versão impressa ISSN 0101-7438 / versão online ISSN 1678-5142

EXTRAÇÃO DE REGRAS DE CLASSIFICAÇÃO A PARTIR DE REDES NEURAIS PARA AUXÍLIO À TOMADA DE DECISÃO NA CONCESSÃO DE CRÉDITO BANCÁRIO

Maria Teresinha Arns Steiner *
Dep. Matemática / Univ. Federal do Paraná (UFPR)
Curitiba – PR
tere@mat.ufpr.br

Júlio Cesar Nievola
PPGIA / Pont. Univ. Católica do Paraná (PUC-PR)
Curitiba – PR
nievola@ppgia.pucpr.br

Nei Yoshihiro Soma
Div. Ciência da Computação / Inst. Tec. de Aeronáutica (ITA)
São José dos Campos – SP
nysoma@comp.ita.br

Tamio Shimizu
Dep. Engenharia de Produção / Univ. de São Paulo (USP)
São Paulo – SP
tmshimiz@usp.br

Pedro José Steiner Neto
Dep. Administração / Univ. Federal do Paraná (UFPR)
Curitiba – PR
pedrosteiner@ufpr.br

* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

Recebido em 02/2006; aceito em 09/2007 após 1 revisão
Received February 2006; accepted September 2007 after one revision

Resumo

A avaliação de risco de crédito é um importante problema administrativo da área de análise financeira. As Redes Neurais têm recebido muita atenção pela sua alta taxa de acurácia preditiva, no entanto não é fácil compreender como elas alcançam as suas decisões. Neste artigo um conjunto de dados de crédito é analisado usando a técnica de extração de regras *NeuroRule* e o *software WEKA* para a extração de regras a partir de uma Rede Neural treinada. Os resultados foram considerados bastante satisfatórios alcançando mais de 80% de acurácia quanto à concessão (ou não) de crédito bancário em todas as simulações.

Palavras-chave: redes neurais; técnica de extração *NeuroRule*; risco de crédito.

Abstract

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 predictive accuracy rate, but how they reach their decisions is not easy to understand. In this paper, we present a real-life credit-risk data set and analyzed it using the NeuroRule extraction technique and the software WEKA. The results were considered very satisfactory, reaching more than 80% of accuracy in granting or denying credit on every simulation.

Keywords: neural networks; NeuroRule extraction technique; credit-risk.

1. Introdução

Com o avanço da tecnologia, não tem sido difícil para as grandes empresas armazenar eficientemente grandes volumes de dados (registros históricos) em seus computadores, para recuperá-los sempre que necessário. Muitas delas, no entanto, têm se deparado com o problema de possuir muitos dados, mas pouco conhecimento (*data rich but knowledge poor*) (Lu *et al.*, 1995).

A correta tomada de decisão, conceder (ou não) crédito bancário, seja para pessoa física, jurídica ou outra, é essencial para a sobrevivência das instituições bancárias. Muitas vezes, o prejuízo causado pelo erro na tomada de decisão de conceder crédito a um único cliente pode prejudicar o lucro obtido em muitas operações bem sucedidas (Steiner *et al.*, 2005).

O uso de ferramentas que possam auxiliar nesta tomada de decisão tem sido feito, principalmente por pesquisadores da área de Inteligência Artificial (IA). Duas abordagens básicas para os problemas de classificação (que é o caso do problema de crédito aqui apresentado) estudadas por pesquisadores de IA, são a abordagem simbólica (baseada em árvores de decisão) e a abordagem conexionista (baseada, principalmente, em RN).

Algumas das vantagens obtidas através do correto uso de ferramentas para a tomada de decisão quanto à concessão de crédito são, dentre outras: envolvimento de um menor número de pessoas na análise de crédito, liberando-as para outras atividades; agilidade no processamento das solicitações de crédito; menor subjetividade no decorrer do procedimento da tomada de decisão; maior acurácia dos resultados, ou seja, menor percentual de erros.

Mineração de Dados (*Data Mining*) é uma nova tecnologia utilizada para aumentar a qualidade e a eficiência das decisões – muitas empresas como, por exemplo, as instituições bancárias, têm obtido um alto retorno de seus investimentos fazendo uso das ferramentas de análise de banco de dados (Witten & Frank, 2005).

O objetivo deste artigo é utilizar, dentre as diversas técnicas de *Data Mining* inseridas no contexto de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in DataBases – KDD*), ferramentas capazes de fazer a classificação de empresas (entidades legais) como "boas ou más" tomadoras de crédito, com base nos registros históricos armazenados pelas instituições. Das diversas técnicas de *Data Mining* para classificação, optou-se por utilizar a técnica de extração de regras de classificação de Redes Neurais (RNs) treinadas para a avaliação de risco de crédito, usando as codificações "termômetro" e "*dummy*" (Baesens *et al.*, 2003) para as variáveis (entradas, atributos), tornando-as binárias.

A acurácia das regras de classificação assim obtidas, são comparadas com: 1) a extração de regras diretamente dos dados (padrões) originais; 2) a extração de regras diretamente dos dados originais fazendo o descarte de alguns deles, conforme explicado na seção 5; 3) a extração de regras dos dados originais, fazendo o descarte de alguns deles e, ainda, utilizando as referidas codificações aos atributos.

Pretende-se com as referidas comparações, verificar qual das quatro alternativas mencionadas fornece regras de classificação com maior taxa de acurácia na tarefa de classificação; o grau de importância de se codificar os atributos preliminarmente ao treinamento da RN; assim como o grau de importância de se treinar a RN preliminarmente a extração de regras.

Na seção 2 deste artigo é feita uma descrição do problema real com a apresentação dos dados e atributos utilizados nos experimentos (simulações); na seção 3 discorre-se sobre

Data Mining e *KDD* e, também, sobre alguns trabalhos correlatos; na seção 4 é apresentado o algoritmo *NeuroRule* aplicado a uma RN de Múltiplas Camadas; na seção 5 são desenvolvidas as quatro referidas simulações e, finalmente, na seção 6 são apresentadas as conclusões.

2. Descrição do Problema Real

Os dados utilizados neste artigo, obtidos junto a um grande banco brasileiro (Lemos *et al.*, 2005), se referem ao crédito de pessoas jurídicas (empresas). O referido banco coloca à disposição de seus clientes linhas de crédito, tanto para capital de giro quanto para investimentos, sendo que a sua clientela é constituída, basicamente, por micros, pequenas e médias empresas.

