Universidade do Minho
Escola de Engenharia

Departamento de Informática

Fábio André Souto da Silva

Credit Scoring as an Asset for Decision Making in Intelligent Decision Support Systems

Outubro, 2011

Universidade do Minho
Escola de Engenharia

Departamento de Informática

Fábio André Souto da Silva

Credit Scoring as an Asset for Decision Making in Intelligent Decision Support Systems

Dissertação
Mestrado em Engenharia Informática

Supervisor:
Cesar Analide

Outubro, 2011

Acknowledgements

First and foremost I offer my sincerest gratitude to my supervisor, Professor Cesar Analide, who has always supported me throughout my dissertation inspiring me to work harder.

I would like also express my gratitude to professor Paulo Novais for the ideas and the words of encouragement given.

A special thanks is required for both my parents and my sister for the support, encouragement and patience towards myself. Their help always allowed me to achieve my objectives.

A last mention to my girlfriend Rita for the patience and understanding demonstrated over the last year and for the conscience of the difficulty attaining a master degree has.

Resumo

Hoje em dia, a análise de risco é um tópico importante para as instituições financeiras, especialmente no contexto de pedidos de empréstimo e de classificação de crédito. Algumas destas instituições têm já implementados sistemas de classificação de crédito personalizados para avaliar o risco dos seus clientes baseando a decisão do pedido de empréstimo neste indicador. De facto, a informação recolhida pelas instituições financeiras constitui uma valiosa fonte de dados para a criação de ativos de informação, de onde mecanismos de classificação de crédito podem ser desenvolvidos.

Historicamente, a maioria das instituições financeiras baseia os seus mecanismos de decisão sobre algoritmos de regressão. No entanto, estes algoritmos já não são considerados o estado da arte em algoritmos de decisão. Este facto levou ao interesse na pesquisa de diferentes algoritmos de aprendizagem baseados em algoritmos de aprendizagem máquina, capaz de lidar com o problema de classificação de crédito.

O trabalho apresentado nesta dissertação tem como objetivo avaliar o estado da arte em algoritmos de decisão de crédito, propondo novos conceitos de optimização que melhorem o seu desempenho. Paralelamente, um sistema de sugestão é proposto no âmbito do tema de decisão de crédito, de forma a possibilitar a perceção de como os algoritmos tomam decisões relativas a pedidos de crédito por parte de clientes, dotando-os de uma fonte de pesquisa sobre como melhorar as possibilidades de concessão de crédito e, ainda, elaborar perfis de clientes que se adequam a determinadas condições e propósitos de crédito.

Por último, todos os componentes estudados e desenvolvidos são combinados numa plataforma capaz de lidar com o problema da classificação de crédito através de um sistema de especialistas, implementado como um sistema multi-agente. O uso de sistemas multi-agente para resolver problemas complexos no mundo de hoje não é uma nova abordagem. No entanto, tem havido um interesse crescente no uso das suas propriedades, em conjunto com técnicas de aprendizagem máquina e *data mining* para construir sistemas mais eficazes. O trabalho desenvolvido e aqui apresentado pretende demonstrar a viabilidade e utilidade do uso deste tipo de sistemas no problema de decisão de crédito.

Abstract

Risk assessment is an important topic for financial institution nowadays, especially in the context of loan applications or loan requests and credit scoring. Some of these institutions have already implemented their own custom credit scoring systems to evaluate their clients' risk supporting the loan application decision with this indicator. In fact, the information gathered by financial institutions constitutes a valuable source of data for the creation of information assets from which credit scoring mechanisms may be developed.

Historically, most financial institutions support their decision mechanisms on regression algorithms, however, these algorithms are no longer considered the state of the art on decision algorithms. This fact has led to the interest on the research of new types of learning algorithms from machine learning able to deal with the credit scoring problem.

The work presented in this dissertation has as an objective the evaluation of state of the art algorithms for credit decision proposing new optimization to improve their performance. In parallel, a suggestion system on credit scoring is also proposed in order to allow the perception of how algorithm produce decisions on clients' loan applications, provide clients with a source of research on how to improve their chances of being granted with a loan and also develop client profiles that suit specific credit conditions and credit purposes.

At last, all the components studied and developed are combined on a platform able to deal with the problem of credit scoring through an experts system implemented upon a multi-agent system. The use of multi-agent systems to solve complex problems in today's world is not a new approach. Nevertheless, there has been a growing interest in using its properties in conjunction with machine learning and data mining techniques in order to build efficient systems. The work presented aims to demonstrate the viability and utility of this type of systems for the credit scoring problem.

Glossary

ACL	Agent Communication Language
AI	Artificial Intelligence
AID	Agent Identification
AMS	Agent Management System
BDI	Belief desire and intentions
CA	Checking Account
CART	Classification and Regression Trees
CBR	Case Based Reasoning
CRMC	Credit Risk Management Cycle
DAI	Distributed Artificial Intelligence
DF	Directory Facilitator
DM	Data Mining
FICO	Fair Isaac Corporation
FIPA	Foundation for Intelligent Physical Agents
FIPA-ACL	FIPA-Agent Communication Language
GA	Genetic Algorithm
GP	Genetic Programming
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronics Engineers
JADE	Java Agent Development Framework
JESS	Java Expert System Shell
MAS	Multi-Agent Systems
ML	Machine Learning
MTS	Message Transport Service
RBF	Radial Basis Function

SBCS Small Business Credit Scoring

SVM Support Vector Machines

Contents

List of Figures

List of Tables

List of Algorithms

1. Introduction

Nowadays, people are becoming increasingly dependent in loans from financial institutions. However, it is not an uncommon situation the fact that some people are incapable of correctly assessing the type and amount of the loan that is affordable to them. As a consequence some people tend to delay their monthly installments or, in extreme cases even become incapable of repaying their debt back to the financial institution. A client's history provides an excellent source of information for predicting the behavior of future clients. In fact, some rules and patterns can be identified in this data history that may be relevant when deciding where future clients have their loan application accepted or not. From the perspective of information as an asset, this client's history data usage creates valuable assets to an organization. The information gathered from these sources is considered to be one of the six types of assets for organizations, namely it falls into the category of IT information asset (1). Furthermore, the best managed companies recognize information as a key asset, and focus more on information than technology while optimizing their business performance (2). In this context, many financial institutions are implementing or improving client classification systems in order to distinguish good from potentially bad clients.

Statistical analysis and deterministic systems are still the most common classification systems financial institution use in their applications, and there is here a practical opportunity to develop alternative systems based on artificial intelligence (AI) (3) and supported by AI techniques. In fact, AI, in general, and data mining (DM) processes, in particular, have an interesting appeal when considering that they may help developing autonomous decision mechanisms able to learn, induce and react to changes and new trends in real time. Only in present times are these financial instructions conducting studies in order to evaluate how techniques from AI, machine learning (ML) and DM can be used to predict client behavior (4), (5).

This dissertation is particularly aimed at credit scoring systems for consumer credit using previous client data records to predict and avoid those classified as bad clients in terms of debt repayment.

1.1. Motivation

Over the past years it was possible to observe the tendency to consider the information inside enterprises as an asset more important than technology itself. In this context, enterprises have focused their investment in information management leaving technology to a role where it is considered a utility. As a consequence, information is considered one of the main competitive advantages leading to the development of the concept of business intelligence, and in turn, the development of information analysis and management (6), (2). With this approach, it is possible to move from the correction of fault and failures to take actions to prevent them from happening while maintaining resources more efficiently. This makes it possible to make better decisions and provide further assistance to the decision making process (6).

One point of concern with this analysis is related to the informational assets that have to be characterized with cost, risk, quality, accessibility, utility and reusability so they can be used in quality information frameworks (7). These assets are managed by data governance and authors such as Khatri and Brown even consider this type of asset to be one of the 6 most important assets of an enterprise. In this approach authors also characterize data assets by the principles of quality, access, lifecycle correlating them with other IT assets within a company (1) (8).

In literature, it is possible to find the process of Data Warehouse as a way to obtain relevant information from a set of data that can be used in the creation of strategic informational assets. In this process, large quantities of data are analyzed and manipulated so knowledge can be extracted capable of being used in strategic decisions. In order to guarantee that the process is successful four principles in the data gathered should be met (9):

- Data profiling: the process of inspecting the characteristics and attributes of data to discover and identify problem areas within the data;
- Data quality: the data quality phase allows the correction of the problems and anomalies found in data profiling;
- Data integration considers the integration of data from multiple sources to combine information across different systems;

- Data augmentation: Data augmentation helps to complete data management cycle by appending records with other data available in the environment.

A different and more recent approach is now being evaluated in the literature by AI techniques applied to management and extraction of knowledge from sources of data in order to produce information assets which in turn may influence the decision making process and unveil future trends and behaviors. As an example Faiz e Edirisinghe (10) introduce the application of neural networks, specialists, fuzzy logic and case based reasoning in asset management as well as their advantages and disadvantages. Waterbury (11), explains how AI improves asset management once again with the help of neural networks, fuzzy logic and system of specialists. These references are then associated with innovative methodologies to manage informational assets and reduce maintenance costs with decision and predictive systems from these systems. There is evidence that informational assets are of great interest to enterprises and institutions but for that to happen it is necessary to build these informational assets from operational data. These steps have been assured by business intelligence platforms and data governance platforms. However recent studies are evaluating the possibility to include techniques from AI and ML processes to improve and create better informational assets from raw data.

Finally, current developments on our society suggest that credit scoring has gained a lot of importance, especially in terms of loan and bankruptcy predictions. Due to this fact, a study on how information assets gathered through financial institutions can be used to support and predict future trends is motivated by their increasing importance and influence on decision making processes.

1.2. Objectives

Information is seen as an important asset in every enterprise, it has a preponderant importance in any enterprise so the way it is treated and manipulates is of great interest.

The aim of the work presented in this dissertation is to prove the utility of ML and AI techniques in large scale decision systems which transform data into information and knowledge to perform their tasks. More precisely, credit scoring applied to consumer credit in unsecured markets problem will be addressed. Consequently, the work developed and detailed is supposed to implement metrics and information flow management taking into consideration the value of information within the enterprise or institution.

The developments made are specifically targeted at decision making and suggestive systems able to support financial institutions, decision makers and clients of those institutions. The list of general objectives is:

- Perform a study on decision algorithms based techniques from ML and AI that rely on data taken from datasets as well as interesting points of optimization for such algorithms;
- Review both current projects on multi-agent systems or other systems capable to support and to make decisions and current trends on system design to integrate machine learning algorithms into these systems;
- Improve previously studied decision algorithms with inspiration on reviewed optimization techniques;
- Develop a suggestion system able to work with the algorithms reviewed and to make suggestions based on incomplete information and learning algorithms to find ideal situations;
- Develop a multi-agent system able to handle the transformation of raw data into information and knowledge performing decisions based on learning algorithms and suggestions to both clients and financial institutions.

As the decision process at financial institutions is important for their operation, there is a demand for good decision systems able to classify clients with the maximum accuracy possible. Moreover, a suggestion system may also help financial institutions find trends and spot interesting client profiles who are perceived by the system as good clients. From the client side, the same suggestion system may help clients follow the necessary requirements to have a loan accepted and understand how they can improve their client report at these institutions. The multi-agent system itself should use the known properties from agents and their relationships to assure that the multi-agent system accomplishes their general goals, being them suggestion or decision making.

In conclusion, the work presented in this dissertation will provide both theoretical studies on decision and suggestion systems as well as a practical implementation of a credit scoring system for the chosen credit scoring problem.

1.3. Investigation Methodology

In order to accomplish the objectives described in section 1.2, Action-Research methodology was followed (12). This methodology initially identifies a problem so hypothesis can formulated that focus the development of the investigation. Consequently the information is recompiled, organized and analyzed continuously building a proposal to solve the problem identified. Finally, conclusions based on the results obtained during the investigation can be taken. In order to follow this research methodology six complementary stages have been followed to achieve the planned objectives. The stages defined are described below:

- Specification of the problem and its characteristics;
- Constant and incremental update and review of the state of the art;
- Idealization and gradual and interactive development of the proposed model;
- Experimentation and implementation of the solution thru the development of a prototype;
- Result analysis and conclusions;
- Diffusion of experiences, results and knowledge acquired with the scientific community.

1.4. Structure of the document

This dissertation is composed by a total of seven chapters being the first the present chapter, the introduction, where a brief overview to the theme of this dissertation is made containing motivation, objectives and investigation methodology used to justify the work presented in this dissertation.

Chapter 2. contains the problem analysis which includes an introduction to the problem of credit scoring detailing its formal definition, historical data and areas of application. Common features to this theme are analyzed such as common data gathering procedures, attributes used for the credit scoring problem and legal considerations on the use of attribute data on different countries. Both traditional and modern models are also mentioned, as well as, some known projects and applications of credit scoring. The chapter ends by providing an overview of current tools and framework used to solve the credit scoring problem at our disposal today.

The following chapter, chapter 3. , reviews classification algorithms present in the literature related to the problem of credit scoring. It is given more importance to the most used or most investigated algorithms in the research community. Apart from these algorithms, some optimization techniques made by authors to some of these algorithms are also reviewed, as well as, algorithm combinations that boost the classification accuracy. The chapter ends with remarks from the comparison of the algorithms discussed assessing their utility strengths and weakness.

In chapter 4. , the concept of multi-agent systems is introduced, detailing the purpose of an agent and the characteristics of a multi-agent system. To better introduce some of the concepts used by this computational paradigm, known properties and standards developed are discussed. Following, it is presented tools and frameworks available for use today which comply with known standards and properties of such multi-agent systems.

Over chapter 5. , it is described the development process of the work presented in this dissertation beginning by the study on classification algorithms through the elaboration of a suggestion system and the planning of the multi-agent system for credit scoring.

In chapter 6. , the results obtained from the developments made in section five are presented and discussed. These include considerations from the proposed decision classifier algorithms and the comparison with work from other authors. Even more, the suggestion system is assessed in a series of business cases demonstrating the utility of such system and the kind of problem it may answer. The results provided also demonstrate the utility and the validity of the proposed multi-agent system. With regards to the multi-agent system, interesting properties inherent to the developed system are also presented such as overall availability and fault tolerance.

Finally, the document ends with chapter 7. , the conclusion, where a synthesis of the work done is presented evidencing its main conclusions. Afterwards, important contributions and work generated are enumerated. This final chapter ends with the suggestions for future work that has yet to be done.

2. Problem Analysis

Credit scoring is an important problem for financial institutions and it may enable them perform activities such as distinguish their clients as good or bad for loan purposes. More specifically, it is intended that with the usage of credit scoring techniques, financial institutions are able to predict client behavior during the time it takes them to repay the loan. In order to build credit scoring models, financial institutions use their clients' credit history. A client's credit history is an individual record of all loans, accounts, late payments and late deposits each client has. With this data, a credit score is attributed to each client, which should then divide clients by their credit worthiness.

Although there are standard scoring systems, not all countries use them. Even worse, the use of credit scoring models by financial institutions is not without mistakes incurring in bad client evaluations. Those evaluations can have two different consequences, classifying a good client as bad thus losing a business opportunity or classifying a bad client as good, resulting in a debt that is not correctly repaid or a defaulted debt.

In Portugal, the Banco de Portugal has seen the number of overdue loans increasing in the last two years according to the statistics made by the Banco de Portugal. In these studies, the number of private customers with overdue loans is about 14,3%, the highest percentage of the last two years (13).On other end, Figure 1 demonstrates the evolution of consumer bankruptcy filings in the US over the last years. The tendency clearly shows that the number of bankruptcy filings has been increasing exponentially over the last couple of years results in increase of people who cannot repay their debts. The two white bars in 1978 and 1994, presented in Figure 1, present years on which reforms on bankruptcy in the US were performed, however they seem to have low impact on the exponential increase of bankruptcy filings. These facts provide factual data which gives even more importance to credit scoring models, as the development of new scoring system may reduce this number obtaining better overall performances enabling the financial institutions to make better decisions.

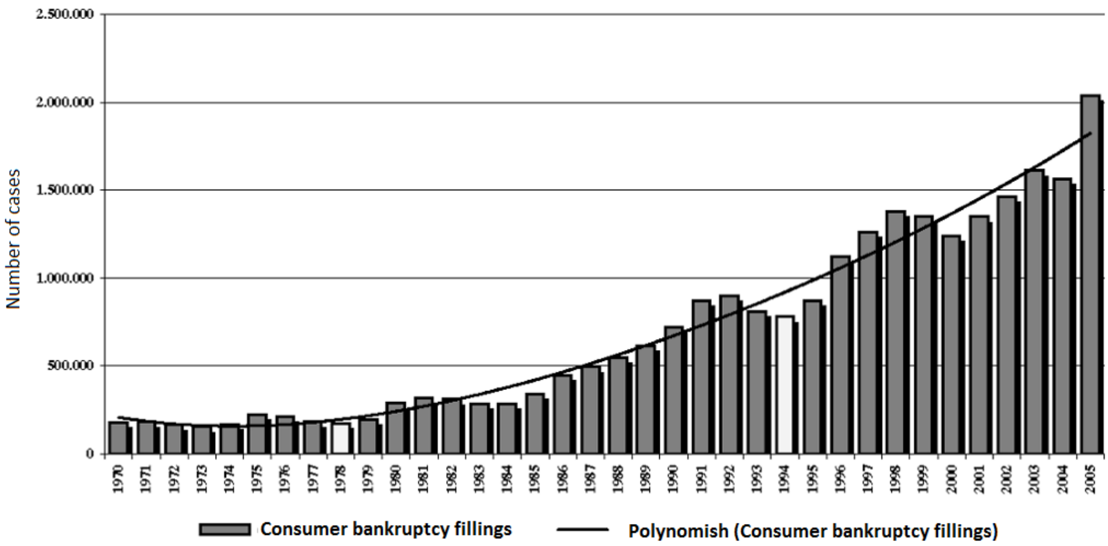

Figure 1 - Consumer bankruptcy fillings (14)

As credit scoring is a complex problem, in this section, it will be discussed its formal definition, history and applications, as well as, common features essential to build credit scoring models and the most used set of attributes in problems of credit scoring. Afterwards, traditional and modern models are described demonstrating both implemented models and models in current study for the problem at hand. This approach will give insight not only on the current state for this problem, but also, current application on the business environment, the usefulness of artificial intelligence (AI) to the process and future trends which might enhance and improve the on this subject.

2.1. Credit Scoring

Credit scoring it is used by the majority of financial institutions today, though, only recently has the research community devoted more focus to this classification problem. Prior to 2000 there were few books that addressed this problem, and even until 2005 only about 15 books had been published on the subject (15). Since 2005, the number of book publication on credit scoring and themes related to credit scoring has increased greatly. Simply searching for the term credit scoring or credit score on most search engines delivers more than 15 books published in recent years. This proves that credit scoring is an increasingly hot topic for financial institutions these days.

Historically, the credit problem is documented since about 5000 years ago and consequently it is considered one of the most ancient financial problems, however the credit scoring problem as it known today is only considered to be about 60 years old (16).

Credit scoring is regarded as a business process to aid the decision of lenders, in this case financial institutions, whether or not lend money to client, assess the risk of their investment and ensure the maximum profit for the loan providers. A simple definition for the credit scoring is given by Anderson (15) which defines it as "(...) the use of statistical models to transform relevant data into numeric measures that guide credit decisions". This definition considers only the model creation and its influence on credit decision. However, more complete definitions were also found in the literature, which detail the influence of this process on credit decisions. Consequently, Thomas, Elderman and Crook in (16) define the process of credit scoring as "(...) a set of decision models and their underlying techniques that aid lenders in granting consumer credit. These techniques decide who will get credit, how much credit should they get, and what operational strategies will enhance the profitability of the borrowers to the lenders". This definition clearly states that credit scoring is more than the traditional client assessment view. It also develops considerations about risk, profit and the use of different models and techniques in the decision process.

