Maria do Carmo Rocha Sousa

A Scorecard for Pay/No Pay Decision-Making in the Retail Banking Industry

Tese submetida à
Faculdade de Ciências da Universidade do Porto
para a obtenção do grau de Mestre
em Engenharia Matemática

Dissertação realizada sob a supervisão
de
Professor Doutor Joaquim Pinto da Costa
e de
Professora Doutora Maria Teresa Mendonça
Departamento de Matemática Aplicada
Faculdade de Ciências da Universidade do Porto
Setembro de 2006

Aos meus pais, Emília e Francisco

This work became possible due to the support of some different people and organizations. In particular, I would like to thank to Manuel Gonçalves and João Sanches for their confidence in me. I am grateful with my supervisors, Professors Joaquim Costa and Teresa Mendonça for their support and guidance over the last year. Thanks to the best-in-the-world team; thanks to my family for their patience; thanks to Carlos, for being my financial guru; thanks to Jaime for his unlimited support.

Maria do Carmo da Rocha Sousa

September 2006

Sumário

A esfera financeira abrange um conjunto alargado de temas com grande mediatismo na sociedade actual, onde a decisão de risco de crédito assume grande destaque.

Na indústria da banca de retalho, o parecer dos analistas prevaleceu na decisão de crédito, sem opção, durante muitos anos. Nas últimas décadas assistiu-se à emergência dos métodos de classificação nesta área. Na década de 60, a massificação dos cartões de crédito conduziu ao desenvolvimento de modelos automáticos apropriados. Actualmente, o sector bancário acelera a implementação de novos modelos, ajustados ao tipo de decisão de crédito e segmentos de clientes e à eficiência de processos, convergindo para os requisitos do Acordo Basileia II.

A generalização da comunicação digital permitiu a intensificação dos pagamentos *online* e por débito directo nas contas de depósito à ordem. Os bancos de retalho têm de garantir um resposta imediata aos pedidos de pagamento, podendo atingir milhões de pedidos por dia. Quando uma conta não tem saldo suficiente, o banco tem de decidir se paga uma transacção a débito (processo de decisão pagar/não pagar). Este processo de decisão deve ser concluído até ao fim do dia, para cumprir os níveis de serviço estipulados para o Sistema de Compensação Interbancário. Optimizar este processo de decisão obriga que a decisão seja consistente, objectiva e rápida, com o mínimo de erros e de perdas.

Actualmente num banco de retalho português, a maior parte das decisões pagar/não pagar são tomadas automaticamente com modelos comportamentais e modelos que reproduzem regras de decisão de especialistas, sendo as decisões críticas sujeitas a avaliação humana. Contudo, os modelos comportamentais utilizados foram desenvolvidos para preverem o incumprimento a seis meses ou mais; adicionalmente, para garantir que a sua implementação é exequível, eles não reproduzem completamente o raciocínio humano. Como tal, não estão materializadas neles algumas características específicas do problema. Tanto o ciclo de receitas dos clientes como o ciclo de pagamentos demoram um mês para estarem completos. Assim, se for tomada uma decisão de 'pagar', espera-se que a conta regularize dentro de 30 dias. Este facto levou-nos a considerar o desenvolvimento de um modelo específico para classificar o risco de crédito a curto prazo para os clientes do segmento *mass-market* deste banco. Neste trabalho são construídos vários modelos de classificação com base neste conceito. Começamos por avaliar modelos de classificação de crédito binários, associando os clientes às classes de bom ou mau risco, de acordo com o seu registo de incumprimento. A detecção de uma região crítica entre as classes típicas de bom e mau risco, conjuntamente com a possibilidade de classificar manualmente alguns clientes de crédito, conduziu-nos ao desenvolvimento de uma grelha de avaliação de risco de crédito tripartido, com uma terceira classe de saída, a classe de revisão, entre a classe dos bons e a dos maus. Com este modelo, 87% das decisões podem ser tomadas automaticamente, o que compara favoravelmente com as grelhas de avaliação de risco de crédito actuais, com uma automatização de 79%.

Palavras-chave: Grelhas de avaliação de risco de crédito, conta de depósito à ordem, *pagar/não pagar*, regularização a um mês, ma*ss-market*, modelos de classificação, ROC, modelos com três classes de saída.

Abstract

The financial sphere covers a wide set of pivotal areas in the actual society, where credit decision-making assumes great relevance.

In the retail banking industry, analysts' judgment prevailed in credit decision-making, without alternative, for long time. In the last decades the emergence of classification methods took place in this area. In the sixties, the expansion of credit cards has lead to the development of appropriated models. Nowadays, the banking sector accelerates the implementation of new models, fitted to the type of credit and segments of customers and operative efficiency, converging to the Basel II Accord requirements.

The ubiquity of digital communications has led to the generalization of online payments in individuals' Demand Deposit Accounts (DDAs). Retail banks have to assure a prompt answer for those payment requests, which can be millions a day. When the DDA has not sufficient balance the bank has to decide whether to pay the debit transaction (a pay/no pay decision-making). This pay/no pay decision must be performed by the end of the day, to fit the Financial Net Settlement System service level's requirements. Optimizing this decision-making entails the decision to be consistent, objective and fast, with the minimum of mistakes and losses.

Currently at a retail Portuguese bank, most of the pay/no pay decisions are automatically managed with behavioural models and models that attempt to reproduce human judgement, while critical decisions are left for manual assessment. However, the automatic behavioural scoring models in use were developed for predicting default in a six-month period or more; furthermore, to keep the implementation straightforward, they do not entirely emulate human reasoning. Therefore, some distinctive features of the problem are not materialized on them. Both customers' income and payments cycles take one month to be completed. Hence, if a 'pay' decision is made, it is expected that the DDA cures within 30 days. This led us to consider the development of a specific model to classify short-term credit risk for mass-market customers of this retail bank. In this work several classification models are built on this assumption. We start by assessing binary scorecards, assigning credit applicants to good or bad risk classes according to their record of defaulting. The detection of a critical region between typical good and bad risk classes, together with the opportunity of manually classifying some of the credit applicants, led us to develop a tripartite scorecard, with a third output class, the review class, in-between the good and bad classes. With this model, 87% decisions can be made automatically, which compares favourably to the actual scorecards, with an automation of 79%.

Keywords: Credit scoring, DDA, pay/no pay, regularization in one month, mass-market, classification models, ROC, three-class output model.

Table of Contents

Chapter 1

Introduction

For the last decades, the emphasis of banking practices has been changing. Previously, banks had focused almost exclusively on large lending and corporate customers. Now, consumer lending is seen as an important and growing part of the bank activity. Although still representing a small fraction by value, it is becoming more and more significant.

Controlling a large bank network has arrived to a high level of complexity. Banks have begun to market their products, not only to whom they have strong relationship but also to whom they have enticed. Mistakes may occur in a lending decision that demands for fastness. In corporate lending, the aim was traditionally to avoid losses. With large consumer lending, banks became to realize that the goal should not be to loss avoidance but profit maximization. Keeping losses under control is a part of that, but profits can be maximized by taking on a small controlled level of bad debts and so expand the consumer lending books [1].

The New Basel Capital Accord[1] requirements, that will regulate banks' lending from 2007, bring out the need of reaching more flexibility and risk sensitivity by improving internal rating based (IRB) models. This in turn will allow banks to measure credit and operational risk [2].

Basel II

The first Basel Capital Accord was created in 1988 because the governors of the Group of Ten (G-10) central banks were concerned that the world's major banks were undercapitalized. (The Group of Ten is made up of 11 industrialized countries - Belgium, Canada, France, Germany, Italy, Japan, the Netherlands, Sweden, Switzerland, the United Kingdom and the United States - which consult and cooperate on economic, monetary and financial matters.) The G-10 feared a repeat of the Latin American debt crisis, which was caused by several things, including a catastrophic depletion of foreign currency reserves). The accord addressed this issue by recommending that banks have a minimum level of capital, which they carefully defined in terms of types of assets and how risky those assets were. Within two years, the accord's banking framework had been adopted by banks in more than 100 countries.

But the financial services world is constantly changing, with new products and new ways of looking at risk. With all these changes, a bank could adhere to the Basel Capital Accord and still be undercapitalized. So, the Basel Committee began a series of discussions with banks and experts in the world banking community on the need to better identify and assess risk.

In Trust but Verify: Compliance in a Regulated World by Jim Goodnight

The fundamental objective of the Committee's work to revise the 1988 Accord has been to develop a framework that would further strengthen the soundness and stability of the international banking system while maintaining sufficient consistency that capital adequacy regulation will not be a significant source of competitive inequality among internationally active banks. The Committee believes that the revised

[1] The New Basel Capital Accord is also known as Basel II.

> Framework will promote the adoption of stronger risk management practices by the banking industry, and views this as one of its major benefits. The Committee notes that, in their comments on the proposals, banks and other interested parties have welcomed the concept and rationale of the three pillars (minimum capital requirements, supervisory review, and market discipline) approach on which the revised Framework is based. More generally, they have expressed support for improving capital regulation to take into account changes in banking and risk management practices while at the same time preserving the benefits of a framework that can be applied as uniformly as possible at the national level.
>
> The Basel II Framework describes a more comprehensive measure and minimum standard for capital adequacy that national supervisory authorities are now working to implement through domestic rule-making and adoption procedures. It seeks to improve on the existing rules by aligning regulatory capital requirements more closely to the underlying risks that banks face. In addition, the Basel II Framework is intended to promote a more forward-looking approach to capital supervision, one that encourages banks to identify the risks they may face, today and in the future, and to develop or improve their ability to manage those risks. As a result, it is intended to be more flexible and better able to evolve with advances in markets and risk management practices.
>
> The efforts of the Basel Committee on Banking Supervision to revise the standards governing the capital adequacy of internationally active banks achieved a critical milestone in the publication of an agreed text in June 2004.
>
> The Committee intends the Framework set out here to be available for implementation as of yearend 2006. However, the Committee feels that one further year of impact studies or parallel calculations will be needed for the most advanced approaches, and these therefore will be available for implementation as of year-end 2007.
>
> *In Basel II: International Convergence of Capital Measurement and Capital Standards*

Meanwhile, banks run to get conditions for fitting Basel II and commercial requirements. Stimulated by the market competition, they are still involved in other important matters with very specific goals, such as empowering their internal organization and processes managing strategies. As a consequence from the actual trend of mergers & acquisitions (M&A) observed in the bank sector, their complexity is enlarged, while synergies are not completed. This leads to the necessity of performing a strict quality control across their branch network to set up with homogeneous management guidelines, in particular in the credit risk decision arena.

1.1. The Portuguese scenario

Portuguese Interbank Clearing System[2] has experienced dramatic improvements since the launch of SIBS (*Sociedade Interbancária de Serviços SA*[3]) in 1983 and particularly with the launch of ATM (Automated Teller Machine) network. In brief, in 2005 this network already had 10 000 ATM's working continuously [3], processing in average about 100 millions of operations monthly [4]. In the same year, about 43% of the Portuguese families had computer e 32% could access directly Internet from home. According to INE

[2] The Interbank Clearing System allows banks to receive and send transactional data of each bank customers and their applications, in particular customers' demand deposit accounts (DDAs).

[3] SIBS is a Portuguese company that provides services to the banking industry.

(Instituto Nacional de Estatística[4]), these data represent an annual average growth of about 17% in the number of families with computer and of 28% in Internet access [5].

This scenario has framed a big transformation in the Portuguese pattern of payment behaviour in the previous years. In particular, it has both promoted the decrease of usage of some older instruments of payment, e.g., real cash and bills of exchange, and the increase of electronic payments by credit transfers on ATM and Internet. Simultaneously, the number of payment orders of direct debits performed in customers' **demand deposit accounts** (DDAs) for all kind of service bills (e.g. phone, instalments, etc) has increased. The number of direct debits in the Net Settlement System[5] has increased from 5 to 69 millions from 2003 to 2005. Nowadays, there are about 6 millions of direct debits each month involving the amount of about 800 million Euros [4].

Despite this scenario, and regardless of the actual goal to decrease by 50% cheques transactions till 2009 [6], payments with cheques are still common practice. Each month about 13 million cheques are submitted to the Portuguese Net Settlement System, representing a total amount of 15 145 million of Euros [4].

As a consequence of these events, banks have to react efficiently to the increase of direct debits in customers' DDAs, while keep dealing with the payment of cheques, also debited in customer's DDAs. This scenario has led banks to set up powerful management systems to control the credit risk involved in those debit transactions. If all debits performed on customers DDAs were to occur only if there were sufficient funds on them, banks would be only concerned about ensuring operational viability of all processes involved. However, it does not happen always that way. If a debit transaction is being posted in customer's DDA and it has not enough balance, bank has to decide whether to pay it – pay/no pay credit decision-making.

In a time that Portuguese banks have special concerns about their consumer-lending portfolio, they face new challenges in managing the credit risk involved in this decision-making. Although the pay/no pay decision-making is not widely discussed, there are emerging signs of the awareness to the theme and of the on-going actions that are being performed to optimize this process. These include making decisions based upon behavioural models allowing banks to reduce delinquency and to make more profitable pay/no pay decisions [7].

[4] INE is the Portuguese Institute of Statistic.

[5] The Net Settlement System is the network system that allows the communication between banks in the Interbank Clearing System.

1.2. Problem formulation

The present work is embodied in the scenario described in the previous section focused in a Portuguese retail bank performing thousands of pay/no pay decisions *per* day.

The pay/no pay decision process is currently supported on models and credit analysts' assessments. Decisions are made automatically; analysts decide whether to pay those transactions that, for policy reasons, were not done by models[6].

Automatic decisions are being supported on behavioural scoring models that were developed to predict default in a six-month period and models that attempt to reproduce the decisions of credit analysts. However, current models suffer from some limitations. First, to keep the implementation straightforward, they do not entirely emulate human reasoning. Furthermore, there are some relevant features that are not materialized on them. The current method is segmented in subsystems according to customers' profitability to assure higher limits to the most profitable ones. This is the current approach of maximizing profitability. Nevertheless, this process has allowed the bank to reduce delinquencies and driven more profitable pay/no pay decisions, as well as experienced cost savings and service-level improvements [7]. The weak points that were identified lead us to expect that those models could be largely improved.

Both customers' income and payments cycles (see Figure 1) take one month to be completed. Therefore, if a 'pay' decision is made, it is expected that the DDA cures within 30 days (the DDA is cured when it does not exceed its balance and overdraft limits). This led us to consider that the process could be improved if the decisions were based in a scorecard that could predict default over a 30-day period.

[6] Transactions involving large amounts, customers with a weak relation at the bank or that are showing an ambiguous behaviour, and segments of customers that due to their commercial value or their singular transactional pattern must be considered separately.

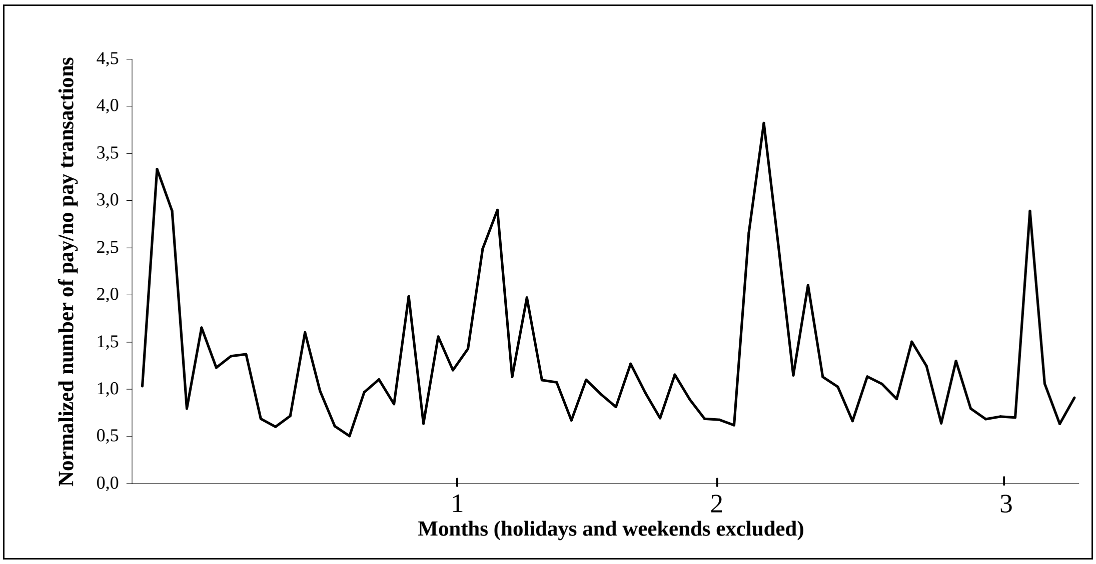

Figure 1: Normalized number of pay/no pay transactions in the period from January to March 2006.

