Universidade de São Paulo
Escola Superior de Agricultura "Luiz de Queiroz"

Técnicas de Data Mining na aquisição de clientes para financiamento de Crédito Direto ao Consumidor - CDC

Adriana Maria Marques da Silva

Dissertação apresentada para obtenção do título de Mestre em Ciências. Área de concentração: Estatística e Experimentação Agronômica

Piracicaba
2012

Adriana Maria Marques da Silva
Bacharel em Estatística

Técnicas de Data Mining na aquisição de clientes para financiamento de Crédito Direto ao Consumidor - CDC

Orientador:
Prof. Dr. **CARLOS TADEU DOS SANTOS DIAS**

Dissertação apresentada para obtenção do título de Mestre em Ciências. Área de concentração: Estatística e Experimentação Agronômica

Piracicaba
2012

Dados Internacionais de Catalogação na Publicação
DIVISÃO DE BIBLIOTECA - ESALQ/USP

Silva Adriana Maria Marques da
Técnicas de Data Mining na aquisição de clientes para financiamento de Crédito Direto ao Consumidor - CDC / Adriana Maria Marques da Silva.- - Piracicaba, 2012.
182 p: il.

Dissertação (Mestrado) - - Escola Superior de Agricultura "Luiz de Queiroz", 2012.

1. Árvore de decisão 2. Crédito direto ao consumidor 3. Financiamento 4. Mineração de dados 5. Redes neurais 6. Regressão logística I. Título

CDD 332.743
S586t

DEDICATÓRIA

Aos meus pais,

***Maria Lailda Marques** e*

Manoel Carlos Santana da Silva

Com amor, DEDICO.

AGRADECIMENTOS

Primeiramente, aos meus familiares, Maria Lailda Marques, Manoel Carlos Santana da Silva, João Paulo Marques da Silva, por estarem ao meu lado, mesmo a quilômetros de distância durante um período desta jornada. Em especial, à minha mãe, pelo carinho e bondade na correção dos meus trabalhos. Também aos meus primos, tios e tias pela confiança e carinho.

Aos meus amigos que trabalho, que muito ajudaram nesta jornada final, por me respeitarem e incentivarem: Andreia Santos, Lyse Nogueira, Daniel Ferreira, Danylo Moya, Alison Ishii, Ronaldo Aoki, Daniela Souza, Carlos Miranda, Reginaldo Perseghetti, Daniel Martins, Bruno Galhardo, Rafael Paes, Rafael Amaro e Cleria Barichello.

Ao meu primeiro e melhor chefe, Ivan Pezzoli, por confiar e me apoiar inúmeras vezes, sempre me motivando e me entusiasmando em toda atividade que eu fizesse.

Ao SAS, pela compreensão e apoio, especialmente ao meu chefe Rodolpho Marcelino e Wander Vasconcelos.

Aos amigos de departamento, Kelli Gonçalves, Thais Cardoso e Otavio Menezes, pela ajuda, compreensão e admiração.

Ao Alexandre Gomes e Henrique Lima, pela ajuda e camaradagem nos problemas técnicos.

À professora Édina, pelos conhecimentos compartilhados e pela amizade.

Aos meus colegas de pós-graduação, Marcelino Rosa, Everton Batista, Cristiane Rodrigues, Josiane Rodrigues, Lilian, Tiago Oliveira, Ana Patricia Peixoto, pela ajuda, conversas risos, almoços, horas de estudo e pelo divertimento.

Á minha amiga de casa, Priscila Neves Faria, pela amizade, companherismo e ajuda.

A minha amiga, Gláucia Tatiana Ferrari, pela amizade, carinho, dedicação, horas de estudo, viagens, divertimento e companherismo.

Ao amigo Ricardo Alves de Olinda, pela ajuda, amizade e dedicação.

Ao Professor Dr. Carlos Tadeu dos Santos Dias, pela orientação e confiança em mim depositadas. Por todo incentivo, críticas e sugestões que foram fundamentais para o desenvolvimento desta pesquisa e para o meu crescimento profissional.

Aos professores de graduação pela formação e por toda ajuda.

Aos docentes do Programa de Pós-Graduação em Estatística e Experimentação Agronômica que auxiliaram em minha formação.

Aos funcionários do Departamento de Ciências Exatas da ESALQ/USP, Eduardo Bonilha e Jorge Alexandre Wiendl, pelo apoio técnico, às secretárias Luciane Brajão e Solange de Assis Paes Sabadin, pelo apoio acadêmico.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pela concessão da bolsa de estudos para a realização deste trabalho.

Aqueles que contribuiram direta ou indiretamente para a realização deste estudo e, por fim, a todos que confiaram em mim.

MUITO OBRIGADA!

SUMÁRIO

RESUMO....11
ABSTRACT....13
1 ESTRUTURA DA DISSERTAÇÃO....15
2 INTRODUÇÃO....17
2.1 Justificativa....20
2.1.1 Justificativa Teórica....21
2.1.2 Justificativa Prática....22
3 REVISÃO BIBLIOGRÁFICA....25
3.1 Regressão Logística....25
3.1.1 Função de ligação Logito....27
3.1.2 Função de ligação Probito....27
3.1.3 Função de ligação Complementar Log-Log (Cloglog)....28
3.1.4 Regressão Logística Simples....29
3.1.4.1 Teste de Significância dos Coeficientes....34
3.1.5 Regressão Logística Múltipla....40
3.1.5.1 Teste de significância dos parâmetros do modelo....43
3.1.5.2 Estimação do Intervalo de Confiança dos Parâmetros....44
3.1.5.3 Razão de Chance....44
3.1.5.4 Seleção de variáveis....48
3.1.5.5 Medidas de qualidade do ajuste....51
3.1.5.6 Estatísticas Pearson Qui-Quadrado e Deviance....51
3.1.5.7 Teste de Hosmer-Lemeshow para adequação do modelo....53
3.1.5.8 Matriz de confusão....54
3.1.5.9 Área abaixo da curva ROC....56
3.2 Árvore de Decisão....58

3.2.1 Utilização da Árvore de Decisão 62

3.2.1.1 Seleção de variáveis 62

3.2.1.2 Importância da variável 63

3.2.1.3 Detecção de interação 63

3.2.1.4 Valores faltantes 64

3.2.1.5 Interpretação do modelo 65

3.2.1.6 Modelagem preditiva 66

3.2.2 Como construir uma árvore de decisão 67

3.2.2.1 Como uma regra é criada usando uma divisão binária 67

3.2.2.2 Mensurar a importância de uma divisão quando a variável resposta é binária 68

3.2.2.2.1 Grau de separação 69

3.2.2.2.2 Redução da impureza como medida para mensurar a importância de uma quebra 71

3.2.2.2.2.1 Índice de impureza GINI 72

3.2.2.2.2.2 Entropia 72

3.2.2.3 Mensurar a importância de uma divisão quando a variável resposta é categórica 73

3.2.2.4 Ajustes para o valor-p quando as variáveis explicativas têm diferentes níveis 73

3.2.2.4.1 Ajuste de Bonferroni 74

3.2.2.4.2 Ajuste de Profundidade 75

3.2.3 Controlar o crescimento da árvore: regras de parada 76

3.2.4 Poda: a seleção da árvore do tamanho certo 77

3.2.5 Algoritmos Conhecidos 83

3.2.5.1 ID3 84

3.2.5.2 C4.5 84

3.2.5.3 CART.....85
3.2.5.4 CHAID.....86
3.2.5.5 Algorítmos SAS.....86
3.3 Rede Neural.....87
3.3.1 O cérebro humano.....88
3.3.2 Os Neurônios.....89
3.3.3 A comunicação entre os Neurônios.....89
3.3.4 O modelo MCP (McCulloch e Pitts).....91
3.3.5 Funções de Ativação.....92
3.3.6 Principais arquiteturas de RNAs.....94
3.3.7 Aprendizado.....98
3.3.7.1 Aprendizado supervisionado.....99
3.3.7.2 Correção de erros.....100
3.3.7.3 Aprendizado por reforço.....102
3.3.7.4 Aprendizado não supervisionado.....103
3.3.8 Perceptron.....103
3.3.8.1 O algorítmo de aprendizado do Perceptron.....104
3.3.8.2 Implementação do algorítmo de aprendizado do Perceptron.....105
3.3.8.3 Considerações sobre o aprendizado do Perceptron.....106
3.3.9 Redes Perceptron de Múltiplas Camadas (MLP).....106
3.3.9.1 A arquitetura de uma rede Perceptron de Múltiplas Camadas (MLP).....108
3.3.9.2 Número de camadas.....109
3.3.9.3 Número de neurônios.....110
3.3.9.4 Treinamento de Redes MLP.....110
3.3.9.5 Camada de saída.....114
3.3.9.6 Camada escondida.....115
4 MATERIAL E MÉTODOS.....119

4.1 Descrição do conjunto de dados 120

4.2 Sistema computacional SAS 122

5 RESULTADOS 125

6 CONCLUSÃO 137

REFERÊNCIAS 139

APÊNDICES 143

RESUMO

Técnicas de Data Mining na aquisição de clientes para financiamento de Crédito Direto ao Consumidor – CDC

O trabalho busca dissertar sobre as técnicas de data mining mais difundidas: regressão logística, árvore de decisão e rede neural, além de avaliar se tais técnicas oferecem ganhos financeiros para instituições privadas que contam com processos ativos de conquista de clientes. Uma empresa do setor financeiro será utilizada como objeto de estudo, especificamente nos seus processos de aquisição de novos clientes para adesão do Crédito Direto ao Consumidor (CDC). Serão mostrados os resultados da aplicação nas três técnicas mencionadas, para que seja possível verificar se o emprego de modelos estatísticos discriminam os clientes potenciais mais propensos dos menos propensos à adesão do CDC e, então, verificar se tal ação impulsiona na obtenção de ganhos financeiros. Esses ganhos poderão vir mediante redução dos custos de marketing abordando-se somente os clientes com maiores probabilidades de responderem positivamente à campanha. O trabalho apresentará o funcionamento de cada técnica teoricamente, e conforme os resultados indicam, data mining é uma grande oportunidade para ganhos financeiros em uma empresa.

Palavras-chave: Mineração de Dados; Regressão Logística; Árvore de Decisão; Rede Neural; Crédito Direto ao Consumidor

ABSTRACT

Data Mining Techniques to acquire new customers for financing of Consumer Credit

The paper intends to discourse about most widespread data mining techniques: logistic regression, decision tree and neural network, and assess whether these techniques provide financial gains for private institutions that have active processes for business development. A company of the financial sector is used as object of study, specifically in the processes of acquiring new customers for adhesion to consumer credit (in Brazil CDC). This research will show the results of the three above mentioned techniques, to check whether the statistical models point out relevant differences between prospects´ intentions to adhere to consumer credit. In the meantime, the techniques are checked whether they leverage financial gain. These gains are expected to came from better focused and directed marketing efforts. The paper presents the operation of each technique theoretically, and as the results indicate, data mining is a great opportunity for a company boost profits.

Keywords: Data Mining; Logistic Regression; Decision Tree; Neural Network; Consumer Credit

1 ESTRUTURA DA DISSERTAÇÃO

A presente dissertação encontra-se dividida nas seguintes partes: Introdução; Justificativa; Desenvolvimento; Resultados e Conclusões. No capítulo 2, Introdução, apresenta-se a contextualização do estudo, além das justificativas teóricas e práticas. No capítulo 3, Revisão de Literatura, são explicadas todas as técnicas utilizadas na aplicação e delineia-se o procedimento utilizado para a obtenção dos objetivos. Neste capítulo são apresentados fundamentos teóricos sobre cada abordagem. No Capítulo 4, Resultados, apresentam-se a descrição do estudo de caso realizado e os modelos obtidos, além das comparações e motivos pelos quais o modelo foi escolhido. No capítulo 5 são apresentadas as conclusões finais do trabalho em decorrência dos resultados obtidos nesta pesquisa.

2 INTRODUÇÃO

Segundo Dilly (2010), a quantidade de informação no mundo dobra a cada 20 meses e o tamanho e a quantidade dos bancos de dados crescem com velocidade ainda maior. Como a quantidade de informação disponível aumenta a cada dia, é essencial tentar aproveitar o máximo possível dessa informação. A forma mais sensata de utilizar essas informações é verificar se há algum conhecimento, padrão ou alguma direção dentro delas.

O banco de dados de um supermercado, por exemplo, contém cada transação realizada por cada cliente. Com todos esses registros, podem-se descobrir padrões nas compras, criar grupos de cliente com um mesmo hábito, descobrir produtos que impulsionam a venda de outros e outros achados. Com todas essas descobertas, pode-se otimizar os resultados financeiros do supermercado.

O processo de explorar grandes quantidades de dados à procura de padrões consistentes, como regras de associação ou sequências temporais, para detectar relacionamentos sistemáticos entre variáveis, é chamado mineração de dados, em português, ou *Data Mining*, em inglês.

Data Mining é parte de um processo maior conhecido como Descoberta de Conhecimento em Base de Dados (*KDD - Knowledge Discovery in Databases*) e se constitui por um leque de técnicas que por meio do uso de algoritmos de aprendizagem ou classificação baseados em estatística, inteligência artificial e aprendizado de máquinas, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões e auxiliando na descoberta de conhecimento.

O ser humano sempre aprendeu observando padrões, formulando hipóteses e testando-as para descobrir regras. A novidade da era do computador é o grande volume de dados que não pode mais ser examinado à procura de padrões em um prazo de tempo razoável. A solução é instrumentalizar o próprio computador para detectar relações que sejam novas e úteis. A mineração de dados surge para essa finalidade e pode ser aplicada tanto para a pesquisa científica como para impulsionar a lucratividade de uma empresa com experiência, inovadora e competitiva.

O processo *KDD* é constituído de várias etapas, sendo a etapa mais importante o *Data Mining*. Como se pode notar pela Figura 1, o processo *KDD* passa por cinco fases. A primeira fase para a descoberta de conhecimento é a seleção dos dados. Nessa fase é importante ter conhecimento de onde se pretende chegar.

Como é de conhecimento geral, em toda análise quantitativa, a qualidade dos dados é essencial para a obtenção de resultados confiáveis. Segundo Diniz e Louzada-Neto (2000), dados limpos e compreensíveis são requisitos básicos para o sucesso do *Data Mining*. Com isso é essencial que a segunda fase, Pré-Processamento, seja realizada com sucesso. Esse passo leva até 80% do tempo necessário para todo o processo, devido às dificuldades de integração de bases de dados heterogêneas (MANNILA, 1996).

