

---

## **Information asset analysis: credit scoring and credit suggestion**

---

Fábio Silva\* and Cesar Analide

Department of Informatics,  
University of Minho, Campus Gualtar,  
4710-057 Braga, Portugal  
Fax: (+351) 253 604 471  
E-mail: f.aandree@gmail.com  
E-mail: analide@di.uminho.pt  
\*Corresponding author

**Abstract:** Risk assessment is important for financial institution, especially in loan applications. Some have already implemented their own credit-scoring mechanisms to evaluate their clients' risk and make decisions based on this indicator. In fact, the data gathered by financial institutions is valuable source of information to create information assets, from which credit-scoring mechanisms can be developed. The purpose of this paper is to, from information assets, create a decision mechanism that is able to evaluate a client's risk. Furthermore, a suggestive algorithm is presented to better explain and give insights on how the decision mechanism values attributes.

**Keywords:** credit scoring; artificial intelligence; knowledge extraction; data mining; intelligent agents; multi-agent systems.

**Reference** to this paper should be made as follows: Silva, F. and Analide, C. (2011) 'Information asset analysis: credit scoring and credit suggestion', *Int. J. Electronic Business*, Vol.

**Biographical notes:** Fábio Silva has a Degree in Computer Engineering from the University of Minho in Portugal since 2009 and is currently a Master student at the Department of Informatics of the University of Minho.

Cesar Analide holds a PhD from the University of Minho since 2004 and is a Professor at the Department of Informatics, School of Engineering, at the same University. He works in the scientific area of artificial intelligence, particularly in the fields of knowledge representation, knowledge extraction and multi-agent systems.

---

### **1 Introduction**

Nowadays, people are becoming increasingly dependent on 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 a loan that is in fact affordable to them. As a consequence, some people tend to delay their monthly instalments or, in extreme cases even become incapable of repaying their debt back to the financial institution. A client history provides an excellent source of information for predicting the

behaviour of future clients. In fact, some rules and patterns can be identified in this data history that may be relevant when deciding where a future client has its loan application accepted or not. From the perspective of information as an asset, this client's data history usage creates valuable assets to an organisation. Moreover, the information gathered from these sources is considered to be one of the six types of assets for organisations, namely it falls into the category of IT information asset (Khatri and Brown, 2010). Furthermore, the best-managed companies recognise information as a key asset and focus more on information than technology while optimising their business performance (Higson and Waltho, 2009). In this context, many financial institutions are implementing or improving client classification systems 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 (Vojtek and Kocenda, 2006). In fact, artificial intelligence and data mining poses an interesting appeal when considering that they may help in making semi-autonomous or even completely autonomous decision mechanisms able to learn and react to changes and new trends in almost real time. Only in present times are these financial institutions conducting studies to evaluate how techniques from artificial intelligence and data mining can be used to predict client behaviour (Eletter et al., 2010; Islam et al., 2009). This paper is particularly aimed at credit-scoring systems using previous records from old clients to predict and avoid those classified as bad clients in terms of debt repayment.

This introduction is followed by a section where related work is discussed presenting relevant classification models used by financial institutions today. In this section, it is also discussed some approaches made seen in the literature to improve classification algorithms. This section ends with a reference of present toolkits and frameworks available today to use when developing classification systems.

In Section 3, the data set used in this paper is presented. Section 4 reveals a study conducted to improve the performance of neural networks in the problem at hand. In the end of this section, results from the algorithms proposed are shown and performance increases observed.

Later, in Section 5, a case study to support the implementation of suggestive algorithms in this context is presented as well as a suggestive algorithm based on genetic programming and the improved classifier explained in Section 4.

In Section 6, an agent-based system architecture is proposed to integrate all algorithms developed and provide all those functionalities in a real-world application prototype.

This paper ends with conclusions of the work developed and some suggestions of future work to further develop these mechanisms.

