# Using Neural Network Rule Extraction for Credit-Risk Evaluation

*Maria Teresinha Arns Steiner*; Pedro José Steiner Neto**; Nei Yoshihiro Soma***; Tamio Shimizu+ and Júlio Cesar Nievola++*

*UFPR – *Mathematics; **Business Departments*
*CP: 19081; CEP: 81531-990, Curitiba, Paraná, Brazil*
****ITA – Computer Sciences Division , Pça. Mal. Eduardo Gomes, 50, Vl. das Acácias*
*CEP: 12228-990, São José dos Campos, São Paulo, Brazil*
*+USP – Engineering Production , São Paulo, São Paulo, Brazil*
*++PUC-PR – Applied Informatics Graduate Program*
*Av. Imaculada Conceição, 1155, CEP 80215-901, Curitiba, Paraná, Brazil*

**Summary**
Credit-risk evaluation is a very important management science problem in the financial analysis area. Neural Networks have received a lot of attention because of their universal approximation property. They have a high prediction accuracy rate, but it is not easy to understand how they reach their decisions. In this paper, we present a real-life credit-risk data set and analyze it by using the *NeuroRule* extraction technique and the *WEKA* software. The results were considered very satisfactory.
*Keywords:*
*Neural Networks, NeuroRule extraction technique, Credit-risk.*

## 1. Introduction

As technology advances, it has not been difficult for large companies to efficiently store large volumes of data (historical records) in their computers to recover them whenever necessary. Many of these companies, however, have faced the problem of having lots of data, but little knowledge - data rich but knowledge poor [8].
Making the correct decision, i.e., granting (or not) bank credit, is essential for banks' survival. Many times, losses caused by a mistake when making a decision on granting credit to a single client can put in jeopardy profits obtained with many well-succeeded operations [16].
Tools that may help decision-making have been used, mainly by researchers in the Artificial Intelligence (AI) area. Two basic approaches for classification problems (which is the case of the credit problem presented in this paper) studied by AI researchers are the symbolic approach (based on decision trees) and the connectionist approach (based mainly on Neural Networks, NN).
Regarding credit grants, some of the advantages obtained through the correct use of tools for decision-making are, among others: involving less people in credit analysis, thus releasing them to perform other activities, agility when processing credit requests, smaller subjectivity along the decision-making procedures and greater result accuracy, i.e., a smaller percentage of mistakes.
Data Mining is a new technology that is used to increase decision quality and efficiency. Several companies, such as banks, for instance, have obtained a high return over investment by making use of database analysis tools.
The purpose in this work is to use, among the several different Data Mining techniques from the KDD (Knowledge Discovery in Databases) context, those tools capable of classifying companies (legal entities) as “good” or “bad” credit takers, based upon the historical records those institutions have stored. From the several classification Data Mining techniques, we made an option for the technique of extracting classification rules from a trained NN to evaluate credit risk and doing this by coding variables (inputs, attributes) as "thermometer" and "dummy" [1], thus making them binary.
The classification rules' accuracy thus obtained in this work is compared with: 1) rule extraction, directly from the original data (patterns); 2) rule extraction, directly from the original data, discarding some of them, as explained in Section 5, in this paper; 3) rule extraction, directly from the original data, discarding some of them, and also using the aforementioned attribute coding.
The purpose of these comparisons is to check which one of the four alternatives supplies classification rules with a greater accuracy rate in the classification task, the importance level of coding attributes prior to training the NN, as well as the importance level of training the NN prior to extracting rules.
In Section 2 the real problem is described along with the presentation of the data and attributes used in the experiments (simulations); in Section 3, Data Mining and KDD are discussed, as well as some correlated works; in Section 4, the *NeuroRule* algorithm is applied to a Multi-layer NN; in Section 5, the four simulations aforementioned are developed and, finally, in Section 6, we present the conclusions.

## 2. Description of the Real Problem

The data used in this work were obtained from a large Brazilian bank [7] and refer to credit for legal entities (companies). This bank makes credit lines available to its clients, for working capital, as well as for investments. Mainly micro, small and medium sized companies make up the bank's body of clients.

By the time this research was made, the bank used an internal application called Credit Analysis (CAN) as a tool to carry out credit analysis. It was from this application, which contained the companies' file and accounting information, that managers sought for support for making decisions with respect to granting (or not) bank credit.

To develop this work we used historical data from a total of 339 clients – legal entities –, from which 266 were provenly Good Credit Payers and 73 were Bad Credit Payers. From each one of these clients 24 different attributes were extracted, chosen by specialists in the area (the bank's credit managers) and specified in Table A herein attached, with their respective original values (columns 1 and 2) and, added to column 1, their type, which can be ordinal or nominal in this problem. In columns 3 and 4 from Table A the value intervals are established for each one of the attributes, as well as the number of patterns contained in each interval, respectively. In columns 5 to 8 is the "thermometer" or "dummy" coding, which is explained in Section 3, for each one of the attributes. Finally, in column 9 the quantity of inputs for the NN that was used is accounted. This is obviously a value that depends on the coding that is used.

## 3. KDD and Data Mining

The aim of the broad area or process called KDD are those techniques and tools that try to transform into knowledge the data stored by companies. The KDD process is a set of continuous activities that share the knowledge that is discovered in databases. According to [4] five steps compose such set: data selection, data pre-processing and cleaning, data transformation, Data Mining, and result interpretation and evaluation. The interconnection between these steps can be seen in Figure 1.

The KDD process starts by understanding the problem's domain and the final objectives to be reached. The available data is arranged into an organized group, the search's target. The data-cleaning step comes next, by means of data pre-processing, integrating heterogeneous data, eliminating incomplete data and others. This step can take up to 80% of the time needed for the whole process, due to the well-known difficulties of integrating heterogeneous databases [10].

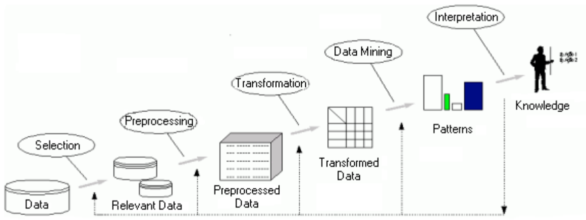


Figure 1. Activities that compose the KDD process [4]

The pre-processed data may also go through a transformation that stores them adequately. During this step the use of data storage (data warehouse) is considerably extended, because with this technology information can be stored more efficiently. All these steps, which occur prior to the Data Mining step, can be seen as a Data Exploratory Analysis [17]. This analysis may also involve, among other alternatives, data standardization, as well as discarding atypical data. In this work, these preliminary steps involved basically data selection, cleaning and coding (in two of the four simulations).