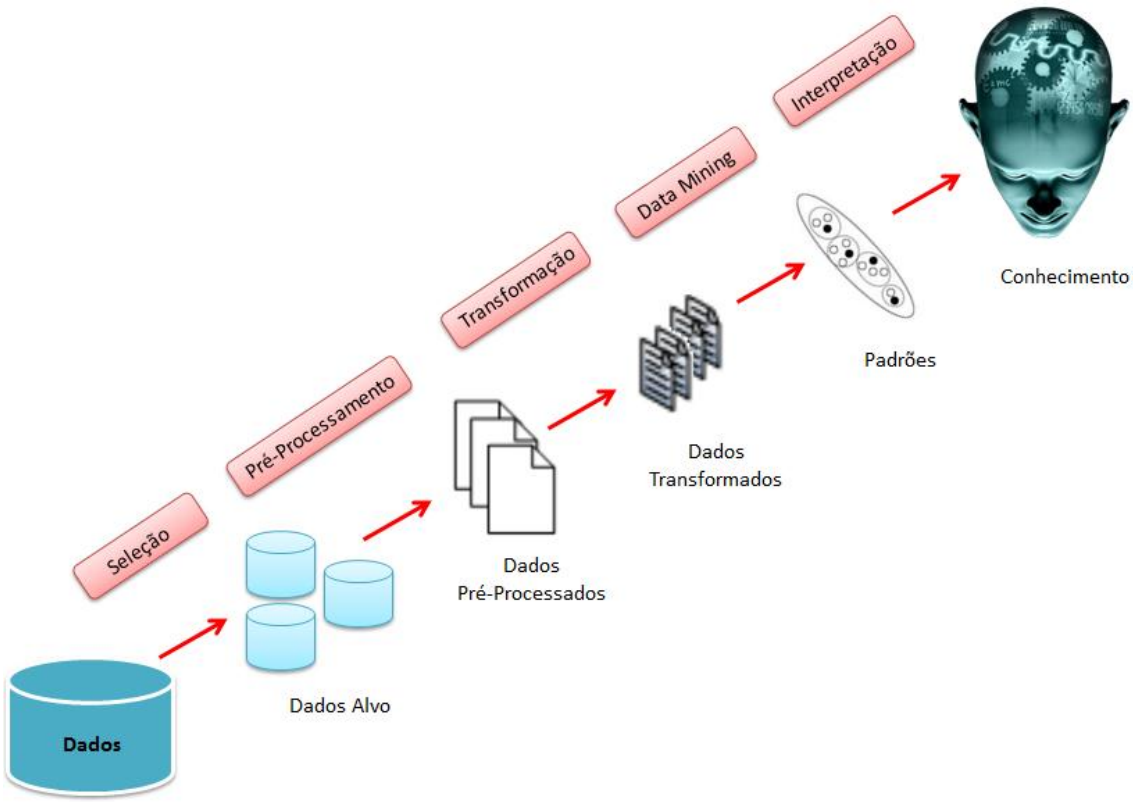

Figura 1 - Etapas que constituem o processo de *KDD*

Os dados pré-processados devem passar por outra transformação, que os armazena adequadamente, visando facilitar o uso das técnicas de *Data Mining*. O objetivo do passo seguinte, *Data Mining*, é a aplicação de técnicas de mineração nos dados pré-processados, o que envolve ajuste de modelos e/ou determinação de

características nos dados. Em outras palavras, exige o uso de métodos inteligentes para a extração de padrões ou conhecimentos dos dados.

No passo final, Interpretação e Análise, existe a possibilidade de retorno a qualquer um dos passos anteriores, dependendo dos resultados e das necessidades exigidas pelo objetivo. Com isso, o resultado final não depende apenas da etapa do *Data Mining*, depende de todo processo: consistência da base de dados (*Data Cleaning*), escolha das variáveis e por último a técnica utilizada.

Teoricamente, *Data Mining* pode ser aplicado em qualquer área de conhecimento. No entanto, existem áreas em que o uso dessa técnica é mais frequente. Conforme Fayyad, Piatetski-Shapiro e Smyth (1996), essas áreas são:

Marketing: redução dos custos com o envio de correspondências através de sistemas de mala direta a partir da identificação de grupos de clientes potenciais. Um exemplo disso é o que o Pão de Açúcar fez com a utilização do SAS. O mercado passa a oferecer um cartão de desconto em troca de informações pessoais que serão utilizadas como entrada para o modelo computacional de *Data Mining*. Com essas informações consegue-se criar grupos de clientes e consequentemente, pode-se oferecer o produto certo para pessoa certa, aumentando a probabilidade de venda.

Detecção de fraude: reclamações indevidas de seguro, chamadas clonadas de telefones celulares, compras fraudulentas com cartão de crédito, fraude na composição quimica do leite e nomes duplicados em sistemas de Previdência Social.

Investimento: diversas empresas têm usado técnicas de mineração de dados para obter ganhos financeiros. São usados especialmente modelos de redes neurais no mercado de ações e na previsão da cotação do ouro e do dólar.

Produção: empresas desenvolvem sistemas para detectar e diagnosticar erros na fabricação de produtos. Estas falhas são normalmente agrupadas por técnicas de Análise de Agrupamentos.

As técnicas de mineração podem ser aplicadas a tarefas (neste contexto, um problema de descoberta de conhecimento a ser solucionado) como

associação, classificação, predição/previsão, sumarização e clusterização. A seguir uma descrição resumida de cada uma delas (FAYYAD; STOLORZ, 1997):

Associação: consiste em determinar quais fatos ou objetos tendem a ocorrer juntos em um mesmo evento ou em uma mesma transação.

Classificação: consiste em construir um modelo que possa ser aplicado a dados não classificados visando categorizar os objetos em classes. Associa ou classifica um item a uma ou várias classes categóricas pré-definidas. Uma técnica estatística apropriada para classificação é a análise discriminante. Os objetivos dessa técnica envolvem a descrição gráfica ou algébrica das características diferenciais das observações de várias populações, além da classificação das observações em uma ou mais classes predeterminadas.

Predição/Previsão: predição é usada para definir um provável valor para uma ou mais variáveis. A previsão é utilizada quando se têm séries temporais (dados organizados cronologicamente), como por exemplo a previsão da cotação de uma ação na bolsa de valores.

Agrupamentos ou Clusterização: é um processo de partição, que visa dividir uma população em subgrupos mais heterogêneos entre si. É diferente da tarefa de classificação, pois não existem classes predefinidas, os objetos são agrupados de acordo com a similaridade. Os *clusters* são definidos por meio do agrupamento de dados baseados em medidas de similaridade ou modelos probabilísticos. A análise de *cluster* (ou agrupamento) é uma técnica que visa detectar a existência de diferentes grupos dentro de um determinado conjunto de dados e, em caso de sua existência, determinar quais são eles.

A Mineração de Dados fornece uma série de idéias e técnicas para uma vasta variedade de profissões. Estatísticos, pesquisadores de Inteligência Artificial e administradores de bancos de dados que usam técnicas diferentes para chegar a um mesmo fim, ou seja, a informação.

2.1 Justificativa

Qualquer técnica estatística empregada corretamente pode reverter em grandes mudanças para qualquer objetivo. Bancos de dados são a fonte para

qualquer incremento, novos conhecimentos e descobertas. Empresas capazes de estudar e entender seu próprio negócio conseguem visualizar novas oportunidades e com isso conseguem uma melhor posição no mercado. O tema dessa dissertação é muito abordado pelas empresas e merece destaque no meio acadêmico, para que as técnicas sejam aperfeiçoadas e que com isso exista um link entre universidade e empresa. A seguir, estão descritas justificativas para este estudo, tanto na parte acadêmica, como no mundo corporativo.

2.1.1 Justificativa Teórica

As técnicas estatísticas e computacionais são grandes aliadas do conhecimento e das descobertas. Desde análises descritivas até modelos mais sofisticados, o poder das melhores decisões, deveriam ser baseados nestes resultados. Desde a década de 70, vem ocorrendo debates sobre as razões para a baixa utilização de modelos pelos gestores de empresas, apesar de ser comprovada a eficácia em diversos modelos disponíveis (LITTLE, 2004). Ainda existe uma certa resistência por parte dos executivos tomadores de decisão, porém, cada dia fica mais nítida a necessidade de um estudo para o conhecimento do negócio em questão.

Segundo Leeflang e Wittink (2000) um modelo é a representação dos elementos mais importantes da percepção de um sistema do mundo real, por isso, a necessidade das pesquisas que envolvem a elaboração de modelos sejam realizadas em parcerias entre a academia e as empresas, possibilitando aos acadêmicos o acesso a um grande conjunto de informações reais e ao mesmo tempo que os modelos gerados possam efetivamente contribuir com os gestores, auxiliando nos processos de tomada de decisão (LEEFLANG; WITTINK, 2000).

A melhor compreensão de como se comporta o negócio de uma empresa, auxiliará a determinar estratégias mais eficazes, bem como possibilitará às empresas adotantes a aprimorar o processo de avaliação e escolha de produtos e serviços, bem como estratégias de *marketing* e estudos de riscos. A importância da realização de estudos no mercado corporativo é o de possibilitar o desenvolvimento e a melhora da competitividade das empresas nacionais.

2.1.2 Justificativa Prática

Como mencionado na introdução, 80% do tempo de uma análise de *Data Mining* é usado pelo processamento dos dados e manipulação dos mesmos. Uma preocupação, apontada pelas empresas que adotaram sistemas de coleta de informações de clientes, está no desafio em transformar estes dados em informações que auxiliem no processo decisório, o que de, certa forma, vem trazendo questionamentos quanto à viabilidade de coletar tantas informações, considerando os altos custos envolvidos comparados aos benefícios gerados, conforme abordam Rigby e Ledingham (2004). Os autores afirmam que a necessidade do negócio é a prioridade maior da empresa e deve prevalecer em relação à capacidade tecnológica.

Muitas empresas armazenam milhares de registros em suas bases de dados, como informações relacionadas ao cliente, histórico de comportamento com seus produtos, entre outros. A imperícia (inabilidade) em obter informações sobre estes dados impede que a organização obtenha conhecimento valioso e aplicável (SUMATHI; SIVANANDAM, 2006). Neste contexto, a utilização de técnicas de mineração de dados mostra-se como uma oportunidade para a realização de estudos acadêmicos e, também, para a geração de novos modelos para as organizações. Este estudo pretende auxiliar na compreensão das técnicas de mineração de dados, que são técnicas de extração de conhecimento de grandes quantidades de dados (HAN; KAMBER, 2006).

A aplicação de técnicas de mineração de dados pode auxiliar na elaboração de novos modelos contextualizados a casos brasileiros, mostrando o potencial da utilização destas técnicas para a gestão de serviços e consumidores. Com o advento de novas interfaces gráficas que facilitam o uso das ferramentas, associado à grande quantidade de informações disponibilizadas, a mineração de dados representa uma grande oportunidade para a realização de estudos e modelos em administração para melhores tomadas de decisão.

A escolha do tema desta dissertação se deu pelo fato de que as técnicas de *Data Mining* são técnicas emergentes, sendo incentivada a sua utilização por diversos autores (HAIR et al., 2005; GUPTA et al., 2006), além de ser

recomendada a utilização de mineração de dados de modo a abrir novas perspectivas para o mercado corporativo (GUPTA et al., 2006).

"Um modelo deve prever, no mínimo, os fatos que o originaram. Um bom modelo é aquele que tem a capacidade de previsão de novos fatos" (BASSANEZI, 2004), sendo assim, a grande preocupação deste trabalho é que o modelo desenvolvido possa ser aplicado no mundo corporativo com o objetivo de ajudar na montagem de estratégia da empresa ou diminuição dos prejuízos.

Apesar de muitos modelos de marketing serem robustos e comprovadamente eficazes, observa-se ainda a pouca utilização de modelos acadêmicos pelas empresas, sendo que Martinez-Lopez e Casillas (2009) recomendam um esforço da academia para reduzir este distanciamento, de modo que os modelos possam ser utilizados com sucesso e aplicados nas atividades do dia-a-dia das empresas. Para Little (2004), os modelos não são muito utilizados pelos gestores por ser difícil de encontrar um bom modelo que inclua as variáveis de interesse do gestor, pela dificuldade de se realizar uma boa parametrização e pelo fato de os gestores não compreenderem os modelos. Para que um modelo seja utilizado por gestores, Little (2004) ressalta que o modelo deverá ser: (1) simples; (2) robusto; (3) fácil de controlar; (4) adaptativo; (5) completo nos elementos importantes e (6) fácil de comunicar. A simplicidade facilita a compreensão. A robustez previne a inconsistência e evita resultados absurdos. A facilidade de controle implica a transparência do modelo, de modo que o gestor saiba o que está ocorrendo. A adaptabilidade permite a inserção no modelo de novas alterações do ambiente. A requisição de ser completo permite que o gestor possa inserir os requisitos/variáveis desejados. A facilidade de comunicação é desejável para permitir a difusão do conhecimento.

Outra dificuldade para o uso de modelos pelos gestores é a necessidade da customização, uma vez que cada universo de produtos, serviços e clientes possui características próprias, que dificilmente são contempladas por um modelo genérico. Na construção do modelo optou-se por avaliar o melhor desempenho do modelo feito por três das principais técnicas de Data Mining: Regressão Logística, Árvore de Decisão e Redes Neurais.

Com este estudo, objetiva-se colaborar com os estudos científicos brasileiros na área de mineração de dados. O mercado corporativo brasileiro ainda necessita de pesquisas que possam aprimorar a gestão e possibilitar a obtenção da excelência em prestação de serviços, o que poderá abrir novas possibilidades de atuação. O estudo também pode ajudar a conscientizar os gestores de empresas de serviços da importância da utilização de boas práticas de gestão, uso da inteligência analítica.

3 REVISÃO BIBLIOGRÁFICA

Neste capítulo exlica-se as técnicas utilizadas na aplicação prática e delineia-se o procedimento utilizado para a obtenção dos objetivos.

3.1 Regressão Logística

A regressão logística surgiu em 1789, com os estudos de crescimento populacional de Malthus. Segundo Cramer (2002), 40 anos depois, Alphonse Quetelet e Pierre- François Verhust, recuperaram a idéia de Malthus para descrever o crescimento populacional na França, Bélgica e Rússia. No entanto, só em 1845, Pierre- François Verhust publicou a formulação utilizada nos estudos de crescimento da população a que chamou de curva logística.

Ainda no séc. XIX, a mesma função foi utilizada para descrever as reações químicas autocatalíticas, porém se manteve apagada na maior parte do século e só foi redescoberto em 1920 por Raymond Pearl, discípulo de Karl Pearson, e Lowell Reed que o aplicaram igualmente ao estudo do crescimento da população dos Estados Unidos da América.