The first realization that statistical techniques used to spate groups could be used to separate good borrowers from bad ones was made by Durand in (17). Despite that fact, only when credit cars arrive to the market did financial institutions realize the usefulness of a credit scoring process. Until then people relied more on manual process to assessment of credit opportunities disliking statistical methods where the decision could be automated. The high number of credit card applicants made it financially impossible to have manual process for credit approval so credit scoring through statistical analysis and automated decision where supported and encouraged (15), (16). Credit scoring proved to have great results especially in market with a high range of clients which produce low valued credit. Manually, these credits would be impossible to assess and still be profitable due to manual labor involved and low profit margins, but with automatic decision provided by credit scoring methods, financial institutions could improve their reach, allowing them to benefit from small credit using speedier loan approval processes with lower operational costs. Of course, like most models and technologies credit scoring might also have its own problems that need to account for. One of these problems is the privacy which concerns the amount of data that is used by this process and the assurance that it is only used for that purpose. Other problem concerns the case of blacklisting clients, which can take some time. This happens due to the fact that in order to blacklist a type client it first has to become relevant to model and techniques used to perform credit scoring thus allowing some amount of wrong evaluations before models update themselves. Moreover, credit scoring is often a process designed to produce a score using statistics and correlations but lacks the causality of the decision. That means that the reason for the score provided by the process might be difficult or even impossible to explain. The main uses of credit scoring are defined by Anderson in (15) in the following list of classes:

- Application score: applications are evaluated as business proposals which are then assessed based in information about current credit considerations of the applicants, past deals from clients;
- Behavioral score: uses information on client's credit score to perform account management based on each client profile. Examples where the behavior score is useful include fraud detection and bankruptcy prediction;
- Collections score: used so companies can make the best decisions on delinquent accounts to balance the costs of collecting against recoverable revenues;
- Customer score: uses information about different clients' accounts to perform activities such as advisory, account management and cross-selling;
- Bureau score: is a credit score provided by a credit bureau, a credit reference agency which collects information from different sources to provide consumer credit information. This score may indicate a predictor of bad behavior or bankruptcy predictor of the data held by a financial institution.

Credit scoring activities may also be managed by processes found in the literature such as the Credit Risk Management Cycle (CRMC), Figure 2. This process deals with an account management cycle, from the day a credit business opportunity is initially assessed until it is fully repaid. In this area, several credit scoring applications should be used together with CRMC to perform some tasks on some stages on the CRMC cycle. The CRMC cycle illustrated by Figure 2 and introduced in (15) comprehends a total of 7 stages from the process:

1. Segmentation: where clients and products to be target are defined, which can be done by customer score;
2. Solicitation: is responsible for the marketing campaigns to entice desired customers;

3. Acquisition: formalization and acceptance of new business applications which are considered good which can be done by customer score and application score;
4. Management: account management dealing with repayments, queries and other normal activities account management activities to which behavior score might help;
5. Collections: used when in the presence of bad behavior in credit repayment preserving the customer relationship. In this stage collection score and customer score might provide useful;
6. Tracing: attempt to find clients who move without informing of address or contact changes;
7. Rehabilitation: used to collect money back from the client on the gravest delinquencies resulting most times in loss of customer relationship.

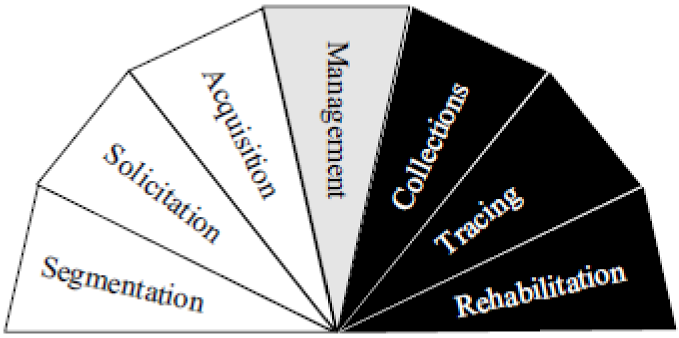

Figure 2 - Credit Risk Management Cycle (15)

The CRMC is filled with opportunities to use credit scoring application to assure the correct client selection and repayment of credit applications. Even in bad scenarios such as delays in credit repayment, credit scoring proves itself useful for the financial institution.

Different types of markets for the application of credit scoring can also be presented and added to specific CRMC cycles. For instance, the unsecured market considers debts where there are no collaterals defined, specific assets of the borrower, to account for the case of a bankruptcy or failure to meet the terms for repayment to financial institutions. These loans are often used by borrowers for small purchases or unexpected expenses.

On the other hand, there is the secured market where debts are secured by collaterals, assets that a borrower specifies as guarantee in the case of bankruptcy or failure to meet the terms for repayment. These loans carry less risk for financial institutions as the collateral may be executed by the financial institution, which takes possession of the asset and may sell it to regain some or the entire amount originally lent to the borrower.

Other markets may also be considered for the use of credit score, for example, store credit, where a store provides credit to its clients so they can buy goods from it. Here, the credit scoring may be used to distinguish clients who are likely to repay their debts without any problems. Service provision is also another area where credit scoring may be applied, to know which clients deserve a service on a credit base which is paid on a later time. The last example presented in this analysis considers enterprise lending, the case where enterprises ask financial institutions for loan to account for a situation where they need to increase their liquidity. In this case credit scoring evaluates if the enterprise is a valuable client and which conditions are the most beneficial for both borrower and lender. Examples of applications in each of these markets are here presented:

- Unsecured: credit cards and personal loans;
- Secured: loan mortgages, loan for motor vehicles;
- Store Credit: credit to commercial stores;
- Service provision: phone contracts and municipal accounts;
- Enterprise lending: trade credit, enterprise loans.

For the purpose of the work presented in this dissertation the application score and customer score, as well as the unsecured and store credit market will be the main areas of study inside the credit scoring problem. In these cases, the methods most used rely on scorecards to make the credit scoring decisions. A simple definition of a scorecard in the context of credit scoring is proposed by Siddqi in (18), who defines a scorecard as "(...) a group of characteristics, statistically determined to be predictive in separating good and bad accounts". These score cards are often made recurring to some statistical techniques such as regressions to provide attempt to provide a quantitative measurement of the likelihood that a customer will display behaviors like loan default or bankruptcy with respect to their current or proposed credit position with a lender. The work presented in this dissertation will also study different techniques from machine learning (ML) and data mining (DM) to perform these scorecards in order to perform credit scoring on consumer credit.

2.2. Common Features

In order to perform credit scoring, on the specific case of consumer credit, there are a number of common features which need to be addressed. One of the most important features any credit scoring process has to deal with is to assure data availability and quality. The first considers ift there are enough cases to train models while the second assures the correctness, soundness and completeness of the data used. It is important that any data gathered for this purpose to be statistical significant for the problem obtained by a random process. Siddiqi in (18), considers the minimum number of client instances used to train a credit scoring model should be of at least 4000 instances of which 2000 represented good borrowers and 2000 bad borrowers.

The type of client attributes used in these models has some similar set of attributes that are considered important to build such mechanisms. Generally, at least 4 types of client attributes indicators are used:

- Demographic;
- Financial;
- Employment;
- Behavioral.

An example of attributes of each of these categories can be found in Table 1. These indicators are important because both by their soundness in helping to estimate a client's risk probability and their explanatory power when a credit-scoring method is employed to analyze a loan application (3).

Table 1 - Client Attribute Characterization (3)

Demographic Indicator	Finance Indicator	Employment Indicator	Behavioral Indicator
Age	Total Assets	Type of Employment	Checking Account (CA)
Sex	Gross Income	Length of current employment	Average Balance in CA
Marital Status	Gross Income of Household	Number of employments over the last x years	Loans outstanding
Number of dependants	Monthly Costs of Household		Loans defaulted or delinquent
House Status			Number of payments per Year
District of Address			Collateral / Guarantee

Although many of the attributes used are often shared between credit scoring models the classes they are divided into often differ. Consequently, another accepted division of the most generic attributes used for the credit scoring problem divides the list of attributes into five different types (19).

These five attributes types are:

- Character: attributes to identify if a client is willing to repay the loan. Examples of such attributes can be personal characteristics, credit history and job stability;
- Capacity: attributes to identify the client financial ability to repay the loan. Examples of such attributes are client's income and expense statements;
- Capital: attributes to identify the current client's assets and acts as a insurance against unfavorable situations. Examples of such attributes is the amount in a savings account;
- Collateral: attributes to identify assets that the client uses as a guarantee if the primary source of repayment for the loan fails. Examples of such attributes is property like buildings, houses and terrains;
- Condition: attributes to identify the macro-economic which may have impact on the client's ability to repay the loan. Examples of such attributes are job nature and inflation rates.

Along with data gathering legal issues must also be of prime concern as any decision made that does not meet the appropriate laws is considered illegal and must be disregarded. Client discrimination based in attributes such race or gender is generally illegal in most countries and may justify legal suites those who ignore these considerations. Table 2 presents a list of variables with use restriction on some countries. In this list 0 denotes no restrictions for use while 1 represents use restriction (14).

Table 2 - Legal restrictions on variable use for credit scoring (14)

Variable	U.S.	Germany	U.K.	France
Gender	1	0	0	0
Marital Status	1	0	0	0
Ethinic Origin	1	1	1	1
Color of Skin	1	0	0	0
Nation Origin	1	0	0	0
Age	0	0	0	0
Political Opinion	0	1	1	1
The trade union Membership	1	1	1	1
Religious Belief	1	1	1	1
Health data	1	1	1	0

On the other hand, some attributes despite being legal might be considered culturally unacceptable, for instance poor health and driving records. Consequently these models should take into consideration the aspects stated above which might involve discarding or aggregating attributes so they can be used without raising legal issues or cultural turmoil (5).

Despite being careful about the data usage credit scoring system must also obey rules and regulations imposed by international treaties which concern the credit risk and its influence on the financial institution. Basel II is an important treaty used worldwide by most financial institutions which defines between other things the amount of capital financial institutions must retain in order to cover the operational risks of conceding credit to customers while retaining a good financial performance. The main reason for this treaty is to ensure that financial institutions have enough capital reserves to face the risk they expose themselves through their lending and investment practices (20). Predictable, a credit scoring model must implement its guidelines in order to be accepted for commercial use in institutions which have signed this treaty.

2.3. Traditional Models

Most financial institutions use statistical pattern recognition models to build their own decision mechanisms (3), (5). Linear Discriminant Analysis is one of such the models, this technique aims to classify a heterogeneous population into homogeneous subsets on which the decision mechanism is further developed.

Some critics to this model mention that this model is only optimal for a small class of distributions and ideal for the definition of bad and good groups in the specific case when no clear boundary exists between the two and when it is pretended to test the significance on individual variables. In Czech and Slovak Republics' financial institutions the most used technique is the Logit Analysis which is an improvement upon the Linear Discriminant analysis technique. This model is considered an extension from the Linear Discriminant Analysis which accounts for the non-normality of the data used.

Logit Analysis has some disadvantages as it does not account for missing values, is sensitive to high correlation among explanatory variables and is also limited by its parametric model, however it is considered by several studies to have better performance than Linear Discriminant analysis model (3).

On the other hand, countries like U.S.A. and Canada use a standard credit scoring system on their financial institutions, the FICO model. This model uses several different variables from the client credit history. In the FICO model, Figure 3, five different types of variables are used each with different weights for the model:

1. Payment history (35%), a client history of re-payment to the financial institutions;

2. Debt (30%), a client current list of debts;
3. Length of credit history (15%), time of creation of the first credit score;
4. Account diversity (10%), the set of account types on the clients history in the same or different financial institutions;
5. The search for new credit (10%), number of inquiries a client has for new loans.

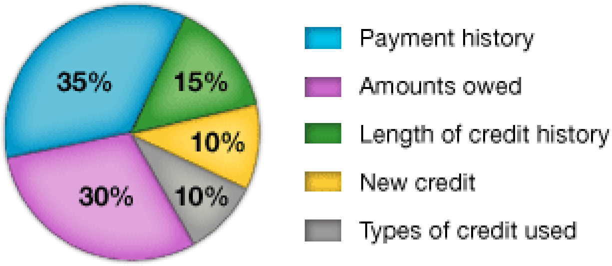

Figure 3 - FICO scoring model

In the end, the FICO scoring model produces results which range from 300 and 850 for each client which represent its credit worthiness (21).

Analyzing Table 1 it possible to see that the type of attributes present in that list are consistent with the attributes also present in the standard credit scoring system FICO (21) presented at the beginning of chapter 2. Apart from the attributes in the demographic indicator, all the other types of attributes are used by the FICO system. FICO also seem to cover all the 5 different types of attributes reviewed by Beares, Beck and Siegel (19).

Although it is a standard model FICO is not without criticism. This model is considered to have become less effective as institutions become more dependent on it. Even more, there were found sly maneuvers to increase one's FICO score such as increasing the limit of a credit account thus increasing the client available credit a factor which has high importance for the FICO system (22).

2.4. Modern Models

Modern scoring mechanism are mostly based in ML and DM techniques associated with AI. According to Faiz (10) AI consists of neural nets, expert systems and fuzzy logic. These techniques provide the ability to monitor assets in real time dealing with the unknown and unpredictable, where the goal may be to reduce asset maintenance expenses, improve utilization and product quality.

A neural network is a system that simulates the operation of the human brain and performs tasks such as data-mining, pattern recognition, classification and process modeling. Despite existing for over 50 year neural networks has only recently increased its practical application due to low cost computer with high processing speed. Examples can be found in a Jordanian bank studies where studies were conducted to evaluate the benefit of using Multi-Layer Feed Forward Neural Networks in a credit scoring system. These structures receive a set of client attributes which are evaluated by a trained neural network to give an output with the assessment. Despite not giving any explanation on how the decision is built the studies conducted led to the conclusion that these structures are in fact good classifiers achieving up to 95% correct evaluations in their tests (4). Other usages of neural networks as an ensemble for multiple classifiers can also be found for the prediction of bankruptcy and credit scoring. The results show that only one architecture performs better than the original Neural Networks algorithm and as so the author suggests studies with when choosing the best ensemble architecture (23). Testing neural networks also showed that they achieve less cost of misclassification than other traditional models like Discriminant Analysis and Logic Regression by accepting less bad clients than other models and still identifying more bad clients correctly (5).

Fuzzy logic is based in what is considered degrees of truth as opposed to the Boolean classification true or false. Fuzzy logic may also behave like the human brain in a sense that it aggregates a number of partial truths into a higher-level truth. With an threshold indicator measuring the level of truth is exceed activities and tasks may be triggered by such system.

Expert systems, unlike the previous two techniques simulate the judgment and experience of a set of organized knowledge experts in a particular field. Frequently, these systems include an expert knowledge base containing accumulated experience and a set of rules used to apply the knowledge base in a particular application. Both the rules and the knowledge can be updated and it may happen in real time as a continuous analysis.

Case Based Reasoning (CBR) is also considered in (10) for the process of decision making. This approach uses a set of problems and answers in a Case Archive. When the system is presented with a new problem the answer is derived from the most similar cases with the necessary modifications. This model is used to solve problem were expertise is important and it works by storing past experience in a form of cases from past events characterized by a set of attributes. Then, when a new case is presented to the model similarities between past cases and the present case are calculated using an appropriated function that adapt solutions from the past cases to the present case, thus classifying it based on the most similar case. If the result proves to be corrected the new case can be added to the set of past cases to help discovering solutions to other problem (24).

Other modern approaches use algorithms from artificial immune systems in order to solve the credit scoring problem. In (25), it can be seen three different algorithms using this approach and it also provides a comparison between other modern and traditional algorithms. This comparative study aims to prove that artificial immune system algorithms are a valid and serious option for the credit scoring problem.

Different approaches also make use of financial liquidity to forecast a client's ability to pay a future installment. From an historical set of financial liquidity of clients and their behavior and by comparison with a present client its risk is calculated and appropriate action may be taken before transgression happens if necessary. In this approach the client's history is analyzed every few months and he is classified as good, medium and bad customer according to his history and future prediction of invoice payment (26).

2.5. Credit scoring projects

During the literature review made, related projects which dealt with credit scoring mechanisms for the bankruptcy problem were found. Tsai and Wu in (27) propose the use of neural networks, more precisely Multilayer Perceptron, to assess credit bankruptcy. Their approach considers the application of single classifiers and ensemble multiple classifiers which are combined to produce a final classification. The conclusions obtained from their work stated that single classifiers performed better than ensemble classifiers. The same problem is also object of study in (28), however this approach is based on Support Vector Machines (SVM) to perform predictions. The work focuses in finding the optimal kernel function parameter values to obtain the best classification results. The parameters to estimate in this approach are C and γ from the radial basis function (RBF). To obtain the best parameter values a grid search 5-fold cross-validation search is used. Comparing results obtained with this approach with regression and neural networks algorithms it was found that this approach outperformed the rest.

Other studies demonstrate the concern to make the decision process more comprehensible where opaque classifiers are used. By opaque classifiers it is intended to mention those classifiers with complex mathematical functions and hard to understand how different attributes are assed in the decision process. In (29), there is a proposition for rule extraction from SVM. To this end two different approaches to rule extraction are reviewed, pedagogical and decompositional. The decompositional approach is related to the internal structure of the SVM from which rules are defined. The pedagogical approach tries to combine input and output of the SVM to derive the rules. The conclusion of this study shows that the performance of the rule extraction SVM has a small performance gap from the traditional SVM algorithm.

The effects of Small Business Credit Scoring (SBCS) are assessed in (30) where an analytical framework is proposed to assess its influence in the lending business of financial institutions. From the proposed analytical framework empirical results on the effects of SBCS on the quantity, price and risk of small business credits are shown and commented. The main conclusions of this analytical framework to assess the effect of SBCS are that the effects of these technologies differ from small to slighter larger credits. Effects are also influenced by the type of technology used, if it is rule based or base on discretion methods.

2.6. Tools and Frameworks

To build any credit evaluation application, many financial institutions make use of existing frameworks with a large set of techniques to help the process of data mining.

The most used frameworks to perform DM and apply ML techniques in enterprises and institutions are proprietary solutions. In order to prove that both Gartner (31) and Forrester (32) reports were analyzed, as well as, the magic quadrant and the wave charts. These studies indicate that the most used solutions in a field of business intelligence are from IBM, SAS, SAP, Oracle and Microsoft. Even more, both these reports besides indicating market leaders among these technologies have also described future trends for this technology and client demands for future releases. Some of highlighted points on current client demand include:

- A move to shift priorities from measurement analyses to predictive analytics;
- The development of better predictive tools;
- A greater focus on content analytics;
- The need to perform forecasting optimization tasks.

A more focused study performed by Forrester (33) specifically on DM tools which hold predictive analytic features points products from SAS Institute, SPSS, IBM and Oracle as market leaders. To this list KXEN and Portrait Software products were also added as market leaders in this specific context. All these vendors implement in their products the ability to perform tasks like:

- Data preparation features;
- Data analysis;
- Visualization of predictive models and results;
- Handle multiple database platforms.

As most of the reviewed tools are only commercially available, some free open source tools are also presented. In this context, open source tools like RapidMiner (34) or Weka (35) provide a vast list of DM and ML techniques that can be used in conjunction with any other application. Those two tools also provide libraries that can be imported to custom programs. Rapidminer (34) for instance is used by the Bank of America. Other example of open source DM and ML tools include Orange (36), written in C++ and phyton, and KNIME (37), written in JAVA, which provide the same basic functionalities of the Weka and Rapidminer. KNIME has won also been distinguished by Gartner as a cool vendor in Analytics, Business Intelligence, and Performance Management areas.

In a more specialized context, namely neural networks, it can be found Encong (38), a comprehensive framework for neural networks. To evaluate evolution algorithms for data mining, KEEL, is also mentioned. This tool allows to evaluate different evolution algorithms as well as to integrate them with other software tools (39).

In this dissertation's work, the Weka Toolkit (35) was used to perform the tests upon some of the algorithms proposed. This decision was made due to the fact that this framework has a collection of ML algorithms for DM tasks which can be applied directly in a dataset or used in a java program and previous experience with the tool on related studies (40). Weka has also an active support community and their program is released as open source software.