Por ocasião da pesquisa, o banco utilizava como ferramenta para realizar sua análise de crédito, um aplicativo interno chamado Análise de Crédito (ANC). A partir deste aplicativo, que contém as informações cadastrais e contábeis das empresas, a gerência se apoiava para tomar as suas decisões com relação a concessão (ou não) de crédito bancário.

Para o desenvolvimento deste artigo foram utilizados registros históricos de um total de 339 clientes pessoa jurídica, dos quais 266 comprovadamente eram adimplentes e 73 inadimplentes. De cada um destes clientes foram extraídos 24 atributos, indicados pelos especialistas da área (gerentes de crédito do banco), especificadas na Tabela A em anexo, com os seus respectivos valores originais (colunas 1 e 2), sendo que na coluna 1 está acrescentado, também, o seu tipo que, para este problema, pode ser ordinal ou nominal. Nas colunas 3 e 4 desta Tabela A encontram-se os intervalos de valores definidos para cada um dos atributos e o número de padrões contidos em cada intervalo, respectivamente. Nas colunas 5 a 8 encontra-se a "codificação termômetro" ou a "codificação *dummy*", conforme o tipo do atributo, explicadas na seção 3. Finalmente, na coluna 9 está contabilizada a quantidade de entradas para a RN utilizada, valor que depende, obviamente, da codificação utilizada.

3. KDD, *Data Mining* e Trabalhos Correlatos

As técnicas e ferramentas que buscam transformar os dados armazenados por empresas em conhecimento, são o objetivo da grande área ou processo chamado *KDD*. O processo *KDD* é um conjunto de atividades contínuas que compartilham o conhecimento descoberto a partir de bases de dados. Segundo Fayyad *et al.* (1996) esse conjunto é composto de cinco etapas: seleção dos dados; pré-processamento e limpeza dos dados; transformação dos dados; *Data Mining*; interpretação e avaliação dos resultados, cuja interligação pode ser visualizada na Figura 1.

O processo *KDD* começa com o entendimento do domínio do problema e dos objetivos finais a serem atingidos. É feito um agrupamento organizado dos dados disponíveis, alvo da prospecção. A etapa da limpeza dos dados (*data cleaning*) vem a seguir, através de um pré-processamento dos dados, fazendo a integração de dados heterogêneos, eliminação de incompletude dos dados e outras. Essa etapa pode tomar até 80% do tempo necessário de todo o processo, devido as bem conhecidas dificuldades de integração de bases de dados heterogêneas (Mannila, 1996).

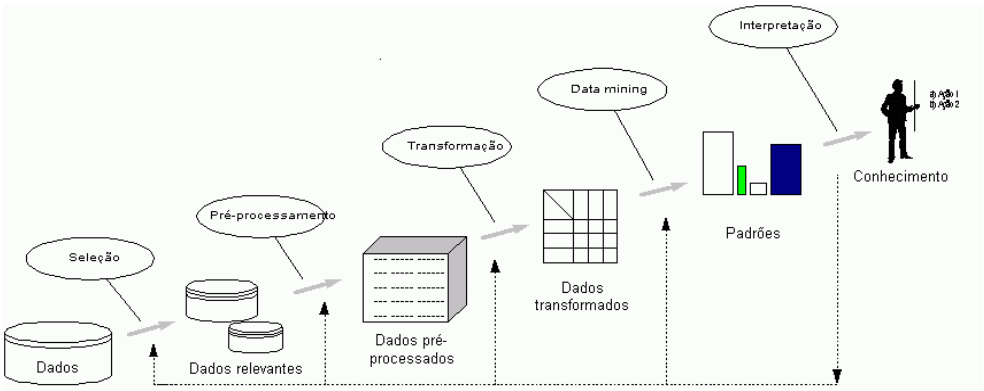

Figura 1 – Atividades que compõem o processo *KDD* (Fayyad *et al.*, 1996).

Os dados pré-processados podem, ainda, passar por uma transformação que os armazena adequadamente; nesta fase, o uso do armazenamento de dados (*data warehouse*) se expande consideravelmente, já que com essa tecnologia as informações podem ser armazenadas de maneira mais eficiente. Todas estas etapas, preliminares a etapa de *Data Mining*, podem ser vistas como uma Análise Exploratória dos Dados (Steiner *et al.*, 2006). Esta análise pode envolver ainda, dentre outras alternativas, a padronização dos dados e o descarte de dados atípicos. No presente artigo, estas etapas preliminares envolveram, basicamente, seleção, limpeza e codificação (em duas das quatro simulações) dos dados.

Tem-se, então, a etapa de *Data Mining*, que começa com a escolha dos algoritmos a serem utilizados. Essa escolha depende, fundamentalmente, do objetivo do processo de *KDD* (Witten & Frank, 2005) que pode ser: classificação, agrupamento ou associação. De modo geral, na etapa de *Data Mining*, os algoritmos utilizados procuram por padrões nos dados.

Diversas ferramentas distintas, como RNs, árvores de decisão, sistemas baseados em regras, programas estatísticos e outras, tanto isoladamente quanto em combinação, podem ser então aplicadas ao problema. Em geral, o processamento de busca é interativo, de forma que os analistas avaliam o resultado, formam um novo conjunto de questões para refinar a busca em um dado aspecto das descobertas e realimentam o sistema com novos parâmetros. Ao final do processo, um relatório das descobertas é gerado, que passa então a ser interpretado pelos analistas de mineração e o conhecimento é descoberto. *Data Mining* é a parte mais interessante do processo *KDD*, sendo que no contexto de negócios é a que mais alavanca e auxilia o empresário a descobrir filões de mercado.

Segundo Freitas (2000) o conhecimento a ser descoberto deve ser correto; compreensível e útil; além disso, o método de descoberta do conhecimento deve ser, por sua vez, eficiente; genérico (aplicável a diversos tipos de problemas); flexível (facilmente modificável).

Dentre as técnicas de *Data Mining* utilizadas em problemas de classificação, que é o caso do problema abordado neste artigo, destacam-se as RNs, que constroem representações internas de modelos ou padrões detectados nos dados, mas essas representações não são apresentadas de forma explícita aos usuários. Neste artigo é feita a extração de regras de classificação a partir dos atributos codificados e da RN já treinada, com o intuito de deixar claro, compreensível ao usuário (gerente de crédito) de que forma que os atributos estão "atuando" para fazer a classificação de cada um dos clientes.