Os modelos logísticos surgiram da necessidade de modelos mais satisfatórios para dados qualitativos e pela dificuldade encontrada ao aplicar a Regressão Linear para variáveis dependentes qualitativas. O modelo de regressão logística é o principal modelo de dados binários, que são aqueles em que a variável de interesse (resposta) assume dois valores possíves. Como existem muitas situações práticas onde as variáveis binárias são encontradas, o estudo sobre o assunto é bastante vasto.

A regressão logística é muito semelhante à regressão linear. Em ambos os casos utiliza-se uma ou mais variáveis explicativas (X) para predizer o valor de uma variável resposta (Y). Entretanto, na regressão logística (ou modelo binário), a variável resposta (Y) possui apenas dois valores possíveis.

Usualmente adota-se o valor 1 como o resultado mais importante da resposta ou aquele que se pretende relacioanar ao acontecimento de interesse (conhecido como “sucesso”) e o valor 0 ao “fracasso” (resultado complementar).

A regressão logística trabalha com chances ao invés de proporções. As chances correspondem à razão entre proporções de dois resultados possíveis. Se p é a probabilidade de sucesso, então $1 - p$ é a probabilidade de fracasso, ou seja:

$$p = P(Y = 1),\ 1 - p = P(Y = 0) \text{ e } chance = \frac{p}{1-p} = \frac{probabilidade\ de\ sucesso}{probabilidade\ de\ fracasso}$$

e sendo p uma probabilidade, o valor previsto deve ser qualquer número limitado entre 0 e 1.

A Regressão Logística modela a média p em termos de uma ou mais variáveis explicativas x. Pode-se tentar relacionar p e x como uma regressão linear:

$$p = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \quad (1)$$

no entanto não seria um bom modelo, pois sempre que $\beta_j \neq 0\ (com\ j = 0, 1, \ldots, k)$, valores extremos de x fornecerão valores para $\beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$ que ficariam fora do conjunto de valores possíveis para $p\ (0 \leq p \leq 1)$.

Por isso, o modelo de regressão logística remove essa dificuldade determinando uma transformação t de modo que $t(p(x))$ pertença ao intervalo $]-\infty; \infty[$, podendo assim ser modelada pela função linear como na eq. (1). A função t é denominada como função de ligação (ISHIKAWA, 2007).

De acordo com Sarma (2009), algumas transformações podem desempenhar esse papel. Assumindo que a variável estimada é denotada por $\hat{y}$ para cada linha no banco de dados, sabe-se que o valor de $\hat{y}$ depende de todas as variáveis usadas para estimá-lo (representadas pelas variáveis independentes x_i), sendo assim, sempre que se tem todas as observações de x_i preenchidas, tem-se $\hat{y} > 0$, ou seja:

$$\hat{\boldsymbol{y}} = \boldsymbol{\beta}'\boldsymbol{x} + \boldsymbol{\varepsilon} \quad (2)$$

em que $\boldsymbol{\beta}$ é o vetor de coeficientes, $\boldsymbol{x}$ é o vetor de variáveis independentes e $\boldsymbol{\varepsilon}$ é uma variável aleatória. Diferentes suposições sobre a distribuição da variável aleatória $\boldsymbol{\varepsilon}$ dá origem a diferentes funções de ligação. Sendo que a probabilidade de resposta é:

$$P(y = 1\ |\boldsymbol{x}) = P(\hat{y} > 0\ |\boldsymbol{x}) = P(\boldsymbol{\beta}'\boldsymbol{x} + \boldsymbol{\varepsilon} > 0) = P(\boldsymbol{\varepsilon} > -\boldsymbol{\beta}'\boldsymbol{x}) = 1 - F(-\boldsymbol{\beta}'\boldsymbol{x})$$

em que $F(.)$ é a função da distribuição acumulada da variável aleatória $\boldsymbol{\varepsilon}$.

3.1.1 Função de ligação Logito

Segundo Sarma (2009), a função de distribuição acumulada será:

$$F(-\boldsymbol{\beta}'x) = \frac{1}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}$$

e, com isso, tem-se que:

$$1 - F(-\boldsymbol{\beta}'\boldsymbol{x}) = 1 - \frac{1}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}$$

$$= \frac{1+e^{\boldsymbol{\beta}'\boldsymbol{x}} - 1}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}$$

$$= \frac{e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}$$

Por isso, a probabilidade de resposta é calculada como

$$P(y=1\,|\boldsymbol{x}) = 1 - F(-\boldsymbol{\beta}'\boldsymbol{x}) = \frac{e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}} \quad (3)$$

e

$$1 - \ P(y=1\,|\boldsymbol{x}) = 1 - \frac{e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}$$

$$= \frac{1+e^{\boldsymbol{\beta}'\boldsymbol{x}} - e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}} = \frac{1}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}} \quad (4)$$

Das eq. (3) e (4) pode-se notar que a função de ligação é:

$$\log\left(\frac{P(y=1\,|\boldsymbol{x})}{1-P(y=1\,|\boldsymbol{x})}\right) = log\left(\frac{\frac{e^{\boldsymbol{\beta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}}{\frac{1}{1+e^{\boldsymbol{\beta}'\boldsymbol{x}}}}\right) = log\left(e^{\boldsymbol{\beta}'\boldsymbol{x}}\right) = \boldsymbol{\beta}'\boldsymbol{x}$$

O $\boldsymbol{\beta}'\boldsymbol{x}$ é chamado de preditor linear, uma vez que é uma combinação linear das variáveis x_i de entrada.

3.1.2 Função de ligação Probito

Segundo Sarma (2009), na função de ligação probito assume-se que a variável aleatória $\boldsymbol{\varepsilon}$ na eq. (2) tem uma distribuição normal com média 0 e desvio padrão igual a 1. Neste caso tem-se que:

$$P(y = 1\,|\boldsymbol{x}) = 1 - F(-\boldsymbol{\beta}'\boldsymbol{x}) = F(\boldsymbol{\beta}'\boldsymbol{x})$$

devido à semetria da distribuição de probabilidade normal, em que

$$F(\boldsymbol{\beta}'\boldsymbol{x}) = \int_{-\infty}^{\boldsymbol{\beta}'\boldsymbol{x}} \frac{1}{\sqrt{2\pi}} e^{-u^2} du$$

Sendo assim,

$$\Phi^{-1}(P(y = 1\,|\boldsymbol{x})) = \boldsymbol{\beta}'\boldsymbol{x}$$

então

$$P(y = 1\,|\boldsymbol{x}) = \Phi^{-1}(\boldsymbol{\beta}'\boldsymbol{x})$$

Definido que Φ^{-1} é a inversa da distribuição de probabilidade normal acumulada.

3.1.3 Função de ligação Complementar Log-Log (Cloglog)

Segundo Sarma (2009), na função de ligação log-log a probabilidade de resposta é calculada como

$$P(y = 1\,|\boldsymbol{x}) = 1 - e^{-e^{\boldsymbol{\beta}'x}}$$

e

$$1 - P(y = 1\,|\boldsymbol{x}) = 1 - 1 + e^{-e^{\boldsymbol{\beta}'x}} = e^{-e^{\boldsymbol{\beta}'x}}$$

Com isso, a função de ligação é definida por:

$$log\big(-log(1 - P(y = 1\,|\boldsymbol{x}))\big) = \boldsymbol{\beta}'\boldsymbol{x}$$

Em estudos de dados binários que envolvem uma variável respotas Y binária e uma ou mais covariáveis X, a probabilidade de sucesso é:

$$P(y = 1\,|x_i) = p(x_i)$$

em que $p(x_i)$ representa o valor esperado de Y dado o valor x_i da variável X_i. A forma específica do modelo de Regressão Logística Simples é:

$$p(x_i) = P(y = 1\ |x_i) = \frac{e^{\beta_0+\beta_1 x_i}}{1 + e^{\beta_0+\beta_1 x_i}}$$

A média condicional de Y dado x_i, quando se usa a distribuição logística é definida por:

$$E(Y_i|X = x_i) = p(x_i)$$

em que $i = 1,2, \dots, n$, ou seja, o valor esperado irá sempre representar a probabilidade de $Y_i = 1$.

Seja a transformação linear $t(x_i) = \beta_0 + \beta_1 x_i$, então:

$$p(x_i) = \frac{e^{t(x_i)}}{1 + e^{t(x_i)}}$$

sendo assim,

$$\frac{p(x_i)}{1 - p(x_i)} = \frac{\frac{e^{t(x_i)}}{1 + e^{t(x_i)}}}{\frac{1}{1 + e^{t(x_i)}}} = e^{t(x_i)}$$

A transformação de $p(x_i)$ que é o ponto importante no estudo de Regressão Logística, aqui, é a transformação logito. Essa transformação é defenida, em termos de $p(x_i)$, como:

$$t(x_i) = log\left(\frac{p(x_i)}{1 - p(x_i)}\right)$$

em que $t(x_i)$ é o logito.

3.1.4 Regressão Logística Simples

A Regressão Logística Simples trata de um modelo no qual a variável resposta assume valores 0 ou 1 e contém apenas uma variável explicativa (x_1). Sabe-se que a observação da variável resposta Y, dado um valor de x será a probabilidade de ocorrência ($p(x_i)$) mais um erro (ε). Com isso, se $Y = 1$ então $1 = p(x) + \varepsilon$ e $\varepsilon = 1 - p(x)$. Já quando $Y = 0$, então $0 = p(x) + \varepsilon$ e assim $\varepsilon = -p(x)$. Como $p(x)$ é sempre um valor positivo, ε assume sempre um valor negativo quando $Y = 0$ e sempre positivo quando $Y = 1$.

Logo, a distribuição condicional da variável resposta segue uma distribuição Bernoulli com probabilidade definida pela média condicional $p(x)$. Conforme a distribuição de Bernoulli, a função de probabilidade de Y é $P(Y = y) = p^y(1-p)^{1-y}$.

Como mencionado anteriormente, utilizando a função de ligação logito, o valor esperado da variável resposta, na regressão logística simples é definido por $p(x_i) = \frac{e^{\beta_0+\beta_1 x_1}}{1+e^{\beta_0+\beta_1 x_1}}$ e desde que as observações sejam independentes, a função de probabilidade é definida por:

$$f(y_i) = p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$$

Quando o vetor da média condicional $\mathrm{E(Y|X)}$ pode assumir qualquer valor quando X varia entre $-\infty$ e ∞, os parâmetros do modelo podem ser estimados utilizando o método dos Mínimos Quadrados (MMQ), pois o objetivo é ajustar um modelo linear. Porém, quando o vetor da média condicional apresentar a forma de uma distribuição acumulada, como no caso da variável dicotômica, a estimação dos parâmetros da função é definida pela máxima verossimilhança (não linear).

Como o objetivo é obter o valor dos parâmetros com o propósito de encontrar os melhores valores para $\hat{\beta}$ utiliza-se, então, o método de máxima verossimilhança, a fim de que os estimadores dos parâmetros maximizem a função que expressa a probabilidade com base nos dados observados.

A função de verossimilhança é definida por:

$$L(y_1, y_2, \dots, y_n, \boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i) = \prod_{i=1}^{n} p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$$

o que representa a expressão:

$$\left[p(x_1)^{y_1}\left(1-p(x_1)\right)^{1-y_1}\right] \cdot \left[p(x_2)^{y_2}\left(1-p(x_2)\right)^{1-y_2}\right] \cdots \left[p(x_n)^{y_n}\left(1-p(x_n)\right)^{1-y_n}\right]$$

Aplicando o logaritmo, tem-se:

$$\ln L(y_1, y_2, \dots, y_n, \boldsymbol{\beta}) = \ln \prod_{i=1}^{n} f(y_i)$$

$$\ln L(y_1, y_2, \dots, y_n, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \ln p(x_i) + (1 - y_i) \ln\left(1 - p(x_i)\right) \right]$$

Para encontrar o valor de $\boldsymbol{\beta}$ que maximiza $\ln L(y_1, y_2, \dots, y_n, \boldsymbol{\beta})$, faz-se a derivada parcial de $\ln L(y, \boldsymbol{\beta})$ com relação a β_0 e em seguida a β_1, igualando as duas derivadas a zero.

$$\ln L(y_1, y_2, \dots, y_n, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \ln\left(\frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}\right) + (1 - y_i) \ln\left(1 - \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}\right) \right]$$

Como $y = \ln u \Longrightarrow y` = \frac{1}{u} u`$.

sendo que $u = \beta_0 + \beta_1 x_1$, então

$$\frac{\partial \, ln \, L(\boldsymbol{y}, \boldsymbol{\beta})}{\partial \beta_0} = \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} \left[y_i \, ln\left(\frac{e^u}{1 + e^u}\right) + (1 - y_i) \, ln\left(1 - \frac{e^u}{1 + e^u}\right) \right]$$

derivando $ln\left(\frac{e^u}{1+e^u}\right)$, tem-se:

$$\left[ln\left(\frac{e^u}{1 + e^u}\right) \right]` = \frac{1}{\frac{e^u}{1 + e^u}} \left(\frac{e^u}{1 + e^u}\right)` = \frac{1}{\frac{e^u}{1 + e^u}} \left(\frac{(e^u)`(1 + e^u) - e^u(1 + e^u)`}{(1 + e^u)^2}\right)$$

$$= \frac{1}{\frac{e^u}{1 + e^u}} \left(\frac{e^u(u)`(1 + e^u) - e^u(u)` e^u}{(1 + e^u)^2}\right)$$

Como $u` = \frac{\partial u}{\partial \beta_0} = 1$,

$$= \frac{1}{\frac{e^u}{1 + e^u}} \left(\frac{e^u(1 + e^u) - e^u e^u}{(1 + e^u)^2}\right)$$

$$= \frac{1}{\frac{e^u}{1 + e^u}} \left(\frac{e^u + e^{u^2} - e^{u^2}}{(1 + e^u)^2}\right)$$

$$= \frac{1 + e^u}{e^u} \cdot \frac{e^u}{(1 + e^u)^2}$$

$$= \frac{1}{1 + e^u}$$

derivando $ln\left(1 - \frac{e^u}{1+e^u}\right)$, tem-se:

$$\left[ln\left(1-\frac{e^u}{1+e^u}\right)\right]` = \frac{1}{1-\frac{e^u}{1+e^u}}\left(1-\frac{e^u}{1+e^u}\right)` = \frac{1}{1-\frac{e^u}{1+e^u}}\left(\frac{-e^u}{(1+e^u)^2}\right)$$

$$= \frac{1}{\frac{1+e^u-e^u}{1+e^u}}\left(\frac{-e^u}{(1+e^u)^2}\right)$$

$$= 1+e^u\left(\frac{-e^u}{(1+e^u)^2}\right)$$

$$= -\frac{e^u}{1+e^u}$$

Retornando à derivada principal, tem-se que:

$$\frac{\partial\, ln\, L(\boldsymbol{y},\boldsymbol{\beta})}{\partial\beta_0} = \sum_{i=1}^{n}\left[\frac{y_i}{1+e^u}+(1-y_i)\left(-\frac{e^u}{1+e^u}\right)\right]$$

$$= \sum_{i=1}^{n}\left[\frac{y_i+(1-y_i)(-e^u)}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}\left[\frac{y_i-e^u+y_ie^u}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}\left[\frac{y_i(1+e^u)-e^u}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}\left[y_i-\frac{e^u}{1+e^u}\right]$$

Sabendo que $\frac{e^u}{1+e^u} = \frac{e^{\beta_0+\beta_1x_1}}{1+e^{\beta_0+\beta_1x_1}}$ e que $\frac{e^{\beta_0+\beta_1x_1}}{1+e^{\beta_0+\beta_1x_1}} = p(x_i)$, então

$$\frac{\partial\, ln\, L(y,\beta)}{\partial\beta_0} = \sum_{i=1}^{n}[y_i-p(x_i)] = 0$$

Sabendo que o estimador de $p(x_i)$ é $\hat{p}(x_i)$:

$$\sum_{i=1}^{n}[y_i-p(x_i)] = 0$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \hat{p}(x_i) = 0$$

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \hat{p}(x_i)$$

ou seja, a soma dos valores observados de y é igual a soma dos valores estimados da probabilidade do evento sucesso.