Then comes the Data Mining step, which starts by choosing the algorithms that will be used. This choice depends fundamentally on the KDD process' objective [18], which may be: classification, grouping or association. In general, the algorithms used during the Data Mining step look for patterns in the data.

Several distinct tools, such as NNs, decision trees, systems based on rules, statistical programs and others, isolated or combined one with each other, can then be applied to the problem. In general, search processing is interactive so as to allow analysts to review results, form a new set of questions in order to refine searches with respect to a certain aspect of results, and feed the system back with new parameters. By the end of the process, a discovery report is produced, which is then interpreted by the mining analysts and the knowledge is discovered. Data Mining is the most interesting part in the KDD process, and in the business context it is the one that mostly levers and helps businessmen to find market niches.

According to [6], the knowledge to be discovered must be correct, understandable and useful. Besides, the discovery method must be efficient, generic (applicable to several types of problems) and flexible (can be easily modified).

Among the Data Mining techniques that are used in classification problems – and this is the case of the problem discussed in this paper – we point out the NNs: they build internal representations of models or patterns they detect in the data, but these representations are not presented explicitly to the users. In this work the extraction of classification rules is made by means of the coded attributes and the trained NN, with the purpose of making clear and comprehensible to the user (credit managers) how the attributes are "acting" to perform the classification of each one of the clients.

Among the numerous works that discuss Data Mining techniques for classification, we can mention: Lu *et al.* [8] and Lu *et al.* [9], who present the algorithm called *NeuroRule* (used in this work), which performs the extraction of rules from a trained NN and obtains rules of the IF-THEN type. In both papers the performance with this approach was checked on a bank credit problem. Fidelis *et al.* [5] present a classification algorithm, which is based on Genetic Algorithms (GAs) and discovers comprehensible rules of the IF-THEN type in the Data Mining context. The proposal was evaluated in two public dominion medical databases: dermatology and breast cancer.

Setiono & Leow [15] present a Fast Method for Extracting Rules from Trained Neural Networks (*FERNN*): firstly, *FERNN* identifies the relevant hidden units using the *C4.5* algorithm [13]. Then, for each relevant hidden unit *FERNN* finds the set of relevant connections from the inputs to the hidden units and, finally, *FERNN* substitutes the decision tree's divided conditions generated by the *C4.5* by rules that involve the network's inputs. Santos *et al.* [14] use a GA to define a topology that is adequate for a NN that will be trained. The proposed system was evaluated on three data sets available in UCI's repository: Iris, Wine and Monks-2.

Baesens *et al.* [1] discuss three methods for comparatively extracting rules from a NN: *NeuroRule*, *Trepan* and *Nefclass*. These were applied to three real credit databases: German Credit, Bene1 and Bene2. Olden & Jackson [12] describe some methods from the literature to "unveil" the mechanisms of a NN. For the evaluation of real estate, one can mention the work of Nguyen & Cripps [11], who compare a NN's predictive performance with the Multiple Regression Analysis for selling residential houses. In their work, Bond *et al.* [2] examine the effect a view to a lake has over the value a house may have.

## 4. The *NeuroRule* Algorithm and Attribute Coding

Among the many existing NN models, we decided to use a Multiple Layer NN [3] that was trained with the back-propagation algorithm (supervised learning).

As mentioned above, the purpose in this work is to extract classification rules from the coded attributes and the trained NN, and for this, we used the *NeuroRule* algorithm, which is describe below – steps 1 to 4 [8].

The *NeuroRule* Algorithm's steps 2 and 3 need perfect rules generated by a trained NN. To generate these rules we used the *WEKA* software (Waikato Environment for Knowledge Analysis), which is available in the World Wide Web (www.cs.waikato.ac.nz/ml/weka). This software contains 10 algorithms for extracting classification rules, namely: JRip, ZeroR, Ridor, Prisma, M5Rules, Part, OneR, Nnge, Decision Table and finally, Conjunctive Rule, all of them described in [18].

### 4.1 The *NeuroRule* Algorithm for Extracting Rules (*NeuroRule* Extraction)

Step 1. By means of clustering, turn the activation values into discrete ones:

1a. Let $\varepsilon \in (0, 1)$. Let $D$ the number of discrete activation values in the hidden layer. Let $\delta_1$ the activation value in the hidden layer for the training set's first pattern. Let $H(1) = \delta_1$, counter(1) = 1, sum(1) = $\delta_1$ and make $D = 1$.

1b. For all patterns $i = 2, 3, ..., k$ in the training set:

Let $\delta$ the activation value.

If there is an index $j'$, such that:

$|\delta - H(j')| = \min_{j \in \{1,2,...,D\}} |\delta - H(j)|$ and $|\delta - H(j')| \leq \varepsilon$,

then make counter($j'$) = counter($j'$) + 1, sum($D$) = sum($D$) + $\delta$

else $D = D + 1$, $H(D) = \delta$, counter($D$) = 1, sum($D$) = $\delta$.

1c. Substitute $H$ for the average value of all activation values that were grouped into this group:

$H(j)$ = sum($j$) / counter($j$), $j = 1, 2, ..., D$.

1d. Check the NN's accuracy with the activation values $\delta^i$ in the hidden nodes replaced by $\delta^d$, the activation value of the cluster to which the activation value belongs.

1e. If the accuracy is smaller than the needed value, decrease $\varepsilon$ and repeat step 1.

Step 2. List the activation values that were made discrete and calculate the network's output.

Generate perfect rules that have a perfect covering of all the examples from the hidden nodes' activation values to the output values.

Step 3. For the discretized hidden node activation values appeared in the rules found in step 2, list the input values that lead to them and generate perfect rules.

Step 4. Generate rules that relate the input values to the output values by rule substitution based on the results of steps 2 and 3 above.

### 4.2 Attribute Coding

The *NeuroRule* algorithm used in this work to extract rules from a trained NN assumes that the data are discrete and are represented as binary inputs using a "thermometer coding" for the ordinal attributes and a "dummy coding" for the nominal attributes [1].

Table 1 illustrates the "thermometer coding" for the "income" ordinal variable, for instance. Firstly the attribute "income" is made discrete with the values 1, 2, 3 or 4. If, for instance, $I_3 = 1$, this means that the original variable "income" > 1,000. In this work, the operation of making attributes discrete was carried out with the help of a specialist.

**Table 1.** An example of the "thermometer coding" procedure for ordinal variables.

| Original input Income (R$) | Categorical Input | $I_1$ | $I_2$ | $I_3$ |
|---|---|---|---|---|
| Income ≤ 1,000 | 1 | 0 | 0 | 0 |
| 1,000 < Income ≤ 2,000 | 2 | 0 | 0 | 1 |
| 2,000 < Income ≤ 3,000 | 3 | 0 | 1 | 1 |
| Income > 3,000 | 4 | 1 | 1 | 1 |

Table 2 illustrates the "dummy coding" for the variable "loan purpose", for instance. This coding scheme makes it easy to generate and interpret IF-THEN rules.