## **2 Related work**

An evaluation of the state of the art in risk assessment and decision models implemented in some financial institution is presented. It discussed models, which were implemented and even some models in study for client classification. Furthermore, a review of some algorithms and optimisations upon them is reviewed, as well as,

some tools and frameworks currently available that may help build autonomous decision systems.

### *2.1 Models*

Some particularities exist when building decision models that must be taken into consideration. Legal issues, for example, must be of prime concern as any decision made that does not meet the appropriate laws are considered illegal and must be disregarded. Client discrimination based on attributes such as race or gender is generally illegal in most countries and may justify legal suites to those who ignore these considerations. Consequently, these models should take into consideration all the aspects stated earlier.

Most financial institutions use statistical pattern recognition models to build their own decision mechanism (Vojtek and Kocenda, 2006; Islam et al., 2009). Among those systems, models that use the Linear Discriminant Analysis can be found. This technique aims to classify a heterogeneous population into homogeneous subsets on which the decision mechanism is further developed. Here, the objective is to choose the linear combinations of explanatory variables, which can separate most subsets of data with the maximum distance between means of these subsets.

Most financial institutions use statistical pattern recognition models to build their own decision mechanism. In Czech and Slovak Republics' national institutions, the most used technique is the Logit Analysis, which is an improvement upon the Linear Discriminant analysis technique (Vojtek and Kocenda, 2006). This model is considered an extension from the Linear Discriminant Analysis, which accounts for the non-normality of the data used.

Modern scoring mechanisms are mostly based on machine learning and data-mining techniques associated with artificial intelligence (Madeira et al., 2003). 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 utilisation and product quality. Artificial intelligence use, according to Faiz and Edirisinghe (2009), neural networks, expert systems and fuzzy logic. 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 modelling. Despite existing for over 50 years, neural networks have only recently increased its practical application owing to low-cost computer with high-speed processing.

Fuzzy logic is based on 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. When a threshold indicator, measuring the level of truth, is exceeded, activities and tasks may be triggered by such system. Expert systems, unlike the previous two techniques, simulate the judgement and experience of a set of organised 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 is also considered (Faiz and Edirisinghe, 2009; Simic and Simic, 2007) 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. Other modern approaches use algorithms from artificial immune systems to solve the credit-scoring problem. In Bhaduri (2009), it can be seen three different algorithms using this approach and it also provides a comparison with others.

## 2.2 Algorithms

As noted earlier, in this context, artificial intelligence techniques use machine learning and data mining to produce results. In this aspect, it can be found in the literature several algorithm optimisation 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 on genetic programming to classify instances (Huang et al., 2005). Another example uses a combination of decision trees with genetic programming and is also able to improve classification (Eggermont et al., 2004).

Neural networks are also focus of optimisation and some approaches try to make use of feature selection algorithms before constructing the neural network, making some attributes more relevant in this structure (O'Dea et al., 2001). Feature selection using decision trees may also be 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 (Ratanamahatana and Gunopulos, 2002).

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

## 2.3 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 mining data.

In this context, open-source tools like RapidMiner (Mierswa et al., 2006) or Weka (Witten and Frank, 2005) provide a vast list of data mining and machine learning 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 (Mierswa et al., 2006) for instance, as stated in their web page, is used by the Bank of America. In fact, both of the above-mentioned tools are referenced in many credit evaluation papers.

In a more specialised context, namely neural networks, it can be found Encong (Heaton, 2010), a comprehensive framework for neural networks. To evaluate evolution algorithms for data mining, KEEL is also mentioned. This tool allows evaluating different evolution algorithms as well as integrating them with other software tools (Alcalá-Fdez et al., 2009).

In this paper, the Weka Toolkit (Witten and Frank, 2005) 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 machine learning algorithms for data-mining tasks, which can be applied directly in a data set or used in a java program. Weka has also an active support community and their program is released as open-source software.