Derivando, agora, em função de β_1, tem-se:

$$\frac{\partial \ln L(y,\beta)}{\partial \beta_1} = \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} \left[y_i \ln\left(\frac{e^u}{1+e^u}\right) + (1-y_i) \ln\left(1 - \frac{e^u}{1+e^u}\right)\right]$$

sendo a derivada de $\ln\left(\frac{e^u}{1+e^u}\right)$, dado que $\frac{\partial u}{\partial \beta_1} = x_1$

$$\left[ln\left(\frac{e^u}{1+e^u}\right)\right]` = \frac{1}{\frac{e^u}{1+e^u}} \left(\frac{e^u}{1+e^u}\right)` = \frac{1+e^u}{e^u} \left(\frac{(e^u)`(1+e^u) - e^u(1+e^u)`}{(1+e^u)^2}\right)$$

$$= \frac{1+e^u}{e^u} \left(\frac{e^u(x_1)(1+e^u) - e^u(e^u)(x_1)}{(1+e^u)^2}\right)$$

$$= \frac{1+e^u}{e^u} \left(\frac{x_1\left[e^u + e^{u^2} - e^{u^2}\right]}{(1+e^u)^2}\right)$$

$$= \frac{1+e^u}{e^u} \left(\frac{e^u x_1}{(1+e^u)^2}\right) = \frac{x_1}{1+e^u}$$

sendo a derivada de $ln\left(1 - \frac{e^u}{1+e^u}\right)$, dado que $\frac{\partial u}{\partial \beta_1} = x_1$

$$\left[ln\left(1 - \frac{e^u}{1+e^u}\right)\right]` = \frac{1}{1 - \frac{e^u}{1+e^u}} \left(1 - \frac{e^u}{1+e^u}\right)` = \frac{1}{\frac{1+e^u-e^u}{1+e^u}} \left(-\frac{e^u x_1}{(1+e^u)^2}\right) =$$

$$= \frac{1+e^u}{1} \left(-\frac{e^u x_1}{(1+e^u)^2}\right) =$$

$$= -\frac{e^u x_1}{1+e^u}$$

Retornando à derivada principal, tem-se que:

$$\frac{\partial \ln L(\boldsymbol{y},\boldsymbol{\beta})}{\partial \beta_1} = \sum_{i=1}^{n}\left[\frac{y_i x_1}{1+e^u} - (1-y_i)\left(\frac{e^u x_1}{1+e^u}\right)\right]$$

$$= \sum_{i=1}^{n}\left[\frac{y_i x_1 - e^u x_1 + y_i x_1 e^u}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}\left[\frac{y_i x_1 (1+e^u) - e^u x_1}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}\left[y_i x_1 - \frac{e^u x_1}{1+e^u}\right]$$

$$= \sum_{i=1}^{n}[x_1(y_i - p(x_i))]$$

então

$$\frac{\partial \ln L(\boldsymbol{y},\boldsymbol{\beta})}{\partial \beta_0} = \sum_{i=1}^{n}[x_i(y_i - p(x_i))] = 0$$

As equações encontradas, a partir das derivadas, são conhecidas como equações de verossimilhança. Em Regressão Logística essas equações não são lineares em β_0 e β_1, o que exige métodos especiais para solução. Estes métodos são de natureza iterativa e têm sido programados em softwares onde a Regressão Logística está disponível (HOSMER; LEMESHOW, 2000).

3.1.4.1 Teste de Significância dos Coeficientes

Pode-se usar a estatística Deviance para testar hipóteses sobre subconjuntos dos parâmetros do modelo, assim como usa-se as somas de quadrados do erro para testar hipóteses semelhantes no modelo de regressão linear normal. Pode-se escrever o modelo completo em duas partes, como:

$$\ln\left(\frac{p(x_i)}{1-p(x_i)}\right) = \boldsymbol{X\beta} = \boldsymbol{X}_{PP}\boldsymbol{\beta}_{PP} + \boldsymbol{X}_{SP}\boldsymbol{\beta}_{SP}$$

em que o modelo completo tem $k+1$ parâmetros. O vator $\boldsymbol{\beta}_{PP}$ é referente aos parâmetros da primeira parte, ou seja, contém $k+1-r$ dos parâmetros no modelo completo, $\boldsymbol{\beta}_{SP}$ contém os parâmetros da segunda parte, ou seja, r parâmetros e que as colunas da matriz $\boldsymbol{X}_{PP}$ e $\boldsymbol{X}_{SP}$ contém as variáveis associadas a esses parâmetros. Neste caso r é o número de parâmetros que deseja-se testar.

A estatística Deviance do modelo completo é descrita por $D(\boldsymbol{\beta})$ e supondo que queira-se testar

$$\begin{cases} H_0: \boldsymbol{\beta}_{SP} = 0 \\ H_1: \boldsymbol{\beta}_{SP} \neq 0 \end{cases}$$

o modelo reduzido será

$$\ln\left(\frac{p(x_i)}{1-p(x_i)}\right) = \boldsymbol{X}_{PP}\boldsymbol{\beta}_{PP}$$

e a estatística Deviance do modelo reduzido será $D(\boldsymbol{\beta}_{PP})$.

Segundo Montegomery, Peck e Vining (2006), a estatística Deviance para o modelo reduzido será sempre maior que a deviance do modelo completo, porque o modelo reduzido contém menos parâmetros. No entanto, se a deviance do modelo reduzido não for muito maior que a deviance do modelo completo indica que o ajuste do modelo reduzido é quase tão bom quanto o ajuste do modelo completo, por isso é provável que os parâmetros em $\boldsymbol{\beta}_{SP}$ sejam iguais a zero. Porém, se a diferença da deviance é maior, pelo menos um dos parâmetros de $\boldsymbol{\beta}_{SP}$ não é zero e então deve-se rejeitar a hipótese nula. Formalmente a diferença entre deviances é

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = D(\boldsymbol{\beta}_{PP}) - D(\boldsymbol{\beta}) \tag{5}$$

e tem $n-(k+1-r)-(n-(k+1)) = r$ graus de liberdade. Se a hipótese nula é verdadeira e se n é grande, a diferença (1.5) tem uma distribuição qui-quadrado com r graus de liberdade. Portanto, o teste estatístico e o critério de decisão são:

$$\begin{cases} \text{Se } D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) \geq \chi^2_r(\alpha) \text{ rejeitar a hipótese nula.} \\ \text{Se } D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) < \chi^2_r(\alpha) \text{ não rejeitar a hipótese nula.} \end{cases}$$

Assim, a comparação dos valores da variável resposta com os valores preditos obtidos dos modelos com e sem a variável em questão é baseada na função do log da verossimilhança $L(\boldsymbol{\beta})$. Esta comparação é definida por:

$$D = -2ln\frac{Verossimilhança\ do\ modelo\ Atual}{Verossimilhança\ do\ modelo\ Saturado}$$

$$D = -2ln\left[\frac{\prod_{i=1}^{n}\left[p(x_i)^{y_i}(1-p(x_i))^{(1-y_i)}\right]}{\prod_{i=1}^{n}[{y_i}^{y_i}(1-y_i)^{(1-y_i)}]}\right]$$

$$D = -2ln\left[\frac{p(x_1)^{y_1}(1-p(x_1))^{(1-y_1)}\dots p(x_n)^{y_n}(1-p(x_n))^{(1-y_n)}}{{y_1}^{y_1}(1-y_1)^{(1-y_1)}\dots {y_n}^{y_n}(1-y_n)^{(1-y_n)}}\right]$$

$$D = -2\begin{bmatrix} ln\left[p(x_1)^{y_1}\left(1-p(x_1)\right)^{(1-y_1)}\right]\dots ln\left[p(x_n)^{y_n}\left(1-p(x_n)\right)^{(1-y_n)}\right]- \\ ln\left[{y_1}^{y_1}(1-y_1)^{(1-y_1)}\right]\dots ln\left[{y_n}^{y_n}(1-y_n)^{(1-y_n)}\right] \end{bmatrix}$$

$$D = -2\begin{bmatrix} \left[y_1\, ln\, p(x_1) + (1-y_1)\, ln\left(1-p(x_1)\right) + \dots + y_n\, ln\, p(x_n) + (1-y_n)\, ln\left(1-p(x_n)\right)\right]- \\ [y_1 \ln y_1 + (1-y_1)\ln(1-y_1) + \dots +\ y_n \ln y_n + (1-y_n)\ln(1-y_n)] \end{bmatrix}$$

$$D = -2\sum_{i=1}^{n}\left[y_i\, ln\, p(x_i) + (1-y_i)\, ln\left(1-p(x_i)\right) - y_i \ln y_i - (1-y_i)\ln(1-y_i)\right]$$

$$D = -2\sum_{i=1}^{n}\left[y_i[ln\, p(x_i) - \ln y_i] + (1-y_i)\left[ln\left(1-p(x_i)\right) - \ln(1-y_i)\right]\right]$$

$$D = -2\sum_{i=1}^{n}\left[y_i\left[ln\frac{p(x_i)}{y_i}\right] + (1-y_i)\left[ln\frac{\left(1-p(x_i)\right)}{(1-y_i)}\right]\right]$$

Como o estimador de máxima verossimilhança de $p(x_i)$, definido em 3.1.4, é $\hat{p}(x_i)$ então a estatística $\widehat{D}$ (Deviance) é:

$$\widehat{D} = -2\sum_{i=1}^{n}\left[y_i\left[ln\frac{\hat{p}(x_i)}{y_i}\right] + (1-y_i)\left[ln\frac{\left(1-\hat{p}(x_i)\right)}{(1-y_i)}\right]\right]$$

devido à propriedade de invariância das funções dos estimadores de máxima verossimilhança.

Para estimar a significância de uma variável independente, compara-se o valor de D com e sem a variável independente na Equação:

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = D(\boldsymbol{\beta}_{PP}) - D(\boldsymbol{\beta})$$

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = D_{para\ o\ modelo\ sem\ a\ variável} - D_{para\ o\ modelo\ com\ a\ variável}$$

então:

$$D(\boldsymbol{\beta}_{PP}) = -2ln\left[\frac{L_{\boldsymbol{\beta}_{PP}}}{L_S}\right] \text{ e } D(\boldsymbol{\beta}) = -2ln\left[\frac{L_{\boldsymbol{\beta}}}{L_S}\right]$$

em que:

$D(\boldsymbol{\beta}_{PP})$ é a Deviance para o modelo sem a variável, primeira parte do modelo, excluindo as variáveis que deseja-se testar;

$D(\boldsymbol{\beta})$ é a Deviance para o modelo com a variável;

$L_{\boldsymbol{\beta}_{PP}}$ é a função de verossimilhança do modelo sem a variável;

$L_{\boldsymbol{\beta}}$ é a função de verossimilhança do modelo com a variável e

L_S é a função de verossimilhança do modelo saturado.

Assim:

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = D(\boldsymbol{\beta}_{PP}) - D(\boldsymbol{\beta}) = -2ln\left[\frac{L_{\boldsymbol{\beta}_{PP}}}{L_S}\right] + 2ln\left[\frac{L_{\boldsymbol{\beta}}}{L_S}\right]$$

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = -2\left[ln\, L_{\boldsymbol{\beta}_{PP}} - ln\, L_S - ln\, L_{\boldsymbol{\beta}} + ln\, L_S\right]$$

$$D(\beta_2|\beta_1) = -2ln\left[\frac{L_{\boldsymbol{\beta}_{PP}}}{L_{\boldsymbol{\beta}}}\right]$$

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = -2ln\left[\frac{verossimilhança\ sem\ a\ variável}{verossimilhança\ com\ a\ variável}\right]$$

Segundo Hosmer e Lemeshow (2000), para o caso de uma única variável independente, quando ela não está no modelo, o Estimador de Máxima Verossimilhança de β_0 é $\ln\left(\frac{n_1}{n_0}\right)$ em que n_1 é o número de indivíduos com a característica de interesse e n_0 é o número de observações que não tem a característica de interesse.

