed, its performance must be evaluated.

3.2.1 Data

In order to build a credit scorecard it is necessary to have a dataset of applicants where the performance of each applicant is known. For each applicant there must be a flag indicating whether that specific account is "good" or "bad". For example, customers who miss three or more consecutive payments may be considered as "bad". In addition to this performance indicator for each applicant there is also a set of information that has been obtained from the application form, e.g. age, annual income, marital status, or a credit bureau, e.g. outstanding loans.

3.2.2 Feature Selection

The initial number of features included in the population is generally larger than the number of features that will be included in the final scorecard. Problems such as overfitting or poor predicting performance can occur if many features are associated with each applicant. Many methods have been proposed for overcoming this problem in statistical approaches, e.g. stepwise procedures (e.g. Neter et al, 1996) and machine learning techniques, e.g. wrappers and filters (e.g. Kohavi and John, 1997). Methods for feature selection in mathematical programming discriminant analysis models will be considered in Chapter 4, but in the benchmarking study of this chapter it is assumed that the variables included in the datasets used represent the optimal set of predictors and no feature selection method is used.

3.2.3 Data Transformation

The datasets used to develop scorecards usually include both numerical and categorical data, e.g. the occupation of an applicant for credit may be represented by a variable with different numerical values for different occupation categories. Categorical data of this type must generally be transformed for scorecard development. Although numerical data can be used directly in developing a scorecard, it may be desirable to transform some numerical data to produce a robust scorecard. For example, if there is a wide range of values associated with a feature, e.g. income, the scorecard may be very sensitive to outliers if the same weight is attached to all values within this range (e.g. Thomas, 2000). Data may also be transformed to

accommodate non-linear relationships, e.g. between an applicant's age and the likelihood of default.

Coarse classification based on weights of evidence (WoE) is widely used in credit scoring to construct features from both categorical and continuous variables in the original dataset (e.g. Thomas et al, 2002). For example, assume that for an original categorical variable k there are L possible categories, where category l, l=1,2,….,L, has g_{kl} goods (i.e. non-defaulting customers) and b_{kl} bads (i.e. defaulting customers) with G and B representing the total number of goods and bads respectively. For category l, l=1,2,….,L, of original variable k, the weight of evidence W_{kl} is given by

$$W_{kl} = \log(g_{kl}B/b_{kl}G).$$

Coarse classification is then used to construct binary features by combining categories with similar weights of evidence, where the similarity of weights may be assessed subjectively. Continuous features can also be coarse classified by first partitioning the range of values for a continuous feature into mutually exclusive categories, e.g. deciles, and then combining categories with similar weights of evidence to produce binary features. Coarse classification increases the number of binary features to be considered, but it is widely used in credit scoring (e.g. Somol et al, 2005).

3.2.4 Performance Measurement

Several measures have been proposed for assessing the performance of scorecards. Among the most popular are accuracy measures and separability measures. Both types of measures should be assessed on observations of known class membership that were not used to develop the scorecard, i.e. performance should be assessed on holdout samples that were not used in developing the classifier. Neither of these measures takes account of the consequences of misclassification. For example, in credit granting decisions, it is generally more costly to give credit to a customer who later defaults than not to give credit to a potential customer who would not have defaulted. In practice, however, it is often difficult to determine misclassification costs (e.g. Adams and Hand, 1999).

3.2.4.1 Accuracy Measures

Overall accuracy is defined as the ratio of the number of correctly classified cases to the number of the total cases. The main weakness of

overall accuracy as a performance measure is that it does not take account of the relative sizes of the classes. Thus, for example, if 5% of cases are bad, then a scorecard that assigns all cases to the good class will have 95% overall accuracy, but will fail to take account of the implications of misclassifying bad applicants. This weakness can be addressed, at least partially, by considering the accuracy in each class. In scorecard development, the accuracy in the bad class is the proportion of correctly classified bad cases in the total number of bad cases, while the accuracy in the good class is the proportion of correctly classified good cases in the total number of good cases.

For a general two-class classification problem, with classes defined as positive (e.g. bad) or negative (e.g. good), and a sample of n cases, assume that the results, in terms of the number of cases, from a classification model are summarized as in the confusion matrix of Table 3.1, where $n=a+b+c+d$:

Predicted Class	True Class	
	Positive	Negative
Positive	*a*	*b*
Negative	*c*	*d*

Table 3.1: Two-by-Two Confusion Matrix

The following terms can be then defined:

True positive rate = $a/(a+c)$

True negative rate = $d/(b+d)$

False positive rate = $b/(b+d)$

False negative rate = $c/(a+c)$

Note that sometimes (e.g. Hand, 1997) the true positive rate is termed "sensitivity" and true negative rate is termed "specificity", so that the false positive rate is 1–specificity.

3.2.4.2 Separability Measures

The receiver operating characteristic (ROC) curve (e.g., Bradley, 1997) can be used to provide information about the predictive accuracy of the model over its entire range of possible threshold values for a specific time period (Sobehart et al, 2000). In the ROC curve, the true positive rate (i.e. sensitivity) is plotted against the false positive rate (i.e. 1–specificity), with

points on the curve given by the results for different threshold values (e.g. Figure 3.1). The ROC curve provides an indication of the average classification performance of a classifier, but a threshold value must be specified in order to make classification decisions. If the classifier could perfectly classify, then the ROC curve would connect the points (0,0) and (0,1) and then the points (0,1) and (1,1). A random classifier is represented by a line at 45 degrees. In practice, ROC curves lie between the perfect and random ROC curves, as shown in Figure 3.1.

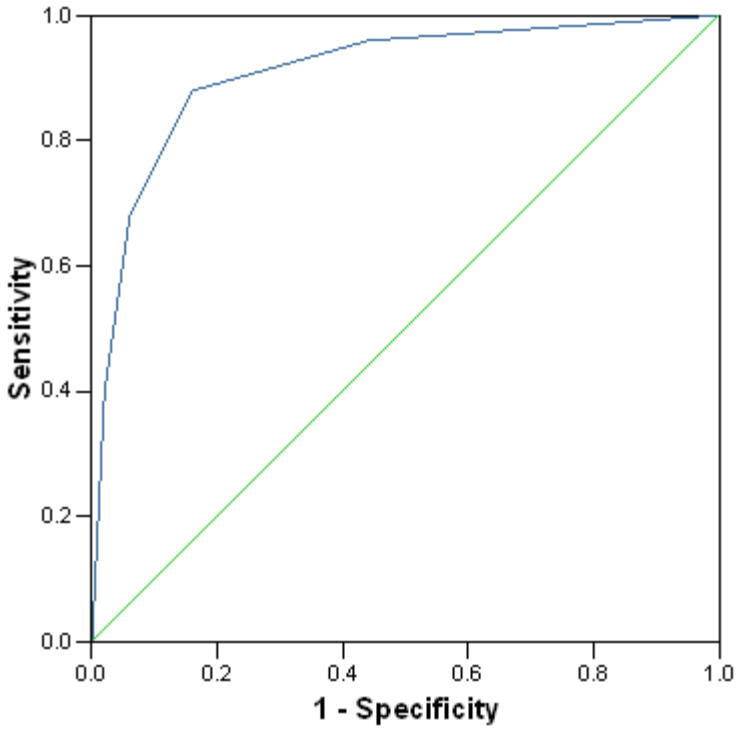

Figure 3.3: The ROC Curve. The diagonal line represents the performance of the random classifier.

A measure commonly used for comparing the performance of classifiers is the area under the ROC curve (AUC). AUC represents the probability that a randomly selected positive case will be classified as positive. Bradley (1997) investigated the use of the AUC as a performance measure for machine learning algorithms on six real world datasets and concluded that there was a good agreement between accuracy and AUC rankings of the classification algorithms.

3.3 Consumer and Small Business Credit Scoring

Many studies have examined issues associated with credit scoring for both individual consumers and small businesses. Some of the studies involving consumer credit will be considered first and this will be followed by discussion of a range of studies that focus on small business credit scoring.

3.3.1 Consumer Credit Scoring

Many different techniques have been applied to the problems of granting credit to individual consumers. In one of the first studies of consumer credit granting, Chatterjee and Barcun (1970) used personal loan data from a New York bank and proposed that each applicant for credit should be classified as belonging to the class with which there was most in common, i.e. similar to nearest neighbour rule. The objective in classifying applicants was to minimise the expected loss from the misclassification, with the jackknife method used to estimate the classification error rate. Unfortunately, this study did not include any comparisons with other methods.

In one of the first published works which used logistic regression in credit decisions, Wiginton (1981) applied logistic regression to credit scoring using data from a major oil company and compared the results obtained by discriminant analysis and a chance classifier. Only three characteristics were used in this study and as the chance classifier outperformed logistic regression and discriminant analysis, it was concluded that neither of these two methods is appropriate for making classification decisions on this dataset. These results may be due to the small number of the predictors used (normally a scorecard includes at least eight to fifteen features, e.g. Mays 2004) and technical issues, e.g. the effect of changing the cutoff value used in discriminant analysis was not investigated.

Desai et al (1997) used datasets from three credit unions in the Southeastern United States for the period 1988 - 1991, to compare the performance of logistic regression, linear discriminant analysis and two types of neural networks (modular neural networks and multilayer perceptron neural networks). These datasets, each consisting of less than 1,000 observations with 18 variables per observation, were also used to compare customised credit scoring models, in which a separate model was developed for each credit union, and generic credit scoring models, in which

the same models were developed for all three credit unions. It was found that the customised neural networks performed well in classifying the bad loans, and that the generic models did not perform as well as the customised models, especially for the bad loans.

Henley and Hand (1996) proposed an adjusted Euclidean distance metric for a *k*-nearest neighbour (*k*-nn) credit scoring method. The performance of this *k*-nn method was compared with logistic regression, linear regression, projection-pursuit regression and classification trees on a dataset of 19,186 mail-order credit customers, where each observation consisted of 16 categorical variables. Using bad rate amongst the accepted cases given a fixed acceptance rate as the performance measure, it was found that the *k*-nn method outperformed the other methods. However the differences are small and there is no statistical test to confirm if this is significant.

West (2000) compared the performance of five different types of neural networks (multilayer perceptron, mixture of experts, radial basis function, learning vector quantization, and fuzzy adaptive resonance) with linear discriminant analysis, logistic regression, classification trees, and *k-nn* in credit scoring. The comparisons were carried out on the German and the Australian credit datasets from the UCI repository (e.g. Frank and Asuncion, 2010) using a 10-fold cross-validation setup. The results indicated that the radial basis function and the mixture of experts performed better than all the other methods, but methods for selecting features were not discussed.

Yobas et al (2000) compared the predictive performance of linear discriminant analysis, neural networks, genetic algorithms and classification trees on a data set of 1,001 credit card payers composed of 14 variables, with classification performance assessed using the leave-one-out and 10-fold cross-validation methods. It was found that linear discriminant analysis outperformed the other methods, but it was noted that the results were affected by the way experiments were conducted, particularly the method of data transformation and the degree of class imbalance.

Baesens (2003) compared the performance of neural networks, linear discriminant analysis, quadratic discriminant analysis, naïve Bayes, logistic regression, linear programming, support vector machines, decision trees and *k-nn*. There were limitations in this study because it used only the simplest linear programming model with basic normalisation, which affected the model's performance. The results indicated that both SVM and neural

networks performed relatively well, and that linear programming did not perform very well overall.

Each of these consumer credit scoring studies suggested a different method for the construction of the most accurate classifier, although the performance differences were not always significant, and this supports the flat maximum effect described in Lovie and Lovie (1986).

3.3.2 Small Business Credit Scoring

Linear discriminant analysis was used by Altman (1968) to produce scores, called *Z*-scores, for predicting bankruptcy of firms based on financial ratios. Although this study can be considered as the start of research in the application of statistical methods for default prediction, the analysis was based on a small sample with 33 bankrupt and 33 non-bankrupt firms. The results indicated that discriminant analysis could be used for bankruptcy prediction, but that more research was required, e.g. because of the small sample.

One of the first to attempt to build scorecards for small business loans was Edmister (1972) who only focused on the selection of the financial ratios that could be useful in predicting SME failure. In this study, multivariate discriminant analysis was used to develop a model to predict small business defaults based on 19 financial ratios from a sample of small and medium sized enterprises over the period 1954-1969. It was reported that working capital/total assets and net operating income/sales ratios were predictive. However, the number of the cases included in the test was very small and also there were several restrictions applied in order to extract the data, making the final sample biased.

Srinivasan and Kim (1987) compared four classification models (discriminant analysis, logistic regression, goal programming and the recursive partitioning algorithm (RPA)), and a judgmental model (the analytic hierarchy process) using data for commercial loans, with error rates estimated by the bootstrap method. The results indicated that RPA gave slightly better results than the other methods. Boyle et al (1992) set up a study similar to Srinivasan and Kim (1987) using consumer credit data and they found that hybrid methods gave better results. However, the datasets used in the experiments were small and there was no data transformation similar to the concept of weight of evidence.

Leonard (1992) was one of the first to test the predictive ability of logistic regression and discriminant analysis models in SME lending. Using commercial loan applications from a major Canadian bank, it was shown that the benefits that have been obtained in consumer credit are available in commercial lending. In this study, loans to firms with assets of less than ten million dollars and for loan values of one million dollars or less were considered, with 283 applications in total, where 204 were approved and 79 rejected. This sample was quite small in comparison with consumer lending. Another feature of the dataset is that the applications were pooled by different areas, with different population bases, types of industry and levels of competition. The analysis incorporated 20 different variables, none of which included personal details of the owners, so that only "hard" data, e.g. financial ratios, balance sheet data, were used. The results indicated that the application of credit scoring techniques to small business credit appeared promising.

A credit scoring system for use as a decision support system in small business loan departments was proposed by Tsaih et al (2004). The main effort in this study was directed at providing a system that would ease the complexity involved in developing a credit scoring system that takes account of information asymmetry and time required to maintain the system. In order to achieve this, a mechanism was developed to update the economic environment and information relevant to the firm and its owner. The proposed credit scoring model was based on the probit model (e.g. Grablowsky and Talley, 1981) and tested using a dataset consisting of 41,000 small firms, with 6,000 defaulting and 35,000 non-defaulting firms. This model also incorporated a number of features related to the owner and the financial results of the firm. The method performed well, achieving 80% accuracy in defaulting firms, but unfortunately there is not much information about the methodology adopted in this study.

Bensic et al (2005) compared the performance of neural networks, logistic regression and classification trees using a dataset of small business loans. Although neural networks were found to perform well in this study, it should be noted that a very small dataset of credit applicants was used (only 160 applicants) and that bad cases were defined as 46 days past due (dpd) rather than 90 dpd as proposed in Basel II (e.g. Basel, 2006b). Also, there was no information about the transformation of the data.

In an expansion of the work of Edmister (1972), Altman and Sabato (2006) developed statistical models for assessing the creditworthiness of SMEs. Logistic regression was used to develop a distress prediction model from a large number of relevant financial ratios. This distress prediction model was found to have a holdout sample prediction power almost 30% better than that of the generic corporate Z-score model of Altman (1969). It was concluded that improving the accuracy of a credit risk model is likely to have beneficial effects on the Basel II capital requirements for SMEs and as such could result in lower borrowing costs for SMEs. It was also argued that banks should develop different credit risk models for SMEs and large corporations.

Lin et al (2007a) explored how different definitions of default and different transformations of data affect the accuracy of models for small business credit scoring. It was concluded that coarse classification, i.e. data transformation in which the raw data are replaced by binary variables, improves the accuracy of the classifier, although it does not seem to matter if WoE or binary variables are used to transform the data. It was also found that the accuracy of the scorecard is affected by the definition of default. In an extension of this study, Lin et al (2007b) compared Merton based models (e.g. Merton, 1974), and retail credit scoring models. The results indicated that retail credit scoring models had better performance when the sample had more bad cases, although the Merton models had better performance when there were higher acceptance rates in the samples.

3.3.3 Discussion

A number of different methods have been used for consumer and small business credit scoring. In published studies of consumer and small business credit scoring, different techniques have been found to produce better results in specific applications, and there is no evidence that one technique will consistently outperform other methods. It is therefore important to ensure that a number of techniques are considered in developing credit scoring systems for both consumer and small business lending. There has, however, been relatively limited use of MP methods in developing scorecards, particularly for lending to small businesses, and in comparative studies in which MP methods have been included, the MP models used are generally very simple, e.g. with basic normalisation constraints. In addition, many of

the comparative studies do not adopt techniques that are commonly adopted in practice, e.g. transformation of categorical data using WoE. For this reason a benchmarking study was performed to compare the performance of a number of techniques, including a relatively simple MP model, on a range of credit scoring problems.

3.4 Benchmarking Study

In this benchmarking study, five two-class credit datasets of different sizes and from different sources were used to compare the performance of credit scoring models developed using statistical techniques, machine learning methods and an MP discriminant analysis model. A range of datasets were used because it is desirable to identify methods for developing classification models that work well in a wide range of problems. Indeed, Dietterich (1998) argues that this is one of the central issues in classification analysis, particularly with regard to machine learning methods.

3.4.1 The Datasets

The datasets used comprised four credit application datasets (Australian, German, Greek and SPSS) and a small business loans dataset. The Australian and German datasets, which were obtained from the UCI repository (e.g. Frank and Asuncion, 2010), have been used in several credit scoring studies, e.g. Piramuthu (1999b), Baesens (2003). The Australian datasets contains data for 690 credit card applications for an Australian bank, with 383 observations in one class and 307 observations in the other class. Each observation consists of six continuous variables and eight categorical variables (four variables with two categories, two with three categories, one with nine categories and one with 14 categories). The German dataset contains data for 1,000 credit card applications, with 700 observations in one class and 300 in the other class, and 20 variables for each observation (eight numeric and 12 categorical). The Greek dataset, which was provided in confidence by a Greek bank, contains 14,413 observations with 11,438 cases in ‘good’ class and 2,975 in the bad class. Each observation consists of 39 variables. The SPSS dataset is provided as an example dataset by SPSS, e.g. SPSS (2001), software and consists of 1,500 cases, with 949 cases in one class and 551 cases in the other class. Each observation consists of 8 variables (one categorical and seven numeric

variables). The small business loans dataset is provided by the Federal Reserve Board (1998) and contains 3,661 cases, with 3,047 observations in group one and 514 observations in group 2. Each observation consists of 9 variables, mainly accounting ratios. More information about every dataset is included in appendix A.

For the benchmarking study, data in each of the datasets were transformed using weight of evidence. For categorical variables, the WoE was calculated for each category and categories with similar WoE were grouped together into one feature. Judgements about similarity were subjective on the basis of visual inspection of WoE histograms. Continuous variables were first divided into deciles, which were then grouped together according to the WoE. These procedures result in each of the original variables being replaced by a small set of indicator variables or features. This approach is very popular in practice in the industry. The main disadvantage of this approach is that the number of variables can become extremely large with a subsequent possibility of overfitting. After transformation, the Australian dataset contained 37 binary features, the German dataset contained 51, the Greek dataset contained 45, the SPSS dataset contained 35, and the SME dataset contained 25. In practice, the nature of features would be taken into account in selecting features for scorecard development (e.g. Anderson, 2007) but, as in other studies (e.g. Piramuthu, 1999a; Liu and Schumann, 2005), these practical considerations were not taken into account and all features were included in this benchmarking study.

3.4.2 Methods Used in the Benchmarking Study

In the benchmarking study, the statistical, machine learning and MP methods that are most widely suggested for developing credit scoring models were applied to each of the five datasets. The statistical methods used were linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic regression, *k*-nearest neighbours (*k*-nn) with $k=3$ and $k=10$, naïve Bayes and classification tree. The machine learning methods considered were a multilayer neural network, a linear support vector machine (SVM), a radial basis function (RBF) SVM and a polynomial kernel SVM, e.g. Cristianini and Shawe-Taylor (2000). The linear SVM uses linear functions to estimate feature weights, whereas the

RBF and polynomial kernel SVMs use different types of functions to generate non-linear classifiers. The MP discriminant analysis model used in this study is the MSD model normalised for invariance under origin shift, i.e. model 2.8.

Commercial software was used to develop classification models for each of the five datasets. MatlabArsenal toolbox (Yan, 2006) was used for the LDA, QDA, logistic regression, *k*-nn and neural network methods. Weka open source software (Witten and Eibe, 2005) was used for the naïve Bayes, classification tree and SVM classifiers. Xpress-MP mathematical programming software (Dash Associates, 2006) was used for the MSD model.

3.4.3 Performance Assessment

Four measures were used for assessing the performance of the classifiers in this benchmarking study, namely accuracy (overall and in each class) and area under the ROC curve, i.e. AUC, where AUC is expressed as a percentage of the maximum possible area under an ROC curve. These measures have been used extensively in other comparison studies, e.g. Piramuthu (1999a), Doumpos et al (2001), Daskalaki et al (2006). In this benchmarking study, accuracy and AUC are calculated from the mean value of the same ten randomizations. In these randomizations, the dataset was randomly partitioned into a subset with 80% of observations and a subset with the remaining 20% of observations. The larger, i.e. 80% of observations, subset was used for training and out-of-sample performance was evaluated on the smaller, i.e. 20% of observations, subset. For each classification model, the average out-of-sample performance for 10 randomizations of each dataset was then determined. Paired *t*-tests were used to compare the average holdout sample hit rates produced by the methods, with the *t*-statistic acting as an indicator of potentially significant differences between hit rates. Analytical results are included in Appendix C.

3.5 Benchmarking Study Results

The results for the Australian, German, Greek, SPSS and the small business loans dataset are discussed in sections 3.5.1, 3.5.2, 3.5.3, 3.5.4 and 3.5.5 respectively.

3.5.1 Australian Dataset Results

For each method used to generate a classifier, the out-of-sample performance over 10 randomisations of the Australian dataset is presented in Table 3.2 in terms of the overall accuracy, accuracy in class 1, accuracy in class 2 and AUC as a percentage of the maximum area under an ROC curve.

Method	Accuracy (%)			AUC
	Overall	Class 1	Class 2	(%)
LDA	90	91	89	97
QDA	90	88	91	95
Logistic Regression	91	91	90	96
3-Nearest Neighbours	87	85	88	93
10-Nearest Neighbours	89	85	93	95
Naïve Bayes	85	84	94	94
Classification Tree	**92**	91	93	94
Neural Network	90	91	90	96
SVM – Linear	91	90	93	91
SVM – RBF	**92**	90	94	92
SVM – Polynomial	89	89	90	89
MSD	90	89	89	96

Table 3.2: Classifier Performance on Australian Dataset

As can be seen from the results in Table 3.2 for the Australian dataset, the classification tree and SVM-RBF produced classifiers with the highest overall accuracy, with on average 92% of the total observations in the holdout samples classified correctly. The paired *t*-test results indicate that the overall accuracy of both these methods is significantly better than logistic regression, 3-*nn*, 10-*nn*, naïve Bayes, neural networks, SVMs, MSD, LDA and QDA. The paired *t*-test results also indicate that the overall accuracy of the MSD model is significantly better than 3-*nn*, and naïve Bayes.

The results in Table 3.2 also show that none of the classifiers had any issues in classifying correctly observations from both classes, i.e. there was no substantial bias in favour of either class. The classifier that was most accurate for the majority class was naïve Bayes (on average 94% of the observations were classified correctly). However the same classifier did not perform as well for the minority class (on average only 84% of the

observations were classified correctly). Under the AUC criterion the best performance was achieved by LDA (97%), while SVM-polynomial had the poorest performance (89%). Overall, the MSD model achieved good results on each of the performance measures.

3.5.2 German Dataset Results

The out-of-sample performance over 10 randomizations of the German dataset is presented in Table 3.3 for each of the methods used to produce classifiers.

Method	Accuracy (%)			AUC
	Overall	Bads	Goods	(%)
LDA	72	74	71	79
QDA	72	70	73	77
Logistic Regression	**75**	50	85	78
3-Nearest Neighbours	71	31	89	71
10-Nearest Neighbours	70	23	91	72
Naïve Bayes	73	61	82	81
Classification Tree	71	41	84	64
Neural Network	70	49	79	71
SVM – Linear	**75**	43	89	66
SVM – RBF	71	10	99	54
SVM – Polynomial	68	48	77	63
MSD	**75**	47	86	78

Table 3.3: Classifier Performance on German Dataset

In the German dataset (table 3.3) the best overall accuracy is achieved by MSD, logistic regression and SVM-linear, with on average 75% of the observations classified correctly. All three methods achieve significantly better overall accuracy than LDA, QDA, neural networks, 3-*nn*, SVM-RBF, SVM-polynomial, and classification tree. The MSD model performed slightly better compared to logistic regression and SVM-linear. The best accuracy for the bad class is achieved by the LDA, with on average 74% of the bad class observations classified correctly. The best performance for the good class was achieved from the SVM-RBF, with on average 99% of the good class observations classified correctly, but this classifier achieved the

worst performance for the bad class (10%). It is clear that there can be a trade-off between the classification of observations from the bad and the good classes. In the results in Table 3.3, only LDA and QDA achieved balanced results, with average accuracies for both classes of about 72%. It can also be seen that all methods except LDA are biased in favour of the good class, i.e. have lower classification accuracy in the bad class. For the AUC criterion, naïve Bayes had the best performance (81%) and SVM-RBF had the poorest performance (54%).

The MSD model managed to predict correctly 86% of the cases from the good class (i.e., one of the highest percentages for this dataset), and 47% from the bad class (c.f. the best accuracy of 74% for the bad class). The MSD model also achieved the third highest AUC metric, i.e. 78%. .

3.5.3 Greek Dataset Results

For each method used to generate a classifier, the average out-of-sample performance on the Greek dataset is presented in Table 3.4. It should be noted that this dataset is extremely imbalanced as the bad class consists only 20% of the whole population. This feature creates problems to the performance of the classifier as it is dominated by the majority class.

Method	Accuracy (%)			AUC
	Overall	Bads	Goods	(%)
LDA	63	63	64	67
QDA	63	62	63	67
Logistic Regression	80	6	99	68
3-Nearest Neighbours	76	18	92	63
10-Nearest Neighbours	77	10	94	65
Naïve Bayes	81	30	56	65
Classification Tree	81	9	97	59
Neural Network	75	8	86	66
SVM – Linear	83	5	100	52
SVM – RBF	82	0	100	58
SVM – Polynomial	**83**	6	100	53
MSD	80	5	100	68

Table 3.4: Classifier Performance on Greek Dataset

The best overall performance on the Greek dataset (Table 3.4) was achieved by the SVM with a polynomial kernel, with on average 83% of observations classified correctly, while the worst overall performance was achieved by the LDA method, with on average 63% of the overall observations classified correctly. Paired *t*-tests indicate that MSD performed significantly better than LDA, QDA, logistic regression, neural network, 3-nn, and naïve Bayes. A feature of the performance on the Greek dataset is that there are examples of extremes with, for example, SVM-RBF, assigning all the bad cases to the good class and failing completely to predict bad cases. Indeed the SVM-RBF achieves the worst performance in the bad class, with no observations classified correctly, but this classifier classified correctly all the observations from the good class. The best performance for the bad class was achieved by LDA (63%). These results have similarities with the results obtained on the German dataset, with poorer classification accuracy in the bad class. Using the AUC criterion the best performance was achieved by the MSD model (68%) and SVM-linear had the worst performance (52%).

The MSD model managed to predict correctly 100% of the cases from the good class (which is one of the highest percentages for this dataset), and 5% from the bad class (which is one of the lowest for this dataset). MSD also achieved one of the highest AUC metric, i.e. 68% and one of the highest overall accuracy ratios.

3.5.4 SPSS Dataset Results

The average out-of-sample performance on the SPSS dataset is presented in Table 3.5 for each method used to generate a classifier.

Method	Accuracy (%)			AUC
	Overall	Bads	Goods	(%)
LDA	74	75	71	82
QDA	73	69	73	81
Logistic Regression	75	59	85	83
3-Nearest Neighbours	70	53	89	76
10-Nearest Neighbours	74	56	91	79
Naïve Bayes	75	70	82	78
Classification Tree	73	60	84	77
Neural Network	71	57	79	77
SVM – Linear	**76**	65	89	73
SVM – RBF	76	55	99	72
SVM – Polynomial	69	56	77	*66*
MSD	75	62	86	82

Table 3.5: Classifier Performance on SPSS Dataset

From the results for the SPSS dataset in Table 3.5, it can be seen that SVM-linear achieved the best overall accuracy rate, with on average 76% of observations classified correctly. Logistic regression, MSD, SVM-RBF, and naïve Bayes also performed well. Paired *t*-tests indicate that MSD performs significantly better than QDA, neural network, 3-*nn*, 10-*nn*, SVM-polynomial, classification tree, and naïve Bayes. The worst overall accuracy (69%) was performed by the SVM-polynomial, which also performed poorly under the bad class accuracy, with on average 56% of bad class observations classified correctly. The best performance under the bad class accuracy was achieved by LDA, with on average 75% of bad class observations classified correctly and the worst performance under this criterion was achieved by the 3-NN, with 53% of bad class observations classified correctly. For the good class accuracy criterion, the best performance was achieved by the SVM-RBF (99%) and the worst performance was achieved by LDA (71%). Under the AUC criterion the best performance was achieved by logistic regression (83%) and the worst by SVM-polynomial (66%). On the SPSS dataset the classifiers generally performed better on the good class, and although performance tended to be poorer on the bad class, the differences in performance were not as extreme as on the German dataset.

The MSD model performed relatively well in its overall (75%), good case (86%) and bad case (62%) predictions. The MSD also achieved one of the best results on the AUC metric (82%).

3.5.5 SME Dataset Results

The average out-of-sample performance on the SME dataset is presented in Table 3.6 for each method used to generate a classifier.

Method	Accuracy (%)			AUC
	Overall	Bads	Goods	(%)
LDA	83	62	87	84
QDA	86	51	92	84
Logistic Regression	**89**	36	98	84
3-Nearest Neighbours	87	22	98	77
10-Nearest Neighbours	88	19	99	72
Naïve Bayes	86	82	82	82
Classification Tree	88	30	97	67
Neural Network	88	35	97	83
SVM – Linear	89	37	97	67
SVM – RBF	87	18	99	58
SVM – Polynomial	85	33	93	59
MSD	88	34	97	77

Table 3.6: Classifier Performance on SME Dataset

As can be seen from Table 3.6, logistic regression achieved the best overall classification performance on the SME dataset, with on average 89% of observations classified correctly. Paired *t*-tests indicate that the overall classification performance of MSD was significantly better than LDA, QDA, and naïve Bayes. The worst overall performance was achieved by LDA (83%). For the bad class, the best accuracy (82%) was achieved by naïve Bayes and SVM-RBF had the worst classification performance (18%), but on the good class the performance of these two methods was reversed, with SVM-RBF achieving 99% accuracy for the good class. Under the AUC criterion the best performance was achieved by the logistic regression (84%) and the worst by the SVM-RBF (58%).

The MSD model performed well based on AUC metric (77%) and

overall accuracy (88%). The MSD model also predicted correctly 97% of the cases from the good class (which is good for this dataset), but as with all methods other than naïve Bayes, the classification performance on the bad class was poorer, with only 34% of cases predicted correctly.

3.6 Summary

A number of methods for developing scorecards for consumer and small business lending have been evaluated in a benchmarking study using different datasets, with data transformed using techniques that are widely used in practice for scorecard development. In particular, this study compared the performance of commonly used statistical methods (linear discriminant analysis, quadratic discriminant analysis, logistic regression, *k*-nearest neighbours, naïve Bayes and classification tree) and machine learning techniques (multilayer neural network and three types of support vector machines) for developing scorecards with the MSD mathematical programming discriminant analysis model. Although the MSD model has been included in other comparative studies (e.g. Baesens, 2003) very simple normalisations were used.