### 3 Problem description

In this paper, the problem of client classification for loan applications will be considered. The objective is to improve classification models based on neural networks and build a system that is able to dynamically update itself as new data becomes available in an autonomous way.

To achieve better results in client classification, some past data about a set of attributes characterising each client of a financial institution is taken into consideration and is used to build the classification model.

Furthermore, the system must also be able to suggest clients explanation on how they can improve their situation, increasing their chances to have loan applications accepted by the financial system.

The goal of this system is to indicate a client what are the most advantageous characteristics he/she may possess to be granted with a loan application. With an incomplete set of attributes characterising 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.

In this work, a data set related to credit scoring was chosen from the UCI Machine Learning Repository. The choice fell upon a German credit data set, where each client is characterised by a set of 20 attributes, followed by the classification of each customer. This data set has two versions: one that contains categorical/symbolic attributes and another where these attributes were all transformed into numerical attributes. For simplicity and opportunity, the data set chosen to develop the study presented in this paper was the one with numerical attributes.

The data set itself is a combination of personal, social and financial information about past bank clients. The complete list of attributes is shown in Table 1.

**Table 1** Data set attributes

<i>Number</i>	<i>Attribute</i>	<i>Description</i>
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	Instalment rate	Instalment 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	Instalment plans	Other instalment plans

**Table 1** Data set attributes (continued)

<i>Number</i>	<i>Attribute</i>	<i>Description</i>
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

#### 4 Classification algorithms

To analyse the data in the data set to build a classification algorithm, some initial tests were conducted using simple classifiers from the Weka Toolkit. The data set 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 Table 2.

**Table 2** Initial classification model's study

<i>Algorithm</i>	<i>Correct evaluation (%)</i>	<i>Error (%)</i>
J48	77.6	22.4
OneR	72.4	27.6
Multilayer Perceptron	73.5	26.5
Naïve Bayes	75.6	24.4

From these results, it is observed that the rate of accuracy is in the range of 70–80%. 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 the capabilities to update the classification model without the need to reevaluate 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.

##### 4.1 Multilayer perceptron

The *Multilayer Perceptron* is an algorithm that uses a feed forward neural network with backpropagation to classify instances.

In this network, a variable number of hidden layers can be used with a different number of neurons. Each neuron has a weight assigned to him/her and uses also a non-linear 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 functions, which is also used in this algorithm. Another interesting property of this type of neural network is that there no connections between neurons in the same

layer; however, neurons are fully connected between layers and it often uses more than three layers in the network.

The backpropagation learning algorithm changes the weights in each neuron after each instance of a data set is processed based on the amount of error in the output compared with the expected result. We represent the error in output node  $j$  in the  $n$ th data point by  $e_j(n) = d_j(n) - y_j(n)$ , where  $d$  is the target value and  $y$  is the value produced by the *Multilayer Perceptron*. We then make corrections to the weights of the nodes based on those corrections that minimise the error in the entire output, given by the equation (1).

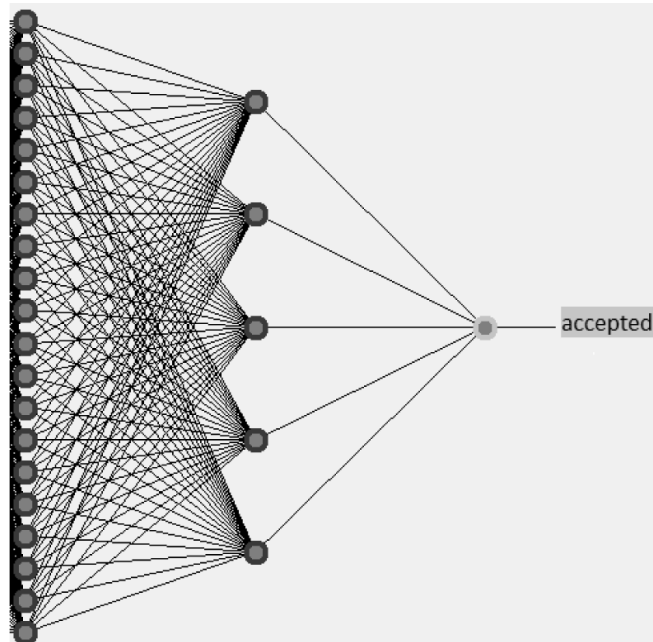
$$\varepsilon(n) = \frac{1}{2} \sum_j e_j^2(n). \quad (1)$$