Dentre os numerosos trabalhos que abordam as técnicas de *Data Mining* para classificação, pode-se citar: Lu *et al.* (1995) e Lu *et al.* (1996) que apresentam o algoritmo chamado *Neurorule* (utilizado no presente artigo) que faz a extração de regras a partir de uma RN treinada, obtendo regras do tipo SE-ENTÃO (*IF-THEN*); o desempenho desta abordagem foi verificado, em ambos os artigos, em um problema de crédito bancário. Fidelis *et al.* (2000) apresentam um algoritmo de classificação baseado em Algoritmos Genéticos (AGs) que descobre regras compreensíveis do tipo *IF-THEN* no contexto de *Data Mining*; a proposta foi avaliada em duas bases de dados médicos de domínio público, de dermatologia e de câncer de mama.

Setiono & Leow (1998) apresentam um método rápido para extrair regras de RN treinadas (*FERNN – a Fast Method for Extracting Rules from Trained Neural Networks*); primeiro, *FERNN* identifica as unidades escondidas relevantes usando o algoritmo C4.5 (Quinlan, 1993); depois, para cada unidade escondida relevante, *FERNN* acha o conjunto de conexões relevantes das entradas para as unidades escondidas e, finalmente, *FERNN* substitui as condições divididas da árvore de decisão gerada pelo C4.5 por regras envolvendo as entradas da rede. Santos *et al.* (2000) usam um AG para definir uma topologia adequada a uma RN a ser treinada; o sistema proposto foi avaliado em três conjuntos de dados disponíveis no repositório *UCI*: *Iris*, *Wine* e *Monks-2*.

Baesens *et al.* (2003) abordam três métodos para a extração de regras de uma RN, comparativamente: *Neurorule*; *Trepan* e *Nefclass* em três bases de dados reais de crédito: *German Credit*, *Bene 1* e *Bene 2*. Olden & Jackson (2002) descrevem alguns métodos da literatura para "desvendar" os mecanismos de uma RN. Na avaliação imobiliária, pode-se citar o trabalho de Nguyen & Cripps (2001), que comparam o desempenho preditivo de RN com a Análise de Regressão Múltipla para a venda de casas de família. No trabalho de Bond *et al.* (2002) é examinado o efeito que a vista de um lago tem sobre o valor de uma casa.

4. Algoritmo *NeuroRule* e Codificação dos Atributos

Dentre os muitos modelos existentes de RNs, decidiu-se por utilizar uma RN de Múltiplas Camadas (Fausett, 1995) a qual foi treinada com o algoritmo *back-propagation* (retro-propagação), de aprendizado supervisionado.

Conforme mencionado anteriormente, neste artigo o objetivo é extrair regras de classificação a partir atributos codificados e da RN treinada e, para isso, fez-se uso do algoritmo *NeuroRule*, descrito mais adiante – passos 1 a 4 (Lu *et al.*, 1995).

Os passos 2 e 3 do Algoritmo *NeuroRule* exigem a geração de regras perfeitas a partir de uma RN treinada. Para a geração destas regras fez-se uso do *software WEKA* (*Waikato Environment for Knowledge Analysis*), disponível em *World Wide Web* (www.cs.waikato.ac.nz/ml/weka). Este *software* contém dez algoritmos para extração de regras de classificação, que são os seguintes: *JRip*; *ZeroR*; *Ridor*; *Prisma*; *M5Rules*; *Part*; *OneR*; *Nnge*; *Decision Table* e, finalmente, *Conjunctive Rule*, todos descritos em Witten & Frank (2005).

Dentre os algoritmos anteriormente citados, o *JRip* ou *Ripper* (*Repeated Incremental Pruning to Produce Error Reduction* ou Poda Incremental Repetida para Produzir Redução de Erro, proposto por William W. Cohen (Witten & Frank, 2005)) foi o que apresentou melhor desempenho em praticamente todas as vezes em que os mesmos foram comparados e, por este motivo, na seção 4.2 é feita a sua descrição.

4.1 Algoritmo *NeuroRule* para Extração de Regras (*NeuroRule Extraction*):

Passo 1. Faça a discretização dos valores de ativação da RN via agrupamento. Uma alternativa é a seguinte:

1a. Seja $\varepsilon \in (0, 1)$ e seja D o número de valores de ativação discretos na camada escondida da RN. Ainda, seja δ_1 o valor de ativação na camada escondida para o primeiro padrão do conjunto de treinamento e seja $H(1) = \delta_1$, contador(1) = 1, soma(1) = δ_1 e faça $D = 1$.

1b. Para todos os padrões $i = 2, 3, ..., k$ do conjunto de treinamento:

Seja δ o valor de ativação.

Se existe um índice j' tal que:

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

então faça contador(j') = contador(j') + 1, soma(D) = soma(D) + δ

caso contrário, $D = D + 1$, $H(D) = \delta$, contador (D) = 1, soma(D) = δ.

1c. Substitua H pela média de todos os valores de ativação que tenham sido agrupados neste grupo, ou seja, $H(j)$ = soma(j)/contador(j), $j = 1, 2, ..., D$.

1d. Cheque a acurácia da RN com os valores de ativação δ^i nos nós escondidos substituídos por δ^d, o valor de ativação do grupo ao qual o valor de ativação pertence.

1e. Se a acurácia ficar abaixo do nível exigido, decresça ε e repita o passo 1.

Passo 2. Enumere os valores de ativação discretizados e calcule a saída da rede.

Gere regras perfeitas (com a máxima acurácia) que tenham uma cobertura perfeita de todos os exemplos dos valores de ativação discretizados dos nós escondidos para os valores de saída.

Passo 3. Para os valores de ativação discretizados que apareceram nos nós escondidos nas regras definidas no passo 2, enumere os valores de entrada que conduziram a eles, gere regras perfeitas.

Passo 4. Gere regras que relacionam os valores de entrada e os valores de saída por substituição de regras com base nos resultados dos passos 2 e 3 anteriores.

4.2 Algoritmo *Ripper* ou *JRip* (desenvolvido na linguagem de programação *Java*):

Para um problema de duas classes, escolha uma delas como positiva e a outra como classe negativa:

1. Aprenda regras para a classe positiva;
2. A classe negativa será a classe *default* (padrão).

Para um problema de várias classes:

1. Ordene as classes de acordo com a prevalência crescente da classe (iniciar com a classe que contém a menor quantidade de exemplos ou padrões);
2. Aprenda primeiro o conjunto de regras para a menor classe (menor número de exemplos), considerada como classe positiva; trate o restante como classes negativas;
3. Repita com a seguinte menor classe, tratando-a como classe positiva.

Construindo um conjunto de regras:

1. Use o algoritmo de cobertura seqüencial:

   Encontre a melhor regra que cubra o conjunto atual de exemplos positivos;

   Elimine tanto os exemplos positivos quanto negativos cobertos pela regra.