A general conclusion from the benchmarking study is that there is not a unique method for developing scorecards that will produce classifiers that perform better than other classifiers under all data conditions. Similar results have been found in other studies (e.g. Srinivisan et al, 1987; Henley, 1995; Desai et al, 1997). The choice of method for developing a classifier should therefore depend on the characteristics of the problem. In general, however, the benchmarking study results indicate that classifiers developed using logistic regression, linear SVM and MSD were found to perform well on the five datasets. The performance of the classifiers was also found to be affected by the proportion of observations in each class, with a tendency for classification to be biased towards the majority class in the case of imbalanced datasets. Methods for dealing with imbalanced datasets, particularly in using MP models to develop classifiers, will therefore be investigated in a later chapter. A limitation of this benchmarking study is that all features generated after data transformation by WoE were used in developing the classifiers. In practice, only a limited number of features would be used in developing a classifier. As there has been only limited

research in feature selection for mathematical programming discriminant analysis models, this topic will be considered in the next chapter.

Chapter 4

4. Feature selection

4.1 Introduction

Many classification decisions such as credit risk assessment and medical diagnosis are based on limited information. For decisions of this type, classification models may be used to assign observations or objects of unknown class to one of a number of specified classes based on the values of a set of features associated with each observation or object. These classification models can be developed using statistical techniques such as discriminant analysis and logistic regression (e.g. Hand, 1997), machine learning methods such as neural networks (e.g. Ripley, 1994), or mathematical programming (MP) discriminant analysis models (e.g. Stam, 1997). The features used in developing a classification model may be the raw variables associated with each observation, or features may be constructed from one or more raw variables. Although a large number of features may be available, it is often desirable to base the classification model on a limited number of features in order to simplify the model and reduce its data requirements. By developing a parsimonious classifier not only will data collection and storage costs be reduced, but it may also be possible to improve classification performance and enhance understanding of the classification criteria (e.g. Guyon and Elisseeff, 2003).

A number of feature selection techniques have been proposed. Some of these techniques are associated with specific methods for developing classification models, but others can be applied more generally. For example, complete enumeration can, at least in principle, be used with all classification model development methods to determine the best subset of features or the best subset of specified size, but the computational effort would generally be prohibitive. Stepwise forward and backward methods can also be used as a general feature selection methodology in which a new feature is added (in stepwise forward methods) or removed (in stepwise backward methods) at each step, with an appropriate criterion used to choose the feature to add or remove. These stepwise feature selection methods, and extensions which allow a feature to be removed/added after it has been added/removed, are widely used in statistical approaches (e.g. Hand, 1997), but it is unlikely that stepwise methods will find the subset of

features that is best in terms of the class separation criterion.

Machine learning techniques for developing computer based classification models often incorporate either a filter-based or a wrapper-based method for feature selection (e.g. Kohavi and John, 1997). In filter-based approaches, features are selected in a pre-processing stage, e.g. using correlation with class membership (Blum and Langley, 1997). Although the filter-based feature selection is rapid (Blum and Langley, 1997), interactions between subsets of features and biases in the induction algorithm used to produce the classifier are ignored (Kohavi and John, 1997). Filter-based methods may also risk discarding useful features as a feature that seems completely useless by itself may be valuable if used in combination with other features (Guyon and Elisseeff, 2003). In wrapper-based approaches, feature selection is linked to the induction algorithm and heuristics, many of which are based on forward or backward stepwise procedures, are generally used to search the feature subset space, with selection criteria related to the performance of the induced classifier (e.g. Kohavi and John, 1997). As the algorithm must be run from the start to test a specific subset of features, wrapper-based methods can be computationally intensive in problems with large training samples and many features.

Filter methods have been used in a number of studies. For example, Tsai (2009) compared five methods of feature selection (*t*-test, correlation, stepwise regression, principal components analysis and factor analysis) as the input to a neural network. In this study, which used both bankruptcy and credit scoring datasets, it was found that none of the feature selection methods performed best on all datasets. Chen and Li (2010) compared the performance of four feature selection methods for input to an SVM, but the number of features to be selected was specified in advance. Ping and Yongheng (2011) used credit scoring data to compare the performance of different feature selection methods as input to an SVM, a classification tree and a *k*-nearest neighbours model. This study found that a rough sets method (Pawlak, 1982) for selecting features for input to an SVM was found to produce the best classifier, but although this approach was called a "hybrid SVM-based" model, it is more appropriate to consider it as a filter method to select features for an SVM. Similarly, although Oreski et al (2012) proposed a hybrid system with genetic algorithm and neural network for credit risk assessment using data from a Croatian bank, the approach uses a genetic algorithm as a filter method to select features for input to a neural network.

Although wrappers have been criticised as "brute force" methods requiring massive amounts of computation, e.g. Pal and Mitra (2004), they have been applied in credit scoring (e.g. Liu and Schumann, 2005). Wang et al (2012) proposed a hybrid feature selection method based on rough sets (Pawlak, 1982) and scatter search (Glover, 1998) as a wrapper for logistic regression, classification tree and neural network models. This approach produced promising results, but as only two small datasets were used for testing, generalized conclusions cannot be made.

There are other machine learning techniques in which features are selected in a pre-processing stage before the classifier is trained. For example, ReliefF (e.g. Robnik-Šikonja and Kokonenko, 2003) is an iterative procedure for determining a measure of each feature's ability to separate observations of different group membership. In the ReliefF procedure for the two-group problem, a series of observations is randomly generated and each observation's K, $K>1$, nearest neighbours in the same group and in the other group are identified. For each observation in this series, each feature's separation measure is updated by adding the average difference between this feature's value in the observation and its value in the K nearest neighbours from the other group, and subtracting the average difference between this feature's value in the observation and its value in the K nearest neighbours from the same group. For feature selection, features are ranked by the value of the separation measure.

In MP methods for developing classification models, an MP model is used to generate a discriminant function that separates the observations in a training sample of known group membership into the specified groups optimally in terms of a group separation criterion (e.g. Stam, 1997). The simplest models for MP discriminant analysis are linear programming (LP) models in which the group separation measure is generally based on the deviations of misclassified observations from the discriminant function, with minimisation of the sum of deviations (MSD) being the most widely used objective. One of the advantages of MP discriminant analysis is that classification accuracy, i.e. the number of correctly classified observations, can be used directly as the group separation criterion in a mixed integer programming (MIP) model by associating a binary variable with each training sample observation. Due to the binary variable requirements, these MIP models for maximising classification accuracy (MCA) or minimising misclassifications can only be applied to relatively small discriminant

problems, although a two-stage MP based approach (Stam and Ragsdale, 1992) and an iterative MP procedure (Glen, 2003) have been proposed for larger problems.

Several approaches have been proposed for feature selection in MP discriminant analysis models. Nath and Jones (1988) proposed a jackknife procedure for feature selection in MP discriminant analysis models, but this procedure is computationally intensive and it may not identify the best subset of features for the group selection criterion of the MP model. Koehler (1991) used MP as a framework for describing the problem of determining the minimum number of features required in discriminant functions that minimise misclassifications, but the problem was not fully formulated in MP terms and a genetic solution algorithm was proposed. Bradley et al (1997) formulated the feature selection problem as an MP model in which the objective function is a parameterised linear combination of the average sum of deviations of misclassified observations and the number of features. The binary variables associated with inclusion of each feature were then approximated in two ways and solution algorithms were proposed, although solutions may not be globally optimal. Glen (1999) has shown that by using integer programming techniques, MP discriminant analysis models can be extended to determine the best subset of features of specified size for the MP model's group selection criterion, e.g. sum of deviations in the MSD model or classification accuracy in the MCA model. The original MP models for determining the best subset of features of specified size (Glen, 1999) were normalised for invariance under origin shift and required a pair of binary variables for each feature. However, by using a special ordered set of type 1 (SOS1), i.e. a set of variables of which at most one may be non-zero, to represent the discriminant function coefficient of each feature, only one binary variable per feature is required in these MP models for feature selection (Glen, 2006). In an MSD based multi-objective approach for gene selection, Sun and Xiong (2003) used only one binary variable per feature, but this model was not normalised for invariance under origin shift and the number of features in the subset cannot be specified.

In MP feature selection discriminant analysis models, features should ideally be selected based on their impact on classification accuracy, i.e. by extending the MCA model, but since the MCA model requires a binary variable for each training sample observation, the feature selection MCA model can only be applied to relatively small problems. In this chapter, two

heuristic feature selection methods based on the MCA model are proposed for two-group discriminant problems with large datasets of observations and these heuristics are tested on three credit datasets.

In section 4.2 the feature selection methods most commonly used in the credit industry are described. MP-based methods for feature selection are considered in more detail in section 4.3. In section 4.4, two MP heuristics based on the MCA model are proposed for the feature selection problem. These heuristics are then tested on three credit datasets. The findings are summarised in section 4.5

4.2 Feature Selection in Credit Scoring

In application scoring, where it is necessary to predict the behavior of an applicant for a loan, a number of features from an application form or credit bureau databases, e.g. age, occupation, education, credit history, are considered in order to predict a customer's behaviour. The metrics most widely used in practice for feature selection are the χ^2-statistic and the information statistic.

To use the χ^2-statistic and the information statistic for feature selection, consider a credit scoring dataset containing G good observations and B bad observations, with each observation consisting of the values of n binary features. For feature j, j=1,2,….,n, let g_j and b_j denote the total number of goods and bads respectively in observations in which feature j has value 1.

4.2.1 The χ^2-statistic

The χ^2-statistic measure is a non-parametric statistic for examining the relationship between categorical variables (Siegel, 1988). To calculate the χ^2-statistic for feature j, j=1,2,….,n,, let $\hat{g}_j$ and $\hat{b}_j$ denote the expected number of goods and bads respectively in observations in which feature j has value 1, where

$$\hat{g}_j = (g_j + b_j)G/(G + B) \qquad \text{and} \qquad \hat{b}_j = (g_j + b_j)B/(G + B).$$

The χ^2-statistic (with one degree of freedom) for feature j, j=1,2,….,n, is then given by

$$\chi^2 = (g_j - \hat{g}_j)^2/\hat{g}_j + (b_j - \hat{b}_j)^2/\hat{b}_j + (\hat{g}_j - g_j)^2/(G - \hat{g}_j) + (\hat{b}_j - b_j)^2/(B - \hat{b}_j)$$

and for feature selection, features are ranked by the value of this statistic.

4.2.2 The information statistic

The information statistic, F_j, for feature j, j=1,2,….,n, is given by

$$F_j = (g_j/G - b_j/B)\log(g_jB/b_jG).$$

For feature selection, features are ranked by the value of this statistic. Information statistic values for predictive features range from 0 to about 3, with higher values indicating a stronger relationship with the outcome variable (e.g. Mays and Yuan, 2004). A value below 0.01 indicates that a feature has very little predictive ability and should not be considered further unless there is a business reason, while a value in the range (0.01, 0.3) indicates that a feature should be considered for more tests (e.g. Mays and Yuan, 2004). The information statistic gives little weight to features that provide information for only a small portion of the sample.

4.3 MP Approaches for Feature Selection

MP discriminant analysis models do not require assumptions about the distributions of deviations from the discriminant function and do not produce estimates of the statistical properties of the function's parameters. Glorfeld and Gaither (1982) criticised the usefulness of LP-based discriminant analysis models partly because of the failure to deal with the feature selection problem. Although Freed and Glover (1982) commented that post-optimal analysis of LP-based discriminant analysis models could help in choosing appropriate features, a detailed approach was not proposed.

Nath and Jones (1988) proposed a variable selection method for use with LP discriminant analysis models based on the jackknife method (e.g. Efron, 1981). This approach, which involves running the LP discriminant analysis model a number of times with each observation omitted in turn, is computationally intensive when applied to problems with a large number of observations. Stam (1997) noted that although the methodology for feature selection proposed by Nath and Jones (1988) was an important contribution to MP-based discriminant analysis, there was a need for further research in this area.

Feature selection methods based on MP techniques were also proposed by Bradley et al (1997) and Bredensteiner and Bennett (1998). Both these approaches use MP methods to minimise the number of features included in the model while minimising the error rate, but both formulations are computationally intensive. Both methods were found to be effective in

eliminating redundant features while minimising cross-validation errors, but these results were obtained from small problems.

Ziari et al (1997) proposed a technique based on resampling estimation procedures, i.e. jackknife and bootstrap, to develop a statistical discriminant MP model using an approach similar to Nath and Jones (1988). This methodology produces parameter estimates and statistical properties that could be used to form confidence regions and to test the significance of the discriminant function's coefficients. The results obtained by this method on small credit scoring datasets were superior to the results from a simple MP model. However, as the computational effort depends on the resampling estimation technique, the sample size and the precision required for the estimates, this methodology is not suitable for use with large datasets.

Glen (1999) proposed integer programming formulations in which a binary variable is associated with each feature in order to solve the feature selection problem. For example, the MSD feature selection model (2.8) can be used to generate a discriminant function in a specified number of features. However, since classifier performance will ultimately be evaluated by prediction accuracy, the use of sum of deviations, rather than classification accuracy, as the selection criterion is a potential disadvantage of model (2.8) as a tool for developing classification models.

4.3.1 The MCA feature selection model

Classification accuracy can be used as the group separation criterion in generating discriminant functions in a specified number of features by defining a binary variable β_i for observation i, i=1,2,....,m, where β_i=1 if the observation is classified correctly. In addition, to prevent observations lying on the discriminant function being regarded as correctly classified, as in MSD model (2.8), define a small rejection interval Δ, Δ>0, about the discriminant function so that observations in this interval are regarded as misclassified. Defining other symbols as before and normalising for invariance under origin shift, MCA discriminant functions in p, $1 \leq p \leq n$, features can be generated by the model:

$$\text{Maximise} \quad \sum_{i=1}^{m} \beta_i \tag{4.1a}$$

$$\text{subject to} \ \sum_{j=1}^{n} X_{ij}(a_j^+ - a_j^-) - a_0 + (U + \Delta)\beta_i \le U \quad i \in G_1 \tag{4.1b}$$

$$\sum_{j=1}^{n} X_{ij}(a_j^+ - a_j^-) - a_0 - (U + \Delta)\beta_i \ge -U \quad i \in G_2 \tag{4.1c}$$

$$\sum_{j=1}^{n} (a_j^+ + a_j^-) = 1 \tag{4.1d}$$

$$a_j^+ + a_j^- - \varepsilon\gamma_j \ge 0 \quad j=1,2,\ldots,n \tag{4.1e}$$

$$a_j^+ + a_j^- - \gamma_j \le 0 \quad j=1,2,\ldots,n \tag{.4.1f}$$

$$\sum_{j=1}^{n} \gamma_j = p \tag{4.1g}$$

a_0 unrestricted; $a_j^+, a_j^- \ge 0$, $j=1,2,....,n$;

$\gamma_j=0,1$, $j=1,2,\ldots,n$; $\beta_i=0,1$ $i=1,2,....,m$.

By solving MIP model (4.1) for $p=1,2,\ldots,n$, the best subset of features for maximising classification accuracy in the training sample can be identified. If feature selection is not required, MCA discriminant functions normalised for invariance under origin shift can be generated by the basic MCA model with (4.1*a*) as objective and constraints (4.1*b*), (4.1*c*) and (4.1*d*). Since discriminant functions generated by this basic MCA model will not include features that do not contribute to the MCA objective, this MCA model can also be used to identify the subset of features for classification accuracy maximisation.

MCA discriminant functions provide a benchmark for assessing the training sample classification performance of linear classifiers (Stam and Joachimsthaler, 1990), where performance is measured by the hit rate, i.e. the proportion of observations classified correctly. MCA discriminant functions in p, $1 \le p \le n$, features generated by model (4.1) therefore benchmark the training sample performance of other linear classifiers in p features, although the training sample hit rate is a positively biased measure of classifier performance (e.g. Huberty, 1994). Classification performance can be estimated from the hit rate in a holdout sample, i.e. a sample of observations of known class membership that is separate from the training sample. No single method of generating discriminant functions, including MCA, will produce good linear classifiers under all data conditions, as shown in results for simulated discriminant problems in which performance was measured by the average holdout sample hit rate (e.g. Stam and

Joachimsthaler, 1990; Glen, 2006). There is, however, evidence that the feature selection MCA model can produce parsimonious classifiers which perform well in comparison with other methods (e.g. Glen, 2001), but this model can only be applied to small problems because of its binary variable requirements.

4.4 MP-Based Heuristics for Feature Selection

In the MCA feature selection model (4.1), the feature selection criterion is classification accuracy, but because a binary variable is required for each feature and each training sample observation, this model can only be applied to problems in which the total number of binary variables is such that the problem can be solved relatively easily. In order to allow MCA based feature selection methods to be used more widely, two MCA-based heuristics are proposed for feature selection in discriminant problems with a large number of observations.

In each of the proposed heuristic procedures for feature selection, assume that M observations of known group membership are available. First generate S training samples each with M_S, $M_S<M$, observations, where each training sample is generated by sampling an equal number of observations from each group without replacement, and where the total number, M_S, of training sample observations is such that the resulting MCA discriminant problem with M_S, observations is computationally tractable. A set of T pairs of training and holdout samples is also generated to evaluate the performance of classifiers generated from specified subsets of features. Each of these pairs of training and holdout samples is generated by partitioning the set of M observations of known group membership into a training sample with M_T observations, $M_S<M_T<M$ and an associated holdout sample containing the remaining $M-M_T$ observations, where the number, M_T, of training sample observations is not limited by the computational requirements of the MCA model.

4.4.1 MCA Heuristic 1: the number of features is specified

The heuristic procedure for selecting a specified number of features will generally be used to determine the best subsets of features in a given range, q to r ($q\geq 1$, $r\leq n-1$), in the number of features. For each value, p, in the

required range, i.e., $q \leq p \leq r$, this heuristic procedure consists of the following stages:

Stage 1: For a specified number, p, of features, apply the MCA feature selection model (4.1) to each training sample, s, $s=1,2,\ldots,S$, with M_S observations, and determine the best subset Φ_{sp} of p features for training sample s.

Stage 2: Determine the subset Ω_p, of p features that occurs most frequently in the subsets Φ_{1p}, Φ_{2p}, …., Φ_{sp}, where Ω_p is not unique if there are ties in the feature subsets occurring most frequently.

Stage 3: Evaluate the performance of MSD discriminant functions in the p features of subset Ω_p by using the MSD model (2.1) normalised for invariance under origin shift to generate the MSD discriminant function in these p features for each of the T training samples with M_T observations and determining the average hit rate in the associated holdout samples.

By repeating stages 1, 2 and 3 for all values of p in the required range, i.e., for $p=q, q+1,\ldots, r$, the subset of features with the best classification performance can be determined.

This heuristic procedure for selecting a specified number p, $q \leq p \leq r$, of features has similarities with voting algorithms for classification (e.g., Bauer and Kohavi, 1999), particularly the bagging (i.e., bootstrap aggregating) algorithm (Breiman, 1996) in which s bootstrap samples with m observations are generated by randomly sampling m observations with replacement from the set of M observations of known class membership and a classifier is generated from each bootstrap sample. The output from these s classifiers is aggregated by voting, where a new observation is assigned to the class to which it is allocated most frequently by these s classifiers, with ties broken by selecting randomly from the classes involved.

A possible disadvantage of this feature selection heuristic is that since selected features must have discriminant function coefficients of at least the threshold value, ε, some features may be selected with coefficient value ε simply to ensure that p features are selected, so that in these cases features may be selected in an arbitrary way. For this reason another heuristic method is also proposed.

4.3.2 MCA Heuristic 2: the number of features is not specified

The heuristic feature selection procedure in which the number of features to be selected is not specified uses the MCA model normalised for invariance under origin shift, i.e., model (4.1) without variables γ_j, $j=1,2,...,n$, and constraints (4.1*e*), (4.1*f*) and (4.1*g*). This heuristic procedure consists of the following stages:

Stage 1: For each training sample, s, $s=1,2,...,S$, with M_S observations, use the MCA model normalised for invariance under origin shift to generate the MCA discriminant function, and determine the subset Ψ_S of variables with non-zero value coefficients in this function.

Stage 2: Rank the features in order of their frequency of occurrence in subsets Ψ_S, $s=1,2,...,S$, with possible ties in this ranking.

Stage 3: Use a stepwise forward approach to evaluate the ranked list of features generated in Stage 2. In this stepwise approach, the most highly ranked feature is evaluated first by generating MSD discriminant functions normalised for invariance under origin shift using the T training samples with M_T observations and determining the average hit rate on the associated holdout samples. Features, or groups of features, are then added in rank order and the process is repeated until all ranked features have been included in this stepwise evaluation. The subset of features with the highest average holdout sample hit rate is then selected.

Note that all tied features of equal rank are introduced at the same step in stage 3 of this heuristic procedure. Ties in ranking could be dealt with by considering all combinations of the tied features, but this approach is not adopted because of the potential computational effort required, particularly at the first step where a group of features may be ranked most highly.

This heuristic procedure will generally require considerably less computational effort than the heuristic procedure for selecting a specified number of features. In some applications, however, a number of features may be included in all the functions generated in the first stage by the MCA model, so that all these features will be ranked equally as occurring most frequently. This second heuristic procedure may therefore not be appropriate for the development of a classifier with a small number of features, and the

first heuristic feature selection procedure should be considered for problems of this type.

4.4 Experimental Studies

The performance of the two feature selection heuristics was tested on the Australian, German and US credit datasets (see Appendix A). As in the experimental studies in Chapter 3, these datasets were transformed using weight of evidence, so that the observations consisted of binary features. Details for the transformed features can be found in Appendix B. A short description of the three datasets, (number of goods and bads and the number of features) is given in Table 4.1.

Datasets	No of Bads	No of Goods	Total	No of Variables Generated
Australian	307	383	690	37
German	300	700	1,000	51
US	996	9,503	10,499	81

Table 4.1: Data description

The MP discriminant analysis models were set up and solved on a personal computer using Xpress-MP (Dash Associates, 2006). For the two MCA-based feature selection methods, 20 training samples with 50 observations in each group were generated from the transformed Australian and German datasets, while 50 training samples with 50 observations in each group were generated from the transformed US dataset because of the larger size of this dataset. The model was run by setting the number of features, p, in the range [15, 37] for the Australian dataset, [20, 51] for the German dataset, and [49, 81] for the US dataset. In order to assess the performance of the MSD model in stage 3 of each heuristic, each dataset was split randomly ten times into two samples, with 80% of observations forming a training sample and the remaining 20% of observations forming an associated holdout sample. The classification models were developed on each of the ten training samples and performance evaluated on the associated holdout samples.

4.4.1 Comparison of the MCA heuristics

The average holdout sample hit rates obtained by MCA heuristics 1 and 2 on the transformed Australian, German and US datasets for cases in which MCA heuristic 2 (i.e. the heuristic in which the number of features to be selected is not specified) identified subsets of p features are presented in Figures 4.1, 4.2 and 4.3, respectively. Paired t-tests were used to compare the average holdout sample hit rates produced by the two heuristics, with the t-statistic acting as an indicator of potentially significant differences between hit rates. Results for the paired t-tests can be found in Appendix C.

For the Australian dataset (Figure 4.1), the average holdout sample hit rates for the two heuristics appear fairly similar for the seven cases for which MCA heuristic 2 identified subsets of p features, with the hit rates tending to decrease as the number of features increases. The paired t-tests for the Australian dataset indicate that only in the case with 33 features, in which heuristic 1 produces a higher hit rate, is the difference in hit rates significant at the 5% level.

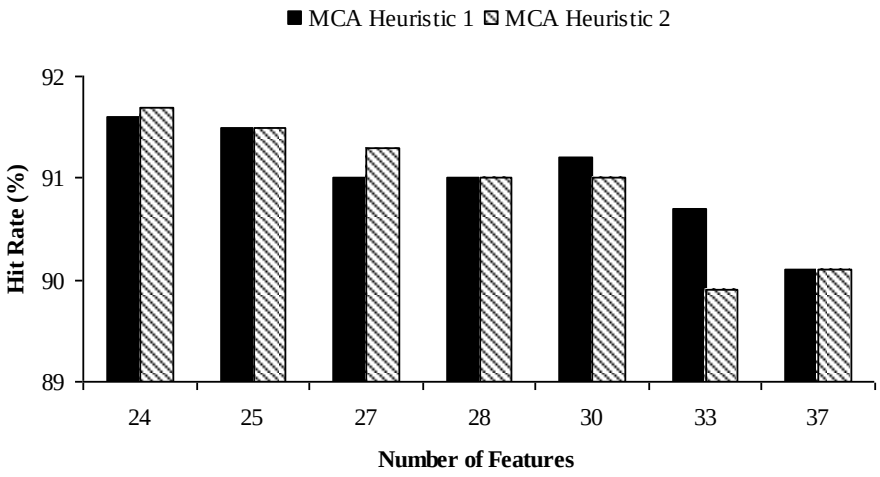

Figure 4.1: Performance of MCA heuristics on Australian dataset

On the German dataset (Figure 4.2), MCA heuristic 1 performs at least as well as MCA heuristic 2 in all cases except that with 27 features. The paired t-tests for the German dataset indicate that for the cases with 26, 32, 41, 45 and 46 features, in which MCA heuristic 1 has better performance, the difference in the two hit rates is significant.

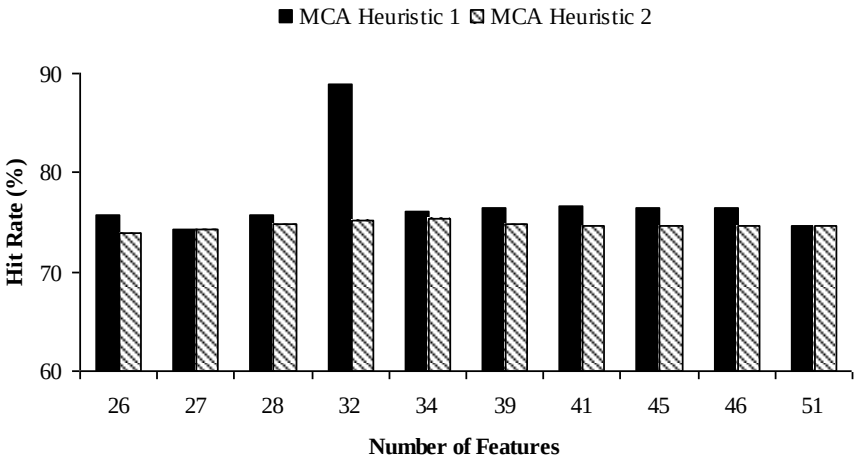

Figure 4.2: Performance of MCA heuristics on German dataset

For the US dataset (Figure 4.3), the average holdout sample hit rate for MCA heuristic 1 is at least as good as MCA heuristic 2 in all but one of the eight comparable cases, although the paired *t*-tests indicate that only for the 60-features case, in which MCA heuristic 1 has the higher hit rate, is the difference in hit rates significant.

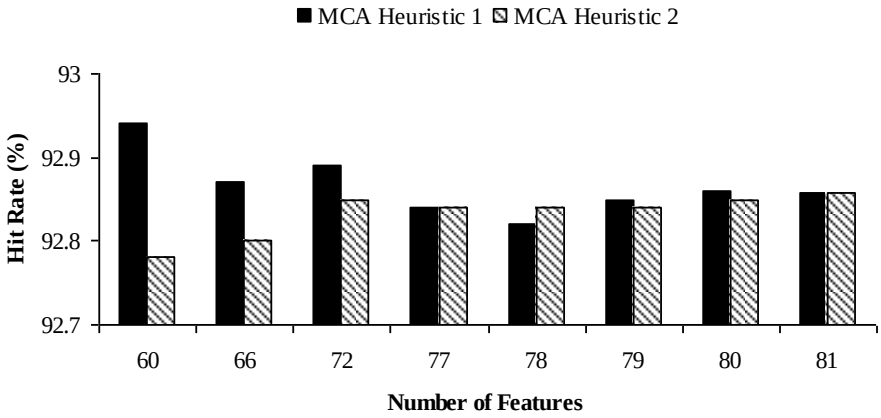

Figure 4.3: Performance of MCA heuristics on US dataset

Overall, the results from the three datasets suggest that although the difference in performance is generally small, MCA heuristic 1 is superior to MCA heuristic 2. In practice, even small improvements in scorecard performance can produce significant benefits for financial institutions (e.g. Henley and Hand, 1997).

4.4.2 Comparison of MCA heuristic 1 with other feature selection methods

As the comparison of holdout sample performance indicates that MCA heuristic 1 performs better than MCA heuristic 2, MCA heuristic 1 was compared with another four feature selection method, the χ^2-statistic, the information statistic, ReliefF and the MSD version of MCA feature selection heuristic 1. The χ^2-statistic and the information statistic are commonly recommended for use in credit scoring (e.g. Thomas et al, 2002), while ReliefF has had limited application in credit scoring (e.g. Liu and Schumann, 2005). The MSD feature selection heuristic, rather than the MSD feature selection model, was used in order to reduce the computational time.

Each of the feature selection methods was first used to produce a rank ordering of the features in the transformed Australian, German and US datasets, with WEKA open source software (Witten and Frank, 2005) used to produce the rankings for the χ^2-statistic, the information statistic and ReliefF. For each feature selection method and each dataset, features were added in rank order, the MSD discriminant function in the associated subset of features was generated for each of the ten large training samples and the performance of each function was evaluated on the paired holdout sample. The average holdout sample hit rates for MCA heuristic 1 and the other four feature selection methods on the transformed Australian, German and US datasets are shown in Figures 4.4, 4.5 and 4.6, respectively. As in the comparison of the two MCA heuristics, the paired *t*-test was used to indicate potentially significant differences between the holdout sample hit rates produced by MCA heuristic 1 and each of the other feature selection heuristics. Results for the paired *t*-tests can be found in Appendix C.

In the results for the Australian dataset (Figure 4.4), the performance of the classifiers generated by all the feature selection methods tends to deteriorate as the number of features increases. MCA heuristic 1 is superior to the other feature selection methods on this dataset for classifiers with 18 to 27 features, while the χ^2-statistic is superior for classifiers with 32 to 36 features. However, the paired *t*-tests for the Australian dataset do not indicate any significant differences between MCA heuristic 1 and the other methods.

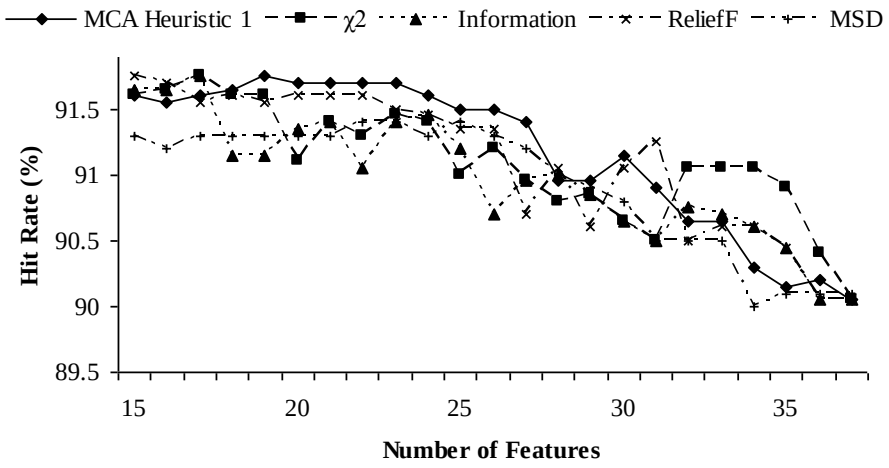

Figure 4.4:
Performance of feature selection methods on Australian dataset

The most obvious characteristic of the results for the German dataset (Figure 4.5) is the peak in the hit rate for MCA heuristic 1 with 32 and 33 features, with paired *t*-tests indicating that MCA heuristic 1 significantly outperforms the other methods for classifiers with 30 to 33 features. Since MCA heuristic 2 also produced a significantly lower hit rate with 32 features (Figure 4.2), this peak in Figure 4.5 suggests that MCA heuristic 1 has identified a strong set of 32 features for scorecard development. MCA heuristic 1 also generally performs better than the other methods on the German dataset for classifiers with 34 to 50 features, although paired *t*-tests indicate that its performance is significantly better than the other methods only for cases with 49 and 50 features. However, in the other results in Figure 4.5, MCA heuristic 1 is significantly poorer than the other methods for classifiers with 20, 22 and 23 features.