3. Learning Algorithms

After reviewing the credit scoring problem detailed in chapter 2. , in this chapter it will be discussed an overview of learning algorithms. These algorithms were selected by their importance for the problem and also by their current usage in the finance industry and research. As the most used models are regression algorithms, and decision trees, that will be formerly presented. On the other hand, neural networks will be addressed due to their presence in a large quantity of research papers on the subject.

Some optimization techniques on classification algorithms will also be reviewed focusing on the strategies implemented. These optimizations aim to target algorithms able to better predict a client risk. This review targets one of the objectives of this dissertation: the study of decision algorithms based techniques from machine learning (ML) and artificial intelligence (AI).

3.1. Classification Algorithms

The algorithms explained here will be focused on the algorithms used in traditional and modern models to perform credit scoring activities such as loan approval. Some of the algorithms used are even very popular for several data mining (DM) tasks as they are counted as the top ten most used algorithms in the area. In this category are included the Multilayer Perceptron, the Decision Tree and the Linear Regression (41).

3.1.1 Multilayer Perceptron

The Multilayer Perceptron is an algorithm that uses a feed forward neural network with back propagation to classify instances. In this network each neuron has a weight attributed to him and it also uses a nonlinear activation function which was developed to model the frequency of action potentials of biological neurons in a brain. The most common activation functions are sigmoid and they are used in this algorithm. This structure uses 3 types of layers in the network, one input layer, one or more hidden layers and one output layer to provide the results. Another property of this type of neural network is that there are no connections between neuron in the same layer, however neurons are fully connected between layers. The back propagation learning algorithm changes the weights in each neuron after each instance of a dataset is processed based on the amount of error in the output compared to the expected result. We represent the error in output node j by $e_j(n) = d_j(n) - y_i(n)$, where d is the target value and y is the value produced by the Multilayer Perceptron. Then corrections are made to the weights of the nodes based on those corrections which minimize the error in the entire output, given by the equation 1 (35).

$$\in_n = \frac{1}{2} \sum_j e_j^2 (n) \tag{1}$$

3.1.2 **Decision Trees**

Decision trees are popular methods, robust for noisy data and capable of learning disjunctive expressions. In particular, C4.5 decision trees are even considered among the most used DM algorithms nowadays, being evaluated as one of the top ten algorithms in DM (41). These decision trees in each of the internal nodes specify a test on some attribute from the input dataset. Each branch descending from a node corresponds to one of the possible values of the attribute specified for that node and each test results in branches that represent different outcomes of the test. The basic algorithm to induce the decision tree is a greedy algorithm that constructs decision trees in a top-down recursive manner. Additionally it implements a divide and conquer strategy to build the model. The algorithm starts with a single node representing all the data in the dataset. If the sample of data considered is of the same attribute class then that node becomes a leaf in the decision tree. Otherwise, the algorithm chooses an attribute that better divides the sample data into individual classes of that attributes. The process is recursive and ends when the sample data in a node is all of the same attribute class or when there are no more attributes available, that were not used yet, to divide the sample data.

The decision tree often uses an entropy-based measure as a heuristic for selecting the attribute that will best split the sample data into separate classes. In each round the algorithm computes the process above described known as the information gain for each attribute and then chooses the one with the highest information gain as the test attribute of the sample data and perform the split point. The best split point is easily evaluated considering each unique value for each attribute in the sample data as a possible split point and calculating the information gain of each one (42).

3.1.3 **Linear Regression**

Linear Regression and Regression methods such as Linear Discriminant Analysis and Logit Analysis are two widely used algorithms by financial institutions to perform credit scoring (3). In the Linear Discriminant Analysis the objective is to choose the linear combination of explanatory variables which can separate most subsets of data with the maximum distance between means of these subsets. In the equation 2, G is a subset of data, w is the weight associated to each explanatory variable in the vector represented by x. This models works by scoring a set of client explanatory variables in different subsets of data achieving for each one, from the equation 2, a score and with this process reduce the problem to one dimension with a defined group of clients which share more common characteristics on which the decision mechanism can be further developed.

$$A_G = \left\{ x \middle| \sum w_i x_i > c \right\} \tag{2}$$

The Logit Analysis is considered an extension of the Linear Discriminant Analysis which accounts for the non-normality of the data used. It uses a vector of application characteristics x to calculate the probability of a client related to the vector of explanatory variables x with the equation 3. The estimation of the parameters w is done using the maximum likehood method.

$$\log\left(\frac{\rho}{1-\rho}\right) = w_0 + \sum w_i \log x_i \tag{3}$$

Although Logit Analysis is considered to have a better performance than the Linear Discriminant Analysis it is susceptive to high correlation variables, it is limited by its parametric model form and cannot handle missing values in their variables.

3.1.4 **Other Classification Algorithms**

The list of classification algorithms reviewed is not exhaustive due to their lower performance, interest to the work of presented in this dissertation or appearance in papers and articles in the literature. Thus, some other algorithms will be mentioned here in a less detailed form.

Attempts to solve the same credit scoring problem can be seen by the k-nearest Neighbor Classifier, a cluster algorithm which groups client according to their similarity. This similarity is calculated by the distance between a client attribute values (3).

There also approaches to the same problem using algorithms inspired in immune systems which use structures called B-cells to classify instances in a dataset. The basic algorithm produces B-cells, selects the stimulated cells above a certain threshold which then are cloned and then mutated the each iteration of the problem adding to the B-cell pool the best mutated cells. Variations on this algorithm are also present in the literature and in some cases are able to increase the predictive performance of the initial algorithm (43), (25).

Other potentially good algorithms to perform classification are evidenced by Wu and Kumar in their list of the most used DM algorithms (41). In this list there is mention to algorithms like Support Vector Machines (SVM), the Naïve Bayesian Classifier, the AdaBoost and the Classification and Regression Trees (CART). The fact that these algorithms are widely used in most DM applications today could mean that they may also be good alternatives to evaluate the credit scoring problem. In reality, there are studies which evaluate the use of the CART algorithm (3) and SVM (44). In these studies the CART algorithm was found to perform worse than the Logit Analysis while the SVM was found similar to the decision tree C4.5 algorithm and Genetic Algorithms (GA).

3.2. Algorithm Optimization

As noted above, in this context, AI techniques use ML and DM to produce results. In this aspect it can be found, in the literature, several algorithm optimization proposals.

Improvements in genetic algorithms used for classification exist and can be found in The Two Stage Genetic Programming Algorithm. This algorithm produces a set of if-then rules as well as a function based in genetic programming (GP) to classify instances (45). Another example uses a combination of decision trees with genetic programming and is also able to improve classification (46).

Neural networks are also focus of optimization and some approaches try to make use of feature selection algorithms before constructing the neural network, making some attributes more relevant in this structure (47). Feature selection using decision trees may be also used to determine a set of attributes, those in the upper levels of the tree, to build the subset of attributes that is considered to be used with the Naïve Bayes classifier (42).

All these algorithm combinations obtain improved results when compared to those versions where they are not combined, leading to the conclusion that combining different algorithms maybe a good source of optimization.

3.2.1 Two Stage Genetic Programming

Classification using genetic programming has been proposed as an alternative to neural networks when only small datasets are provided or when the datasets available contain irrelevant attributes which affects the performance of the neural network. Improvements in genetic algorithms used for classification exist and can be found in the Two Stage Genetic Programming algorithm. This algorithm uses a function based in genetic programming and if-then rules to classify instances.

With the generated if-then rules, the decision maker can better understand the contents of a dataset and exploit this knowledge for marketing strategies, failure prediction and association rules but lack the ability to forecast instances which do not fall into any of the rules above. Function based algorithm provide better capabilities of forecasting but lack the existence of intelligence rules. To combine the advantages of these two approaches the Two Stage Algorithm was proposed in order to integrate function and induced based methods.

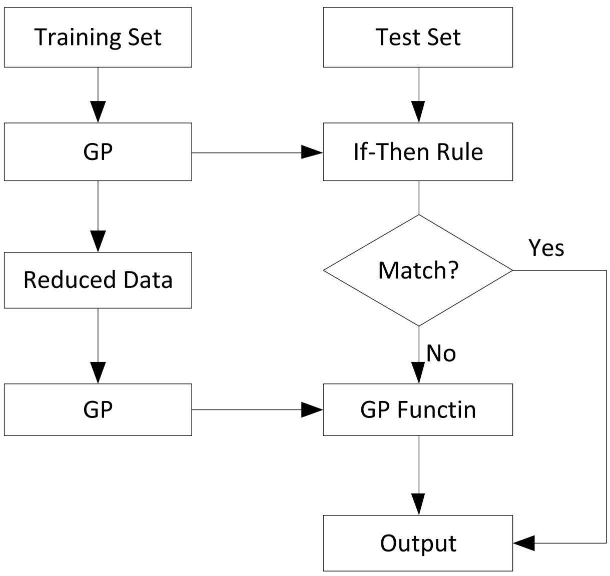

Figure 4 - Two Stage Genetic Programming Algorithm (45)

In this algorithm, illustrated in Figure 4, if-then rules using genetic programming characterizing the dataset are derived initially. Consequently, the data not covered by the if-then rules generated, a reduced part of the dataset, is used in another genetic programming algorithm to form the discriminant function for forecasting. The evaluation of each client is made accordingly to Figure 4, when the existent rules can classify it stops there if not it is then provided to the second genetic algorithm for classification (45).

3.2.2 **Genetic Programming for Data Classification**

Decision trees are usually build using a divide and conquer strategy, however this approach uses ML algorithms and evolutionary algorithms instead, it is a tree-based genetic programming algorithm. While greedy algorithm like the C4.5 is aimed at optimizing locally the construction of the decision tree, genetic programming algorithm performs a global search through the space of possible solutions. This algorithm performs a search by partitioning space of the numerical attributes so then the genetic algorithm can concentrate in finding the tree shape and the right combination of attributes. For this process full atomic representations are used with atom in both the internal and leaf nodes and they are represented by an attribute, operator and value e.g. (Age > 26). Particularly, leaf nodes contain class assignment atoms like $(class := C)$ where C is the predicted category.

The initial simple representation of the tree uses the full atomic representation where for numerical attributes it is used a < operator and for nominal attributes an = operator is used. This initial tree also uses all possible combinations for attribute-value combinations of the dataset, the objective is to give flexibility to the genetic programming algorithm to choose the best attribute-value combination in the tree. A refined representation is also possible by using gain or gain ratio that unlike in algorithm like C4.5 determine $k-1$ threshold values resulting in k partitions where the gain ration criteria determining if there advantages in partitioning the data into a larger number of data subsets (35). Other approach to partition the domain of numerical attributes is clustering similar collections of data based on similarity. The difference is that clustering algorithms do not use target class as it is, but rather divide it natural groups.

Results from this algorithm show that it performs better than C4.5 algorithm to classify instances in a range of datasets with different number of instances and attributes (46).

3.2.3 **Combining Feature Selection with Neural Networks**

Neural network are considered good classifier for credit scoring and evaluating client risk, (4), (5), (47). However, neural networks are also focus of optimization and some approaches try to make use of feature selection algorithms before constructing the neural network, making some attributes more relevant in this structure.

This procedure has essentially two phases: the first, using ideas from information theory, important attributes are selected; in a second phase, a neural network is trained using the dataset with the attributes selected as important.

The first phase uses a set of tupples with a defined number of attributes and calculates the information content of each one of the attributes by their ability to classify the tupples removing the attributes with less information content from the dataset.

In phase two, the resulting dataset is used for classification with a feed-forward network using back-propagation as a learning algorithm to train the neural network. The final results comparing standard neural network and the algorithm proposed show that there is an increase in the accuracy of the predicted values thus showing the improvement made (47).

3.2.4 **Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection**

The Naive Bayesian Classifier is a simple classifier based on probability theory. Here the algorithm works by assigning a probability for each attribute to fall into one of the classes of classification. In equation 4 it is demonstrated Bayes' rule to calculate the probability of $P(C_k|v_1\, v_2\, \ldots\, v_n)$ which is in fact the probability of classifying a tuple of attributes v_j into a class C_k..

$$\frac{P(v_1 \wedge v_2 \wedge \ldots \wedge v_n|C_k)P(C_k)}{P(v_1 \wedge v_2 \wedge \ldots \wedge v_n)} \tag{4}$$

Feature selection using decision trees is used to determine a subset of attributes, those in the upper levels of the tree, that are considered to be used with the Naïve Bayes classifier. In this approach, 10% of the samples in the dataset are randomly selected. Then a decision tree algorithm based in the C4.5 algorithm is used on the selected data and the attributes from the first 3 levels are selected. These steps are repeated from the beginning 5 times and from the union of all selected attributes a list of important attributes is built. Finally the Naive Bayesian

Classifier is used on the subset of data only using the attributes selected in the previous step. Results show that the improved Naive Bayesian Classifier obtained a better performance than the normal Naive Bayesian Classifier and the C4.5 decision tree algorithm consistently in classifying test using multiple datasets (42).

3.3. Algorithm Comparison

The algorithms presented cannot be directly compared and assessed. Due to the fact that the test environment was different for some of the presented algorithms, only the algorithms with similar test environments were compared. Neural networks have shown in many studies to perform better than the Discriminant Analysis and Logistic Regression model achieving a better classification rate and less misclassification of bas clients in the context of credit scoring (5). Some improvements in the neural network algorithm presented, namely by the addition of feature selection to the algorithm, have also shown that there is still room for optimization upon these structures (47). Other proposals from the literature also suggest the use of genetic algorithms has a possible optimization in neural networks using them in the neuron weight calculation (5).

Despite the good accuracy rate, neural networks are often criticized because they do not provide an explanation on how the decision mechanism produces its decision. Unlike other algorithms like decision tree where the decision path is easily obtained neural network behave more like a black box generating an output from some input (3). To overcome this limitation other algorithms were also proposed in the literature like the genetic programming algorithms and decision trees. The decision tree algorithm also shows promising results while providing a clear explanation for each given classification from the derived tree (42). The genetic programming algorithms presented show two different approaches to solve the same problem while maintaining good accuracy rates. The first generates a set of rules that help understand how attributes are used in the predictive analyses but when no rule covers the case presented adopts an alternative classification method where the decision process is not so clear. The second attempts to build a decision tree based on full atomic representations. Results provided in (46) and (45) show these two genetic programming attempts perform better than conventional decision tree algorithm C4.5.

Finally an optimization used upon the Naive Bayes classifier was also presented, using feature selection (42). This approach proved also to be slightly more efficient than decision trees.

The explanation for the decisions despite not being impossible to deduce are more complicated than the conventional decision trees and rule generation algorithms.

4. Multi-Agent Systems

Multi-agent Systems (MAS) emerged from the combination of Artificial Intelligence (AI) with distributed computational models, generating a new paradigm: Distributed Artificial Intelligence (DAI). This new paradigm, DAI, is a field of AI dedicated to the study of problem resolution through distributed computational systems.

The main objective of MAS is the study, construction and application of multi-agent systems. A MAS is made by several intelligent agents that interact with each other and with other systems perusing a set of objectives or executing a set of tasks or, even, both at the same time. Multi-agent systems aim to solve problems that are beyond the capacity of an individual agent (48).

4.1. Agents

Nowadays, the use of MAS to solve complex real world problems is already present. Agent collaboration to perform complex tasks is a concept used by many of these systems.

One of the most commonly accepted definitions for the term agent by the scientific community is from Wooldridge and Jennings (49), where they define an agent to be "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives".

Differences between agents may exist as some might be more reactive, responding quickly to some input, or more deliberative, using information from their sensors to build an internal representation of the surrounding environment in order to plan and act upon it or, even, have a hybrid approach joining reactivity with deliberation (48).

Wooldridge and Jennings (1995) define two different notions of agents, a weak notion of agent and a strong notion of agent, being the strong notion of agent an extension of the weak notion of agent. The weak notion of agent includes those exhibiting some or all of the following characteristics:

- Autonomy: ability to execute actions autonomously, without outside interference, in order to achieve it objectives;
- Social Ability: agents are able to interact with other agents, systems and possible humans to satisfy their objectives;
- Reactivity: allows an agent to perceive the environment, react to changes and respond in time to such changes according to his objectives;
- Pro-activeness: exhibition of a goal-directed behavior, taking initiative in order to satisfy its objectives.

These properties assure the agent can perform autonomous actions, communicate with other agents and its surroundings, react to events, and exhibit a goal-directed behavior taking actions after perceiving its environment.

On the other side, a strong agent, apart from the weak agent properties, may also exhibit human-like properties like:

- Mobility: ability to move around an electronic network;
- Rationality: takes into consideration that the agent will act and perform in order to reach its objectives and will not take actions that prevent him from achieving its results;
- Veracity: the assumption that an agent will not in conscience send false information to others;
- Benevolence: assumes that the agent will not have conflicting objectives and it will always try to do what it is asked.

A strong agent is then able to exhibit human-like behavior such as emotions, beliefs, intentions and obligations (50).

Despite the notion of weak and strong agents, an agent can also be divided into classes according to their internal architecture. This architecture is related to how the agent acts and uses reasoning to react in its environment. The available agent architectures are:

- Reactive: the agent senses the world, and based on is internal rules act rapidly in its environment. This agent does not possess an internal representation of the world and its actions are triggered by internal atomic rules of the type condition/reaction. ;
- Deliberative: this type of agent contains an internal representation of the world that is updated by the information collected by is sensors in the environment. Actions are

performed after a deliberative process where the agents consider what it knows of its environment which actions are better to reach its goals. This type of agent is considered slower to take action in comparison to the reactive agent ,its internal rules are more complex;

- Hybrid: hybrid architecture combines the best of both world between the reactive and deliberative architectures. It uses a reactive scheme to react to important changes in its environment in a faster time. The deliberative part is used for long term objectives with more careful reasoning's. This architecture is ideal to deal fast to certain environmental conditions handled by the reactive part while the deliberative part tries to find better alternative actions;
- Belief desire and intentions (BDI): this architecture has the goal of building an internal representation of the agent's knowledge based on its mental states which it will use to determine its course of action.

The use of agents should consider all the aspects detailed before to assure that its behavior is accurate and adequate to perform its objectives.

4.2. Characteristics of Multi-Agents Systems

A MAS uses agents and its interactions in order to achieve a global objective. Most times, MAS are used to perform complex tasks through the interaction of its agents so that their global objectives can be achieved. The main advantages of MAS are their flexibility, extensibility and reusability, it is easy to add and remove agents from the system. It is also considered a robust, reliable and computationally efficient system. Furthermore, its development and maintenance costs are low (51).

With respect to its architectures, a MAS may be open or closed. The former, an open architecture, does not consider a formal architecture and, for instance, its agents do not know how many of them there are in the system or their functions. Consequently, a discovery service is fundamental to obtain agent interaction. The latter, closed architecture, defines the number of agents present in the system, their interaction and relationships which are not likely to change over time. Additionally properties of a MAS may concern:

- Coordination;
- Communication;

- Organization.

In Figure 5 it is demonstrated the different types of coordination that may exist in a MAS. Firstly they can be divided into two classes, cooperation where agents help each other in order to achieve their objectives and completion where agents compete with each other. The cooperation side can be subdivided into a central cooperation where there are agents responsible for coordinative actions of other agents and a distributed cooperation where agents have different specialties used by other agents without formal coordination.