2. Cada vez que uma regra é colocada no conjunto de regras, calcule o novo comprimento da descrição:

   Pare de adicionar novas regras quando o novo comprimento da descrição for *d* bits maior que o menor comprimento de descrição encontrado até então.

Crescendo uma regra:

1. Inicie com a regra vazia;
2. Adicione conjunções enquanto elas melhorarem o ganho de informação *FOIL* (*First-Order Induction Learning* ou Aprendizado por Indução de Primeira Ordem (Quinlan, 1990));
3. Pare quando a regra não cobrir mais exemplos negativos;
4. Pode a regra imediatamente usando o incremento da poda do erro reduzido (*reduced error pruning incremental*):

   Medida para poda: $v = (p - n) / (p + n)$, em que:

   $p =$ número de exemplos positivos cobertos pela regra no conjunto de validação; e

   $n =$ número de exemplos negativos cobertos pela regra no conjunto de validação.

   Método de Poda: retire qualquer seqüência final de condições que maximize *v*.

Otimizando o conjunto de regras:

1. Para cada regra *r* no conjunto de regras *R*:

   Considere 2 regras alternativas:

   Regra de substituição (r*): cresça nova regra a partir do zero;

   Regra de revisão (r'): adicione conjunções para estender *r*.

   Compare a regra *r* com as regras r* e r'.

2. Escolha o conjunto de regras que minimize o *MDL* (*Minimum Description Length Principle* ou Princípio da Descrição de Mínimo Comprimento: representa o modelo da forma mais compacta possível com o máximo de informações dos dados):

   Repita geração e otimização de regras para o restante dos exemplos positivos.

4.3 Codificação dos Atributos

O algoritmo *Neurorule* utilizado neste artigo para a extração de regras a partir de uma RN treinada, assume que os dados são discretizados e representados como entradas binárias usando a "codificação termômetro" para os atributos ordinais e a "codificação *dummy*" (artificial) para os atributos nominais (Baesens *et al.*, 2003).

A Tabela 1 ilustra a "codificação termômetro" para a variável ordinal "renda", por exemplo. O atributo "renda" é, primeiramente, discretizado nos valores 1, 2, 3 e 4; se, por exemplo,

$I_3 = 1$, isto significa que a variável original "renda" > 1.000. No presente artigo, esta discretização foi feita com o auxílio de um especialista.

A Tabela 2 ilustra a "codificação *dummy*" para a variável "propósito do empréstimo", por exemplo. Este esquema de codificação facilita a geração e interpretação das regras SE-ENTÃO.

Tabela 1 – Um exemplo do procedimento "codificação termômetro" para variáveis ordinais.

Entrada original Renda (R$)	Entrada Categórica	I_1	I_2	I_3
Renda ≤ 1.000	1	0	0	0
1.000 < Renda ≤ 2.000	2	0	0	1
2.000 < Renda ≤ 3.000	3	0	1	1
Renda > 3.000	4	1	1	1

Tabela 2 – Um exemplo do procedimento "codificação *dummy*" para variáveis nominais.

Entrada original Propósito	I_1	I_2
Propósito = carro	0	0
Propósito = imóvel	0	1
Propósito = outro	1	0

5. Implementação e Resultados

Neste artigo foram desenvolvidas quatro simulações com os dados com o objetivo de obter regras de classificação, conforme descritas de 5.1 a 5.4 a seguir. O modo para teste, em todas as quatro simulações, foi o método padrão do *software WEKA*, ou seja, "10 arquivos para validação cruzada" (*10-fold cross-validation*) (Witten & Frank, 2005).

5.1 Primeira Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito

Nesta 1ª simulação, as regras de classificação foram obtidas diretamente dos dados originais do problema, sem levar em consideração a RN treinada. Esta 1ª simulação foi feita com o objetivo de verificar a importância (ou não) de se codificar os atributos, assim como a de se treinar uma RN preliminarmente a extração de regras.

Para esta 1ª simulação foram considerados os dados originais do problema, ou seja, os 339 padrões (pessoas jurídicas), em que 266 pertencem ao conjunto *A* (adimplentes, resposta = 1) e 73 pertencem ao conjunto *B* (inadimplentes, resposta = 0). Para cada um destes padrões tem-se 24 atributos contidos na Tabela A em anexo.

Dentre os dez métodos para obtenção das regras de classificação contidos no *software WEKA*, o método *JRip* apresentou a maior taxa de acurácia. Os resultados obtidos estão contidos na Tabela 3 a seguir.

Tabela 3 – Resultados para a 1ª Simulação: Regras de Classificação obtidas diretamente a partir dos dados originais com o auxílio do *software WEKA*.

REGRAS		RESULTADO
SE [(cliente = novo) E (número de funcionários ≥ 0) E (faturamento anual bruto ≥ 54.000)]	então	Inadimplente (33/11)
Caso Contrário	então	Adimplente (306/51)

Através da Tabela 3, 1ª regra, tem-se que 33 padrões foram classificados como inadimplentes, sendo que 11 deles com erro, ou seja, dos 33 padrões inadimplentes, 11 padrões, na verdade, eram adimplentes. Da mesma forma, pela 2ª regra, 306 padrões foram classificados como adimplentes, sendo que 51 com erro. Estes valores podem ser representados pela matriz de confusão apresentada na Tabela 4. Assim, tem-se que a acurácia para esta 1ª simulação é de: 1 – [(11 + 51) / 339] = 81,71%.

Tabela 4 – Matriz de Confusão para a 1ª Simulação.

REAL\CLASSIFICAÇÃO	ADIMPLENTES	INADIMPLENTES
ADIMPLENTES	306	51
INADIMPLENTES	11	33

5.1.1 Codificação dos Atributos e Treinamento da Rede Neural

Finalizada a 1ª simulação, cada um dos 24 atributos foi transformado segundo as codificações "termômetro" e "*dummy*", tornando-os binários, conforme explicado na seção 4.2 anterior, obtendo-se os dados da Tabela A em anexo, em que cada padrão possui 54 atributos (e não mais 24). Considerando-se estes 54 atributos, uma RN de Múltiplas Camadas (3 Camadas) foi treinada com o algoritmo *back-propagation*. Depois de testar várias topologias, cada uma delas com diferentes pesos iniciais, os melhores resultados obtidos foram com uma topologia contendo 4 neurônios na camada escondida. Esta topologia, com 54 entradas, 4 neurônios na camada escondida e um neurônio na camada de saída (adimplente ou inadimplente), (54 – 4 – 1), classificou erroneamente apenas 13 dos 339 padrões, ou seja, a acurácia desta RN foi de 1 – [13 / 339] = 96,17%.