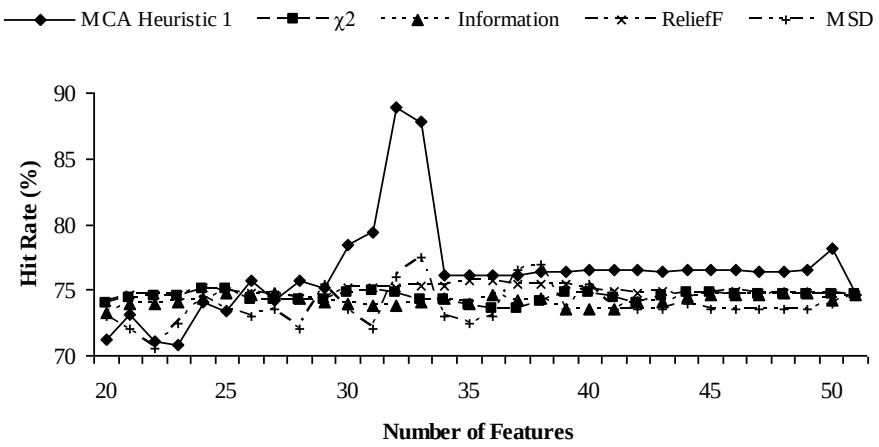

Figure 4.5:
Performance of feature selection methods on German dataset

On the US dataset (Figure 4.6), the performance of the classifiers generated for the features selected by the five methods is rather uneven, with the χ^2-statistic producing the best classifiers (with 69 features). For classifiers with 53 to 60 features, MCA heuristic 1 generally performs better than the other methods, but the paired *t*-tests indicate that MCA heuristic 1 has significantly better performance than the other methods only in the case with 55 features.

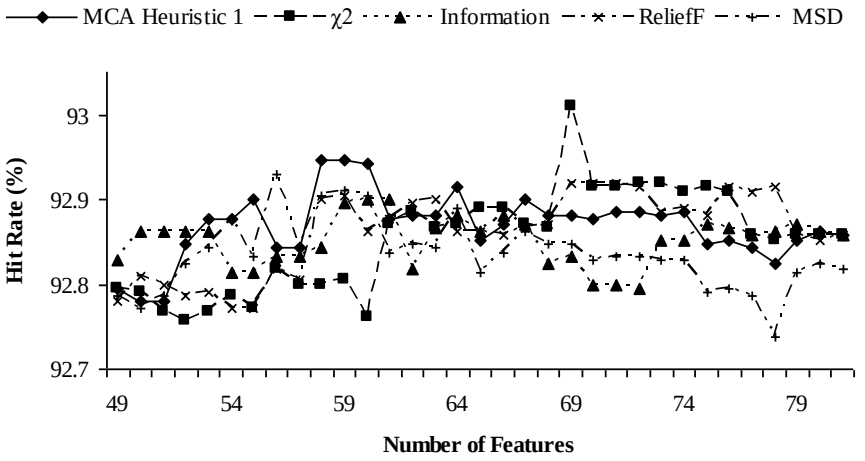

Figure 4.6:
Performance of feature selection methods on US dataset

Overall, the results from these comparisons show that classifier

performance varies from dataset to dataset, and although none of the feature selection methods outperforms the others on all three datasets, MCA heuristic 1 generally performs well. Since small differences in scorecard performance can have substantial financial impact (e.g. Henley and Hand, 1997), the results suggest that MCA heuristic 1 may be useful for feature selection in practice.

4.5 Summary

In developing classification models with a limited number of features, feature selection should ideally be based on the impact on classification accuracy. Although the MCA model can be extended for feature selection, this extended MCA model can only be applied to relatively small problems. Two heuristic methods of feature selection based on the MCA model for two-group discriminant problems are developed in this chapter. In the first heuristic the number of features to be selected is specified, while in the second heuristic the number of features to be selected is not specified. Both these feature selection heuristics use classification accuracy as the feature selection criterion but can be modified to take account of other factors. For example, if misclassification costs are available the model can be extended to select features in order to minimise misclassification costs rather than the misclassification rate. The MCA feature selection heuristics can also be extended to take account of other requirements, e.g. coefficients of certain features must be non-negative.

The two MCA based feature selection heuristics were applied to three credit scoring datasets and the results suggest that the first MCA heuristic, i.e. for a specified number of features, is generally superior to second MCA heuristic in which the number of features is not specified. The performance of classifiers generated using the features selected by the first heuristic was then compared with classifiers generated using the features selected by four other methods. Although none of the feature selection methods in this comparative study consistently performed better than the others on all three datasets and for feature subsets of all sizes, the first MCA heuristic generally performed well, suggesting that this MCA-based feature selection heuristic is a useful tool for developing parsimonious classifiers.

Chapter 5

5. Imbalanced datasets

5.1 Introduction

Irrespective of the technique used in classification model development, the training sample of observations of known class membership should ideally contain approximately the same number of observations in each class (e.g. Lewis, 1992). In practice, however, it may be difficult to obtain a balanced training sample of observations because of the nature of the classification decision for which the classification model is required. For example, in developing a model to assess the credit worthiness of applicants for credit, i.e. credit scoring (e.g. Thomas et al, 2002), where applicants are classified as good (i.e. unlikely to default) or bad (i.e. likely to default) using application form data, less than 10 per cent of cases will typically be classified as bad (e.g. Vinciotti and Hand, 2003). The degree of class imbalance in the data can be even greater in other applications, such as the identification of fraudulent credit-card transactions, where fraudulent cases typically comprise less than 0.2 per cent of total transactions (e.g. Brause et al, 1999).

5.1.1 Difficulties in Learning from Imbalanced Datasets

A classification model developed from an imbalanced dataset may be unduly influenced by the observations in the dominant class and of limited practical value. For example, if the dominant class accounts for 99% of cases, then although a classifier that assigns all observations to the dominant class will have 99% accuracy, this classifier does not take account of the implications of misclassification. For example, in assessing applications for credit it is more costly to accept an applicant who is likely to default than to reject an applicant who is unlikely to default (e.g. Adams and Hand, 1999). Methods for dealing with class imbalance must therefore be considered in developing a classification model from an imbalanced dataset. The difficulties associated with developing classifiers from imbalanced datasets arise in all types of classification problems, but only the two-class problem will be considered in this chapter.

In using linear discriminant analysis to develop classifiers, the common

covariance matrix is estimated as a weighted average of estimates of the two within-class covariance matrices, with weights normally based on the observed class sizes. If one class is much larger than the other, then the estimate will be biased towards the larger class (e.g. Klecka, 1981) and the discriminant function will have a similar bias. Standard logistic regression also weights all observations equally (e.g. Hand and Vinciotti, 2003), causing difficulties with imbalanced datasets. In using nearest neighbour methods to develop classifiers, the nearest neighbours of a minority class observation will often belong to the majority class and as a result the classifier will tend to assign a new observation to the majority class, making nearest neighbour methods vulnerable to imbalanced datasets (e.g. Tan, 2005). Brown and Mues (2012) examined the performance of logistic regression, LDA , QDA, SVMs, decision trees, *k*-nn, NNs, a gradient boosting algorithm and random forests in a credit scoring context. The results indicated that QDA, SVMs and decision trees performed very poor compared to the other algorithms.

MP methods for developing classifiers may also be unduly influenced by observations in the majority class. For example, if the MSD model (2.1) is applied to an imbalanced dataset with many fewer observations in group 1 (G_1) than in group 2 (G_2), i.e. $m_1 \ll m_2$, the resultant discriminant function will tend to be biased in favour of the majority group, i.e. G_2, so that this function will misclassify a much higher proportion of observations in the minority group, i.e. G_1, than in the majority group.

Machine learning methods also face difficulties in developing classifiers from imbalanced datasets. Classification trees are particularly sensitive to imbalanced datasets as many tests are required to separate the minority class cases from majority class cases, and therefore overfitting is very likely (e.g. Japkowicz and Stephen, 2002). Neural networks and support vector machines have also been found to perform poorly on imbalanced datasets, although Japkowicz and Stephen (2002) conclude that support vector machines are more robust than neural networks and that neural networks are superior to classification trees on imbalanced datasets.

5.1.2 Methods for Dealing with Imbalanced Datasets

An approach that can be adopted to deal with imbalanced datasets in all methods for developing classification models is to pre-process the data to

produce a more balanced dataset by either over-sampling the minority class or under-sampling the majority class (e.g. Japkowicz and Stephen, 2002). As over-sampling the minority class generally involves sampling with replacement, some observations may be replicated, leading to increased likelihood of overfitting, while under-sampling the majority class may cause some important data regions to be ignored. A method that has similarities to undersampling is the logistic regression with state dependent sample selection. In this method, the dependent variable is defined as binary, as in simple logistic regression method. The main difference is that the model is solved towards finding the probability of having a bad applicant, i.e. minority class, given its specific characteristics. Along with that, a significant number of observations that comes from the majority class needs to be deleted in order to get a more balanced dataset. In reviewing machine learning methods for developing classification models from imbalanced datasets, Chawla et al (2002) argue that under-sampling the majority class is more effective than over-sampling the minority class. Over-sampling and under-sampling strategies that concentrate on sampling observations close to the class boundary have also been proposed (e.g. Japkowicz and Stephen, 2002), but an iterative procedure must be used to identify observations to be sampled as the form of the class boundary in not known in advance. A method for generating synthetic minority class observations by interpolating between adjacent minority class observations was proposed by Chawla et al (2002), but it was noted that these synthetic observations will be biased as majority class observations adjacent to minority class observations are ignored.

In a study using both real and simulated unbalanced datasets, Louzada et al (2012) compared the performance of standard logistic regression and logistic regression with state-dependent sample selection (Cramer, 2004), which involves discarding a large proportion of majority class observations. Although there was no significant difference in the predictive performance of these two methods, differences in the distributions of default probabilities were found. This study also confirmed the benefit of working with balanced datasets where possible. Brown and Mues (2012) found that on unbalanced credit scoring datasets random forests (Breiman, 2001), in which a set of classification trees with randomly selected features is used to determine class membership by voting, and gradient boosting (Friedman, 2001), in which classification error is iteratively reduced, performed well in

comparison with a number of more commonly used statistical and machine learning techniques for developing classifiers.

Another approach for dealing with an imbalanced dataset is to take account of the consequences of misclassification, particularly the misclassification costs, in developing a classifier. These costs can be incorporated directly into MP models for minimising misclassifications (e.g. Bajgier and Hill, 1982) or maximising classification accuracy (e.g. Glen, 2001), but because a binary variable is required for each training sample observation, these mixed integer programming (MIP) models can only be applied to relatively small datasets. Misclassification costs can also be taken into account in statistical methods for estimating the probability of class membership by using a cost based threshold for class assignment (e.g. Hand and Vinciotti, 2003). Machine learning methods based on minimising misclassifications can also be extended to incorporate misclassification costs, but, as with MP and statistical methods, it is often difficult to determine misclassification costs in practice (e.g. Adams and Hand, 1999). Statistical methods can also be extended to focus on observations close to the unknown class boundary by iteratively assigning higher weights to observations close to the boundary derived at the previous iteration, as in the weighted logistic regression procedures proposed by Hand and Vinciotti (2003). Although the procedures proposed by Hand and Vinciotti (2003) are not designed specifically for imbalanced datasets, these procedures were found to outperform standard logistic regression on an imbalanced personal loan dataset when the threshold class assignment probability reflected the higher cost of misclassifying minority class, i.e. defaulting, cases.

An advantage of using MP methods to develop classifiers is that additional constraints can be incorporated in MP models to balance misclassification metrics across the classes. In this chapter, methods for extending MP models to deal with imbalanced datasets are described. The performance of classifiers produced by these extended MP discriminant analysis models and those produced by a standard MP model and logistic regression is then compared on four real datasets.

5.2 Mathematical Programming Methods for Imbalanced Datasets

As with other methods for developing classification models, MP methods for developing classifiers can address the difficulties associated

with imbalanced datasets by over-sampling from the minority class or under-sampling from the majority class, but over-sampling can result in overfitting while some data regions may be ignored with under-sampling. For the two-group discriminant problem with m_1 observations in group 1 and m_2 in group 2, Glover (1990) suggested that the objective function of the MSD model, and its goal programming extension in which both external and internal deviations are considered, can be modified to achieve balance across both groups by multiplying the objective function coefficients of deviation variables for group 1 and group 2 by m_2 and m_1, respectively. For example, the objective function of MSD model (2.1) then becomes

$$\text{Minimise} \qquad m_2 \sum_{i \in G_1} d_i + m_1 \sum_{i \in G_2} d_i \qquad (5.2)$$

but the performance of this modified MSD model on imbalanced datasets has not been investigated.

Koehler (1990) noted that for classifiers developed by both statistical and MP discriminant analysis techniques, classification errors are generally unevenly distributed across the two groups, and suggested that this problem could be addressed in MIP models for minimising misclassifications by imposing constraints on the difference in the proportion of misclassified observations in each group. For example, if z_1 and z_2 denote the number of misclassified observations in groups 1 and 2 respectively, then for γ>0 and small, the required constraints are:

$$-m_1 m_2 \gamma \leq m_2 z_1 - m_1 z_2 \leq m_1 m_2 \gamma$$

The use of this approach has not been investigated and, because of the binary variable requirements of the underlying MIP model for minimising misclassifications, it can only be applied to relatively small problems. Although Koehler (1990) only considered the problem of balancing errors in MIP models for minimising misclassifications, a similar approach can be applied to MSD-based models by imposing constraints on the difference in the mean deviation in each group:

$$-\delta \leq \frac{1}{m_1} \sum_{i \in G_1} d_i - \frac{1}{m_2} \sum_{i \in G_2} d_i \leq \delta \qquad (5.3)$$

where δ>0 and small.

Glover and Better (2007) did not consider the difficulties associated with imbalanced datasets, but suggested that for non-separable discriminant problems, it may be useful to impose an additional constraint on MSD-based models to balance the violations in each group:

$$m_1 \sum_{i \in G_2} d_i = m_2 \sum_{i \in G_1} d_i \qquad (5.4)$$

Note that constraint (5.4) can be viewed as a limiting case of (5.3) with δ=0. Although constraint (5.4) can clearly be used to deal with imbalanced datasets, its use has not been tested and it may be over-restrictive as it constrains the mean deviation in each group to be equal. In practice, the main difficulty in applying MSD models to severely imbalanced datasets is that the discriminant function generated will tend to assign observations to the majority class. This difficulty can be addressed by adding a constraint to ensure that the mean deviation in the minority class (G_1) does not exceed the mean deviation in the majority class (G_2), i.e.

$$\frac{1}{m_1} \sum_{i \in G_1} d_i \leq \frac{1}{m_2} \sum_{i \in G_2} d_i \qquad (5.5)$$

5.3 Experimental Studies

Experimental studies were performed to evaluate the performance of three MP methods for dealing with imbalanced datasets, namely (i) MSD model with balancing objective function (5.2), (ii) MSD model with range constraints (5.3), and (iii) MSD model with balancing constraint (5.5). In these experimental studies, the impact of the range parameter, δ, in the MSD model with range constraints (5.3) was investigated by using four range parameter values, namely δ=0.001, δ=0.0005, δ=0.0001 and δ=0.00001,. For comparative purposes, standard logistic regression (e.g. Hosmer et al, 2000) and the basic MSD model normalised for invariance under origin shift were also included in the experimental studies. Statistical approaches (e.g. Hand and Vinciotti, 2003) and machine learning methods (e.g. Eitrich et al, 2007) for dealing with imbalanced datasets were not included in this study as there are many variants of these methods and the results are dependent on parameter values, some of which, e.g. misclassification costs, may be difficult to determine in practice.

Four datasets consisting of application data or transaction data from financial institutions were used in the experimental studies. Dataset 1 contained 13,516 observations, each consisting of 12 variables, with 184 (1.4%) bad cases; dataset 2 contained 15,050 observations, each consisting of 8 variables, with 218 (1.4%) bad cases; dataset 3 contained 29,389 observations, each consisting of 11 variables, with 1006 (3.4%) bad cases;

dataset 4, contained 10,375 observations, each consisting of values of 21 variables, with 375 (3.6%) bad cases.

The observations in each of these four datasets consisted of the values of a set of both continuous and categorical variables. The original variables in each dataset were, as in Chapters 3 and 4, transformed to binary features by coarse classification based on weight of evidence (e.g. Thomas et al, 2002). This approach is widely used in credit scoring (e.g. Somol et al, 2005), although it leads to an increase in the total number of features. After coarse classification, there were 32 binary features in dataset 1, 18 binary features in dataset 2, 25 binary features in dataset 3, and 37 binary features in dataset 4.

For the experimental studies, each dataset was randomly partitioned ten times into a training sample with 80% of observations and a holdout sample consisting of the remaining 20% of observations. For each of the classification model development techniques, a classifier was developed from each training sample, its performance evaluated on the associated holdout sample, and the average classification performance of each technique evaluated over the ten randomisations. Average classification performance was evaluated in terms of the overall average accuracy (percentage of correctly classified observations in holdout sample), good class accuracy (percentage of correctly classified good cases in holdout sample), and bad class accuracy (percentage of correctly classified bad cases in holdout sample). Overall average accuracy is widely recommended for assessing scorecard performance (e.g. Thomas et al, 2002), but it is particularly important to consider performance in each class when classifiers are developed from imbalanced datasets. Scorecard performance should, in practice, also be monitored after implementation, with modifications made if necessary (e.g. Mays, 2004).

In the experimental studies, the MP discriminant analysis models were set up and solved using Xpress-MP (Dash Optimization, 2006) and WEKA open source software (Witten and Frank, 2005) was used for logistic regression.

5.3.1 Experimental Results

The performance of the classifiers developed by each method on datasets 1, 2, 3 and 4 are summarised in Tables 5.1, 5.2, 5.3 and 5.4,

respectively, in terms of the average percentage accuracies overall and in each class. For each dataset it can be seen that the classifiers developed by standard methods, i.e. logistic regression and the basic MSD model, have, as expected, high average accuracy in total and in the good, i.e. majority, class, but have very low accuracy in the bad, i.e. minority, class.

For each dataset it can also be seen that, in comparison with the classifiers generated by standard methods, the classifiers generated by the extended MSD models have higher accuracies in the bad class, but lower accuracies both overall and in the good class. For classifiers generated by the MSD model with range constraints (5.3), the accuracy in the bad class increases and accuracies overall and in the good class decrease with reduction of the range parameter, δ, from 0.001 to 0.00001. The performance of the MSD model with balancing constraint (5.5) is similar to that of the MSD model with range constraints (5.3) for δ=0.00001, i.e. balancing constraint (5.5) is effectively the limiting case of balancing constraints (5.3) with δ=0. On datasets 1, 3 and 4, it can be seen (Tables 5.1, 5.3 and 5.4) that the MSD model with balancing objective function (5.2), the MSD model with range constraints (5.3) for δ=0.00001 and the MSD model with balancing constraint (5.5) all generate classifiers with similar performance. However, on dataset 2, the MSD model with balancing objective function (5.2) has a larger difference between good and bad class accuracies than the MSD model with range constraints (5.3) for δ=0.00001 and the MSD model with balancing constraint (5.5).

Method	**Accuracy (%)**		
	Total	Goods	Bads
Logistic Regression	99	99	1
MSD – Basic Model	**99**	100	0
MSD – Balancing Objective	75	76	61
MSD – Range Constraints: δ=0.001	79	80	58
MSD – Range Constraints: δ=0.0005	77	77	59
MSD – Range Constraints: δ=0.0001	76	76	60
MSD – Range Constraints: δ=0.00001	76	76	60
MSD – Balancing Constraint	75	76	61

Table 5.1: Results for Dataset 1

Method	Accuracy (%)		
	Total	Goods	Bads
Logistic Regression	98	99	1
MSD – Basic Model	**99**	99	3
MSD – Balancing Objective	74	74	57
MSD – Range Constraints: δ=0.001	71	71	60
MSD – Range Constraints: δ=0.0005	69	69	62
MSD – Range Constraints: δ=0.0001	69	69	63
MSD – Range Constraints: δ=0.00001	69	69	63
MSD – Balancing Constraint	69	70	63

Table 5.2: Results for Dataset 2

Method	Accuracy (%)		
	Total	Goods	Bads
Logistic Regression	96	100	0
MSD – Basic Model	**98**	98	3
MSD – Balancing Objective	70	70	68
MSD – Range Constraints: δ=0.001	74	75	63
MSD – Range Constraints: δ=0.0005	71	71	66
MSD – Range Constraints: δ=0.0001	70	70	68
MSD – Range Constraints: δ=0.00001	70	70	68
MSD – Balancing Constraint	70	71	68

Table 5.3: Results for Dataset 3

Method	Accuracy (%)		
	Total	Goods	Bads
Logistic Regression	96	100	0
MSD – Basic Model	**99**	99	7
MSD – Balancing Objective	85	85	76
MSD – Range Constraints: δ=0.001	88	89	70
MSD – Range Constraints: δ=0.0005	87	87	73
MSD – Range Constraints: δ=0.0001	85	85	75
MSD – Range Constraints: δ=0.00001	85	85	76
MSD – Balancing Constraint	85	85	76

Table 5.4: Results for Dataset 4

Overall, the results show that the standard MSD model and logistic regression fail to perform well on imbalanced datasets. The results also suggest that the extended MSD models, i.e. with balancing objective (5.2), range constraints (5.3) or balancing constraint (5.5), outperform the standard methods in achieving balanced performance in each class, although there is some evidence that the MSD model with balancing objective (5.2) is not as effective as the MSD model with additional constraints.

5.4 Summary

There are difficulties in generating classifiers from imbalanced datasets as traditional methods tend to produce classifiers that are biased towards the majority class. The difficulties associated with imbalanced datasets can be addressed by pre-processing the data to produce balanced datasets or by considering the costs associated with misclassifying observations in each class, but these approaches have limitations. In this chapter it has been shown that MP methods can be extended, either by modifying the objective function or incorporating additional constraints, to develop classifiers from imbalanced datasets without the need to pre-process the data or incorporate misclassification costs. Although some of these extensions have been proposed previously, none of these extended models have been applied to imbalanced datasets. In this study, extended MSD models have been applied

to four real imbalanced datasets from financial institutions and it has been shown that these extended models can produce classifiers with balanced performance over the majority and minority classes.without assumptions about misclassification costs or the need to pre-process the data.

Chapter 6

6. Ordinal Classification

6.1 Introduction

In developing classifiers it is usually assumed that the class values are unordered and the groups are defined in a nominal way. The resulting classifier will not focus on the order of the observations but only assign cases to one of the nominal predefined classes. However, there are problems in which it is not enough to assign a case to one of the predefined classes, but it is also required to rank observations. This category of problems, known as ranking, sorting or ordinal classification, considers the features as evaluation criteria and the groups (and observations) are defined in an ordered way from the most to the least preferred.

Ordinal classifiers have a variety of applications. Zopounidis and Doumpos (2002) described applications of ordinal classifiers in fields such as stock evaluation, e.g. Zopounidis et al (1999), pattern recognition, e.g. Zopounidis and Doumpos (1998), job evaluation, e.g. Spyridakos et al (2001), and financial management, e.g. Doumpos et al (2001). Sorting has been used also in credit scoring, e.g. Zopounidis et al (1998). Thomas et al (2001) stress the need for ordinal scorecards when the score is used in decisions such as pricing a product or defining the percentage of applicants to accept. Ordinal classifiers can also be used in scorecard calibration or recalibration. The purpose of calibrating or recalibrating a scorecard is to make sure that a scorecard will have specific properties, e.g. positive scores or differences in scores having constant meaning (e.g. Thomas et al, 2001). Calibration is also necessary to keep the scorecard aligned with the changes in the constantly changing economy. Basel II (2006) stresses the importance of calibration for keeping the scorecard up to date under changing economic conditions.

Statistical methods have been proposed for developing ordinal classifiers. For example, ordinal logit and probit models (e.g. Borooah, 2002) are ordinal statistical methods which are similar to the models applied to nominal datasets as described in Chapter 2. The main difference is the extra cut-offs that are necessary to discriminate between the different classes. Isotonic regression is a statistical method for ordinal classification problems (e.g. Barlow et al, 1972) which is based on the same concepts as

linear regression, but with extra constraints that limit the weights of the features. This approach has been found to produce fairly good results in scorecard calibration (e.g. Schwalb et al, 2003).

Methods from machine learning have also been used in ordinal classification problems. For example, Kotsiantis and Pintelas (2003) proposed a cost sensitive approach that can be used for sorting observations according to specified criteria. The main disadvantage of this approach is the use of cost weights that are based on the intuition of the researcher. Frank and Hall (2001) suggested an approach in which an ordinal classification problem is converted into a set of binary classification problems. An advantage of this method is that it is not necessary to change the structure of the algorithm used every time, but each binary classification problem must be solved, so that for a problem with k classes it is necessary to use k–1 models to estimate the probability of class membership. For problems where it is necessary to sort observations this method is therefore computationally intensive. Another weakness of this approach is that it discards important information from the class variable that can be used for classifying the observations. For example, if the original class variable has three values, such as cool, mild, and hot, the original dataset is split into k-1 datasets, i.e. 2, where in one sub-problem the class variable values are "higher" than cool, (i.e. mild and hot) and in the other sub-problem the class variable values are "higher" than mild, (i.e. hot). The algorithm is then applied in each of the sub-problems. So, nothing will be known explicitly about the class variable of the observations as the new target variable is aggregated. For example, in the "cool, mild, and hot" application, observations with class variable "mild" will appear in the sub-problems with a different label. Shashua and Levin (2003) proposed a methodology that extends the use of support vector machines for ordinal classification problems by splitting the target variable into a number of different consecutive classes, and trying to optimise the same criterion as in the two-class problem, but with more constraints. As in the approach of Frank and Hall (2001), observations are aggregated into larger groups.

A mathematical programming method for treating ordinal classification problems was proposed by Srinivasan (1976). This LP-based method focuses on the ordinal nature of the dependent variable and tries to replicate the performance of ordinal regression, but the solution process is time consuming. Jacquet-Lagreze and Siskos (1982) proposed an LP-based

method, the UTA (utilité additive) method, for ranking problems. The UTA method uses an LP model to generate an additive utility function from a weak-order preference ranking of a reference set, i.e. a training sample, of observations based on the ranking of the observations and the values of the features associated with each observation. This additive utility function consists of piecewise-linearisations of the marginal utility functions of the features. In the UTA method, it is assumed that each function's marginal utility function is monotone non-decreasing. This limitation can be overcome by using a mixed integer programming approach (Glen, 2008) but the marginal utility must be either monotone non-decreasing or monotone non-increasing. In addition, the UTA method has only been applied to problems with a small number of observations in the training sample and problems with binary features have not been considered. The UTA method has also been extended for other applications, e.g. the UTA based discriminant analysis method can be used to generate non-linear discriminant functions by considering the class membership, rather than the ranking, of observations (e.g. Zopounidis and Doumpos, 2000).

Additive utility discriminant analysis methods are described in Section 6.2. Methods for overcoming common problems in these additive utility approaches are also described in Section 6.2. An experimental comparison of additive utility discriminant analysis methods is described in Section 6.3 using a two-class credit scoring classification problem. In Section 6.4, possible practical applications of ordinal classifiers in scorecard development are discussed and a new LP model is introduced for ranking observations. This method is tested on a small dataset in which the observations are ordered and the results are compared with results obtained using statistical methods. The conclusions from this chapter are summarised in Section 6.5.

6.2 Additive Utility Discriminant Analysis

Mathematical programming discriminant analysis models offer great flexibility over common statistical methods as they do not assume anything about the distribution of the population and can easily incorporate different goals in the objective function or have additional constraints included in the formulation. However, standard MP discriminant analysis methods treat the groups in a nominal way and do not consider any information related to the ordinal nature of some classification problems. In order to overcome this

kind of problem, the LP-based UTA model was proposed by Jacquet-Lagreze and Siskos (1982). The UTA method can be used to solve the problem of multi-criteria choice and ranking on a set of alternatives by constructing an additive utility function from a weak order preference defined by the user on a subset of reference observations.

The UTA method is based on the concept of preference disaggregation, in which the global preferences of the decision maker are disaggregated, as specified in the ranking of the reference set of observations, i.e. training sample of observations, by considering the marginal utility of each feature associated with the observations, e.g. Doumpos et al (2001). The ranked observations are described by the values of a set of features, where these features may have increasing and decreasing preference. For example in credit scoring, income can be considered to be of increasing preference while number of credit bureau searches can be considered to be of decreasing preference. In the UTA method, the marginal utility of each feature is assumed to be monotone non-decreasing, so that features with monotone non-increasing marginal utility must be transformed to monotone non-decreasing form. The LP-based UTA model attempts to determine the marginal utility function of each feature that replicates the specified ranking of observations as far as possible.

Using a notation similar to that used by Glen (2008), let X_{ij} denote the value of feature j, j=1,2,...n in training sample observation i, i=1,2,...m, and let $u_j(\cdot)$ denote the marginal utility function of feature j, j=1,2,...n. It is assumed that the utility function, $U(\cdot)$, is an additive function of the marginal utility functions of the features, so that the utility $U(X_{i1}, X_{i2},\ldots, X_{in})$ of observation i, i=1,2,...,m, is given by:

$$U(X_{i1}, X_{i2},..., X_{in}) = \sum_{j=1}^{n} u_j(X_{ij}) \qquad (6.1)$$

The marginal utility function, $u_j(\cdot)$ of feature j, j=1,2,...,n is approximated by a piecewise linear function with s_j segments with s_j+1 ordered breakpoints, P_{jk}, k=0,1,2,...,s_j, where P_{j0} is the lowest point on the scale for feature j. For observation i, i=1,2,...,m, the value X_{ij} of feature j is represented as a linear combination of weights, a_{ijk}, k=0,1,2,...,s_j, at the s_j breakpoints, these weights being non-zero for at most two adjacent breakpoints, so that for $P_{j,r-1} \le X_{ij} \le P_{jr}$, $1 \le r \le s_j$, the weights a_{ijk} are given by:

$$\alpha_{ij,r-1} = \frac{P_{jr} - X_{ij}}{P_{jr} - P_{j,r-1}} \tag{6.2a}$$

$$\alpha_{ijr} = \frac{X_{ij} - P_{j,r-1}}{P_{jr} - P_{j,r-1}} \tag{6.2b}$$

$$\alpha_{ijk} = 0, k \neq r-1, r \tag{6.2c}$$

If v_{jk}, $v_{jk} \geq 0$, denotes the marginal utility of feature j, j=1,2,...,n, at breakpoint k, k=0,1,2,...,s_j, then for observation i, i=1,2,...,m, the marginal utility, $u_j(X_{ij})$, of feature j, j=1,2,...,n, is given by

$$u_j(X_{ij}) = \sum_{k=1}^{s_j} a_{ijk} v_{jk} \tag{6.3}$$

so that the utility of observation i, i=1,2,...,m can be expressed as

$$U(X_{i1}, X_{i2}, ..., X_{in}) = \sum_{j=1}^{n} \sum_{k=1}^{s_j} a_{ijk} v_{jk} \tag{6.4}$$

6.2.1 The UTA Discriminant Analysis Model

Consider a two-group discriminant problem in which the m observations are known to belong to either group 1, i.e. G_1, or group 2, i.e. G_2, where $G_1 \cap G_2 = \emptyset$. In the UTA discriminant analysis model, it is assumed that the marginal utility function of each feature is monotone non-decreasing, so that it is assumed that

$$v_{jk} - v_{j,k-1} \geq 0, \; j=1,2,...,n, \; k=1,2,...,s_j. \tag{6.5}$$

It is also necessary to normalise the additive utility function by setting the marginal utility of each feature to zero at the lowest value on the scale for this feature and the maximum utility must be constrained to one, i.e.

$$v_{j0} = 0 \quad j=1,2,...,n \tag{6.6a}$$

$$\sum_{j=1}^{n} v_{js_j} = 1 \tag{6.6b}$$

For observation i, i=1,2,...,m let d_i, $d_i \geq 0$, denote the error in the utility of observation i, where $d_i > 0$ if observation i is misclassified and $d_i = 0$ if observation i is correctly classified. In order to prevent observations with utility equal to the cut-off value, a_0, being considered correctly classified,

introduce a rejection interval Δ, $\Delta \geq 0$, where it is assumed that group 1 observations are correctly classified if their utilities are a_0 or more, while group 2 observations are correctly classified if their utilities are $a_0 - \Delta$ or less. The additive utility discriminant analysis LP model is used to determine the values of coefficients v_{jk}, $j=1,2,\ldots,n$, $k=1,2,\ldots,s_j$, and the cut-off value a_0 that minimises the sum of errors of misclassified observations:

$$\text{Minimise} \quad \sum_{i=1}^{m} d_i \tag{6.7a}$$

$$\text{subject to} \quad \sum_{j=1}^{n}\sum_{k=1}^{s_j} a_{ijk} v_{jk} - a_0 + d_i \geq 0 \qquad i \in G_1 \tag{6.7b}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{s_j} a_{ijk} v_{jk} - a_0 - d_i \leq -\Delta \qquad i \in G_2 \tag{6.7c}$$

$$v_{jk} - v_{j,k-1} \geq 0 \qquad j=1,2,\ldots,n, \qquad k=1,\ldots,s_j \tag{6.7d}$$

$$v_{j0} = 0 \qquad j=1,2,\ldots,n, \tag{6.7e}$$

$$\sum_{j=1}^{n} v_{js_j} = 1 \tag{6.7f}$$

$$a_0 \geq 0,\ v_{jk} \geq 0,\ j=1,2,\ldots,n;\ k=1,\ldots,s_j$$

The main difference between model (6.7) and the UTA formulation of Jacquet-Lagreze and Siskos (1982) is that the value of feature j, $j=1,2,\ldots,n$, of observation i, $i=1,2,\ldots,m$, is expressed in terms of the weights defined in equation (6.2). Studies using UTA discriminant analysis models similar to (6.7) have reported good performance (Zopounidis and Doumpos, 1999; Zopounidis and Doumpos 2001; Doumpos et al 2006). However, all these studies either use small datasets or use simulated populations raising questions around the significance of the results.