Next, using a method of the gradient descent it is found the change in each neuron weight to be the value from equation (2) where  $y_i$  is the output of the previous neuron and  $\eta$  is the learning rate, which is carefully selected to ensure that the weights converge to a response fast enough, without producing oscillations.

$$\Delta w_{ji}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_i(n). \quad (2)$$

In this paper, the *Multilayer Perceptron* used contains an input layer with one neuron for each client attribute, a hidden layer composed of 5 neurons and an output layer with one neuron responsible for delivering the client classification. Figure 1 illustrates the internal structure of the neural network used with the data set from Section 3.

**Figure 1** Example of the internal constitution of a neural network



## 4.2 Decision trees

One problem that neural networks face is the fact that they consider each attribute as equal, and do not consider attributes of a given data set that may be more relevant for the decision model than others. In this section is discussed a feature selection algorithm based on C4.5 decision trees.

Decision trees are popular methods, robust for noisy data and capable of learning disjunctive expressions. These decision trees for each of the internal nodes specify a test on some attribute from the input data set. 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 data set. 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 attribute. The process is recursive and ends when the sample data in a node is all of the same attribute class or when there is no more attributes to divide the sample data by.

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 described earlier, 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.

## 4.3 Feature selection algorithm

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

The assumption that serves as a 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. From this information, two feature selection algorithms are proposed. Both of them use the *J48* classifier from the Weka Toolkit with a confidence factor of 0.25 to produce a decision tree from the data set. Then, the first algorithm chooses all the attributes presented in such decision tree as important and delivers the set. Not all attributes from a data set may be presented in a decision tree and those that are not considered as less important in the process of classifying instances.

The second algorithm aims to get a reduced list of the most relevant set of attributes for decision making in a data set. Those are placed in the upper levels of the decision tree. In this case, all attributes presented in the first three levels of a decision tree are selected and returned as the most important algorithms.



#### 4.4 *Neural networks with feature selection*

From the above-mentioned work, some approaches are now considered to implement feature selection upon neural networks making them more aware of relevant attributes to whom special consideration should be given. The internal structure of each Multilayer Perceptron applied in this section has the same components detailed in Section 4.1.

To accomplish feature selection upon the neural network, two approaches will be considered.

The first approach uses the selection algorithm in Section 4.3 that uses all attributes in a decision tree described earlier. Then, the data is filtered and those attributes not featuring in the feature selection set are eliminated from the data set. With the new data set, we present it to the neural network, more precisely a Multilayer Perceptron and train the network with the modified and normalised data set. The algorithm was named “Multilayer Perceptron with Feature Selection 1”.

The second approach uses the second feature selection algorithm presented in Section 4.3. With the given attributes from the feature selection algorithm, now a special normalisation of the data set is performed. The attributes indicated from the feature selection are normalised within a range from 0 to 2 and all the other attributes are normalised within a range from 0 to 1. Neural networks are very sensible to the input data and normalising the data set in different ways will lead the network to pay more attention to the values with greater amplitude. The algorithm was named “Multilayer Perceptron with Feature Selection 2”. In the next section, we will see that this last approach yields good results when compared with other alternatives and inclusive of the simple Multilayer Perceptron algorithm.

#### 4.5 *Results from neural networks with feature selection*

With the German data set used for this project, a number of tests were made using the algorithms detailed earlier. In Table 3, it is presented a short summary of the results in terms of correct predictions. All tests were made using the data set described in Section 3 and a test split of 66% for training data and 33% to evaluate the behaviour of each algorithm.