The aim of this project is the development of a model optimized for the classification of short-term credit risk for mass-market customers of the retail network of the bank. In particular, the model will predict default in a 30-day period. The model will support pay/no pay decision-making, granting credit to those DDAs that, in spite of having insufficient balance on them, tend to regularize their balance in the following month. Since it will be focused on the prediction of default in very short time, it is expected that they will perform better than traditional scorecards specially addressed to predict default in six-month period or more.

We also intend to improve the model by maximizing profits, which we plan to materialize with the introduction of the expected loss for each incorrect decision during the development of the model, rather than just predicting the customers default risk, as in traditional scorecards.

1.3. Thesis' structure

This thesis follows in Chapter 2 with a brief history of consumer credit and credit scoring. Some classical classification methods are succinctly presented: logistic regression, classification trees and artificial neural networks. The fundamental principles of the Receiver Operating Characteristic (ROC) are presented in this chapter to be later used for assessing pay/no pay credit models performance.

In Chapter 3 the scope to develop the credit model for the pay/no pay decision is established. Some standard issues in models development are depicted, such as the data sources and the sample dataset, the level of prediction and the target classes. Although we had at our disposal a considerable volume of data, their selection was focused on the individuality of the risk of credit to be predicted. The target classes were chosen by marking the examples in the sample dataset as *defaulter* or *non-defaulter*, whether they regularized or not in a 30-day period. This was our approach to differentiate from the methodologies used for constructing models adjusted to conventional credit products, focused on predicting default in a six-month period or more.

The data treatment and preparation is one of the most important parts of the entire process and one of the most time consuming and difficult [8], often cited as taking 50 to 75% of the total project effort [9]. The main steps of this task are exposed in Chapter 4, as well as a short exposition of an empirical estimation of the misclassification losses based on historical data collected in the scope of pay/no pay decision. This was the approach for maximizing profits in the pay/no pay decision-making.

Chapter 5 introduces the experimental methodology and the results for several models achieved from the combination of classifiers with two sets of input features and two different relations between misclassification errors. Models performance was assessed using Receiver Operating Characteristic (ROC) curves and estimated losses. The chapter ends with the presentation of a three-class output model.

Finally, results and conclusions are discussed, and future work is outlined in Chapter 6.

Chapter 2

Brief history of consumer credit

Consumer credit has been around for 5000 years since the time of the Babylonians. For the majority of times, credit exhibitions were related to farmers' exchanges, which were already dealing with their cash flow problems by borrowing and planting to pay back the harvest.

By the time of Greek and Roman empires, banking and credit institutions were well advanced. The state of credit was not to known great improvement in the following millennium. Over 1350, commercial pawnshops charging interest, born by the time of Crusades, were disseminated all over Europe. Credit morality had become a critical issue during Middle Ages (in Islamic countries it still is, nowadays). The decency of charging interest on loans was questioned, mainly for large charges, for which charging was considered usury and therefore unacceptable.

In 1800, the rising of the middle classes led to the creation of private banks, mainly to offer overdrafts and to fund business and living expenses.

When consumers started to buy motorcars, in the 1920s, finance companies had rapidly expanded to fulfil this need. In the second half of the twentieth century consumer lending had a colossal expansion, triggered by the advent of credit cards in decade of 60s.

Nowadays, signs of credit are everywhere. Mainly in the Occidental World, it is hard to imagine living without a credit card and purchasing a car or a house without a loan. Banking industry offers credit lines for trips, consumer products, education, etc. Credit no longer has frontiers.

2.1. Scorecards: from Fisher's discriminant to machine learning techniques

Credit scoring is essentially a way to identify different groups in a population when we cannot see the characteristics that define the groups but only related ones [1].

The first approach to differentiate between groups took place in Fisher's original work in 1936 for general classification problems of varieties of plants. The objective was to find the best separation between two groups in a 1D dimensional space, searching for the best combination of variables such that the groups were separated the most in that subspace. Durand [10] brought this methodology to financial world in 1941 to distinguish between good and bad loans.

The World War II brought out the first expert systems. As credit analysts were called to fight, finance houses and mail-order firms requested them to write down their rules for deciding whom to give loans.

Some of these were numerical scoring systems and others were sets of conditions that needed to be satisfied – expert systems.

In the early 1950s, Bill Fair and Earl Isaac created the first consultancy directed to finance houses, retailers and mail-orders firms, making use of statistically derived models in lending decision (today, this company has a strong presence in the financial services industry [11]).

The boom of credit cards in 1960s demanded the automation of the credit decision task. This in turn required the use of better credit scoring systems, which were feasible due to the growth of computing power. The value of credit scoring became noticed and it was recognized to be much a better predictor than any other judgmental scheme.

Once perceived the value of credit scoring, it was extended to other products in 1980s. By this time, logistic regression and linear programming were introduced.

Recently, artificial intelligence techniques imported from statistical learning theory, such as classification trees and neural networks, have arisen. Classification trees are well suited for comprehension. Artificial Neural Network (ANN) technology is a well-established method of classification [12] and has received increasing attention in the financial area. Support Vector Machine (SVM) is the state-of-art neural network technology based on statistical learning [2].

The number of learning algorithms is vast. Many frameworks, adaptations to real-life problems, intertwining of base algorithms were, and continue to be, proposed in the literature, ranging from statistical approaches to state-of-the-art machine learning algorithms, parametric to non parametric procedures. Our study will not attempt to cover them all. In practice, the choice of a classifier is a difficult problem and it is often based on which classifier(s) happen to be available, or best known, to the user [13]. Our working tool will be the SAS Software, one of the world's leading information delivery system for accessing, managing, analyzing, and presenting data. Integrated in Base SAS, the SAS Enterprise Miner Software contains a collection of analytical tools that can be used to create and compare multiple models. Enterprise Miner provides three tools for **predictive modelling**: generalized linear models, classification and regression trees, and artificial neural networks.

2.1.1. Generalized linear models

Linear models represent the relationship between a continuous response variable and one or more predictor variables (either continuous or categorical) in the form $Y = XW + e$, where Y is the vector of observations of the response variable, X is the matrix determined by the predictors, W is the vector of parameters, e is a vector of random disturbances, independent of each other and usually having a normal distribution. These models are appropriated for linear relationships between the response and one or more predictors. The X matrix (and the W vector) is usually extended with an additional variable, with all data points taking the same constant value on this variable. This allows obtaining linear relationships without the restriction of passing through the origin.

When in the presence of nonlinear relationships, general nonlinear models can be used to tackle the problem. However, there are some nonlinear models, known as **generalized linear models** that can be fitted using simpler linear methods. To understand generalized linear models, first notice that the linear models have the following three characteristics [14]:

1) The response has a normal distribution with mean m
2) A coefficient vector W defines a linear combination XW of the predictors X
3) The model equates the two as $m = XW$

In generalized linear models, these characteristics are generalized as follows:

1) The response has a distribution that can be normal, binomial, Poisson, gamma, or inverse Gaussian
2) A coefficient vector W defines a linear combination XW of the predictors X
3) A link function $f()$ defines the link between the two as $f(m) = XW$

The most important and common case in which linearity is not enough is when Y and m are bounded. The linear model is inadequate in these cases because complicated and unnatural constraints on W would be required to make sure that m stays within range. Typically, the link function is used to transform the m to a scale on which it is unconstrained. The identity link specifies that the expected mean of the response variable is identical to the linear predictor, rather than to a non-linear function of the linear predictor. The canonical link functions for a variety of probability distribution are given below.

Probability Distribution	**Canonical Link Function**	**Meaning** $f()$	**Parameter restriction**
Normal	Identity	$f(m) = m$	m real
Binomial	Logit	$f(m) = \log \frac{m}{1-m}$	$m \in (0,1)$
Poisson	Log	$f(m) = \log m$	$m > 0$
Gamma	Reciprocal	$f(m) = 1/m$	$m > 0$

Other link functions are possible, such as:

- The probit (or normit) function:

$$f(m) = \Phi^{-1}(m)$$

which is the inverse of the cumulative standard normal function which is

$$\Phi(x) = (2p)^{-1/2} \int_{-\infty}^{x} \exp\left(-z^2/2\right) dz .$$

- The complementary log-log function (the term Cloglog will be used to refer to this link function)

$$f(m) = \log(-\log(1-m))$$

is the inverse of the cumulative extreme-value function (also called the Gompertz distribution), which is $F(x) = 1 - \exp(-\exp(x))$.

Logistic Regression

Logistic regression is a part of the generalized linear models. It is similar to a linear regression model but is suited to models where the dependent variable is dichotomous, that is, the dependent variable can take the value 1 with probability of success q or the value 0 with probability of failure $1-q$. This type of variable is called a Bernoulli (or binary) variable.

The independent or predictor variables in logistic regression can take any form. That is, logistic regression makes no assumption about the distribution of the independent variables. They do not have to be normally distributed, linearly related or of equal variance within each group. The relationship between the predictor and the response variables is not linear; instead, the *logit* link function is used: $\log \frac{m}{1-m} = W^T X$. Or, stated equivalently, $m = \frac{e^{W^T X}}{1+e^{W^T X}} = p(c_1 \mid X_1 W)$.

Fitting a logistic classifier model implies finding estimates of W that maximize the likelihood of the model (that is, the probability of the data given the model). It can be shown that this model is correct when both the class-conditional densities $p(X \mid C_0)$ and $p(X \mid C_1)$ are multi-normal with equal covariance matrices, where C_0 and C_1 represent the two target classes. The hyper-plane of all points X satisfying the equation $W^T X = 0$ forms the decision boundary between the two classes; these are the points for which $p(C_1 \mid X, W) = p(C_0 \mid X, W) = 0.5$.

2.1.2. Classification trees[7]

The root of the majority of the work on decision trees is in Breiman et al [17] and Quinlan's ID3 algorithm [18] from statistical and machine learning perspectives.

Decision trees are hierarchical decision systems in which conditions are sequentially tested until a class is accepted. To this end, the feature space is split into unique regions, corresponding to the classes, in a *sequential manner*. Upon the arrival of a feature vector, the searching of the region to which the feature vector will be assigned is achieved via a sequence of decisions along a path of nodes of an appropriately constructed tree. The most popular schemes among decision trees are those that split the space into hyperrectangles with sides parallel to the axes. The sequence of decisions is applied to individual features, and the questions to be answered are of the form "is feature $x_k \leq a$?" where a is a threshold value. Such trees are known as ordinary binary classification trees (OBCTs). From this process results a

[7] This section is based on [16].

model that can be presented as a tree or traduced into linguistic rules corresponding to "if-then" finite sequences, which confers the model outputs great interpretability. Other types of trees are also possible that split the space into convex polyhedral cells or into pieces of spheres. In the following one will refer only to OBCTs.

The basic idea behind an OBCT is demonstrated via the simplified example in Figure 2. By a successive sequential splitting of the space, regions are created corresponding to the various classes. Figure 2 also shows the respective binary tree with its decision nodes.

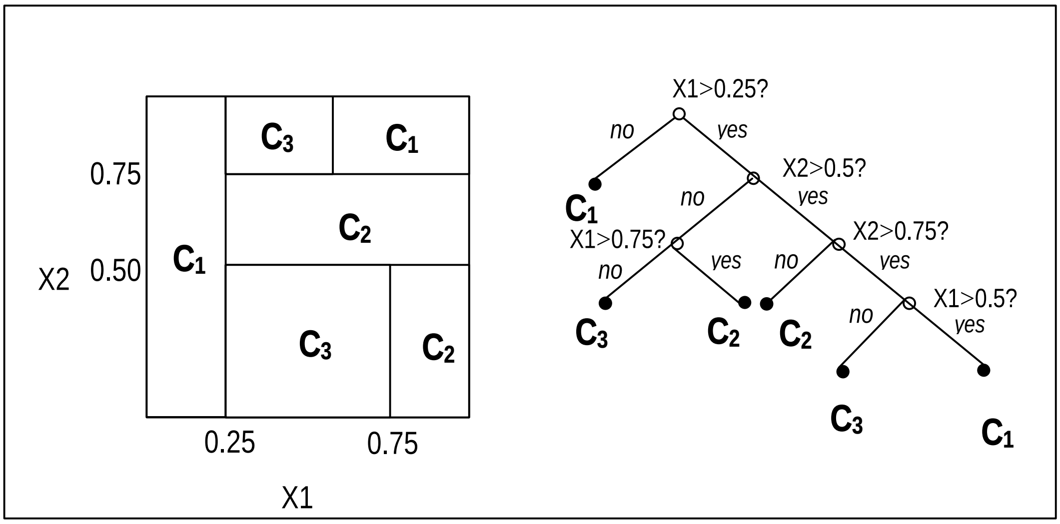

Figure 2: Simplified example of an OBCT.

In the general case, in order to develop a binary decision tree, the following design elements have to be considered in the training phase:

- At each node, the set of candidate questions to be asked has to be decided. Each question corresponds to a specific binary split into two descendant nodes. Each node t is associated with a specific subset S_t of the training set S. The splitting of a node is equivalent to the split of the subset S_t into two disjoint descendants subsets, S_{t+1} and S_{t+2}. The first of the two consists of the examples in S_t that correspond to the answer "Yes" of the question and those of the second to the "No" answer. The first node (root) of the tree is associated with the training set S.
- A *splitting criterion* must be adopted according to which the best split from the set of candidate ones is chosen.
- A stop-splitting rule is required that controls the growth of the tree and a node is declared as a terminal (*leaf*).
- A rule is required that assigns each leaf to a specific class.

Expectedly, there is more than one method to approach each of the above design elements.

Set of questions

For the OBCT type of trees the questions are of the form “Is $x_i \leq a$?”. For each feature, every possible value of the threshold a defines a specific split of the subset S_t. Thus in theory, an infinite set of questions has to be asked if alpha varies in an interval in *IR*. In practice, only a finite set of questions need to be considered. Since the number of training examples *N* if finite, any of the features x_i can take at most $N_i \leq N$ different values. Thus, for feature x_i, one can use as possible a values the midvalues of two consecutive distinct values of x_i. The same has to be repeated for all features. Thus, in such case, the total number of candidate questions is majorized by $\sum_{i=1}^{p} N_i$. However, only one of them has to be chosen to provide the binary split at the current node of the tree. This is selected to be the one that leads to the best split of the associated subset S_t. The best split is decided according to a splitting criterion.

Splitting criterion

Every binary split of a node t, generates two descendant nodes, be them $t+1$ and $t+2$, each one associated with two new subsets, S_{t+1} and S_{t+2}, respectively. There is a class of impurity measures that quantify how impure each node t of the tree is, where purity occurs when all cases in a node belong to just one class.

From the root node to the leaves, every split must generate subsets that are more homogeneous compared to the ancestor set S_t. One can define the goodness of the split s (comprising a threshold a and a feature x_i) as the decrease of impurity from the ancestor node to the descendants, reaching more “class homogeneous” descendants subsets. If one then adopt a split *s* at the node t, with the proportion of examples going into node $t+1$ being $p(t+1)$ and the proportion going into node $t+2$ being $p(t+2)$, using the impurity function I, one can measure the change of impurity originated by that split as:

$$\Delta I(s,t) = I(t) - \left[I(t+1)\,p(t+1) + I(t+2)\,p(t+2) \right].$$

The greater this difference, the greater the decrease of impurity and purer nodes are reached. Therefore, one chooses the split that maximizes this expression, which is equivalent to minimize $\left(I(t+1)\,p(t+1)+I(t+2)\,p(t+2)\right)$ while the difference remains positive; otherwise the split must not be performed.

Impurity measures commonly used in classification trees include:

- Entropy index

$$I(s,t)=-\sum_{i=1}^{K} p\left(C_i\middle|t\right)\log_2 p\left(C_i\middle|t\right)$$

- Gini Index

$$I(s,t)=\sum_{\substack{i,j=1\\ i\neq j}}^{K} p(C_i|t)\,p(C_j|t)$$

where $p\left(C_i\middle|t\right)$ denotes the probability that a vector in the subset S_t, associated with a node t, belongs to class C_i, $i=1,\ldots,K$.