Model (6.7) can be modified to use maximisation of the number of correctly classified observations as the objective function, resulting in a MIP formulation, or it can be extended to consider more than one goal in a goal programming formulation. These extensions face the same problems identified earlier, i.e. only a limited number of observations can be considered for MIP based models and there are difficulties in assigning appropriate weights in the goal programming models.

6.2.2 The Additive Utility Discriminant Analysis Model

In the UTA discriminant analysis model (6.7), it is assumed that the marginal utility function of each feature is monotone non-decreasing. Glen (2008) developed an MIP model for generating additive utility discriminant functions in which it is only necessary to assume the marginal utility function of each feature is monotone. In this additive utility discriminant analysis (AUDA) approach, let δ_j, j=1,…,n, be a binary variable such that δ_j=1 if the marginal utility of feature j is monotone non-decreasing and δ_j=0 otherwise. The marginal utility of feature j, j=1,2,…,n, must be either monotone non-decreasing or monotone non-increasing, and because $0 \le v_{jk} \le 1$, the requirement corresponding to (6.5) can be expressed as:

$$v_{jk} - v_{j,k-1} - \delta_j \ge -1, \; j=1,2,\ldots,n, \; k=1,2,\ldots,s_j. \tag{6.8a}$$

$$v_{jk} - v_{j,k-1} - \delta_j \le 0, \; j=1,2,\ldots,n, \; k=1,2,\ldots,s_j. \tag{6.8b}$$

Constraints (6.6a) and (6.6b) must also to be modified:

$$v_{j0} + \delta_j \le 1 \quad j=1,2,\ldots,n \tag{6.9a}$$

$$v_{j0} - \delta_j \le 0 \quad j=1,2,\ldots,n \tag{6.9b}$$

$$\sum_{j=1}^{n} (v_{j0} + v_{js_j}) = 1 \tag{6.9c}$$

Using the same notation as for the UTA discriminant analysis model (6.7), the AUDA model can be defined as below:

$$\text{Minimise} \quad \sum_{i=1}^{m} d_i \tag{6.10a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} \sum_{k=1}^{s_j} a_{ijk} v_{jk} - a_0 + d_i \ge 0 \qquad i \in G_1 \tag{6.10b}$$

$$\sum_{j=1}^{n} \sum_{k=1}^{s_j} a_{ijk} v_{jk} - a_0 - d_i \le -\Delta \qquad i \in G_2 \tag{6.10c}$$

$$v_{jk} - v_{j,k-1} - \delta_j \ge -1 \qquad j=1,2,\ldots,n, \; k=1,\ldots,s_j \tag{6.10d}$$

$$v_{jk} - v_{j,k-1} - \delta_j \le 0 \qquad j=1,2,\ldots,n, \; k=1,\ldots,s_j \tag{6.10e}$$

$$v_{j0} + \delta_j \le 1 \qquad j=1,2,\ldots,n \tag{6.10f}$$

$$v_{j0} - \delta_j \leq 0 \qquad j=1,2,\ldots, n \qquad (6.10g)$$

$$\sum_{j=1}^{n} (v_{j0} + v_{js_j}) = 1 \qquad (6.10h)$$

$a_0 \geq 0$, $v_{jk} \geq 0$, $j=1,2,\ldots, n$; $k=1,\ldots,s_j$, $\delta_j=0,1$, $j=1,2,\ldots, n$

Although model (6.10) allows the marginal utility of each feature to be monotone non-decreasing or monotone non-increasing, it can also be used when the form of the utility function is known. For example, when modelling the default rate for a credit card portfolio, it is expected that the higher the income of an applicant, the lower the probability of default. This relationship is then assumed monotone decreasing, and can be pre-specified before running the model. Model (6.10) can also be extended for feature selection (Glen, 2008).

6.2.3 Difficulties in Using Additive Utility Discriminant Analysis Methods

Before applying the additive utility method it is necessary to specify the number of segments in the piecewise linearisation of each feature. In splitting the range of a feature into a number of segments, it is necessary to ensure that there are no null intervals, i.e. intervals containing no observations. It is therefore necessary to be aware of the distribution of the values of each feature. The split of the feature will affect the calculation of the marginal utility. Doumpos and Zopounidis (2001) presented a five-stage heuristic for defining the number of intervals for each feature, but this heuristic assumes that there are sufficient observations in every interval and it does not consider how to treat binary and categorical variables.

The problem of splitting a feature into intervals and transforming the feature has been considered in credit scoring, e.g. Anderson (2007), by introducing binary variables based on the chi square statistic or the weight of evidence (WoE), as outlined in Chapter 3. Alternatively, i.e. instead of introducing binary variables, it is possible to replace the value of a feature by its WoE. The main advantage of this approach is that scores assigned to the attributes will reflect the ranking of their bad/good odds (e.g. Thomas et al, 2002), provided that the coefficient assigned to the feature is positive. As a result, transforming raw data using WoE will achieve the required monotonicity of the features when applying additive utility methods. Moreover, by using WoE it is possible to transform categorical variables to

continuous variables. In the following section the performance of additive utility methods is examined after first performing the WoE transformation to the data.

6.3 Experimental Studies

The performance of logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), neural networks (NNs), MSD without balancing constraints, MSD with balancing constraint (see Chapter 5), additive utility discriminant analysis (AUDA) and the additive utility discriminant analysis using WoE transformation (AUDA-WoE), were compared on three datasets used in the experimental studies in Chapter 3, i.e. the German, SPSS and Greek datasets. These datasets were chosen due to the mixture of categorical and continuous features and the size of the dataset. Open source MatlabArsenal toolbox (Yan, 2006) was used to train and evaluate the QDA and LDA classifiers. Weka open source software (Witten and Eibe, 2005) was used to train the neural networks, and for logistic regression. Xpress-MP software was used for the MSD, AUDA and AUDA-WoE models. Ten randomisations of the datasets were used, with 80% of each randomization used for model development and the remaining 20% used for performance assessment. The performance measures that were used in the previous experimental studies, i.e. overall accuracy, accuracy in bad and good classes, and area under the ROC curve (AUC), were used in this study and the average values for each measure over the ten randomisations estimated.

6.3.1 German Dataset

The results for the German dataset are presented in Table 6.1.

Method	Accuracy (%)			AUC
	Overall	Bad	Good	(%)
Logistic Regression	77	54	87	81
LDA	73	75	72	81
QDA	73	70	74	79
NN	76	59	84	79
MSD	74	53	81	77
MSD - Balancing	74	76	73	80
AUDA	65	10	90	75
AUDA-WoE	74	22	96	79

Table 6.1: Performance on the German Dataset

From Table 6.1, it can be seen that AUDA-WoE performs better than the additive utility method on the German dataset, with AUDA-WoE achieving 74% overall accuracy compared to 65% for AUDA. The best overall accuracy is achieved by logistic regression and the worst by AUDA. The best performance for the bad class is achieved by MSD with balancing constraints and the worst performance by additive utility method. The best performance for the good class is achieved by AUDA-WoE and the worst by LDA. It appears that AUDA-WoE is biased in favour of the good class as only 22% of the bad class observations are correctly classified. Only the LDA and QDA methods and MSD with balancing constraints achieved balanced results with almost equal accuracies for both classes. Paired *t*-tests were applied to examine the significance of the results. Analytical results of the *t*–tests are included in Appendix C. Under the overall accuracy metric, logistic regression was significantly better than AUDA-WoE, LDA, QDA and UTA discriminant analysis. Under the AUC measure there is no significant difference between the methods.

6.3.2 SPSS Dataset

The results for the SPSS dataset are presented in Table 6.2.

Method	Accuracy (%)			AUC
	Overall	Bad	Good	(%)
Logistic regression	76	61	85	83
LDA	75	75	75	83
QDA	73	70	75	82
NN	73	58	81	79
MSD	71	77	60	77
MSD - Balancing	73	80	64	79
AUDA	72	26	98	83
AUDA-WoE	88	82	91	94

Table 6.2: Performance on the SPSS Dataset

From Table 6.2 it can be seen that overall, AUDA-WoE performed better than the other methods. AUDA-WoE achieved 88% overall accuracy, with the MSD model having the poorest performance under this criterion. The best accuracy for the bad class was achieved by AUDA-WoE and the worst by AUDA. The best performance for the good class was achieved by AUDA and the worst by QDA. However, although AUDA achieved the best results for good class accuracy, it performed poorly in the bad, i.e. the minority, class. AUDA also had the largest discrepancy in performance between the good and bad classes, with all the other methods achieving either balanced or reasonably balanced results. Under the AUC metric, the best performance was achieved by AUDA-WoE and the worst by the MSD model. Paired *t*-tests were applied to examine the significance of the results and analytical results are included in Appendix C. Under the overall accuracy and AUC measures, AUDA-WoE was significantly better than logistic regression, LDA, QDA and AUDA.

6.3.3 Greek Dataset

The results for the Greek dataset are presented in Table 6.3.

Method	Accuracy (%)			AUC
	Overall	Bad	Good	(%)
Logistic Regression	79	3	99	67
LDA	63	56	64	68
QDA	63	56	65	67
NN	76	21	91	63
MSD	70	10	99	60
MSD - Balancing	56	60	51	63
AUDA	70	15	90	58
AUDA-WoE	74	30	86	66

Table 6.3: Performance on the Greek Dataset

The Greek dataset is the most imbalanced, with the bad class accounting for 20% of the population while the good class consists the remaining 80%. The results for this dataset are dominated by the majority class. From the results in Table 6.3, it is clear that the performance of the additive utility method is improved when the WoE transformation is used. It is also worth noting that AUDA-WoE performs better than logistic regression which fails to classify correctly more than 3% of the minority class. Paired *t*-tests were applied to examine the significance of the results (see Appendix C). Logistic regression was significantly better than AUDA-WoE, LDA, QDA and additive utility method under the overall accuracy criterion, but, as with the other two datasets, AUDA-WoE achieved better results than the additive utility method.

The results from these three datasets show that no single method outperforms the other methods on all datasets, confirming that it is important to consider a number of methods in developing classification models

6.4 Applications of Ordinal Classification in Credit Scoring

Credit scoring is concerned with predicting the correct status of an applicant for credit. Usually the class variable is treated as nominal and the score is only used to assign the applicant to one of the two classes. There are, however, decisions in credit scoring in which the distance of an applicant's score from the cut-off value or the relative position of each score from the cut-off value is important. By considering the distance of each score from the cut-off value and the distance between two scores, the

scorecard can generate ordinal measures. The use of this category of scorecards in different decisions has been emphasised by Thomas et al (2001) who noted that decisions such as what percentage of applicants to accept or reject is related to the ordinal nature of the score. Also in decisions which are related to customisable characteristics of the products it is essential to use an ordinal scorecard. For example, it is possible to set the value of the interest or the annual fee paid by an applicant, according to the score of an applicant and the distance from the cut-off value.

6.4.1 Calibration

The use of credit scoring in different stages of the credit cycle such as originations, i.e. application scoring, accounts management, i.e. behavioural scoring, and collections, i.e. collections scoring, was described in Chapter 3. In all these problems, scoring is a two–class classification problem, e.g. to accept or to reject an application, to increase the credit limit or not. However, when developing or monitoring the scorecard it is essential to satisfy specific properties for a scorecard (e.g. Anderson, 2007), such as the properties found in a survey by Thomas et al (2001). For example, only positive scores or positive feature weights may be required by the users of scorecards, so that they can easily explain their decision to an applicant. It may be also necessary for the difference between the scores to have a constant meaning across the range of the score and for continuous characteristics to have monotone good/bad odds. Mays (2005) noted that it is common in credit scoring to calibrate the results of logistic regression. It is also common to add some base points in order to move the score in a specific score interval. Thomas et al (2001) proposed a mathematical programming model that can be used for recalibration of the scorecard and satisfy other properties. Although properties can be incorporated using ad hoc techniques, Thomas et al (2001) note that this could create contradictions, e.g. trying to have reference scores with specific marginal good/bad odds, might create negative weights for some features.

These properties can be incorporated in a scorecard with the help of a linear programming model. In this study, the focus will be solely on the ordering of the observations and how this can be achieved through the use of an MP model. The model proposed by Thomas et al (2001) is described below using a notation similar to that used earlier. Consider an ordinal

classification problem with m ranked observations in increasing order, described by the values of n features, with X_{ij} representing the value of feature j in observation i. The LP model is used to determine the function, defined by cutoff value a_0 and the coefficient a_j, j=1,2,….,n, of feature j, that minimizes the errors in the ranking of the observations. Defining e_{ir} as the amount that should be added to the score generated for observation r, r>i, i=1,2,…,m–1, r=2,3,…,m, to ensure that score for observation r is at least as large as score for observation, the model proposed by Thomas et al (2001) can be expressed as:

$$\text{Minimise} \qquad \sum_{i,r} e_{ir} \tag{6.11a}$$

$$\text{subject to} \qquad a_0 + \sum_j a_j X_{ij} \le a_0 + \sum_j a_j X_{i+1j} - e_{ir} \quad 1 \le i < r \le m \tag{6.11b}$$

$a_j \ge 0$, j=0,1,2,....,n; $e_{ir} \ge 0$, i=1,2,....,m-1, r=2,3,…,m, with i<r.

There are several issues with model (6.11). Firstly, the constant term, a_0, which corresponds to a constant that should be added to all scores to ensure that all scores are positive, cannot be determined by this model. A constant term of this form could be determined by adding an additional set of constraints to ensure that all scores are positive, but this would increase the size of the model. Secondly, this model is not normalised. Thomas et al (2001) suggest adding a constraint that requires the coefficients a_j, j=1,2,….,n, to sum to a constant, e.g. 100, which is appropriate if the feature weights must be non-negative. Thirdly, as model (6.11) has n+1 a_j variables, m(m–1)/2 e_{ir} variables and m(m–1)/2 constraints, it is intractable for problems of even moderate size. Thomas et al (2001) suggest an approximate model in which only adjacent observations are compared and which requires fewer variables and constraints. However, by defining the "error" variables in a different way, an exact model can be developed for generating the ranking function.

6.4.2 New Ordinal LP model

The LP presented in this section is based on the same principles as the LP model proposed by Thomas et al (2001) for calibrating a scorecard with m ranked observations. Define $d_i^-, d_i^- \ge 0,$ and $d_i^+, d_i^+ \ge 0,$ as, respectively, the amount that must be subtracted from or added to the score for observation i, i=1,2,….,m, to preserve the ranking of observations. The LP

model for determining the coefficient a_j of feature j, j=1,2,….,n, in the ranking function, where the function is normalised by summing the coefficients to a constant, e.g. 1, is then:

$$\text{Minimise} \qquad \sum_{i=1}^{m}(d_i^- + d_i^+) \tag{6.12a}$$

$$\text{subject to} \qquad \sum_j a_j X_{ij} - d_i^- + d_i^+ \le \sum_j a_j X_{i+1j} - d_{i+1}^- + d_{i+1}^+ \quad 1 \le i < m \tag{6.12b}$$

$$\sum_j a_j = 1 \tag{6.12c}$$

$$a_j \ge 0,\ j=0,1,2,....,n;\ d_i^-, d_i^+ \ge 0,\ i=1,2,\ldots,m$$

In model (6.12) it is assumed that the ranking function's feature coefficients, a_j, j=1,2,….,n, must be non-negative. Ranking functions in which these coefficients may take negative values can be generated by representing free variables a_j, j=1,2,....,n, by a pair of non-negative variables, a_j^+ and a_j^- as in (2.2), and substituting (2.3) for the normalisation constraint (6.12*c*).

The ordinal MP model (6.12) was tested on a small dataset consisting of 25 ranked observations of road projects, (e.g. Beuthe and Scannella, 2001). These road projects had been ranked by experts using 6 features for each project. The rankings generated by model (6.12) were compared with the expert rankings using the leave-one-out (LOO) approach as the dataset consists of a small number of observations. The performance of the ordinal MP model was then evaluated using Kendall's τ and Spearman's rank correlation coefficient, ρ (e.g. Salkind, 2007). For comparison, rankings were also generated using ordinal logistic regression, ordinal probit and ordinal negative logistic (e.g. Salkind, 2007) and Kendall's τ and Spearman's rank correlation coefficient, ρ, determined, as shown in Table 6.4. Although, because of the small size of the dataset, it is not possible to draw wide-ranging conclusions from the results in Table 6.4, these results are included in order to give some indications about the potential power of the different methods.

Method	τ	ρ
Ordinal MP	0.871	0.969
Ordinal Logistic Regression	0.890	0.972
Ordinal Probit	0.867	0.966
Ordinal Negative Logistic	0.881	0.973

Table 6.4: Statistics for the Different Models.

From Table 6.4, it can be seen that although ordinal logistic regression performs best according to Kendall's τ, and ordinal negative logistic regression performs best according to Spearman's ρ, the ordinal MP approach performs reasonably well according to both metrics. Further research using larger datasets is clearly required to evaluate the ordinal MP model (6.12). It should also be noted that model (6.12) can be extended to incorporate other conditions that may be required in calibrating or recalibrating scorecards (e.g. Thomas et al, 2001).

6.5 Summary

Various methods have been proposed for ordinal classification problems, with applications in job evaluation, financial management, stock evaluation and calibration/recalibration of scorecards. The statistical methods that have been proposed for this type of problems, e.g. ordinal logistic regression, either make assumptions about the distributions of the underlying populations or use computationally intensive algorithms to obtain solutions. MP-based methods have also been proposed for ordinal classification problems. The most established of these MP approaches are additive utility methods in which an LP model is used to generate a piecewise linear utility function from a weak order preference ranking defined by the user on a subset of reference observations. These additive utility methods can also be modified to generate non-linear discriminant functions, but methods based on the original UTA method have only been applied to relatively small datasets and have not addressed the difficulties associated with binary features.

In this chapter, a general additive utility discriminant analysis model has been extended to deal with binary features. This additive utility discriminant analysis model has been applied to three credit scoring datasets and the

performance compared with other methods for generating binary classifiers. Although different methods for developing classifiers performed best on these datasets, the results suggest that the additive utility discriminant analysis model is a useful tool for classifier development.

Some of the issues involved in using an LP-based model proposed for calibrating or recalibrating scorecards have also been considered in this chapter and a revised LP model has been outlined. This new LP model has applied to a small dataset. The results from this small dataset indicate that further research on larger datasets would be appropriate.

Chapter 7

7. Conclusions

7.1 Thesis summary

This thesis has investigated issues related to the application of MP methods to the binary classification problem, with an emphasis on credit scoring applications. Credit scoring is a binary classification problem that is of high importance to financial institutions that provide credit as lenders need to be able to predict if an applicant for credit is likely to repay or default. Previous experimental studies with MP discriminant analysis models have used inappropriate normalisation constraints and/or small datasets or did not use the data transformations widely applied in practice. This thesis has investigated the performance of MP models using appropriate normalisation constraints and data transformations on real datasets. In addition, other important issues have been discussed in relation to the application of MP to the binary classification problem. In particular, this thesis has considered the choice of appropriate features for inclusion in the classification model, the performance of MP discriminant analysis models on imbalanced datasets, the development of non-linear classifiers based on MP methods, and the application of MP models to ordinal problems.

The main methods that can be used to construct a classifier, together with their strengths and weaknesses, were outlined in Chapter 2. The methods most widely used in practice for classification model development are statistical techniques (e.g. linear regression, linear and quadratic discriminant analysis, logistic regression, classification trees) and machine learning methods (e.g. neural networks, expert systems, nearest neighbour methods, support vector machines). MP discriminant analysis models can also be used to develop classifiers, but are not as widely used as statistical and machine learning approaches. The simplest MP methods use LP models to generate a discriminant function that optimises a metric based on the deviations of misclassified observations from the discriminant function, with objectives such as minimisation of the sum of deviations (MSD), i.e. the l_1-norm, or maximisation of the minimum deviation (MMD), i.e. the l_∞-norm. One of the advantages of MP methods for developing classifiers is that classification accuracy can be used as the objective function to maximise the

number of correctly classified training sample observations (or minimise the number of misclassified observations), i.e. the l_0-norm, by using an MIP formulation. However, the size of problem to which this MIP approach can be applied is restricted because a binary variable must be associated with each training sample observation. Non-linear programming methods can be used to develop classifiers based on the l_p-norm for values of p other than 0, 1 and ∞, although there are computational difficulties in solving these non-linear programming models. MP methods each generate a linear discriminant function, but non-linear discriminant functions can be generated by first transforming the original features or by additive utility MP discriminant analysis models that generate piecewise linear discriminant functions.

Chapter 3 considered the credit scoring problem and methods for developing scorecards. The emphasis of this thesis is on application scoring, i.e. assessment of new applications for credit, but other types of scoring, e.g. behavioural scoring and collection scoring, are noted. The most common uses of application scoring relate to personal loans and credit cards, but these methods can also be applied to portfolios of small business loans, which can be an important element in determining a bank's capital requirements, as recognised in the Basel II regulations. Chapter 3 concluded with a benchmarking study comparing the performance of classifiers developed by different techniques using six datasets representing different experimental conditions in terms of their size and origin, with features generated by the WoE transformation. The results from this benchmarking study confirm the results from previous studies, e.g. Baesens (2005), about the performance of the classifiers and that, in particular, there is no single method that outperforms all other methods under all data conditions. The MSD model was included in this comparative study, but all WoE-generated features were used for classifier development, whereas only a limited set of features would be used in practice.

The development of parsimonious classifiers requires efficient and effective methods for feature selection. Traditional methods of feature selection were outlined in Chapter 4. Ideally, features should be selected in terms of their impact on classification accuracy, but traditional methods use proxies for classification accuracy in the selection process. Features are selected on the basis of their contribution to classification accuracy in the feature selection extension of the MIP discriminant analysis model for

maximising classification accuracy. However, this MIP approach can only be applied to discriminant problems with a relatively small number of observations as a binary variable is required for each observation. Two heuristic methods of feature selection based on the MIP model for maximising classification accuracy in two-group discriminant problems were proposed in Chapter 4. The number of features to be selected is specified in advance in one heuristic, but not specified in advance in the other. The two heuristics were applied to three credit scoring datasets, with the heuristic involving a specified number of features generally outperforming the heuristic in which the number of features is not specified. The performance of classifiers developed by the heuristic involving a specified number of features was then compared with classifiers developed by four other feature selection methods, and although none of the five feature selection methods consistently outperformed the other methods, the MIP based heuristic generally performed well.

In Chapter 5 the difficulties associated with imbalanced binary classification problems, i.e. problems with many more observations in one class than in the other, were considered. Imbalanced datasets, which are often found in credit scoring and fraud detection applications, can lead to the production of classifiers that are dominated by the majority class, so that in extreme cases all observations are assigned to the majority class. A common method for dealing with imbalanced datasets involves pre-processing the data to produce a more balanced dataset by either under-sampling the majority class or over-sampling the minority class, but pre-processing the data can bias classifier performance. Alternatively, the costs associated with misclassifying observations can be considered in developing classifiers, but in practice it is often difficult to determine misclassification costs. It was shown in Chapter 5 that MP discriminant analysis models can be extended to balance misclassification metrics across the classes either by modifying the objective function or by incorporating additional constraints, so that it is not necessary to pre-process the data or identify misclassification costs. These extended MP models for developing classifiers were applied to four imbalanced datasets from financial institutions and were found to produce classifiers with balanced performance across the two classes.

The use of MP in generating non-linear discriminant functions and ordinal classification was outlined in Chapter 6. The additive utility UTA method (Jacquet-Lagrèse and Siskos, 1982) uses an LP model to produce an

additive utility ranking function from a weak-order preference ranking of a training sample of observations, but it is assumed that the marginal utility of each feature is monotone non-decreasing. This additive utility approach can be modified so that it is only necessary to assume that the marginal utility of each feature is monotone. A further extension allows an additive utility approach to be used to generate piecewise linear representations of non-linear discriminant functions, but this approach, which cannot deal with binary features, has only been applied to relatively small discriminant problems. In Chapter 6 it was shown that binary features can be accommodated in additive utility discriminant analysis by using the WoE data transformation, and this approach was applied to three credit scoring datasets. The simplest applications of credit scoring are concerned with accepting or rejecting applicants for credit, with applicants classified as good, i.e. unlikely to default, or bad, i.e. likely to default. Credit scoring is also used for ordinal classification, e.g. ranking applicants by risk of default, or for estimating the probability of default. MP approaches have been proposed to calibrate scorecards so that, as far as possible, scores have specified properties (e.g. Thomas et al, 2001). A simple LP model for ranking applicants for credit is presented in Chapter 6, but because an appropriate credit scoring dataset was not available, the use of this model was demonstrated on a small ordered dataset from another domain.

This thesis has demonstrated that MP discriminant analysis models can be used to develop linear and non-linear classifiers from large datasets of the type encountered in credit scoring applications. It has also been shown that MP based heuristic methods can be used to select features on the basis of their impact on classification accuracy in order to develop parsimonious classifiers. Methods for extending MP discriminant analysis models to deal with imbalanced datasets have also been developed and tested in this thesis. These techniques have been used to develop classifiers from a range of datasets and the performance of these methods compared with classifiers developed by statistical and machine learning methods. Although no single method of classifier development has been found to outperform all other methods under all data conditions, the results show that MP methods can be a valuable tool in this area.

7.2 Limitations of the research

The experimental studies in this thesis were based solely on application scoring data. The use of MP-based techniques on a number of similar problems, e.g. behavioural scoring and collection scoring, has therefore not been considered. Moreover, the performance of two-stage MP-based models was not tested even if these models seem to be a good alternative for classification problems. In addition, as only a very limited number of methods for imbalanced datasets was tested, it would be worth including more methods in the comparison studies. In Chapter 6 only a small (and irrelevant) dataset was used for testing the new ranking classifier suggested. It would be worth obtaining a larger dataset for testing the performance of this classifier.

7.3 Issues for further research

7.3.1 Application of MP Methods to Peer-to-Peer Lending

Peer-to-peer (P2P) lending allows individuals to lend money to other individuals through P2P websites. P2P lending has become popular due to the economic crisis and reduced lending by banks to individuals and small businesses. The website operator generates income through a fee charged to users, with lenders earning interest from borrowers. The advantages to borrowers are that interest rates and initiation charges may be lower than available elsewhere and the absence of early repayment fees. However, as borrowers who default cannot be easily pursued, it is very important to have methods for rapidly predicting the behaviour of applicants using every possible piece of information. MP methods may be particularly suitable for this problem as special relationships and qualitative characteristics can be incorporated into the classifier.

7.3.2 Using MP Methods in Combination with Other Techniques

This thesis has examined the performance of MP methods individually. MP methods could also be used in combination with techniques from statistics and machine learning to improve performance and increase flexibility. For example, it was noted in Chapter 2 that neural networks do not offer explanation about how a specific decision is reached. By using an MP model in combination with a neural network it may be possible to improve performance and provide explanation.

7.3.3 Application of MP Methods to Collection Data

Collection scoring is closely related to use of behavioural scoring (Lewis, 1992) as it uses a behavioural score to set up a strategy for recovering debts from, for example, 60+ days-past-due customers. The additive utility discriminant analysis model may be appropriate for this application.

7.3.4 Use of Macro-Economic Factors in MP-Based Methods

Most techniques for developing scorecards use data which relate to a specific period. As a result, these data may not be appropriate for predicting the behaviour of accounts under different economic conditions. For instance, loan accounts opened during a recession may be expected to behave differently from accounts opened during a period of economic growth. For credit scores to accurately predict the probability of default, a credit scorecard should not be static but should reflect changes in the economy (e.g. Thomas et al, 2005). One way to achieve this is through the inclusion of macro-economic factors in a credit scoring model (e.g. Crook and Belloti, 2007; Wendling and Goncalves, 2007). The use of macro-economic in MP methods for developing scorecards has not been investigated and is therefore an area for future research.

References

Abad, P.L. and Banks, W.J. (1993). New LP based heuristics for the classification problem, *European Journal of Operational Research*, 67, 88-100.

Adams, N.M. and Hand, DJ (1999). Comparing classifiers when the misallocations costs are uncertain. *Pattern Recognition* 32, 1139-1147.

Akkoc, S. (2012). An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. *European Journal of Operational Research*. 222(1), 168-178.

Altman, E.I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23 (September), 589-609.

Altman, E.I., and Sabato, G. (2006). Effects of the New Basel Capital Accord on Bank Capital Requirements for SMEs. *Journal of Financial Services Research* 28, 1/2, 15-42.

Anderson, J.A. (1969). Constrained discrimination between k populations. *Journal of the Royal Statistical Society*. Series B, 31, 123-139.

Anderson, R. (2007). *The credit scoring toolkit: Theory and practice for retail credit risk management and decision automation*. OUP Oxford: London.

Andreeva, G. (2004). Credit risk in the context of European integration: assessing the possibility of Pan-European scoring, *Unpublished PhD thesis, University of Edinburgh*, UK.

Apilado, V.P., Warner, D.C. and Dauten, J.J. (1974). Evaluative techniques in consumer finance, *Journal of Financial and Quantitative Analysis*, 9, 275-283.

Arminger, G., Enache, D., Bonne, T., (1997). Analyzing credit risk data: a comparison of logistic discriminant classification tree analysis and feed-forward networks. *Computational Statistics*, 12, 293-310.

Baesens, B. (2003). Developing intelligent systems for credit scoring using machine learning techniques. Leuven, K.U. Leuven, Faculteit Economische en toegepaste economische wetenschappen, 264 pp.

Bajgier, S.M. and Hill, A.V. (1982). An experimental comparison of statistical and linear programming approaches to the discriminant problem. *Decision Sciences*, 13, 604-618.

Banks, W.J. and Abad, P.L. (1994). On the performance of linear programming heuristics applied on a quadratic transformation in the

classification problem. *European Journal of Operational Research*. 74, 23-28.

Barlow, R. E., Bartholomew, D.J., Bremner, J. M., and Brunk, H. D. (1972). *Statistical inference under order restrictions; the theory and application of isotonic regression*. New York: Wiley.

Basel Committee on Banking Supervision (2006a). International convergence of capital measurement and capital standards: a revised framework, Bank for International Settlements. http://www.bis.org

Basel Committee on Banking Supervision (2006b). Sound credit risk assessment and valuation for loans, *Bank for International Settlements*. http://www.bis.org

Bauer, E. and Kohavi, R. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36, 105-139.