Para extrair regras de classificação "perfeitas" a partir da referida RN treinada (54 – 4 – 1), os 13 padrões classificados erroneamente pela RN foram excluídos da amostra de dados.

5.2 Segunda Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito, com a exclusão dos 13 padrões anteriormente mencionados

Para esta 2ª simulação foram considerados os 326 padrões originais (13 padrões excluídos), em que 256 pertencem ao conjunto *A* e 70 pertencem a *B*. Dentre os métodos para obtenção das regras de classificação d*o software WEKA*, o método *JRip* novamente apresentou os melhores resultados, os quais estão contidos na Tabela 5 a seguir. A acurácia para esta 2ª simulação é de: 1 – [(48 + 9) / 326] = 82,51%. A forma de interpretação dos resultados contidos na Tabela 5 é análoga a da Tabela 3 anterior.

Tabela 5 – Resultados para a 2ª Simulação: Regras de Classificação obtidas diretamente a partir dos dados originais, com a exclusão de 13 padrões, com o auxílio do *software WEKA*.

REGRAS		RESULTADO
SE [(tempo de conta no banco ≤ 8 meses) E (número de funcionários ≥ 1) E (faturamento anual bruto ≥ 75.000)]	então	Inadimplente (31/9)
Caso Contrário	então	Adimplente (295/48)

5.3 Terceira Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito com os atributos codificados e com a exclusão de 13 padrões

Para esta 3ª simulação também foram considerados os 326 padrões, em que 256 pertencem ao conjunto *A* e 70 pertencem a *B*. Para cada um destes padrões tem-se 54 atributos transformados em binários pelas codificações "termômetro" e "*dummy*", conforme seção 4.2 e Tabela A em anexo.

Dentre os métodos contidos *no software WEKA*, novamente o método *JRip* apresentou os melhores resultados, que estão contidos na Tabela 6 a seguir. A acurácia para esta 3ª simulação é de: 1 – [(1 + 12 + 40) / 326] = 83,74%.

Tabela 6 – Resultados para a 3ª Simulação: Regras de Classificação obtidas diretamente a partir dos dados codificados, com os 13 padrões descartados, com o auxílio do *software WEKA*.

REGRAS		RESULTADO
SE [(idade da conta na agência ≤ 12) E (existência de bens imóveis em nome dos sócios > 0) E (sede da empresa = própria ou alugada) E (bairro = outros)	então	Inadimplente (10/1)
SE [(idade da conta na agência ≤ 12) E (faturamento bruto anual > 60.000) E (risco atribuído pela agência = C)	então	Inadimplente (33/12)
Caso contrário	então	Adimplente (283/40)

5.4 Quarta Simulação: obtenção das regras de classificação a partir da RN de Múltiplas Camadas já treinada, fazendo uso do Algoritmo *NeuroRule*, considerando os dados originais do problema de crédito com os atributos codificados, com a exclusão dos 13 padrões, com o auxílio do *software WEKA*

Para esta 4ª simulação foram considerados novamente os 326 padrões (conjunto *A* com 256 padrões e o conjunto *B* com 70 padrões). Para cada um destes padrões tem-se os mesmos 54 atributos transformados pelas já mencionadas codificações.

Desta forma, utilizou-se o Algoritmo *NeuroRule* para obter as regras de classificação. Inicialmente os valores de ativação foram agrupados em cada um dos quatro neurônios da camada escondida (*a*, *b*, *c* e *d*). Para os dados apresentados neste artigo, os valores obtidos para *a*, *b*, *c* e *d*, para todos os 326 padrões, foram "1" ou "0", conforme já mencionado.

Definidos os agrupamentos em cada um dos quatro neurônios ("1" e "0"), foram obtidas as regras de classificação da camada escondida para a camada de saída utilizando o *software WEKA* e, em seguida, as regras de classificação da camada de entrada para a camada escondida para, finalmente, juntá-las para obter as regras de classificação da camada de entrada para a camada de saída. A situação para esta 4ª simulação está ilustrada na Figura 2, onde i_1, i_2, ..., i_{54} são as entradas da RN; w_{ij}'s e w_{hi}'s são as conexões (pesos) entre as camadas de entrada e escondida e entre as camadas escondida e de saída, respectivamente e $\theta_{i,1}$, i = 1, 2, 3 e 4 e $\theta_{h,2}$, h=*1* são as *bias* das camadas escondida e de saída, respectivamente.

Dentre os métodos contidos *no software WEKA*, o método *Part* foi o que apresentou a maior acurácia para as regras de classificação da camada escondida para a camada de saída. Os resultados obtidos estão contidos na Tabela 7, sendo que a acurácia foi de 99,08%.

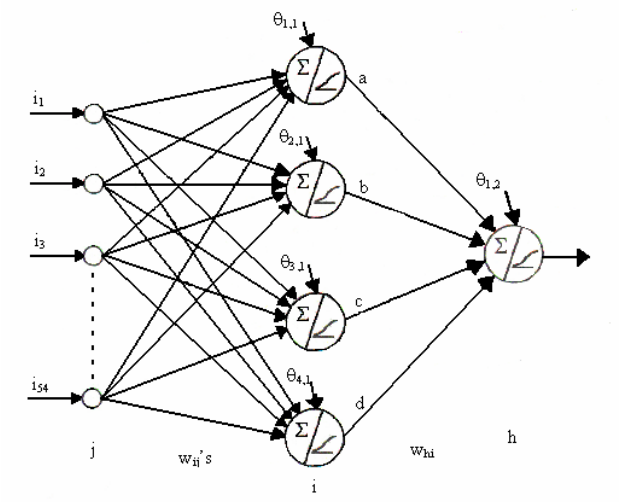

Figura 2 – RN de Múltiplas (3) Camadas, (54 – 4 – 1) que, dentre as topologias testadas, apresentou a melhor acurácia para os dados do problema.

Tabela 7 – Regras de classificação da camada escondida para a camada de saída.

REGRAS		RESULTADO
SE (b = 1)	então	1 = Adimplente (180/1)
SE (a = 0) E (d = 0)	então	0 = Inadimplente (66/2)
SE (c = 0)	então	1 = Adimplente (75/0)
CASO CONTRÁRIO	então	0 = Inadimplente (5/0)

Para obter as regras de classificação da camada de entrada para a 1ª unidade (a) da camada escondida, o método *JRip* foi o que apresentou melhores resultados; a acurácia foi de 92,94%. As regras estão contidas na Tabela 8.