Other less usual measures were not considered, such as error bounds. It is commonly accepted that the properties of the resulting final tree seem to be rather insensitive to the choice of the splitting criterion [17]. Nevertheless, this is very much a problem dependent behaviour.

<u>Stop-splitting rule</u>

The natural question that now arises is when one decides to stop splitting a node and declares it as a leaf of the tree. A possibility is to stop splitting when the purity improvement of the best split is below an adopted threshold. Other alternatives are to stop splitting either if the cardinality of the subset S_t is small enough or if S_t is pure, in the sense that all points in it belong to a single class. Experience has shown that the use of a threshold value for the impurity decrease as a stop-splitting rule does not lead to satisfactory results. Many times it stops tree growing either too early or too late. The most commonly used approach is to grow the tree up to a large size first and then prune nodes according to a pruning criterion. A number of pruning criteria have been suggested. A commonly approach is to combine an estimate of the error probability with a complexity measuring term (e.g. number of terminal nodes) [19].

<u>Class assignment rule</u>

Once a node is declared to be a leaf, then it has to be given a class label. A commonly used rule is the majority rule, i.e., the leaf is labelled as the class most represented in the leaf.

2.1.3. Artificial neural networks

Artificial neural networks, or *neural networks* for short, were originally inspired on the central nervous system and on the neurons (and their axons, dendrites and synapses), which constitute one of its most significant information processing elements. With time, they have evolved quite independently from the biological roots, giving rise to more practical implementations, based on statistics and signal processing.

It must at once be admitted that a specific architecture of neural networks will be exclusively used, namely the *multi-layer perceptron* (MLP), one type of a feed-forward network [20]. A MLP is a layered structure consisting of nodes or units (called *neurons*) and one-way connections or links between the nodes of successive layers, such as the structure of Figure 3.

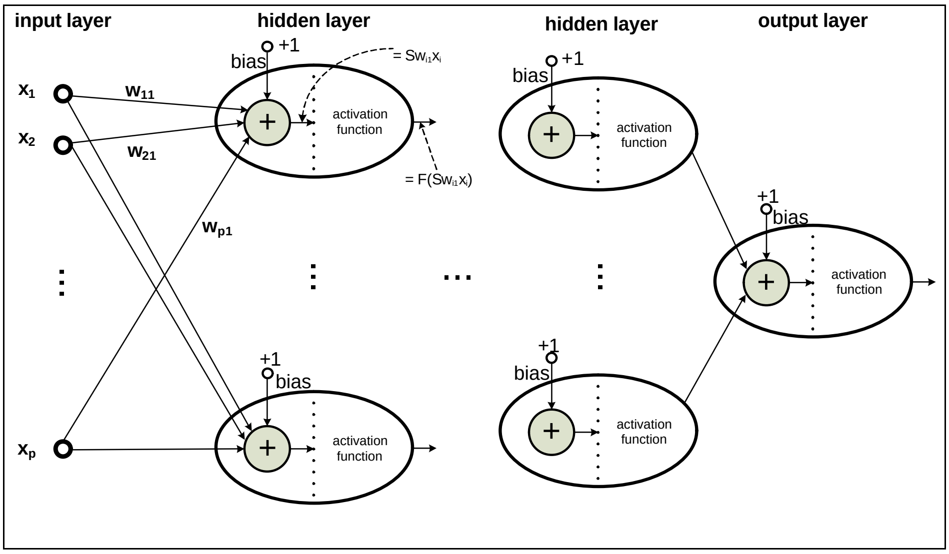

Figure 3: MLP architecture.

The first layer is called the *input layer*, the last layer is the *output layer*, while the middle layers are called the *hidden layers*. The links between nodes are one-way connections that just scale the signal through the link by the (synaptic) weight. These weights can be positive or negative, corresponding to excitation or inhibition of the flow of electrical signals in an actual neuron. The idea of a MLP is to transform an input signal as follows: the input signal is fed into the input nodes and subsequently led through the links of the network via the nodes of the hidden layers to the nodes of the output layer.

The processing inside an artificial neuron consists of two parts. The first part simply aggregates (sums) the weighted inputs; the second part is effectively a non linearity, usually called the *excitation function*, which transforms the resulting sum. Usually, this function is taken to be a sigmoid function like

$$F(x) = \frac{e^x}{e^x + 1}$$

which maps the real line onto the interval (0,1), or

$$F(x) = \tanh(x) = \frac{e^x - 1}{e^x + 1}$$

mapping the real line onto the interval (-1,1). Bias members are taken to be loaded with a signal of magnitude +1.

How do we construct a neural network classifier? Or, how do we find a set of weights such that as many examples in the training dataset as possible are correctly classified by the network? This problem is usually solved by using the so-called *back-propagation algorithm*: start with a random set of weights; then, feed the examples to the network one by one, each time comparing the output the network generates with the output that should have been generated in view of the known class of the example. When there is a difference between the two outcomes, update the weights according to a special procedure, which pushes the weights a fraction in the right direction; this is repeated for all examples in the dataset. Usually, the examples are fed into the network many times, each time constituting an *epoch* of training.

2.2. Measuring models performance

Predictive modelling tries to find good rules (models) for guessing (predicting) the values of one or more variables in a dataset (target) from the values of other variables in the data set. Our target is the quality of the individual, which can assume two values: defaulter or non-defaulter. More than discriminating between these two possibilities, we will be interested in predicting the probability of defaulting. The models to use will then yield a scored dataset as a result of their training. A scored dataset consists of a set of posterior probabilities for each level of the target variable.

The construction (training) of a model can the optimized to estimate only the probabilities of each class of the target variable, without incorporating any business objectives for which the predictor will be used. In our case the model would try to predict the probability of (not) default. Next, we would be left with the decision of selecting a score cutoff, where individuals with risk score greater than or equal to a threshold would be accepted; others, below this cutoff would be rejected. This second stage would need to incorporate the adopted measure of business performance, be it profit, loss, volume of acquisitions, market share, etc.

The ROC curves are well-suited for this second operation, enabling both to select the best cutoff for a given model and to compare multiple models, detecting dominant models, points of intersection, etc.

This two-step strategy has the benefit of being flexible in regards to changes in the measure of business performance. To accommodate a change in the loss or profit value, only the cutoff needs to be redefined, while the model is kept unchanged. Moreover, after selecting the best working point (cutoff), it is possible to perform a sensibility analysis, investigating how changes in the performance measure affect the performance of the model. It is important to stress that, often, losses or profits can not be *estimated* with great certainty by experts on the company. Therefore, this two-step strategy is easily adapted to future changes. However, caution is in order: if the adopted measure of business performance leads to heavily different losses between the different possible errors, the operating point of the model will be strongly shifted away from the point for which it was trained (equal losses), possibly leading to a substantial degradation in performance.

Another strategy would be to incorporate in the construction of the model the adopted measure of business performance. The training of the model would be focussed not in the minimization of the misclassification rate but in the optimization of the profit or loss. In this case, the second stage of optimal cutoff selection is (almost) unnecessary.[8] By integrating the business performance in the model construction we expect to attain an 'optimal' classifier, tuned for the business criterion.

2.2.1. ROC curves and optimal cutoff selection

When designing a classifier we are essentially trying to minimize two types of errors: the error committed in identifying someone as *defaulter* when one is in fact a *non-defaulter* individual and the opposite type of error of diagnosing someone as non-defaulter when one is in fact a *defaulter*. A *confusion matrix* can be used to lay out the different errors:

	Predicted class	
True Class	*Defaulter,* $\hat{D}$	*Non-defaulter,* $\hat{ND}$
Defaulter, D	$p(D,\hat{D})$	$p(D,\hat{ND})$
Non-defaulter, ND	$p(ND,\hat{D})$	$p(ND,\hat{ND})$

Confusion Matrix C1

[8] However, not all models allow the incorporation of the loss or profit matrix in the construction process; others use it in a simplified or approximated mode. Therefore, a second stage of tuning the cutoff may reveal appropriate.

In the above confusion matrix, $p(ND,\hat{D})$ represents the probability of the model predicts as a defaulter and the real class is non-defaulter; the other probabilities follow accordingly. Note that $p(D,\hat{D})+p(D,\hat{ND})=p(D)$, the a priori Default probability in the population. Likewise, $p(ND,\hat{D})+p(ND,\hat{ND})=p(ND)$, the a priori Non-default probability in the population and naturally $p(D)+p(ND)=1$.

We would like to minimize both $p(D,\hat{ND})$ and $p(ND,\hat{D})$. At one extreme case if our classifier predicts Defaulter for any individual we would have $p(D,\hat{ND})=0$, but a presumably high $p(ND,\hat{D})$ (in fact, equal to $p(ND)$); at the other end if the trained classifier predicts always non-defaulter we would have $p(ND,\hat{D})=0$ but a non zero $p(D,\hat{ND})$ (in fact, equal to $p(D)$).

So, designing a classifier resumes to finding the best trade off between these two types of errors. And the best trade off depends on the costs associated with each decision. Consider a generic loss matrix, LM1:

	Predicted class	
True Class	*Defaulter*	*Non-defaulter*
Defaulter	l_1	l_2
Non-defaulter	l_3	l_4

Loss Matrix, LM1

It is easy to see that the Expected Loss, $E[L]$, for a classifier with the confusion matrix C1 is:

$$E[L]=l_1\times p(D,\hat{D})+l_2\times p(D,\hat{ND})+l_3\times p(ND,\hat{D})+l_4\times p(ND,\hat{ND})$$

Now

$$\begin{aligned}
&l_1\times p(D,\hat{D})+l_2\times p(D,\hat{ND})=\\
&=p(D)\left(l_1\times p(D,\hat{D})/p(D)+l_2\times\left(1-p(D,\hat{D})/p(D)\right)\right)=\\
&=p(D)\left((l_1-l_2)\times p(D,\hat{D})/p(D)+l_2\right)
\end{aligned}$$

where $p(D,\hat{D})/p(D)=p(\hat{D}\mid D)$ is usually known as ***sensitivity*** or *true positive rate*.

In the same way

$$l_3 \times p(ND,\hat{D}) + l_4 \times p(ND,\hat{ND}) =$$
$$= p(ND)\left((l_3 - l_4)\times p(ND,\hat{D})\big/ p(ND) + l_4\right) =$$
$$= p(ND)(l_3 - l_4)\left(1 - p(ND,\hat{ND})\big/ p(ND)\right) + p(ND)\times l_4$$

where $p(ND,\hat{ND})\big/ p(ND) = p(\hat{ND} \mid ND)$ is usually known as *true negative rate or* ***specificity***.

Then, it results that

$$E[L] = p_D(l_1 - l_2)\times sensitivity + l_2 \times p_D + p_{ND}(l_3 - l_4)\times(1 - specificity) + l_4 \times p_{ND}$$

Summarizing, the loss of a classifier depends ***linearly*** only on two parameters, the *sensitivity* and *specificity,* weighted by coefficients derived from the loss matrix and the a priori class probability on the population. This means that we can analyse the performance of a model in a 2D space, the (1- *specificity*, *sensitivity*) space.

However, not all points in this space are possible for a model. Having trained a classifier to output the probability of defaulting, we can vary the cutoff parameter (the probability value at which we start declaring a client as defaulter) and, as we do so, trade *specificity* by *sensitivity*. The ROC of a classifier shows this tradeoff, plotting the achievable (1-*specificity*, *sensitivity*) values for a range of cutoffs. Here is an example of an ROC chart for two different classifiers.

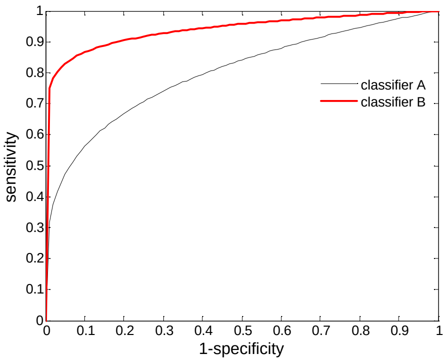

Figure 4: ROC chart for two different classifiers.

Each point on the curve represents a cutoff probability. Points closer to the upper-right corner correspond to low cutoff probabilities. Points in the lower left correspond to higher cutoff probabilities. The extreme points (1,1) and (0,0) represent no-data rules where all cases are classified into class Defaulter or class Non-Defaulter, respectively.

Now, as shown above, the expected loss can be represented as a linear function of $sensitivity$ and $1 - specificity$: $E[L] = a\times(1 - specificity) + b\times(sensitivity) + c$. To minimize the loss we just have to walk in the direction opposite to the gradient of $E[L]$. It is not difficult to see that the represented classifier B is dominant over classifier A, in the sense that, for any (reasonable) loss matrix considered, there is always an operation point of classifier B that is better than the best operating point of classifier A.

A different situation is depicted in the Figure 5. The ROC curve of classifier A intersects the ROC curve of the classifier C. Now, the best classifier depends on the loss matrix. The dashed isocost line (a isocost line is a line of constant cost, perpendicular to the gradient of the loss function) shows the least loss line for a loss matrix M2 where the costs of missing a positive case severely outweighs the cost of raising a false alarm; in this case classifier C provides the best operating point. Conversely, the dotted isocost line shows the least loss line for a loss matrix M1 where the costs of missing a negative case severely outweighs the cost of raising a positive alarm. Now is classifier A that provides the best operating point.

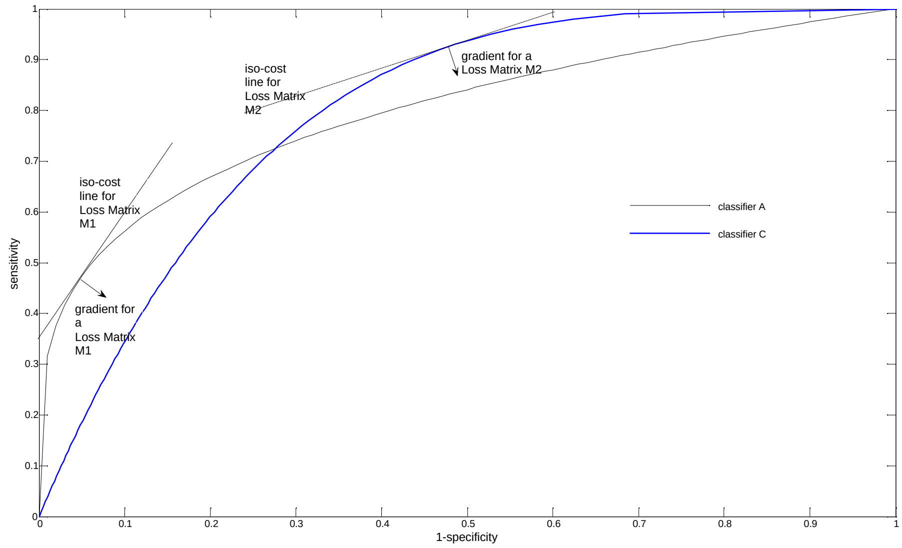

Figure 5: Non-dominant ROCs.

The ROC is most helpful when comparing two or more methods. This will be our working tool for experimentally comparing different methods.

2.2.2. Normalized loss matrix

We have seen that the expected loss can be represented as E[L] $= a \times (1 - specificity) + b \times (sensitivity) + c$, where *a*, *b*, *c* are parameters dependent on the entry values of the loss matrix (and the a priori *default* probability in the population). From the equations previously derived, we see that the slope of isocost lines is given by $-\frac{l_1 - l_2}{l_3 - l_4}\left(\frac{p_D}{p_{ND}}\right)$. Now, when comparing models or when choosing the optimal operating point of a model, the slope of the isocost lines (or, equivalently, the gradient of the expected loss function) is the ***only*** value of interest (a sufficient statistic) from the loss matrix.

Consequently, we can normalize the loss matrix $\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix}$ as far as we keep that proportion unchanged.

$$\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & l_2 - l_1 \\ l_3 & l_4 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & l_2 - l_1 \\ l_3 - l_4 & 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} 0 & \frac{l_2 - l_1}{l_3 - l_4} \\ 1 & 0 \end{bmatrix}.$$

Chapter 3

A credit model for the pay/no pay decision-making

Lenders use behaviour scoring models to adjust credit offers and decide marketing and regulation policy to be applied to each customer. These models can be developed with classification methods. In the following sections, a description of the main issues inherent to the adopted methodology to develop a behavioural scoring model for pay/no pay decision-making is given.