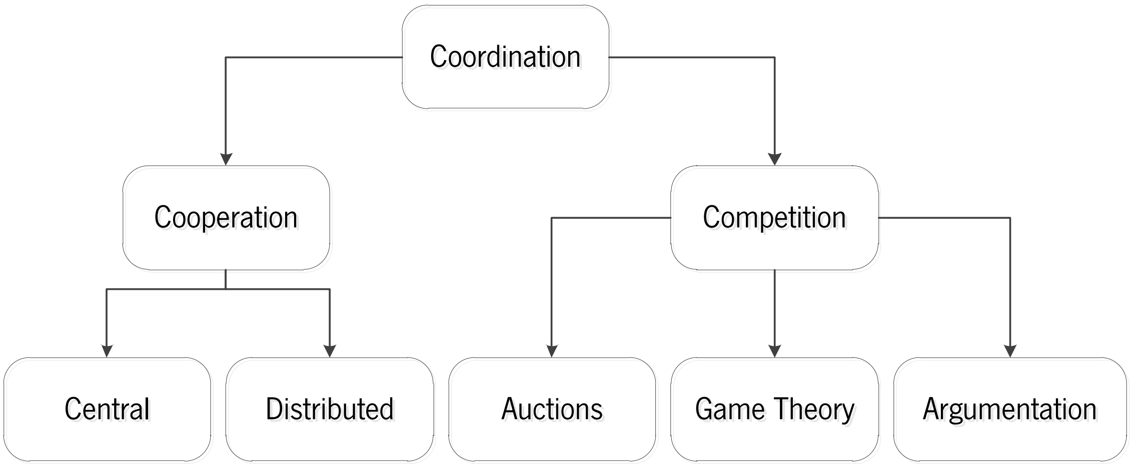

Figure 5 - Agent coordination types in a MAS

Another important characteristic of MAS is how communication between agents is made. In order to guarantee that two agents understand each other it is necessary to make sure that both agents are capable of reaching each other and share the same language protocol stack so what one agents says is understandable and has meaning to the other. Information can be passed through messages, as with the JADE framework (52), or stored in a global memory unit or blackboard system, like in SICStus PROLOG's Linda library (53). When communication between agents is made through messages, this communication can be direct or assisted. In a assisted communication, an agent must communicate with a facilitator agent which may them pass the message to other facilitator agent so the facilitator agent can deliver the message to the intended agent connected to that facilitator agent. Both agents, the destination and the sender might share the same facilitator agent, in which case, the facilitator agents simply acts as an intermediary between the other to agents.

Agents in a MAS may also be organized in different manners which differ from one system to another. These organization types can be divided into 5 five different groups which will now be presented.

The different types of organizations that can be used in a MAS are:

- Community of Specialists: all agents act as experts and communicate freely with each other in order to satisfy their objectives. Moreover, in this type of organization all agent have the same importance;
- Hierarchical: an agent hierarchy is defined in the MAS and communication is made with agents immediately above the hierarchy. In this scenario the most important agent are located at the top of the hierarchy;
- Based on markets: in this type of organization there agents that behave as contractor who have service they which to have done or to do some work and facilitators that receive offers and through auctions and contracts allocate tasks between agents according to their contract specifications;
- Federation: it is based in the hierarchical model but in each level of the hierarchy there several agents which use a facilitator agent to communicate to the upper level of the hierarchy;
- Task allocation: in this organization there are agents responsible to allocate tasks, the contractor agent, which it receives request and delegates the tasks in those requests to other agents whenever they are ready to perform them.

In order to build a MAS these definitions help answer the main problems that may arise when developing it. It is important to choose the right type of communication, organization and coordination of the system so it can reach its global objectives.

4.3. Standards

MAS are already a mature technology which has many standards defined for its creation and implementation. These standards were developed by the Foundation for Intelligent Physical Agents (FIPA), an Institute of Electrical and Electronics Engineers (IEEE) Computer Society standards organization that promotes agent-based technology and the interoperability of its standards with other technologies (54).

FIPA has defined a total of 25 standard specifications for MAS (54). In these standards it can be found specifications for components such as Agent Communication Language (ACL), message structures, ontology specification and transport protocols. These standards can be divided into different categories:

- Agent Communication;
- Agent Management;
- Agent Message Transport;
- Abstract Architecture;
- Applications.

Each of these categories, evidenced in Figure 6, is responsible for the correct functioning of a MAS, however FIPA declares that the agent communication is the core category at the heart of the FIPA multi-agent system.

.

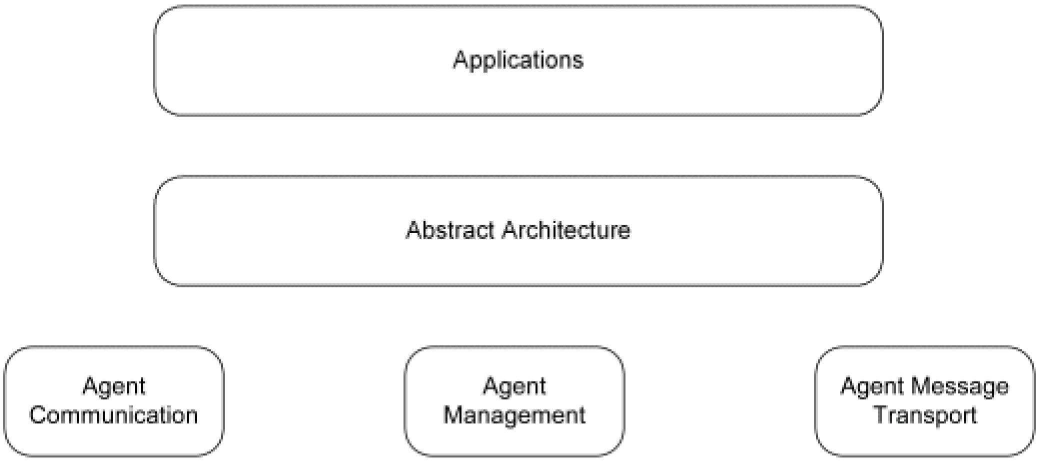

Figure 6 - FIPA Standard Categories

In the work developed and detailed in this dissertation, JADE is the multi-agent tool used (52), which is complaint with all the FIPA standards, one of the main reasons for its choice.

4.4. Multi-Agent Systems Tools and Frameworks

Today there are some mature frameworks and tools to build multi-agent applications. The following list presents a number of currently available tools and frameworks which are used to implement MAS:

- JADE, Java Agent Development Framework is a framework developed in JAVA which allows the creation of applications based in agents. Furthermore it is complaint with the FIPA standards implementing successfully the Agent Management System (AMS), the Directory Facilitator (DF) and Agent Communication Channel (ACC). It also uses the FIPA-Agent Communication Language (FIPA-ACL) in the messages exchanged between agents (52);

- JESS, Java Expert System Shell is a very adequate tool for the development of expert systems. Through a language similar to the LISP language, it is possible to provide a set of rules to the system that are activated when certain conditions are met. The use of JAVA and a interpreted functional language it allows the creation of easily adapted and portable intelligent systems (55);
- FIPA OS, this tool is an open source project for the development of agents in compliance with the FIPA standards. In this tool there are 4 mandatory components, Agent Shell to implement agents, Task Manager which fractions tasks into individual elements, Conversation Manager which allows a specific protocol to exchange messages and the Message Transport Service (MTS) (56);
- Zeus, is a set of tools for the development of agents. Each agent is composed by three layers, an agent definition layer responsible for the definition of the agent capabilities, an organizational layer where the relationships and interactions between agents are defined and the coordination layer where the negotiation and coordination techniques are implemented. Besides that, Zeus allows the definition of restrictions of tasks and ontology and it also provides agent discovery mechanisms (57);

The intention is not to provide a complete list of tools and platforms but rather a list of examples from which multi-agent DM systems may be developed.

4.5. Data Mining and Multi-Agent Systems

MAS and DM tasks have been developed in the past as two separate emerging areas. However, recently, there has been noticed a change in this regard. Research in agent mining systems presents itself as an opportunity to develop new approaches to solve challenges taking advantage of the agent interaction and learning algorithms from DM (58).

A recent study also point the fact that DM evolution is being made by moving to soft computing techniques, with the help of cloud computing and MAS, used in fields like services or research analysis fields, for example (59). Moreover, recent studies also consider the interest for DM solutions in the banking services as a mean to solve some of its classical problems such as credit scoring, risk management and customer relationship. (60). The bank industry has also been a case study before with approaches trying to make a multilayered multi-agent data mining architecture for the banking domain (61).

Other uses of a multi-agent system associated with data mining are related to their distributed computing properties. The collaborative agent work in a multi-agent system allows agents to share their internal information when needed, while performing their own work for the greater benefit of the system. In this context distributed data mining tasks can be successfully implemented in a multi-agent system (62). Distributed data mining systems are also used to tackle complex knowledge discovery problems, such as supply chain finance whereas there are heterogeneous data sources which have to be taken into account (63).

The use of a multi-agent system as the base for a distributed data mining system has been observed in some research papers over the years. The idea is to combine the intrinsic properties of multi-agent systems and their agent communication to develop distributed and collaborative DM tasks. One of the uses for such systems is the discovery of relevant pattern in a data warehouse through the use of intelligent data mining agents, acting in collaboration and performing collaborative action (64).

5. Development

This chapter of the dissertation documents the development process done from which improved classifiers and suggestion algorithms were built. In order to perform such work, some initial data was gathered from a German credit scoring dataset on which the proposed algorithms were tested and compared with other approaches from different authors.

The chapter ends with the specification of a MAS to perform both credit evaluation and suggestion in a distributed environment taking advantage of the characteristics of agents in MAS environments.

5.1. Case Study

The selected case study refers to a problem of client classification for loan applications. With the help of an initial dataset, one of the objectives is to improve classification models based on relevant algorithms to the problem at hand, and also to build a system that is able to dynamically update itself as new data becomes available in an autonomous way.

In order to achieve better results in client classification, some past data about a set of attributes characterizing each client of a financial institution is taken into consideration and is used to build the classification model. These data is considered to be a dataset of attributes in which each line characterizes one client and each column represents a client's attribute. Through the analysis of the dataset it is hoped that client pattern emerge and become useful to classify future client who fit any of those patterns.

Furthermore, the system must also be able suggest clients explanation on how they can improve their situation, increasing their chances to have loan applications accepted by the financial system. Regarding this aspect, the suggestion might be made with the knowledge extracted from the dataset such as client patterns related to good and bad client loans. Much of this information is already stored in the trained classifier algorithm, so it is possible to build algorithms to extract such information from the classifier even when not all client attributes are known, thus making suggestions on the unknown attributes (65). The goal of this suggestion system is to indicate a client what are the most advantageous characteristics he may possess to be granted with a loan application. With an incomplete set of attributes characterizing the client, the system will be able to find, with the help of its own classification models, the set of missing attributes the client must possess. This suggestion algorithm may also be used to promote new services or financial products to such clients.

The dataset credit scoring of a German institution was chosen (66) to perform this study. In this dataset, each client is characterized by a set of 20 attributes, followed by the classification of each customer's loan. The number of entries in the dataset is 1000 client records with no incomplete information. Although the number of entries seems low it is consistent with the available data that financial institutions use to build their models. Due to legal restrictions, the number of client records used to build credit scoring mechanisms is often not superior to 5000 records due to client privacy protection laws (18).

Table 3 - Credit Dataset Description

Number	Attribute	Description
1	Status	Status of checking account
2	Duration	Loan duration in months
3	Credit History	Client's credit history
4	Purpose	Purpose of the loan
5	Credit amount	Loan amount
6	Savings	Saving accounts or bonds
7	Employment duration	Duration of the present employment
8	Installment rate	Installment rate of disposable income
9	Personal status	Personal marital status and sex
10	Debtors	Other debtors / guarantors
11	Residence	Present residence since
12	Property	Property owned
13	Age	Age in years
14	Installment plans	Other installment plans
15	Housing	House status
16	Existing credits	Number of existing credits at the bank
17	Job	Job category
18	Liable people	Number of people liable to
19	Telephone	Existence of telephone number
20	Foreign worker	Native or foreign worker
21	Classification	Credit classification

The dataset is a combination of personal, social and financial information about bank clients. In fact, there are attributes present in the dataset to fill the 4 common categories of attributes for these type of scoring systems as evidenced in chapter 2. For instance, as demographic indicators there is attribute age, as a financial indicator there is the savings attribute, as the employment indicator there is the employment duration attribute and as the behavioral indicator there is the credit history indicator.. The complete list of attributes is shown in Table 3.

This dataset has two versions: one which contains categorical/symbolic attributes and another where these attributes were all transformed into numerical attributes. For simplicity and opportunity, the dataset chosen to develop the study presented in this paper was the one with numerical attributes.

The dataset presented in Table 3 has already been present in some studies and projects from different users. For instance, there are thesis which use this dataset to gather conclusions about the use of artificial neural networks in the credit risk assessment problem (5). Other uses for this dataset are the improvement of classifier algorithms proposing new variations of known algorithms which combine feature selection (47), (42). It was also used to assess original classifier algorithms based on genetic programming (46), (45) and on support vector machines (SVN) (44).

5.2. Improving Classification Algorithms

In order to improve classification algorithms, a feature selection step is introduced in some algorithms and then the results are compared to the initial algorithm and optimization algorithms from different authors. With this approach it is intended to prove the effectiveness of a good feature selection of the data provided to classification algorithms and the increase of the accuracy in such algorithms.

5.2.1 Feature Selection

The proposed feature selection algorithms use decision trees and their properties to select some of the most relevant attributes in a given dataset.

The assumption that serves as the base for this algorithm is that decision trees consider the best set of attributes that classify the sample data for the upper branches in a decision tree. With this information in mind two different feature selection algorithms are proposed. In order to implement them in a testing environment the J48 classifier from the Weka Toolkit (35) was used with a confidence factor of 0.25 to produce a decision tree which was then used to perform the feature selection.

The first algorithm chooses all the attributes presented in such decision tree as important and delivers the set. The presented algorithm is described in Algorithm 1 where the necessary are described and commented. Sometimes, not all attributes from a dataset may be presented in a decision tree which might lead to the suspicion that they are not as relevant for the problem of credit scoring as the other attributes. Consequently, with this algorithm those attributes which are not present in the decision tree are considered as less important in the process of classifying instances and it is hoped that with their removal future classifying algorithms will perform better as they will not be confused by the presence of such algorithms.

```
reduce_datset( data ) {
   /* build initial decision tree to perform selection */
   tree = build_decision_tree( data ) ;
   /* Remove attributes that are not present in the
   first three layers of the decision tree */
   for ( attribute a in data ) {
      if ( tree.has_attribute ( a ) == False)
      then
         remove_attribute( data , a ) ;
   }
   return data ;
}
```

Algorithm 1 - Feature Selection by reducing the dataset

The second algorithm, presented in Algorithm 2, aims to get a reduced list of the most relevant set of attributes for decision making in a dataset. With this information it is hoped that the better set of attributes that classifies most influences a classifiers decision is placed on upper levels of a decision tree. This knowledge can then be used to discriminate the selected attributes in the dataset before presenting the dataset to another classifying algorithm. Discrimination can be obtained by removing attributes from the dataset or normalizing attributes values with different interval ranges. In this case, the attributes presented in the first three levels of a decision tree are selected and returned as the most important algorithms. The decision on how to discriminate the attributes is delayed to the stage where the feature selection is integrated with the classifier algorithms.

Concluding, two feature selection algorithms were presented which aim to identify a smaller set of attributes in a dataset. Future steps include the use of this knowledge to influence the data in the dataset without changing its meaning so the new classifying algorithms can devote more attention to those attributes and less attention to the less importance attributes.

```
attribute_discrimination( data ) {
   /* build initial decision tree to perform selection */
   tree = build_decision_tree( data );
   /* Discriminate attributes that are present in the
   first three layers of the decision tree by giving them
   a larger interval range in the data normalization
   process */
   for ( attribute a in data ) {
      if ( tree.attribute_in_first_three_layers( a ) == False)
      then
         remove_attribute( data , a ) ;
    }
    return data;
 }
```

Algorithm 2 - Feature selection by attribute discrimination

5.2.2 **Algorithm Description**

With respect to the previous work on feature selection, some approaches are now considered to implement feature selection upon classifier algorithms. As an example, the Multilayer Perceptron and Linear Regression algorithms will be used due to the fact that they implement widely known algorithm used and studied in the credit scoring problem.

Neural network algorithms are being studied as a possible replacement for the common linear regression methods used in the majority of financial institutions today. The internal structure of each Multilayer Perceptron applied in this section has the same logic as stated in section 3.1.1 when the Multilayer Perceptron is introduced. Figure 7 illustrates a Multilayer Perceptron, where the internal layers and the number of neurons used in each layer are represented. Likewise, regression algorithms are still considered the most used algorithms by financial institutions today and that is the main justification to also use the Linear Regression algorithm in the implementations of the proposed algorithms.

In order to make the algorithm more general, the description below will make reference to classifier algorithms, although the results presented will only consider the implementation of the described algorithm with Multilayer Perceptron or Linear Regression algorithms as classifiers in the described algorithms.

As stated before, the use of feature selection is considered as a way to make the classifier algorithm more aware of relevant attributes to whom special consideration should be given.

In order to explore different alternatives to join feature selection with the classifier algorithms, two algorithms were developed combining both. The main goal of this approach is to prove that feature selection can, in fact, improve classifier algorithms and its use brings better results than when it not considered. More alternatives and different combinations could be developed and tested but due to objective containments these two approaches are the only considered.

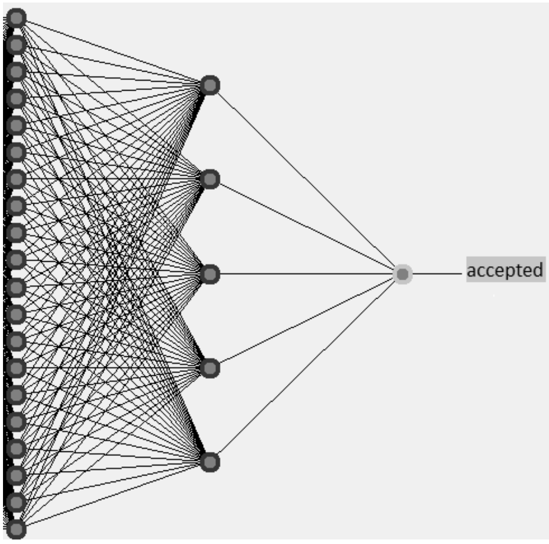

Figure 7 - Multilayer Perceptron Internal Representation

The first approach uses the selection algorithm described in section 5.2. where the attributes are selected by the feature selection algorithm if they take part in a decision tree after it was trained with the chosen dataset. The combination of this feature selection methods and the classifier algorithm is detailed in Algorithm 3. This approach uses a filtered dataset and feeds it to the chosen classifier algorithm. The feature selection is used to remove from the dataset the attributes not featuring in the resulting set of attributes from feature selection algorithm. With the new dataset we present it to the classifier algorithm and train the classifier with the reduced dataset. This algorithm aims to test that if by reducing the number of attributes in a dataset according to their importance, given by the feature selection, better evaluations are possible. Also it aims to tackle the hypothesis in which the presence of some attributes can in a sense confuse the classifier algorithm reducing its accuracy. It should also be stressed that due to the fact that these structures might be sensible to attribute interval ranges all the attributes in this reduced dataset are normalized between 0 and 1. As a note for future reference the algorithm was named "classifier with feature selection 1".

```
classifier_with_feature_selection_one( data ) {
   /* perform feature selection and build new dataset */
   data2 = reduce_dataset( data ) ;
   /* performs attribute interval normalization on the dataset
   between values 0 and 1 */
   normalize_attributes( data2 , 0 , 1 )
   /* buildand train the classifier with the new dataset */
   classifier = new Classifier( data2 ) ;
   /* return the trained classifier */
     return classifier;
}
```

Algorithm 3 - Classifier with feature selection 1

The second algorithm developed will use the second feature selection algorithm presented in section 5.2.   where the attributes are selected if they are present in the first three levels of a decision tree trained with the dataset. The resulting algorithm from the combination of this feature selection algorithm and the classifier algorithm is presented in Algorithm 4.

```
classifier_with_feature_selection_two( data ) {
   /* perform feature selection and build new dataset */
   data2 = discriminate_attributes( data ) ;
   for ( attribute a in data ) {
      if ( data2.has_attribute( a ) )
      then
         /* normalize attribute a values in the dataset data
        with values between 0 and 4 */
        normalize_attribute ( data , a, 0 , 4 );
      else
         /* normalize attribute a values in the dataset data
         with values between 0 and 4 */
         normalize_attribute( data , a , 0 , 2 );
   }
   /* build and train the classifer with the new dataset */
   classifier = new Classifier( data ) ;
  /* return the trained classifier */
 return classifier
}
```

Algorithm 4 - Classifier with feature selection 2

The developed algorithm uses the given attributes from the feature selection as a mean to separate important attributes from less important attributes, however, unlike before the less important attributes will not be removed from the dataset, instead, a special attribute normalization will be performed on the dataset. The set of attributes indicated from the feature selection algorithm are normalized within a range from 0 to 4 and all the remaining attributes are normalized within a range from 0 to 1. As stated before, classifier algorithm might be sensible to the input range. In order to take advantage of this property normalizing the dataset with different interval ranges on selected attributed will try to direct the classifier algorithm to take more attention to those attributes with greater interval range. The remaining attributes are still present in the dataset to prevent a possible situation where their elimination reduces the accuracy of the classifier algorithm, though it is intended to reduce their importance. Once again, as a note for future reference, the presented algorithm was named "Classifier with feature selection 2".