Bellotti, T., and Crook, J. (2007). Credit scoring with macroeconomic variables using survival analysis. *Credit scoring and credit control conference X*.

Bensic, M., Sarlija, N. and Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. *Intelligent Systems in Accounting, Finance and Management*, 13, 133-150.

Beuthe, M and Scannella, G. (2001). Comparative analysis of UTA multicriteria methods. *European Journal of Operational Reseasrch*, 130, 246-262.

Blum, A.L. and Langley, P. (1997). Selection of relevant features and examples in machine learning, *Artificial Intelligence*, 97, 245-271.

Bolton, R.J. and Hand, D.J. (2002). Statistical fraud detection: A review. *Statistical Science*, 17(3), 235-255.

Borooah, V. K. (2001). Logit and probit: Ordered and multinomial models. *In Quantitative Applications in the Social Sciences*. Sage University paper series: London.

Boyle, M. Crook, J. Hamilton, R. and Thomas, L.C. (1992). Methods for credit scoring applied to slow payers. In Thomas, L.C., Crook, J.N. and Edelman, D. B.. Credit scoring and credit control, Oxford University Press, Oxford, pp 75-90.

Bradley, A.P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition* 30, 1145-1159.

Bradley P.S., Mangasarian O.L. and Street W.N. (1997). Feature Selection via Mathematical Programming, *INFORMS Journal on Computing*, 10, 209-217.

Brause, R., Langsdorf, T. and Hepp, M.: *Credit Card Fraud Detection by Adaptive Neural Data Mining*, J.W.Goethe-University, Comp. Sc. Dep., Report 7/99, Frankfurt, Germany (1999), also by http://www.cs.uni.frankfurt.de/fbreports/07.99.ps.gz

Bredensteiner E.J. and Bennett K.P. (1998). Feature minimisation within decision trees, *Computational Optimizations and Applications*, 10,111-126.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). Classification and regression trees, Wadsworth, Pacific Grove, CA.

Breiman, L. (1996). Bagging Predictors, *Machine Learning*, 24, 123-140.

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistics*, 26, 801–849.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.

Brown, I. and Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3), 3446-3453.

Bryant, K. (2001), ALEES: an agricultural loan evaluation expert system, *Expert Systems with Applications*, Vol. 21 pp.75-85.

Bugera, V., Konno, H. and Uryasev, S. (2002). Credit Cards Scoring with quadratic utility functions. *Journal of Multi-criteria decision analysis*. 11,197-211.

Caouette, J.B., Altman, E.I. and Narayanan, P. (1998). Managing credit risk, the next great financial challenge. *Wiley Frontiers in Finance*: New York.

Capon, N. (1982). Credit Scoring Systems: A Critical Analysis, *Journal of Marketing*, 46 (Spring 1982), 82-91.

Cavalier, T.M., Ignizio, J.P. and Soyster, A.L. (1989). Discriminant analysis via mathematical programming: on certain problems and their causes. *Computers and Operations Research*, 16 (4), 353-362.

Charnes, A., Cooper, W.W., and Ferguson, R.O. (1977). Optimal estimation of executive compensation by linear programming. *Management Science*, 2, 138-155.

Chatterjee, S., and Barcun, S. (1970). A Nonparametric Approach to Credit Screening. *Journal of the American Statistical Association*, 65, 150-154.

Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P. (2002), SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research (JAIR),* 16, 321-357.

Chen, M.C., and Huang, S.H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications*, 24, 433-441.

Chen, F.L., Li, F-C.(2010). Combination of feature selection approaches with SVM in credit scoring. *Expert Systems with Applications*, 37(7), 4902-4909.

Choo, U.C. and Wedley, W.C. (1985). Optimal criterion weights in repetitive Multicriteria decision-making. *Journal of the operational research society*, 36,983-992.

Cramer, J. S. (2004). Scoring bank loans that may go wrong: A case study. *Statistica Neerlandica*, 58(3), 365–380.

Cristianini, N., and Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, Cambridge, New York.

Daskalaki, S., Kopanas, I. and Avouris, N. (2006). Evaluation of classifiers for an uneven class distribution problem. *Applied Artificial Intelligence: An International Journal,* 20 (5), 381-417.

Dash Associates (2006). XPRESS-MP User guide and reference manual. *Dash Associates*, Blisworth, England.

De Andres, J., Lorca, P. de Cos Juez, F.J., and Sanchez-Lasheras, F. (2011). Bankruptcy forecasting: A hybrid approach using Fuzzy c-means clustering with multivariate adaptive regression splines (MARS). *Expert Systems with Applications*, 38(3), 1866-1875.

Desai, V.S., Conway, D.G., Crook, J.N. and Overstreet, G.A. (1997). Credit scoring in the credit union environment using neural networks and genetic algorithms. *IMA Journal of Mathematics Applied in Business and Industry*, 8, 323-346.

Dietterich, T.G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation* 10, 1895-1923.

Doumpos, M. and Zopounidis, C. (2001). Developing sorting models using preference disaggregation analysis: An experimental investigation, in Zopoundis, C., Pardalos, P.M. and Baourakis, G. (Editors), *Fuzzy sets in Management, Economics and Marketing*, World Scientific, London, UK, pp 51-67.

Doumpos, M, Zanakis, S.H., and Zopounidis, C. (2001). Multicriteria preference disaggregation for classification problems with an application to global investing risk, *Decision Sciences*, 32, 2, 333-385.

Dreiseitl, S., and Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, 35, 352-359.

Duarte Silva, A.P. and Stam, A. (1994). Second order mathematical programming formulations for discriminant analysis. *European Journal of Operational Research*, 72,4-22.

Duda, R.O., Hart, P.E. and Stork, D. G. (2001). *Pattern Classification*. Wiley.

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3, 32–57.

Edmister, R. O. (1972). An empirical test of financial ratio analysis for small business failure prediction. *Journal of Financial and Quantitative Analysis*, 7, 1477-1493.

Efron, B. (1981). Nonparametric estimates of standard error: The jackknife, the bootstrap and other methods. *Biometrika*, 68 (3), 589-599.

Eisenbeis, R.A. (1977). Pitfalls in the application of discriminant analysis in business, finance, and economics. *Journal of Finance*, (3), 875-900.

Eisenbeis, R.A.(1978). Problems in applying discriminant analysis in credit scoring models. *Journal of Banking and Finance*, (2) 205-219.

Eisenbeis, R.A. (1996). Recent developments in the application of credit scoring techniques to the evaluation of commercial loans. *IMA Journal of Mathematics Applied in Business and Industry* 7, 271-290.

Eisenbeis, R.A. and Avery, R.B. (1972). *Discriminant Analysis and Classification Procedures: Theory and Applications*. Lexington, Mass: D.C. Heath and Co.

Eitrich, T., Kless, A., Druska, C., Meyer, W. and Grotendorst, J. (2007). Classification of Highly Unbalanced CYP450 Data of Drugs Using Cost Sensitive Machine Learning Techniques. *Journal of Chemical Information and Modelling*, 47, 92-103.

Erenguc, S.S. and Koehler, G.J. (1990). Survey of mathematical programming models and experimental results for linear discriminant analysis, *Managerial and Decision Economics*, 11,215-225.

Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research*, 210(2), 368-378.

Fisher, R.A. (1936). The use of multiple measurements in taxonomy problems, *Annals of Eugenics*, 7, 179-188.

Fogarty, T.C., and Ireson, N.S. (1994). Evolving Bayesian classifiers for credit control-a comparison with other machine learning methods. *IMA Journal of Mathematics Applied in Business and Industry*, 5, 63-75.

Frank E. and Hall M. (2001). A simple approach to oridinal classification. In *Proceedings of the European Conference on Machine Learning*, pages 145-165.

Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Freed, N., and Glover, N (1981a). A linear programming approach to the discriminant problem, *Decision Sciences*, 12, 68-74.

Freed, N., and Glover, N (1981b). Simple but powerful goal programming models for discriminant problems, *European Journal of Operational Research*, 7, 44-60.

Freed, N., and Glover, N (1982). Linear programming and statistical discrimination- The LP side. *Decision Sciences*, 13, 172-175.

Freed, N., and Glover, N (1986a). Evaluating alternative linear programming models to solve the two-group discriminant problem. *Decision Sciences*, 17, 151-162.

Freed, N., and Glover, N (1986b). Resolving certain difficulties and improving the classification power of LP discriminant analysis formulations. *Decision Sciences*, 17, 589-595.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19, 1–141

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.

Gallagher, R.J., Lee, E.K. and Patterson, D.A. (1997). Constrained discriminant analysis via 0/1 mixed integer programming. *Annals of Operations Research*, 74, 65-88.

Gehrlein, W.V. (1986). General mathematical programming formulations for the statistical classification problem, *Operations Research Letters*, 5,299-304.

Gehrelein, W.V. and Wagner, B.J. (1997). A two-stage least cost credit scoring model. *Annals of Operations Research*, 74,159-171.

Glen, JJ. (1999). Integer programming methods for normalization and variable selection in mathematical programming discriminant analysis models. *Journal of the Operational Research Society*, 50,1043-1053.

Glen, JJ. (2001). Classification accuracy in discriminant analysis. *Journal of the Operational Research Society*, 52, 328-339.

Glen, J.J. (2003). An iterative mixed integer programming method for classification accuracy maximizing discriminant analysis. *Computers & Operations Research*, 30, 181-198.

Glen, J.J. (2005). Mathematical programming models for piecewise-linear discriminant analysis. *Journal of the Operational Research Society*, 56, 331-341.

Glen, J.J. (2006). A comparison of standard and two-stage mathematical programming discriminant analysis models. *European Journal of Operational Research*, 171, 496-515.

Glen, J.J.(2008). An additive utility mixed integer programming model for nonlinear discriminant analysis. *Journal of Operational Research Society*, 59, 1492-1505.

Glorfeld, L.W. and Gaither, N. (1982). On using Linear Programming in Discriminant problems, *Decision Sciences*, 13, 167-171.

Glorfeld, L.W. and Hardgrave, B.C. (1996). An improved method for developing neural networks: The case of evaluating commercial loan creditworthiness, *Computers and Operations Research*, 23 (10), 933-944.

Glover, F. Keene, S. and Duea, B. (1988). A new class of models for the discriminant problem. *Decision Sciences,* 19, 269-280.

Glover, F. (1990). Improved linear programming models for discriminant analysis, *Decision Sciences*, 21,771-785.

Glover, F. (1998). A template for scatter search and path relinking. In J.-K. Hao, E.Lutton, E. Ronald, M. Schoenauer, & D. Snyers (Eds.), *Artificial evolution. Lecture notes in computer science* (1363, pp. 125–137). Springer.

Glover, F. and Better, M. (2007). Improved classification and discrimination by successive hyperplane and multi-hyperplane separation. Draft available in http://www.opttek.com/News/pdfs/Improved%20Classification.pdf.

Gordon, A.D. (1981). Classification: Methods for the Exploratory Analysis of Multivariate Data. Chapman and Hall: London.

Grablowsky, B.J. and Talley, W.K. (1981) Probit and discriminant functions for classifying credit applicants: a comparison. *Journal of Economics and Business*, 33, 254-261.

Grinold, R.C. (1972). Mathematical programming methods for pattern classification, *Management Sciences*, 19, 272-289.

Guo, Z., Lu, L., Xi, S., and Sun, F. (2009). An effective dimension reduction approach to Chinese document classification using genetic algorithm. *Lecture Notes in Computer Science*, 480-489. Springer-Verlag Berlin Heidelberg

Guyon I., and Elisseeff A., (2003). An introduction to variable and feature selection, *Journal of Machine Learning Research*, 3,1157-1182.

Hand D.J. (1981). *Discrimination and Classification*. Wiley: Chichester.

Hand D.J. (1997). *Construction and assessment of classification rules*. Wiley: Chichester.

Hand, D.J. and Henley, W.E. (1997). Statistical classification methods in consumer credit. *Journal of the Royal Statistical Society*, Series A 160, 523-541.

Hand, D.J. and Jacka, S.D. (eds), (1998). *Statistics in Finance*, Edward Arnold, London.

Hand D.J. and Vinciotti, V. (2003). Choosing *k* for two-class nearest neighbour classifiers with unbalanced classes. *Pattern Recognition Letters*, 24: 1555-1562.

Henley, W.E. (1996). Statistical aspects of credit scoring, *PhD thesis*, Open University.

Henley, W.E. and Hand, D.J. (1996). Construction of a k-NN credit scoring system. *IMA Journal of Mathematics Applied in Business and Industry*, 8, 305-321.

Hosmer, D.W., and Lemeshow, S. (1989). *Applied logistic regression*, Wiley: New York.

Huang, C.-L., Chen, M.-C., Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33, 847-856.

Huberty, C.J. (1994). Applied discriminant analysis. New York: Wiley.

Jacquet-Lagreze, E. and Siskos, Y. (1982). Assessing a set of additive utility functions for multicriteria decision making: The UTA method. *European Journal of Operational Research*, 10, 151-164.

Jacquet-Lagreze, E. (1995). An application of UTA discriminant model for the evaluation of R & D projects. In P.M. Pardalos, Y. Siskos, C. Zopounidis. *Advances in Multicriteria Analysis*. Kluwer Academic Publishers, Dordrecht.

Jacquet-Lagreze, E. and Siskos, Y. (2001). Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research*, 130, 233-245.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6, 429-449.

Joanes, D.N. (1993). Reject inference applied to logistic regression for credit scoring, *IMA Journal of Mathematics Applied in Business and Industry*, 5, 35-43.

John, G. H., Kohavi, R. and Pfleger, K. (1994). Irrelevant features and subset selection problem. *Machine Learning Proceedings of the 11th International Conference* , Morgan Kaufmann, 121-129.

Klecka, W.R. (1981). Discriminant analysis. Quantitative Applications in the Social Sciences, Sage University Press, London.

Koehler, G.J. (1990). Considerations for mathematical programming models in discriminant analysis. *Managerial and Decision Economics*, 11,227-234.

Koehler, G.J. and Erenguc, S.S.(1990). Minimising misclassifications in linear discriminant analysis. *Decision Sciences*, 21, 63-85.

Koehler, G.J. (1991). Linear discriminant functions determined by genetic search. *ORSA, Journal on Computing*, 3,345-357.

Kohavi, R. and John, G.H. (1997). Wrappers for feature subset selection, *Artificial Intelligence*, 97, 273-324.

Kononenko, I. and Bratko, I. (1991). Information-Based evaluation criterion for classifier's performance, *Machine Learning*, 6, 67-80.

Kotsiantis, S. and Pintelas, P. (2003). A cost sensitive technique for ordinal classification problems, in *Lecture notes in Artificial Intelligence*, Springer Verlag Vol 3025, 220-229.

Laitinen, E. K. (1999). Predicting a Corporate Credit Analyst's Risk Estimate by Logistic and Linear Models. *International Review of Financial Analysis*. 8(2) 97-121

Lam, K.F., Choo, E.U., and Wedley, W.C. (1993). Linear goal programming in estimation of classification probability. *European Journal of Operational Research*, 67,101-110.

Lam, K.F., Choo, E.U. and Moy, J.W. (1996). Minimizing deviations from the group mean: A new linear programming approach for the two-group classification problem. *European Journal of Operational Research*, 88,358-367.

Lam, K.F. and Moy, J.W. (1997). An experimental comparison of some recently developed linear programming approaches to the discriminant analysis. *Computers and Operations Research*, 24(7), 593-599.

Lane, S. (1972). Submarginal credit risk classification. *Journal of Financial and Quantitative Analysis*, 7, 1379-1385.

Leonard, K.J. Credit scoring models for the evaluation of small-business loan applications. *IMA Journal of Mathematics Applied in Business and Industry* 4, (1992), 89-95.

Lewis, E. M. (1992) *An Introduction to Credit Scoring*. Athena Press: San Rafael.

Li, S.-T., Shiue, W., and Huang, M.-H. (2006). The evaluation of consumer loans using support vector machines. *Expert Systems with Applications*, 30, 772-782.

Liittschwager, J.M. and Wang, C. (1978). Integer Programming solution of a classification problem. *Management Science*, 24, 1515-1525.

Limsombunchai, V., Gan, C., and Lee, M. (2005). An analysis of credit scoring for agricultural loans in Thailand. *American Journal of Applied Sciences* 2 (8): 1198-1205.

Lin, S-M, Ansell, J. and Andreeva, G. (2007a). Predicting default of a small business using different definitions of financial distress. *X Credit Scoring and Credit Control Conference*. Edinburgh.

Lin, S-M, Ansell, J. and Andreeva, G. (2007b). Merton models or credit scoring: modeling default of a small business. *X Credit Scoring and Credit Control Conference*. Edinburgh.

Liu, Y. and Schumann, M. (2005). Data mining feature selection for credit scoring models, *Journal of the Operational Research Society*, 1-10.

Lin, C.C., Chang, C.C., Li, F.C., and Chao, T.C. (2011). Features selection approaches combined with effective classifiers in credit scoring. *IEEE International Conference on Industrial Engineering and Engineering Management*, 752-757.

Loucopoulos, C. and Pavur, R. (1997). Experimental evaluation of the classificatory performance of mathematical programming approaches to the three-group discriminant problem: The case of small samples. *Annals of Operations Research*, 74, 191-209.

Louzada, F., Ferreira-Silva, P. and Diniz, C.A.R. (2012). On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data. *Expert Systems with Applications*, 39(9), 8071-8078.

Lovie, A.D. and Lovie, P. (1986). The flat maximum effect and linear scoring models for prediction. *Journal of Forecasting*, 5, 159-186.

Mahmood, M.A. and Lawrence, E.C. (1987). A performance analysis of parametric and nonparametric discriminant approaches to business decision making. *Decision Sciences*, 18, 308-326.

Mangasarian O. (1965). Linear and Nonlinear Separation of patterns by Linear Programming. *Operations Research*, 13, 444-452.

Markowski, E.P. and Markowski, C.A. (1985). Some difficulties and improvements in applying linear programming formulations to the discriminant problem. *Decision sciences*, 16, 237-247.

Markowski, C.A. (1990). On the balancing of error rates for the LP discriminant methods. *Managerial and Decision Economics*, 11, 235-241.

Mays, E. (1998). Credit Risk Modelling: Design and Application, Global Professional Publishing, Ohio.

Mays, E. (2004). Scorecard development, in *Credit Scoring for Risk Managers*, Ed. Mays E., Thomson, Ohio.

Mays, E., and Yuan, J. (2004). Variable analysis and reduction, in *Credit Scoring for Risk Managers*, Ed. Mays E., Thomson, Ohio.

Merton, R. (1974). On the pricing of corporate debt: The risk structure of interest rates. *Journal of Finance*, 29: 449-470.

Metaxiotis, K, and Psarras, J. (2003). Expert systems in business: applications and future directions for the operations researcher, *Industrial Management and Data Systems*, 103 (5), 361-368.

Michalopoulos, M., Dounias, G., Hatas, D., and Zopounidis, C. (2001). An automated knowledge generation approach for managing credit scoring problems, in *Fuzzy Sets in Management, Economics and Marketing*.

Nath, R. (1984). Estimation of misclassification probabilities in the linear programming approaches to the two-group discriminant problem. *Decision Sciences*, 15, 248-252.

Nath, R. and Jones, T.W. (1988). A variable selection criterion in the linear programming approaches to discriminant analysis. *Decision Sciences*, 19, 554-563.

Negnevitsky, M. (2002). Artificial Intelligence: *A guide to intelligent systems*. Pearson, London, UK.

Neter, J., Kutner, M.H., Nachtsheim, C.J. and Wasserman, W. (1996). *Applied linear statistical models*, Irwin, London.

Nikbakht, E. and Tafti, M.H.A. (1989). Application of expert systems in evaluation of credit card borrowers. *Managerial Finance* 15(5), 19-27.

Oreski, S., Oreski, D., and Oreski, G. (2012). Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. *Expert Systems with Applications*, 39(16), 12605-12617.

Orgler, Y.E. (1970). A credit scoring model for commercial loans. *Journal of Money, Credit, and Banking*, November, 435-445

Orgler, Y.E. (1971). of bank consumer loans with credit scoring models. *Journal of Bank Research*, Spring.

Orgler, Y.E. (1975), *Analytical methods in Loan Evaluation*, Lexington, Mass: Lexington Books, D.C. Heath.

Pal, S. K. and Mitra, P. (2004). Pattern recognition algorithms for data mining: Scalability, Knowledge Discovery and Soft Granular Computing. CRC Press.

Pampel, F.C. (2000). *Logistic regression: A primer*. Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-132. Thousand Oaks, CA:Sage.

Pavur, R., Wanarat, P. and Loucopoulos, C. (1997). Examination of the classificatory performance of MIP models with secondary goals for the two-group discriminant problem. *Annals of Operations Research*, 74, 173-189.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer and Information Sciences*, 11, 341–356.

Ping, Y. and Yongheng, L. (2011). Neighborhood rough set and SVM based hybrid credit scoring classifier. *Expert Systems with Applications*, 38(9), 11300-11304.

Pinson, S. (1992), A multi-expert architecture for credit risk assessment: the CREDEX system, in O'Leary, D.E., Watkin, P.R. (Eds), *Expert Systems in Finance*, Elsevier Science Publishers, Oxford, pp.37-64.

Piramuthu S., (1999a). Feature selection for financial credit-risk evaluation decisions. *INFORMS Journal on Computing*, 11, 258-266.

Piramuthu S., (1999b). Financial credit risk evaluation with neural and neurofuzzy systems. *European Journal of Operational Research*, 112, 310-321.

Piramuthu S., (2004). Evaluating feature selection methods for learning in data mining applications, *European Journal of Operational Research*, 156, 483-494.

Poon, M. (2007). Scorecards as devices for consumer credit: the case of Fair Isaac and Company Incorporated. *The Sociological Review*, 55, 284-306.

Press S.J., and Wilson S. (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, 73,699-705.

Reichert, A.K., Cho, C.C., and Wagner, G.M. (1983). An examination of the conceptual issues involved in developing credit-scoring models, *Journal of Business and Economics Statistics*, 1(2), 101-114.

Retzlaff-Roberts, D.L. (1996). A ratio model for discriminant analysis using linear Programming. *European Journal of Operational Research*, 94, 112-121.

Ripley, B.D. (1994). Neural networks and related methods for classification. *Journal of Royal Statistical Society*, 56, 409-456.

Robnik-Sikonja,M. and Kononenko, I.(2003). Theoretical and Empirical Analysis of ReliefF and RReliefF. *Machine Learning*, 53:23-69.

Rosen, J.B. (1965). Pattern separation by convex programming. *Journal of Mathematical Analysis and Applications*, 10, 123-134.

Rubin, P.A. (1994). A comment regarding polynomial discriminant functions, *European Journal of Operational Research*, 72,29-31.

Rubin, P.A. (1997). Solving mixed integer classification problems by decomposition. *Annals of Operations Research*, 74, 51-64.

Safavian, S.R. and Landgrebe, D. (1991). A survey of decision tree classifiers methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21 (3), 660-674.

Salkind, N.J. (2007). Encyclopedia of Measurement and Statistics. Thousand Oaks (CA). Sage.

Sarkar, D. (2005). Solving mixed integer formulation of the KS maximization problem – Dual based methods and results from large practical problems. *Credit scoring and Credit Control IX*, Edinburgh.

Schwalb, O., Lee, T.H. and Zheng, S. (2003). An algorithm for score calibration based on cumulative bad rates. *International Journal of Information Technology and Decision Making*, 2(1), 93-103.

Shaw, M.J. and Gentry, J.A. (1988). Using an Expert System with Inductive Learning to evaluate Business Loans, *Financial Management*, 45-56.

Shashua, A. and Levin, A. (2003). Ranking with large margin principle: two approaches. *Advances in Neural Information Processing Systems*, 15, 937-944.

Siegel, S. (1988). *Nonparametric statistics for the behavioral sciences*, New York; London : McGraw-Hill.

Sikonja, M.R. and Kononenko, I. (2003). Theoretical and Empirical Analysis of Relief and RReliefF. *Machine Learning*, 53, 1-2, 23-69.

Siskos, J. and Zopounidis, C. (1985). The evaluation criteria of the venture capital investment activity: An interactive assessment. *European Journal of Operational Research*, 31, 304-313.

Smith, C.A.B. (1947). Some examples of discrimination, *Annals of Eugenics*, 13, 272-282.

Smith, F.W. (1968). Pattern classifier design by linear programming, *IEEE Transactions on Computers*, C-17 (4), 367-372.

Sobehart, J., Keenan, S. and Stein, R. (2000). Validation methodologies for default risk models, *Credit* 51-56.

Somol P., Baesens B., Pudil P. and Vanthienen J. (2005). Filter-versus wrapper-based feature selection for credit scoring, *International Journal of Intelligent Systems*, 20,985-999.

SPSS for Windows, Rel. 11.0.1. (2001). Chicago: SPSS Inc.

Spyridakos, A., Siskos, Y., Yannacopoulos, D. and Skouris, A. (2001). Multicriteria job evaluation for large organizations. *European Journal of Operational Research* 130, 375-37.

Srinivasan, V. (1976). Linear programming computational procedures for ordinal regression. *Journal of the ACM*, 23(3):475-487.

Srinivasan, V. and Kim, Y.H. (1987) Credit granting: a comparative analysis of classification procedures. *The Journal of Finance*, 42, 665-683

Srinivasan, V., and Ruparel, B. (1990), CGX: an expert support system for credit granting, *European Journal of Operational Research*, Vol. 45 pp.293-308.

Stam, A. and Joachimsthaler, E.A. (1989). Solving the classification problem in discriminant analysis via linear and nonlinear programming methods. *Decision Sciences*, 20, 285-293.

Stam, A. (1990). Extensions of mathematical programming-based classification rules: A multicriteria approach. *European Journal of Operational Research*, 48,351-361.

Stam, A. and Joachimsthaler, E.A. (1990). A comparison of a robust mixed-integer approach to existing methods for establishing classification rules for the discriminant problem. *European Journal of Operational Research*, 46, 113-122.

Stam, A. and Jones, D.G. (1990). Classification performance of mathematical programming techniques in discriminant analysis: results for small and medium sample sizes. *Managerial and Decision Economics*, 11, 243-253.

Stam, A. and Ragsdale, C.T. (1990). A robust nonparametric procedure to estimate response functions for binary choice models. *Operations Research Letters*, 9, 51-58.

Stam, A. and Ragsdale, C.T. (1992). On the classification gap in mathematical programming-based approaches to the discriminant problem. *Naval Research Logistics*, 39, 545-559.

Stam, A. (1997). Nontraditional approaches to statistical classification: Some perspectives on L_p-norm methods. *Annals of Operations Research*, 74, 1-36.

Stiglitz, J.E. and Weiss, A. (1981). Credit rationing in markets with imperfect information. *The American Economic Review*, 71 (3), 393-410.

Sueyoshi, T. (1999). DEA-discriminant analysis in the view of goal programming. *European Journal of Operational Research*, 115, 564-582.

Sueyoshi, T. (2001). Extended DEA-discriminant analysis. *European Journal of Operational Research*, 131, 324-351.

Sueyoshi, T. (2006). DEA-Discriminant Analysis: Methodological comparison among eight discriminant analysis approaches. *European Journal of Operational Research*, 169, 247-272.

Sun, M. and Xiong, M. (2003). A mathematical programming approach for gene selection and tissue classification, *Bioinformatics*, 19, 1243-1251.

Sundbom, T. (2007). Mathematical programming based approaches in credit scoring. *Unpublished thesis*, University of Uppsala.

Tan, S. (2005). Neighbour-weighted k-nearest neighbour for unbalanced text corpus. *Expert Systems with Applications*, (28), 667-671.

Thomas, L.C., Banasik, J, Crook, J.N. (2001). Recalibrating scorecards. *Journal of the Operational Research Society*, 52, 981-988.

Thomas, L.C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers, *International Journal of Forecasting*, 16, 149-172.

Thomas, L.C., Edelman D.B., and Crook J.N. (2002) *Credit Scoring and its Applications.* Philadelphia: SIAM

Tian, X., and Deng, F. (2004). A credit scoring model using support vector machine, *Proceedings of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, China.

Tsai, M.C., Lin, S.P., Cheng, C.C.L. (2009). The consumer loan default predicting model – An application of DEA – DA and neural network. *Expert Systems with Applications*, 36(9), 11682-11690.

Tsai, C-F. (2009). Feature Selection in bankruptcy prediction, *Knowledge-Based Systems*, 22(2), 120-127.

Tsaih, R., liu, Y.-J., Liu, W. and Lien, Y.-L. Credit scoring system for small business loans. *Decision Support Systems* 38, (2004), 91-99.

Vapnik, V. (1995). *Nature of statistical learning theory*, New York, Springer-Verlag.

Vellido, A., Lisboa, P.J.G., and Vaughan, J. (1999). Neural networks in business: a survey of applications (1992-1998). *Expert Systems with Applications*, 17, 51-70.

Verikas, A., and Bacauskiene, M. (2002). Feature selection with neural networks, *Pattern Recognition Letters*, 23, 1323-1335.

Viaene, S., Derrig, R., Baesens, B., Dedene, G. (2002). A comparison of state-of-the-art classification techniques for expert automobile insurance fraud detection. *Journal of Risk and Insurance* 69(3), 373-421.

Vinciotti, V. and Hand, D.J. (2003). Scorecard construction with unbalanced class sizes, *Journal of the Iranian Statistical Society*, 2, 189-205.

Wanarat, P. and Pavur, R. (1996). Examining the effect of second order terms in mathematical programming approaches to the classification problem. *European Journal of Operational Research*, 93, 582-601.

Wang, Y., Wang, S., and Lai, K.K. (2005). A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, 13 (6), 820-831.

Wang, J., Hedar, A-R, Wang, S. and Shouyang, M.J.(2012). Rough set and scatter search metaheuristic based feature selection for credit scoring. *Expert Systems with Applications*, 39(6), 6123-6128.

Wendling, F., and Goncalves, R. (2007). Use of macro-economic factors in credit scoring – Application to point in time risk evaluation of SMEs. *Credit scoring and credit control conference X*.

West D. (2000). Neural Network Credit Scoring Models. *Computers & Operations Research*. 27: 1131-1152.

Westgaard, S., and Van der Wijst, N. (2001). Default probabilities in a corporate bank portfolio: A logistic model approach. *European Journal of Operational Research*, 135, 338-349.

Wiginton J.C. (1980) A note on the comparison of Logit and Discriminant models of Consumer Credit Behavior. *Journal of Financial and Quantitative Analysis*, 15,757-770.

Williams, H.P. (1999). *Model building in mathematical programming*. Wiley: Chichester.

Wilson, J.M. (1996). Integer Programming Formulations of Statistical Classification Problems, *Omega*, 24, 681-688.

Witten, I.H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*, 2nd Edition, Morgan Kaufmann, San Francisco.

Yan, R. (2006). MatlabArsenal: A Matlab package for classification algorithms.

Yobas, M.B., Crook, J.N., and Ross, P. (2000). Credit scoring using neural and evolutionary techniques, *IMA Journal of Mathematics Applied in Business and Industry*, 11, 111-125.

Zadeh, L.A. (1965). Fuzzy Sets, *Information and Control*, 8 (3), 338-353.

Ziari, H.A., Leatham, D.J. and Ellinger, P.N. (1995). An application of mathematical programming techniques in credit scoring of agricultural loans. *Agricultural Finance Review*, 55, 74-88.

Ziari, H.A., Leatham, D.J. and Ellinger, P.N. (1997). Development of statistical discriminant mathematical programming model via resampling estimation techniques. *American Journal of Agricultural Economics*, 79, 1352-1362.

Zopounidis, C. and Doumpos, M. (1998). Developing a multicriteria decision support systems for financial classification problems: The FINCLAS system. *Optimization Methods and Software*, 8, 277-304.

Zopounidis, C. Pardalos, P., Doumpos, M., and Mavridou, T. (1998). Multicriteria decision aid in credit cards assessment. In *Managing in Uncertainty: Theory and Practice*, Zopounidis C. Pardalos, P. (eds) Kluwer Academic Publishers: Dordrech; 163-178.