3.1. Available data and sources of information

The applicability of data mining techniques and classification methods depends on the acquisition of data related to the occurrence to model. Generally data come from disparate sources, internal and external, and are deposited in a data warehouse.

From internal sources, data are collected from different ongoing processes, such as pay/no pay decision-making, and applications as loans, mortgages, deposits, revolving and DDAs. Externally, the bank accesses Banco de Portugal (BdP) databases of Centralisation of Information on Cheque Defaulters (CICD) and Central Credit Register (CCR). CICD centralises the information compulsorily reported by credit institutions, and discloses through the banking system the list of cheque defaulters [21]. CCR is a database administered by BdP, using credit information reported by the participants (the institutions which grant credit). The CCR provides a range of services linked to the processing and circulation of the information. All data protection regulations are safeguarded, following the standards of the National Commission for Data Protection. The main goal of the CCR is to provide a back up for participants in their assessment of the risks attached to extending credit. Therefore, the participants can assess aggregated information on the credit liabilities of each client to the financial system as a whole. The CCR contains information on credit liabilities contracted within the financial system, both positive (when contractual obligations are being duly fulfilled) and negative (when there is credit in arrears) [22].

3.2. Level of prediction

The first decision in this project was to define the level of prediction that was intended, i.e. if the model would score the customer, the account (the DDA) or the transaction to be decided in pay/no pay decision-making.

Some credit managers support that credit risk is related to customers in a full vision, whatever are the pay/no pay transactional features. If a customer is not faulty he will pay the bank back, independently on the amount of credit being granted him in this process. Others support that, to evaluate the risk of deciding to pay this kind of transactions it is not crucial to have a holistic vision of the customer; they would better like to have a straight vision over the account. By focusing on the product, they are allowed to better distinguish the risk involved in such specific operations and therefore they can be much more proactive granting credit. Nevertheless, credit managers agree that scoring the transaction is unreliable because of the lack of information related to the customer and the product views. So far it has not been found any evidence that transactional features by themselves are sufficient to conclude whether the customer will default. Hence, the scorecard must not be developed to score transactions, which would be unrealistic and useless.

We have decided to develop a scorecard that predicts default at DDA level. It is important to have a holistic vision of the customer and it is possible that particular transactional features, combined with customer and DDA information, might better reproduce the reality we are attempting to model. With this in perspective, we have constructed a sample dataset with a single record by DDA to which was joined each observed value for every relevant and meaningful characteristic at customer, DDA and transaction views. Technical procedures are detailed in the following sections.

3.3. Target segment of the model

In the development of a credit scoring model it must be clearly defined the population to which the model will be applied. Thus, those who would not be given credit for (higher level) policy reasons should be removed from the sample, as should those who would be given them automatically. The former might include underage applicants, customers with credit in arrears, and those with no credit bureau file. The latter might include customers with specific savings products or employees of the lender [1].

The purpose is to build a model focused on customers of the mass-market segment of the retail net of the bank; these customers were *a priori* known so it was not our main intention to perform any segmentation in the population to split it into subpopulations and build different scorecards for each one[9]. Regardless of our purposes, if one uses a classification tree, then the highest-level splits may give good indications of what the appropriate segments of the population might be [1].

[9] If one intended to develop a credit model for pay/no pay decision that would be applied to other segments of customers, it would be reasonable to follow a similar methodology as for the mass-market.

A credit model can be constructed imposing restrictive conditions, mainly to move away branded risk from bank credit portfolios. Some actual restrictions consist in excluding from destination segments those customers that have internal alerts, e.g. credit in arrears and DDA over limit occurrences, or customers with fraud alerts received from CICD or CCR of BdP [21][22]. In this project, no such exclusions were performed, because a significant range of pay/no pay decisions are in DDAs of such customers, as can be noticed from Figure 6. Instead, such information was included in the input set of characteristics.

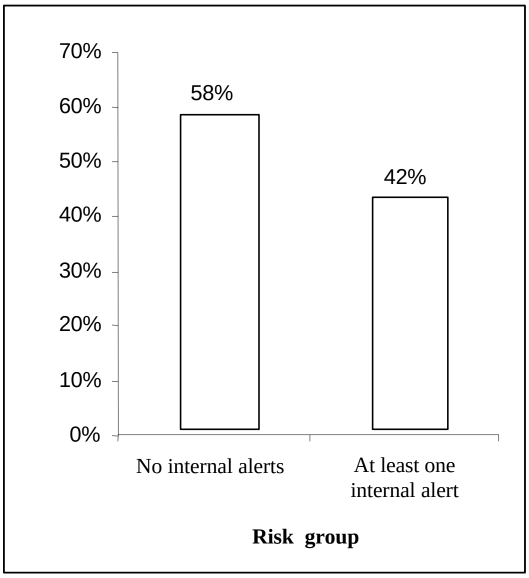

Figure 6: Pay/no pay decision-making risk structure – Percentage of customers in each risk group 'No internal alerts' and 'At least one internal alert'.

Customers with weak relation with the bank are not covered by the model and were not considered in the model design. For the current project, it was considered that a customer has insufficient relation with the bank if he has a DDA for less than three months.

Customers that the bank has special concerns to deal with, e.g. customers with stronger relationship, huge financial assets or profitability, should not be scored by the model under development. Nevertheless, for sample selection no particular exclusions were performed: on one hand because such customers are not flagged in the files for pay/no pay decision purposes, and on the other hand because, by the nature of the credit, they represent a residual fraction of pay/no pay users of mass-market segment. Some of these cases will probably fall far away from the standard ranges of values of some characteristics and will be removed in the data treatment and preparation phase, to be described in Chapter 4

3.4. Sample selection

To implement our methodology a sample of customers' DDAs with pay/no pay decisions in the past is needed. An important issue is to ensure that the sample is representative of potential users of such credit easiness in the future, the through-the-door population. It must also be sufficiently diverse in order to reflect different types of repayment behaviour and to make possible to identify which characteristics best explain differences between good and bad customers. A compromise is established between the need of having a dataset close to the future through-the-door population and the size of the observation and performance windows. These should be large enough to capture the best transactional characteristics and incorporate a reasonable history of repayment patterns, respectively. This may represent a constraint in the development of scorecards for some products of credit, such as loans and mortgages, whose development may require several years of historical data. Pay/no pay scoring model is intended to be strongly predictable in a short-term period – one month. Based on business experts' knowledge, that goal requires at most three months of historical clean data, which is easily compiled and processed. Furthermore, it simplifies the assignment of understanding external influences on dataset, such as fortuitous or seasonal occurrences. Therefore, no such constraints bounded the ongoing model development.

In this project the sample comprises DDAs with pay/no pay decisions in January 2006, the decision period. For those accounts, data was collected for the previous three months – the observation window – at all related levels: customer, DDA and transaction. The performance was evaluated for each customer's DDA according to his behaviour in the 30-day period after having had a pay/no pay decision – the performance window (see Figure 8).

The main idea was to look to pay/no pay decisions at January 2006 and evaluate whether or not their balance has turned positive in the following 30-day period. If so, the DDA was labelled with *non-defaulter*; if not, with *defaulter*. Then, the model can be developed selecting the features in the observation window – and relations between them – that better allow approaching DDA balance recovery.

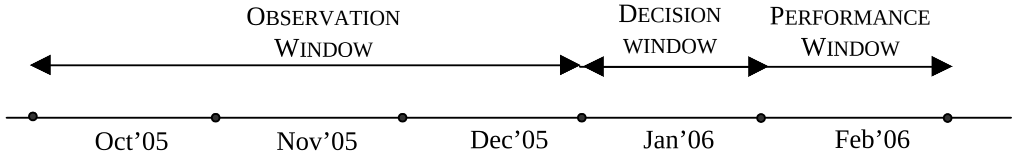

Figure 7: Time line model for application credit scoring development.

The selected sample has 187 733 records matching each DDA with a pay/no pay decision during the decision period. For each DDA, all relevant information was gathered, which includes the information considered in the ongoing pay/no pay decision process. Given that only a three months period of historical data is needed, the lack of historical information does not represent a significant restriction when developing our model, as for the other products of credit mentioned before.

3.5. Definition of the target class

In order to predict whether the customer's DDA cures within a 30-day period after a pay/no pay decision, the model was built on an analysis of good payers versus bad payers, i.e. a model with a binary[10] target. Each customer's DDA in the sample was given a performance definition, 'non-defaulter' if he was less than 30 consecutive days in excess[11] and 'defaulter' in the opposite condition.

The chosen definition of 'good' is obviously connected to the pattern of the pay/no pay decision-making, and customers' income and payments cycles (see Figure 8 to observe pay/no pay cycle). If the customer's DDA is going to cure it is expected that it happens in the following month. The cure curve along one period of three months after one day of pay/no pay approvals[12] demonstrates that 98% of the DDAs cures within the first following month; after that, its evolution is very slow, as can be noticed in Figure 8. This means that once the one-month period is exceeded, customer's DDA balance will remain negative.

[10] Binary target models are also known as dichotomous.

[11] A DDA is considered to be in excess if it exceeds its balance or overdraft limits.

[12] One means by 'one day of approvals' the entire set of DDAs that have had, at least, one transaction paid in pay/no pay decision-making.

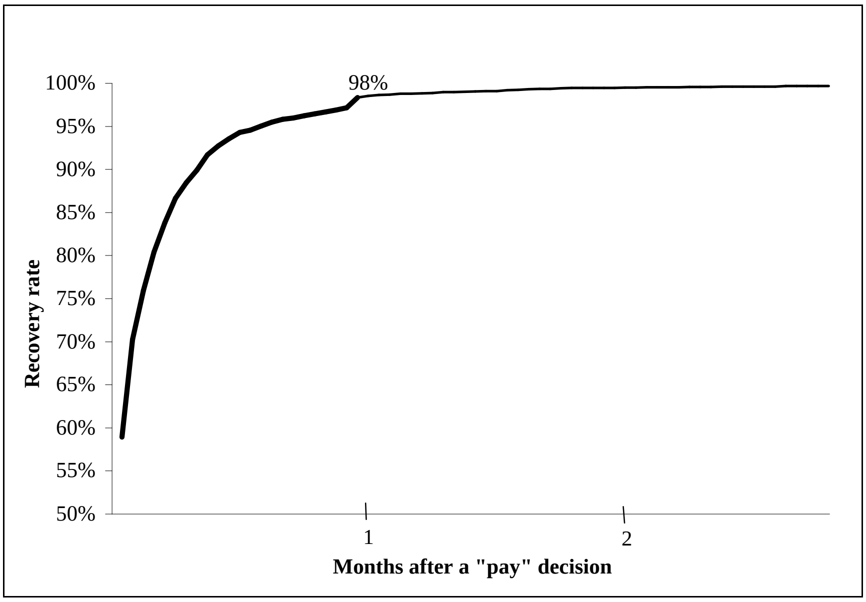

Figure 8: Standard cure curve of one day of approvals for a three month period. Each point of the cure curve corresponds to the percentage of DDAs that cured since the day of the pay decision.

3.6. Proportion of each target class

It is a delicate matter and there is no straight answer to the questions of how large the sample must be and what is the optimal proportion of each target class, *defaulter* and *non-defaulter*, in the sample. Some approaches consider an equal number of examples of each class in the sample; others reflect the *defaulter:non-defaulter* odds in the population as a whole. Normally, the second is strongly oriented to the *non-defaulters;* keeping the same odds in the sample would mean there might not be enough of bad subpopulation to identify their characteristics. If the distribution of each class in the sample is not the same as for the real proportion of *defaulters* and non-defaulters in the real population, then it must be performed an adjustment to the results obtained from the sample. In some approaches this is done automatically, as the probabilities of each class in the whole population, be them p_D and p_{ND}, are used in the calculations. For other methods, it must be done *a posteriori*. As an example, lets consider a classification tree that was build on a sample with equal target class proportions, but in the real population the ratio from *defaulters* to *non-defaulters* is 20:80, then for a node with the ratio *defaulters*:*non-defaulters* equal to 1:2, the true odds are

$$\frac{(odds\ in\ node)\cdot(odds\ in\ true\ population)}{(odds\ in\ sample\ population)} = \frac{\frac{1}{2}\cdot\frac{2}{8}}{\frac{1}{1}} = 1:8 \cdot$$

In this project, since the sample contains a significant number of registers, the real proportion of target classes in population, presented in Figure 9, will be preserved. The number of registers corresponding to defaulter examples, 33 792, is significant enough to capture their intrinsic characteristics.

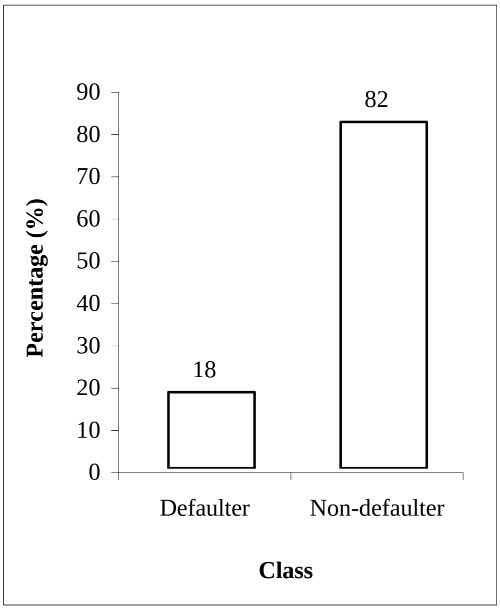

Figure 9: Proportion of each target class in the sample dataset and in the real population, i.e. pay/no pay users.

3.7. Set of features

The aim of this project is to build a model that is strongly oriented to predict default associated with short-term credit risk in customers' DDAs. There are plenty of available data in bank's database to develop models of different natures, in particular a pay/no pay decision model.

The selected characteristics for model development are those directly linked to all transactions that occur in customers' DDAs, having straight impact in its balance, and those deriving directly from the structure and volume of monthly debits and credits. We also considered information that portrays signs of customers' behaviour in their relation with other products of credit. In this perspective internal and external data were collected, mainly actual or past flawed experiences with any financial institution, such as missing payment of instalments and default warnings.

There were two major groups of information used to develop pay/no pay decision model, the information arising directly from customers and DDA data views. Additionally, we also made use of the set of information arising directly from pay/no pay decision process, i.e. we have used the registers of all transactions decided in this process in the month before the decision period and the information that had been considered by credit analysts in the manual decision. It is known that in a short-term credit, decision analysts give more attention to the most recent events; in the pay/no pay decision it usually never goes further than three months to the decision day. This is a judgemental procedure that we attempted to reproduce in our model using a three-month observation window.

The characteristics gathered from customers' DDA, as well as a brief description, are provided in Table 1.

Features code	**Description**
F01	Number of days since the last credit transaction in DDA.
F02	Number of days since the last debit transaction in DDA.
F03_1 F03_2 F03_3	Maximum balance during the month.
F04_1 F04_2 F04_3	Minimum balance during the month.
F05_1 F05_2 F05_3	Average balance of DDA during the month.
F06_1 F06_2 F06_3	Current DDA balance at the end of month.
F07_1 F07_2 F07_3	Number of debit transactions performed by customer initiative.
F08_1 F08_2 F08_3	Value of debit transactions performed by customer initiative.
F09_1 F09_2 F09_3	Number of credit transactions performed by customer initiative.
F10_1 F10_2 F10_3	Value of credit transactions performed by customer initiative.
F11_1 F11_2 F11_3	Number of days in the month that the DDA has exceeded any of its overdraft limits, independently of: 1) At the end of the month, this situation remains or has been overcome; 2) The days that the account has exceeded its limits were consecutive or not.
F12_1 F12_2 F12_3	Number of consecutive days in the month that the DDA balance has exceeded any of its limits. This period may have begun in a previous month, but always ends in the month that the information refers to, since that period is the longest with end date in the given month.
F13_1 F13_2 F13_3	Number of days since the DDA started to be over limit.
F14_1 F14_2 F14_3	Salary indicator.
F18	Average value of the pay/no pay transactions in the DDA during the month (once the pay/no pay transaction is a debit, this value was treated with negative sign).
F19	Average of the projected DDA balance after the payment of each transaction in pay/no pay in the month, even if the payment did not occur.
F20	Month average of the unavailable values, e.g. cheques or ATM deposits in the DDA that were not yet confirmed.
F22	Internal alerts flag.
F23	Number of customer's credit products.
F24	Customer's financial asset.
F25	Customer's profitability in the previous year.
F26	Counts the number of days that DDA is with negative balance, if it is with negative balance.
F27	Number of DDA parties.