**Table 2.** An example of the "dummy coding" procedure for nominal variables.

| Original input Purpose | $I_1$ | $I_2$ |
|---|---|---|
| Purpose = car | 0 | 0 |
| Purpose = real estate | 0 | 1 |
| Purpose = other | 1 | 0 |

## 5. Implementation and Results

In this work, four simulations with the data were developed with the purpose of obtaining classification rules, as described in 5.1 to 5.4, below. The test mode in all four simulations was the *WEKA* software's standard method, i.e., 10-fold cross-validation [18].

### 5.1 First Simulation: obtaining classification rules directly from the *WEKA* software, considering the credit problem's original data

In this 1st simulation, the classification rules were obtained directly from the problem's original data, without considering the trained NN. This 1st simulation was performed in order to verify the importance (or not) of codifying attributes, as well as of training a NN prior to extracting rules.

For this 1st simulation the problem's original data were considered, i.e., 339 patterns (legal entities), from which 266 belong to set **A** (Good Credit Payers, answer=1) and 73 belong to set **B** (Bad Credit Payers, answer=0). For each one of these patterns there are 24 attributes, shown in Table A.

Among the 10 methods for obtaining classification rules that exist in the *WEKA* software, the **JRIP** method was the one that presented the greatest accuracy rate. The results that were obtained are in Table 3, below.

From Table 3, 1st rule, we have that 33 patterns were classified as Bad Credit Payer, 11 of them erroneously, i.e., from 33 Bad Credit Payer patterns, 11 were, in fact, Good Credit Payers. The same way, according to the 2nd rule, 306 patterns were classified as Good Credit Payers, from which 51 erroneously. These values can be represented by the confusion matrix presented in Table 4. Therefore, we have that the accuracy for this 1st simulation was of: 1-[(11 + 51) / 339] = 81.71%.

**Table 3.** Results for the 1st Simulation: Classification Rules obtained directly from the original data, with the help of the *WEKA* software

| Rules | | Result |
|---|---|---|
| IF [(client = new) AND (number of employees ≥ 0) AND (gross annual turnover ≥ 54,000)] | then | Bad Credit Payer (33/11) |
| ELSE | then | Good Credit Payer (306/51) |

**Table 4.** Confusion Matrix for the 1st Simulation**.**

| Real/Classification | Good Credit Payer | Bad Credit Payer |
|---|---|---|
| **Good Credit Payer** | 306 | 51 |
| **Bad Credit Payer** | 11 | 33 |

#### 5.1.1 Attribute Coding and Neural Network Training

Once the 1st simulation was finished, each one of the 24 attributes was transformed according to the "thermometer" or "dummy" coding, thus making them binary, as can be seen in Table A, this way obtaining 54 attributes. Considering these 54 attributes, a Multi-Layer NN (three layers) was trained with the back-propagation algorithm. After testing several topologies, each one of them with different initial weights, the best results were obtained with a topology that contained 4 neurons in the hidden layer. This topology, with 54 inputs, 4 neurons in the hidden layer and 1 neuron in the output layer (Good Credit Payer or Bad Credit Payer), (54 – 4 – 1), classified erroneously only 13 of the 339 patterns, i.e., this NN's accuracy was of 1 - [13 / 339] = 96.17%.

To extract "perfect" classification rules from this trained NN (54 – 4 – 1), the 13 patterns the NN classified erroneously were removed from the data sample.

### 5.2 Second Simulation: obtaining the classification rules directly from the WEKA software, considering the credit problem's original data and excluding the 13 patterns mentioned above

For this 2nd simulation 326 original patterns were considered (13 patterns were excluded), from which 256 belong to set **A** and 70 to set **B**.

Among the methods for obtaining classification rules that exist in the *WEKA* software, the **JRIP** method was again the one that presented the best results, shown in Table 5 below. The accuracy for this 2nd simulation was of: 1 - [(48 + 9) / 326] = 82.51%.

**Table 5.** Results for the 2nd Simulation: Classification Rules obtained directly from the original data, with the exclusion of 13 patterns and with the help of the *WEKA* software

| **Rules** | | **Result** |
|---|---|---|
| IF [(age of account in bank ≤ 8 months) AND (number of employees ≥ 1) AND (gross annual turnover ≥ 75,000)] | then | Bad Credit Payer (31/9) |
| ELSE | then | Good Credit Payer (295/48) |

## 5.3 Third Simulation: obtaining the classification rules directly from the *WEKA* software, considering the credit problem's original data with coded attributes and excluding 13 patterns

For this 3rd simulation 326 original patterns were considered, from which 256 belong to set **A** and 70 to set **B**. For each one of these patterns there are 54 attributes transformed into binary by coding "thermometer" or "dummy", according to Sub-section 4.2 and Table A.

Among the methods contained in the *WEKA* software, the **JRIP** method once more presented the best results, which are in Table 6 below. The accuracy for this 3rd simulation was of:

1 - [(1 + 12 + 40) / 326] = 83.74%.

**Table 6.** Results for the 3rd Simulation: Classification Rules obtained directly from the coded data, with 13 discarded patterns and the help of the *WEKA* software

| **Rules** | | **Result** |
|---|---|---|
| IF [(account age in agency ≤ 12) AND (partners own real property > 0) AND (company's premises = owned or rented) AND (district = others) | then | Bad Credit Payer (10/1) |
| IF [(account age in agency ≤ 12) AND (gross annual turnover > 60,000) AND (risk the agency ascribes = C) | then | Bad Credit Payer (33/12) |
| ELSE | then | Good Credit Payer (283/40) |

## 5.4 Fourth Simulation: obtaining the classification rules directly from the trained Multi-Layer NN, making use of the *NeuroRule* Algorithm, considering the credit problem's original data with coded attributes, excluding 13 patterns and with the help of the *WEKA* software

For this 4th simulation, once more were considered the 326 patterns, set **A** with 256 patterns and set **B** with 70 patterns. For each one of these patterns there are the same 54 attributes transformed by the coding mentioned above. Thus, the *NeuroRule* Algorithm was used to obtain the classification rules.

Initially, the activation values were clustered into each one of the four neurons in the hidden layer (*a*, *b*, *c* and *d*). For the data presented in this paper, the values obtained for *a*, *b*, *c* and *d*, for all 326 patterns, were "1" or "0".

Once the clusters in each one of the four neurons were defined, classification rules were generated from the hidden node activation values to the output values using the *WEKA* software. Next, classification rules were generated from the input values to the hidden layer using the *WEKA* software again. Finally, classification rules were generated relating the input values and the output values by rule substitution.