Over chapter 6. the results from both approaches will be presented and conclusions discussed. More specifically, it will be demonstrated that Algorithm 4 achieves a better accuracy rate than Algorithm 3. Moreover, it will also be demonstrated that Algorithm 4 yields good results even when compared to other alternatives and other optimized algorithms from different authors.

5.3. Suggestion System

An objective of the work presented in this dissertation was to develop a suggestion system for the credit scoring for loan application problem. Over the next section, it is presented the context where such systems are useful for both financial institutions and their clients. Following the context description the developed algorithm is detailed explaining its properties.

5.3.1 Context

After deciding upon a classification algorithm, it was felt that client classification might be improved if a suggestion mechanism exists in the system. For instance, before applying for a loan, a client normally uses a simulator or a conversation with a consulter from a financial institution to discuss how to purpose and guarantee itself as a good borrower. During this process the client may only be interested on evaluating his chances of being granted a loan from that financial institution and from these scenarios which is the best for him. In this context a suggestion model could be useful to the client and may also help the financial institution advice his client on the better set of actions he or she can take to improving his chances of being granted with the desired loan.

Other usefulness of a suggestion system could be explained a client to whom a loan application was refused using the presented classification model. With a suggestive algorithm he or she may find a solution for his problem. The client would give the system an incomplete set of information of a predetermined set of attributes he cannot change and the system would calculate how changes in the not specified attributes would increase his chances to be granted with the loan. These changes might consider increasing the client's savings amount available in the financial institution in a different account, reducing the amount of the loan by a percentage or, even change the installment plans. With this new information the client could be advised about these new potential situations that allow him to be granted with the desired loan. From the client's perspective it would allow him to become aware on how to improve its credit scoring card or to understand its loan limitations. On the other hand, from a financial institutions perspective the suggestion system could be used both as a client suggestion system easing the process of finding answers for the client's needs and as a cross selling platform where the suggestion system proposes financial products from a financial institution that could improve a client's credit score (67).

5.3.2 **Genetic Programming**

The purposed suggestion system is based on the classical genetic programming algorithm, represented in Figure 8, which is an evolutionary algorithm (68). In short genetic algorithms (GA) begin with an initial population randomly created.

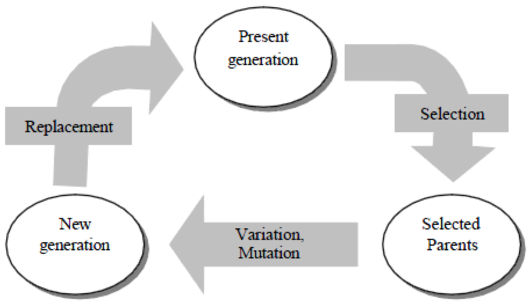

Figure 8 - Genetic algorithm evolution cycle (68)

Each individual in this population is called a chromosome and each chromosome is composed of one or more genes. A gene is a representation of a variable inside a chromosome. After the creation of the initial population this population suffers four operations performed by genetic operators:

- Selection;
- Variation;
- Mutation;
- Replacement.

The selection operator selects individuals from the population to reproduce child individuals which are a combination of the parent's genes. This selection is probabilistic but it also discriminates individuals with better fitness to have higher probabilities of being selected. The concept of individual fitness will be explained.

The combination between two parent individuals which produce two child individuals in a population is performed by the variation operator. This operator simulates the reproduction of new individuals in a population from the combination of the parents' genes. The most used technique to combine the two parents' genes is the crossover technique. In this technique a cut point in the chromosome is created separating two sets of genes from each individual. This cut point is the same for the two individuals in which the crossover will occur. Over the next step the left set of genes from an individual is joined with a right set of genes from other individual thus creating a new complete chromosome.

Mutation is the operator responsible to alter some genes in a chromosome in order to differentiate the chromosome fitness and introduce variance in a genes value. Until now, a gene never altered its value rather it was be changed by another as it was seen with the variance operator. With the mutation operator, a selected individual in the population sees some of its genes altered randomly by this operator.

The last operator is the replacement operator which is responsible to substitute the old generation of chromosomes by a new generation of chromosomes. To replace the old generation, the best individuals between the individuals at the beginning of a cycle and the individuals generated by the genetic operators are chosen until the maximum number of individual per generation is reached. It is important to note that the number of individuals at the beginning of each cycle between generations is not altered. The evaluation of the best individuals is made by a fitness function which is responsible to use the problem variables represented by a chromosome gene and calculate a fitness number.

The complete GA cycle starts by an initial population containing a fixed number of individuals randomly generated. Then, each individual is evaluated by a fitness function ordering the individuals according to their fitness values. With the new calculated fitness values the selection operator selects some individuals so the variance operator can generate new individuals by the combination of genes between individuals. On some selected individuals the mutation operator is then used to induce variance on the gene's values. With the resulting set of individuals from the beginning of the cycle and the generated individuals a fitness function is again used so the replacement operator can chose the best individuals which are used as the initial population at the beginning of the next cycle. This GA cycle is repeated until the results are proven satisfactory or until the number of cycle iterations reaches its programmed limit.

5.3.3 **Suggestion Algorithm**

The suggestive algorithm is inspired in GA because they are considered to be a generic optimization algorithm and easily adaptable to different problems and objectives. In order to a GA to search through a set of fixed and variables attributes with different machine learning (ML) classifiers some changes to the generic GA cycle have been done.

The idea is to use genetic algorithms to perform a search in the global space of possible solutions and deliver the positive answers to the client, in the case study example, when the loan approval is possible. As the generic GA cycle does not consider the existence of immutable genes, the known attributes from a client, those which he or she cannot change or is not allowed by business rules are separated from the unknown client attributes. Those unknown attributes are then represented as genes in a chromosome. One of the problems to be solved now is the evaluation of each chromosome by a fitness function capable of translating if the attribute values represented by the chromosome's genes do in fact grant a loan for that client. The solution for this problem was to create a fitness function which could be represented by the classifying algorithm which decides whether or not a client receives a loan. To accomplish this, each time an individual needs to be evaluated by a fitness function, the attributes represented in each gene are joined with the client's known attributes preserving the original order and then fed to classifier algorithm responsible for the loan approval decision. The result of the evaluation becomes the fitness of that individual in the GA cycle. The fitness function is described in equation 5

$$Fitness = evaluation(\ classifier, chromosome_{genes}, fixed_{attributes}) \quad (5)$$

With a chromosome and gene representation for the problem and a valid fitness function it is possible to successfully build a custom GA able to perform the search for solution for each client with known and unknown attributes. The simple algorithm used to search such responses is a set of steps explained below:

1. Select each missing client attribute as a gene in a chromosome;
2. If not created, randomly create the initial population of chromosomes; otherwise, select the best clients from the set generated earlier;
3. Perform the selection operator and select pairs of chromosomes;

4. In selected pairs of chromosomes perform the crossover operator by calculating a split point to exchange genes between each pair of chromosomes;
5. Perform the mutation operator and assign a random value to one gene in selected chromosomes;
6. Join the gene information with the known immutable client attributes and use the Multilayer Perception with Feature Selection 2 as the objective function;
7. If the maximum time of calculation not exceeded, if there are still negative client classifications or if the number of desired alternatives is not met start from the beginning; otherwise, the algorithm ends here.

In the credit data system, each individual in the population will be the set of attributes that were not specified by a client. Those attributes are then generated randomly between the space of possible solutions for each attribute type. Figure 9 illustrates this initial step where the attributes A(X) represent fixed attributes given by the client and V(X,Y) represent the attributes randomly discovered so the modified genetic algorithm can be used to perform suggestions.

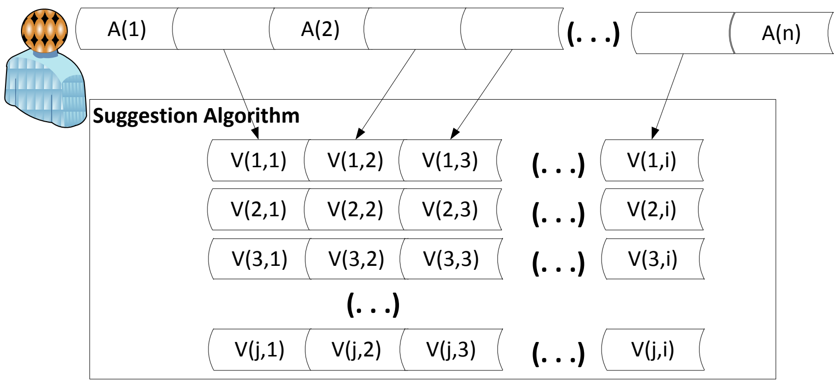

Figure 9 - Suggestion System initial step

After performing the selection and mutation operators, the attributes are joined with the immutable client attributes and a classification of each pseudo-client is done, retaining the raw classification value as the client score to select the chromosome population for the next iteration and chose the best classified clients from the possible set. When the algorithm reaches the end of a stage, the population selected for the next iteration is the set of chromosomes that achieved better classification from the previous generation or the present modified generation that have a different combination of attributes. This last step assures that the answers to the initial problem are all different. Figure 10 shows how the discovered situations are presented to the client by the algorithm. Each of the presented solutions represents a positive answer to loan request.

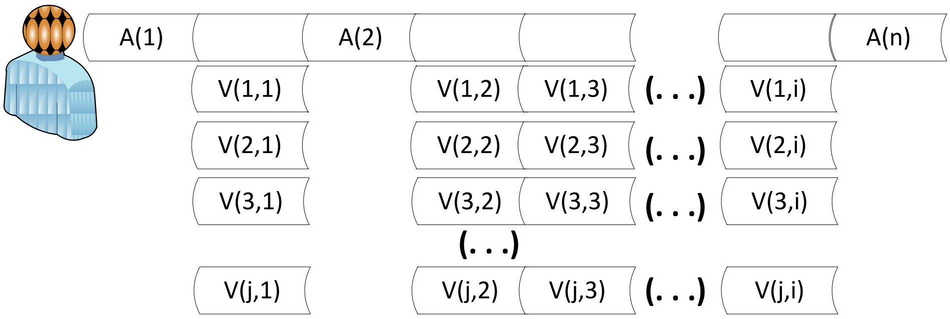

Figure 10 - Suggestive System with discovered suggestions

The pseudo-code process presented in Algorithm 5 details the implementation of the algorithm described.

```
suggestion_algorithm( known_atributes, pop, mut_rate, cros_rate, itera, ans ) {
   /* initial variables */
   INITIAL_POPULATION = pop ;
   MUTATION_RATE = mut_rate ;
   CROSSOVER_RATE = cros_rate ;
   MAX_ITERATIONS = itera ;
   DESIRED_ANSWERS = ans;
   /* build first chromosome with unknown attributes */
   for ( attribute a in data ) {
       if ( known_attributes.has_attribute( a ) == False )
       then
           cromossome.add_gene( a ) ;
   }
   /*build initial population */
   population = chromosome.clone( INITIAL_POPULATION ) ;
   population.randomize( ) ;
   evaluate_chromosomes( population, known_attributes, classifier) ;
   count = 1 ;
   /* intiate GA cycle to search for compatible answers */
   do {
       new_population = perform_crossover_operator( population, CROSSOVER_RATE) ;
       perform_mutation_operator(new_population, MUTATION_RATE) ;
       /* join known attributes with the attributes discovered in each chromosome
       and evaluate each complete simulation with the classifier algorithm saving
       the classifier's evaluation */
       evaluate_chromosomes( new_population, known_attributes, classifier) ;
       /* select the best chromosomes from both the old and new generation of chromosomes
       until the maximiun population number is reached */
       population = select_next_population( population , new_population );
       count += 1 ;
   /* repeat process while the number of iterations or the number of positive answers
   is not reached */
   } while ( MAX_ITERATIONS != count or DESIRED_ANSWERS = population.positive_evaluations )
   return population ;
 }
```

Algorithm 5 - Client suggestion

The design of Algorithm 5 allows the use of any type of known and unknown attribute information as each gene and chromosome are dynamically created at the beginning of each cycle. This means that the algorithm does not make initial assumptions about the chromosome size which is the number of genes it contains. This property allows the algorithm to easily perform searches from situations where all the attributes are unknown, to the more specialized contexts where only certain attributes are unknown.

This algorithm is also compatible with different classifying algorithms as it only needs that the classifier returns a positive or negative answer. The algorithm implementation uses negative values to represent a loan not approved and positive values to indicate an approved loan. The internal structure of the classifier is not relevant to this algorithm. That makes it possible to use algorithms which behave like a black box such as the neural networks to act a classifier algorithm to this problem. Due to the fact that this work is evaluating both new and old approaches to the credit scoring problem, in terms of classifier algorithm, it was important to assure that the classifier used was not a constraint in this algorithm. This algorithm has already been presented to the research community in (67) in a conference devoted to the Soft Computing scientific area.

The functionality and result from this algorithm are presented in Chapter 5. along with the main conclusions gathered by the use of this algorithm in different objectives. More specifically, it will be proved the utility of the proposed algorithm to suggest alternatives to clients and allow consulters from financial institutions to better advise their clients or perform cross selling operations recommending financial products which improve a client's credit score.

5.4. Multi-Agent System for Credit Scoring

In order to fully integrate the studied classification and suggestion algorithms in a flexible yet robust system a multi-agent system was designed. The aim of this system is to behave as an expert system, in which it is intended to have agents performing its tasks through information and task sharing, so the final system objectives are accomplished. In this section, the main components of the developed system are described from general system behavior to individual agent tasks and responsibilities.

5.4.1 System Behavior

The proposed MAS present in this dissertation is composed of different types of agents with specialized characteristics which, when working together, are able to assess clients and suggest alternatives in a credit scoring system (69). The number of agents in the proposed MAS is not fixed, rather, different types of agents with specific functions and responsibilities are defined allowing to have redundancy on the MAS's agents and also different implementations on some types of agents. In order to assure the correct functionality of the system, six agents where defined:

- Feeder: responsible to gather the initial information about past clients and their respective;
- Model: which builds and trains the decision classifier used on the process of credit scoring and suggestion activities;
- Decision: responsible to evaluate loan applications based on client attributes;
- Suggestion, implements the suggestion algorithm detailed in section 6.4 to perform suggestions;
- Configuration: configures restrictions on client attributes according to business rules or legal implications;
- Inquiry: agent responsible for the interaction with clients gathering their requests and present the final answer to them.

The interaction between agents and system users and the general MAS organization is represented in Figure 11. This representation tries to combine the classical three layer architecture, used in most software engineering process to separate data, business logic and user interface, and a MAS context. From the analysis of Figure 11 it is easy to understand that the Feeder agent is responsible for the data gathering. Likewise, Model, Decision, Suggestion and Configuration Agents are responsible to implement the business logic of the system which means building and training all the algorithms to perform the suggestion credit scoring activities as well as enforcing all the necessary configuration rules on the system. The remaining Inquiry agent type is responsible for the presentation layer in the sense that it has the responsibility to interact with end users gathering their request and presenting the solution to such requests.

An information flow can be also be observed, in Figure 11, from the moment data is imported into the system to the moment where knowledge is shared to the Inquiry agent. The Feeder agent is used to import a data into the system and is also responsible for monitoring the data source for new data. A Model agent, upon receiving the new data, updates its decision algorithms according to its internal classifier algorithm, which can be unique to each agent. These trained models are then shared with Decision and Suggestion agents in order to keep them informed about the latest and more accurate models. Both Suggestion and Decision agents use their classifying models to classify and suggest instances. These actions are performed upon request by the Inquiry agents present in the multi-agent system, which are the point of contact between the system and its end-users. A configuration agent exists to overview the multi-agent system, enforce some logical rules on the system, such as the maximum number of agents of each type, and enforce business policies, such as forcing classifiers to deny loans to specific instances with certain values even when the classifier considers it a good client.

All communication between agents is supported by an Agent Management System (AMS) and Directory Facilitator (DF) internal to the multi-agent platform chosen, developed in JADE (52). It is requested that, upon creation and deletion, all agents in the system are registered or unregistered in the AMS service so they can be found by other agents. The AMS and DF are not represented in Figure 11because they are already part of the JADE framework and a part of most MAS platforms.

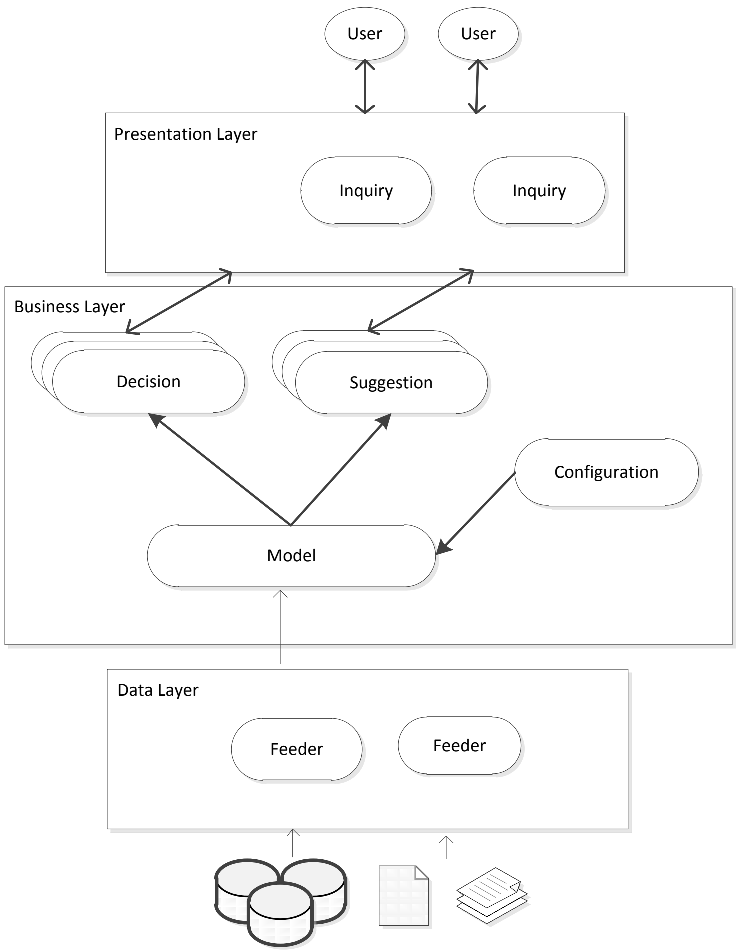

Figure 11 - Multi-Agent System for Credit Scoring

As stated before, with these six different agent types, the system is built and acts in order to achieve its global objectives:

- Classify client's loan applications based on client information stored by financial institutions;
- Suggest possible scenarios based on incomplete set of client information which can then be used for tasks like client suggestion, client advice, client profiling studies, decision explanation or even cross-selling studies by financial institutions.

In this context, the suggestion capability provides a complementary feature to the credit scoring problem, which by analyzing an incomplete set of client information and looking alternatives in the remaining unknown client information shows situations that would grant those clients the loan applications. Another utility for the suggestion system is that it may also give hints on how the scoring system works when it is not easy to deduce that information from the classifier used to score clients.