Table 1: Characteristics used in the model construction. When the described features appear with three codes, they refer to each value captured in the observation window. Their termination indicates the month they refer to. The termination _1 is related to December'05, _2 refers to November'05 and _3 refers to October'05.

Chapter 4

Data treatment and preparation

Organizations gather as much information as possible about transactional data from customers' applications on their data warehouses, trying to obtain complete and truthful information systems. Commonly, sources of information are very disparate and vulnerable to human mistakes, wrong assumptions and exceptional occurrences in the processing of the information. As a result, the nature of all information is not always as reliable as required; the impact can be especially negative when used for model development.

A data treatment and preparation stage was performed before training the models. Data were explored to avoid noisel and redundant data variables, to select the most useful in models design, and the best range of values to consider in each variable. Data preparation is one of the most important parts of the entire process and one of the most time consuming and difficult [8], often cited as taking 50 to 75% of the total project effort [9]. The main steps of this process are described in the following sections.

4.1. Extraction and aggregation of data

The information of the three main data sets was extracted from the data warehouse and placed together in a database, gathering all relevant and meaningful data in a single file, see Figure 10. This file was then used as the sample dataset, the foundation of the scorecard we were attempting to develop.

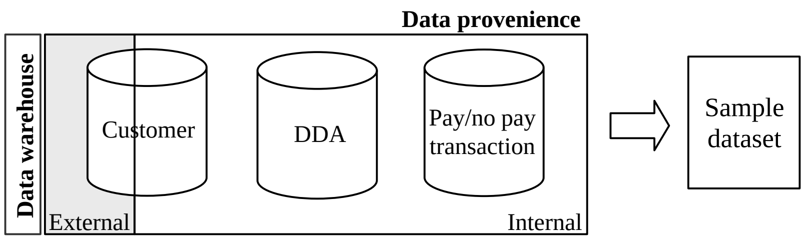

Figure 10: Main datasets used to develop pay/no pay decision model and their provenience: internal or external.

We organized the sample in such way that each DDA, corresponding features and the respective target value matches just one record in the file. To reach this we had to perform some classic techniques of data manipulation[13] using SAS Software.

4.2. Data partition

The sample dataset was randomly partitioned into three subsets for training, validation and assessment (testing). The training dataset was used for preliminary model fitting, the validation dataset was used to monitor and tune the model weights during estimation. The test dataset was just used for model assessment. The original sample dataset was randomly divided in three subsets: 70% of it to be the training set, 20% and 10% to be the validation and test data groups, respectively.

4.3. Missing values, transformation of variables and data filtering

The distribution of continuous and categorical data was analysed to determine the extension of missing data and the most appropriated action. Measures of location, scale, skewness and kurtosis for all continuous variables are presented in Table 2. Table 3 presents the extension of missing data and the number of possible values in categorical variables.

[13] This is a crucial task in this kind of projects and is usually performed by IT (Information Technology) companies' teams.

Feature	Minimum	Maximum	Mean	Std. Dev.	Missing %	Skewness	Kurtosis
F01	-3	99999	140.98	3162.9	0%	31.518	994.05
F02	0	99999	120.98	3162.6	0%	31.546	995.22
F03_1	-2754	92689	1204.9	4420	0%	10.751	157.36
F03_2	-1997	143129	1365.2	5798.6	0%	14.295	276.16
F03_3	-28903	261096	1435.2	7516.6	0%	23.623	747.16
F04_1	-1,12E+05	40206	-534.2	4621.8	0%	-14.32	275.21
F04_2	-93639	38122	-408.8	4195.1	0%	-14.28	278.44
F04_3	-30032	22419	-278.7	2059.6	0%	-5.011	85.971
F05_1	-25229	57321	322.04	2397.1	0%	12.593	296.79
F05_2	-13149	44753	369.55	1904.5	0%	10.526	186.58
F05_3	-28903	169743	487.59	4394.5	0%	30.107	1121
F06_1	-59168	76020	280.21	3105.6	0%	10.208	356.26
F06_2	-44436	49738	332.61	2439.6	0%	58.445	198.93
F06_3	-28903	35787	357.68	2324.8	0%	58.976	98.101
F07_1	0	133	15.452	18.015	0%	20.999	62.568
F07_2	0	151	14.816	17.149	0%	22.157	74.217
F07_3	0	174	15.966	17.946	0%	22.626	83.236
F08_1	0	392647	3213.7	13147	0%	17.346	432.37
F08_2	0	191100	3006.6	10747	0%	10.497	140.61
F08_3	0	281477	2924.8	11043	0%	15.384	326.78
F09_1	0	59	32.185	54.452	0%	45.701	27.58
F09_2	0	60	3.133	5.311	0%	44.275	26.172
F09_3	0	60	31.955	52.476	0%	44.678	26.253
F10_1	0	395131	3131.9	13031	0%	17.685	452.04
F10_2	0	217519	2989.2	11283	0%	11.14	159.69
F10_3	0	280393	2645.4	9395	0%	16.469	414.8
F11_1	0	31	8.241	10.622	0%	10.655	-0.311
F11_2	0	30	7.613	99.489	0%	11.343	-0.08
F11_3	0	31	7.795	10.315	0%	11.199	-0.154
F12_1	0	552	13.083	30.647	0%	68.133	78.046
F12_2	0	834	12.881	34.837	0%	10.74	192.37
F12_3	0	1687	12.524	50.303	0%	21.905	656.59
F13_1	0	552	77.365	28.637	0%	83.094	106.64
F13_2	0	552	16.595	34.266	0%	50.424	49.091
F13_3	0	1748	26.227	62	0%	13.118	319.84
F18	-1,49E+05	-0.655	-620.6	4232.7	0%	-25.28	803.29
F19	-1,00E+05	8708.4	-694.1	4093.4	0%	-15.84	312.61
F20	0	355714	564.01	8443.8	0%	37.919	1571.9
F23	0	190	0.943	47.245	0%	32.529	1285.2
F24	-28903	401433	2063.4	12320	0%	20.12	576.37
F25	-1,49E+06	164671	-1757	56011	0%	-20.8	452.7
F26	0	1749	16.372	56.703	0%	16.458	447.99

Table 2: Measures of location, scale, skewness and kurtosis of interval variables in the sample dataset.

Feature	Nº of possible Values	Missing %
F14_1	2	1%
F14_2	2	1%
F14_3	2	0%
F22	2	0%
F27	5	0%

Table 3: Missing data in categorical variables in the sample dataset.

From Table 2 and Table 3, we observe that the only variable with missing values is the characteristic salary indicator (F14). A careful analysis of these observations, complemented with a validation on a small random sample in the central system of the bank, revealed that the data was missing on those cases which DDAs had no deposits of customer's salary. Thus, missing values were replaced by zeros.

As we were attempting to build predictive models through statistical approaches such as neural networks and logistic regression, categorical variables were converted into numerical values. In the target class, *non-defaulter* and *defaulter* were replaced by 0 and 1, respectively. The variable F22 (corresponding to internal alerts) was converted to 1 if the customer had risk incidents and to 0 on the opposite situation.

Analysing the measures of location, scale, skewness and kurtosis presented in Table 2, complemented with the histograms of each variable, we observed values clearly out of standard ranges, with big deviations from medium values. Therefore, examples with values that were deviated more than three standard deviations from the mean were eliminated.

4.4. Feature selection

In supervised learning, variable selection is used to find a subset of the available inputs that accurately predict the output [23]. The objective of variable selection is three-fold: improving prediction performance of the predictor, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the data [24].

Feature evaluation is carried out according to a goodness criterion. There is a pallet of algorithms that have been already developed for feature selection, using different evaluation criteria and searching strategies. There are feature selection methods that use a measure to evaluate the goodness of individual features, such as correlation. Features are ranked according to their values on the selected measure. The first ranked features are chosen from the entire set of available characteristics, according to prior domain knowledge or a user-specified threshold value. Another group of feature selection methods evaluates the goodness of a group features. The idea is to find a good features subset, instead of good individual features.

Features selection algorithms can be divided into 'filter' and 'wrapper', according to the nature of the methods used to evaluate features. Wrapper algorithms evaluate features with the classification accuracy provided by a target classification algorithm. Filter algorithms are independent of any learning algorithms and use a particular measure that reflects the characteristics of the dataset to evaluate features. A target classifier is included in the feature selection process of the wrapper approach. This leads to the improvement of classification accuracy of the wrapper classifier, but it also contributes to the increase of the computation time. If the wrapped classification is time-consuming, or if the training dataset is large, the application of the wrapper approach may be unrealistic due to the enormous time required. Furthermore, because the derived feature subset is biased to the wrapped classification algorithm, good performance may not occur when the feature subset is used to build models with other classification algorithms [25]. These weaknesses had, somehow, limited the application of the wrapper methodology in our project, mainly due to the large size of the training dataset.

In this project, a preliminary assessment was performed using the R^2 measure as selection criterion. The method was applied to determine a ranked subset of input variables, the most related to the target. The features most useful for predicting the target variable were then identified based on R^2 output measures and linear models framework, as described bellow.

The following two-step process was performed to apply this criterion to the binary target:

1. The squared correlation coefficient (R^2) was computed for each input variable, rejecting those with a squared correlation lower than 0.005. The squared correlation coefficient, also referred as coefficient of determination, is the proportion of target variation explained by a single input variable. The coefficient ranges from 0, when there is no linear relationship between an input and the target, to 1, for an input that explains all of the target variability.

2. A forward stepwise regression was used to evaluate the remaining significant variables, rejecting those with a stepwise R^2 improvement lower than 0.0005. The sequential forward selection process starts by selecting the input variable that has the highest squared correlation coefficient. At each successive step, an additional input variable is chosen, providing the largest increase in the model R^2. The stepwise process ends when no remaining input variables can meet R^2 cutoff criterion.

This selection criterion was applied to the training dataset and the following results were generated.

The R^2 for the target variable

All model effects can be ranked by their R^2 value. The listing of the R^2 for the target variable is presented in Table 4. This information determines which effects are chosen in the Sum of Squares (SS) and R^2 portion for effects chosen for the target.

Feature	R^2	Feature	R^2	Feature	R^2	Feature	R^2
F13_1	0.351260	F09_1	0.040108	F10_1	0.015116	F18	0.004046
F11_1	0.316850	F22	0.038261	F14_1	0.013598	F04_1	0.003900
F12_1	0.282977	F05_1	0.037199	F06_3	0.012662	F02	0.002779
F26	0.168448	F07_3	0.031356	F05_3	0.012542	F27	0.002322
F13_2	0.165992	F06_1	0.029766	F14_2	0.011292	F04_2	0.002268
F12_2	0.147338	F09_2	0.026407	F03_3	0.011137	F19	0.002259
F11_2	0.143785	F03_1	0.025929	F14_3	0.009885	F04_3	0.000971
F11_3	0.089666	F05_2	0.021272	F10_2	0.009791	F25	0.000335
F12_3	0.089067	F09_3	0.019716	F08_2	0.008589	F01	0.000152
F13_3	0.084611	F03_2	0.019465	F08_3	0.008493	F23	0.000086
F07_1	0.062546	F06_2	0.019110	F10_3	0.008423	F24	0.000033
F07_2	0.041378	F08_1	0.015352	F20	0.007155		

Table 4: R^2 value for each input feature.

S and R^2 portion for effects chosen for the target

According to the settled rules, a subset of input variables was determined. The list of the chosen input variables is listed in Table 5, where it can be found the ranked selected features and the associated R^2 measure, F-value, p-value, SS and error mean square (MSE)[14] in each column.

Feature	R^2	F value	p-value	SS	MSE
F13_1	0.351260	63369	<.0001	5952.319259	0.093932
F11_1	0.056968	11267	<.0001	965.360927	0.085684
F09_1	0.015983	3248750261	<.0001	270.849416	0.083370
F26	0.004352	891324894	<.0001	73.749017	0.082741
F07_1	0.004528	934682258	<.0001	76.724336	0.082086
F13_2	0.002510	520437185	<.0001	42.531836	0.081723
F12_3	0.003127	652008014	<.0001	52.989469	0.081271
F14_1	0.001713	358241326	<.0001	29.026095	0.081271
F22	0.001665	349274705	<.0001	28.215615	0.080783
F12_1	0.000625	131223957	<.0001	10.588941	0.080694

Table 5: Input variables that were chosen according to the settled rules.

[14] MSE measures variation due to either random error or to other inputs that are not in the model. This value should get smaller as important inputs are added to the model.

Table 5 lists the sequentially selected features, which are ranked by the R^2 statistic. The relationship between these inputs and the target can be further evaluated using a modeling method. R^2 measures the sequential improvement in the model as the input variables are selected. One can observe that 35% of the variation in the target is explained by its linear relationship with F13_1. The R^2 statistic for F11_1 indicates that this interaction accounts for an additional 5,7% of the target variation.

<u>The final ANOVA table for the target</u>

The analysis of variance (ANOVA), presented on Table 6, gives the information about how well the selected inputs as a set predicted target.

Effect	R^2	SS
Model	0.442731	7502.35
Error		9443.25
Total		16946

Table 6: Final ANOVA for the target.

The inputs collectively explain 44,27% of the total variety in the target.

The attained set using R^2 criterion, presented in Table 5, was used as the input set for training three different methods: logistic regression, classification trees and artificial neural networks. The results of the models trained from this input dataset will be lately compared to models built with all the available characteristics, presented in section 3.7, for establishing the winning subset of characteristics to use as the input set in the training of each method one intended to use.

Ten variables were chosen from the 47 available features. The set attained from the R^2 criterion are: F13_1, F11_1, F09_1, F26, F07_1, F13_2, F12_3, F14_1, F22 and F12_1. It is noticed that the selected features refer to December'05, which is the closest month to the decision window. It is also observed that the top features are those reflecting the value of the DDA balance (F13_1 and F11_1) and customer relation with its DDA (F09_1) in the observation window.

4.5. Loss matrix estimation

There are two main groups of transactions in the pay/no pay process, cheques and others. The income of each of these transactions is different: cheques always generate income, whether the transaction is approved; in the other cases it only happens if the transaction is approved. Therefore, we have decided to distinguish them when evaluating their contributions to the profit/loss matrix.

As a consequence of a 'pay' decision, DDA balance keeps or turns negative, depending on its previous balance (if it was already negative or not). In practice, the bank decides to grant credit to the applicant on the debtor balance of the DDA.

If in a given month, a DDA has *n* debit transactions of any type paid in pay/no pay decision-making, the bank charges an amount M that is the maximum between a fixed fee, the and the sum of interest on the credit granted on customer's DDA to pay those transactions.

If the transaction is a cheque, an additional fee is charged, whatever the decision. For each cheque transaction that is paid in a customer DDA with no sufficient funds, bank charges a fee, f_+; on the other hand, if a cheque is refused the account is charged from a value f_-.

Our approach[15] to determine the profit/loss matrix settles in the following principles:

- Classifying as defaulter an actual defaulter customer only generates profits in refusals of cheques. The profit is then given by the charged appropriate fee, f_-, weighted by the expected proportion of cheques in pay/no pay transactions structure, p_C, i.e. $p_C f_-$.
- The error of classifying an actual defaulter as non-defaulter generates a loss that is equal[16] to the value of the transaction under decision; since the mean value of cheques is higher (Figure 11), the costs of wrong classifications are differentiated by group of transactions. Therefore, the expected cost of a bad decision in cheques, l_C, and the expected cost of a bad decision in other cases, l_O, are weighed by expected proportions of each group in pay/no pay transactions structure, which means this loss is $p_C l_C + (1 - p_C) l_O$.
- The error of classifying an actual non-defaulter as defaulter produces a loss corresponding to the uncharged fees and revenue from charging the fee f_- in the refusal of cheques transactions. Weighing those fees by the appropriate proportions in pay/no pay transactions structure the loss is given by $p_C(f_+ - f_-) + M$.
- The profit of classifying as non-defaulter an actual non-defaulter comes from the charged fees. Given that the charged fees are higher in cheques transactions, to evaluate this profit one has to

[15] Our approach for evaluating the income from a pay/no pay decision does not consider indirect income such as commercial benefits from keeping relation with good customers active. Although quantifying the income would conduct us to valuable results it would also require considering some non-trivial business assumptions. As that would take us beyond the objectives of the current work, they were not taken in consideration.