The situation for this 4th simulation is pictured in Figure 2, where $i_1$, $i_2$, ..., $i_{54}$ are the NN's inputs, $w_{ij}$*'s* and $w_{hi}$*'s* are the connections (weights) between the input and the hidden layers, and between the hidden and the output layers, respectively, and $\theta_{i,1}$, $i$ = 1, 2, 3 and 4, and $\theta_{h,2}$, $h=1$, are respectively the hidden and output layers' bias.

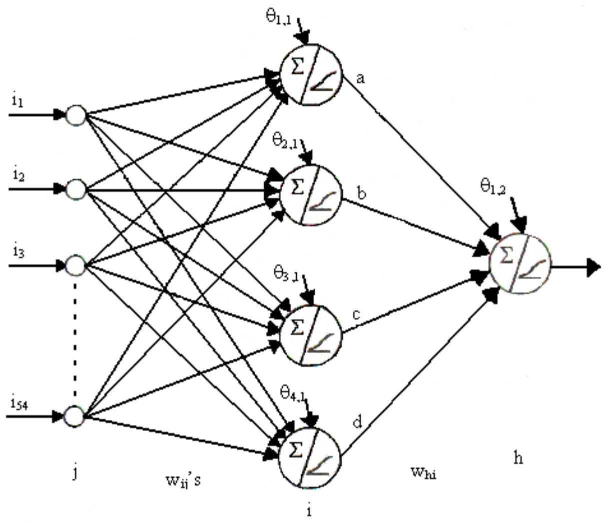


**Figure 2.** The multi-layer (3) NN (54 – 4 – 1) that, among the topologies that were tested, presented the best accuracy for the data in the problem.

Among the methods contained in the *WEKA* software, the **PART** method was the one that presented the greatest accuracy for the classification rules from the hidden layer to the output layer. The results obtained are presented in Table 7 where, as we have already mentioned, *a*, *b*, *c* and *d* are the activation values for the four neurons in the hidden layer. Accuracy was of 99,08%.

To get the classification rules from the input layer to the 1st unit (*a*) of the hidden layer, the **JRIP** method was the one that showed the best results: accuracy was of 92.24%. Those rules are in Table 8.

**Table 7**. Classification Rules from the hidden layer to the output level

| **Rules** | | **Result** |
|---|---|---|
| IF ($b$ = 1) | then | 1 = Good Credit Payer (180/1) |
| IF ($a$ = 0) AND ($d$ = 0) | then | 0 = Bad Credit Payer (66/2) |
| IF ($c$ = 0) | then | 1 = Good Credit Payer (75/0) |
| ELSE | then | 0 = Bad Credit Payer (5/0) |

**Table 8.** Classification Rules from the input layer to the 1st neuron ($a$) of the hidden layer

| **Rules** | | **Result** |
|---|---|---|
| IF [($i_{18}$ = 1) AND ($i_{31}$ = 0) AND ($i_{47}$ = 0) AND ($i_{22}$ = 1)] | then | $a$ = 1 (=Good Credit Payer) (18/1) |
| IF [($i_{18}$ = 1) AND ($i_{47}$ = 0) AND ($i_{10}$ = 0) AND ($i_{23}$ = 0) AND ($i_{49}$ = 0)] | then | $a$ = 1 (Good Credit Payer) (19/0) |
| IF [($i_{47}$ = 0) AND ($i_{18}$ = 1) AND ($i_6$ = 0) AND ($i_{32}$ = 0)] | then | $a$ = 1 (Good Credit Payer) (9/1) |
| IF [($i_{13}$ = 1) AND ($i_{46}$ = 0) AND ($i_{31}$ = 0) AND ($i_{49}$ = 0) AND ($i_{11}$ = 0)] | then | $a$ = 1 (Good Credit Payer) (6/0) |
| ELSE | then | $a$ = 0 = (Bad Credit Payer) (274/21) |

To get the classification rules from the input layer to the 2nd unit ($b$) of the hidden layer, the **JRIP** method was once more the one that showed the best results: accuracy was of 91.10%. Those rules are in Table 9.

**Table 9.** Classification Rules from the input layer to the 2nd neuron ($b$) of the hidden Layer

| **Rules** | | **Result** |
|---|---|---|
| IF [($i_5$ = 0) AND ($i_{49}$ = 0)] | then | $b$ = 0 = (Bad Credit Payer) (104/7) |
| IF [($i_4$ = 0) AND ($i_{50}$ = 0) AND ($i_{52}$ = 1) AND ($i_{47}$ = 0)] | then | $b$ = 0 = (Bad Credit Payer) (19/1) |
| IF [($i_4$ = 0) AND ($i_{24}$ = 1) AND ($i_{51}$ = 0) AND ($i_8$ = 0)] | then | $b$ = 0 = (Bad Credit Payer) (18/4) |
| ELSE | then | $b$ = 1 = (Good Credit Payer) (185/17) |

Again, the **JRIP** method showed the best results when obtaining the classification rules from the input layer to the 3rd unit ($c$) of the hidden layer: accuracy was of 99.08%. Those rules are in Table 10.

The classification rules from the input layer to the 4th unit ($d$) of the hidden layer, were obtained using the **JRIP** method, which once more was the one that showed the best accuracy: 81.90%. Those rules are in Table 11.

**Table 10.** Classification Rules from the input layer to the 3rd neuron ($c$) of the hidden layer

| **Rules** | | **Result** |
|---|---|---|
| IF [($i_{40}$ = 1) AND ($i_{18}$ = 1) AND ($i_{10}$ = 1) AND ($i_{36}$ = 0)] | then | $c$ = 1 = (Good Credit Payer) (11/2) |
| IF [($i_{32}$ = 1) AND ($i_6$ = 0) AND ($i_{31}$ = 1)] | then | $c$ = 1 = (Good Credit Payer) (3) |
| ELSE | then | $c$ = 0 = (Bad Credit Payer) (312/1) |

**Table 11.** Classification Rules from the input layer to the 4th neuron ($d$) of the hidden layer

| **Rules** | | **Result** |
|---|---|---|
| IF [($i_{17}$ = 0) AND ($i_7$ = 0)] | then | $d$ = 1 = (Good Credit Payer) (137/35) |
| IF [($i_{33}$ = 1) AND ($i_{50}$ = 0)] | then | $d$ = 1 = (Good Credit Payer) (15/3) |
| ELSE | then | $d$ = 0 = (Bad Credit Payer) (174/21) |

Finally, in order to get the classification rules from the input layer to the output layer, we "joined" the rules obtained previously, thus getting the classification rules from the input layer to the output layer. The results are presented in Table 12.