Dado que $\frac{\partial \ln L(\boldsymbol{y},\boldsymbol{\beta})}{\partial \beta_0} = \sum_{i=1}^{n}[y_i - p(x_i)]$ e $p(x_i) = \frac{e^{\beta_0+\beta_1 x_1}}{1+e^{\beta_0+\beta_1 x_1}}$. Como a variável independente não está no modelo reduzido, tem-se que $\frac{\partial \ln L(\boldsymbol{y},\boldsymbol{\beta})}{\partial \beta_0} = \sum_{i=1}^{n}\left[y_i - \frac{e^{\beta_0}}{1+e^{\beta_0}}\right]$. Igualando essa expressão a zero:

$$\sum_{i=1}^{n}\left[y_i - \frac{e^{\beta_0}}{1+e^{\beta_0}}\right] = 0$$

$$\sum_{i=1}^{n}\left[y_i + y_i e^{\beta_0} - e^{\beta_0}\right] = 0$$

$$\sum_{i=1}^{n}\left[y_i - e^{\beta_0}(1-y_i)\right] = 0$$

$$\sum_{i=1}^{n} y_i - e^{\beta_0}\sum_{i=1}^{n}(1-y_i) = 0$$

$$e^{\beta_0} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n}(1-y_i)}$$

Como $\sum_{i=1}^{n} y_i = n_1$ (número de casos de sucesso) e $\sum_{i=1}^{n}(1-y_i) = n_0$ (número de casos em que $Y = 0$), aplicando ln em ambos os lados têm-se:

$$ln\, e^{\beta_0} = ln\left(\frac{n_1}{n_0}\right) \text{ então } \hat{\beta}_0 = ln\left(\frac{n_1}{n_0}\right)$$

Sendo assim, para o caso de Regressão Logística Simples:

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = -2ln\left[\frac{verossimilhança\ sem\ a\ variável}{verossimilhança\ com\ a\ variável}\right]$$

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = -2ln\left[\frac{\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}}{\prod_{i=1}^{n}[p(x_i)^{y_i}(1-p(x_i))^{(1-y_i)}]}\right]$$

Como visto anteriormente, o ln do denominador é definido por:

$$ln\left(\prod_{i=1}^{n}\left[p(x_i)^{y_i}\left(1-p(x_i)\right)^{(1-y_i)}\right]\right) = \sum_{i=1}^{n}\left[y_i\, ln\, p(x_i) + (1-y_i)\, ln\left(1-p(x_i)\right)\right]$$

já o $ln\left(\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}\right)$ é definido por:

$$ln\left(\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n_0}{n}\right)^{n_0}\right) = n_1 ln\frac{n_1}{n} + n_0 ln\frac{n_0}{n}$$

$$= n_1 \ln n_1 - n_1 \ln n + n_0 \ln n_0 - n_0 \ln n$$

$$= n_1 \ln n_1 + n_0 \ln n_0 - [n_1 \ln n + n_0 \ln n]$$

$$= n_1 \ln n_1 + n_0 \ln n_0 - [(n_1 + n_0) \ln n]$$

$$= n_1 \ln n_1 + n_0 \ln n_0 - n \ln n$$

Então:

$$D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) = -2\sum_{i=1}^{n}\big[y_i\, ln\, p(x_i) + (1-y_i)\, ln\big(1-p(x_i)\big) - (n_1 \ln n_1 + n_0 \ln n_0 - n \ln n)\big]$$

sob a hipótese nula que β_1 é igual a zero, a estatística $D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP})$ tem distruição qui-quadrado com 1 grau de liberdade, com a suposição do tamanho n ser suficientemente grande. Rejeita-se H_0 se $D(\boldsymbol{\beta}_{SP}|\boldsymbol{\beta}_{PP}) \geq \chi_r^2(\alpha)$ (HOSMER; LEMESHOW, 2000).

Segundo Hosmer e Lemeshow (2000), antes de concluir que um ou todos os coeficientes são não nulos, tem-se que observar a estatística do teste de Wald. O teste de Wald pode ser obtido comparando a estimativa de máxima verossimilhança de determinado coeficiente, $\hat{\beta}_j$, com a estimativa do seu erro padrão. Assim as hipóteses são as seguintes:

$$\begin{cases} H_0: \beta_j = \beta_j^* \\ H_1: \beta_j \neq \beta_j^* \quad (j = 1, \dots, k) \end{cases}$$

e a estatística teste definida pela seguinte expressão:

$$w_j = \frac{\hat{\beta}_j - \beta_j^*}{\sqrt{var(\hat{\beta}_j)}}$$

em que $\sqrt{var(\hat{\beta}_j)}$ é o desvio padrão estimado do estimador do parâmetro β_j e β_j^* é o valor que se deseja testar. A estatística w_j apresenta uma distribuição qui-quadrado com número de graus de liberdade igual ao número de restrições. Os valores críticos, α_j, para as estimativas dos parâmetros são os níveis para os quais se o valor do teste de Wald calculado para um determinado β_j for maior que o α_j, se rejeita a hipótese nula para um dado nível de significância. No caso do teste dos coeficientes nulos, $\beta_j^* = 0$, e então

$$W = \frac{\hat{\beta}_1}{\widehat{SE}(\hat{\beta}_1)} \sim N(0,1)$$

em que $\widehat{SE}(\hat{\beta}_1)$ é o desvio padrão estimado do estimador do parâmetro β_j.

Observando que $\hat{\beta}_1$ e $\widehat{SE}(\hat{\beta}_1)$ são estimadores de máxima verossimilhança de β_1 e $SE(\beta_1)$ respectivamente, rejeita-se a hipótese nula de $\beta_1 = 0$ se $|W| > z_{\alpha/2}$. Conforme Hosmer e Lemeshow (2000), os Intervalos de Confiança são os seguintes:

Coeficiente de Inclinação:

$$\hat{\beta}_1 \pm z_{1-\alpha/2}\widehat{SE}(\hat{\beta}_1)$$

Intercepto:

$$\hat{\beta}_0 \pm z_{1-\alpha/2}\widehat{SE}(\hat{\beta}_0)$$

Logito:

$$\hat{t}(x) \pm z_{1-\alpha/2}\sqrt{\hat{V}ar(\hat{t}(x))} = \hat{\beta}_0 + \hat{\beta}_1 x \pm z_{1-\alpha/2}\left(\hat{V}ar(\hat{\beta}_0) + x^2\hat{V}ar(\hat{\beta}_1) + 2xCov(\hat{\beta}_0, \hat{\beta}_1)\right)$$

3.1.5 Regressão Logística Múltipla

A Regressão Logística Múltipla, assim como a Regressão Logística Simples, contém a variável resposta como uma variável dicotômica, porém possui mais de uma variável independente (X_i). Sabendo que a probabilidade condicional da variável resposta, considerando k variáveis independentes ($\boldsymbol{x}' = (x_1, x_2, \ldots, x_k)$) é definida por:

$$P(Y = 1|x) = p(x)$$

Neste caso, como trata-se de k variáveis independentes, o logito da Regressão Linear Múltipla é definido por:

$$t(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

Com isso, o Modelo de Regressão Linear Múltipla será:

$$p(x) = \frac{e^{t(x)}}{1 + e^{t(x)}} = \frac{e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k}}{1 + e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k}}$$

escrevendo o modelo linearizado tem-se:

$$\ln\left(\frac{p(x_i)}{1-p(x_i)}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon$$

Mesmo linearizado, este modelo apresenta erros heterocedásticos (com variância não constante) o que torna não aconselhável a utilização do método de mínimos quadrados para a estimação dos parâmetros do modelo.

Sendo a função de máxima verossimilhança $L(y_1, \dots, y_n, \boldsymbol{\beta})$:

$$L(y_1, \dots, y_n, \boldsymbol{\beta}) = \prod_{i=1}^{n} f(y_i)$$

em que $f(y_i) = p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$ é a função de probabilidade de y_i e n o número de observações.

A maximização desta função é um problema equivalente a maximização do seu logaritmo, já que a função logaritmo é uma função monótona crescente. Para facilitar a obtenção do maximizante, tem-se o logaritmo da função de verossimilhança ou função log-verossimilhança, como descrito anteriormente:

$$L(y_1, \dots, y_n, \boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \ln\left(\frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}\right) + \sum_{i=1}^{n} (1-y_i) \ln\left(1-\frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_k x_k}}\right)$$

O estimador de máxima verossimilhança dos $k+1$ componentes de $\boldsymbol{\beta}$ correspondem, por definição, aos valores desses parâmetros que maximizam $L(y_1, \dots, y_n, \boldsymbol{\beta})$. Para obter este máximo, torna-se necessário calcular a primeira e a segunda derivada de $L(y_1, \dots, y_n, \boldsymbol{\beta})$, designadas por Gradiente e matriz Hessiana G.

$$G_{i,j} = \frac{\partial^2 L(y_1, \dots, y_n, \boldsymbol{\beta})}{\partial \beta_i \partial \beta_j}, \qquad i,j = 0,1, \dots, k$$

Não é possível encontrar diretamente uma solução para este problema que assegure a condição necessária para o máximo de $L(y_1, \dots, y_n, \boldsymbol{\beta})$. Assim, este problema de maximização é resolvido por meio de um algoritmo de otimização. Um dos algoritmos de otimização mais utilizados é o de Newton-Raphson. Amemiya (1985) demonstra que o log da função de verossimilhança é globalmente côncavo,

assim o algoritmo de Newton-Raphson converge para um único máximo (os estimadores de máxima verossimilhança) independentemente dos valores de inicialização adotados.

Se os elementos da matriz Hessiana são avaliados como os estimadores de máxima verossimilhança $\beta = \hat{\beta}$, para estimar os valores das variâncias e covariâncias dos coeficientes basta inverter a matriz Hessiana (MONTEGOMERY; PECK; VINING, 2006).

$$Var(\widehat{\boldsymbol{\beta}}) = -G(\widehat{\boldsymbol{\beta}})^{-1} = (\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X})^{-1} \quad (6)$$

Segundo Hosmer e Lemeshow (2000) e Montegomery, Peck e Vining (2006) o ajuste do modelo estimado é $Var(\widehat{\boldsymbol{\beta}}) = (\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X})^{-1}$, em que $\boldsymbol{X}$ é uma matriz $n \times k + 1$ contendo os dados de cada observação e $\boldsymbol{V}$ é uma matriz diagonal $n \times n$ cujos elementos da diagonal principal são $\hat{p}(x_i)(1 - \hat{p}(x_i))$. Assim as matrizes $\boldsymbol{X}$ e $\boldsymbol{V}$ são:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1k} \\ 1 & x_{21} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{bmatrix}$$

e

$$\boldsymbol{V} = \begin{bmatrix} \hat{p}(x_1)(1 - \hat{p}(x_1)) & 0 & \cdots & 0 \\ 0 & \hat{p}(x_2)(1 - \hat{p}(x_2)) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{p}(x_n)(1 - \hat{p}(x_n)) \end{bmatrix}$$

Lembrando que o *j*-ésimo elemento da diagonal da matriz $V(\widehat{\boldsymbol{\beta}})$ é a variância estimada $\hat{\beta}_j$, podendo ser denotada por $Var(\hat{\beta}_j)$, e os elementos fora da diagonal principal são covariâncias de $\hat{\beta}_j$ e $\hat{\beta}_i$, denotadas de $Cov(\hat{\beta}_i, \hat{\beta}_j)$, o estimador do erro padrão é definido por:

$$\hat{d}_v(\hat{\beta}_j) = [\hat{V}ar(\hat{\beta}_j)]^{1/2} \quad j = 0,1,2,\ldots,k$$

em que $\hat{d}_v$ representa a estimaiva do erro padrão.

3.1.5.1 Teste de significância dos parâmetros do modelo

Assim como na Regressão Linear, a primeira etapa é verificar a significância dos parâmetros associados às variáveis no modelo. O teste baseado na estatística G é o mesmo para o caso univariado, mas agora, substitui-se os valores ajustados pelo vetor $\boldsymbol{\beta}^{\prime}$ que contém $k+1$ parâmetros e testa-se as seguintes hipóteses:

$$\begin{cases} H_0: \beta_0 = \beta_1 = \beta_2 = \cdots = 0 \\ H_1: pelo\ menos\ um\ parâmetro\ difere\ de\ 0\ (\beta_j \neq 0). \end{cases}$$

Como mencionado, a estatística G nq eq. (5) tem distribuição χ^2_k e rejeita-se se H_0 se $G > \chi^2_k(\alpha)$. Ao rejeitar H_0 conclui-se que pelo menos um coeficiente ou talvez todos os k coeficientes são diferentes de zero.

Após concluir que pelo menos um parâmetro é diferente de zero, realiza-se o teste univariado de Wald, em que as hipóteses são:

$$\begin{cases} H_0: \hat{\beta}_j = 0 \\ H_1: \hat{\beta}_j \neq 0 \end{cases}$$

ou seja, testa-se a significância da variável X_j. Para isso calcula-se a estatística já descrita anteriormente:

$$W_j = \frac{\hat{\beta}_j - 0}{\sqrt{\hat{V}ar(\hat{\beta}_j)}}, \qquad j = 1, \ldots, k+1$$

Ao calcular W_j rejeita-se H_0 se W_j for menor que o valor do percentil da distribuição da estatística teste W_j, ou seja, X_j não é significativa para o modelo, ou, conclui-se por meio do valor-p que se for maior que um α pré definido, o parâmetro é significativo para o modelo (não se rejeita H_0).

Após retirar-se r variáveis não significativas, realiza-se novamente o teste G, em que, agora compara-se o valor de G com k variáveis iniciais menos o valor G sem as r variáveis retiradas. Caso esta diferença seja menor que a estatística $\chi^2_k(\alpha)$, as r variáveis que foram retiradas não entram no modelo.

3.1.5.2 Estimação do Intervalo de Confiança dos Parâmetros

Os métodos usados na estimação do intervalo de confiança do modelo de Regressão Logística Múltipla são os mesmos da Regressão Linear Simples. Então os intervalos de confiança são definidos a seguir.

Coeficientes:

$$\hat{\beta}_j \pm z_{\alpha/2}\sqrt{\hat{V}ar(\hat{\beta}_j)}$$

Para obter o intervalo de confiança da transformação logito estimada $\left(\hat{t}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_k x_k = \boldsymbol{X}'\widehat{\boldsymbol{\beta}}\right)$é necessário saber a soma das variâncias para cada variável. Como $Var\left(\widehat{\boldsymbol{\beta}}\right) = \left(\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X}\right)^{-1}$ (ver eq. (6)), então:

$$\hat{V}ar[\hat{t}(x)] = \hat{V}ar\left[\boldsymbol{X}'\widehat{\boldsymbol{\beta}}\right] = \boldsymbol{X}'\hat{V}ar\left[\widehat{\boldsymbol{\beta}}\right]\boldsymbol{X} = \boldsymbol{X}'\left(\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X}\right)^{-1}\boldsymbol{X}$$

Logo, o intervalo de confiança é definido por:

$$\hat{t}(x_i) \pm z_{\alpha/2}\sqrt{\boldsymbol{X}'(\boldsymbol{X}'\boldsymbol{V}\boldsymbol{X})^{-1}\boldsymbol{X}}$$

3.1.5.3 Razão de Chance

Uma análise para exploração dos dados diz respeito ao cálculo dos odds e dos odds-ratio (razões de chance). O odds pode ser interpretado como a comparação de dois números: o primeiro traduz a probabilidade de ocorrência de um evento e o segundo, a probabilidade do mesmo evento não ocorrer, ou seja:

$$odds = \frac{p(evento)}{1 - p(evento)} = \frac{probabilidade\ do\ evento\ ocorrer}{probabilidade\ do\ evento\ não\ ocorrer}$$

Já o odds-ratio é a razão entre os odds, ou seja

$$odds - ratio = \frac{odds(Y = 1\,|X = 1)}{odds(Y = 1\,|X = 0)}$$

Sendo assim, a razão de chance é uma medida de associação que indica o quanto mais ou menos provável é a probabilidade de obter uma resposta positiva, consoante ao valor da variável independente. Por exemplo, para variáveis explicativas dicotômicas, considerar-se que Y indica se o indivíduo está em situação regular ou devedora, e X (variável indenpendente) seja a presença ou ausência de

um determinado fator de risco (medida criada a partir da característica do indivíduo), então a razão de chance indica o quanto mais provável é a ocorrência do evento, neste caso, de o indivíduo estar em situação devedora, consoante ao fator de risco estar ou não presente.