[16] In our approach, the loss of a bad classification of an actual defaulter never goes further the value of the transaction under decision (it would not behave this way if it was considered the costs of keeping relation with bad customers active). Although it is not hard to believe that such loss is probably inferior to the value of the transaction, we considered the worst scenario in which the credit is totally lost, i.e. DDA balance will never recover since the approval.

weighted the expected revenues, f_+ and *M*, by the correspondent proportion in pay/no pay transactions structure, hence the profit is given by $p_C f_+ + M$.

These principles frame a practical evaluation of the expected loss of a single decision in the pay/ no pay decision-making and can be represented in the loss matrix given in Table 7.

	Predicted class	
True class	Defaulter	Non-defaulter
Defaulter	*0*	$p_C l_C + (1 - p_C) l_O$
Non-defaulter	$p_C(f_+ - f_-) + M$	*0*

Table 7: Loss matrix used in model development.

The given matrix allows constructing a model that is adjusted to the event under formulation as it incorporates profits and losses generated in each decision.

To apply this matrix one has to determine its parameters. Although all fees and interests are pre-defined for each DDA, some scenarios can point exclusions and the amount to charge can be reduced. Hence, rather than using the standard pre-defined fees, which would lead to unrealistic and inflated profits, matrix parameters were estimated empirically using a sample of historical decisions in pay/no pay decision-making. Mean charged fees and expected costs were then calculated for each of the two groups, cheques and others.

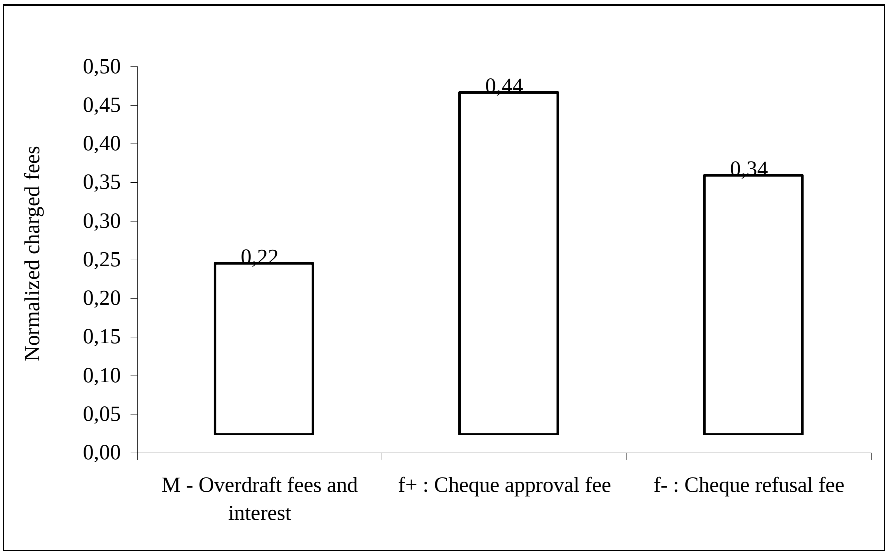

Figure 11: Normalized average charged fees by transaction in pay/no pay decision-making for mass-market segment.

Independently of the type of transaction, when the decision is 'pay', the average overdraft fees and interests charged, M, are 0.22 (normalized value). A cheque approval generates higher income, 0.44, because it is charged an additional fee f_+. The charged cheque refusal fees, f_-, are about 0.34.

Since the income of deciding a cheque is higher, our approach differentiates the contributions of each of the two groups[17] to the expected profit using their proportion in pay/no pay transactions structure (Figure 12).

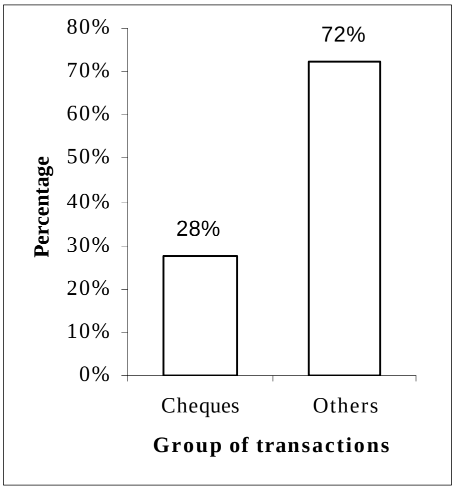

Figure 12: Percentage of each two groups of transactions in pay/no pay structure.

The value under decision in each group is also disparate; therefore one decided to calculate separately expected costs of misclassification.

[17] Another possible approach would be to build separated models for each two groups of transactions, cheques and others.

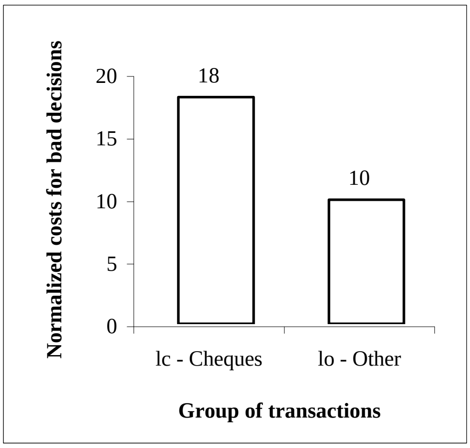

Figure 13: Normalized expected costs of misclassification in each two groups of pay/no pay transactions.

As can be deducted from the Figure 13 the costs of misclassifying a cheque are superior to the costs of misclassifying other type of transactions and the correspondent relation is 18:10.

The true values loss matrix used to develop the model is then:

	Predicted class	
True class	*Defaulter*	*Non-defaulter*
Defaulter	0.00	12.20
Non-default	0.25	0.00

The normalized loss matrix used to develop the model is then:

	Predicted class	
True class	*Defaulter*	*Non-defaulter*
Defaulter	0	49
Non-defaulter	1	0

Chapter 5

Experimental results

Next we present experimental results for several models based on logistic regression, classification trees and neural networks. All models were designed from the same input dataset. A prior vector proportional to each class in the sample was incorporated in the training of each model.

Two different subsets of features and two different loss matrices were considered in model training. The subsets of features considered were the set attained in Chapter 4 from the R^2 criterion, named set A, and the original set with all the available features, named set B. Experimental results are presented for different classification methods and subsets of features combined with the default loss matrix M_d – corresponding to equal losses for misclassification errors – and the loss matrix previously estimated in Chapter 4, which puts more weight on clients wrongly predicted as non-defaults, M_e.

Models were designed on two phases:

1. A standard training of the model. We evaluated two alternatives: training with the default loss matrix and with the estimated loss matrix. The training tries to minimize the loss under the supplied loss matrix. More than just providing the predicted class for a given example point (client), the model outputs a score for each point, where the score means the probability of default.

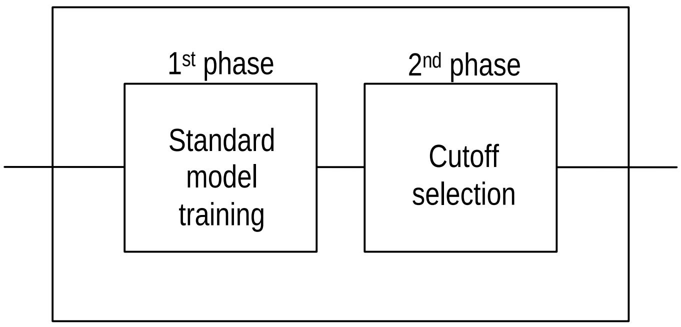

Figure 14: Model design.

2. Selection of the best cutoff for the actual business rules. For illustrative purposes, we present results with the two same matrices. So a model was first trained with a matrix (M_d or M_e) and then optimized for each of the two matrices. A few notes are in order:

a. If the true probability of defaulting p_d of a given client was known, the best cutoff for a general loss matrix $\begin{bmatrix} l_1 & l_2 \\ l_3 & l_4 \end{bmatrix}$ would be easily determined from:

   i. Expected loss of predicting as defaulter: $l_1 \times p_d + l_3 \times (1 - p_d)$

   ii. Expected loss of predicting as non-defaulter: $l_2 \times p_d + l_4 \times (1 - p_d)$

   Therefore, decide as defaulter when

$$l_1 \times p_d + l_3 \times (1 - p_d) < l_2 \times p_d + l_4 \times (1 - p_d) \Leftrightarrow p_d > \frac{1}{1 + \frac{l_2 - l_1}{l_3 - l_4}}$$

   For the default matrix, it comes that the threshold is 0.5; for the estimated loss matrix the threshold is 0.02.

b. The score output by the model after the first phase is just an approximation of the true probability. The approximation is better around the working point determined by the matrix supplied to the model, rendering almost unnecessary to optimize the cutoff when the second phase makes use of the same matrix. When carried out, this stage makes only use of the validation set to optimize the cutoff. A simple criterion was used to tune the cutoff by exhaustive search: the minimization of the loss on the validation set. Finally, the performance of the overall model was assessed on the test set. Because the two matrices incorporate such different balancing of errors, it is expected that the best model will always result from the model incorporating the loss matrix since the first phase. Results are presented for both the theoretical (T) and the experimentally tuned cutoffs (E).

Models' performance was assessed using the scored test set, searching for a dominant ROC curve, when existing, or comparing expected losses on the evaluated operating point. Expected loss, misclassification rate, probability of default, sensitivity and specificity are provided when comparing results.[18]

[18] These parameters were naturally obtained from the scored test datasets.

5.1. Results for logistic regression

Logistic regression models were built with three different link functions, logit, probit and cloglog. For both subsets of features, set A and set B, and each loss matrix, M_d or M_e, a logistic regression[19] was performed for these three link functions. Performing all possible combinations, twelve models were built. The different configurations under evaluation are presented in Table 8.

Classification method	**Loss Matrix on 1st phase**	**Link function**	**Set of features**	**Model ID**
Logistic regression	Default	Logit	B	RDLB
			A	RDLA
		Probit	B	RDPB
			A	RDPA
		Cloglog	B	RDCB
			A	RDCA
	Estimated	Logit	B	RELB
			A	RELA
		Probit	B	REPB
			A	REPA
		Cloglog	B	RECB
			A	RECA

Table 8: Configurations evaluated for the logistic regression, with the corresponding model ID.

[19] SAS procedure uses a modified Newton-Raphson algorithm to fit the model.

5.1.1. Cutoff selected for minimizing estimated losses

Table 9 presents a summary of the regression models' operating points, when the cutoff was selected on the second phase for minimizing the estimated losses (corresponding to matrix M_e).

Model ID	Cutoff	Expected loss	Specificity	Sensitivity	Misclassification rate
	E | T	E | T	E | T	E | T	E | T
RDLB	0.0236 | 0.0200	0.730 | 0.729	0.294 | 0.263	0.983 | 0.986	0.582 | 0.607
RDLA	0.0247 | 0.0200	0.783 | 0.785	0.232 | 0.189	0.983 | 0.986	0.632 | 0.667
RDPB	0.0047 | 0.0200	0.831 | 0.803	0.033 | 0.140	0.996 | 0.989	0.793 | 0.706
RDPA	0.0047 | 0.0200	0.831 | 0.803	0.033 | 0.140	0.996 | 0.989	0.793 | 0.706
RDCB	0.0594 | 0.0200	1.490 | 0.819	0.526 | 0.000	0.876 | 1.000	0.411 | 0.819
RDCA	0.0594 | 0.0200	1.490 | 0.819	0.526 | 0.000	0.876 | 1.000	0.411 | 0.819
RELB	0.0246 | 0.0200	0.729 | 0.728	0.302 | 0.258	0.982 | 0.986	0.575 | 0.610
RELA	0.0258 | 0.0200	0.785 | 0.787	0.242 | 0.189	0.981 | 0.986	0.624 | 0.667
REPB	0.0186 | 0.0200	0.724 | 0.731	0.272 | 0.283	0.986 | 0.984	0.599 | 0.590
REPA	0.0225 | 0.0200	0.771 | 0.780	0.238 | 0.217	0.983 | 0.984	0.628 | 0.644
RECB	0.0258 | 0.0200	0.753 | 0.756	0.278 | 0.231	0.982 | 0.986	0.594 | 0.633
RECA	0.0200 | 0.0200	0.798 | 0.798	0.169 | 0.169	0.987 | 0.987	0.683 | 0.683

Table 9: Measures determined at the regression models' operating points, for the case that the cutoff was selected on a second phase with matrix M_e. Results are presented for both the experimentally (E) and theoretical (T) tuned cutoffs.

<u>Evaluating sets of input features influence in models performance</u>

When analysing the results of Table 9, one observe that the influence of the subset of input features on the final performance of the models is not significant. Fixing the other training parameters under evaluation, i.e. link function and loss matrix, models' performances are roughly independent of the set of the features. Nevertheless, for those models for which a slight difference can be noticed, set B is always associated with the winning model, e.g. RECB with a minimum loss 0.753 compares favourably with RECA with minimum loss 0.798 (the pair of models REPB and REPA, RELB and RELA, RDLB and RDLA compare similarly). The pre-selection of the subset of input features with the R^2 criterion brought no improvement to the logistic regression.

<u>Evaluating link functions effects</u>

The effect of the link functions in the logistic regression is specially noticed for models trained in the first phase with the default matrix. Under such setup, it is noticed in Table 9, that cloglog link function provides the worst results, whilst logit shows the best effect. The ROCs of RDLA, RDPA and RDCA models, presented in Figure 15, strengthen this conclusion, as it is visible the weakness of cloglog for all points in the curve. This leads to the remark that, independently on the losses for misclassification, this model performance is always inferior to the others.

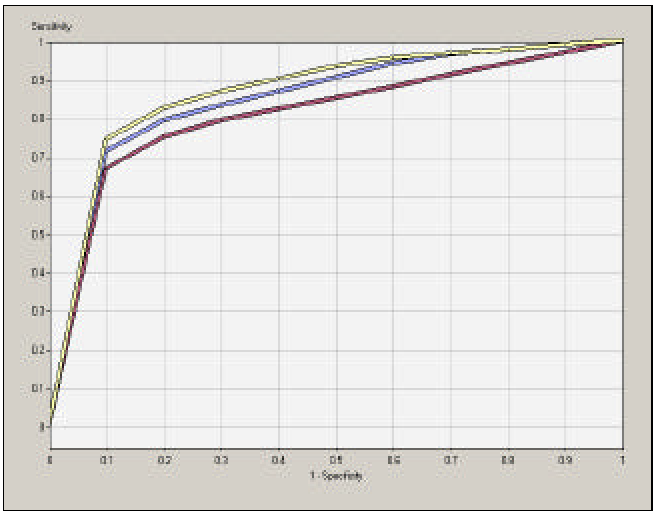

Figure 15: ROC curves of the models RDLA, RDPA and RDCA. The top curve is from RDLA, middle curve is of RDPA and down curve is RDCA's ROC.

Regression models built incorporating in their training the estimated losses perform similarly, whatever the link function. Nevertheless, logit and probit are associated with the best models, RELB and REPB, both with a minimum loss of 0.729 and 0.724 respectively.

Evaluating loss matrices effect

Fixing the link function and the set of features for evaluating the additional value of incorporating the expected losses for misclassification in the training, it is noticed that this approach leads to the best model. While for the logit link function there was no significant difference from one matrix to the other, in terms of minimum losses, for both probit and cloglog the best models were reached when the estimated losses were incorporated in the training. The noticeable improvement is attained for cloglog, for which the expected loss for models built with the estimated matrix are about a half of those trained with equal losses for misclassification.

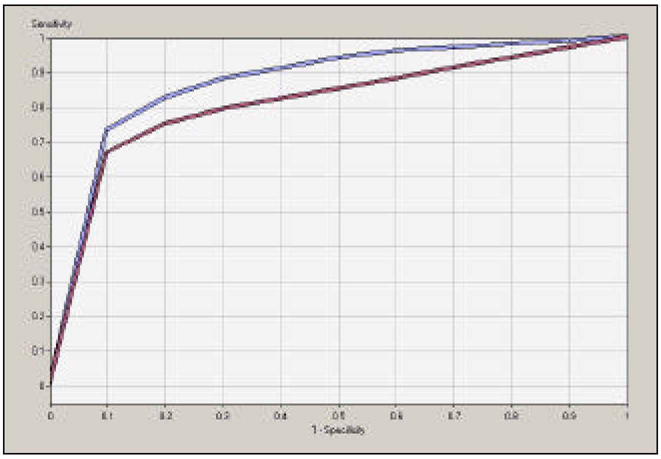

Figure 16: ROC curves of the models RDCB and RECB. The dominant curve is RECB's ROC.