Transforming inputs $i_1$ to $i_{54}$ into their true meanings, as can be seen in Table A herein attached, we have the final classification rules for this 4th simulation, which are presented in Table 13.

**Table 12.** Classification Rules from the input layer to the output layer (coded attributes)

| **Rules (the symbol "¬" = "no"** | | **Result** |
|---|---|---|
| IF (b = 1):<br>IF {¬ [$i_5$ = 0 and $i_{49}$ = 0] OR ¬ [$i_4$ = 0 and $i_{50}$ = 0 and $i_{52}$ = 1 and $i_{47}$=0] OR<br>¬ [$i_4$ = 0 and $i_{24}$ = 1 and $i_{51}$ = 0 and $i_8$ = 0]} | then | Good Credit Payer (180/12) |
| IF [(a = 0) and (d = 0)]:<br>IF {¬ [$i_{18}$ = 1 and $i_{31}$ = 0 and $i_{47}$ = 0 and $i_{22}$ = 1] OR ¬ [$i_{18}$ = 1 and $i_{47}$=0 and $i_{10}$ = 0 and $i_{23}$ = 0 and $i_{49}$ = 0] OR<br>¬ [$i_{47}$ = 0 and $i_{18}$ = 1 and $i_6$=0 and $i_{32}$ = 0] OR<br>¬ [$i_{13}$ = 1 and $i_{46}$ = 0 and $i_{31}$ = 0 and $i_{49}$ = 0 and $i_{11}$=0]} AND {¬ [$i_{17}$ = 0 and $i_7$ = 0] OR ¬ [$i_{33}$ = 1 and $i_{50}$ = 0]} | then | Bad Credit Payer (66/21) |
| IF (c = 0):<br>IF {¬ [$i_{40}$ = 1 e $i_{18}$ = 1 e $i_{10}$ = 1 e $i_{36}$ = 0] OU ¬ [$i_{32}$ = 1 e $i_6$=0 e $i_{31}$ = 1] } | then | Good Credit Payer (75/1) |
| ELSE | then | Bad Credit Payer (5/0) |

**Table 13.** Classification Rules from input layer to the output layer (attributes with their real meanings)

| Rules | | Result |
|---|---|---|
| IF {[(age of account at BB > 12 months) and (partners own movables > 12,000)] OR [(age of account at BB > 36 months) and (partners own movables > 0) and (risk ascribed by the bank ≠ B) and (partners own real property > 0)] OR [(age of account at BB > 36 months) and (client at another bank = no) and (risk ascribed by the bank = A) and (activity sector = services)]} | then | Good Credit Payer (180/12) |
| IF {[(district = others) and (company insurance policy = yes) and (partners own real property > 0) and (gross annual turnover ≤ 180,000)] OR [(district = others) and (partners own real property > 0) and (time in activity ≥ 6 years) and (gross annual turnover > 60,000) and (partners own movables > 12,000)] OR [(partners own real property > 0) and (district = others) and (age of account at BB > 0) and (financial applications at BB > 8,000)] OR [(number of employees ≤ 10) and (partners own real property > 30,000) and (company insurance policy = yes) and (partners own movables > 12,000) and (time in activity ≥ 3 years)]} AND {[(company's premises = rented) and (activity sector = industry)] OR [(financial applications at BB ≤ 4,000) and (partners own movables > 0)]} | then | Bad Credit Payer (66/21) |
| IF {[(account history ≠ new client) and (district = others) and (time in activity < 6 years) and (sales on installment < 20%)] OR [(financial applications at BB ≤ 8,000) and (age of account at BB > 0) and (company insurance policy = no]} | then | Good Credit Payer (75/1) |
| ELSE | then | Bad Credit Payer (5/0) |

Since the main purpose on developing the present study was to find a friendly way to offer to credit analysts (loan managers) good advice on credit decision making, a tentative model for a spreadsheet with a very easy way to show all combinations for the variables and the decision to be made on each combination is presented in table 14. This table should provide loan officers not only a guide to make better decisions, but also could help them on improving their relationship with customers, by pointing out targets for a stronger relationship.

In order to be more effective, a loan officer would receive a printed version for the spreadsheet and based on the customer relationship information could easily identify the major reasons for a credit denial and suggest to the customer where to act. It could also increase the accuracy on credit decision making by bringing speed to the process, reducing the time a customer has to wait, and therefore reducing the risk of losing the operation.

In order to use the spreadsheet, the loan officer would first check whether the customer presents any of the three combinations for Rule 1, based on 9 attributes. If any combination is reached, the loan could be granted. In case a costumer does no meet any specifications for Rule 1, a second set of combinations, called Rule 2, should then be tested. If the customer presents any of the 7 combinations using a total of 16 attributes, the loan should be denied. A final set of combinations, called Rule 3, using 7 attributes should then be tested and if conditions are matched, the loan should be granted. Any other situation would result in a denial for the credit operation. It should be noted that, although Rules 1 to 3 use a total of 32 attributes, as some of them are used in more than one combination, only 15 out of the initial 24 were used to performs the test, and therefore should be considered more important.

Table 14 is the one that shows the set of rules, and could be used by any loan officer at the company. It can be seen that this table could be customized for every branch since the set of rules probably would differ. Further details about the implementation can be obtained directly with the corresponding author.

## 6. Conclusion

The purpose with this paper is to present tools that may help to identify and foresee which clients will be good credit payers (or not) in relation to credit from banks.
Although Data Mining problems involve, in general, thousands or even millions of data, different from the problem presented here (339 x 24 in the 1$^{st}$ simulation, 326 x 24 in the 2$^{nd}$ and 326 x 54 in the 3$^{rd}$ and 4$^{th}$ simulations), the conclusion arrived at can be used as support for larger problems. Besides, the ideal is that the number of patterns had been much higher than 339 (or 326), mainly for simulations 3 and 4, in which the number of attributes was of only around 1/6 (= 54) of the number of patterns. It would be interesting that this ratio (n$^{o}$ of attributes/n$^{o}$ patterns) had been of the order of 1/10 or smaller.
This paper presented a way of extracting classification rules from a problem whose attributes were coded, making them binary, and whose patterns were trained with a NN. The extraction was accomplished by applying the *NeuroRule*

algorithm (4th. simulation) and with the help of the *WEKA* software (all simulations). The accuracy of the classification rules obtained in the 4th. simulation was compared to the accuracy of other three less elaborated simulations to check the importance (or not) of coding the attributes, as well as of training a NN (or not) prior to extracting classification rules. The simulations and the corresponding classification rules' accuracy from all four simulations are summarized in Table 15, below.