Tabela 8 – Regras de Classificação da Camada de Entrada para o 1º neurônio da Camada Escondida (*a*).

REGRAS		RESULTADO
SE [(i_{18} = 1) E (i_{31} = 0) E (i_{47} = 0) E (i_{22} = 1)]	então	*a* = 1 (=Adimplente) (18/1)
SE [(i_{18} = 1) E (i_{47} = 0) E (i_{10} = 0) E (i_{23} = 0) E (i_{49} = 0)]	então	*a* = 1 = (Adimplente) (19/0)
SE [(i_{47} = 0) E (i_{18} = 1) E (i_{6} = 0) E (i_{32} = 0)]	então	*a* = 1 = (Adimplente) (9/1)
SE [(i_{13} = 1) E (i_{46} = 0) E (i_{31} = 0) E (i_{49} = 0) E (i_{11} = 0)]	então	*a* = 1 = (Adimplente) (6/0)
CASO CONTRÁRIO	então	*a* = 0 = (Inadimplente) (274/21)

Para obter as regras de classificação da camada de entrada para a 2ª unidade (*b*) da camada escondida, novamente o método *JRip* apresentou os melhores resultados; a acurácia foi de 91,10%. As regras estão contidas na Tabela 9.

Tabela 9 – Regras de Classificação da Camada de Entrada para o 2º neurônio da Camada Escondida (*b*).

REGRAS		RESULTADO
SE [(i_{5} = 0) E (i_{49} = 0)]	então	*b* = 0 = (Inadimplente) (104/7)
SE [(i_{4} = 0) E (i_{50} = 0) E (i_{52} = 1) E (i_{47} = 0)]	então	*b* = 0 = (Inadimplente) (19/1)
SE [(i_{4} = 0) E (i_{24} = 1) E (i_{51} = 0) E (i_{8} = 0)]	então	*b* = 0 = (Inadimplente) (18/4)
CASO CONTRÁRIO	então	*b* = 1 = (Adimplente) (185/17)

O método *JRip* apresentou os melhores resultados também na obtenção das regras de classificação da camada de entrada para a 3ª unidade (*c*) da camada escondida; a acurácia foi de 99,08%. As regras estão contidas na Tabela 10.

Tabela 10 – Regras de Classificação da Camada de Entrada para o 3º neurônio da Camada Escondida (*c*).

REGRAS		RESULTADO
SE [(i_{40} = 1) E (i_{18} = 1) E (i_{10} = 1) E (i_{36} = 0)]	então	*c* = 1 = (Adimplente) (11/2)
SE [(i_{32} = 1) E (i_{6} = 0) E (i_{31} = 1)]	então	*c* = 1 = (Adimplente) (3)
CASO CONTRÁRIO	então	*c* = 0 = (Inadimplente) (312/1)

As regras de classificação da camada de entrada para a 4ª unidade (*d*) da camada escondida, foram obtidas utilizando-se o método *JRip* que, novamente, apresentou a melhor acurácia, 81,90%. As regras estão contidas na Tabela 11.

Tabela 11 – Regras de Classificação da Camada de Entrada para o 4º neurônio da Camada Escondida (*d*).

REGRAS		RESULTADO
SE [(i_{17} = 0) E (i_7 = 0)]	então	*d* = 1 = (Adimplente) (137/35)
SE [(i_{33} = 1) E (i_{50} = 0)]	então	*d* = 1 = (Adimplente) (15/3)
CASO CONTRÁRIO	então	*d* = 0 = (Inadimplente) (174/21)

Finalmente, para obter as regras de classificação da camada de entrada para a camada de saída, "juntou-se" as regras anteriormente obtidas, obtendo-se as regras de classificação da camada de entrada para a camada de saída. Os resultados estão apresentados na Tabela 12.

Tabela 12 – Regras de Classificação da Camada de Entrada para a Camada de Saída (atributos codificados).

REGRAS (onde o símbolo "¬" = "não"; "1" = Adimplente; "0" = Inadimplente		RESULTADO
SE (b = 1) – (da Tabela 7) (Analisando as Tabelas 7 e 9 (neurônio *b*)): SE {¬ [i_5 = 0 E i_{49} = 0] OU ¬ [i_4 = 0 E i_{50} = 0 E i_{52} = 1 E i_{47} = 0] OU ¬ [i_4 = 0 E i_{24} = 1 E i_{51} = 0 E i_8 = 0]} (são 9 atributos envolvidos nesta 1ª regra)	então	1 = (Adimplente) (180/12)
SE [(a = 0) E (d = 0)] – (da Tabela 7) (Analisando as Tabelas 7, 8 e 11 (neurônios *a* e *d*)) SE {¬ [i_{18} = 1 E i_{31} = 0 E i_{47} = 0 E i_{22} = 1] OU ¬ [i_{18} = 1 E i_{47} = 0 E i_{10} = 0 E i_{23} = 0 E i_{49} = 0] OU ¬ [i_{47} = 0 E i_{18} = 1 E i_6=0 E i_{32} = 0] OU ¬ [i_{13} = 1 E i_{46} = 0 E i_{31} = 0 E i_{49} = 0 E i_{11} = 0]} E {¬ [i_{17} = 0 E i_7 = 0] OU ¬ [i_{33} = 1 E i_{50} = 0]} (são 16 atributos envolvidos nesta 2ª regra)	então	0 = (Inadimplente) (66/21)
SE (c = 0) – (da Tabela 7) (Analisando as Tabelas 7 e 10 (neurônio *c*)) SE {¬ [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] } (são 7 atributos envolvidos nesta 3ª regra)	então	1 = (Adimplente) (75/1)
CASO CONTRÁRIO	então	0 = (Inadimplente) (5/0)

Transformando as entradas i_1 a i_{54} em seus verdadeiros significados, conforme Tabela A em anexo, tem-se as regras de classificação finais para esta 4ª simulação, apresentadas na Tabela 13.

Tabela 13 – Regras de Classificação da Camada de Entrada para a Camada de Saída (atributos com os seus significados reais).