Uma razão de chance igual a 1 indica ausência de relação associativa entre a variável explicativa e a variável dependente. Uma razão de chance menor que 1 indica que a variável explicativa está associada negativamente à variável resposta, ou seja, quanto menor a razão de chance, maior a probabilidade de o cliente apresentar menores riscos de incumprimento, indicando que o fator de risco apresenta algum poder para discriminar quem são os bons pagadores. Já uma razão de chance maior que 1 significa que quanto maior é a razão de chance, maior é a probabilidade de o cliente apresentar maiores riscos de incumprimento, evidenciando que o fator de risco considerado apresenta poder para discriminar maus pagadores.

Na maioria dos modelos, os coeficientes estimados das variáveis independentes representam uma inclinação ou taxa de alteração de uma função da variável dependente por acréscimo de uma unidade na variável independente.

No modelo de Regressão Logística $\beta_1 = t(x+1) - t(x)$ é o coeficiente de inclinação que representa a variação na transformação logito para o acréscimo de uma unidade na variável independente X. Toda a interpretação depende da natureza da variável independente. No exemplo citado anteriormente, existe a situação da interpretação dos coeficientes de Regressão Logística quando a variável independente é dicotômica. Segundo Hosmer e Lemeshow (2000) esta situação pode ser apresentada como:

	$X = 1$	$X = 0$
$Y = 1$	$p(1) = \frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}}$	$p(0) = \frac{e^{\beta_0}}{1+e^{\beta_0}}$
$Y = 0$	$1 - p(1) = \frac{1}{1+e^{\beta_0+\beta_1}}$	$1 - p(0) = \frac{1}{1+e^{\beta_0}}$
Total	1	1

O odds para o evento $X = 1$ é definido como $\frac{p(1)}{1-p(1)}$ e a odds para o evento quando $X = 0$ é $\frac{p(0)}{1-p(0)}$. Sendo assim, o odds-ratio (razão de chance) é definido como o odds de $X = 1$ pelo odds de $X = 0$, como:

$$\psi = \frac{\frac{p(1)}{1-p(1)}}{\frac{p(0)}{1-p(0)}} = \frac{\frac{\frac{e^{\beta_0+\beta_1}}{1+e^{\beta_0+\beta_1}}}{\frac{1}{1+e^{\beta_0+\beta_1}}}}{\frac{\frac{e^{\beta_0}}{1+e^{\beta_0}}}{\frac{1}{1+e^{\beta_0}}}} = \frac{e^{\beta_0+\beta_1}}{e^{\beta_0}} = \frac{e^{\beta_0}e^{\beta_1}}{e^{\beta_0}} = e^{\beta_1}$$

O log de odds-ratio conhecido como logito é:

$$t(1) = \ln\left(\frac{p(1)}{1-p(1)}\right) \qquad \text{e} \qquad t(0) = \ln\left(\frac{p(0)}{1-p(0)}\right)$$

então o ln da razão de chances é:

$$\ln\psi = \ln\left[\frac{\frac{p(1)}{1-p(1)}}{\frac{p(0)}{1-p(0)}}\right] = \ln\left[e^{\beta_1}\right] = \beta_1$$

Considerando o exemplo mencionado anteriormente, se $\psi = 0{,}5$ a ocorrência de ser um mau pagador é a metade entre aqueles que não tem o fator de risco do que entre os indivíduos que tem fator de risco. Se $\psi = 9{,}5$, então a chance de um indivíduo ser mau pagador é 9 vezes maior em indivíduos com o fator de risco do que um indivíduo sem o fator de risco.

O estimador de ψ tende a ter distribuição assimétrica. A assimetria amostral de $\psi = 0{,}5$ é devido ao fato que ela varia entre 0 e ∞, com valor 0 ocorrendo quando $\psi = 1$. Para tamanhos amostrais grandes, a distribuição de ψ será normal e portanto simétrica. Assim o intervalo de confiança de $100(1-\alpha)\%$ será:

$$exp\left[\hat{\beta}_j \pm z_{\alpha/2}\sqrt{\hat{V}ar\left(\hat{\beta}_j\right)}\right] \tag{7}$$

Quando tratar-se de uma variável independente com mais de duas categorias pode-se usar um conjunto de variáveis dicotômicas para representá-las.

Fixa-se um grupo como referência com o qual os outros grupos serão comparados. O método para especificação das variáveis dicotômicas envolve fazer todas elas iguais a zero para o grupo de referência e fixar uma única variável de planejamento igual a 1 para cada um dos outros grupos. Sendo assim se a variável independente contiver k categorias, serão criadas $k-1$ variáveis dicotômicas para explicá-las:

Categorias de Y	D_1	D_2
A	0	0
B	1	0
C	0	1

O intervalo de confiança para a razão de chance será exatamente o mesmo que apresentado na eq. (7). Segundo Hosmer e Lemeshow (2000) esse método de codificação de variáveis de planejamento é o mais utilizado na literatura e conhecido como codificação de célula referente, pois o interesse é estimar o risco de um grupo “com a ocorrência” em relação ao outro grupo “sem a ocorrência”.

Tratando-se de uma variável independente contínua o log das chances para uma variação de c unidades em X fornece a diferença logito $t(x+c)-t(x)=c\beta_1$, e a razão de chances será:

$$\psi(c)=\psi(x+cx)=e^{c\beta_1}$$

O intervalo de confiança para a razão de chance $\psi(c)$ (HOSMER; LEMESHOW, 2000) é definida por:

$$exp\left[c\hat{\beta}_j \pm z_{\alpha/2}\sqrt{\hat{V}ar(\hat{\beta}_j)}\right]$$

A interpretação do coeficiente estimado para uma variável contínua é similar ao de uma variável nominal. A principal diferença é que é necessário definir que quantidade c seria uma mudança significativa nas variáveis contínuas.

3.1.5.4 Seleção de variáveis

Quando se selecionam dados no âmbito de um problema de classificação, a tendência é acrescentar o maior número de variáveis possíveis, de forma a melhor caracterizar o problema. Acontece, normalmente, que muitas das variáveis não estão associadas a variável resposta (*target*), havendo nestes casos, dois tipos de variáveis: as variáveis completamente irrelevantes, ou seja, que em nada distiguem a variável resposta; e as variáveis redundantes, ou seja, que em nada acrescentam a discriminação da variável resposta dado que alguma outra variável já acrescentou a mesma informação. Por esta razão, é comum em estudos deste gênero, considerarem-se diversas abordagens de forma a encontrar as relações tidas entre as variáveis independentes e a variável resposta.

O propósito da seleção de variáveis consiste em, a partir de um conjunto inicial de F variáveis, selecionar um subconjunto H, tal que H<F, tendo sido H apurado segundo um determinado critério que permita identificar as variáveis relevantes para o problema em análise. A eliminação de variáveis inúteis permite reduzir a dimensão dos dados e a sua complexidade e portanto, reduzir o tempo de processamento dos métodos. Além disso, segundo Hosmer e Lemeshow (2000), a seleção de variáveis é um passo muito importante, pois tendencialmente, com um menor número de variáveis o modelo será mais robusto.

Para alcançar o objetivo na seleção de variáveis é necessário: (1) um plano de seleção de variáveis, (2) um método para a validação do modelo em termos das variáveis individuais e também do ponto de vista do ajuste com todas no modelo (HOSMER; LEMESHOW, 2000).

Na obtenção de um modelo estatístico procura-se o mais parcimonioso, mas que explique bem os dados. A vantagem em minimizar o número de variáveis é que o modelo resultante provavelmente é mais estável numericamente e é mais fácil de ser generalizado, pois quanto mais variáveis o modelo tiver, maiores serão os erros padrão estimados e o modelo fica cada vez mais dependente dos dados observados.

Conforme Hosmer e Lemeshow (2000), as etapas para a seleção de variáveis são as seguintes:

1) O processo de seleção começa com uma análise exploratória univariada cuidadosa para cada variável. Deve-se tomar cuidado com a variável independente, pois dependendo de seu tipo podem ocorrer tabelas de contigência com caselas zero, que produzirá uma estimativa pontual univariada para uma das razões de chances iguais a zero ou infinito.

2) Depois é feita a seleção para uma análise multivariada. A variável cujo teste univariado tiver valor-p < 0,25 é candidata a entrar no modelo multivariado juntamente com outras variáveis consideradas importantes pelo especialista responsável pela análise.

O valor de nível de significância $\alpha = 0{,}25$ é usado como critério para seleção de variáveis, pois o uso do valor tradicional ($\alpha = 0{,}05$) frequentemente falha na identificação de variáveis conhecidas como importantes.

3) Nesta etapa, a importância de cada variável incluída no modelo deve ser verificada. Por isso, deve-se calcular a estatística de Wald e uma comparação de cada coeficiente estimado com o coeficiente do modelo univariado contendo apenas aquela variável. As variáveis que não contribuírem para o modelo baseado neste critérios devem ser eliminadas e um novo modelo deve ser ajustado. O novo modelo é comparado com o modelo anterior (sempre com mais variáveis) por meio do teste da razão de verossimilhança. Os coeficientes estimados para as variáveis restantes devem ser comparados com aqueles do modelo completo. É necessário, verificar as variáveis cujos coeficientes têm mudanças marcantes em magnitude. Este processo de eliminação, reajustamento e verificação é feito até que todas as variáveis importantes estejam incluídas no modelo e aquelas excluídas não tenham importância estatística.

4) Após a obtenção do modelo com todas as variáveis essenciais, é interessante considerar os termos de interação entre as variáveis. Primeiro, incluí-se no modelo principal cada interação e compara-se o modelo de interação com o modelo principal. Selecionam-se as interações significativas e ajusta-se um novo modelo. O novo modelo é comparado com o modelo principal. Se não existir efeito de interação o processo está completo, mas, se existir o efeito de interação, o processo continuará até que se determine o modelo completo com as interações.

Outra maneira para selecionar variáveis é o método *Stepwise*. Neste tipo de seleção, as variáveis são selecionadas tanto por inclusão como por exclusão no modelo em um uso sequencial baseado exclusivamente em critério estatístico. Existem duas outras versões do procedimento de seleção:

a) Seleção *forward* com teste para eliminação *backward*;

b) Eliminação *backward* seguido de um teste de seleção *forward*. A seleção *stepwise* é útil porque ela constrói modelos em forma sequencial e permite o exame de um conjunto de modelos que podem não ter sido examinados.

A seleção *stepwise* é um algoritmo estatístico que verifica a importância das variáveis e também em incluí-las ou excluí-las com base numa regra de decisão fixada. A importância de uma variável é definida em termos de uma medida da significância estatística do coeficiente da variável (HOSMER; LEMESHOW, 2000).

Segundo Hosmer e Lemeshow (2000), na Regressão Linear *Stepwise*, o teste F é usado desde que os erros sejam assumidos com distribuição Normal. Na Regressão Logística *Stepwise*, os erros são assumidos a partir da distribuição Binomial e a significância é avaliada pelo teste razão de verossimilhança qui-quadrado.

Assim, em cada passo do procedimento, a variável mais importante, em termos estatísticos, será a variável que produz a maior mudança no log de verossimilhança relativo a um modelo não contendo a variável (modelo com maior estatística da razão de verossimilhança G) (HOSMER; LEMESHOW, 2000).

Depois que o modelo de Regressão Logístico é ajustado, podem ocorrer alguns problemas numéricos:

a) Frequência de zeros em uma tabela de contigência: Uma prática comum para evitar uma estimativa do ponto indefinido é adicionar 1,5 para cada célula. Este valor adicionado permite a mudança da análise de uma tabela de contingência simples, mas raramente é satisfatório para um conjunto de dados mais complexo (HOSMER; LEMESHOW, 2000).

A presença de uma célula de contagem zero deve ser detectada na análise univariada dos dados, pois esta célula causará problemas de estágio de modelagem de análise. Para contornar este problema, pode-se juntar as categorias

da variável em uma forma significativa para eliminá-la, ou se a variável é no mínimo de escala ordinal, tratá-la como se ela fosse contínua.

b) Covariáveis discriminam perfeitamente: É quando um conjunto de covariáveis separa completamente os grupos respostas. Se uma covariável é conhecida, o valor da variável resposta com certeza é conhecido.

c) Colinearidades: Como no caso da Regressão Linear, o ajuste do modelo via Regressão Logística é também sensível para colinearidades entre as variáveis independentes no modelo.

Hosmer e Lemeshow (2000) destacam que os problemas numéricos de uma célula de contagem zero, separação completa e colinearidade, são sempre manifestados por erros padrão estimados extraordinariamente grandes e algumas vezes, por coeficientes estimados grandes.

3.1.5.5 Medidas de qualidade do ajuste

Após a estimação do modelo, o mais adequado é avaliar a qualidade do ajuste do mesmo. Com isso, o interesse é testar as hipóteses:

$$\begin{cases} H_0: o\ modelo \text{ é } adequado \\ H_1: o\ modelo \text{ não é } adequado \end{cases}$$

Para verificar a qualidade do ajuste, é necessário verificar se o valor estimado pelo modelo proposto é igual aos valores reais. O esperado é que as distâncias entre $\boldsymbol{y}$ (vetor da variável resposta) e $\hat{\boldsymbol{y}}$ (vetor dos valores ajustados) sejam pequenas.

Existem algumas estatísticas testes capazes de testar essas hipóteses, são elas:

3.1.5.6 Estatísticas Pearson Qui-Quadrado e Deviance

Na Regressão Logística, segundo Hosmer e Lemeshow (2000), existem muitas formas de medir a diferença entre o valor esperado e o valor ajustado. Uma forma é ajustar a j-ésima covariável padrão como $\hat{y}_j$.

$$\hat{y}_j = m_j \hat{p}_j = m_j \frac{e^{\hat{t}(x_j)}}{1 + e^{\hat{t}(x_j)}}$$

em que:

m_j é o número de observações que tiveram os mesmos valores, para $j = 1, \dots, J$ sendo que J é o número de observações x distintas;

$\hat{p}_j$ é a probabilidade condicional da variável resposta, denotada aqui como $\hat{p}(x_i)$;

$\hat{t}(x_j)$ é a transformação logito estimada.