Coding the problem's attributes, training a NN and finally extracting classification rules from this trained NN for credit evaluation increased the accuracy percentage by 8% = 89.57 – 81.71, thus reaching almost 90%. From the user's (bank managers, credit analysts) point of view there is always an advantage in using this tool, because it shows its results (classification rules) in a form that is easy to understand, showing in details what information (attributes) from the companies that were analyzed were the most relevant for classifications with sufficient accuracy.

This way, bank managers can check if the results obtained by means of this technique are or not in line with their experience. With the technique presented in this paper, contained in the 4th simulation, it is possible to analyze new credit proposals with an adequate safety margin and, consequently, supply the users with an additional analysis tool. Obviously, it is possible to apply this technique to other types of problems as, for instance, medical diagnosis, evaluation engineering, insurance evaluation, lawsuit analysis and materials quality evaluation.

## References

[1] Baesens, B.; Setiono, R.; Mues, C.; Vanthienen, J. (2003). Using Neural Network Rule Extraction and Decision Tables for Credit-Risk Evaluation. *Management Science*, **49**, n. 3, 312-329.

[2] Bond, M.T.; Seiler, V. L.; Seiler, M.J. (2002). Residential Real Estate Prices: a Room with a View. *The Journal of Real Estate Research*, **23**, n. 1, 129-137.

[3] Fausett, L. (1995). *Fundamentals of Neural Networks - Architectures, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, New Jersey.

[4] Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P.; Uthurusamy, R. (1996). *Advances in Knowledge Discovery & Data Mining*. AAAI/MIT.

[5] Fidelis, M.V.; Lopes, H.S.; Freitas, A.A. (2000). Um Algoritmo Genético para Descobrir Regras de Classificação em Data Mining. *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação*, **IV**, 17-29.

[6] Freitas, A.A. (2000). *Uma Introdução a Data Mining. Informática Brasileira em Análise*. CESAR - Centro de Estudos e Sistemas Avançados do Recife. Ano II, n. 32.

[7] Lemos, E.P.; Steiner, M.T.A.; Nievola, J.C. (2005). Análise de Crédito Bancário por meio de Redes Neurais e Árvores de Decisão: uma Aplicação Simples de Data Mining. *Revista de Administração da Universidade de São Paulo* (RAUSP), **40**, n.3, 225-234.

[8] Lu, L.; Setiono, R.; Liu, H. (1995). NeuroRule: A Connectionist Approach to Data Minig. *Proceedings of the 21st. VLDB Conference*, Zürich, Switzerland, 478-489.

[9] Lu, H.; Setiono, R.; Liu H. (1996). Effective Data Mining using Neural Networks. *IEEE Transactions on Knowledge and Data Engineering*, **8**, n. 6, 957-961.

[10] Mannila, H. (1996). Data mining: machine learning, statistics, and databases. International Conference on Scientific and Statistical Database Management, Stockholm, 1-8.

[11] Nguyen, N.; Cripps, A. (2001). Predicting Housing Value: A Comparison of Multiple Regression Analysis and Artificial Neural Networks. *The Journal of Real Estate Research*, 22, n. 3, 313-336.

[12] Olden, J.D.; Jackson, D.A. (2002). Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, **154**, 135-150.

[13] Quinlan, J.C. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann , San Mateo.

[14] Santos, R.T.; Nievola, J.C.; Freitas, A.A. (2000). Extracting Comprehensible Rules from Neural Networks via Genetic Algorithms. *IEEE*, 130-139.

[15] Setiono, R.; Leow, W.K. (1998). FERNN: An Algorithm for Fast Extraction of Rules from Neural Networks. Technical Report, National University of Singapure, Department of Information Systems and Computer Science.

[16] Steiner, M.T.A.; Soma, N.Y.; Shimizu, T.; Nievola, J.C.; Steiner Neto, P.J. (2005). Using Neural Network Extraction for Credit-risk Evaluation. *IFORS 2005* (*17 th. Triennial Conference of the International Federation of Operational Research Societies hosted by INFORMS*), FB-20.

[17] Steiner, M.T.A. ; Soma, N.Y.; Shimizu, T.; Nievola, J.C.; Steiner Neto, P.J. (2006). The Influence of Exploratory Data Analysis on the KDD Process: An Application to Medical Diagnosis. *International Journal of Operations and Quantitative Management*, **12**, n.1, 1-11.

[18] Witten, I.H.; Frank, E. (2000). *Data Mining: Pratical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, San Francisco, California.

Maria Teresinha Arns Steiner got her Master's and Ph.D.'s degrees in Production Engineering, on Operations Research area, at Federal University of Santa Catarina, Brazil, and her Pos-Doc, at the Technological Institute of Aeronautics, São José dos Campos, SP, Brazil. She is an Associate Professor at Federal University of Paraná. Brazil. She lectures on Engineering Undergraduate Programs and on Numerical Methods in Engineering Graduate Program. e-mail: tere@mat.ufpr.br

Pedro José Steiner Neto got his Master's degree in Engineering Management at Florida Institute of Technology, USA and his Ph.D.'s degree in Business at São Paulo University, Brazil. He is an Associate Professor at Federal University of Paraná. He lectures on Business Undergraduate and Graduate Programs. e-mail: pedrosteiner@ufpr.br

Nei Yoshihiro Soma got his Master's degree in Systems Analysis and Applications, at the National Institute of Spatial Research, SP, Brazil; his Ph.D.'s degree in Applied and Computational Mathematics at University of Sheffield, Sheffield, England, and his Pos-Doc on Operations Research area at Universita degli Studi, Bologna, Italy. He is the Computer Division Chief and Full Professor at the Technological Institute of Aeronautics. He lectures on Computer Sciences Graduate Program. e-mail: nysoma@ita.br

Tamio Shimizu got his Master's degree in Computational and Electronic Engineering at the Technological Institute of Aeronautics and his Ph.D.'s degree in Production Engineering at São Paulo University, Brazil. He is Full Professor at São Paulo University. He lectures on Production Engineering Graduate Program at São Paulo University. He was a post-doctoral fellow at several institutes such as Rensselaer Polytechnic Institute, USA, Tokyo University, Japan, and others institutes. e-mail: tmshimiz@usp.br

Júlio Cesar Nievola got his Master's and Ph.D.'s degrees in Electrical Engineering, on Information Systems, at Federal University of Santa Catarina. He is Full Professor at Pontifícia Universidade Católica do Paraná. He lectures on Computer Science and Computer Engineering Undergraduate Programs and on Applied Informatics Graduate Program. email: nievola@ppgia.pucpr.br