Analysis of sensitivity and specificity measures

Establishing the cutoff based on the estimated losses for misclassification leads to models with high sensitivity. This measure ranges from 0.876 in models trained with default loss matrix and cloglog link function (RDCB and RDCA) to 0.996 in models trained with that matrix using probit linking function (RDPB and RDPA). The price of avoiding misclassifying *defaulter* examples is the associated low specificity value. In exception to RDCB and RDCA, with high losses, models' specificity does never goes further than 0.302, which means that for these models, up to 0.698 *non-defaulter* clients have to be rejected in the pay/no pay decision process.

5.1.2. Cutoff selected on the equal loss assumption

The model that best separates the two target classes, weighted by the actual proportion of the target classes in the whole population, was obtained by selecting the cutoff to reduce the overall loss for equal misclassification losses. Basic measures determined at logistic regression models' operating points, attained for the default loss matrix are presented in Table 10.

Model ID	Cutoff	Expected loss	Specificity	Sensitivity	Misclassification rate
	E | T	E | T	E | T	E | T	E | T
RDLB	0.595 | 0.500	0.0965 | 0.0965	0.977 | 0.970	0.569 | 0.600	0.096 | 0.096
RDLA	0.592 | 0.500	0.0964 | 0.0969	0.980 | 0.973	0.559 | 0.586	0.096 | 0.097
RDPB	0.496 | 0.500	0.0938 | 0.0940	0.984 | 0.985	0.552 | 0.550	0.094 | 0.094
RDPA	0.496 | 0.500	0.0938 | 0.0940	0.984 | 0.985	0.552 | 0.550	0.094 | 0.094
RDCB	0.502 | 0.500	0.0952 | 0.0955	0.988 | 0.988	0.526 | 0.526	0.095 | 0.095
RDCA	0.502 | 0.500	0.0952 | 0.0955	0.988 | 0.988	0.526 | 0.526	0.095 | 0.095
RELB	0.591 | 0.500	0.0966 | 0.0966	0.977 | 0.970	0.569 | 0.600	0.097 | 0.097
RELA	0.558 | 0.500	0.0970 | 0.0972	0.977 | 0.973	0.568 | 0.584	0.097 | 0.097
REPB	0.555 | 0.500	0.0973 | 0.0983	0.976 | 0.970	0.570 | 0.591	0.097 | 0.098
REPA	0.551 | 0.500	0.0971 | 0.0975	0.978 | 0.974	0.563 | 0.579	0.097 | 0.097
RECB	0.561 | 0.500	0.0994 | 0.0980	0.979 | 0.975	0.545 | 0.571	0.099 | 0.098
RECA	0.539 | 0.500	0.0989 | 0.0986	0.980 | 0.977	0.542 | 0.557	0.099 | 0.099

Table 10: Measures determined at the regression models' operating points, for the case that the cutoff was selected on a second phase with matrix M_d. Results are presented for both the experimentally (E) and theoretical (T) tuned cutoffs.

The comparison of the models built from logistic regression trained with the default matrix reveals that all fit the misclassification rate of 10%. The exceptions are RDPB and RDPA with an associated misclassification rate of 9%, comparing favourably to the others.

The measures of specificity and sensitivity are quite dissimilar. Specificity ranges from 0.976 in REPB to 0.988 in RDCB and RDCA, while sensitivity varies in the opposite way from 0.526 in RDCB and RDCA to 0.570 in REPB.

Following with this methodology it is quite obvious the influence of the prior vector, i.e. the actual proportion of target classes in the true population. Given that the dominant class *non-defaulters* represents 82% of the whole population, the cutoff tends to avoid error on *non-defaulter* examples, and therefore the specificity measure is high. The obtained models have not a fine accuracy in predicting *defaulter* examples, due to the fact they are expected to appear less than *non-defaulter.* In brief, the constructed models classify correctly a high percentage of *non-defaulter* examples, more than 97%, while they fail in the classification of *defaulters*. For all models the percentage of actual defaulters correctly classified is above 60%. The best model is RDPB.

5.2. Results for classification trees

For both subsets of features, A and B, and each loss matrix, M_d or M_e, classification trees were built using two different splitting criteria, entropy measure and gini index. Again, performing all possible combinations, eight models were built; their configurations are shown in Table 11.

Classification method	Loss Matrix on 1st phase	Splitting criteria	Set of features	Model ID
Classification trees	Default	Entropy	B	TDEB
			A	TDEA
		Gini	B	TDGB
			A	TDGA
	Estimated	Entropy	B	TEEB
			A	TEEA
		Gini	B	TEGB
			A	TEGA

Table 11: Configurations evaluated for the classification tree, with the corresponding model ID.

5.2.1. Cutoff selected for minimizing estimated losses

The performance of tree based models, when the cutoff was selected on the second phase for minimizing the estimated losses are presented in Table 12.

Model ID	Cutoff	Expected loss	Specificity	Sensitivity	Misclassification rate
	E | T	E | T	E | T	E | T	E | T
TDEB	0.0483 | 0.0200	0.896 | 0.819	0.461 | 0.000	0.949 | 1.000	0.451 | 0.819
TDEA	0.0483 | 0.0200	1.028 | 0.819	0.442 | 0.000	0.935 | 1.000	0.469 | 0.819
TDGB	0.0500 | 0.0200	1.025 | 0.819	0.440 | 0.000	0.936 | 1.000	0.470 | 0.819
TDGA	0.0500 | 0.0200	1.028 | 0.819	0.442 | 0.000	0.935 | 1.000	0.469 | 0.819
TEEB	0.0193 | 0.0200	0.697 | 0.697	0.284 | 0.284	0.988 | 0.988	0.589 | 0.589
TEEA	0.0483 | 0.0200	1.028 | 0.819	0.442 | 0.000	0.935 | 1.000	0.469 | 0.819
TEGB	0.0226 | 0.0200	0.708 | 0.708	0.327 | 0.327	0.982 | 0.982	0.554 | 0.554
TEGA	0.0230 | 0.0200	0.750 | 0.750	0.310 | 0.310	0.979 | 0.979	0.569 | 0.569

Table 12: Measures determined at the tree models' operating points, for the case that the cutoff was selected on a second phase with matrix M_e. Results are presented for both the experimentally (E) and theoretical (T) tuned cutoffs.

From analysis of Table 9 it can be noticed that the models trained with all available features (set B) have superior performance than those constructed with set A.

From a thorough analysis of the splitting effects on the final model performance it is noticed that they provide similar results.

In general, trees that were constructed including the matrix M_e in the training outperform those that were trained with matrix M_d. In exception to models TDEA and TEEA that perform equally, models TEEB, TEGB and TEGA attain losses inferior to models TDEB, TDGB and TDGA, because they fail less in defaulter examples. Since the cost for misclassifying a defaulter customer is higher, models must succeed in those examples, with the unavoidable failure in classifying *non-defaulters*. As the actual proportion of *non-defaulters* examples in the whole population is superior, misclassification rate is penalized by the inducted bias; therefore those models with lower losses have higher misclassification rates.

The model with the minimum expected loss is the model TEEB.

5.2.2. Cutoff selected on the equal loss assumption

To determine the tree that best classifies *defaulter* and *non-defaulter* examples, weighted by their actual proportion in the whole population, the default loss matrix was used to set each tree model cutoff on a second phase. Again, the goal is to achieve the model that produces lower losses which, for equal losses, is equivalent to reach the model with the lower associated misclassification rate.

Model ID	**Cutoff**	**Expected loss**	**Specificity**	**Sensitivity**	**Misclassification rate**
	E | T	E | T	E | T	E | T	E | T
TDEB	0.483 | 0.500	0.0837 | 0.0837	0.974 | 0.974	0.655 | 0.655	0.084 | 0.084
TDEA	0.474 | 0.500	0.0866 | 0.0866	0.976 | 0.976	0.630 | 0.630	0.087 | 0.087
TDGB	0.509 | 0.500	0.0826 | 0.0826	0.978 | 0.978	0.645 | 0.645	0.083 | 0.083
TDGA	0.481 | 0.500	0.0865 | 0.0865	0.978 | 0.978	0.620 | 0.620	0.086 | 0.086
TEEB	0.641 | 0.500	0.0977 | 0.0977	0.997 | 0.997	0.471 | 0.471	0.098 | 0.098
TEEA	0.474 | 0.500	0.0866 | 0.0866	0.976 | 0.976	0.630 | 0.630	0.087 | 0.087
TEGB	0.372 | 0.500	0.1701 | 0.1701	0.847 | 0.847	0.754 | 0.754	0.170 | 0.170
TEGA	0.580 | 0.500	0.0949 | 0.0949	0.985 | 0.985	0.542 | 0.542	0.095 | 0.095

Table 13: Measures determined at the tree models' operating points, for the case that the cutoff was selected on a second phase with matrix M_d. Results are presented for both the experimentally (E) and theoretical (T) tuned cutoffs.

Focusing the analysis on Table 13, it is seen that models that were trained in a first phase with matrix M_d are more accurate than models trained with the matrix M_e.

Selecting cutoff based on equal loss assumption leads to models succeeding in classifying *non-defaulter* examples. This happens because the actual proportion of *non-defaulter* examples in the whole population is superior to the proportion of defaulters, highlighting the influence of the prior vector in the resulting models. The overall misclassification rate ranges from 0.08 in TDEB and TDGB to 0.09 in TDEA and TDGA.

Models measures are very close one to the others. In fact, there is not a factual evidence of the superiority of any of them. The models with higher performance are TDEB and TDGB, and for both, the misclassification rate is equal to 0.08. They differ on their measures of specificity and sensitivity, respectively, equal to 0.97 and 0.66 for TDEB and 0.98 and 0.65 for TDGB. Differences do not seem significative.

5.3. Results for neural networks

For the neural network method one have exclusively used MLP architecture. Fixing a structure of one hidden layer, different numbers of hidden neurons were evaluated in order to find the best configuration. Five configurations were tested, varying the number of neurons from one to twenty, with a stepwise of five. The *back-propagation algorithm* was used to develop the neural network classifier. Performing all possible combinations of loss matrices, input sets of features and MLP configurations twenty models were built; their parameters are presented in Table 14.

Classification method	Loss Matrix on 1st phase	Number of neurons	Set of features	Model ID
Neural Network MLP 1 hidden layer	Default	1 neuron	B	ND01B
			A	ND01A
		5 neurons	B	ND05B
			A	ND05A
		10 neurons	B	ND10B
			A	ND10A
		15 neurons	B	ND15B
			A	ND15A
		20 neurons	B	ND20B
			A	ND20A
	Estimated	1 neuron	B	NE01B
			A	NE01A
		5 neurons	B	NE05B
			A	NE05A
		10 neurons	B	NE10B
			A	NE10A
		15 neurons	B	NE15B
			A	NE15A
		20 neurons	B	NE20B
			A	NE20A

Table 14: Configurations evaluated for the neural networks, with the corresponding model ID.

5.3.1. Cutoff selected for minimizing estimated losses

Table 15 presents a summary of the neural networks' operating points, when the cutoff was selected on the second phase for minimizing the estimated losses (corresponding to matrix M_e).

Model ID	Cutoff	Expected loss	Specificity	Sensitivity	Misclassification rate
	E | T	E | T	E | T	E | T	E | T
ND01B	0.0212 | 0.0200	0.697 | 0.698	0.343 | 0.330	0.982 | 0.983	0.541 | 0.552
ND01A	0.0317 | 0.0200	0.793 | 0.784	0.233 | 0.122	0.981 | 0.993	0.632 | 0.720
ND05B	0.0289 | 0.0200	0.733 | 0.729	0.376 | 0.295	0.975 | 0.983	0.516 | 0.580
ND05A	0.0197 | 0.0200	0.765 | 0.766	0.207 | 0.215	0.987 | 0.986	0.652 | 0.646
ND10B	0.0171 | 0.0200	0.680 | 0.691	0.301 | 0.339	0.988 | 0.983	0.575 | 0.545
ND10A	0.0303 | 0.0200	0.798 | 0.823	0.115 | 0.020	0.992 | 0.998	0.726 | 0.803
ND15B	0.0255 | 0.0200	0.693 | 0.686	0.361 | 0.297	0.981 | 0.988	0.527 | 0.578
ND15A	0.0337 | 0.0200	0.807 | 0.823	0.178 | 0.024	0.985 | 0.997	0.676 | 0.800
ND20B	0.0294 | 0.0200	0.709 | 0.727	0.316 | 0.230	0.983 | 0.989	0.564 | 0.633
ND20A	0.0213 | 0.0200	0.753 | 0.755	0.228 | 0.206	0.986 | 0.988	0.635 | 0.653
NE01B	0.0215 | 0.0200	0.715 | 0.716	0.306 | 0.285	0.983 | 0.985	0.572 | 0.589
NE01A	0.0246 | 0.0200	0.788 | 0.796	0.211 | 0.166	0.984 | 0.987	0.650 | 0.686
NE05B	0.0249 | 0.0200	0.743 | 0.764	0.246 | 0.202	0.986 | 0.988	0.620 | 0.656
NE05A	0.0192 | 0.0200	0.770 | 0.766	0.197 | 0.215	0.987 | 0.986	0.660 | 0.646
NE10B	0.0233 | 0.0200	0.673 | 0.683	0.293 | 0.256	0.989 | 0.992	0.581 | 0.611
NE10A	0.0208 | 0.0200	0.774 | 0.767	0.218 | 0.204	0.985 | 0.987	0.643 | 0.655
NE15B	0.0208 | 0.0200	0.774 | 0.767	0.218 | 0.204	0.985 | 0.987	0.643 | 0.655
NE15A	0.0208 | 0.0200	0.774 | 0.767	0.218 | 0.204	0.985 | 0.987	0.643 | 0.655
NE20B	0.0294 | 0.0200	0.709 | 0.727	0.316 | 0.230	0.983 | 0.989	0.564 | 0.633
NE20A	0.0294 | 0.0200	0.768 | 0.752	0.340 | 0.210	0.974 | 0.988	0.545 | 0.649

Table 15: Measures determined at the neural networks' operating points, for the case that the cutoff was selected on a second phase with matrix M_e. Results are presented for both the experimentally (E) and theoretical (T) tuned cutoffs.

The influence of the input set of features in models performance is only slightly perceptible, with set B being associated with the models with lower expected losses.

Although losses have slight differences from one model to the other, it can not be easily made an association between the number of neurons and models' performance. In the set of models with cutoff based on the estimated losses, NE10B outperforms the others.

5.3.2. Cutoff selected on the equal loss assumption

To determine the neural network that best classifies *defaulter* and *non-defaulter* examples, weighted by their actual proportion in the whole population, the loss matrix M_d was used to set models cutoff. In Table 16 are depicted the major indicators of models performance, achieved at models' operating points. No classifier has shown to be significantly better than the others. Although it is not a strong dominance, the simpler model, ND01B, is the one with the lower associated expected loss, outperforming the others.