Zopounidis, C., Doumpos, M. and Zanakis, S.(1999). Stock evaluation using a preference disaggregation methodology. *Decision Sciences*, 30 (2), 313-336.

Zopounidis, C. and Doumpos, M. (1999). A multicriteria decision aid methodology for sorting decision problems: the case of financial distress. *Computational Economics*, 14, 197-218.

Zopounidis, C. and Doumpos, M. (2002). Multicriteria classification and sorting methods: A literature review. *European Journal of Operational research*, 138, 229-246.

- APPENDIX –

APPENDIX A
Data description included in experimental studies

Table A-1: Greek Dataset Characteristics

	Name	Type
1	Residential Status	Discrete
2	Marital Status	Discrete
3	Age	Continuous
4	Income	Continuous
5	Area	Discrete
6	Occupation Code	Discrete
7	Time in Job	Continuous
8	Dependents	Continuous
9	Time in Address	Continuous
10	Phones	Discrete
11	Card Type	Discrete
12	Sex	Discrete
13	OtherCard1	Discrete
14	OtherCard2	Discrete
15	OtherCard3	Discrete
16	OtherCard4	Discrete
17	OtherCard5	Discrete
18	OtherCard6	Discrete
19	OtherCard7	Discrete
20	OtherCard8	Discrete
21	OtherCard9	Discrete
22	OtherCard10	Discrete
23	Stand order	Discrete
24	Status Delivery	Discrete
25	Secondary Card1	Discrete
26	Secondary Card2	Discrete
27	Mortgage	Discrete
28	Consumer loan	Discrete
29	Bank type	Discrete
30	Mutual fund	Discrete
31	Insurance	Discrete
32	Minimum Payment	Discrete
33	Balance transfer	Discrete
34	Account type	Discrete
35	Credit limit	Discrete
37	Mobile phone	Discrete
38	Home phone	Discrete
39	Business phone	Discrete

Table A-2: German Dataset Characteristics

	Name	**Type**
1	Status of existing checking account	Discrete
2	Duration in months	Continuous
3	Credit history	Continuous
4	Purpose	Discrete
5	Credit amount	Continuous
6	Savings account	Discrete
7	Present employment since	Continuous
8	Installment rate in percentage of disposable income	Continuous
9	Personal status and gender	Discrete
10	Other debtors/guarantors	Discrete
11	Date beginning permanent residence	Continuous
12	Property	Discrete
13	Age in years	Continuous
14	Other installment plans	Discrete
15	Housing	Discrete
16	Number of existing credits at this bank	Continuous
17	Job	Discrete
18	Number of dependents	Continuous
19	Telephone	Discrete
20	Foreign Worker	Discrete

Table A-3: SME Dataset Characteristics

	Name	**Type**
1	Cash / Total Assets	Continuous
2	Liabilities / Total Assets	Continuous
3	Equity / Total Assets	Continuous
4	Sales / Total Assets	Continuous
5	Cash / Net Sales	Continuous
6	Profit / Sales	Continuous
7	Working Capital / Assets	Continuous
8	Account Payable / Sales	Continuous
9	Account Receivable / Liabilities	Continuous

Table A-4: SPSS Dataset Characteristics

	Name	Type
1	Age in years	Continuous
2	Level of education	Discrete
3	Years with current employment	Continuous
4	Years at current address	Continuous
5	Household Income	Continuous
6	Debt to income ratio	Continuous
7	Credit Card Debt	Continuous
8	Other debt	Continuous

Table A-5: US Dataset Characteristics

	Name	Type
1	Age	Continuous
2	Expenditure_Jan	Continuous
3	Expenditure_Feb	Continuous
4	Expenditure_Mar	Continuous
5	Expenditure_Apr	Continuous
6	Expenditure_May	Continuous
7	Expenditure_Jun	Continuous
8	Expenditure_Jul	Continuous
9	Expenditure_Aug	Continuous
10	Expenditure_Sept	Continuous
11	Expenditure_Oct	Continuous
12	Expenditure_Nov	Continuous
13	Expenditure_Dec	Continuous
15	Dependents	Continuous
16	Months in previous address	Continuous
17	Additional income	Continuous
18	Income	Continuous
19	Selfemployed	Discrete
20	Professional	Discrete
21	Unemployed	Discrete
22	Management	Discrete
23	Military	Discrete
24	Clerical	Discrete
25	Sales	Discrete
26	Other job	Discrete
27	Months at current address	Continuous
28	Number of credit bureaus inquiries	Continuous
29	Major credit card	Discrete
30	Department store credit card	Discrete
31	Gasoline credit card	Discrete
32	Months employed	Continuous
33	Current trade item accounts	Continuous
34	Bank savings account	Discrete
35	Bank checking account	Discrete
36	Major derogatory reports	Continuous
37	Minor derogatory reports	Continuous
38	Number of open and current trade lines	Continuous
39	Number of trade active lines	Continuous
40	Number of trade lines 30 days past due	Continuous
41	Number of 30 day delinquencies within 12 months	Continuous
42	Dollar amount of averaging revolving balance	Continuous

Table A-6: Australian – Company dataset Characteristic

	Name	Type
1	Company Type	Discrete
2	Home Phone	Discrete
3	Residential Status	Discrete
4	Product Type	Discrete
5	Age of Applicant	Continuous
6	Time in Current Address	Continuous
7	Time in Business	Continuous
8	Turnover in Current Year	Continuous
9	NBPT Current Year	Continuous
10	NBPT Current Year - rate	Continuous
11	Average Net	Continuous
12	Term	Continuous

Table A-7: Australian – Individuals dataset characteristics

	Name	Type
1	Security Flag	Discrete
2	Home Phone	Discrete
3	Residential Status	Discrete
4	Product Type	Discrete
5	Age of Applicant	Continuous
6	Time in Current Address	Continuous
7	Time in Current Employment	Continuous
8	Time in previous Employment	Continuous
9	Time in Previous Address	Continuous
10	Number of Dependants	Continuous
11	Number of Credit Cards	Continuous
12	Term	Continuous
13	Amount Financed	Continuous

Table A-8: Australian – Sole Trader dataset characteristics

	Name	Type
1	Security Flag	Discrete
2	Home Phone	Discrete
3	Residential Status	Discrete
4	Product Type	Discrete
5	Age of Applicant	Continuous
6	Time in Current Address	Continuous
7	Time in Current Employment	Continuous
8	Time in previous Employment	Continuous

Table A-9: Fraud scoring dataset

	Name	Type
1	Time of Transaction	Continuous
2	Type of Merchant	Discrete
3	Overseas Transaction	Discrete
4	Time Since Prev Ret Txn	Continuous
5	Retail Txns Last Hour	Continuous
6	Retail Txns Last 24h	Continuous
7	Retail Txns Last 7 days	Continuous
8	Retail Txns Last 28 days	Continuous
9	Time Since Prev Cash Txn	Continuous
10	Cash Txns Last hour	Continuous
11	Cash_Txns_Last_24h	Continuous
12	Cash Txns Last 7 days	Continuous
13	Cash Txns Last 28 days	Continuous
14	Hst % Txn same value L12M retail	Continuous
15	Hst % Txn same merchant L12M retail	Continuous
16	Transactions L72hrs 3 days	Continuous
17	Av value of retail tx in last hour	Continuous
18	Country of Origin Risk Group	Continuous
19	Av Value Last 5 Txns	Continuous
20	Amount	Continuous
21	MCC	Continuous
22	Dev_Val_Flag	Discrete

APPENDIX B

Data Transformation
Greek dataset

Table B-1: Coarse classification, Greece, Residential status

ATTRIBUTE	WOE	GROUP
Tenant	-0.42996	1
Other	-0.26966	1
Owner	0.229263	2

Table B-2: Coarse classification, Greece, Marital status

ATTRIBUTE	WOE	GROUP
Single	-0.21641	1
Other	-0.22424	1
Married	0.231861	2

Table B-3: Coarse classification, Greece, occupation code

ATRIBUTE	WOE	GROUP
<=4	-0.43691	1
5-8	-0.064	2
9-11	-0.22231	2
12-13	-0.10662	2
14	0.173432	3
15	0.87811	2
16-18	0.505812	1

Table B-4: Coarse classification, Greece, Age

ATTRIBUTE	WOE	GROUP
<=25	-0.58239	1
26-28	-0.02216	2
29-32	0.055356	2
33-35	0.052109	2
36-38	0.063555	2
39-41	0.157897	2
42-45	-0.07757	2
46-50	0.185309	3
51-57	0.214784	3
58<=	0.216558	3

Table B-5: Coarse classification, Greece, Phones

ATTRIBUTES	WOE	GROUP
1	0.009043	1
2	-0.18924	1
3	0.00247	1
4	-0.8077	2

Table B-6: Coarse classification, Greece, Time in Job

ATTRIBUTES	WOE	GROUP
0 MONTHS	0.233042	1
1	-0.50682	2
2-3	-0.45572	2
4	-0.45501	2
5-6	-0.1303	2
7-10	0.227133	3
11-18	0.319044	3
19<=	0.453796	3

Table B-7: Coarse classification, Greece, dependents

ATTRIBUTES	WOE	GROUP
0	-0.0253	1
≠ 0	0.074927	2

Table B-8: Coarse classification, Greece, Time in address

ATTRIBUTE	WOE	GROUP
0	0.060111	1
1-3	-0.2898	2
4-5	-0.26396	2
6-10	0.105687	3
11-19	0.156332	3
20-27	0.127661	3
28-75	-0.01869	4

Table B-9: Coarse classification, Greece, area

ATTRIBUTE	WOE	GROUP
1	-0.10023	1
2-3	-0.23203	2
4	-0.20903	3
5-6	-0.26946	1
7-8	0.07706	1
9-10	0.121934	3
11-12	0.255985	1
13	0.189038	2

Table B-10: Coarse classification, Greece, income

ATTRIBUTE	WOE	GROUP
0	0.239574	1
<=68	-0.26364	2
69-85	-0.38478	2
86-100	-0.26761	2
101-121	-0.26878	2
122-150	0.095805	3
151-188	0.100954	3
189-262	0.211448	3
263<=	0.455627	4

US Dataset

Table B-11: Coarse classification, US, dependents

ATTRIBUTE	WOE	GROUP
0	-0.48538	1
≠0	0.223048	2

Table B-12: Coarse classification, US, months in previous address

ATTRIBUTE	WOE	GROUP
1	0.265914	1
2	-0.03251	2
3	0.040281	2
4	0.094538	2
5	-0.02892	3
6	0.131068	3
7	-0.17215	4
8	-0.25248	4
9	-0.11717	4

Table B-13: Coarse classification, US, months in current address

ATTRIBUTE	WOE	GROUP
1	0.084517	1
2	-0.06001	1
3	0.481184	2
4	-0.01131	3
5	-0.08807	3
6	-0.17191	3
7	-0.07081	3
8	-0.04723	3
9	-0.03891	3
10	0.224069	4

Table B-14: Coarse classification, US, months employed

ATTRIBUTE	WOE	GROUP
1	0.506201	1
2	-0.06949	2
3	0.174633	2
4	-0.13567	3
5	-0.1492	3
6	-0.18093	3
7	-0.01678	3
8	-0.07499	3
9	0.004858	4
10	0.111508	4

Table B-15: Coarse classification, US, additional income

ATTRIBUTE	WOE	GROUP
1	0.114534	1
2	0.16138	1
3	-0.31291	2
4	-0.47261	2
5	0.178897	2

Table B-16: Coarse classification, US, age

ATTRIBUTE	WOE	GROUP
1	-0.67596	1
2	-0.16984	1
3	0.075588	2
4	0.038422	2
5	0.065937	2
6	0.211647	3
7	0.107333	3
8	0.23638	4
9	0.276409	4
10	0.175279	4

Table B-17: Coarse classification, US, active trade lines

ATTRIBUTE	WOE	GROUP
1	0.268277	1
2	-0.51304	2
3	-0.37227	2
4	-0.48393	2
5	-0.133	2
6	-0.02139	2
7	0.411221	3
8	0.325492	3
9	0.448259	3
10	0.492641	3

Table B-18: Coarse classification, US, income

ATTRIBUTE	WOE	GROUP
1	-0.56538	1
2	-0.3718	1
3	-0.26501	1
4	-0.08256	2
5	0.015937	2
6	0.043475	2
7	0.24991	3
8	0.115238	3
9	0.701376	4
10	0.874926	4

Table B-19: Coarse classification, US, average balance

ATTRIBUTE	WOE	GROUP
1	0.14228	1
2	0.401624	1
3	0.160614	1
4	0.059849	1
5	-0.02401	2
6	-0.06676	2
7	-0.10036	2
8	-0.22894	3
9	-0.44073	3
10	-0.13357	3

Table B-20: Coarse classification, US, average expenses

ATTRIBUTE	WOE	GROUP
1	-0.56152	1
2	-0.01533	2
3	0.006779	2
4	-0.12033	3
5	-0.16	3
6	0.062801	3
7	0.186731	4
8	0.383442	4
9	0.173861	4
10	0.323782	4

Table B-21: Coarse classification, US, expenditure_January

ATTRIBUTE	WOE	GROUP
1	-1.43167	1
2	-1.68025	1
3	0.728113	2
4	-0.39603	2
5	0.27167	2
6	1.203261	3
7	1.054138	3
8	1.271725	3
9	1.792297	3
10	1.978491	3

Table B-22: Coarse classification, US, expenditure_February

ATTRIBUTE	WOE	GROUP
1	-0.81165	1
2	-1.7965	1
3	0.172644	2
4	0.423447	2
5	0.492555	2
6	1.255448	3
7	1.457957	3
8	1.457957	3
9	1.341697	3

Table B-23: Coarse classification, US, expenditure_March

ATTRIBUTE	WOE	GROUP
1	-0.57713	1
2	-1.00285	1
3	0.26806	2
4	0.600048	2
5	0.804988	2
6	1.341697	3
7	1.543287	3
8	0.947527	3

Table B-24: Coarse classification, US, expenditure_April

ATTRIBUTE	WOE	GROUP
1	-0.41422	1
2	-0.64618	1
3	-0.99239	1
4	0.136062	2
5	0.494906	2
6	0.740117	2
7	1.082524	3
8	0.897915	3
9	0.678843	3

Table B-25: Coarse classification, US, expenditure_May

ATTRIBUTE	WOE	GROUP
1	-0.23129	1
2	-2.37068	1
3	-0.6592	1
4	0.173861	2
5	0.658143	2
6	0.719314	2
7	0.873932	3
8	0.430301	3
9	0.2262	3

Table B-26: Coarse classification, US, expenditure_June

ATTRIBUTE	WOE	GROUP
1	-0.36355	1
2	-1.90513	1
3	-0.23262	1
4	0.283016	2
5	0.614715	2
6	0.547745	2
7	0.479158	3
8	0.338412	3
9	0.309334	3

Table B-27: Coarse classification, US, expenditure_July

ATTRIBUTE	WOE	GROUP
1	-0.23317	1
2	-1.93019	1
3	-0.51088	1
4	0.381399	2
5	0.572514	2
6	0.699896	2
7	0.293007	3
8	0.365167	3
9	0.0407	3

Table B-28: Coarse classification, US, expenditure_August

ATTRIBUTE	WOE	GROUP
1	-0.06065	1
2	-1.81464	1
3	-0.48379	1
4	0.172475	2
5	0.581646	2
6	0.387574	2
7	0.383442	3
8	0.370277	3
9	-0.13357	3

Table B-29: Coarse classification, US, expenditure_September

ATTRIBUTE	WOE	GROUP
1	0.014082	1
2	-1.5568	1
3	-0.45606	1
4	0.238622	2
5	0.273743	2
6	0.331221	2
7	0.305223	3
8	0.253263	3
9	-0.0381	3

Table B-30: Coarse classification, US, expenditure_October

ATTRIBUTE	WOE	GROUP
1	0.085782	1
2	-0.36693	1
3	0.23759	2
4	0.028238	3
5	-0.01448	3
6	-0.02202	3
7	-0.01849	3
8	-0.0381	3

Table B-31: Coarse classification, US, expenditure November

ATTRIBUTE	WOE	GROUP
1	0.053512	1
2	-0.11188	1
3	0.523102	2
4	0.413449	2
5	0.388604	2
6	0.335336	2
7	-0.05522	3
8	-0.46051	3
9	-0.68008	3

Table B-32: Coarse classification, US, expenditure_December

ATTRIBUTE	WOE	GROUP
1	-0.01138	1
2	0.380669	1
3	1.076589	1
4	0.149738	2
5	0.40292	2
6	0.235522	2
7	-0.00538	3
8	-0.35777	3
9	-0.66776	3

SME DATASET

Table B-33: Coarse Classification, SME, Cash / Total Assets

ATTRIBUTE	WOE	GROUP
0.000	-0.12348	1
0.012	-1.04209	1
0.030	-0.40036	1
0.055	-0.40046	1
0.094	0.147411	2
0.152	0.005381	2
0.241	0.436404	3
0.382	0.387859	3
0.680	0.922282	3
2.839	0.968802	3

Table B-34: Coarse Classification, SME, Liabilities / Total Assets

ATTRIBUTE	WOE	GROUP
0.000	1.733212	1
0.000	0.487804	1
0.094	0.15022	2
0.247	0.244692	2
0.412	-0.00109	3
0.589	-0.32142	3
0.784	-0.38289	3
1.034	-0.64305	3
1.862	-0.84963	3
908.733	0	3

Table B-35: Coarse Classification, SME, Equity / Total Assets

ATTRIBUTE	WOE	GROUP
-0.862	-0.85355	1
-0.034	-0.56761	1
0.176	-0.32037	1
0.381	-0.32646	1
0.556	-0.00735	1
0.718	0.179124	2
0.876	0.15022	2
0.994	0.465391	3
1.000	1.812563	3
16.560	-1.77969	3

Table B-36: Coarse Classification, SME, Sales / Total Assets

ATTRIBUTE	WOE	GROUP
0.400	0.588442	1
0.932	0.287173	1
1.490	0.028599	1
2.071	-0.1464	2
2.802	-0.28765	2
3.704	-0.16013	2
4.949	0.005513	2
7.069	-0.11563	2
13.791	-0.15012	2
50000.000	0.156433	3

Table B-37: Coarse Classification, SME, Cash / Net Sales

ATTRIBUTE	WOE	GROUP
0.000	-0.25416	1
0.004	-0.92304	1
0.011	-0.58914	1
0.018	-0.3821	1
0.028	-0.22506	1
0.043	0.214853	2
0.066	0.337492	2
0.106	0.871714	2
0.195	1.20718	3
13.929	1.161821	3

Table B-38: Coarse Classification, SME, Profit / Sales

ATTRIBUTE	WOE	GROUP
-0.065	-0.19427	1
0.000	-0.2403	1
0.025	-0.11519	1
0.063	-0.2564	1
0.126	0.216864	2
0.215	0.059536	2
0.330	0.128241	2
0.500	0.280824	3
0.714	-0.47344	3
5.864	0.569613	3

Table B-39: Coarse Classification, SME, Working Capital / Assets

ATTRIBUTE	WOE	GROUP
0.000	-0.23813	1
0.091	0.166221	1
0.203	-0.27738	1
0.341	-0.2584	1
0.482	-0.05825	1
0.615	-0.2705	1
0.760	0.018714	2
0.899	0.079209	2
1.000	0.601399	2
1.958	-1.08654	3

Table B-40: Coarse Classification, SME, Account Payable / Sales

ATTRIBUTE	WOE	GROUP
0.0000	1.02204	1
>0.0000	-0.49915	2

Table B-41: Coarse Classification, SME, Account Receivable / Liabilities

ATTRIBUTE	WOE	GROUP
0.0000	0.669324	1
>0.0000	-0.3895	2

SPSS DATASET

Table B-42: Coarse Classification, SPSS, Level of Education

ATTRIBUTE	WOE	GROUP
High School	-0.03789	1
Post Undrg	0.185841	2
Did not Comple	0.4904	2
College Degree	-0.34952	3
Some College	0.021828	3

Table B-43: Coarse Classification, SPSS, Age

ATTRIBUTE	WOE	GROUP
1	-1.07919	1
2	-0.76682	1
3	-0.5845	2
4	-0.57939	2
5	-0.39525	3
6	-0.40057	3
7	-0.22522	4
8	-0.29236	4
9	-0.29478	4
10	0.59576	5

Table B-44: Coarse Classification, SPSS, Years with current employment

ATTRIBUTE	WOE	GROUP
1	-0.7197	1
2	-0.36778	1
3	-0.25599	1
4	0.274636	2
5	0.267256	3
6	0.161896	3
7	0.64195	3
8	0.459628	3
9	1.480708	4
10	0.372617	5

Table B-45: Coarse Classification, SPSS, Years at Current Address

ATTRIBUTE	WOE	GROUP
1	-0.77677	1
2	-0.26924	1
3	-0.15163	1
4	-0.21599	1
5	-0.00468	1
6	0.201659	2
7	0.052309	3
8	0.825813	3
9	0.443713	3
10	0.619477	3

Table B-46: Coarse Classification, SPSS, Household Income

ATTRIBUTE	WOE	GROUP
1	-0.48541	1
2	-0.49103	1
3	-0.48305	1
4	-0.66503	1
5	0.12842	2
6	-0.02573	3
7	-0.06989	3
8	-0.26604	4
9	-0.29584	4
10	-0.14726	5

Table B-47: Coarse Classification, SPSS, Debt to Income ratio

ATTRIBUTE	WOE	GROUP
1	1.292537	1
2	1.135968	1
3	0.761275	2
4	0.825813	2
5	1.145807	2
6	0.411837	3
7	0.750247	3
8	0.354268	4
9	0.252657	4
10	0.51922	4

Table B-48: Coarse Classification, SPSS, Credit Card Debt

ATTRIBUTE	WOE	GROUP
1	0.603728	1
2	0.252657	1
3	0.73726	2
4	0.537239	2
5	0.190295	3
6	0.489341	3
7	0.066092	4
8	0.092315	4
9	0.323827	5
10	0.188694	5

Table B-49: Coarse Classification, SPSS, Other Debt

ATTRIBUTE	WOE	GROUP
1	0.907159	1
2	0.212652	2
3	0.449578	2
4	0.12927	2
5	0.316527	3
6	0.335575	3
7	0.309816	3
8	0.149473	4
9	-0.21989	5
10	-0.22209	5

COMPANY DATASET

Table B-50: Coarse Classification, Company, Age of applicant

ATTRIBUTE	WOE	GROUP
240-379	0.068902858	1
380-421	-0.163949507	1
422-458	0.657861775	2
459-490	0.457191079	2
491-519	0.758016149	3
520-554	0.284561512	3
555-589	0.550115572	3
590-627	0.193538115	4
628-677	0.773259124	5
678+	1.241669127	5

Table B-51: Coarse Classification, Company, Months in Current Address

ATTRIBUTE	WOE	GROUP
low-1	-1.552957574	1
2-10	0.348268875	1
11-18	-0.086536897	1
19-35	0.525124348	2
36-47	0.112913852	2
48-60	0.386498786	3
61-93	0.874425232	3
94-128	0.832008628	3
129-192	1.148549529	3
193+	0.625246679	3

Table B-52: Coarse Classification, Company, Months in Business

ATTRIBUTE	WOE	GROUP
0-1	-1.455903625	1
2-17	0.150208239	2
18-35	-0.232071678	2
36-48	0.0381312	2
49-71	0.160504558	2
72-84	1.022448152	3
85-119	1.652788959	3
120-154	0.799796994	3
55-216	0.951458608	3
217+	0.670490198	3

Table B-53: Coarse Classification, Company, NBPT Current Years rate

ATTRIBUTE	WOE	GROUP
low--0.0001	-0.002854355	1
0	-0.88973747	1
0.0001-0.0323	0.393728671	2
0.0324-0.0665	0.706689761	2
0.0666-0.105	0.288774442	2
0.1051-0.1632	0.523267337	2
0.1633-0.2466	0.432531561	2
0.2467-0.3756	0.524010556	2
0.3757-0.9809	0.270080165	2
0.981+	0.093468908	2

Table B-54: Coarse Classification, Company, NBPT Current Years

ATTRIBUTE	WOE	GROUP
low-0	-1.007007314	1
1	0.22891885	1
2-12481	0.534063863	2
12482-29814	0.937369143	2
29815-42000	0.884652361	2
42001-60826	0.833509003	2
60827-100000	0.632116186	2
100001-181000	0.541319034	2
181000 – 190000	-0.456677195	3
190000+	-0.550056745	3

Table 255: Coarse Classification, Company, Turnover Current Years

ATTRIBUTE	WOE	GROUP
0	-1.650802231	1
1-49964	0.14217506	1
49965-109989	0.518796391	2
109990-169810	1.529650198	2
169811-250000	0.908580315	2
250001-378739	0.762854868	2
378740-568817	0.161134083	3
568818-891225	0.432531561	3
891226-1544496	0.276887229	3
1544497+	-0.03298881	3

Table B-56: Coarse Classification, Company, Average Net

ATTRIBUTE	WOE	GROUP
0	-1.05428277	1
0.0833-1033.3333	0.13428119	1
1040-2666.4166	0.59829894	2
2666.6666-3500	0.55257346	2
3501-4166.6666	0.16862089	2
4174.5833-5000	0.5161042	2
5001-6666.25	0.22193809	2
6666.6666-8666.6666	0.72297102	3
8725 – 9120	0.71797232	3
9120 +	0.8645367	3

Table B-57: Coarse Classification, Company, Term

ATTRIBUTE	WOE	GROUP
1.low-22	0.039809	1
2.23-24	0.451749	1
3.25-36	-0.61943	2
4.37-60	-0.04413	2

Table B-58: Coarse Classification, Company, Company Type

ATTRIBUTE	WOE	GROUP
LTD	-1.574936481	1
NAC	-1.097385646	1
P/L	0.082804575	2

Table B-59: Coarse Classification, Company, Home Phone

ATTRIBUTE	WOE	GROUP
N	-1.448999643	0
Y	0.43078875	1

Table B-60: Coarse Classification, Company, Residential Status

ATTRIBUTE	WOE	GROUP
L	-0.0321133	1
O	-1.7792115	1
R	-0.2616754	2
W	0.61562816	3

Table B-61: Coarse Classification, Company, Product

ATTRIBUTE	WOE	GROUP
Lease	72.45652174	0
Rent	13259.54348	1

INDIVIDUALS DATASET

Table B-62: Coarse Classification, Individuals, Security Provided

ATTRIBUTE	WOE	GROUP
N	-0.003025119	0
Y	0.359189667	1

Table B-63: Coarse Classification, Individuals, Home Phone

ATTRIBUTE	WOE	GROUP
N	0.818661568	0
Y	-0.040953773	1

Table B-64: Coarse Classification, Individuals, Residential Status

ATTRIBUTE	WOE	GROUP
L	0.58223387	1
R	0.51598357	1
W	-0.8653307	2

Table B-65: Coarse Classification, Individuals, Product

ATTRIBUTE	WOE	GROUP
Lease	-0.49152233	0
Rent	0.422704221	1

Table B-66: Coarse Classification, Individuals, Age of Applicant

ATTRIBUTE	WOE	GROUP
low-348	-0.839188647	1
349-389	-0.526868285	1
390-426	-0.227021857	1
427-461	-0.101303386	2
462-492	-0.094138695	2
493-525	0.173817581	2
526-560	0.710236001	3
561-602	0.792920503	3
603-657	0.741182183	3
658+	0.946533082	3

Table B-67: Coarse Classification, Individuals, Current Address

ATTRIBUTE	WOE	GROUP
low-5	-0.35441296	1
6-12	-0.259117539	1
8-12	-0.180928046	1
13-20	-0.169318468	1
21-30	-0.119554118	1
31-42	0.034292589	2
43-60	0.141431787	2
61-96	0.450593666	3
97-168	0.557626045	3
169+	0.330950208	3

Table B-68: Coarse Classification, Individuals, Current Employment

ATTRIBUTE	WOE	GROUP
low-5	-0.278618813	1
6-11	-0.191057532	1
12-18	-0.397542057	1
19-24	-0.166976416	1
25-35	0.090821786	2
36-41	0.075806105	2
42-53	-0.06284328	2
54-66	0.313178228	3
67-83	0.431069987	3
84+	0.740147522	3

Table B-69: Coarse Classification, Individuals, Amount Financed

ATTRIBUTE	WOE	GROUP
low-1490	0.344547044	1
1491-1985.45	-0.120932478	1
1985.46-2272.73	-0.043622092	1
2272.74-2545.46	0.053107458	1
2545.47-2785.54	-0.119115947	1
2785.55-3136.35	0.069160224	2
3136.36-3363.62	0.165320002	2
3363.63-3702.73	-0.032507007	3
3702.74-4318.17	-0.038289603	3
4318.18+	-0.179268055	3

Table B-70: Coarse Classification, Individuals, Term

ATTRIBUTE	WOE	GROUP
low-24	0.161703773	1
25-35	0.160479982	1
36	0.030886855	1
37+	-0.5010954	2

Table B-71: Coarse Classification, Individuals, No of Credit Cards

ATTRIBUTE	WOE	GROUP
0	-0.819200804	1
>0	0.452805597	2

Table B-72: Coarse Classification, Individuals, No of Dependants

ATTRIBUTE	WOE	GROUP
0	-0.095489117	1
1	0.120787459	1
2	0.213793415	1
3+	0.062686938	1
Missing	2.228536201	2

Table B-73: Coarse Classification, Individuals, Time in Previous Address

ATTRIBUTE	WOE	GROUP
0	0.290522686	1
1-5	-0.433327676	2
6-17	-0.490165419	2
18-30	-0.315299445	3
31-59	-0.317540476	3
60+	0.242387009	3

Table B-74: Coarse Classification, Individuals, Time in previous employment

ATTRIBUTE	WOE	GROUP
0	0.340540686	1
1-7	0.303334676	1
8-15	-0.090509669	2
16-28	-0.214602225	2
29-51	-0.313900346	2
52+	-0.545687009	2

SOLE TRADER DATASET

Table B-75: Coarse Classification, Sole Trader, Security Provided

ATTRIBUTE	WOE	GROUP
N	0.000829333	1
Y	-0.01115652	2

Table B-76: Coarse Classification, Sole Trader, Home Phone

ATTRIBUTE	WOE	GROUP
N	0.860443741	1
Y	-0.024544304	2

Table B-77: Coarse Classification, Sole Trader, Residential Status

ATTRIBUTE	WOE	GROUP
L	0.75848669	1
O	0.80267698	1
R	-0.5276889	2

Table B-78: Coarse Classification, Sole Trader, Product

ATTRIBUTE	WOE	GROUP
Lease	-1.314361232	1
Rent	0.054115764	2

Table B-79: Coarse Classification, Sole Trader, Age of Applicant

ATTRIBUTE	WOE	GROUP
1.low-282	-0.991394786	1
2.283-323	-0.092241023	1
3.324-357	0.318582437	1
4.358-391	-0.136426626	2
5.392-430	-0.132391651	2
6.431-473	0.129211516	2
7.474-513	0.689930283	3
8.514-557	0.19544105	3
9.558-614	0.887244144	3
10.615+	0.450107811	3

Table B-80: Coarse Classification, Sole Trader, Time in current address

ATTRIBUTE	WOE	GROUP
low-3	-0.1402012	1
4-7	-0.428139518	1
8-12	0.00678652	1
13-20	-0.665451326	2
21-30	-0.321011415	2
31-42	0.387896888	3
43-60	0.257289589	3
61-96	0.953273651	3
97-168	0.472006125	3
169+	0.481730999	3