Em outras palavras, é como se fossem criados grupos, onde as observações são as mesmas (ver exemplo no apêndice 1). Sabendo-se como calcular $\hat{y}$, a qualidade do ajuste pode ser avaliada com a estatística qui-quadrado de Pearson, que compara as probabilidades observadas e esperadas de sucesso e fracasso em cada grupo de observações. O número esperado de sucesso é $m_j \hat{p}_j$ e o número esperado de fracassos é $m_j(1 - \hat{p}_j)$. A estatística de Pearson é (MONTEGOMERY; PECK; VINING, 2006):

$$\chi^2 = \sum_{j=1}^{J} \left\{ \frac{(y_j - m_j \hat{p}_j)^2}{m_j \hat{p}_j} + \frac{\left[(m_j - y_j) - m_j(1 - \hat{p}_j) \right]^2}{m_j(1 - \hat{p}_j)} \right\} =$$

$$= \sum_{j=1}^{J} \frac{(y_j - m_j \hat{p}_j)^2}{m_j \hat{p}_j (1 - \hat{p}_j)}$$

A estatística χ^2 pode ser comparada a uma distribuição qui-quadrado com $n - (k + 1)$ graus de liberdade. Pequenos valores para a estatística (ou um valor-p grande) implica que o modelo proporciona um ajuste satisfatório aos dados.

A qualidade do ajuste também pode ser avaliada utilizando o resíduo de Deviance. A estatística de Deviance como o dobro da diferença do log da verossimilhança entre o modelo saturado e o modelo completo (que é o modelo atual), que foi ajustado para os dados com probabilidade de sucesso estimado $\hat{p}_j = \frac{e^{\hat{\boldsymbol{\beta}}' \boldsymbol{x}}}{1 + e^{\hat{\boldsymbol{\beta}}' \boldsymbol{x}}}$. A Deviance é definida como:

$$D = 2\ln\frac{L(modelo\ saturado)}{L(modelo\ completo)} = 2\sum_{j=1}^{J}\left[y_j \ln\left(\frac{y_j}{\hat{y}_j}\right) + +\left(m_j - y_j\right)\ln\left(\frac{m_j - y_j}{m_j(1-\hat{p}_j)}\right)\right]$$

Note que, no cálculo da Deviance, $y_j \ln\left(\frac{y_j}{\hat{y}_j}\right) = 0$ se $y_j = 0$ e se $y_j = m_j$ tem-se $\left(m_j - y_j\right)\ln\left(\frac{m_j - y_j}{m_j(1-\hat{p}_j)}\right) = 0$. Quando o modelo de regressão logística é ajustado adequadamente e o tamanho da amostra é grande, a Deviance segue uma distribuição qui-quadrado com $n-(k+1)$ graus de liberdade em que $(k+1)$ é o número de parâmetros no modelo. Pequenos valores de Deviance (ou valor-p grande) implica que o modelo proporciona um bom ajuste aos dados, enquanto grandes valores da Deviance indicam que o modelo atual não é adequado (MONTEGOMERY; PECK; VINING, 2006).

3.1.5.7 Teste de Hosmer-Lemeshow para adequação do modelo

Hosmer e Lemeshow propuseram um teste para verificar a adequabilidade do modelo quando não há réplica nas variáveis regressoras. Neste procedimento as observações são classificadas em g grupos com base nas probabilidades estimadas de sucesso e geralmente, cerca de 10 grupo são usados (quando $g = 10$, os grupos são chamados de decis de risco) e o número de sucessos observados O_j e fracassos $N_j - O_j$ são comparados com a frequencia esperada em cada grupo, $N_j\bar{p}_j$ e $N_j(1-\bar{p}_j)$, em que N_j é o número de observações em cada grupo e a probabilidade média de sucesso estimada em cada um dos j-ésimo grupo é definida por $\bar{p}_j = \sum_{i\in grupoj}\frac{\hat{p}_i}{N_j}$.

$$HL = \sum_{j=1}^{n}\frac{(O_j - N_j\bar{p}_j)^2}{N_j\bar{p}_j(1-\bar{p}_j)}$$

Se o modelo de regressão logística está correto, a estatística de Hosmer Lemeshow (2000) segue uma distribuição qui-quadrada com $g-2$ graus de liberdade quando a amostra é grande. Grandes valores de estatística HL implicam que o modelo não tem um adequado ajuste aos dados (MONTEGOMERY; PECK; VINING, 2006).

3.1.5.8 Matriz de confusão

A matriz de confusão resume os resultados do modelo. Esta tabela, também conhecida como tabela de classificação, é o resultado da classificação cruzada da variável resposta Y com os valores dicotômicos cujos valores são derivados da probabilidade estimada pelo modelo (HOSMER; LEMESHOW, 2000).

Com o modelo ajustado atribui-se um valor estimado de $\hat{y}$ (ou 0, ou 1) a partir da probabilidade estimada pelo modelo ($\hat{p}(x_i)$) para cada indivíduo. Assim o i-ésimo indivíduo será classificado como 1 se $\hat{p}(x_i) \leq c$ (em que c é um ponto de corte previamente definido, conhecido como *cutoff*) e 0 caso contrário. Um valor, segundo Hosmer e Lemeshow (2000), comum para c é $0{,}5$. Para um determinado *cutoff* é possível determinar a matriz de confusão, como apresentada a seguir:

		Valores Previstos		ERROS	
		1	0	1	0
Valores observados	1	d Verdadeiro Positivo (VP)	c Falso negativo (FN)		$\frac{c}{c+d} = \alpha$
	0	b Falso Positivo (FP)	a Verdadeiro Negativo (VN)	$\frac{b}{a+b} = \beta$	

Por meio da matriz de confusão é possível determinar a porcentagem de classificações corretas do modelo ajustado, que são as medidas de especificidade e de sensitividade. Sensitividade é a razão do grupo com classificação favorável do grupo com a variável de interesse (classificado $\hat{y} = 1$, observado $y = 1$) sobre o total desse grupo observado, ou seja:

$$sensitvidade = \frac{VP}{VP + FN}$$

A especificidade é a razão do outro grupo com classificação favorável, com a outra variável (classificação $\hat{y} = 0$ e observado $y = 0$) sobre o total desse grupo observado:

$$especificidade = \frac{VN}{VN + FP}$$

A razão geral do modelo de classificação correta é estimada como:

$$\frac{d + a}{a + b + c + d} \times 100\%$$

e o erro total do modelo de classificação como:

$$\frac{b + c}{a + b + c + d} \times 100\%$$

Segundo Chorão (2005) é importante realçar nessa matriz, vários aspectos importantes:

1) Erro tipo I

   Designado por α (dimensão do teste), é a razão de observações em situação 1 ($y = 1$) classificados como sendo 0 ($\hat{y} = 0$). Imagine uma instituição financeira que tenha uma taxa α elevada (clientes devedores sendo classificados como clientes regulares) significa que a instituição é muito generosa com a concessão de crédito estando, então, exposta ao risco de crédito.

2) Erro tipo II

   Designado por β (complementar da potência do teste) é a razão de observações em situação 0 ($y = 0$) classificados como 1 ($\hat{y} = 1$). Na instituição financeira citada, se β é elevado por um longo período haverá perdas nas vendas e concomitantemente quebra nos lucros. Esta instituição está exposta ao risco comercial, ou seja, ao risco de perda de quota de mercado.

3) Cutoff

   Os erros α e β estão dependentes do *cutoff* considerado para classificar a observação com 0 ou 1. Além disso, a matriz de confusão é muitas vezes usada para comparar diferentes modelos de classificação, tendo como hipótese que os dois tipos de erros têm a mesma importância para a instituição.

3.1.5.9 Área abaixo da curva ROC

A curva ROC (Receiver Operating Characteristic), também conhecida como curva de Lorenz (HENLEY; MCNEIL, 1982) é baseada nos conceitos de sensitividade e especificidade. Estatísticas (medida de classificação correta) que podem ser obtidas a partir da construção de matrizes de confusão criadas a partir do resultado da classificação dos indivíduos, gerado pelo modelo.

De acordo com Hosmer e Lemeshow (2000), para fazer a curva plota-se a probabilidade de detenção do verdadeiro sinal (sensitividade) e o falso sinal (1-especificidade) para completo alcance dos possíveis pontos de corte.

A área abaixo da curva ROC, que varia entre 0 e 1, fornece uma medida da capacidade do modelo discriminar entre indivíduos com o fator de interesse versus os que não tem o fator de interesse. Contudo, quando se considera um teste onde estão presentes duas populações, uma com indivíduos 1 (presença do fator de interesse) e outra de indivíduos 0 (ausência do fator de interesse), muito raramente se observa uma perfeita separação entre as duas populações. Os resultados deste teste apresentam uma sobreposição conforme nota-se na Figura 2.

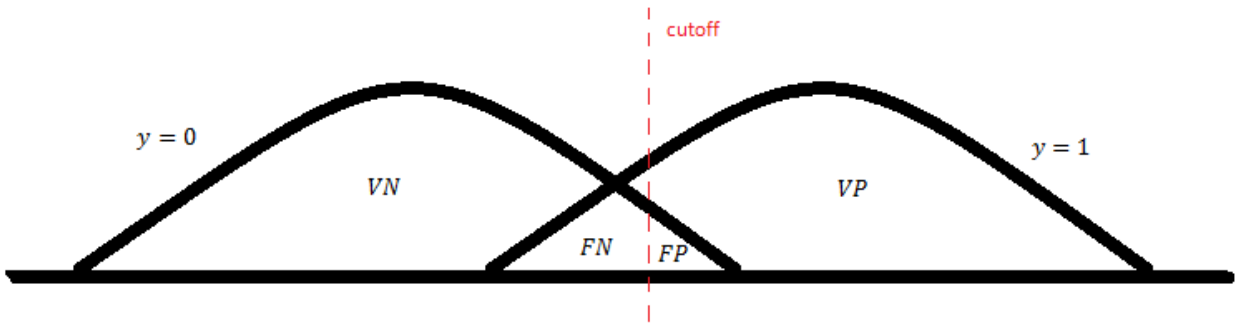

Figura 2 - Funções de densidade de duas populações

Para a direita do *cutoff* (teste positivo) identifica-se uma área correspondente ao falso positivo (FP) e outra ao verdadeiro positivo (VP). Para a esquerda do ponto de corte (teste negativo) identifica-se uma área correspondente aos falsos negativos (FN) e outra aos verdadeiros negativos (VN).

Quanto menor for a sobreposição das distribuições, menor é a área correspondente ao falso positivo. Assim, valores de corte elevado conduzem a um teste pouco sensível e muito específico; por outro lado, valores de *cutoff* baixos conduzem a um teste muito sensível e pouco específico.

O objetivo é escolher um ponto de corte ótimo, que maximize a escolha de sensibilidade e especificidade, deve-se plotar um gráfico semelhante ao gráfico da Figura 3, em que são sugeridos diversos pontos de corte e o ponto ótimo é o cruzamento da curva de sensibilidade e especificidade.

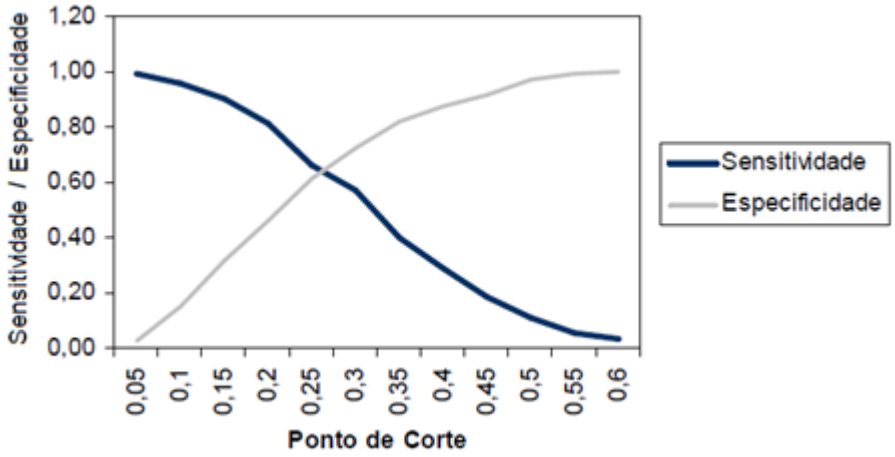

Figura 3 - Plotagem de Sensitividade e Especificidade contra os pontos de corte

Já a Figura 4 ilustra a curva ROC, cuja área abaixo da curva é a medida de discriminação (varia entre 0 e 1), ou seja, a capacidade preditiva do modelo classificar corretamente as observações como 0 ou 1.

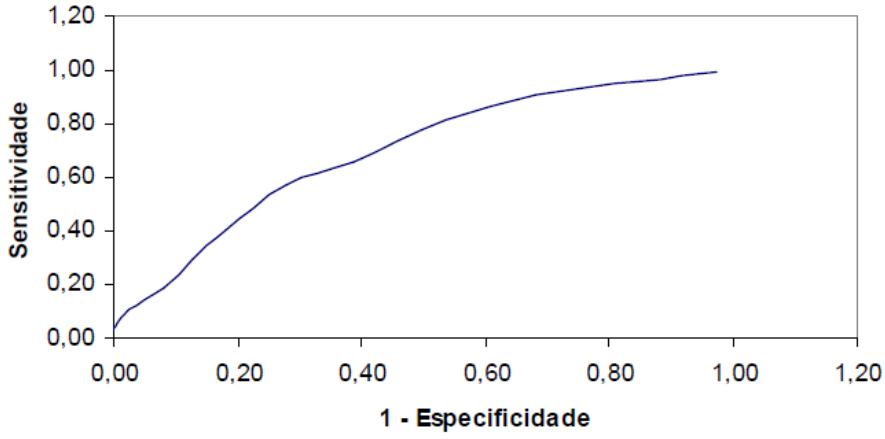

Figura 4 - Plotagem de Sensitividade versus 1- Especificidade para possíveis pontos de corte

O cálculo da área abaixo da curva ROC é bastante intuitivo: Seja n_1 o número de indivíduos com $Y = 1$ e n_0 o número de indivíduos com $Y = 0$. Existem $n_1 \times n_0$ pares em que os indivíduos com $Y = 1$ são combinados com os indivíduos com $Y = 0$. Destes, $n_1 \times n_0$ pares é determinada a proporção das vezes em que os indivíduos com $Y = 1$ tem a maior das 2 probabilidades. Imagine um caso em que tem-se 575 indivíduos. O número de observações com $Y = 1$ é 147 e $Y = 0$ é 428.

Logo, $147 \times 428 = 62916$ comparações podem ser feitas. Daí contando o número de vezes que o indivíduo com $Y = 1$ tem maior probabilidade que o indivíduo com $Y = 0$ tem-se $43972{,}5$ (contagem da estatística U de Mann-Whitney). Assim a razão $^{43972{,}5}/_{62926} = 0{,}6989$, em que $0{,}6989$ é a área abaixo da curva ROC .