Model ID	Cutoff	expected loss	Specificity	Sensitivity	Misclassification rate
	E | T	E | T	E | T	E | T	E | T
ND01B	0.499 | 0.500	0.0885 | 0.0886	0.981 | 0.981	0.596 | 0.596	0.089 | 0.089
ND01A	0.510 | 0.500	0.0926 | 0.0927	0.983 | 0.983	0.563 | 0.564	0.093 | 0.093
ND05B	0.523 | 0.500	0.0907 | 0.0909	0.980 | 0.978	0.590 | 0.597	0.091 | 0.091
ND05A	0.524 | 0.500	0.0916 | 0.0915	0.983 | 0.981	0.570 | 0.582	0.092 | 0.091
ND10B	0.516 | 0.500	0.0912 | 0.0911	0.981 | 0.980	0.581 | 0.588	0.091 | 0.091
ND10A	0.596 | 0.500	0.0939 | 0.0948	0.986 | 0.978	0.545 | 0.574	0.094 | 0.095
ND15B	0.551 | 0.500	0.0940 | 0.0941	0.982 | 0.978	0.562 | 0.581	0.094 | 0.094
ND15A	0.710 | 0.500	0.0959 | 0.1050	0.987 | 0.958	0.529 | 0.612	0.096 | 0.105
ND20B	0.509 | 0.500	0.0941 | 0.0941	0.982 | 0.981	0.562 | 0.567	0.094 | 0.094
ND20A	0.500 | 0.500	0.0906 | 0.0906	0.981 | 0.981	0.584 | 0.584	0.091 | 0.091
NE01B	0.610 | 0.500	0.0942 | 0.0937	0.981 | 0.974	0.565 | 0.602	0.094 | 0.094
NE01A	0.552 | 0.500	0.0935 | 0.0934	0.984 | 0.980	0.556 | 0.573	0.094 | 0.093
NE05B	0.462 | 0.500	0.0954 | 0.0957	0.976 | 0.979	0.583 | 0.566	0.095 | 0.096
NE05A	0.524 | 0.500	0.0916 | 0.0915	0.983 | 0.981	0.570 | 0.582	0.092 | 0.091
NE10B	0.502 | 0.500	0.0950 | 0.0952	0.978 | 0.978	0.573 | 0.574	0.095 | 0.095
NE10A	0.493 | 0.500	0.0903 | 0.0902	0.981 | 0.981	0.588 | 0.586	0.090 | 0.090
NE15B	0.482 | 0.500	0.0904 | 0.0903	0.979 | 0.981	0.593 | 0.585	0.090 | 0.090
NE15A	0.482 | 0.500	0.0904 | 0.0903	0.979 | 0.981	0.593 | 0.585	0.090 | 0.090
NE20B	0.509 | 0.500	0.0941 | 0.0941	0.982 | 0.981	0.562 | 0.567	0.094 | 0.094
NE20A	0.509 | 0.500	0.0908 | 0.0908	0.982 | 0.981	0.580 | 0.584	0.091 | 0.091

Table 16: Measures determined at neural networks' operating points, for the case that the cutoff was selected on a second phase with matrix M_d. Results are presents for both the experimentally (E) and theoretical (T) tuned cutoffs.

From the previous results it is observed that the loss matrix used during the first phase does not bias the model final performance. It can be noticed that models' performance is roughly invariable when they only differ on the loss matrix they were trained with. The fundamental step is the last, when the final cutoff is set. A natural remark is that neural network classifiers are well suited for future cutoff readjustments (following the methodology described on Chapter 2, section 2.2).

5.4. Three-class output model

The results presented in the last sections show that none of the models could discriminate pleasingly the *defaulters* from the *non-defaulters*. In fact, if we plot the distribution of the defaulters and of the non-defaulters versus the probability predicted by a model, it is observed a certain overlap between both, meaning that the model is not effective in distinguishing them. In Figure 17 the regression model REPB obtained by training with the M_e matrix, set B of features, and probit link function is used to exemplify this state of affairs. The same information is depicted in Figure 18, using the normalized cumulative histogram for the non-defaulters and one minus the normalized cumulative histogram for the defaulters. Note that for a selected cutoff point, the cumulative histogram for the non-defaulters provides the specificity and one minus the cumulative histogram for the defaulters provides the sensitivity.

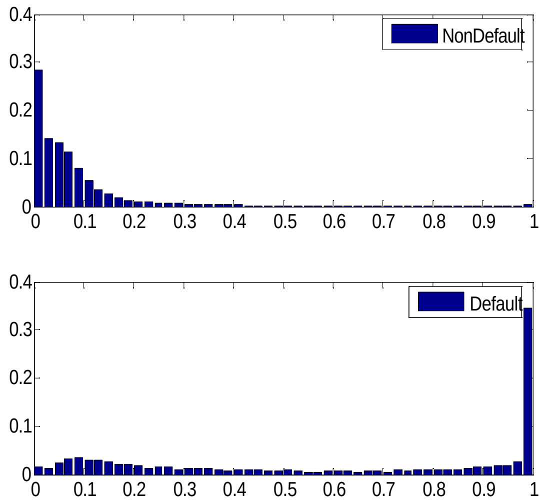

Figure 17: Histogram for REPB scored test dataset.

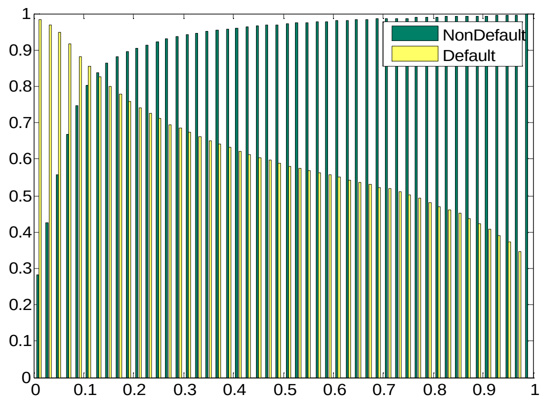

Figure 18: Normalized cumulative histogram for the non-defaulters and one minus the normalized cumulative histogram for the defaulters.

When varying the cutoff value we are just trading off between the two types of possible errors. Pushing the cutoff near the values obtained for the estimated matrix, almost all defaulters are correctly predicted, while about 70% of the non-defaulters are incorrectly predicted as defaulters. Relaxing the cutoff to values around the value obtained for the default matrix, the errors are reversed, with almost all non-defaulters correctly predicted, while about 40% of the defaulters are incorrectly predicted as non-defaulters.

When deploying a system of this kind, there is the opportunity to define a third type of decision, the ***review*** class: an example predicted as review will not be object of an automatic decision but will be evaluated manually by human experts, possibly making used of some additional information.

Therefore, we investigated the possibility of designing models with three output classes: *defaulter*, *review* and *non-defaulter*. The training of such a model comprises the definition of some objective function to be optimized. In the models previously designed the objective was to minimize the loss. Following the same path, would mean the need of defining appropriate losses for each possible decision. The lack of enough information to pursue this course, led us to define a different objective function.

Considering a generic confusion matrix[20] for a three-output model

	Predicted D	Review	Predicted ND
True D	p_1	p_2	p_3
True ND	p_4	p_5	p_6

The goal was to find a model with two cutoffs simultaneously providing

- Low error rates p_3 and p_4 (assuming that all manual decision are correct).
- Reduced number of manual decisions.

The lack of standard formulations and implementations to solve a problem of this kind, led us to start with a simple approach. Building on the models previously designed, the two cutoffs were determined experimentally on the validation set as follows:

- A cutoff was initialized as 0.0. Next, it was iteratively raised until a predefined probability p_3 (= 0.025) was obtained.
- A cutoff was initialized as 1.0. Next, it was iteratively lowered until a predefined probability of error p_4 (= 0.050) was obtained.

[20] In this matrix D is used for representing *default* and ND *non-default.*

Finally, the percentage of automatic correct decisions ($p_1 + p_6$), the percentage of defaulters in the approved set (p_3 / p_6), the misclassification rate ($p_3 + p_4$) and the percentage of automatic decisions ($p_1 + p_3 + p_4 + p_6$) were measured on the test set.

Seven models were chosen from all under evaluation. The following criterions were considered:

- The model must grant automation above 80%;
- The percentage of actual defaulters approved by the model must be up to 5.0%.

Table 17 presents these models and the associated measures previously introduced in this text.

Model	Cutoff low	Cutoff high	Automatic correct decisions $p_1 + p_6$	Approved defaulters p_3 / p_6	Misclassification rate $p_3 + p_4$	Automation $p_1 + p_3 + p_4 + p_6$
RDLB	0.1012	0.2995	75.0%	3.7%	7.2%	82.2%
RELB	0.1016	0.2934	75.2%	3.7%	7.4%	82.6%
REPB	0.1049	0.3049	74.9%	3.5%	7.2%	82.1%
TDGA	0.0696	0.2373	78.6%	4.9%	8.0%	86.6%
TEEA	0.0677	0.2324	78.6%	4.9%	8.0%	86.6%
ND01B	0.1098	0.2732	77.6%	3.7%	7.4%	85.0%
NE20A	0.1209	0.2815	79.1%	4.7%	8.2%	87.3%

Table 17: Summary of the chosen models' associated measures.

The models obtained from the logistic regression are suitable for granting a low percentage of approved defaulters (up to 3.7%); however they are not as powerful when their associated automation is considered. The model ND01B is clearly a better option than logistic regression models once it grants higher automation (85.0%), for a similar misclassification rate (7.4%) and a percentage of approved defaulters (3.7%). The higher automation is reached for models developed with trees and neural networks classifiers (TDGA, TEEA, and NE20A), for which it is about 87.0%. With these models at least 4.7% of the actual defaulters will be approved that would rather be preferred to be far less.

Assuming that the percentage of actual defaulters approved automatically does not consider the effects of the recovery actions that can be performed, we accepted a value up to 5%. The tree-based model was considered the most adequate for the pay/no pay decision-making, providing 87% of automatic decisions. Furthermore, a decision tree scheme is suitable for comprehension and implementation.

Chapter 6

Conclusion and discussion

This study focuses on the application of classification methods to develop a suitable model for pay/no pay decision-making for the mass-market customers of a retail Portuguese bank.

Several binary classifiers based on logistic regression, classification trees and neural networks were developed for classifying short-term credit risk. In particular, the models were developed for predicting whether a customer will default in a 30-day period after having a transaction paid in this decision-making. The rationale behind this approach was that models specifically targeted for predicting default in a 30-day period would outperform traditional scorecards currently in use, which incorporate in their design a six-month period or more for defaulting.

An exhaustive study was conducted, assessing different configurations for the classification models. A prior vector proportional to each class in the sample was incorporated in the training of each model. Two different subsets of features were considered in model training: a subset of features attained from the R^2 criterion and the original set with all the available features. The subsets of features were combined with a default loss matrix corresponding to equal losses for misclassification errors and a loss matrix that puts more weight on clients wrongly predicted as *non-defaults*. The costs of misclassification were empirically estimated to portray the business performance of the different decisions.

The effect of the link function (logit, probit and cloglog) on the logistic regression training was evaluated, as well as the splitting criteria (entropy and gini) effect on the final tree models. In the first case, it was not found evidence of a strong superiority of any link function, in spite of the fact that the cloglog link function has shown to be the weaker when the logistic regression was trained in a first phase with the default loss matrix. Similarly, the performance of the tree models was quite insensitive to the splitting criterion, since both the entropy measure and the gini index have attained similar results.

Neural network methods were applied based on the MLP architecture. Fixing a structure with one hidden layer, different numbers of hidden neurons were tested in order for establishing the appropriate configuration. Five configurations were tested, varying the number of neurons from one to twenty, stepwise five. No classifier has shown to be significantly superior to the others and no relation was found between the number of neurons and models performance. From the models developed from neural networks it was observed that the loss matrix used during the first phase does not bias the final model performance. Models performance is roughly invariant when they only differ on the loss matrix they were trained with. The fundamental phase is the second, when the final cutoff is set. A natural remark is that neural network classifiers are well suited for future cutoff readjustments.

All methods have shown a similar performance. This fact is probably a consequence of having a large training dataset. Moreover, linear classifiers performed as well as nonlinear ones. Although the data is clearly non-linearly separable, the nonlinear classification methods did not find any intrinsic property of the problem.

Although an extensive study was conduced, the attained discrimination between classes was not satisfactory. In fact, the results attained with the binary classifiers show that none could discriminate pleasingly the *defaulters* from the *non-defaulters*. It was observed a certain overlap between the distribution of the defaulters and of the non-defaulters, when analysed over the predicted probability of defaulting, meaning that the models were not effective in distinguishing them. Different loss matrices just make different tradeoffs between the two types of possible errors.

When deploying a system of this kind, there is the opportunity of defining a third type of decision, the ***review*** class: an example predicted as review will not be object of an automatic decision but will be evaluated manually by human experts, possibly making used of some additional information. Therefore, we investigated the possibility of designing models with three output classes: *default*, *review* and *non-default*. The train of the tripartite models was driven to find two cutoffs simultaneously providing low error rates and high automation rate. The lack of standard formulations and implementations to solve a problem of this kind, led us to start from the binary models previously designed and optimize the two cutoffs. The final model allows 87% automatic decisions, comparing favourably to the actual scorecards, with an automation of 79%. More principled approaches for optimally determining the boundaries between the good, review and bad classes are currently being investigated. A complementary model is also under development for managing the resulting credit in arrears.

A tree-based model was selected from the designed models, mainly because of its simple architecture and high automation. Its weakness, the percentage of actual defaulters approved by the model, 4.9%, was considered acceptable, because it was assumed that this percentage does not consider the effects of the recovery actions executed by the bank. Without such actions, 4.9% of the approvals would never be recovered; performing them, the percentage of non-recoveries will be far less. Noticeably, this model seems to be well adjusted for the current needs in pay/no pay decision-making, thus it will be purposed as a **challenger** to the models currently in use at the retail bank.

At the top of our objectives for future work is the development of new models focused on other segments of customers of the retail network of the retail bank. Methods that provide simple models, effortless for comprehension and implementation at the bank, will be favoured. The methodology adopted in this work for establishing adjusted models will be followed, in view of the high automation rate and good profit maximization possible with this approach. It was noticed from this work that the results could have been improved if new features and combinations were brought to the input set of characteristics. For this reason the selection of features will be at the spotlight.

References

[1] L.C. Thomas and D. B. Edelman and J. N. Crook, *Credit Scoring and Its Applications*. SIAM, 2002.

[2] S. Li, W. Shiue and M. Huang, "*The evaluation of consumer loans using support vector machines*". *Expert systems with Application, vol.30, pp. 772-782, 2006.*

[3] João Perdigão, Computerworld: Opinião. http://www.computerworld.com.pt/public/text.asp?text_id=109, 2005.

[4] Banco de Portugal, Estatísticas Monetárias e Financeiras. Boletim Estatístico - Agosto de 2006, B.12 Sistemas de Pagamento, August 2006.

[5] Agência Financeira, "Mais de 40% das famílias portuguesas têm computador". http://www.agenciafinanceira.iol.pt/noticia.php?id=618713, May 2005.

[6] M. Gago, "SIBS lança plano para cortar uso de cheques para metade". Diário de Notícias, pp.10, 20-07-2006.

[7] Fair Isaac, "Leading Portuguese bank fuels growth with customer-focused risk strategies". Viewpoints, vol. 30, pp. 15, Summer 2006.

[8] L. Soibelman and H. Kim, "Data Preparation Process for Construction Knowledge Generation through Knowledge Discovery in Databases". *Journal of Computing in Civil Engineering,* Vol. 16, No. 1, pp. 39-48, January 2002

[9] W. Thompson and D. Duling, "What's new is SAS Enterprise Miner 5.2". Data Mining and Predictive Ring (SUGI 31), Paper 082-31, 2005.

[10] Durand, D., *Risk elements in consumer installment financing*. National Bureau of Economic Research, New York, 1941.

[11] Gareth Herschel, "Magic Quadrant for Customer Data Mining, 1Q06". Gartner RAS Core Research Note G00132466, January 2006.

[12] L.C. Thomas and R. W. Oliver and D. J. Hand, "A survey of the issues in consumer credit model search, Journal of Operational Research Society", vol. 56, pp.1006-1015, 2005.

[13] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review". *IEEE Transactions on Pattern,* vol. 22, no. 1, pp. 4-37, 2000.

[14] The MathWorks, MATLAB R14 Statistical Toolbox Help: Generalized Linear.

[15] Statistical Science website: Generalized Linear Models. http://www.statsci.org/glm/intro.html, accessed 27 September 2006.

[16] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Elsevier 2003.

[17] L Breiman and J Friedman and R. Olshen and C Stone, *Classification and Regression Trees*. Wadsworth Int. Group, 1984.

[18] J. R. Quinlan, *Induction of decision trees*. Machine Learning, 1986.

[19] B. Ripley, *Pattern Recognition and Neural Networks*. Cambridge Press, 1996.

[20] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[21] Banco de Portugal, *Cheques Restrição ao seu uso*. Cadernos do Banco de Portugal, 2001.

[22] Banco de Portugal, *Central de Responsabilidades de Crédito*. Cadernos do Banco de Portugal, 2001.

[23] Caruana, Rich and de Sa, Virginia R., "Benefiting from the Variables that Variable Selection Discards". Journal of Machine Learning Research (JMLR), vol. 3, pp. 1245-1264, March 2003.

[24] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection. Journal of Machine Learning Research", vol. 3, pp. 1157-1182, March 2003.

[25] Y. Liu and M. Schumann, "Data Mining feature selection for credit scoring models". Journal of the Operational Research Research Society, vol. 56, pp 1099-1108, 2005.