**Table 3** Results of the proposed algorithms

<i>Algorithm</i>	<i>Accuracy (%)</i>	<i>Error (%)</i>
Multilayer Perceptron simple	73.5	26.5
Multilayer Perceptron with Feature Selection 1	69.7	30.3
Multilayer Perceptron with Feature Selection 2	76.0	24.0

With these initial results, we have the normal Multilayer Perceptron algorithm as a base of comparison between this method and the others proposed.

Multilayer Perceptron with Feature Selection 1 represents the first algorithm that combines the first feature selection algorithm proposed with the first evolution of the Multilayer Perceptron algorithm proposed in Section 4.4. The test shows a decrease in the accuracy of the neural network. This can be explained with the loss of information introduced by the combination of the feature algorithm in the data set.

From this result, it is fair to conclude that reducing the data set may not improve the client classification.

The second approach, Multilayer Perceptron with Feature Selection 2, represents the second algorithm proposed in Section 4.4 that combines the second feature selection algorithm with the Multilayer Perceptron. Here, we see an improvement in the accuracy of the neural network. The larger range in the selected attributes induces, in the neural network, a special attention to such attributes in relation to others leading to better results than the simple multilayer algorithm. The Multilayer Perceptron with Feature Selection 2 algorithm also performs almost like the Naïve Bayes, in terms of accuracy.

In Table 4, we see the behaviour of the Naïve Bayes and Multilayer Perceptron with Feature Selection 2 algorithm proposed in Section 4.4. Here, the tests conducted also used the German credit data set where all instances of the data set were used for training and evaluation of each algorithm.

**Table 4** Results with Multilayer Perceptron with Feature Selection 2, J48 and Naïve Bayes

<i>Algorithm</i>	<i>Accuracy (%)</i>	<i>Error (%)</i>
Naïve Bayes	75.6	24.6
J48	90.2	9.8
Multilayer Perceptron with Feature Selection 2	97.2	2.8

As the test shows, when a case that was initially handled in the financial institution and given for learning the second algorithm proposed in Section 4.4 shows a better performance than Naïve Bayes and the J48 algorithms. This leads to the conclusion that the proposed algorithm retains information better and when presented with the same cases it outperforms other algorithms.

In the application designed, the second algorithm proposed in Section 4.4 shall be used to classify clients and their loan applications.

#### 4.6 Result comparison with previous work on the data set

As stated in Section 3, the data set chosen has already been used in other papers. Unfortunately, the algorithms produced by Huang et al. (2005) and Eggermont et al. (2004) use metrics of evaluation not directly comparable with the metrics used in this paper.

In Table 5, we present a comparison between the results achieved in this paper with other algorithms.

**Table 5** Results with Multilayer Perceptron with Feature Selection 2, Algorithm, J48 and Naïve Bayes

<i>Algorithm</i>	<i>Accuracy (%)</i>	<i>Error (%)</i>
a Combining feature selection and neural networks for solving classification problems	76	24
b Selective Bayesian classifier	75	25
c Multilayer Perceptron with Feature Selection 2	76	24

The results use 66% of the data presented in the data set for training the classifiers and the 33% left to evaluate the answers given by each classifier. In Table 5, we compare the algorithm

- a Combining Feature Selection and Neural Networks for Solving Classification Problems presented by O'Dea et al. (2001)
- b Selective Bayesian Classifier algorithm presented by Ratanamahatana and Gunopulos (2002) and the last algorithm presented in this paper in Section 4.4
- c Multilayer Perceptron with Feature Selection 2.

The first algorithm (a) uses a neural network with a reduced version of the data set. The reduced data set comes from attribute selection based on information theory. The second algorithm (b) also uses a reduced version of the data set 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 that are considered more relevant. The last algorithm, (c) *Multilayer Perceptron with Feature Selection 2*, uses the full data set, however, feature selection is used to distinguish important attributes in the neural network.