**Table 14.** Classification Rules for the analyzed problem with 15 attributes

| Attrbutes | Account age in agency | Area of Activity | Time in Activity | Number of employees | Company's Premises | District | Gross Annual Turnover | Client in another bank | Business Insurance | Financial Application in the Agency | Sales on Installment | Account History | Real Property owned by partners | Movables owned by partners | Risk ascribed by the Bank | Result |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1st. Rule | > 12 | | | | | | | | | | | | | > 12.000 | | Good Player |
| | > 36 | | | | | | | | | | | | > 0 | > 0 | ≠ B | |
| | > 36 | services | | | | | | no | | | | | | | A | |
| 2nd. Rule | | industry | | | rented | others | ≤ 180.000 | | yes | | | | > 0 | | | Bad Player |
| | | | | | | others | ≤ 180.000 | | yes | ≤ 4.000 | | | > 0 | > 0 | | |
| | | industry | ≥ 6 | | rented | others | > 60.000 | | | | | | > 0 | > 12.000 | | |
| | | | ≥ 6 | | | others | > 60.000 | | | ≤ 4.000 | | | > 0 | > 12.000 | | |
| | > 0 | industry | | | rented | others | | | | > 8.000 | | | > 0 | | | |
| | | industry | ≥ 3 | ≤ 10 | rented | | | | yes | | | | > 30.000 | > 12.000 | | |
| | | | ≥ 3 | ≤ 10 | | | | | yes | ≤ 4.000 | | | > 30.000 | > 12.000 | | |
| 3rd. Rule | | | < 6 | | | others | | | | | < 20 | ≠ new client | | | | Good Player |
| | > 0 | | | | | | | | no | ≤ 8.000 | | | | | | |
| 4th. Rule | else | | | | | | | | | | | | | | | Bad Player |

**Table 15.** Summary of the accuracies obtained in the four simulations (patterns x attributes)

| Simulations | Accuracy (%) |
|---|---|
| First Simulation: obtaining classification rules directly from the WEKA software, considering the credit problem's original data (339 x 24). | 81.71 |
| Second Simulation: the same as the first simulation and excluding 13 patterns (326 x 24). | 82.52 |
| Third Simulation: the same as the second simulation with coded attributes (transformed into binary - 326 x 54). | 83.74 |
| Fourth Simulation: the same as the third simulation obtaining the classification rules from the trained Multi-Layer NN, making use of the NeuroRule Algorithm (326 x 54). | 89.57 |

## ATTACHMENT

**Table A.** Attributes considered in this work and their respective coding, "thermometer" or "dummy", changing them into binary

| Attributes | Attributes' Original Values | Intervals | Number of patterns in each interval | Input 1 | Input 2 | Input 3 | Input 4 | Number of Inputs |
|---|---|---|---|---|---|---|---|---|
| 1. There are file restrictions to the company **(nominal attribute)** | | | | **$I_1$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 15 | 0 | --- | --- | --- | |
| | 2 = no | 2 = no | 324 | 1 | --- | --- | --- | |
| 2. There were file restrictions to the company during the last 5 years, but they were raised **(nominal attribute)** | | | | **$I_2$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 14 | 0 | --- | --- | --- | |
| | 2 = no | 2 = no | 325 | 1 | --- | --- | --- | |
| 3. Account age in agency (**ordinal attribute**) | Num. of Months | Num. of Months | | **$I_3$** | **$I_4$** | **$I_5$** | **$I_6$** | 4 |
| | | 1=[0] | 71 | 0 | 0 | 0 | 0 | |
| | | 2=(0,12] | 67 | 0 | 0 | 0 | 1 | |
| | | 3=(12, 36] | 86 | 0 | 0 | 1 | 1 | |
| | | 4=(36,72] | 59 | 0 | 1 | 1 | 1 | |
| | | 5=>72 | 56 | 1 | 1 | 1 | 1 | |
| 4. Area of Activity **(nominal attribute)** | | | | **$I_7$** | **$I_8$** | --- | --- | 2 |
| | 1 = commerce | 1 = commerce | 171 | 0 | 0 | --- | --- | |
| | 2 = industry | 2 = industry | 71 | 1 | 0 | --- | --- | |
| | 3 = services | 3 = services | 97 | 0 | 1 | --- | --- | |
| 5. Time in Activity (**ordinal attribute**) | Num. of Years | Num. of Years | | **$I_9$** | **$I_{10}$** | **$I_{11}$** | **$I_{12}$** | 4 |
| | 1 = > 9 | 1 = > 9 | 76 | 1 | 1 | 1 | 1 | |
| | 2 = 6 to 9 | 2 = (6, 9] | 33 | 0 | 1 | 1 | 1 | |
| | 3 = 3 to 5 | 3 = (3, 6] | 86 | 0 | 0 | 1 | 1 | |
| | 4 = 1 to 2 | 4 = (1, 3] | 36 | 0 | 0 | 0 | 1 | |
| | 5 = < 1 | 5 = < 1 | 108 | 0 | 0 | 0 | 0 | |
| 6. Number of employees **(ordinal attribute)** | Numerical Value | Num. of Employees | | **$I_{13}$** | **$I_{14}$** | **$I_{15}$** | --- | 3 |
| | | 1 = 0 | 94 | 0 | 0 | 0 | --- | |
| | | 2 = [1, 3] | 101 | 0 | 0 | 1 | --- | |
| | | 3 = [4, 10] | 84 | 0 | 1 | 1 | --- | |
| | | 4 = > 10 | 60 | 1 | 1 | 1 | --- | |
| 7. Company's Premises **(nominal attribute)** | | | | **$I_{16}$** | **$I_{17}$** | --- | --- | 2 |
| | 1 = owned | 1 = owned | 47 | 0 | 0 | | | |
| | 2 = rented | 2 = rented | 151 | 0 | 1 | | | |
| | 3 = lent | 3 = lent | 141 | 1 | 0 | | | |
| 8. District **(nominal attribute)** | | | | **$I_{18}$** | --- | --- | --- | 1 |
| | 1 = downtown | 1 = downtown | 146 | 1 | --- | --- | --- | |
| | 2 = others | 2 = others | 193 | 0 | --- | --- | --- | |
| 9. Main clients **(nominal attribute)** | | | | **$I_{19}$** | **$I_{20}$** | --- | --- | 2 |
| | 1 = persons | 1 = persons | 307 | 1 | 0 | --- | --- | |
| | 2 = companies | 2 = companies | 32 | 0 | 1 | --- | --- | |
| | 3 = mixed | 3 = mixed | 0 | 0 | 0 | --- | --- | |
| 10. Gross Annual Turnover **(ordinal attribute)** | Numerical Value | Numerical Value (1,000) | | **$I_{21}$** | **$I_{22}$** | **$I_{23}$** | --- | 3 |
| | | 1 = [0; 60] | 101 | 0 | 0 | 0 | --- | |
| | | 2 = (60; 180] | 113 | 0 | 0 | 1 | --- | |
| | | 3 = (180; 1,000) | 100 | 0 | 1 | 1 | --- | |
| | | 4 = >1,000 | 25 | 1 | 1 | 1 | --- | |