5.4.2 **Agent Description**

After the initial system explanation, in this section each agent will be described with its main interaction functionalities and alternative implementations. The different agents present on the system may appear cloned or with different implementations preserving their functionality. The case for cloned agent types is related to redundancy, system availability, parallel task handling and the system support failure of individual agents without compromising the entire system. On the other hand, different implementations for agent types concern the combination of alternative algorithms and situations for each agent. The only function which must be preserved in these cases is the agent interface for each type, thus leaving the internal implementation to preference of the user. As an example of this utility, two credit scoring systems on two different financial institutions could be considered where those two MAS have different classifiers for the credit scoring problem while preserving the MAS.

The description of each agent type is presented in what follows, explaining their main functionality and interactions as well as the agent interfaces developed for each agent, allowing different implementations without losing their system abilities.

Feeder Agent

In order to build a data mining multi-agent system, there is a need to find and collect data from which classifiers can be built and trained. As most ML and AI techniques require an initial set of data to train the algorithms the Feeder agent becomes responsible to locate and provide this information to the system. This agent acts as a sensor type agent, responsible to monitor the data sources configured in its environment retrieving new data to the system when it becomes available. The interactions of this agent type with other agent from the developed MAS are presented in Figure 12. In this figure it is seen that after gathering the data from a data source the feeder agent must pass it to a Model agent, which is responsible to use it to create or update his models. It must also store the retrieved information on a relational database so it acts a data backup of the information already retrieved.

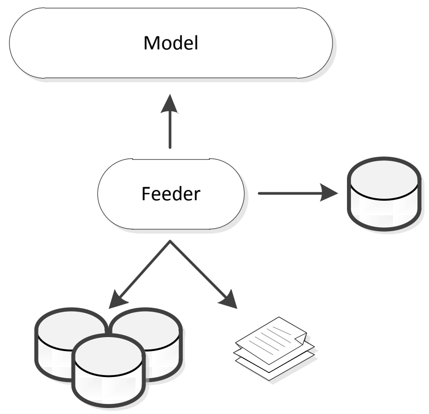

Figure 12 - Feeder agent interaction

At any given time, there could be multiple Feeder agents on the system, each one monitoring a data source, which can be a data warehouse cube, a logical table or, even, simple raw data files. There are, however, a couple of restrictions on the type of data sources used as they must concern the same client attributes and implement a feeder interface with a standard set of method signatures to assure the correct data format is passed to the model agents present in the MAS.

Model Agent

The Model agent is responsible for building classifier models which can use classifier algorithms from ML, AI and DM techniques. Its aim is to learn from past experiences how to assess each situation client situation. The agent itself is only responsible to train a classifier. The classifier algorithm used should be updatable in order to decrease the training time, however if that is not the case, the data stored in the backup database might be used to retrain the classifier algorithm with new data preserving the knowledge present in past data.

Figure 13 details the interaction of the Model agent with other agent types in the system. The model agent is responsible to pass to the Decision and Suggestion agents the trained classifier along with the filter information on each attributes, configuration rules and the model version number. The configuration rules are obtained from the Configuration agent, which in turn is responsible to pass that information to the Model agent. Anytime new data becomes available or new configuration rules are defined the Model agent will pass its information to the Suggestive and Decision agents.

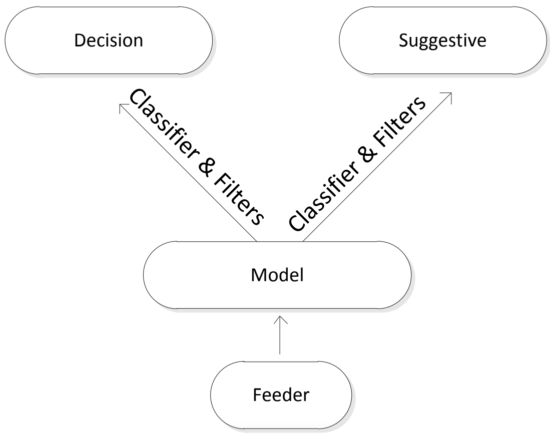

Figure 13 Interactions with the Model agent

Different implementations of the Model agent are possible in order to allow the use of more than one possible classifier algorithm. With respect to this, when a new classifier algorithm is required the work to adapt the system to that change is to develop a new model agent with the given classifier substituting the present Model agents on the MAS. In order to build its internal classifiers a DM and ML algorithms library from Weka (35), was used. Furthermore, some optimization upon classifiers is possible, building hybrid algorithms connected to the credit scoring problem (40), which can also be used as classifier models in this agent.

It is also possible to allow the use of more than one model agent with the same behavior to introduce redundancy in the system. This will prevent the failure of the classifier training in the system, if a Model agent would crash there could be others to pick up the work.

The main advantage of this Model agent is to be able to make isolated training of the decision system components, meaning that other agents, dependent on this knowledge representation, can continue to operate successfully while the system itself is being updated.

Decision Agent

Whenever a new decision model becomes available, the system has to update its Decision agent to reflect the change on the MAS. Figure 14 shows the information flow responsible for the update of the Decision agent and how it interacts with other agents present in the developed system. This agent is responsible to store the trained classifier passed to him by a model agent, and will use this knowledge to answer evaluation requests made by the Inquiry agent.

After receiving the first model, trained by a Model agent, the Decision agent is able to respond to requests from other agents requesting an evaluation of an individual with the complete list of attributes filled. Each response to requests carries not only the predicted value, but also the model version in which it was obtained. Consequently future agents might be able to justify the reason why evaluation may differ over time.

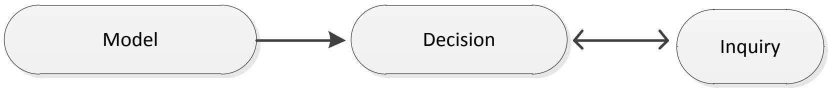

Figure 14 - Interactions with the Decision agent

Due to service level issues, when the agent becomes stressed by the number of requests received, it has the option to launch a clone agent in the system, in order to maintain service levels, distributing incoming requests by other Decision agents.

Suggestion Agent

This agent behavior is based on the suggestion algorithm detailed on section 5.3.   . In order to make suggestions, the Suggestion agent uses the available knowledge inside classifier algorithms to look for more advantageous solutions to incomplete evaluation requests. In the proposed MAS, the Suggestion agent receives the model created by a Model agent and uses it to perform searches in the classifier algorithm, in order to obtain the most advantageous situations for the request presented. The searches conducted by the Suggestive agent are ordered by a Inquiry agent to whom the answers found are then transmitted. These interaction between agents are detailed on Figure 15.

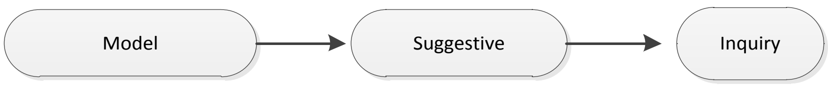

Figure 15 - Interactions with the Suggestive agent

As the classifier algorithm might not be simple to understand, a genetic algorithm was developed, which is able to use an incomplete set of attributes and search in the available domain for the values of the missing attributes, granting successful evaluations in the decision model. The suggestion are then presented as a ranked list of possibilities for the missing attributes ordered by the final credit score of each situation.

Due to service level issues, as it happens with the Decision agent, when it becomes stressed by the number of requests received, the agent has the option to launch a clone agent incoming requests with another Suggestion agent.

Configuration Agent

The Configuration agent is responsible for the multi-agent system supervision. This agent enforces logical rules and business policies, such as business rules to grant or deny loans directly, without the need to go through the evaluation of a classifier trained by a Model agent. When applied, these policies could, for example, deny loan based on attributes values like unemployed people or people with bad loan history.

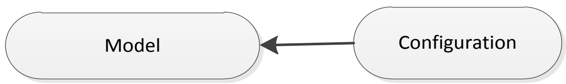

Figure 16 - Configuration agent interaction

These configurations are determined by process owner for the credit scoring problem and introduced in the system manually by the Configuration agent as shown by Figure 16. These restrictions are then passed to the Model agent so it can incorporate these restrictions on the trained classifier creating situation where the restriction substitutes the value of the classifier assessment. Each time a new set of configurations is passed to a Model agent, they replace any existing restrictions and force the Model agent to disseminate the changes made through the agents which it has interaction with, namely the Decision and Suggestive agents while increasing the model revision counter.

Inquiry Agent

The Inquiry agent, which interacts with the final end-user, receives classifying and suggestive requests and passes them to the available Suggestion and Decision agents. The human-agent interaction is possible through a GUI, which reacts to the users' requests, delegating the actions to other agents in the system. The decision for the approval of loan applications is made when the full list of attributes is available and suggestion is performed when some values are missing.

The Inquiry agents in the MAS interact directly with Decision and Suggestive agents which hold the classifiers and suggestion algorithm respectively to deal with its requests. After, performing the request on a specialized agent, the results are then retrieved to the Inquiry agent which in turn displays it on the agent's GUI to the user. These interactions are present in Figure 17, where the interaction between these agents is illustrated.

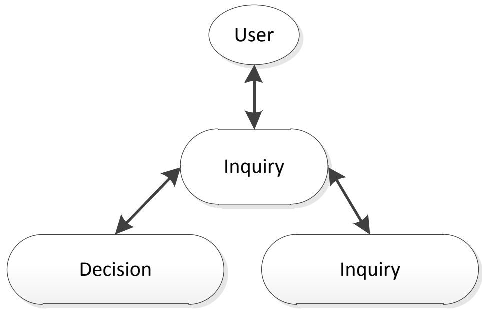

Figure 17 - Inquiry agent interaction

Other implementations for the Inquiry agent might not use a GUI to receive the users' requests, rather they can be taken in a form of a classical web service by the agent. This would enable web based apps to deliver the requests to the system and receiving the final result as the result of the calling the web service.

5.4.3 **Agent Messages**

In order to pass information between agents five different messages were designed to be exchanged between agents. These messages are built upon the Agent Communication Language (ACL) message standard defined by FIPA and implemented in the JADE mulit-agent framework. Each message defines a performative action, a conversation ID, a receiver and some content. The content used in this system were serializable instances of defined Java class. For the correction of the system 5 f these classes were defined to be encapsulated inside the standard FIPA-ACL message. The classes developed in Java were:

- InstancesMessage;
- ModelMessage;
- QueryMessage;
- SuggestionMessage;
- ConfigurationMessage.

Whenever a message is sent from one agent for another it is sent with a performative REQUEST, it represents that that agent is asking another to perform some task for him. The answer from the other agent uses the prerogative INFORME. This Request/Inform action-response is used by the inquiry, decision and suggestion agents. Whenever the inquiry agent has some request it request an answer from a decision agent if a client classification is needed or it request an answer from the suggestive agent if a suggestion is needed. Both these agents, decision and suggestive respond with an informe performative to the inquiry agent. Speaking of content of the shared messages in the case of a query by an inquiry agent it uses a query message to pass the instance. If the message is used for classification then the query message must have a complete instance of data defined which will be read by a decision agent who will then return the query message with the missing fields model version and response filled. On the other hand when a suggestion is needed the query message uses a query message that implement an incomplete instance which will be used as the known attributes. The suggestion agent will then return a suggestion message with the resulting population of suggestion and the model version number.

The notification from a model agent to the suggestive and decision agents is made by a message with the performative INFORME and the content encapsulated in the ModelMessage class. This message contains the classifier trained, the normalizeDataset to perform necessary filters on the presented data, the model version and restriction on attributes used.

The configuration agent uses the same strategy to inform the model agent of attribute restrictions. It also uses a INFORME performative along with a Configuration class to pass the restriction values.

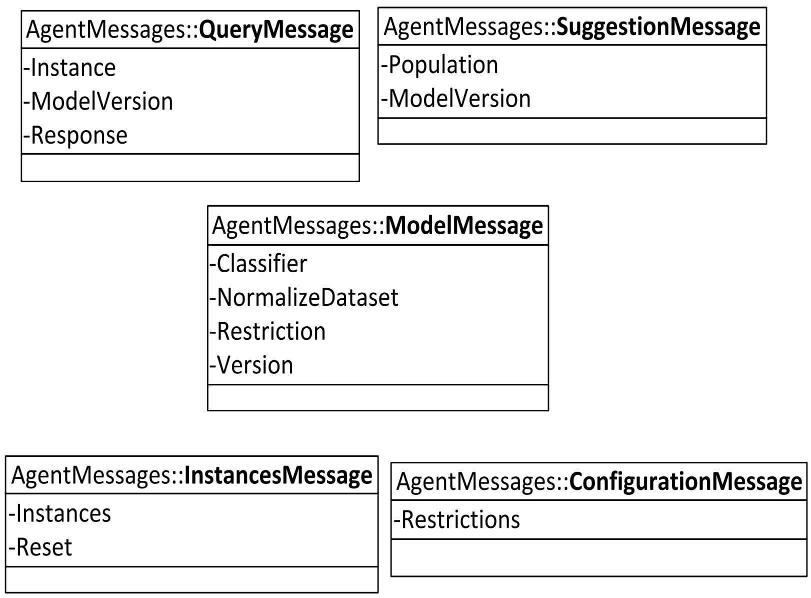

Figure 18 - Agent messages' content

As the last type of message interaction used, the InstancesMessage is used by the feeder agent to inform the model agent about the existence of new instances to train classifier models.

For this purpose, the instances are passed to into a InstancesMessage which is then encapsulated inside a ACL message with a inform performative.

All the messages used and their attributes are illustrated in Figure 18 where the different class diagrams for each message content is displayed. This set of messages assures the information flow through agents as well as agent interaction to perform collaborative tasks such as decisions and suggestions.

5.4.4 **MAS Standards Used**

The agents and interaction described on the section 5.4. are implemented with the help of a set of standards already developed. The MAS framework used was the JADE framework (52) as already stated before. This framework implements the MAS standards developed by the FIPA association (54) which were also used to perform some tasks on the developed MAS.

In order to allow the discovery of agents in the system some potentialities of the chosen framework were used, namely the AMS and DF services. These two services are implemented and managed by the platform as agents on the system where other agents can register and subscribe other agents. Their services are used by every agent that needs to pass information to other agents thus allowing the dynamic discovery of the agents to whom the agents' messages need to be delivered.

In order to deliver agent messages to other agents on the system the Agent Message Transport defined on the FIPA standards and implemented on the JADE framework is used, as well as, the ACL for the structure and specification of each message. An ACL message encapsulates the content of a message in an envelope with the address of the destination agents, the language used for the content, a performative indicating whether the message is a information request or an answers for example and the actual content of the message.

6. Results

Over this chapter results from the experiments with classifier algorithm, suggestion algorithms and the multi-agent system for credit scoring are presented. In the next three sections, details about each experiment conducted will be described, as well as the main conclusions on each part.

This chapter aims to prove the utility of the research conducted during the work presented in this dissertation and also the validity of the conclusions gathered.

6.1. Results with Classification Algorithms

In order to analyze the data in the dataset to build a classification algorithm, some initial tests were conducted using simple classifiers from the Weka Toolkit (35). The dataset was presented to some of the most common classification algorithms and their results were assessed later. The results of these simple tests are shown in the Table 4.

Table 4 - Initial classification models study

Algorithm	Correct Evaluation (%)	Error (%)
J48	77.6	22.4
OneR	72.4	27.6
Multilayer Perceptron	73.5	26.5
Naïve Bayes	75.6	24.4
Linear Regression	76.8	23.2

From these results it is observed that the rate of accuracy is in the region of 70 to 78 %. Classifiers like the J48 and OneR are not simple to update once they require that each time their model has to be build it is required that all model data must be provided. Here a simple update in the model will require a full evaluation which is something that is not ideal. Naïve Bayes and neural networks such as Multilayer Perceptron provide us the capabilities to update the classification model without the need to revaluate all the data presented. Their internal structure is able to update an initial model just considering the new data available and the current model state.

6.1.1 **Evaluation Methodology**

In order to evaluate the accuracy of the developed classification algorithms it was felt the need to choose an evaluation methodology that could prove both the accuracy of such algorithms and their utility by comparing its results with previous work on classification algorithms on the dataset. Due to these restrictions, some of the works found used mainly a percentage split over the dataset to separate it into two different dataset, one for training the classifiers and the other for testing those classifiers.

The work presented in (47) and (42) used a 66% split on the dataset where 66% of the initial dataset was used for training purposes and the remaining 33% to test and evaluate the algorithms' accuracy, a procedure depicted in Figure 19.

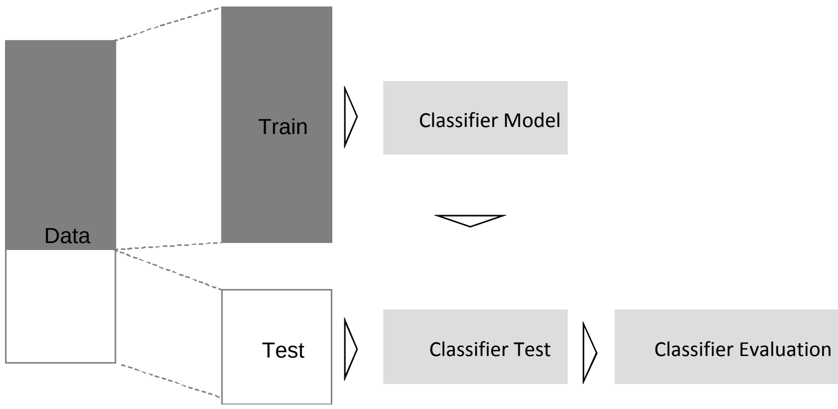

Figure 19 - Classifier's evaluation methodology

This evaluation methodology was then used to compare the work presented in this dissertation with the work of previous authors on the same dataset.

6.1.2 **Classifiers with Feature Selection**

With the dataset used as a case study in this project, a number of tests were made using the algorithms detailed in section 5.2. and 5.3. . In order to access the result of such the developed algorithms, the classifiers Liner Regression and Multilayer Perceptron from the Weka toolkit were used as the base classifier algorithms for our algorithms with feature selection. Those two choices aim to assess the impact of the use of feature selection on classic regression classifiers used in the credit scoring problem as well as the impact on modern algorithms which are currently being assessed like neural networks here represented by the Multilayer Perceptron. As stated in section 6.1.1 the evaluation methodology followed to obtain the accuracy results was the percentage split were the dataset was divided into a training set and a test set with 66% and 33% of the original dataset respectively. For these two initial classifiers their initial results are compared with the results when they are integrated in the developed classifier with feature selection 1 and classifier with feature selection 2 algorithms. In Table 5 the comparative result is presented with the accuracy rates in the dataset for each situation. In this table the term accuracy represents the rate of correctly identified instances and the error rate the rate of misclassified instances in the dataset.

Table 5 - Results of the proposed algorithms

Algorithm	Accuracy (%)	Error (%)
Multilayer Perceptron	73.5	26.5
Multilayer Perceptron with Feature Selection 1	69.7	30.3
Multilayer Perceptron with Feature Selection 2	76.0	24.0
Linear Regression	76.8	23.2
Linear Regression with Feature Selection 1	75.9	24.1
Linear Regression with Feature Selection 2	77.6	22.4

These initial results indicate that the integration of each classifier with the classifiers with feature selection 2 algorithm present the best accuracy than the initial classifier algorithm or the developed classifiers with feature selection 1. From these results it is possible to conclude that classifiers can be improved or degraded by some feature selection algorithms. The objective is to know which feature selection algorithms are worthwhile and those who are not. From the initial proposed feature selection algorithms proposed in section 5.2. only the second developed feature selection algorithm shows promise to improve classifying algorithms.

The selective normalization of attributes between different ranges according to a method responsible to identify the most important attributes may influence some classifiers to pay more attention to those attributes thus increasing accuracy results. On the other hand, the first feature selection algorithm tries to reduce the dataset preserving the considered important attributes and discarding the others. This approach consistently decreased the accuracy of the classifiers across the two initial classifiers studied. This leads to the suggestion that after the initial steps of the creation of a dataset with relevant attributes to a problem, further reducing the dataset does not show promise on increasing classifying results.