Table B-81: Coarse Classification, Sole Trader, Term

ATTRIBUTE	WOE	GROUP
low-24	0.07595	1
25-34	-1.22431	1
35 - 36	0.037777	1
37+	-0.39452	2

APPENDIX C

T – tests results

Chapter 3 – Experimental Study

Australian - Overall												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0751	0.0826	0.0070	0.1283	0.3479	0.0887	0.0837	0.1587	0.2685	0.1011	0.5857
LDA	0.0751		0.5693	0.0161	0.3691	0.8348	0.8880	0.0186	0.0730	0.0684	0.4216	0.8562
QDA	0.0826	0.5693		0.0122	0.5020	0.9509	0.6783	0.0023	0.0176	0.0035	0.5184	0.5658
3-NN	0.0070	0.0161	0.0122		0.0072	0.0369	0.0179	0.0007	0.0001	0.0003	0.0059	0.0016
10-NN	0.1283	0.3691	0.5020	0.0072		0.5022	0.4052	0.0120	0.0045	0.0166	0.6744	0.2205
Neural	0.3479	0.8348	0.9509	0.0369	0.5022		0.8748	0.0520	0.0797	0.1393	0.5189	0.7701
MSD	0.0887	0.8880	0.6783	0.0179	0.4052	0.8748		0.0314	0.0601	0.0803	0.4549	0.7936
DT	0.0837	0.0186	0.0023	0.0007	0.0120	0.0520	0.0314		0.7993	0.1612	0.0081	0.0516
SVM(RBF)	0.1587	0.0730	0.0176	0.0001	0.0045	0.0797	0.0601	0.7993		0.1934	0.0049	0.0011
SVM(Linear)	0.2685	0.0684	0.0035	0.0003	0.0166	0.1393	0.0803	0.1612	0.1934		0.0049	0.0255
SVM(Polynomial)	0.1011	0.4216	0.5184	0.0059	0.6744	0.5189	0.4549	0.0081	0.0049	0.0049		0.3234
Naïve	0.5857	0.8562	0.5658	0.0016	0.2205	0.7701	0.7936	0.0516	0.0011	0.0255	0.3234	

Table C-1: T-tests for the Australian dataset. The case of the overall accuracy

Australian - Majority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0842	0.0703	0.1483	0.0046	0.6042	0.1708	0.0426	0.0136	0.0369	0.9255	0.0003
LDA	0.0842		0.0068	0.3395	0.0052	0.8601	0.8252	0.0114	0.0056	0.0087	0.5924	0.0005
QDA	0.0703	0.0068		0.0235	0.1029	0.1769	0.0225	0.0510	0.0224	0.0754	0.2643	0.0027
3-NN	0.1483	0.3395	0.0235		0.0037	0.3671	0.3096	0.0057	0.0004	0.0059	0.1384	0.0023
10-NN	0.0046	0.0052	0.1029	0.0037		0.0351	0.0052	0.7163	0.2660	1.0000	0.0629	0.0703
Neural	0.6042	0.8601	0.1769	0.3671	0.0351		0.9579	0.0523	0.0407	0.0431	0.7298	0.0163
MSD	0.1708	0.8252	0.0225	0.3096	0.0052	0.9579		0.0134	0.0041	0.0180	0.6478	0.0001
DT	0.0426	0.0114	0.0510	0.0057	0.7163	0.0523	0.0134		0.4895	0.3434	0.0224	0.2869
SVM(RBF)	0.0136	0.0056	0.0224	0.0004	0.2660	0.0407	0.0041	0.4895		0.2925	0.0104	0.6557
SVM(Linear)	0.0369	0.0087	0.0754	0.0059	1.0000	0.0431	0.0180	0.3434	0.2925		0.0296	0.0963
SVM(Polynomial)	0.9255	0.5924	0.2643	0.1384	0.0629	0.7298	0.6478	0.0224	0.0104	0.0296		0.0046
Naïve	0.0003	0.0005	0.0027	0.0023	0.0703	0.0163	0.0001	0.2869	0.6557	0.0963	0.0046	

Table C-2: T – tests for the Australian dataset. The case of the majority accuracy

Australian - Minority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		1.0000	0.0001	0.0059	0.0022	0.6744	0.2719	0.9135	0.2412	0.3626	0.1137	0.0012
LDA	1.0000		0.0001	0.0071	0.0043	0.6662	0.2500	0.9198	0.2958	0.4008	0.1443	0.0021
QDA	0.0001	0.0001		0.1406	0.0727	0.0505	0.7405	0.0702	0.2241	0.2283	0.5070	0.0278
3-NN	0.0059	0.0071	0.1406		0.7078	0.0042	0.1438	0.0100	0.0015	0.0021	0.0049	0.2471
10-NN	0.0022	0.0043	0.0727	0.7078		0.0130	0.1902	0.0000	0.0003	0.0013	0.0001	0.4525
Neural	0.6744	0.6662	0.0505	0.0042	0.0130		0.1912	0.9056	0.4194	0.5402	0.2659	0.0022
MSD	0.2719	0.2500	0.7405	0.1438	0.1902	0.1912		0.4387	0.6989	0.6436	0.9675	0.0761
DT	0.9135	0.9198	0.0702	0.0100	0.0000	0.9056	0.4387		0.3023	0.3661	0.0607	0.0010
SVM(RBF)	0.2412	0.2958	0.2241	0.0015	0.0003	0.4194	0.6989	0.3023		0.7804	0.2443	0.0000
SVM(Linear)	0.3626	0.4008	0.2283	0.0021	0.0013	0.5402	0.6436	0.3661	0.7804		0.2967	0.0000
SVM(Polynomial)	0.1137	0.1443	0.5070	0.0049	0.0001	0.2659	0.9675	0.0607	0.2443	0.2967		0.0002
Naïve	0.0012	0.0021	0.0278	0.2471	0.4525	0.0022	0.0761	0.0010	0.0000	0.0000	0.0002	

Table C-3: T-tests for the Australian dataset. The case of the minority accuracy

Australian - AUC												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.1590	0.0048	0.0045	0.0540	0.1655	0.1665	0.0205	0.0012	0.0000	0.0000	0.0144
LDA	0.1590		0.0002	0.0032	0.0341	0.0758	0.0099	0.0193	0.0007	0.0000	0.0000	0.0103
QDA	0.0048	0.0002		0.0506	0.6693	0.1207	0.0053	0.1560	0.0040	0.0002	0.0000	0.3469
3-NN	0.0045	0.0032	0.0506		0.0020	0.0090	0.0101	0.4087	0.1626	0.0357	0.0004	0.0180
10-NN	0.0540	0.0341	0.6693	0.0020		0.1324	0.1161	0.2831	0.0024	0.0003	0.0000	0.3497
Neural	0.1655	0.0758	0.1207	0.0090	0.1324		0.6239	0.0524	0.0022	0.0002	0.0000	0.0308
MSD	0.1665	0.0099	0.0053	0.0101	0.1161	0.6239		0.0368	0.0015	0.0001	0.0000	0.0367
DT	0.0205	0.0193	0.1560	0.4087	0.2831	0.0524	0.0368		0.0591	0.0085	0.0004	0.4185
SVM(RBF)	0.0012	0.0007	0.0040	0.1626	0.0024	0.0022	0.0015	0.0591		0.1768	0.0048	0.0017
SVM(Linear)	0.0000	0.0000	0.0002	0.0357	0.0003	0.0002	0.0001	0.0085	0.1768		0.0049	0.0001
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0004	0.0000	0.0000	0.0000	0.0004	0.0048	0.0049		0.0000
Naïve	0.0144	0.0103	0.3469	0.0180	0.3497	0.0308	0.0367	0.4185	0.0017	0.0001	0.0000	

Table C-4: T-tests for the Australian dataset. The case of the AUC

German - Overall												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0013	0.0111	0.0013	0.0000	0.0007	1.0000	0.0002	0.0005	1.0000	0.0000	0.4629
LDA	0.0013		0.9102	0.2963	0.0468	0.1404	0.0037	0.1154	0.4543	0.0498	0.0009	0.0102
QDA	0.0111	0.9102		0.3476	0.0985	0.1743	0.0162	0.1402	0.4774	0.0939	0.0063	0.0243
3-NN	0.0013	0.2963	0.3476		0.0985	0.7094	0.0001	0.8049	0.5961	0.0030	0.0018	0.0000
10-NN	0.0000	0.0468	0.0985			0.5890	0.0000	0.4715	0.0184	0.0000	0.0054	0.0000
Neural	0.0007	0.1404	0.1743	0.7094	0.5890		0.0051	0.8516	0.3571	0.0038	0.0698	0.0086
MSD	1.0000	0.0037	0.0162	0.0001	0.0000	0.0051		0.0029	0.0010	1.0000	0.0000	0.3691
DT	0.0002	0.1154	0.1402	0.8049	0.4715	0.8516	0.0029		0.4800	0.0027	0.0519	0.0022
SVM(RBF)	0.0005	0.4543	0.4774	0.5961	0.0184	0.3571	0.0010	0.4800		0.0022	0.0027	0.0024
SVM(Linear)	1.0000	0.0498	0.0939	0.0030	0.0000	0.0038	1.0000	0.0027	0.0022		0.0001	0.4175
SVM(Polynomial)	0.0000	0.0009	0.0063	0.0018	0.0054	0.0698	0.0000	0.0519	0.0027	0.0001		0.0000
Naïve	0.4629	0.0102	0.0243	0.0000	0.0000	0.0086	0.3691	0.0022	0.0024	0.4175	0.0000	

Table C-5: T-tests for the German dataset. The case of the overall accuracy

German - Majority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0021	0.0001	0.0022	0.0909	0.3666	0.0000	0.0019	0.0000	0.0103
LDA	0.0000		0.0027	0.0000	0.0000	0.0007	0.0000	0.0000	0.0000	0.0000	0.0017	0.0000
QDA	0.0000	0.0027		0.0000	0.0000	0.0021	0.0000	0.0000	0.0000	0.0000	0.0206	0.0001
3-NN	0.0021	0.0000	0.0000		0.0088	0.0006	0.0121	0.0002	0.0000	0.9302	0.0000	0.0000
10-NN	0.0001	0.0000	0.0000	0.0088		0.0001	0.0006	0.0001	0.0000	0.0030	0.0000	0.0000
Neural	0.0022	0.0007	0.0021	0.0006	0.0001		0.0015	0.0377	0.0000	0.0005	0.3676	0.2612
MSD	0.0909	0.0000	0.0000	0.0121	0.0006	0.0015		0.1212	0.0000	0.0120	0.0000	0.0018
DT	0.3666	0.0000	0.0000	0.0002	0.0001	0.0377	0.1212		0.0000	0.0019	0.0001	0.0371
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
SVM(Linear)	0.0019	0.0000	0.0000	0.9302	0.0030	0.0005	0.0120	0.0019	0.0000		0.0000	0.0000
SVM(Polynomial)	0.0000	0.0017	0.0206	0.0000	0.0000	0.3676	0.0000	0.0001	0.0000	0.0000		0.0007
Naïve	0.0103	0.0000	0.0001	0.0000	0.0000	0.2612	0.0018	0.0371	0.0000	0.0000	0.0007	

Table C-6: T-tests for the German dataset. The case of the majority accuracy

German - Minority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0000	0.0000	0.8768	0.1051	0.0012	0.0000	0.0003	0.4841	0.0000
LDA	0.0000		0.0029	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
QDA	0.0000	0.0029		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0014
3-NN	0.0000	0.0000	0.0000		0.0001	0.0008	0.0004	0.0064	0.0000	0.0004	0.0000	0.0000
10-NN	0.0000	0.0000	0.0000			0.0000	0.0000	0.0001	0.0001	0.0000	0.0000	0.0000
Neural	0.8768	0.0000	0.0000	0.0008	0.0000		0.3501	0.0237	0.0000	0.0418	0.6826	0.0042
MSD	0.1051	0.0000	0.0000	0.0004	0.0000	0.3501		0.0590	0.0000	0.1244	0.6774	0.0001
DT	0.0012	0.0000	0.0000	0.0064	0.0001	0.0237	0.0590		0.0000	0.4226	0.0194	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
SVM(Linear)	0.0003	0.0000	0.0000	0.0004	0.0000	0.0418	0.1244	0.4226	0.0000		0.0667	0.0000
SVM(Polynomial)	0.4841	0.0000	0.0000	0.0000	0.0000	0.6826	0.6774	0.0194	0.0000	0.0667		0.0001
Naïve	0.0000	0.0000	0.0014	0.0000	0.0000	0.0042	0.0001	0.0000	0.0000	0.0000	0.0001	

Table C-7: T-tests for the German dataset. The case of the minority accuracy

German - AUC												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0028	0.1536	0.0003	0.0005	0.0000	0.7329	0.0000	0.0000	0.0000	0.0000	0.0076
LDA	0.0028		0.0019	0.0001	0.0003	0.0000	0.1116	0.0000	0.0000	0.0000	0.0000	0.0219
QDA	0.1536	0.0019		0.0005	0.0020	0.0000	0.1392	0.0000	0.0000	0.0000	0.0000	0.0041
3-NN	0.0003	0.0001	0.0005		0.1796	0.9093	0.0002	0.0033	0.0000	0.0065	0.0000	0.0000
10-NN	0.0005	0.0003	0.0020			0.4138	0.0004	0.0001	0.0000	0.0000	0.0000	0.0000
Neural	0.0000	0.0000	0.0000	0.9093	0.4138		0.0001	0.0023	0.0000	0.0120	0.0004	0.0001
MSD	0.7329	0.1116	0.1392	0.0002	0.0004	0.0001		0.0000	0.0000	0.0000	0.0000	0.0086
DT	0.0000	0.0000	0.0000	0.0033	0.0001	0.0023	0.0000		0.0000	0.3282	0.2211	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000	0.0002	0.0000
SVM(Linear)	0.0000	0.0000	0.0000	0.0065	0.0000	0.0120	0.0000	0.3282	0.0000		0.0540	0.0000
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0004	0.0000	0.2211	0.0002	0.0540		0.0000
Naïve	0.0076	0.0219	0.0041	0.0000	0.0000	0.0001	0.0086	0.0000	0.0000	0.0000	0.0000	

Table C-8: T-tests for the German dataset. The case of the AUC

SPSS - Overall												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0093	0.0031	0.0002	0.0615	0.0011	0.6221	0.0172	0.4371	0.7730	0.0000	0.0003
LDA	0.0093		0.6281	0.0104	0.9278	0.0530	0.0506	0.6522	0.0307	0.0671	0.0025	0.0059
QDA	0.0031	0.6281		0.0008	0.4845	0.0590	0.0065	0.7932	0.0016	0.0064	0.0001	0.0036
3-NN	0.0002	0.0104	0.0008		0.0000	0.0988	0.0001	0.0005	0.0000	0.0000	0.1339	0.9548
10-NN	0.0615	0.9278	0.4845	0.0000		0.0223	0.0432	0.1211	0.0010	0.0013	0.0000	0.0020
Neural	0.0011	0.0530	0.0590	0.0988	0.0223		0.0003	0.0568	0.0000	0.0009	0.0100	0.1202
MSD	0.6221	0.0506	0.0065	0.0001	0.0432	0.0003		0.0033	0.1000	0.5579	0.0000	0.0001
DT	0.0172	0.6522	0.7932	0.0005	0.1211	0.0568	0.0033		0.0001	0.0024	0.0002	0.0017
SVM(RBF)	0.4371	0.0307	0.0016	0.0000	0.0010	0.0000	0.1000	0.0001		0.6144	0.0000	0.0000
SVM(Linear)	0.7730	0.0671	0.0064	0.0000	0.0013	0.0009	0.5579	0.0024	0.6144		0.0000	0.0002
SVM(Polynomial)	0.0000	0.0025	0.0001	0.1339	0.0000	0.0100	0.0000	0.0002	0.0000	0.0000		0.4275
Naïve	0.0003	0.0059	0.0036	0.9548	0.0020	0.1202	0.0001	0.0017	0.0000	0.0002	0.4275	

Table C-9: T-tests for the SPSS dataset. The case of the overall accuracy

SPSS - Majority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0003	0.2165	0.0047	0.0042	0.0055	0.0000	0.0117	0.0000	0.0000
LDA	0.0000		0.0253	0.0003	0.0000	0.0007	0.0000	0.0003	0.0000	0.0000	0.0362	0.0297
QDA	0.0000	0.0253		0.0371	0.0001	0.0970	0.0001	0.0159	0.0000	0.0010	0.8643	0.0042
3-NN	0.0003	0.0003	0.0371		0.0000	0.9743	0.0039	0.2950	0.0000	0.0960	0.0055	0.0000
10-NN	0.2165	0.0000	0.0001	0.0000		0.0067	0.1184	0.0074	0.0001	0.0243	0.0000	0.0000
Neural	0.0047	0.0007	0.0970	0.9743	0.0067		0.0622	0.6482	0.0003	0.1804	0.0950	0.0002
MSD	0.0042	0.0000	0.0001	0.0039	0.1184	0.0622		0.0856	0.0000	0.3727	0.0002	0.0000
DT	0.0055	0.0003	0.0159	0.2950	0.0074	0.6482	0.0856		0.0001	0.3743	0.0163	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0001	0.0003	0.0000	0.0001		0.0001	0.0000	0.0000
SVM(Linear)	0.0117	0.0000	0.0010	0.0960	0.0243	0.1804	0.3727	0.3743	0.0001		0.0037	0.0000
SVM(Polynomial)	0.0000	0.0362	0.8643	0.0055	0.0000	0.0950	0.0002	0.0163	0.0000	0.0037		0.0003
Naïve	0.0000	0.0297	0.0042	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0003	

Table C-10: T-tests for the SPSS dataset. The case of the majority accuracy

SPSS - Minority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0443	0.2048	0.5458	0.0003	0.2740	0.0345	0.0031	0.2664	0.0002
LDA	0.0000		0.0019	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0002	0.0000	0.0293
QDA	0.0000	0.0019		0.0004	0.0003	0.0012	0.0000	0.0012	0.0000	0.1181	0.0006	0.5311
3-NN	0.0443	0.0000	0.0004		0.1012	0.2202	0.0061	0.0027	0.3636	0.0002	0.0064	0.0000
10-NN	0.2048	0.0000	0.0003	0.1012		0.7190	0.0136	0.0506	0.3022	0.0000	0.8760	0.0000
Neural	0.5458	0.0001	0.0012	0.2202	0.7190		0.0947	0.2582	0.4188	0.0314	0.7545	0.0055
MSD	0.0003	0.0000	0.0000	0.0061	0.0136	0.0947		0.2975	0.0013	0.1045	0.0257	0.0022
DT	0.2740	0.0000	0.0012	0.0027	0.0506	0.2582	0.2975		0.0003	0.0231	0.0365	0.0002
SVM(RBF)	0.0345	0.0000	0.0000	0.3636	0.3022	0.4188	0.0013	0.0003		0.0000	0.2509	0.0000
SVM(Linear)	0.0031	0.0002	0.1181	0.0002	0.0000	0.0314	0.1045	0.0231	0.0000		0.0006	0.0015
SVM(Polynomial)	0.2664	0.0000	0.0006	0.0064	0.8760	0.7545	0.0257	0.0365	0.2509	0.0006		0.0000
Naïve	0.0002	0.0293	0.5311	0.0000	0.0000	0.0055	0.0022	0.0002	0.0000	0.0015	0.0000	

Table C-11: T-tests for the SPSS dataset. The case of the minority accuracy

SPSS – AUC												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.2052	0.0002	0.0000	0.0007	0.0000	0.0207	0.0000	0.0000	0.0000	0.0000	0.0004
LDA	0.2052		0.0001	0.0000	0.0021	0.0000	0.3959	0.0001	0.0000	0.0000	0.0000	0.0010
QDA	0.0002	0.0001		0.0003	0.0418	0.0004	0.0053	0.0007	0.0000	0.0000	0.0000	0.0071
3-NN	0.0000	0.0000	0.0003		0.0002	0.2571	0.0000	0.4222	0.0000	0.0062	0.0000	0.0765
10-NN	0.0007	0.0021	0.0418	0.0002		0.0514	0.0034	0.0003	0.0000	0.0000	0.0000	0.0164
Neural	0.0000	0.0000	0.0004	0.2571	0.0514		0.0000	0.6366	0.0001	0.0029	0.0000	0.5123
MSD	0.0207	0.3959	0.0053	0.0000	0.0034	0.0000		0.0001	0.0000	0.0000	0.0000	0.0014
DT	0.0000	0.0001	0.0007	0.4222	0.0003	0.6366	0.0001		0.0000	0.0006	0.0000	0.0165
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000		0.0038	0.0000	0.0000
SVM(Linear)	0.0000	0.0000	0.0000	0.0062	0.0000	0.0029	0.0000	0.0006	0.0038		0.0000	0.0002
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000
Naïve	0.0004	0.0010	0.0071	0.0765	0.0164	0.5123	0.0014	0.0165	0.0000	0.0002	0.0000	

Table C-12: T-tests for the SPSS dataset. The case of the AUC

SME - Overall												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0019	0.0023	0.0025	0.0982	0.0058	0.0207	0.9023	0.0000	0.0000
LDA	0.0000		0.0029	0.0000	0.0000	0.0003	0.0001	0.0000	0.0003	0.0000	0.0383	0.0002
QDA	0.0000	0.0029		0.0010	0.0011	0.0002	0.0032	0.0001	0.0420	0.0000	0.0005	0.0000
3-NN	0.0019	0.0000	0.0010		0.4344	0.3682	0.3142	0.1262	0.9445	0.0001	0.0000	0.0000
10-NN	0.0023	0.0000	0.0011	0.4344		0.4725	0.4479	0.1949	0.7797	0.0000	0.0000	0.0000
Neural	0.0025	0.0003	0.0002	0.3682	0.4725		0.9390	0.9706	0.4887	0.0318	0.0000	0.0000
MSD	0.0982	0.0001	0.0032	0.3142	0.4479	0.9390		0.9086	0.4775	0.1000	0.0001	0.0000
DT	0.0058	0.0000	0.0001	0.1262	0.1949	0.9706	0.9086		0.4587	0.0005	0.0000	0.0000
SVM(RBF)	0.0207	0.0003	0.0420	0.9445	0.7797	0.4887	0.4775	0.4587		0.0135	0.0004	0.0000
SVM(Linear)	0.9023	0.0000	0.0000	0.0001	0.0000	0.0318	0.1000	0.0005	0.0135		0.0000	0.0000
SVM(Polynomial)	0.0000	0.0383	0.0005	0.0000	0.0000	0.0000	0.0001	0.0000	0.0004	0.0000		0.0000
Naïve	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-13: T-tests for the SME dataset. The case of the overall accuracy

SME - Majority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.1250	0.0017	0.0002	0.1062	0.0786	0.0251	0.0462	0.0000	0.0000
LDA	0.0000		0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0003
QDA	0.0000	0.0003		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0087	0.0000
3-NN	0.1250	0.0000	0.0000		0.0003	0.0003	0.0569	0.0002	0.0445	0.0002	0.0000	0.0000
10-NN	0.0017	0.0000	0.0000	0.0003		0.0000	0.0068	0.0000	0.8310	0.0000	0.0000	0.0000
Neural	0.0002	0.0000	0.0000	0.0003	0.0000		0.5005	0.1617	0.0004	0.0105	0.0000	0.0000
MSD	0.1062	0.0000	0.0000	0.0569	0.0068	0.5005		0.8540	0.0232	0.6187	0.0001	0.0000
DT	0.0786	0.0000	0.0000	0.0002	0.0000	0.1617	0.8540		0.0006	0.4575	0.0000	0.0000
SVM(RBF)	0.0251	0.0000	0.0000	0.0445	0.8310	0.0004	0.0232	0.0006		0.0023	0.0000	0.0000
SVM(Linear)	0.0462	0.0000	0.0000	0.0002	0.0000	0.0105	0.6187	0.4575	0.0023		0.0000	0.0000
SVM(Polynomial)	0.0000	0.0002	0.0087	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000		0.0000
Naïve	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-14: T-tests for the SME dataset. The case of the majority accuracy

SME - Minority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000		0.0000	0.0000	0.6221	0.5549	0.0036	0.0056	0.7822	0.0599	0.0000
LDA	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0424
QDA	0.0000	0.0000		0.0000	0.0000	0.0000	0.0002	0.0000	0.0002	0.0002	0.0001	0.0001
3-NN	0.0000	0.0000	0.0000		0.0000	0.0002	0.0095	0.0000	0.3405	0.0000	0.0000	0.0000
10-NN	0.0000	0.0000	0.0000	0.0000		0.0001	0.0025	0.0000	0.7883	0.0000	0.0000	0.0000
Neural	0.6221	0.0000	0.0000	0.0002	0.0001		0.7484	0.0369	0.0091	0.6146	0.2154	0.0000
MSD	0.5549	0.0000	0.0002	0.0095	0.0025	0.7484		0.2674	0.0376	0.5177	0.7314	0.0000
DT	0.0036	0.0000	0.0000	0.0000	0.0000	0.0369	0.2674		0.0322	0.0000	0.0039	0.0000
SVM(RBF)	0.0056	0.0000	0.0002	0.3405	0.7883	0.0091	0.0376	0.0322		0.0042	0.0155	0.0000
SVM(Linear)	0.7822	0.0000	0.0002	0.0000	0.0000	0.6146	0.5177	0.0000	0.0042		0.0019	0.0000
SVM(Polynomial)	0.0599	0.0000	0.0001	0.0000	0.0000	0.2154	0.7314	0.0039	0.0155	0.0019		0.0000
Naïve	0.0000	0.0424	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-15: T-tests for the SME dataset. The case of the minority accuracy

SME – AUC												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.6788	0.6571	0.0000	0.1147	0.0095	0.0000	0.0000	0.0000	0.0000	0.0000	0.0145
LDA	0.6788		0.8276	0.0000	0.1258	0.0160	0.0000	0.0000	0.0000	0.0000	0.0000	0.0086
QDA	0.6571	0.8276		0.0000	0.1269	0.0290	0.0000	0.0000	0.0000	0.0000	0.0000	0.0060
3-NN	0.0000	0.0000	0.0000		0.4960	0.0003	0.5234	0.0000	0.0000	0.0000	0.0000	0.0000
10-NN	0.1147	0.1258	0.1269	0.4960		0.1730	0.4712	0.5281	0.0705	0.1244	0.0891	0.1740
Neural	0.0095	0.0160	0.0290	0.0003			0.0001	0.0000	0.0000	0.0000	0.0000	0.7650
MSD	0.0000	0.0000	0.0000	0.5234	0.4712	0.0001		0.0000	0.0000	0.0000	0.0000	0.0000
DT	0.0000	0.0000	0.0000	0.0000	0.5281	0.0000	0.0000		0.0087	0.9462	0.0000	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0705	0.0000	0.0000	0.0087		0.0046	0.0000	0.0000
SVM(Linear)	0.0000	0.0000	0.0000	0.0000	0.1244	0.0000	0.0000	0.9462	0.0046		0.0000	0.0000
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0000	0.0891	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000
Naïve	0.0145	0.0086	0.0060	0.0000	0.1740	0.7650	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-16: T-tests for the SME dataset. The case of the AUC

Greek - Overall											
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0000	0.0000	0.0002	0.0253	0.0000	0.0001	0.0000	0.0000	0.0000
LDA	0.0000		0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
QDA	0.0000	0.0001		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3-NN	0.0000	0.0000	0.0000		0.0014	0.2224	0.0000	0.0000	0.0000	0.0000	0.0000	0.0063
10-NN	0.0000	0.0000	0.0000	0.0014		0.0086	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Neural	0.0002	0.0000	0.0000	0.2224	0.0086		0.0002	0.0001	0.0001	0.0000	0.0000	0.7938
MSD	0.0253	0.0000	0.0000	0.0000	0.0000	0.0002		0.0000	0.0004	0.0000	0.0000	0.0000
DT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000		0.2180	0.0003	0.0045	0.0000
SVM(RBF)	0.0001	0.0000	0.0000	0.0000	0.0000	0.0001	0.0004	0.2180		0.0326	0.0260	0.0000
SVM(Linear)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0003	0.0326		0.9084	0.0000
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0045	0.0260	0.9084		0.0000
Naïve	0.0000	0.0000	0.0000	0.0063	0.0000	0.7938	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-17: T-tests for the Greek dataset. The case of the Overall Accuracy

Greek - Minority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0000	0.0000	0.0001	0.0004	0.0054	0.0000	0.0093	0.6514	0.0000
LDA	0.0000		0.0018	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
QDA	0.0000	0.0018		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3-NN	0.0000	0.0000	0.0000		0.0007	0.0034	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10-NN	0.0000	0.0000	0.0000	0.0007		0.0016	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Neural	0.0001	0.0000	0.0000	0.0034	0.0016		0.0001	0.0001	0.0000	0.0001	0.0001	0.8812
MSD	0.0004	0.0000	0.0000	0.0000	0.0000	0.0001		0.0001	0.0000	0.6491	0.0238	0.0000
DT	0.0054	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001		0.0000	0.0000	0.0010	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000
SVM(Linear)	0.0093	0.0000	0.0000	0.0000	0.0000	0.0001	0.6491	0.0000	0.0000		0.0032	0.0000
SVM(Polynomial)	0.6514	0.0000	0.0000	0.0000	0.0000	0.0001	0.0238	0.0010	0.0000	0.0032		0.0000
Naïve	0.0000	0.0000	0.0000	0.0000	0.0000	0.8812	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-18: T-tests for the Greek dataset. The case of the Minority Accuracy

Greek - Majority												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.0000	0.0000	0.0000	0.0000	0.0001	0.0022	0.0000	0.8241	0.8296	0.4005	0.0000
LDA	0.0000		0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
QDA	0.0000	0.0001		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3-NN	0.0000	0.0000	0.0000		0.0003	0.0158	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
10-NN	0.0000	0.0000	0.0000	0.0003		0.0016	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Neural	0.0001	0.0000	0.0000	0.0158	0.0016		0.0001	0.0002	0.0001	0.0001	0.0000	0.5246
MSD	0.0022	0.0000	0.0000	0.0000	0.0000	0.0001		0.0000	0.0522	0.0522	0.1039	0.0000
DT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0002	0.0000		0.0000	0.0000	0.0000	0.0000
SVM(RBF)	0.8241	0.0000	0.0000	0.0000	0.0000	0.0001	0.0522	0.0000		1.0000	0.7263	0.0000
SVM(Linear)	0.8296	0.0000	0.0000	0.0000	0.0000	0.0001	0.0522	0.0000	1.0000		0.7976	0.0000
SVM(Polynomial)	0.4005	0.0000	0.0000	0.0000	0.0000	0.0000	0.1039	0.0000	0.7263	0.7976		0.0000
Naïve	0.0000	0.0000	0.0000	0.0000	0.0000	0.5246	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-19: T-tests for the Greek dataset. The case of the Majority Accuracy

Greek - AUC												
	Logit	LDA	QDA	3-NN	10-NN	Neural	MSD	DT	SVM(RBF)	SVM(Linear)	SVM(Polynomial)	Naïve
Logit		0.2613	0.0614	0.0000	0.0005	0.0007	0.5893	0.0000	0.0000	0.0000	0.0000	0.0006
LDA	0.2613		0.9515	0.0002	0.0150	0.3182	0.2278	0.0000	0.0000	0.0000	0.0000	0.0427
QDA	0.0614	0.9515		0.0000	0.0037	0.1126	0.0308	0.0000	0.0000	0.0000	0.0000	0.0099
3-NN	0.0000	0.0002	0.0000		0.0093	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0004
10-NN	0.0005	0.0150	0.0037			0.0346	0.0000	0.0000	0.0000	0.0000	0.0000	0.3326
Neural	0.0007	0.3182	0.1126	0.0000	0.0346		0.0005	0.0000	0.0000	0.0000	0.0000	0.0732
MSD	0.5893	0.2278	0.0308	0.0000	0.0000	0.0005		0.0000	0.0000	0.0000	0.0000	0.0003
DT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.3745	0.0000	0.0000	0.0000
SVM(RBF)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.3745		0.0000	0.0001	0.0000
SVM(Linear)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0330	0.0000
SVM(Polynomial)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0330		0.0000
Naïve	0.0006	0.0427	0.0099	0.0004	0.3326	0.0732	0.0003	0.0000	0.0000	0.0000	0.0000	

Table C-20: T-tests for the Greek dataset. The case of the AUC

Australian Data	No of Variables																					
	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Chi Square	1.000	0.168	0.193	0.758	0.193	0.147	0.297	0.351	0.397	0.479	0.229	0.460	0.225	0.664	0.785	0.266	0.380	0.373	0.373	0.086	0.359	0.343
Gain ratio	0.343	0.168	0.193	0.237	0.133	0.209	0.297	0.115	0.260	0.685	0.357	0.095	0.225	0.897	0.785	0.266	0.380	0.830	0.914	0.415	0.546	0.343
ReliefF	0.193	0.193	0.343	0.678	0.104	0.343	0.343	0.443	0.309	0.343	0.394	0.394	0.072	0.836	0.448	0.716	0.406	0.713	0.918	0.415	0.546	0.343
MSD	0.604	0.572	0.604	0.546	0.450	0.498	0.498	0.595	0.595	0.604	0.869	0.753	0.785	0.943	0.940	0.627	0.553	0.843	0.843	0.712	0.946	0.876

Table C-21: T – tests for the Australian dataset. Comparisons of all the methods

German Data	No of Variables															
	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Chi Squared	0.000	0.154	0.000	0.004	0.081	0.100	0.059	0.912	0.120	0.153	0.000	0.000	0.000	0.000	0.092	0.073
Gain	0.001	0.212	0.000	0.002	0.569	0.175	0.162	0.557	0.136	0.091	0.000	0.000	0.000	0.000	0.189	0.098
ReliefF	0.000	0.129	0.001	0.005	0.094	0.131	0.258	0.377	0.057	0.277	0.000	0.000	0.000	0.000	0.487	0.620
MSD	0.028	0.175	0.022	0.024	0.171	0.780	0.017	0.398	0.011	0.260	0.000	0.000	0.000	0.000	0.003	0.001

Table C-22: T – tests for the German dataset. Comparisons of all the methods

German Data	No of Variables														
	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
Chi Squared	0.047	0.038	0.042	0.095	0.075	0.017	0.016	0.059	0.044	0.040	0.034	0.031	0.035	0.015	0.002
Gain	0.243	0.118	0.099	0.020	0.015	0.025	0.025	0.024	0.041	0.029	0.034	0.024	0.055	0.017	0.000
ReliefF	0.545	0.395	0.273	0.281	0.115	0.129	0.127	0.100	0.038	0.068	0.073	0.067	0.071	0.037	0.001
MSD	0.002	0.254	0.059	0.037	0.113	0.029	0.031	0.030	0.053	0.026	0.028	0.024	0.023	0.021	0.006

Table C-23: T – tests for the German dataset. Comparisons of all the methods (Contin.)