REGRAS (onde o símbolo "¬" = "não"; "1" = Adimplente; "0" = Inadimplente)		**RESULTADO**
SE {[(tempo de conta no banco > 12 meses) E (existência de bens móveis em nome dos sócios > 12.000] **OU** [(tempo de conta no banco > 36 meses) E (existência de bens móveis em nome dos sócios > 0) E (risco atribuído pelo banco ≠ B) E (existência de bens imóveis em nome dos sócios > 0)] **OU** [(tempo de conta no banco > 36 meses) E (cliente em outro banco = não) E (risco atribuído pelo banco = A) E (setor de atividade = serviços)]}	então	Adimplente (180/12)
SE {[(bairro = outros) E (seguro empresarial = sim) E (existência de bens imóveis em nome dos sócios > 0) E (faturamento bruto anual ≤ 180.000)] **OU** [(bairro = outros) E (existência de bens imóveis em nome dos sócios > 0) E (tempo de atividade ≥ 6 anos) E (faturamento bruto anual > 60.000) E (existência de bens móveis em nome dos sócios > 12.000)] **OU** [(existência de bens imóveis em nome dos sócios > 0) E (bairro = outros) E (tempo de conta no banco > 0) E (aplicações financeiras no banco > 8.000)] **OU** [(número de funcionários ≤ 10) E (existência de bens imóveis em nome dos sócios > 30.000) E (seguro empresarial = sim) E (existência de bens móveis em nome dos sócios > 12.000) E (tempo de atividade ≥ 3 anos)]} **E** {[(sede da empresa = alugada) E (setor de atividade = indústria)] **OU** [(aplicações financeiras no banco ≤ 4.000) E (existência de bens móveis em nome dos sócios > 0)]}	então	Inadimplente (66/21)
SE {[(histórico da conta ≠ cliente novo) E (bairro = outros) E (tempo de atividade < 6 anos) E (vendas a prazo < 20%)] **OU** [(aplicações financeiras no banco ≤ 8.000) E (tempo de conta no banco > 0) E (seguro empresarial = não]}	então	Adimplente (75/1)
Caso Contrário	então	Inadimplente (5)

A acurácia para esta 4ª simulação é de: 1 – [(12 + 1 + 21) / 326] = 89,57%, conforme a matriz de confusão apresentada na Tabela 14.

Tabela 14 – Matriz de Confusão para a 4ª Simulação.

REAL\CLASSIFICAÇÃO	ADIMPLENTES	INADIMPLENTES
ADIMPLENTES	255 (180 + 75)	13 (12 + 1)
INADIMPLENTES	21	71 (66 + 5)

6. Conclusões

O objetivo deste artigo é apresentar ferramentas que possam auxiliar a identificar e prever quais clientes serão adimplentes (ou não) em relação ao crédito em instituições bancárias.

Apesar dos problemas de *Data Mining* envolverem, em geral, milhares ou até milhões de dados, diferentemente do problema aqui apresentado (339 x 24 na 1ª simulação; 326 x 24 na 2ª e 326 x 54 na 3ª e 4ª simulações), as conclusões aqui obtidas podem servir de respaldo para problemas de maior porte. Além disso, o ideal seria que o número de padrões fosse bem maior do que 339 (ou 326), principalmente, para a 3ª e 4ª simulações onde o número de atributos é cerca de apenas 1/6 (= 54) do número de padrões (326); seria interessante que esta relação (nº. atributos/nº. padrões) fosse da ordem 1/10 ou menor.

É apresentada uma forma de se extrair regras de classificação de um problema cujos atributos foram codificados, tornando-os binários, e cujos padrões foram treinados com uma RN. A extração deu-se com a aplicação do algoritmo *NeuroRule* e com o auxílio do *software WEKA* (4ª simulação). A acurácia das regras de classificação assim obtidas, foi comparada com a acurácia de outras três simulações, menos elaboradas, para verificar a importância (ou não) de se codificar os atributos, assim como a de se treinar uma RN preliminarmente a extração de regras de classificação. As simulações e as respectivas acurácias das regras de classificação em todas as quatro simulações estão sintetizadas na Tabela 15 a seguir.

Tabela 15 – Síntese das acurácias obtidas nas quatro simulações (padrões x atributos).

Simulações	Acurácia (%)
Primeira Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito (339 x 24).	81,71
Segunda Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito, com a exclusão de 13 padrões (326 x 24).	82,52
Terceira Simulação: obtenção das regras de classificação diretamente do *software WEKA* considerando os dados originais do problema de crédito, com os atributos codificados (transformados em binários) e com a exclusão de 13 padrões (326 x 54).	83,74
Quarta Simulação: obtenção das regras de classificação a partir da RN de Múltiplas Camadas treinada, fazendo uso do Algoritmo *NeuroRule*, considerando os dados originais do problema de crédito com os atributos codificados e com a exclusão de 13 padrões, com o auxílio do *software WEKA* (326 x 54).	89,57

Através da Tabela 15, segunda simulação, pode-se verificar a importância da exclusão dos 13 padrões atípicos: a acurácia apresentou uma pequena melhora. A terceira simulação mostra que a codificação dos atributos, juntamente com a exclusão dos 13 padrões atípicos, fornece um acréscimo na acurácia. Finalmente, na quarta simulação, tem-se que codificar os atributos do problema, treinar uma RN e, finalmente, extrair regras de classificação a partir desta RN treinada para o problema em pauta, aumentou o percentual de acurácia em 8% (= 89,57 – 81,71) em relação à primeira simulação, alcançando quase 90%.

Do ponto de vista do usuário (gerente bancário, analista de crédito), sempre existe a vantagem em se utilizar esta ferramenta, porque a mesma mostra seus resultados (regras de classificação) em uma forma fácil de compreender, detalhando quais informações (atributos) das empresas analisadas foram mais relevantes para as suas classificações com acurácia satisfatória.

Desta forma, o gerente bancário pode checar se os resultados obtidos por esta técnica "conferem" (ou não) com a sua experiência e utilizá-la na análise de novas propostas de crédito com uma margem de segurança satisfatória como um apoio adicional as suas tomadas de decisões. Obviamente, é possível utilizar esta técnica a outros tipos de problemas como, por exemplo, no diagnóstico médico; na engenharia de avaliações; na avaliação de seguros; na análise de processos judiciais e na avaliação da qualidade de materiais.

Referências Bibliográficas

(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**(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**(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**(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**(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**(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. (1990). Learning Logical Definitions from Relations. *Machine Learning*, **5**, 239-266.

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

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

(16) 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.

(17) 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* (*17th Triennial Conference of the International Federation of Operational Research Societies hosted by INFORMS*), FB-20.

(18) 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**(1), 1-11.

(19) Witten, I.H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann Publishers, 2nd edition, San Francisco, California.

Anexo

Tabela A – Atributos considerados e as suas respectivas codificações "termômetro" e "*dummy*", tornando-os binários.