Due to the nature of the case study's problem, credit scoring, additional consideration has to be made concerning the utility of the prediction made both with future and past data. In the following analysis it will be studied the propensity some algorithms to repeat known errors, in other words, misclassifying instances in the dataset which have already been present in the classifiers training. This test was made using the full dataset as the training and testing dataset for each classifier. In Table 6, the results from this study are presented including also the results from the developed classifier with feature selection 2 applied to the Multilayer Perceptron and Linear Regression algorithms.

Table 6 - Results with the best developed algorithm, J48 and Naive Bayes

Algorithm	Accuracy (%)	Error (%)
Naïve Bayes	75.6	24.4
J48	90.2	9.8
Linear Regression	76.5	23.4
Multilayer Perceptron	97.0	3.0
Linear Regression with feature selection 2	77.2	22.8
Multilayer Perceptron with feature selection 2	96.3	3.7

As the tests show not all algorithms behave in the same manner. The Linear Regression, Linear Regression with feature selection 2 and Naïve Bayes algorithms were found to have the same rate of accuracy as when they were evaluated with the percentage split test. These algorithms do not seem to take full advantage of the instances already presented to them making the same rate of mistakes when presented with fewer instances for training and previously unknown testing instances. However, algorithms like the J48, Multilayer Perceptron and Multilayer Perceptron with feature selection 2 increase significantly their accuracy rates achieving results ranging from 90% to 97%. This fact demonstrates that these algorithms do not easily make mistakes on situation they have already been trained on. In terms of business understandings, in these algorithms with a case that was initially handled in the financial institution and given for learning the probability of repeating it is lower than with Linear Regression, linear regression with feature selection 2 and Naïve Bayes algorithms. This fact suggests that the neural network and decision tree algorithms retain information better and also that the presence of the feature selection on the Multilayer Perceptron with feature selection 2 does not alter this behavior on the original Multilayer Perceptron algorithm.

Over the next section the results achieved with the developed algorithms to combine feature selection mechanism with classifying algorithms will be compared to other optimization approaches made by other author.

6.1.3 **Comparison with Previous Work on the Dataset**

As stated in section 5.1. , the dataset chosen has already been used in other papers and now a comparative result study between the accuracy rate of other authors work and the work developed in this dissertation will be presented. Unfortunately, some of the algorithms presented, namely the algorithms produced by (45) and (46) use metrics of evaluation not directly compatible with the metrics used in the results produced in section 6.1.2.

The results use the percentage split evaluation test with 66% of the data presented in the dataset used for training the classifiers and the remaining 33% to evaluate the answers given by the classifier. This methodology is explained in section 6.1.1 in detail and was the methodology used by the different authors present in the comparative study.

In Table 7 it is presented the results from the comparative study and the results of the best algorithms developed, in this case classifier with feature selection 2 using the Multilayer Perceptron and the Linear Regression algorithms as the classifier algorithms in each situation presented and the results achieved by two other author with different algorithm proposals on their own work.

Table 7 - Result comparison of some algorithms

	Algorithm	Accuracy (%)	Error (%)
(a)	Combining Feature Selection and Neural Networks for Solving Classification Problems	76.0	24.0
(b)	Selective Bayesian Classifier	75.0	25.0
(c)	Multilayer Perceptron with Feature Selection 2	76.0	24.0
(d)	Linear Regression with Feature Selection 2	77.6	23.4

In Table 7 it is compared the algorithm (a) Combining Feature Selection and Neural Networks for Solving Classification Problems presented by (47), the (b) Selective Bayesian Classifier algorithm presented by (42) and the classifier with feature selection 2 presented in section 5.2.2 applied with the Multilayer Perceptron and the Linear Regression algorithms on two different situations. The first algorithm (a) uses a neural network with a reduced version of the dataset. The reduced dataset comes from attribute selection based on information theory. More information on this algorithm can be found in section 3.2.3 . The second algorithm (b) also uses a reduced version of the dataset in conjunction with the Bayesian Classifier. The reduced version once again is obtained from attribute selection but unlike the previous attempt it is based on a decision tree classifier which considers the position of attributes in the tree to select those which are considered more relevant. More information about this algorithm can be found in section 3.2.4 . The last algorithms, (c) Multilayer Perceptron with Feature Selection 2 and (d) Linear Regression with Feature Selection 2 use the classifier with feature selection 2 algorithms applied to the Multilayer Perceptron and Linear Regression algorithms. Both these algorithms use the the full dataset, however, feature selection is used to distinguish important attributes in the neural network.

Again, according to the results presented in Table 7 the accuracy results obtained from the work presented in this dissertation and other approaches by different authors on the same dataset with the objective of improving classifier algorithms seem similar. Despite having different strategies, all authors improved the accuracy rates from their initial starting classifiers with their proposals. This fact suggests the utility of the work developed in this dissertation as an alternative approach to optimize algorithm accuracy with similar results as other known authors.

6.1.4 **Conclusions**

Over section 6.1. results for the evaluation of different clients represented as instances in a dataset with developed algorithms was presented. These results prove that it is possible to improve classifying algorithm with one of the proposed strategies. This strategy uses different interval ranges in attribute normalization in order to distinguish more important attributes in a dataset on the classifiers used. The detailed implementation algorithm for this approach can be found in section 5.2.2 in the classifier with feature selection 2 algorithms.

On other note, from the results presented in this section it is possible to prove that algorithms like the J48, Multilayer Perceptron retain information about past instances in a dataset in a better way than algorithms like Linear Regression and Naïve Bayes. Moreover, it was also demonstrated that the implementation of the developed classifier with feature selection algorithm 2 with the Multilayer Perceptron classifier preserved the ability of the initial algorithm to retain information on past clients. On the other hand, when the algorithm was implemented with the Linear Regression classifier, which had demonstrated less ability to avoid making errors on information already used on the classifier training, the algorithm had not achieved the same results. This fact suggests that this property is directly related to the type of classifier used on the implementation of this algorithm something that should be taken in to consideration for the credit problem in order to avoid making mistakes evaluating information and situations which has already appeared in the past.

6.2. Experiments with the Suggestion System

The suggestion algorithm described in section 5.3. is intended to have multiple purposes in the credit scoring case study. In this section it will be presented some results of the use of this algorithm in different scenarios inside the case study.

To start with, it will be demonstrated the utility of such algorithm to perform client suggestion activities, in which the algorithm suggests to a client some attribute values which might make its loan application approved easier. Furthermore, it will also be demonstrated the utility of such algorithm from a financial institution view in order to help the work of its consultants. Another use for such algorithm is the cross-selling of financial products to clients as a way to, both, improve clients' credit score and the revenue of the financial institution. It can also point which type of clients are more useful in each financial product.

Finally, the overall conclusions about the usage of the suggestive algorithm are presented based on the results obtained from the experiments described.

6.2.1 Client Suggestion

Current credit scoring mechanisms hardly provide a recommendation system to a client who has applied for a loan application. In this sense there is little a client can do to understand how he could improve its credit score in order to have its loan application approved by a financial institution. To tackle this problem it was demonstrated with the help of the developed suggestion algorithm in section 5.3. a couple of experiments where a suggestive system proves itself useful for the client.

The first scenario considers a client who is assessing its possibilities to have a loan application approved by a financial institution. Sometimes clients want to know their loan application limits before committing to the final use of the money they borrow. With a suggestion system for loan applications this client can introduce its current attributes and search for combinations loan amounts and loan purposes approved by financial institutions. A final step could be to chose from the loan amount and purpose scenarios the most suited for its desires.

To prove the utility and usability of this scenario, a fictional client was created simulating a 40 year old customer with good credit history, self-employed, liable for 2 people with a rented house.

After these inputs the system was able to calculate different alternatives for loan amounts and loan purposes which the client may be granted by the financial institution. The results produced by the algorithm were:

- Saving accounts between 0€ and 200€;
- Balance of checking account 100€ between 500€;
- Installments only can take up to 66% of his disposable income.
- Purposes include buy a new car, repairs and domestic appliances

With the results above the client can know choose from the allowed scenarios those which fit its needs better and apply for it.

The second scenario considers a client whose loan application was refused. Here, the client will try to understand which of its attributes are more influential in the loan decision so he can deal with them. The necessary steps in this situation are:

1. provide the system with a number of attributes which the client cannot change. For instance, in this group can be included a client's gender, marital status, liable people and whether or not he is a foreign citizen;
2. leave any attribute which can be changed to a new value due to different interpretations or financial operations blank;
3. chose the maximum number scenarios for the approval of the loan so the client has different alternatives to chose from.

To prove the utility of the suggestive algorithm in this scenario it was simulated a client whose initial loan application was refused. Due to that reason the list of unchangeable attributes was made and presented to the decision system. These steps resulting in presenting the system with a male divorced client aged 54, unskilled employee employed for more than 7 years with a 20% installment rate on disposable income with no other debtors or guarantors, no property, no installment plans, a telephone number and a rented house.

The suggestive algorithm was able to discover new arrangements in the on attribute values that the client can change which allow him to have its loan application approved. These scenarios include:

- a checking account with a balance above 200€;

- the purpose of a loan could be a used car, vacation, business, repairs and domestic appliances;
- the amount of the loan could be up to 10000€;
- No previous credits taken.

As a note on the results here presented are only to exemplify potential use for the algorithm on a client perspective. Also, as the suggestion system is not intended to be exhaustive, different runs might give different results.

6.2.2 **Financial Suggestion**

Employees from financial institutions have to deal with clients who ask them why certain loan applications where refused or loan scenarios for their needs. Entering their client data in the system with some missing attributes might help these employees some justifications to the clients on why its loan application was refused and some likely attribute values responsible for the refusal or positive scenarios for the attributes given by the client. In order to test this approach a male married customer with a history of delays in paying its former loan installments, who's applying for a 5000€ loan for educational purposes, age 30, intending to repay it in 18 months, was simulated. The answers obtained for approved loans were:

- Demand a savings account at the bank between 500€ and 1000€;
- To have, at least, a rented house;
- Does not need guarantors;
- Other installment plans have to be at the same bank;
- Savings account between 100€ and 500€.

This information can now be gathered and presented to the client as a possible explanation of why its loan application was refused.

Other use for this suggestion system by the financial institution could be the discovery of the most typical client type to have a loan application under a certain type of conditions. To prove this use of the algorithm, it was possible to determine a type of client who has a loan application approved to buy a new car when the buyer is a married woman with real estate property and history of delays in credit payment history. Running the suggestive algorithm under these conditions provided the following results for the client type under these conditions:

- Balance of the checking account between 0 or 200€ or greater than 200€;

- Unskilled, skilled or management job;
- has a guarantor or another co-applicant;
- Up to 50% installment rate in percentage of disposable income

With these two experiments the value of the developed suggestive algorithm in a financial institution was demonstrated. The algorithm provides answers to client's justification request in a satisfactory level and is also able to discover client pattern present in the classifier used on the by the suggestive algorithm.

A special case of the situation presented in the last experiment is the need to relate financial products to the case of loan application being approved. In this approach the suggestive algorithm might be used be the financial institution to search for clients whose credit score could be improved by acquiring financial products from the financial institutions. This approach enables the financial institution to disseminate its financial products through its clients where those clients are considered worthwhile. Simulating loan applications in which client have savings, property and good credit history at a financial institution it would be beneficial which other attributes appear most times connected to these situation. The results of running the suggestion system include as an example:

- Checking account exist at the financial institution with a positive balance;
- Male and women married or single;
- Up to liable people;
- Duration of the loan on the longest suggestion is 72 months.

With the results presented, business rules could now be made stating that any loan application which falls within the proposed suggestions should invite clients to transfer or create savings based on financial products of the financial institution. To justify these actions clients could be informed that application on these condition have better possibilities of being granted with a loan.

For the situation specified above the suggestion system provides the following client type where the presence of such financial product has advantage to have its loan application approved.

6.2.3 Conclusions

The results provided by the suggestion algorithm do, indeed, prove its utility on the credit scoring system under different scenarios and different objectives.

In the client side, the algorithm is useful to assess and explore loan opportunities based on a set of fixed attributes a client cannot change. In some cases the suggestion algorithm finds alternative situation to change the decision of a loan application by a financial institution.

From a financial institution, the algorithm is helpful giving explanations to financial employees on why certain clients' loan application are refused. Other use considered and validated was the use of the discovery of client types for certain loan situation allowing these employees to develop marketing techniques to clients which fall into those patterns. A special case for this last case is the opportunity to perform cross-selling activities mixing financial products and loan applications in the same package thus granting the client with the necessary guarantees to have its loan application approved while also selling related financial products to clients.

Finally, it the suggestion algorithm proposed proved, through the experiments presented, to be a robust algorithm handling well different sizes of unknown attributes dynamically. Also the fact that the algorithm is independent from the classifier algorithm used to assess loan application does not make restrictions on the group of decision classifiers which could be used in the system to enable the use of the suggestion algorithm.

6.3. MAS Platform Results

In this section are presented results from the use of the developed multi-agent system from credit scoring. These results aim to prove the functionality of the developed system the properties evidenced in section 5.4. .

This section starts by asserting the correct functionality of the MAS to decide on loan applications and perform loan suggestions. Afterwards, some fault tolerance tests are made to the system in order to prove its reliance to failure and service disruption. Finally, some conclusions are presented based on the developed system results.

6.3.1 **Classification and Suggestion Functionality**

In order to test the correct functionality of the MAS, a simulation including loan application decisions and client suggestion were performed. The MAS was initially fed the credit dataset described on section 5.1. through a file specified to a Feeder agent. It is important to say that the MAS already had 1 Model agent, 2 Decision agents and 2 Suggestion agents.

An Inquiry agent was then brought up on the system so the user could enter its requests for the system. Starting by the assessment of loan applications in Figure 20 it is demonstrated the decision presented by the MAS which was then showed to the user through the Inquiry agent GUI to the end user. Success and accuracy rates for the MAS on loan decisions are not presented in this section because this study has already been conducted in section 6.1. .

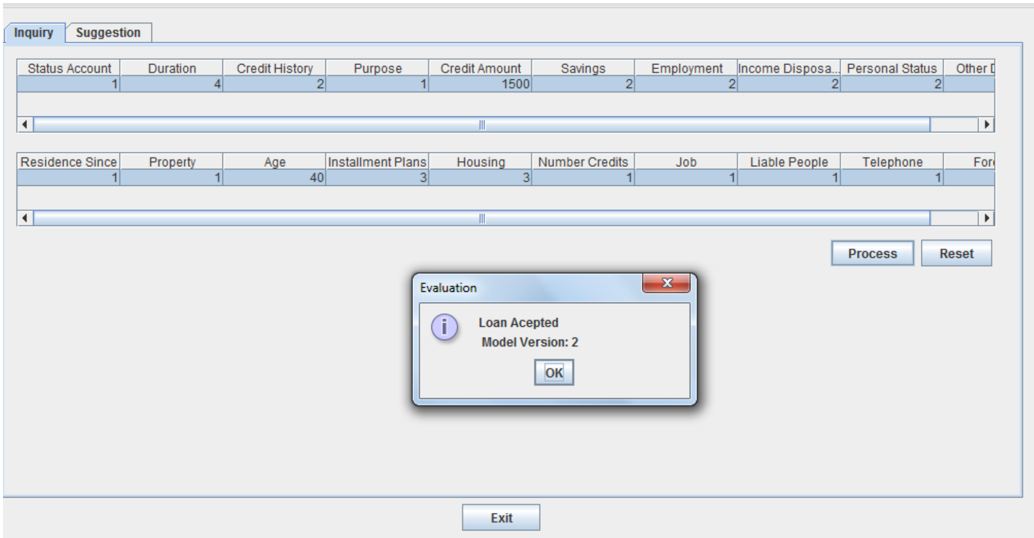

Figure 20 - Example of a loan application decision by the MAS

On the same system, a suggestion request was also simulated to prove that the suggestion algorithm did in fact find scenarios where the attributes introduced in the first row of each table could be used for in approved loan applications. Figure 21 shows the results for the simulation of a suggestion requested to the system on a 30 year old client with housing for free and a loan with 40 months of duration.

The results presented as suggestions then were assessed as a complete loan application which revealed that all the suggestion did in fact result in approved loan applications. From this result it is proven that the suggestive capability of the developed MAS does in fact work and it also provides positive loan application scenarios were possible.

These two type of experiments show that the system behaves as predicted using different agents to produce answers to the requested problems.

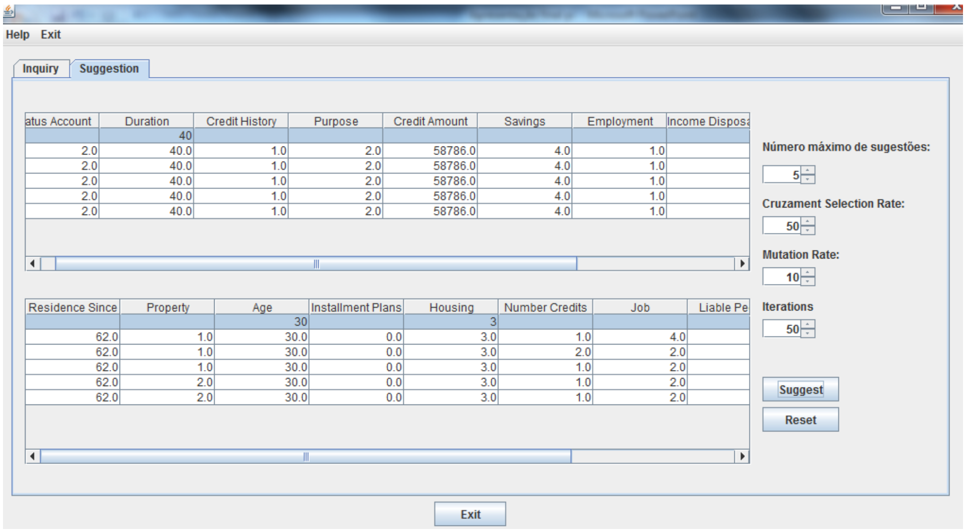

Figure 21 - Suggestion results from the MAS

6.3.2 **Fault tolerance**

As the credit scoring problem is an important activity in a financial institution we presented the developed MAS to some tests in order to assess its robustness and resilience to failure. The developed test environment considered was initially composed:

- 2 Model agents;
- 2 Decision agents;
- 2 Suggestion agents.

From this initial configuration another Feeder agent was introduced in the system and loaded the dataset instances into the MAS. Afterwards, an inquiry agent was introduced to perform some loan decision and suggestion requests to assess the initial system functionality. After confirming the correct functionality of the MAS some tests were conducted to evaluate the effect of having some of the agent crashing on the system.

First, crashing one Decision and one Suggestion agents from the testing system, did not result in a system failure, since the Inquiry agent was always redirected for the available Decision and Suggestion agents. This proved that when redundant agents are present the system does not fail just by the crash of one of these agents. In the presence of all the Suggestive agents not being operational the inquiry agent still performed loan decision requests without being influenced by the failure of all the Suggestive agents.

On other approach upon the initial system, about 10 Inquiry agents were created and made to launch automated loan application for approval and suggestion requests. These action put stress on the MAS. The response of the system was that when stressed, the Suggestive and Decision agents cloned themselves, helping them cope with the increase in inquiries by the Inquiry agents. On these tests a total of 5 Suggestive agents and 5 Decision agents were present at the end of the stress test proving that the system adapted itself to the increase of requests.

6.3.3 **Conclusions**

The results provided by the MAS proved some of the properties described in section 5.4. . To start with, with a use of the developed system it was possible to perform both suggestive and loan decision tasks using known classifiers. Both these action required the interaction of some agents in the MAS which happened without any issues even when redundant agents were present.

Furthermore, it was also possible to determine that the failure of individual agents does not result always in a general system failure, rather the redundant agent are able to rise to the occasion and perform the tasks available. These results did not account for the fact where the server where the MAS server represents a single point of failure in the system, however the MAS platform could also be configured to use more than one server to run this MAS.