As the results show, the rate of accuracy achieved by the algorithm presented in this paper is similar to the rate of accuracy achieved by other authors in their attempt to improve other classifiers.

## 5 Suggestion to clients

### 5.1 Study case

After deciding upon a classification algorithm, it was felt that pure classification of client might be improved if a suggestion mechanism exists in the system. A client, before applying for a loan, normally uses a simulator or a conversation with a consultant from a financial institution. In this process, the client may only be interested in evaluating his or her chances of being granted a loan from that financial institution.

In this context, a suggestion model might be useful to the client and may also help the financial institution to advise his or her client in the better set of actions he or she can take for improving his or her chances of being granted with the desired loan. Let us imagine 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 or her problem. He or She would give the system an incomplete set of information of a predetermined set of attributes he or she cannot change and the system would calculate how changes in the not specified attributes would increase his or her chances to be granted with the loan. These changes might be increasing is credit amount available in the financial in a different account, reducing the amount of the loan by a percentage or, even change the instalment plans. With this new information, the client could be advised about these new potential situations that allow him or her to be granted with the desired loan.

## 5.2 Proposed algorithm

The classification model used in the suggestive algorithm is Multilayer Perceptron with Feature Selection 2. As it is derived from a neural network, the process of building a suggestive mechanism becomes more difficult, since neural networks do not provide any explanation on the results given. 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. The algorithm used to search such responses is a set of steps explained here:

- select each missing client attribute as a gene in a chromosome
- if not created, randomly create the initial population of chromosomes; otherwise, select the best clients from the set generated earlier
- perform the selection operator and select pairs of chromosomes
- in selected pairs of chromosomes perform the crossover operator by calculating a split point to exchange genes between each pair of chromosomes
- perform the mutation operator and assign a random value to one gene in selected chromosomes
- join the gene information with the known immutable client attributes and use the multilayer perception with feature selection 2 as the objective function
- if the maximum time of calculation is 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. 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. The classification algorithm used in this algorithm, Multilayer Perceptron with Feature Selection 2, is supposed to be already trained and to have an initial filter that normalises the client set of attributes according to the rules created in the training step of the classification algorithm.

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.

## 5.3 Results

Some interesting results came to light when investigating the properties of certain types of clients. Simulating an unemployed person, who wants a loan for a new car valued up to 50,000 €, it could be seen that, in a certain set of conditions, a loan application could be accepted by the decision mechanism. These conditions must be, according to the

suggestive system, a person with up to 38-years old, with a rented house, where the loan only takes up to 27% of his or her unemployment allowance, no bad history in previous credits, full payback in up to 74 months, with savings or property in his or her name and with no liable people. This proves the utility of this suggestive system, in the client perspective. From the financial institution perspective, it was also possible to understand from the attributes present in the accepted simulations what attributes are more important to accept a loan application from an unemployed person with low risk for the financial institution. Those attribute values are identified through their repetition in the set of client attributes generated by the suggestive algorithm. Although the client attributes generated are different from each other, some value of some attributes might not be and here is the information needed to help understand the decision process applied by the classification algorithm. For instance, the suggestive system shows that only people with no bad history and with savings or property in their name are fixed while other may vary.

This simulation proves the usefulness of the developed algorithm and provides answers presented in the initial case study.