Atributos	Valores originais dos atributos	Intervalos	Número de padrões em cada intervalo	Entrada 1	Entrada 2	Entrada 3	Entrada 4	Número de entradas
1. Existência de restrições em nome da empresa **(atributo nominal)**				$\mathbf{I_1}$	---	---	---	1
	1 = sim	1 = sim	15	0	---	---	---	
	2 = não	2 = não	324	1	---	---	---	
2. Existência de restrições baixadas nos últimos 5 anos, em nome da empresa **(atributo nominal)**				$\mathbf{I_2}$	---	---	---	1
	1 = sim	1 = sim	14	0	---	---	---	
	2 = não	2 = não	325	1	---	---	---	
3. Tempo de conta na agência **(atributo ordinal)**	No. de Meses	No. de Meses		$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{I_5}$	$\mathbf{I_6}$	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. Setor de Atividade **(atributo nominal)**				$\mathbf{I_7}$	$\mathbf{I_8}$	---	---	2
	1 = comércio	1 = comércio	171	0	0	---	---	
	2 = indústria	2 = indústria	71	1	0	---	---	
	3 = serviços	3 = serviços	97	0	1	---	---	
5. Tempo de Atividade **(atributo ordinal)**	No. de Anos	No. de Anos		$\mathbf{I_9}$	$\mathbf{I_{10}}$	$\mathbf{I_{11}}$	$\mathbf{I_{12}}$	4
	1 = > 9	1 = > 9	76	1	1	1	1	
	2 = 6 a 9	2 = (6, 9]	33	0	1	1	1	
	3 = 3 a 5	3 = (3, 6]	86	0	0	1	1	
	4 = 1 a 2	4 = (1, 3]	36	0	0	0	1	
	5 = < 1	5 = < 1	108	0	0	0	0	
6. Número de Funcionários **(atributo ordinal)**	Valor Numérico	No. de Funcionários		$\mathbf{I_{13}}$	$\mathbf{I_{14}}$	$\mathbf{I_{15}}$	---	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. Sede da Empresa **(atributo nominal)**				$\mathbf{I_{16}}$	$\mathbf{I_{17}}$	---	---	2
	1 = próprio	1 = próprio	47	0	0			
	2 = alugado	2 = alugado	151	0	1			
	3 = cedido	3 = cedido	141	1	0			
8. Bairro **(atributo nominal)**				$\mathbf{I_{18}}$	---	---	---	1
	1 = centro	1 = centro	146	1	---	---	---	
	2 = outros	2 = outros	193	0	---	---	---	
9. Principais clientes **(atributo nominal)**				$\mathbf{I_{19}}$	$\mathbf{I_{20}}$	---	---	2
	1 = pessoas físicas	1 = p. físicas	307	1	0	---	---	
	2 = pessoas jurídicas	2 = p. jurídicas	32	0	1	---	---	
	3 = misto	3 = misto	0	0	0	---	---	
10. Faturamento Bruto Anual **(atributo ordinal)**	Valor Numérico	Valor Numérico (1.000)		$\mathbf{I_{21}}$	$\mathbf{I_{22}}$	$\mathbf{I_{23}}$	---	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	---	

Atributos	Valores originais dos atributos	Intervalos	Número de padrões em cada intervalo	Entrada	Entrada	Entrada	Entrada	Número de entradas
11. Cliente em outro banco **(atributo nominal)**				I_{24}	---	---	---	1
	1 = sim	1 = sim	143	1	---	---	---	
	2 = não	2 = não	196	0	---	---	---	
12. Bens Imóveis **(atributo ordinal)**	Valor Numérico	Valor Numérico (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. Bens Móveis **(atributo ordinal)**	Valor Numérico	Valor Numérico (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. Seguro Empresarial **(atributo nominal)**				I_{31}	---	---	---	1
	1 = sim	1 = sim	126	1	---	---	---	
	2 = não	2 = não	213	0	---	---	---	
15. Aplicações Financeiras na agência **(atributo ordinal)**		Valor Numérico (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 = não	5 = 0	285	0	0	0	0	
16. Vendas a Prazo **(atributo nominal)**				I_{36}	---	---	---	1
	1 = < 20%	1 = [0, 20)	74	1	---	---	---	
	2 = ≥ 20%	2 = ≥ 20	265	0	---	---	---	
17. Experiência de Crédito na agência **(atributo nominal)**	Anos	Anos		I_{37}	I_{38}	---	---	2
	1 = > 2	1 = > 2	75	0	0	---	---	
	2 = ≤ 2	2 = (0, 2]	264	1	0	---	---	
	3 = não	3 = [0]	0	0	1	---	---	
18. Histórico da Conta Corrente **(atributo nominal)**				I_{39}	I_{40}	I_{41}	---	3
	1 = normal	1 = normal	240	0	0	0	---	
	2 = cheques devolvidos	2 = cheques devolvidos	2	1	0	0	---	
	3 = cliente novo	3 = cliente novo	88	0	1	0	---	
	4 = pequenos atrasos freqüentes	4 = pequenos atrasos freqüentes	9	0	0	1	---	
19. Sócios da empresa possuem restrições **(atributo nominal)**				I_{42}	---	---	---	1
	1 = sim	1 = sim	9	0	---	---	---	
	2 = não	2 = não	330	1	---	---	---	
20. Sócios da empresa tiveram restrições baixadas nos últimos 5 anos **(atributo nominal)**				I_{43}	---	---	---	1
	1 = sim	1 = sim	51	0	---	---	---	
	2 = não	2 = não	288	1	---	---	---	

Atributos	Valores originais dos atributos	Intervalos	Número de padrões em cada intervalo	Entrada	Entrada	Entrada	Entrada	Número de entradas
21. Sociedade entre Cônjuges **(atributo nominal)**				$\mathbf{I_{44}}$	---	---	---	1
	1 = sim	1 = sim	116	0	---	---	---	
	2 = não	2 = não	223	1	---	---	---	
22. Existência de Bens Imóveis em nome dos sócios **(atributo ordinal)**	Valor Numérico	Valor Numérico (1.000)		$\mathbf{I_{45}}$	$\mathbf{I_{46}}$	$\mathbf{I_{47}}$	---	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. Existência de Bens Móveis em nome dos sócios **(atributo ordinal)**	Valor Numérico	Valor Numérico (1.000)		$\mathbf{I_{48}}$	$\mathbf{I_{49}}$	$\mathbf{I_{50}}$	---	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. Risco atribuído pelo Banco **(atributo nominal)**	Escala	Escala		$\mathbf{I_{51}}$	$\mathbf{I_{52}}$	$\mathbf{I_{53}}$	$\mathbf{I_{54}}$	4
	1=A (melhor)	1=A (melhor)	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 (pior)	5=E (pior)	3	0	0	0	0	
Total de Atributos: 24	---	---	---	---	---	---	---	54