Finally, the use of different implementation of certain agent types did not comprise the operation of the system as it was able to conciliate different implementation of the same agent types.

7. Conclusion and Future Work

Over this chapter a synthesis of the work done during the elaboration of this dissertation is introduced and compared to the initial objectives proposed. A quick review of the most important works and studies is made. Also, it is also presented a list of relevant work done in concurrence with the work of this dissertation such as paper publication, grants and internships related to the problem studied. The chapter ends with consideration for future work and new directions for the developed work.

7.1. Synthesis of the Work Done

The work presented in this dissertation began by an initial study on the problem of credit scoring detailing current trends and models on both financial institutions and research conducted on the case study problem. Following the evaluation of the decision models used for credit scoring, a review of some of the most used learning algorithms on those models was also conducted evaluating not only the classical algorithms but also optimization on classical learning algorithms. This initial study completed one of the objective of the work presented in this dissertation, the study of decision algorithms based on techniques from ML and AI that rely on data taken from datasets as well as interesting points of optimization for such algorithms.

Multi-agent system theory and applications were also reviewed during the course of this work due to the relevance of such system of some of the developments made in this area. Some related detailed on this dissertation also made reference to MAS projects and their combination with DM and ML tasks for predictive actions. On this stage, another global objective of the work presented in this dissertation was achieved, which was the review of current projects on multi-agent systems capable to perform decisions and current trends on system design to integrate ML algorithms into these systems.

On the development side of this dissertation's work there is the effort to improve classifier accuracy rates by using feature selection algorithms on the data provided to such classifiers. The studies conducted on this field showed promise to one of the proposed alternatives which seemed to have better results than the original classifier. This was yet another objective of the presented dissertation.

The suggestion algorithm was yet another of the developments made on this dissertation's work which took into consideration incomplete set of information about clients. For this incomplete information the suggestion algorithm was able to formulate positive loan scenarios with the known information using theory from genetic algorithms to produce its suggestion. The algorithm was then proved useful on a number of situations like client suggestion, cross-selling at financial institutions and the explanation of the decision process on the classifier and model used to perform credit scoring. With the proposed suggestion algorithm another of the initial objectives was done successfully.

The work presented in this dissertation ended with the design and implementation of a MAS able to take advantage of the previous studies. This MAS was organized according to an expert system where agents have specialized tasks. Only with the collaboration and cooperation between agents and their tasks is it possible to achieve the global MAS goals which the results proved possible on the developed system. The integration of the previous developments in the multi-agent system created for the credit scoring problem enabled the achievement of the last dissertation work objective.

7.2. Relevant Work

During this research several parallel activities and tasks were performed that have contributed to the evolution of this work. The list of such activities is described below:

1. Publications on Journals
    - Fábio Silva, Cesar Analide, "Information Asset Analysis: Credit Scoring and Credit Suggestion", International Journal of Electronic Business (IJEB), Vol. 9, no. 3, Special Issue: "Expertise, Innovation and Technology: Are We in the e-XXI Century", Inderscience Publishers, 2011.

IJEB is a Journal used to develop, promote and coordinate the development and practice of electronic business methods to meet the needs of accelerating technological change and changes in the global economy.

2. Publication on conferences with peer-review:
    - Fábio Silva, Cesar Analide, Paulo Novais, "Credit Scoring Data for Information Asset Analysis", in Soft Computing Models in Industrial and Environmental Applications, 6th International Workshop (SOCO 2011), Corchado E., Snasel V., Sedano J., Hassanien A.E., Calvo J.L., Slezak D., (Eds.) Spriger - Series Advances in Intelligent and Soft Computing, vol. 87, DOI: 10.1007, ISBN 978-3-642-19644-7_24, (International Workshop on Soft Computing Models in Industrial Applications, Salamanca, Spain, April/2011), 2011.

      This conference is about soft computing, a collection or set of computational techniques in machine learning, computer science and some engineering disciplines, which investigate, simulate, and analyze very complex issues and phenomena. This conference is mainly focused on industrial and environmental soft computing applications.
    - Fábio Silva, Cesar Analide, "Multi-Agent System for Credit Scoring", in Proceedings of the 15th Portuguese Conference on Artificial Intelligence, Luís Antunes, H. Sófia Pinto, Rui Prada, Paulo Trigo (Eds.), ISBN: 978-989-95618-4-7, Lisbon, Portugal, 10-13th October, 2011.

      With focus on a fundamental subject of Artificial Intelligence, the 6th edition of Multi-Agent Systems: Theory and Applications - MASTA 2011 Thematic Track, is a forum for presenting and discussing the most recent and innovative work on Multi-Agent Systems, held as a thematic track of EPIA, the Portuguese Conference on Artificial Intelligence.
    - Fábio Silva, Cesar Analide, "Design of an application for credit scoring and client suggestion", in Proceedings of E-Activity and Leading Technologies, E-ALT 2010, Oviedo, Spain, 8-10 November 2010.

      The scope of the E-ALT2010 conference covers all areas around Internet technologies and applications, from programming to e-business, going through all the technical and wider issues arising from them.
3. Grants

- The work related to the to the paper submitted for the Proceedings of E-Activity and Leading Technologies was supported in part by a Scholarship APPIA, Portuguese Association for Artificial Intelligence.

4. Internship at MillenniumBCP
    - During this research I have undergone an internship at MillenniumBCP where I was responsible for building decision mechanism prototypes also related to data mining and trend analysis problems. This experience has improved my vision on how to deal with the problems stated in this work, thus enriching it.
5. Developed prototypes
    - In order to test results and assumption presented throughout this work a prototype of a multi-agent system was developed. This system enabled the validation of some research results as well as the utility of the global system in comparison with more classical approaches.

All the documents that support these works are available through the author.

7.3. Future Work

Although the major objectives proposed at the beginning of the dissertation seem to be achieved successfully, there is room for improvement on the work conducted so far. In this context, future work on the developments report in this dissertation could involve:

- developing new classification algorithms exploring more optimization techniques available in the literature order to obtain better evaluations. In order to achieve this new algorithms have to be evaluated based on their utility for the credit scoring problem.
- improvements on the suggestion algorithm, so the provided results could be present in natural language instead of list of positive scenarios. Other possibility to improve the suggestive algorithm could be the integration of more state of the art theory on genetic algorithm to try to boost the results presented;
- development of new Implementations for the Feeder, Model and Inquiry agents in the developed MAS. These new implementations could extend the functionality of the system like supporting new data source in the case of the Feeder agent, supporting new classifiers in the case of the Model agent and the ability to receive requests from web services and rest operations in the case of new Inquiry agent implementations;

- development of ensemble agents in the proposed MAS to allow the use of multiple classifiers on the system resulting decision from the combination of the results of these classifiers. This would also improve the robustness of the MAS as the evaluation of loan application would be made according to different set of classifiers instead of relying on the interpretation of a single classifier.

On a more business related future work, the results and conclusions about credit scoring models and the utility of the suggestive system still caress the review of financial institutions employees specialized on the subjected. The work here presented has been validated on some conference and journal papers but would be better contextualized if the data used to produce results was imported from real financial institutions.

The comparison between accuracy rates from the developed system and commercial credit scoring mechanism like the FICO scoring system would also be beneficial to further validate the work here presented.

Finally, an adaptation on the MAS to enforce the Basel II rules applied to the credit scoring problem would also present great benefit to the developed MAS which could allow its commercial use on national and international financial institutions, moving the developed system from a prototype and proof of concept to a commercially ready application.

Bibliography

1. *Designing Data Governance.* **Khatri, Vijay and Brown, Carol V.** 2010. Communications of the ACM. Vol. 53, pp. 148-152.

2. **Higson, Chris and Waltho, Dave.** *Unlocking the Value of Information.* s.l. : EURIM, 2009.

3. *Credit-scoring methods.* **Vojtek, Martin and Koenda, Evzen.** 2006, Czech Journal of Economics and Finance.

4. *Neuro-Based Artificial Intelligence Model for Loan Decisions.* **Eletter, Shorouq F., Yaseen, Saad G. and Elrefae, Ghaleb A.** 2010, American Journal of Economics and Business Administration, pp. 27-34.

5. **Islam, Samsul, Zhou, Lin and Li, Fei.** *Application of Artificial Intelligence (Artificial Neural Net-work) to Assess Credit Risk: A Predictive Model for Credit Card Scoring.* s.l. : Dissertation. School of Management Blekinge Institute of Technology, 2009.

6. **Higson, Chris and Waltho, Dave.** *Valuing Information as an Asset.* s.l. : SAS whitepaper, 2010.

7. *Information Assets and their Value.* **Engelsman, Wilco.** 2007. 6th Twent Student Conference on IT.

8. *Agent-based Distributed Data Mining: The KDEC Scheme.* **Klusch, Matthias, Lodi, Stefano and Moro, Gianluca.** 2003. Intelligent Information Agents- The Agent Link Perspective Lecture Notes in Computer Science.

9. *Make Your Data a Strategic Asset.* **Wright, Ken.** 2004. SUGI 29 Proceedings.

10. *Decision Making for Predictive Maintenance in Asset Information Management.* **Faiz, Rizwan B. and Edirisinghe, Eran.** 2009. Interdisciplinary Journal of Information, Knowledge, and Management. Vol. 4, pp. 23-36.

11. **Waterbury, Bob.** *Artificial INtelligence Expands Frontiers in Asset Management.* s.l. : Control Magazine Sarle, 2000.

12. **Somekh, Bridget.** *Action Research: a methodology for change and development.* s.l. : Open University Press, 2005.

13. **Banco de Portugal.** *Statistical bulletin - August 2011.* s.l. : Banco de Portugal, 2011.

14. **Jentzsch, Nicola.** *Financial Privacy: An International Comparison of Credit Reporting Systems.* s.l. : Springer, 2007. 9783540733775.

15. **Anderson, Raymond.** *The credit scoring toolkit: theory and practice for retail credit risk management and decision automation.* s.l. : Oxford University Press, 2007. 9780199226405.

16. **Thomas, L. C., Edelman, David B. and Crook, Jonathan N.** *Credit scoring and its applications.* s.l. : SIAM, 2002. 9780898714838.

17. **Durand, David.** *Risk Elements in Consumer Instalment Financing.* s.l. : National Bureau of Economic Research, Inc, 1941.

18. **Siddiqi, Naeem.** *Credit Risk Scorecards Developing and Implementing Intelligent Credit Scoring.* s.l. : John Wiley and Sons, 2006. 9780471754510.

19. **Beares, P. R., Beck, R. E. and Siegel, S. M.** *Consumer Lending.* s.l. : American Bankers Association, 2001.

20. **Basel Committee on Banking Supervision.** *International Convergence of Capital Measurement and Capital Standards: A Revised Framework.* Switzerland : Bank for International Settlements, 2006.

21. **Fair Isaac Corporation.** *Understanding Your FICO Score.* s.l. : Fair Isaac Corporation, 2010.

22. **Foust, Dean and Pressman, Aaron.** Credit On The Edge. *Business Week.* 2 7, 2008.

23. *Using neural network ensembles for bankruptcy prediction and credit scoring.* **Tsai, Chih-Fong and Wu, Jhen-Wei.** s.l. : Elsevier, 2008. Expert Systems with Applications. pp. 2639–2649.

24. *An approach to efficient business intelligent system for financial.* **Simic, Dragan and Simic, Svetlana.** 2007. Soft Computing – A Fusion of Foundations, Methodologies and Applications. pp. 1185-1192.

25. *Credit scoring using artificial immune system algorithms: A comparative study.* **Bhaduri, Antariksha.** 2009. World Congress on Nature Biologically Inspired Computing. pp. 1540–1543.

26. *A data mining approach to credit risk evaluation and behaviour scoring.* **Madeira, Sara C., Oliveira, Arlindo L. and Conceição, Catarina S.** s.l. : Springer Berlin/Heidelberg, 2003. Progress in Artificial Intelligence, Lecture Notes in Computer Science. pp. 184-188.

27. *Using neural network ensembles for bankruptcy prediction and credit scoring.* **Tsai, Chih-Fong and Wu, Jhen-Wei.** s.l. : Elsevier, 2008, Expert Systems with Applications, Vol. 34, pp. 2639 - 2649.

28. *Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters.* **Min, Jae H. and Lee, Young-Chan.** s.l. : Elsevier, 2005, Expert Systems with Applications, Vol. 28, pp. 603 - 614.

29. *Comprehensible credit scoring models using rule extraction from support vector machines.* **Martens, David, et al.** [ed.] Elsevier. 2007, European Journal of Operational Research, Vol. 183, pp. 1466 - 1476.

30. *Credit Scoring and the Availability, Price, and Risk of Small Business Credit.* **Berger, Allen N., Frame, W. Scott. and Miller, Nathan H.** 2005, Journal of Money, Credit, and Banking, pp. 191-222.

31. **Sallam, Rita L., et al.** *Magic Quadrant for Business Intelligence Platforms.* s.l. : Gartner, 2011.

32. **Evelson, Boris.** *The Forrester Wave™: Enterprise Business Intelligence Platforms, Q4 2010.* s.l. : Forrester, 2010.

33. **Kobielus, James.** *The Forrester Wave™: Predictive Analytics And Data Mining Solutions, Q1 2010.* s.l. : Forrester, 2010.

34. *Yale: rapid prototyping for complex data mining tasks.* **Mierswa, Ingo, et al.** 2006. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.

35. **Witten, Ian H. and Frank, Eibe.** *Data Mining: Practical Machine Learning Tools and Techniques.* San Francisco : Elsevier, 2005.

36. **Orange.** *Orange.* [Online] [Cited: 10 21, 2011.] http://orange.biolab.si/.

37. **KNIME.** *KNIME - Professional Open-Source Software.* [Online] [Cited: 10 21, 2011.] http://www.knime.org/.

38. **Heaton, Jeff.** *Programming Neural Networks with Encog 2 in Java.* s.l. : Heaton Research Inc., 2010.

39. *KEEL: a software tool to assess evolutionary algorithms for data mining problems.* **Alcalá-Fdez, J., et al.** 2009, Soft Computing - A Fusion of Foundations, Methodologies and Applications - Special Issue on Evolutionary and Metaheuristics based Data Mining (EMBDM), pp. 307-318.

40. *Design of an Application for Credit Scoring and Client Suggestion.* **Silva, Fábio and Analide, Cesar.** 2010. Proceedings of the IASK International Conference E-Activity and Leading Technologies & InterTIC.

41. **Wu, Xindong and Kumar, Vipin.** *The Top Ten Algorithms in Data Mining.* s.l. : Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2009.

42. *Scaling up the naïve Bayesian classifier: using decision trees for feature selection.* **Ratanamahatana, Chotirat A. and Gunopulos, Dimitrios.** 2002. Proceedings of Workshop on Data Cleaning and Preprocessing.

43. *Consumer Credit Scoring using an Artificial Immune System Algorithm.* **Leung, Kevin, Cheong, France and Cheong, Christopher.** s.l. : IEEE Xplore, 2007. IEEE Congress on Evolutionary Computation. pp. 3377-3384.

44. *Credit scoring with a data mining approach based on support vector machines.* **Huang, Cheng-Lung, Chen, Mu-Chen and Wang, Chieh-Jen.** s.l. : Elsevier, 2007. Expert Systems with Applications. pp. 847–856.

45. *Two-stage genetic programming (2SGP) for the credit scoring model.* **Huang, Jih-Jeng, Tzeng, Gwo-Hshiung and Ong, Chorng-Shyong.** s.l. : Elsevier, 2005. Applied Mathematics and Computation.

46. *Genetic programming for data classification: partitioning the search space.* **Eggermont, Jeroen, Kok, Joost N. and Kosters, Walter A.** 2004. Proceedings of the 2004 Symposium on Applied Computing.

47. *Combining feature selection and neural networks for solving classification problems.* **O' Dea, Paul, Griffith, Josephine and O' Riordan, Colm.** 2001. Proc. 12th Irish Conf. Artificial Intell. Cognitive Sci. pp. 157-166.

48. **Wooldridge, Michael.** *An Introduction to MultiAgent Systems.* s.l. : John Wiley & Sons, 2002.

49. *Intelligent agents: theory and practice.* **Wooldridge, Michael and Jennings, Nicholas R.** 1995. The Knowledge En-gineering Review. pp. 115-152.

50. *Anthropopathy and its Assessment in Virtual Entities.* **Analide, Cesar, et al.** 2004. Innovations through Information Technology, Proceedings of the IRMA 2004 In-ternational Conference.

51. **Weiß, Gerhard.** *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence.* EUA : MIT Press, 1999.

52. **Bellifemine, Fabio L., Caire, Giovanni and Dominic, Greenwood.** *Developing Multi-Agent Systems with JADE.* s.l. : JohnWiley, 2007.

53. **Deransart, Pierre, Ed-Dbali, AbdelAli and Cervoni, Laurent.** *Prolog: The Standard.* s.l. : Springer-Verlag, 1996.

54. *The Foundation for Intelligent Physical Agents (FIPA).* [Online] 2005. [Cited: 9 15, 2011.] http://http://www.fipa.org.

55. **Friedman-Hill, E.** *JESS: The Expert System Shell for the Java Platform.* USA : Distributed Computing Systems, 2002.

56. **Poslad, S., Bourne, R. and Hayzelden, L.** *Agent Technology for Communications.* s.l. : John Wiley and Sons, 2000.

57. *Software Agent technologies.* **Nwana, H. and Wooldridge, M.** 1997, British Telecommunications Technology Journal, pp. 16-27.

58. *Agent-Mining Interaction: An Emerging Area.* **Cao, Longbing, Luo, Chao and Zhang, Chengqi.** 2007. Autonomous Intelligent Systems: Multi-Agents and Data Mining. pp. 60-73.

59. *A Review on Data mining from Past to Future.* **Venkatadri, M. and Reddy, Lokanatha C.** 2011, International Journal of Computer Applications, pp. 19-22.

60. *Data Mining Applications in the Banking Industry in China (1998-2007).* **Dan, Zhao.** 2008. International Conference on Information Management, Innovation Management and Industrial Engineering.

61. *Application of Multilayered Multi-Agent Data Mining Architecture to Bank Domain.* **XinLi, Huang and Chosler, Rajiv.** 2007. International Conference on Wireless Communications, Networking and Mobile Computing.

62. *A Multi-Agent Distributed Data Mining Model Based on Algorithm Analysis and Task Prediction.* **Zhou, Dan, Rao, Wenbi and Lv, Fangsu.** 2010. 2nd International Conference on Information Engineer-ing and Computer Science (ICIECS).

63. *An Agent-Based Architecture for Supply Chain Finance Cooperative Context-Aware Distributed Data Mining Systems.* **Xiang, Liu.** 2008. Proceedings of the 2008 Third International Confe-rence on Internet and Web Applications and Services.

64. *CAKE – Classifying, Associating and Knowledge Discovery - An Approach for Distributed Data Mining (DDM) Using PArallel Data Mining Agents (PADMAs).* **Khan, Danish.** 2008. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

65. *Information Asset Analysis: Credit Scoring and Credit Suggestion.* **Silva, Fábio and Analide, Cesar.** s.l. : Inderscience Publishers, 2011, In-ternational Journal of Electronic Business (IJEB) Special Issue: "Expertise, Innovation and Technology: Are We in the e-XXI Century".

66. **Hofmann, Hans.** Statlog (German Credit Data) Data Set . *UCI - Machine Learning Repository.* [Online] [Cited: 10 21, 2011.] http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29.

67. *Credit Scoring Data for Information Asset Analysis.* **Silva, Fábio, Analide, Cesar and Novais, Paulo.** 2011. Com-puting Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011.

68. **Dianati, Mehrdad, Song, Insop and Treiber, Mark.** *An Introduction to Genetic Algorithms and Evolution Strategies.* University of Waterloo, Ontario, N2L 3G1. Canada : s.n., 2002.

69. *Multi-Agent System for Credit Scoring.* **Silva, Fábio and Analide, Cesar.** 2011. Proceedings of the 15th Portuguese Conference on Artificial Intelligence, Multi-Agent Systems: Theory and Applications.