## 6 System configuration

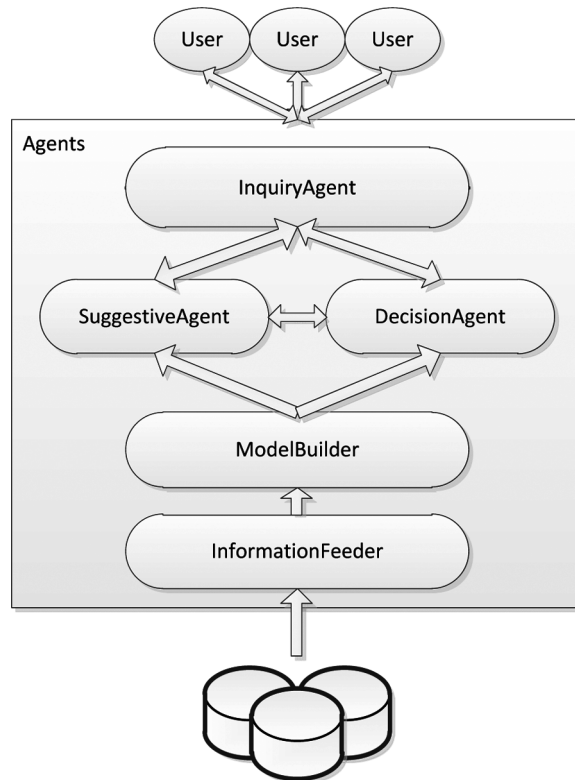
In this section, we use the algorithms described earlier in this paper to build an agent-based application to assess clients in a financial institution. This system uses an agent platform where different agents are responsible for different tasks in the system. In this system, a total of 5 different types of agents, presented in Figure 2, were created:

- *InformationFeeder*: An agent that must gather new data and pass it into the *modelbuilder* agent that builds the decision model.
- *ModelBuilder*: An agent that, based on the Multilayer Perceptron with Feature Selection 2 classification algorithm explained in Section 4.4, builds the decision model with information received from the *InformationFeeder*. After building the model, it sends it to the *DecisionAgent* and to the *SuggestiveAgent*, so they can give their answers based on the new model.
- *DecisionAgent*: An agent that must take the model built by the agent *ModelBuilder* and respond to customers enquires on complete loan applications based on its decision model.
- *SuggestiveAgent*: An agent that, with a model received from the *ModelBuilder* agent and some incomplete information from the customer, suggests him/her a series of favourable combinations to be successful in a loan application.
- *InquiryAgent*: An agent that makes requests to the agents *DecisionAgent* and *SuggestiveAgent* and delivers the responses to the clients.

With these five different types of agents, the system builds and acts in a predefined way. An *InformationFeeder* is used to import a series of data into the system. Whenever new data becomes available to take into consideration, this agent is created and it looks for agent of type *ModelBuilder* and delivers the data to him. The *ModelBuilder* agent upon receiving the new data updates its decision algorithm. If it is the first time some

information is given to him/her, then a feature selection algorithm is used to determine the most relevant attributes and, then those attributes are normalised in the same order as the Multilayer Perceptron with Feature Selection 2 algorithm proposed in Section 4.4. If the model already had evaluated a set of data, then the new data must be normalised according to rules created the first time the model was created and the neural network should then be retrained.

**Figure 2** Multi-agent system



When a new classification model is available, the *DecisionAgent* updates its decision model with the data and normalising rules for the input data. This agent then waits for enquiries from the *InquiryAgent* agent and delivers the responses to him/her.

In a similar manner, the *SuggestiveAgent* updates its classification model and also waits for enquiries from the *InquiryAgent* to deliver the set of suggestions based on an input to the client.

Finally, the *InquiryAgent* is responsible for receiving the requests from a user and redirecting them to the appropriate agent: the *SuggestiveAgent* or *DecisionAgent*. It then waits for the response and delivers it to the client. Each time a new client is introduced in the system an *InquiryAgent* is created and, as a consequence, more than one of these agents may exist in the system at the same time. This is a basic assumption within a multi-agent system, as the one presented here.

## 7 Conclusions and future work

The algorithms described in this paper provided good results in the classification of clients for loan application in some financial institution. The proposed classification algorithm showed improvements when compared with his standard version. The suggestive algorithm also produced good results evaluating alternatives to client situations. The final system architecture proposed to integrate each individual algorithm in a multi-agent system poses as an example of how these algorithms can be used in real-world applications

Different data sets could be used to train the classifier. Also, different classifiers could be improved to have a more comprehensive list to compare performances between each algorithm and some more work could be done allowing different *ModelBuilder* agents that induce different decision models at the same time.