US Data	No of Variables																
	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65
Chi_Squared	1.000	0.828	0.708	0.025	0.047	0.127	0.041	0.710	0.507	0.061	0.086	0.034	0.937	0.930	0.755	0.461	0.438
Gain	0.782	0.546	0.662	0.158	0.156	0.080	0.041	0.710	0.475	0.450	0.552	0.265	1.000	0.796	0.670	0.443	0.832
ReliefF	0.487	0.079	0.079	0.763	0.769	0.191	0.032	0.662	0.662	0.009	0.360	0.446	0.694	0.201	0.766	0.556	0.853
MSD	0.343	0.343	0.343	0.343	0.209	-	0.029	0.472	-	0.121	0.168	0.182	0.168	0.242	0.168	0.177	0.168

Table C-24: T – tests for the US dataset. Comparisons of all the methods

US Data	No of Variables														
	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
Chi_Squared	0.767	0.664	0.830	0.267	0.438	0.584	0.280	0.479	0.601	0.029	0.058	0.279	0.168	0.798	0.443
Gain	0.830	0.619	0.888	0.438	0.350	0.527	0.363	0.872	0.868	0.472	0.033	0.050	0.016	0.798	0.279
ReliefF	0.850	0.443	0.147	0.229	0.041	0.038	0.035	0.111	0.045	0.397	0.591	0.343	0.070	0.678	0.726
MSD	0.173	0.168	0.173	0.173	0.168	0.170	0.333	0.170	0.133	0.174	0.174	0.187	0.115	0.328	0.225

Table C-25: T – tests for the US dataset. Comparisons of all the methods (Contin.)

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression		0.2620	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model	0.2620		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.3210	0.3253	0.2654	0.2361	0.2015
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.3210		0.2354	0.1105	0.0988	0.5644
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.3253	0.2354		0.0988	0.1122	0.3541
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.2654	0.1105	0.0988		0.0855	0.3666
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.2361	0.0988	0.1122	0.0855		0.4111
MSD – Balancing Constraint	0.0000	0.0000	0.2015	0.5644	0.3541	0.3666	0.4111	

Table C-26: T – tests for Dataset 1. The case of minority accuracy.

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression		0.3220	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model	0.3220		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.2566	0.6200	0.4423	0.3699	0.1808
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.2566		0.3832	0.0889	0.3155	0.1777
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.6200	0.3832		0.1988	0.2626	0.5552
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.4423	0.0889	0.1988		0.2988	0.3666
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.3699	0.3155	0.2626	0.2988		0.1550
MSD – Balancing Constraint	0.0000	0.0000	0.1808	0.1777	0.5552	0.3666	0.1550	

Table C-27: T – tests for Dataset 1. The case of majority accuracy.

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression		0.1211	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model	0.1211		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.2955	0.1121	0.1221	0.0998	0.1825
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.2955		0.0988	0.8550	0.6556	0.3411
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.1121	0.0988		0.5911	0.3577	0.3389
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.1221	0.8550	0.5911		0.7711	0.2551
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.0998	0.6556	0.3577	0.7711		0.2211
MSD – Balancing Constraint	0.0000	0.0000	0.1825	0.3411	0.3389	0.2551	0.2211	

Table 38: T - tests for Dataset 2. The case of the minority accuracy.

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression		0.0888	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model	0.0888		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.3011	0.2911	0.1988	0.0988	0.3252
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.3011		0.5121	0.6255	0.5998	0.0505
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.2911	0.5121		0.1788	0.1998	0.7002
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.1988	0.6255	0.1788		0.1122	0.2411
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.0988	0.5998	0.1998	0.1122		0.0998
MSD – Balancing Constraint	0.0000	0.0000	0.3252	0.0505	0.7002	0.2411	0.0998	

Table C-29: T – tests for Dataset 2. The case of majority accuracy.

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.0178	0.2100	0.1880	0.3223	0.2990
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.0178		0.7485	0.5901	0.2310	0.2620
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.2100	0.7485		0.8201	0.2877	0.2872
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.1880	0.5901	0.8201		0.3224	0.8285
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.3223	0.2310	0.2877	0.3224		0.3852
MSD – Balancing Constraint	0.0000	0.0000	0.2990	0.2620	0.2872	0.8285	0.3852	

Table C-30: T –tests for Dataset 3. The case of minority accuracy.

	Logistic Regression	MSD – Basic Model	MSD – Balancing Objective	MSD – Range Constraints: δ=0.001	MSD – Range Constraints: δ=0.0005	MSD – Range Constraints: δ=0.0001	MSD – Range Constraints: δ=0.00001	MSD – Balancing Constraint
Logistic Regression			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Basic Model			0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD – Balancing Objective	0.0000	0.0000		0.2211	0.0999	0.5228	0.6697	0.0877
MSD – Range Constraints: δ=0.001	0.0000	0.0000	0.2211		0.7485	0.0987	0.6255	0.3693
MSD – Range Constraints: δ=0.0005	0.0000	0.0000	0.0999	0.7485		0.1471	0.0875	0.2135
MSD – Range Constraints: δ=0.0001	0.0000	0.0000	0.5228	0.0987	0.1471		0.2365	0.3255
MSD – Range Constraints: δ=0.00001	0.0000	0.0000	0.6697	0.6255	0.0875	0.2365		0.4152
MSD – Balancing Constraint	0.0000	0.0000	0.0877	0.3693	0.2135	0.3255	0.4152	

Table C-31: T – tests for Dataset 3. The case of majority accuracy.

Greek - Overall								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0000	0.0000	0.0766	0.0000
Log	0.0000		0.0000	0.0000	0.0000	0.0000	0.0088	0.0000
MSD	0.0000	0.0000			0.0000	0.0000	0.0001	-
MSD - Balancing	0.0000	0.0000	0.0000		0.0000	0.0000	0.0000	0.0000
LDA	0.0000	0.0000	0.0000	0.0000		0.0004	0.0000	0.0000
QDA	0.0000	0.0000	0.0000	0.0000	0.0004		0.0000	0.0000
NN	0.0766	0.0088	0.0001	0.0000	0.0000	0.0000		0.0001
AUDA	0.0000	0.0000	-	0.0000	0.0000	0.0000	0.0001	

Table C-32: Greek dataset: The case of overall accuracy

Greek - Acc+								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0014	0.0016	0.0026	0.0000
Log	0.0000		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
MSD	0.0000	0.0000		0.0000	0.0000	0.0000	0.0013	0.0000
MSD - Balancing	0.0000	0.0000	0.0000		0.0852	0.8210	0.0000	0.0000
LDA	0.0014	0.0000	0.0000	0.0852		0.0033	0.0003	0.0001
QDA	0.0016	0.0000	0.0000	0.8210	0.0033		0.0003	0.0001
NN	0.0026	0.0000	0.0013	0.0000	0.0003	0.0003		0.0405
AUDA	0.0000	0.0000	0.0000	0.0000	0.0001	0.0001	0.0405	

Table C-33: Greek dataset: The case of minority accuracy

Greek - Acc-								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Log	0.0000		0.0135	0.0000	0.0000	0.0000	0.0000	0.0000
MSD	0.0000	0.0135			0.0000	0.0000	0.0000	0.0000
MSD - Balancing	0.0000	0.0000	0.0000		0.4520	0.3254	0.0000	0.0000
LDA	0.0000	0.0000	0.0000	0.4520		0.0000	0.0000	0.0000
QDA	0.0000	0.0000	0.0000	0.3254	0.0000		0.0000	0.0000
NN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000		0.0654
AUDA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0654	

Table C-34: Greek dataset: The case of majority accuracy

Greek - AUC								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.1243	0.0000	0.0000	0.1348	0.1805	0.1207	0.0000
Log	0.1243		0.0000	0.0000	0.4339	0.7539	0.0417	0.0000
MSD	0.0000	0.0000		0.3254	0.0000	0.0000	0.1077	0.0009

MSD - Balancing	0.0000	0.0000	0.3254		0.0000	0.0000	0.2510	0.0565
LDA	0.1348	0.4339	0.0000	0.0000		0.0909	0.0255	0.0000
QDA	0.1805	0.7539	0.0000	0.0000	0.0909		0.0447	0.0000
NN	0.1207	0.0417	0.1077	0.2510	0.0255	0.0447		0.0280
AUDA	0.0000	0.0000	0.0009	0.0565	0.0000	0.0000	0.0280	

Table C-35: Greek dataset: The case of AUC

German - Overall								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0040	0.0055	0048	0.4262	0.9911	0.3204	0.0000
Log	0.0040		0.5408	0.5001	0.0009	0.0014	0.7915	0.0000
MSD	0.0055	0.5408		0.0001	0.0000	0.0000	0.9563	0.0000
MSD - Balancing	0.0048	0.5001	0.0001		0.0000	0.0000	0.9400	0.0000
LDA	0.4262	0.0009	0.0000	0.0000		0.1053	0.2460	0.0000
QDA	0.9911	0.0014	0.0000	0.0000	0.1053		0.3429	0.0000
NN	0.3204	0.7915	0.9563	0.9400	0.2460	0.3429		0.0024
AUDA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0024	

Table C-36: German dataset: The case of Overall Accuracy

German - Acc+								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001
Log	0.0000		0.6386	0.0000	0.0001	0.0004	0.3741	0.0000
MSD	0.0000	0.6386		0.0000	0.0000	0.0000	0.2696	0.0000
MSD - Balancing	0.0000	0.0000	0.0000		0.0000	0.0000	0.0021	0.0000
LDA	0.0000	0.0001	0.0000	0.0000		0.0004	0.0074	0.0000
QDA	0.0000	0.0004	0.0000	0.0000	0.0004		0.0610	0.0000
NN	0.0000	0.3741	0.2696	0.0021	0.0074	0.0610		0.0000
AUDA	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-37: German dataset: The case of Minority Accuracy

German - Acc-								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0030	0.0000	0.0000	0.0000	0.0014	0.0001
Log	0.0000		0.1369	0.0000	0.0000	0.0000	0.1594	0.0762
MSD	0.0030	0.1369		0.0210	0.0398	0.0796	0.5230	0.0469
MSD - Balancing	0.0000	0.0000	0.0210		0.0120	0.0231	0.0020	0.0000
LDA	0.0000	0.0000	0.0398	0.0120		0.0230	0.0032	0.0000
QDA	0.0000	0.0000	0.0796	0.0231	0.0230		0.0030	0.0000
NN	0.0014	0.1594	0.5230	0.0020	0.0032	0.0030		0.0421
AUDA	0.0001	0.0762	0.0469	0.0000	0.0000	0.0000	0.0421	

Table C38: German dataset: The case of majority Accuracy

German - AUC								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.2293	0.5820	0.2920	0.2553	0.6632	0.9861	0.0504
Log	0.2293		0.1900	0.7501	0.7568	0.1552	0.3800	0.0019
MSD	0.5820	0.1900		0.0000	0.0001	0.6164	0.6758	0.0028
MSD - Balancing	0.2920	0.7501	0.0000		0.1250	0.2011	0.3230	0.0001
LDA	0.2553	0.7568	0.0001	0.1250		0.0021	0.3982	0.0008
QDA	0.6632	0.1552	0.6164	0.2011	0.0021		0.7605	0.0055
NN	0.9861	0.3800	0.6758	0.3230	0.3982	0.7605		0.1632
AUDA	0.0504	0.0019	0.0028	0.0001	0.0008	0.0055	0.1632	

Table C39: German dataset: The case of AUC

SPSS - Overall								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Log	0.0000		0.0000	0.0000	0.0923	0.0009	0.0000	0.0022
MSD	0.0000	0.0000		0.0021	0.0016	0.0102	0.0305	0.4355
MSD - Balancing	0.0000	0.0000	0.0021		0.0251	0.0365	0.0362	0.2654
LDA	0.0000	0.0923	0.0016	0.0251		0.0103	0.0441	0.0036
QDA	0.0000	0.0009	0.0102	0.0365	0.0103		0.5446	0.1274
NN	0.0000	0.0000	0.0305	0.0362	0.0441	0.5446		0.3673
AUDA	0.0000	0.0022	0.4355	0.2654	0.0036	0.1274	0.3673	

Table C-40: SPSS dataset: The case of overall accuracy

SPSS - Acc+								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0051	0.0042	0.0012	0.0000	0.0000
Log	0.0000		0.5859	0.0000	0.0000	0.0000	0.0073	0.0000
MSD	0.0000	0.5859		0.0000	0.0000	0.0000	0.1708	0.0000
MSD - Balancing	0.0051	0.0000	0.0000		0.0021	0.0000	0.0000	0.0000
LDA	0.0042	0.0000	0.0000	0.0021		0.0021	0.0000	0.0000
QDA	0.0012	0.0000	0.0000	0.0000	0.0021		0.0000	0.0000
NN	0.0000	0.0073	0.1708	0.0000	0.0000	0.0000		0.0000
AUDA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-41: SPSS dataset: The case of minority accuracy

SPSS - Acc-								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0043	0.0000	0.0000	0.0000	0.0000	0.0001	0.0003
Log	0.0043		0.0000	0.0000	0.0000	0.0000	0.0003	0.0000
MSD	0.0000	0.0000		0.0012	0.1780	0.1993	0.0016	0.0000
MSD - Balancing	0.0000	0.0000	0.0012		0.0000	0.0000	0.0000	0.0000
LDA	0.0000	0.0000	0.1780	0.0000		1.0000	0.0009	0.0000
QDA	0.0000	0.0000	0.1993	0.0000	1.0000		0.0015	0.0000
NN	0.0001	0.0003	0.0016	0.0000	0.0009	0.0015		0.0000
AUDA	0.0003	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table C-42: SPSS dataset: The case of majority accuracy

SPSS - AUC								
	AUDA-WoE	Log	MSD	MSD - Balancing	LDA	QDA	NN	AUDA
AUDA-WoE		0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Log	0.0000		0.0000	0.0053	0.8904	0.0018	0.0000	0.8361
MSD	0.0000	0.0000		0.7565	0.0001	0.0000	0.0017	0.0004
MSD - Balancing	0.0000	0.0053	0.7565		0.0364	0.2654	0.5654	0.0350
LDA	0.0000	0.8904	0.0001	0.0364		0.0852	0.0050	0.8563
QDA	0.0000	0.0018	0.0000	0.2654	0.0852		0.0038	0.0207
NN	0.0000	0.0000	0.0017	0.5654	0.0050	0.0038		
AUDA	0.0000	0.8361	0.0004	0.0350	0.8563	0.1930	0.0207	

Table C-43: SPSS dataset: The case of AUC

APPENDIX D

Coding for the MP models used in our analysis
UTADIS

```
model UTADIS
!Uses US dataset
!Uses "mmxprs";

setparam("XPRS_MIPTOL",0.00000005)

declarations
 NM1=116      ! No observations in group 1
 NM2=160     ! No observations in group 2
 NL1=29                  !No of observations in Group 1-Holdout sample
 NL2=40             ! No observations in group 2
 NV=6       ! No of original variables
 NG=5        ! No of "grades" for each original variable
 NM=NM1+NM2   ! Total number of observations
 NL=NL1+NL2
 NN=(NV*NG)     ! Total no of variables
 NG1=NG-1
 M1=1..NM1
 M2=1..NM2
 L1=1..NL1
 L2=1..NL2
 M=1..NM
 L=1..NL
 V=1..NV
 G=1..NG
 G1=1..NG1
 N=1..NN
 X1: array(M1,N) of real  ! Group 1 observation, variable value
 X2: array(M2,N) of real  ! Group 2 observation, variable value
 Y1:array(L1,N) of real
 Y2:array(L2,N) of real
 D: real               ! Reject interval (one-sided)
 a0: mpvar              ! Constant in function
 a: array(V,G) of mpvar   ! Variable coefficient
 d1: array(M1) of mpvar   ! deviation of Group 1 observation
 d2: array(M2) of mpvar   ! deviation of Group 2 observation
 DT: array(M1) of real     ! Deviations of training sample obs. - for O/P only
 DH: array(L1) of real
 DT2: array(M2) of real     ! Deviations of training sample obs. - for O/P
only
 DH2: array(L2) of real
             ! No. training sample obs. correctly classified - O/P only

end-declarations

D:= 0.001
NTC:= 0
NHC:= 0
NHC1:= 0
NTC1:= 0
```

```
fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Liver\UTA_samples\5_Se
g\s_91.txt',F_INPUT)
 forall(i in M1) do
  forall(j in N) read(X1(i,j))
 end-do
fclose(F_INPUT)

fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Liver\UTA_samples\5_Se
g\s_92.txt',F_INPUT)
 forall(i in M2) do
  forall(j in N) read(X2(i,j))
 end-do
fclose(F_INPUT)

!Read in data-continuous variables then binary variables
fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Liver\UTA_samples\5_Se
g\h_91.txt',F_INPUT)
 forall(i in L1) do
  forall(j in N) read(Y1(i,j))
 end-do
fclose(F_INPUT)

fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Liver\UTA_samples\5_Se
g\h_92.txt',F_INPUT)
 forall(i in L2) do
  forall(j in N) read(Y2(i,j))
 end-do
fclose(F_INPUT)

! Constraints

! Group 1
forall(i in M1)
 CA(i):= sum(j in V, k in G)X1(i,((j-1)*NG)+k)*a(j,k) - a0 - d1(i) <= -D

! Group 2
forall(i in M2)
 CB(i):= sum(j in V, k in G)X2(i,((j-1)*NG)+k)*a(j,k) - a0 + d2(i) >= 0



! Normalisation
   CC:= sum(j in V) a(j,NG) = 1


! Constraints for monotone weights
forall(j in V, k in G1)
 CE(j,k):= a(j,k+1) - a(j,k) >= 0
forall(j in V)
a(j,1)=0
```

```
! Objective - Minimise sum of deviations
MSD:= sum(i in M1)d1(i) + sum(i in M2)d2(i)

setparam("XPRS_verbose",true)

minimize(MSD)

! Calculate deviation for each observation in training and holdout samples
forall (i in M1) do
  DT(i):=sum(j in V, k in G)X1(i,((j-1)*NG)+k)*getsol(a(j,k))-getsol(a0)
   if (DT(i)<0) then
    NTC:=NTC+1
   end-if
end-do

forall (i in M2) do
  DT2(i):=sum(j in V, k in G)X2(i,((j-1)*NG)+k)*getsol(a(j,k))-getsol(a0)
   if (DT2(i)>=0) then
    NTC1:=NTC1+1
   end-if
end-do

forall (i in L1) do
  DH(i):=sum(j in V, k in G)Y1(i,((j-1)*NG)+k)*getsol(a(j,k))-getsol(a0)
   if (DH(i)<0) then
    NHC:=NHC+1
   end-if
end-do

forall (i in L2) do
  DH2(i):=sum(j in V, k in G)Y2(i,((j-1)*NG)+k)*getsol(a(j,k))-getsol(a0)
   if (DH2(i)>=0) then
    NHC1:=NHC1+1
   end-if
end-do


! Print utility deviation for each observation in training and holdout samples
fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Liver\Results\UTA\out_U
TA_100.txt',F_OUTPUT)
writeln("Liver - Data")
writeln
writeln("Sum of deviations : ", getobjval)
writeln("Constant :", getsol(a0))
writeln
forall(j in V) do
forall(k in G)
  writeln("         Coefficient ", k , " : ", getsol(a(j,k)))
end-do

! Print utility deviation for each observation in training and holdout samples
writeln
writeln
writeln ("                  ")
```

```
writeln ("Group  Ob. No. ", "Utility Deviation Training")
forall (i in M1)
 writeln("  1", strfmt(i,8,0), strfmt(DT(i),16,6))
forall (i in M2)
 writeln("  2", strfmt(i,8,0), strfmt(DT2(i),16,6))
writeln
writeln ("Group  Ob. No. ", "Utility Deviation Holdout")
forall (i in L1)
 writeln("  1", strfmt(i,8,0), strfmt(DH(i),16,6))
forall (i in L2)
 writeln("  2", strfmt(i,8,0), strfmt(DH2(i),16,6))
writeln
writeln ("No. Misclas.", strfmt(NHC,15,0),  strfmt(NTC,15,0))
writeln
writeln ("No. Misclas.", strfmt(NTC1,15,0), strfmt(NHC1,15,0))
writeln
writeln ("Hit Rate (%)  ",((NTC+NTC1)/276)*100)
writeln ("Holdout Hit Rate (%)  ",((NHC+NHC1)/69)*100)
fclose(F_OUTPUT)
end-model
```

MSD Variable Selection

```
model 'MSD-model'
uses "mmxprs"
declarations
n=1..14 !attributes
m1=1..30 !Group 1
m2=1..30 !Group 2
X1:array(m1,n) of real !Data of Group1
X2:array(m2,n) of real !Data of Group2
a01: mpvar !a(0+) coefficient
a02: mpvar !a(0-) coefficient
a:array(n) of mpvar !a(j)
d1:array(m1)of mpvar !deviations in group1
d2:array(m2)of mpvar !deviations in group2
U:real
E:real
A:array(n)of mpvar !the á number
B:array(n)of mpvar !the â number
end-declarations
initializations from
'\\hssk2.hss.ed.ac.uk\mse\pghome\s0565423\Models_Run\australian_sampl
es\sample1.txt'
X1 X2 U  E
end-initializations
!Objective Function
MN:=sum(i in m1)d1(i)+sum(i in m2)d2(i)

!Group1 constraint
forall(i in m1) do
sum(j in n)X1(i,j)*a(j)-a01+a02-d1(i)<=0
end-do

!Group2 constraint
forall(i in m2) do
```

```
sum(j in n)X2(i,j)*a(j)-a01+a02+d2(i)>=0
end-do

!Normalization
a01+a02=1

!Number definitions
forall(j in n) do
-a(j)+(U+E)*A(j)<=U
end-do
forall(j in n) do
a(j)-U*A(j)<=0
end-do
forall(j in n) do
a(j)+(U+E)*B(j)<=U
end-do
forall(j in n) do
a(j)+U*B(j)>=0
end-do
forall(j in n)
A(j)+B(j)<=1

!Bounds
forall(j in n)
A(j) is_binary
forall(j in n)
B(j) is_binary
a01 is_binary
a02 is_binary
forall(i in m1)
d1(i)>=0
forall(i in m2)
d2(i)>=0

!Objective function
minimize(MN)

!Print solution
fopen('C:\Documents and
Settings\s0565423\desktop\Australian_runs\MSD\out1.txt',F_OUTPUT)
writeln("Australian Data")
writeln
writeln("Solution value is: ", getobjval)
writeln
!exportprob(0,"",MN)
forall(j in n)
 writeln(" Coefficient ", j , " : ", getsol(a(j)))
writeln
forall(i in m1)
 writeln(" Deviation ", i , " : ", sum(j in n)(X1(i,j)*getsol(a(j)))-
getsol(a01)+getsol(a02))
forall(i in m2)
 writeln(" Deviation ", i , " : ", sum(j in n)(X2(i,j)*getsol(a(j)))-
getsol(a01)+getsol(a02))
writeln
```

```
 writeln(" Coefficient a0 is : ", getsol(a01)-getsol(a02))
forall(j in n)
 writeln( "the A value is: ", getsol(A(j)))
forall(j in n)
 writeln( "the B value is: ", getsol(B(j)))
fclose(F_OUTPUT)
end-model
```

Integer Programming

```
model damcagb
 ! MCA model - Greek Bank Data
 !  Group 1 observations above function

uses "mmxprs"
setparam("XPRS_MIPTOL", 0.00000005)
declarations
 !P=34
 M1=1..370   ! No observations in group 1
 M2=1..343   ! No observations in group 2
 L1=1..93
 L2=1..86
 N=1..8      ! No variables
 X1: array(M1,N) of real     ! Group 1 observation, variable value
 X2: array(M2,N) of real     ! Group 2 observation, variable value
 Y1: array(L1,N) of real     ! Group 1 observation, variable value
 Y2: array(L2,N) of real     ! Group 2 observation, variable value
 D: real      ! Reject interval
 U: real      ! "Large" number
 E: real   !Small number
 a0: mpvar                   ! Constant in function
 a: array(1..2,N) of mpvar   ! Variable coefficient
 b1: array(M1) of mpvar      ! BV for correct classification in G1
 b2: array(M2) of mpvar      ! BV for correct classification in G2
 s1: array(L1) of mpvar      ! BV for correct classification in G1
 s2: array(L2) of mpvar      ! BV for correct classification in G2


 !d:mpvar
 !g:array(N)of mpvar !the ã number
 NHC:real
 NHC1:real
end-declarations

NHC:=0
NHC1:=0
D:= 0.0005
U:= 100
E:=0.001


initializations from 'C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Yeast\s10.txt'
 X1
end-initializations
```

```
initializations from 'C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Yeast\s10.txt'
 X2
end-initializations

initializations from 'C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Yeast\h10.txt'
 Y1
end-initializations

initializations from 'C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Yeast\h10.txt'
 Y2
end-initializations


a0 is_free

forall(i in M1) b1(i) is_binary
forall(i in M2) b2(i) is_binary
forall(i in L1) s1(i) is_binary
forall(i in L2) s2(i) is_binary
!forall(j in N) g(j) is_binary

! Constraints
!Training Samples
! Group 1
forall(i in M1)
 CA(i):= sum(j in N)X1(i,j)*a(2,j) - sum(j in N)X1(i,j)*a(1,j) -
  a0 - (U+D)*b1(i) >= -U

! Group 2
forall(i in M2)
 CB(i):= sum(j in N)X2(i,j)*a(2,j) - sum(j in N)X2(i,j)*a(1,j) -
  a0 + (U+D)*b2(i) <= U



! Normalisation
 CD:= sum(i in 1..2,j in N)a(i,j) = 1


! Objective - Maximise Classification Accuracy
MCA:= sum(i in M1)b1(i)+ sum(i in M2)b2(i)

!sum(i in M1)b1(i)<=300
!sum(i in M1)b1(i)>=270
!sum(i in M2)b2(i)=8

! Define a(1,j) and a(2,j) as SOS1
forall(j in N) ASET(j):= sum(i in 1..2) (100*i+10*j)*a(i,j) is_sos1

!Attribute Selection
!forall(j in N)
!CE(j):=a(1,j)+a(2,j)-E*g(j)>=0
```

```
!forall(j in N)
!CF(j):=a(1,j)+a(2,j)-g(j)<=0
!CG:=sum(j in N)g(j)=P


setparam("XPRS_verbose",true)

!sum(i in M1)b1(i)=90
!sum(i in M2)b2(i)<=80
!sum(i in M1)b1(i)>=sum(i in M2) b2(i)
maximize(MCA)

 !Holdout Samples
! Group 1
forall(w in L1)do
DH(w):= sum(j in N)Y1(w,j)*getsol(a(2,j)) - sum(j in
N)Y1(w,j)*getsol(a(1,j))
   if (DH(w)>=getsol(a0)) then
   NHC:=NHC+1
   end-if
   end-do

! Group 2
forall(q in L2)do
 DH2(q):= sum(j in N)Y2(q,j)*getsol(a(2,j)) - sum(j in
N)Y2(q,j)*getsol(a(1,j))
   if (DH2(q)<getsol(a0)) then
    NHC1:=NHC1+1
   end-if
   end-do


! Print solution
fopen('C:\Documents and
Settings\s0565423\desktop\Data_Comparison_1\Yeast\Results\IP\Yeast_10
.txt',F_OUTPUT)
writeln("Yeast Data")
writeln
writeln("Number of observations classified correctly : ", getobjval)
writeln
writeln("Specificity: ", getsol(sum(i in M1)b1(i)))
writeln
writeln("Sensitivity: ", getsol(sum(i in M2)b2(i)))
writeln
writeln
writeln("Discriminant Function:")
writeln(" a(2,1)-a(1,1) ",   " = ", getsol(a(2,1))-getsol(a(1,1)))
writeln(" a(2,2)-a(1,2) ",   " = ", getsol(a(2,2))-getsol(a(1,2)))
writeln(" a(2,3)-a(1,3) ",   " = ", getsol(a(2,3))-getsol(a(1,3)))
writeln(" a(2,4)-a(1,4) ",   " = ", getsol(a(2,4))-getsol(a(1,4)))
writeln(" a(2,5)-a(1,5) ",   " = ", getsol(a(2,5))-getsol(a(1,5)))
writeln(" a(2,6)-a(1,6) ",   " = ", getsol(a(2,6))-getsol(a(1,6)))
writeln(" a(2,7)-a(1,7) ",   " = ", getsol(a(2,7))-getsol(a(1,7)))
writeln(" a(2,8)-a(1,8) ",   " = ", getsol(a(2,8))-getsol(a(1,8)))
forall(w in L1)
```

```
writeln("  1",  strfmt(DH(w),15,6))
forall(q in L2)
writeln("  2",  strfmt(DH2(q),15,6))
writeln("GroupA", ":", (NHC))
writeln("GroupB", ":", (NHC1))
!writeln ("No. Misclas.", strfmt(NHC,15,0),  strfmt(NHC1,15,0))
writeln
!forall(j in N)
!writeln(" a(2,j)-a(1,j) ",   " = ", getsol(a(2,j))-getsol(a(1,j)))
fclose(F_OUTPUT)
end-model
```

APPENDIX E

Publications

1. K. Falangis (2006). "Testing the accuracy of MP models in credit environment". *ATINER: 4th International Conference on Business, Economics, Management and Marketing*.
2. K. Falangis and J. J. Glen (2007). "A MP based heuristic for Feature Selection". *10th Credit Scoring and Credit Control Conference*, Edinburgh.
3. K. Falangis (2007). "The use of MSD model in credit scoring". *Operational Research an International Journal*, 7 (3), 481-503.
4. K.Falangis and J. J. Glen (2010). "Heuristics for feature selection in mathematical programming discriminant analysis models" . *Journal of Operational Research Society*, 61, 804-812.