(continues)

| Attributes | Attributes' Original Values | Intervals | Number of patterns in each interval | Input | Input | Input | Input | Number Of Inputs |
|---|---|---|---|---|---|---|---|---|
| 11. Client in another bank **(nominal attribute)** | | | | **$I_{24}$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 143 | 1 | --- | --- | --- | |
| | 2 = no | 2 = no | 196 | 0 | --- | --- | --- | |
| 12. Real property **(ordinal attribute)** | Numerical Value | Numerical Value (1,000) | | **$I_{25}$** | **$I_{26}$** | **$I_{27}$** | --- | 3 |
| | | 1 = [0] | 301 | 0 | 0 | 0 | --- | |
| | | 2 = (0, 50] | 10 | 0 | 0 | 1 | --- | |
| | | 3 = (50, 100] | 13 | 0 | 1 | 1 | --- | |
| | | 4 = > 100 | 15 | 1 | 1 | 1 | --- | |
| 13. Movables **(ordinal attribute)** | Numerical Value | Numerical Value (1,000) | | **$I_{28}$** | **$I_{29}$** | **$I_{30}$** | --- | 3 |
| | | 1 = [0] | 273 | 0 | 0 | 0 | --- | |
| | | 2 = (0, 10] | 24 | 0 | 0 | 1 | --- | |
| | | 3 = (10, 50] | 32 | 0 | 1 | 1 | --- | |
| | | 4 = > 50 | 10 | 1 | 1 | 1 | --- | |
| 14. Business Insurance **(nominal attribute)** | | | | **$I_{31}$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 126 | 1 | --- | --- | --- | |
| | 2 = no | 2 = no | 213 | 0 | --- | --- | --- | |
| 15. Financial Application in the Agency **(ordinal attribute)** | | Numerical Value (1.000) | | **$I_{32}$** | **$I_{33}$** | **$I_{34}$** | **$I_{35}$** | 4 |
| | 1 = > 8,000 | 1 = > 8 | 38 | 1 | 1 | 1 | 1 | |
| | 2 = 4,000 to 8,000 | 2 = (4, 8] | 3 | 0 | 1 | 1 | 1 | |
| | 3 = 2,000 to 4,000 | 3 = (2, 4] | 1 | 0 | 0 | 1 | 1 | |
| | 4 = < 2,000 | 4 = (0, 2] | 12 | 0 | 0 | 0 | 1 | |
| | 5 = no | 5 = 0 | 285 | 0 | 0 | 0 | 0 | |
| 16. Sales on Installment **(nominal attribute)** | | | | **$I_{36}$** | --- | --- | --- | 1 |
| | 1 = < 20% | 1 = [0, 20) | 74 | 1 | --- | --- | --- | |
| | 2 = ≥ 20% | 2 = ≥ 20 | 265 | 0 | --- | --- | --- | |
| 17. Credit Experience with the Agency **(nominal attribute)** | Years | Years | | **$I_{37}$** | **$I_{38}$** | --- | --- | 2 |
| | 1 = > 2 | 1 = > 2 | 75 | 0 | 0 | --- | --- | |
| | 2 = ≤ 2 | 2 = (0, 2] | 264 | 1 | 0 | --- | --- | |
| | 3 = no | 3 = [0] | 0 | 0 | 1 | --- | --- | |
| 18. Account History **(nominal attribute)** | | | | **$I_{39}$** | **$I_{40}$** | **$I_{41}$** | --- | 3 |
| | 1 = normal | 1 = normal | 240 | 0 | 0 | 0 | --- | |
| | 2 = bounced checks | 2 = bounced checks | 2 | 1 | 0 | 0 | --- | |
| | 3 = new client | 3 = new client | 88 | 0 | 1 | 0 | --- | |
| | 4 = frequent small payment delays | 4 = frequent small payment delays | 9 | 0 | 0 | 1 | --- | |
| 19. Partners to the company have file restrictions **(nominal attribute)** | | | | **$I_{42}$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 9 | 0 | --- | --- | --- | |
| | 2 = no | 2 = no | 330 | 1 | --- | --- | --- | |
| 20. Partners to the company had file restrictions during the last 5 years, but were raised **(nominal attribute)** | | | | **$I_{43}$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 51 | 0 | --- | --- | --- | |
| | 2 = no | 2 = no | 288 | 1 | --- | --- | --- | |

(continues)

| Attributes | Attributes' Original Values | Intervals | Number of patterns in each interval | Input | Input | Input | Input | Number Of Inputs |
|---|---|---|---|---|---|---|---|---|
| 21. Society between spouses **(nominal attribute)** | | | | **$I_{44}$** | --- | --- | --- | 1 |
| | 1 = yes | 1 = yes | 116 | 0 | --- | --- | --- | |
| | 2 = no | 2 = no | 223 | 1 | --- | --- | --- | |
| 22. Real Property owned by partners **(ordinal attribute)** | Numerical Value | Numerical Value (1,000) | | **$I_{45}$** | **$I_{46}$** | **$I_{47}$** | --- | 3 |
| | | 1=[0] | 230 | 0 | 0 | 0 | --- | |
| | | 2= (0, 30] | 38 | 0 | 0 | 1 | --- | |
| | | 3= (30, 120] | 37 | 0 | 1 | 1 | --- | |
| | | 4 = > 120 | 34 | 1 | 1 | 1 | --- | |
| 23. Movables owned by partners **(ordinal attribute)** | Numerical Value | Numerical Value (1,000) | | **$I_{48}$** | **$I_{49}$** | **$I_{50}$** | --- | 3 |
| | | 1=[0] | 237 | 0 | 0 | 0 | --- | |
| | | 2=(0, 12] | 33 | 0 | 0 | 1 | --- | |
| | | 3= (12, 24] | 38 | 0 | 1 | 1 | --- | |
| | | 4= > 28 | 31 | 1 | 1 | 1 | --- | |
| 24. Risk ascribed by the Bank **(nominal attribute)** | Scale | Scale | | **$I_{51}$** | **$I_{52}$** | **$I_{53}$** | **$I_{54}$** | 4 |
| | 1=A (best) | 1=A (best) | 85 | 1 | 0 | 0 | 0 | |
| | 2=B | 2=B | 125 | 0 | 1 | 0 | 0 | |
| | 3=C | 3=C | 111 | 0 | 0 | 1 | 0 | |
| | 4=D | 4=D | 15 | 0 | 0 | 0 | 1 | |
| | 5=E (worst) | 5=E (worst) | 3 | 0 | 0 | 0 | 0 | |
| Total Attributes: 24 | --- | --- | --- | --- | --- | --- | --- | 54 |