## References

- Alcalá-Fdez, J., Sánchez, L., García, S., Jesús, M.J., Ventura, S., Guiu, J.M.G., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C. and Herrera, F. (2009) 'KEEL: a software tool to assess evolutionary algorithms for data mining problems', *Soft Comput.*, Vol. 13, No. 3, pp.307–318.
- Bhaduri, A. (2009) 'Credit scoring using artificial immune system algorithms: a comparative study', *World Congress on Nature Biologically Inspired Computing, 2009, NaBIC 2009*, Coimbatore, India, pp.1540–1543.
- Eggermont, J., Kok, J.N. and Kusters, W.A. (2004) 'Genetic programming for data classification: partitioning the search space', *Proceedings of the 2004 Symposium on Applied Computing*, Nicosia, Cyprus, pp.1001–1005.
- Eletter, S.F., Yaseen, S.G. and Elrefae, G.A. (2010) 'Neuro-based artificial intelligence model for loan decisions', *American Journal of Economics and Business Administration*, Vol. 2, No. 1, pp.27–34.
- Faiz, R.B. and Edirisinghe, E.A. (2009) 'Decision making for predictive maintenance in asset information management', *Interdisciplinary Journal of Information, Knowledge, and Management*, Vol. 4, pp.23–36.
- Heaton, J. (2010) *Programming Neural Networks with Encog 2 in Java*, Heaton Research Inc., Chesterfield, UK.
- Higson, C. and Waltho, D. (2009) *Valuing Information as an Asset*, White Paper, SAS Institute, UK.
- Huang, J-J., Tzeng, G-H. and Ong, C-S. (2005) 'Two-stage genetic programming (2SGP) for the credit scoring model', *Applied Mathematics and Computation*, Vol. 174, No. 2, March, pp.1039–1053.
- Islam, Md.S., Zhou, L. and Li, F. (2009) *Application of Artificial Intelligence (Artificial Neural Network) to Assess Credit Risk: A Predictive Model for Credit Card Scoring*, MSc Dissertation, School of Management, Blekinge Institute of Technology, Sweden.
- Khatri, V. and Brown, C.V. (2010) 'Designing data governance', *Communications of the ACM*, Vol. 53, pp.148–152.
- Madeira, S., Oliveira, A. and Conceição, C. (2003) 'A data mining approach to credit risk evaluation and behaviour scoring', *Progress in Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 2902, Springer, Berlin/Heidelberg, Beja, Portugal, pp.184–188.

- Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006) 'Yale: rapid prototyping for complex data mining tasks', *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '06*, ACM, New York, NY, USA, pp.935–940.
- O'Dea, P., Griffith, J., O'Riordan, C., Griffith, J. and Riordan, C.O. (2001) 'Combining feature selection and neural networks for solving classification problems', *Proc. 12th Irish Conf. Artificial Intell. Cognitive Sci.*, pp.157–166.
- Ratanamahatana, C.A. and Gunopulos, D. (2002) 'Selective Bayesian classifier: feature selection for the Naive Bayesian classifier using decision trees', *Proceedings of the 3rd International Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields*, September, Bologna, Italy, pp.613–623.
- Simic, D. and Simic, S. (2007) 'An approach to efficient business intelligent system for financial prediction', *Soft Computing – A Fusion of Foundations, Methodologies and Applications*, Vol. 11, pp.1185–1192.
- Vojtek, M. and Kocenda, E. (2006) 'Credit-scoring methods (in English)', *Czech Journal of Economics and Finance (Finance a uver)*, Vol. 56, Nos. 3–4, March, pp.157–167.
- Witten, I.H. and Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, California, USA.