Uma regra sugestiva para a intepretação da área abaixo da curva ROC é:

- Se $ROC < 0{,}5$ –discriminação péssima;
- Se $ROC = 0{,}5$ – sem discriminação (mostra que a discriminação não é melhor que uma chance ao acaso);
- Se $0{,}5 < ROC < 0{,}7$ – discriminação fraca;
- Se $0{,}7 \leq ROC < 0{,}8$ – discriminação aceitável;
- Se $0{,}8 \leq ROC < 0{,}9$ – discriminação excelente;
- Se $ROC \geq 0{,}9$ – discriminação excepcional.

3.2 Árvore de Decisão

A árvore de decisão é utilizada como um instrumento de apoio à tomada de decisão que consiste numa representação gráfica das alternativas disponíveis geradas a partir de uma decisão inicial. Uma das grandes vantagens de uma árvore de decisão é a possibilidade de transformação/decomposição de um problema complexo em diversos sub-problemas mais simples.

As Árvores de Decisão tem-se tornado populares para explorar, identificar e classificar estruturas complexas, exigindo-se que tenham um tamanho amostral razoável para a obtenção de bons resultados (MCLACHLAN, 1992). Existem dois tipos de árvores de decisão: árvores de regressão, quando a variável resposta é quantitativa e as árvores de classificação, quando a variável resposta é classificatória. Neste estudo aborda-se apenas as árvores de classificação.

Segundo Berry e Linoff (2004), Árvore de Decisão é uma ferramenta muito poderosa e amplamente popular para classificação e predição, sendo seu grande atrativo o fato de que árvores de decisão representam regras que podem ser

expressas em linguagem comum, de modo que os seres humanos possam entendê-las.

O algorítmo da árvore de decisão é muito flexível porque opera com todos os tipos de variáveis, seja nas variáveis independentes como na dependente, não impondo nenhuma restrição às suas distribuições. Uma árvore de decisão tem o poder de discriminar porque decompõe a relação complexa existente entre a variável resposta e as várias variáveis explicativas em sub-problemas mais simples usando, recursivamente, a mesma estratégia em cada sub-problema. O objetivo é encontrar uma árvore com a menor taxa de erro, menor complexidade, com poucos nós terminais e que esteja adequada aos objetivos do estudo, tornando-se fácil de interpretar.

Uma árvore de decisão representa uma segmentação hierárquica dos dados. O segmento original é o conjunto de dados inteiro que é conhecido como o nó raiz da árvore. Ele é o primeiro a ser dividido em dois ou mais segmentos por meio da aplicação de uma série de regras simples. Cada regra atribui uma observação para um segmento com base no valor de uma entrada para essa observação. De um modo semelhante, cada segmento resultante é ainda dividido em sub-segmentos, cada sub-segmento é dividido em mais sub-segmentos e assim por diante. Esse processo continua até que o particionamento não seja mais possível. Tal processo de segmentação é conhecido como particionamento recursivo e resulta em uma hierarquia de segmentos dentro de segmentos. A hierarquia é chamada de árvore e cada segmento ou sub-segmento é chamado de nó.

Qualquer segmento ou sub-segmento que está dividido em mais sub-segmentos é conhecido como *nó intermediário*. Um nó com todos os seus sucessores forma um *ramo da árvore*. Os segmentos finais que não são mais particionados são conhecidos como nós terminais ou *folhas da árvore*. Cada folha é definida por uma combinação única de regras usadas previamente. As folhas são subconjunto disjunto dos dados originais, não há sobreposição entre eles e cada registro no conjunto de dados pertence a uma e somente uma folha.

Um modelo de árvore de decisão é composto por:

- definição do nó, ou regra, a fim de atribuir a cada registro de um conjunto de dados um nó folha;
- probabilidades posteriores de cada nó folha;
- a atribuição de um nível pretendido para cada folha;

Definições do nó são desenvolvidos usando os dados de treinamento e são expressos por regras simples. Probabilidades posteriores são calculadas para cada nó usando os dados de treinamento. A atribuição do nível pretendido para cada nó é feito também durante a fase de treinamento e as probabilidades posteriores são dadas pela proporção de níveis da variável resposta dentro de cada nó e a atribuição do nível é baseada nessa probabilidade, quando não se tem nenhum outro atributo em questão, como o custo ou despesas.

Imagine um exemplo em que um futuro sorveteiro quer saber o que predispõe as pessoas a comprarem sorvete. Entre todas as pessoas observadas, 46% compra sorvete. Esta população é representada no nó raiz da árvore, no topo do diagrama. A Figura 5 mostra detalhadamente o caminho da árvore e suas regras.

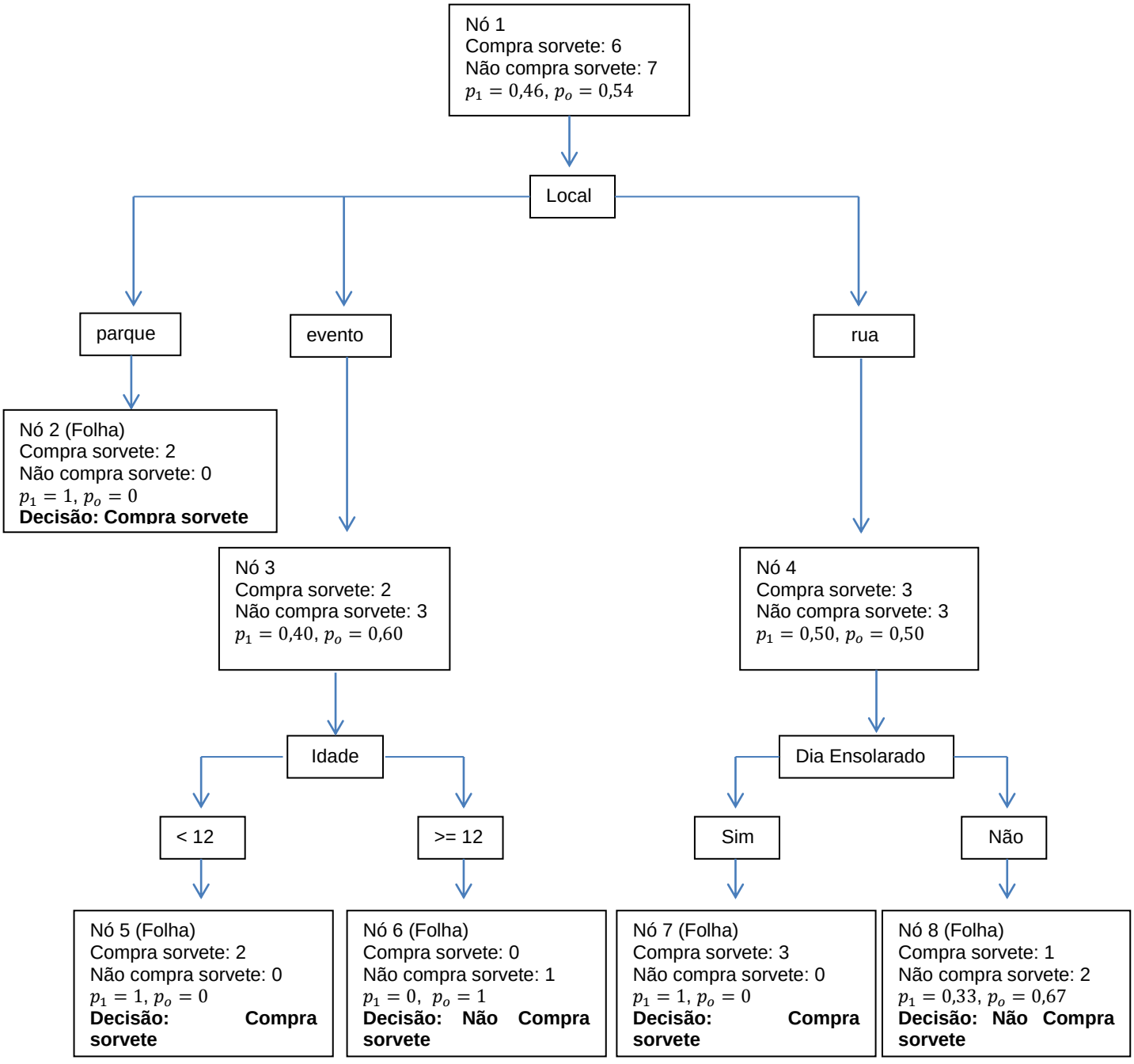

Figura 5 - Exemplo de árvore de decisão para uma variável *target* binária (compra ou não compra)

Árvores de decisão simples são atraentes porque possuem uma representação clara de como as variáveis independentes determinam o alvo. Árvores também são atraentes porque aceitam vários tipos de variáveis: nominal, ordinal e intervalar. Variáveis nominais têm valores categóricos sem ordem inerente. Variáveis ordinais são categóricas com valores ordenados, por exemplo: ' frios', 'bom', 'quente', e 'muito quente'. Variáveis intervalares são variáveis que podem ser calculadas. Temperatura é uma variável intervalar, quando seus valores são expressos em graus. Uma variável pode ser de qualquer tipo, independentemente dela servir como uma variável *target* (o propósito para criação da árvore) ou como uma variável *input* (as variáveis de entrada para o modelo - são aquelas variáveis disponíveis para uso nas regras de divisão).

As árvores também têm suas deficiências. Quando os dados não contêm uma relação simples entre as variáveis de entradas e a variável resposta, a árvore pode acabar sendo uma árvore simplista demais. Uma árvore dá a impressão de que certos insumos exclusivamente explicam as variações no alvo. Um conjunto completamente diferente de insumos poderia dar uma explicação diferente e talvez até melhor. E como mencionado anteriormente, sempre procura-se por uma árvore com a menor taxa de erro, menor complexidade, com poucos nós terminais e que esteja adequada aos objetivos do estudo, tornando-se fácil de interpretar.

3.2.1 Utilização da Árvore de Decisão

As árvores de decisão não são necessariamente utilizadas apenas para modelagem preditiva. Existe uma lista de opções para a utilização de uma árvore de decisão, que são:

3.2.1.1 Seleção de variáveis

Os dados chegam ao analista, normalmente, com muitas variáveis. A primeira missão é encontrar alguma coisa interessante nos dados, que normalmente contém variáveis redundantes ou irrelevantes que ficam no caminho. A tarefa preliminar é determinar quais variáveis são susceptíveis de ser preditiva.

Uma prática comum é excluir variáveis de entrada (independente) com pouca correlação com a variável resposta. Uma prática alternativa é a utilização de insumos que aparecem nas regras de divisão de uma árvore. Árvores avisam relações a partir da interação dos insumos. Por exemplo, comprar sorvete pode não ter correlação com o Local a menos que o tempo esteja ensolarado e quente. A árvore nota as duas entradas. Além disso, as árvores descartam entradas redundantes. Dia ensolarado e temperatura, por exemplo, podem se correlacionar com a compra de sorvetes, mas a árvore só precisa de uma das entradas.

O analista usaria, normalmente, as variáveis selecionadas como as variáveis de entrada em um modelo como o de regressão logística, por exemplo. Porém as árvores não selecionam todas as variáveis importantes para uma regressão. A solução sensata é incluir algumas variáveis a partir de outra técnica, tais como correlação. Nenhuma técnica de seleção é capaz de profetizar quais variáveis vão ser eficazes em outras ferramentas de modelagem.

3.2.1.2 Importância da variável

O analista pode querer usar técnicas de seleção de variáveis para fornecer uma medida de importância de cada variável, em vez de apenas enumerá-las. Intuitivamente, as variáveis usadas em uma árvore têm diferentes níveis de importância. O que torna uma variável importante é a força da influência e o número de casos influenciados.

Alguns *softwares* implementam uma fórmula que define a importância de uma regra de divisão: para uma variável *target* intervalar, a importância de uma divisão é a redução na soma de erros quadrados entre o nó e os ramos imediatos. Para uma variável *target* categórica, a importância é a redução no índice de Gini, normalmente.

3.2.1.3 Detecção de interação

A partir das variáveis selecionadas em uma regressão, normalmente considera-se possíveis efeitos de interação. Considere a modelagem do preço de casas familiares. Suponha que os preços da maioria das casas no conjunto de

dados são proporcionais a uma combinação linear da metragem quadrada e a idade da casa, mas as casas que fazem fronteira um campo de golfe são vendidas a um preço acima do que seria esperado a partir da combinação do tamanho e idade. Para criação do melhor modelo seria necessário um indicador que informe se a casa faz fronteira com o campo de golfe ou não. Dados raramente vêm com as variáveis mais úteis!

No entanto, parece plausível que as casas que fazem fronteira com o campo de golfe são aproximadamente do mesmo tamanho e foram construídas na mesma época. Se nenhuma das outras casas forem desse tamanho e nem foram construídas durante esse tempo, então essa combinação de tamanho e tempo fornece uma indicação sobre a casa fazer fronteira com o campo de golfe. A regressão deve conter três variáveis: metragem quadrada, idade e o indicador de campo de golfe. O indicador é construído a partir da metragem quadrada e idade, portanto, representa uma interação entre esses dois insumos.

Normalmente tenta-se multiplicar o tamanho pela idade, porém não seria significativo. Uma sugestão, então, é desenvolver uma árvore e criar um indicador para cada folha. Para uma observação particular, o indicador é igual a um (1) quando a observação pertence a aquela folha e caso contrário é igual a zero (0). A regressão conterá metragem quadrada, idade, e vários indicadores, um para cada folha da árvore. Se a árvore cria uma folha com apenas as casas que fazem fronteira com o campo de golfe, então, terá-se-á incluido os efeitos de interação direita. Os indicadores para as outras folhas não iriam estragar o ajuste. Indicadores para nós não-folha são desnecessários porque seriam iguais a soma de indicadores de seus descendentes.

3.2.1.4 Valores faltantes

É comum trabalhar com dados nos quais boa parte das variáveis contém uma quantidade considerável de dados faltantes. Árvores de decisão são mais tolerantes à falta de dados do que os modelos de regressão, por exemplo. Em uma regressão, ao combinar várias entradas, uma observação faltante em qualquer variável *input* deve ser descartada. Para o mais simples dos algoritmos de árvore, as

observações que precisam ser excluídas são aquelas em que não se tem a variável *target*.

Valores faltantes podem causar uma perda enorme de dados em dimensões elevadas. Por exemplo, suponha que cada uma das k variáveis de entrada tenha α por cento de dados faltantes. Nesta situação, a proporção esperada de dados disponíveis (sem *missing*) é definida por $(1-\alpha)^k$. Se tem-se 1% de dados ausentes $(\alpha = 0{,}01)$ para 100 variáveis *input*, tem-se apenas 37% dos dados para análise. No caso de 200 variáveis com $\alpha = 0{,}01$, tem-se 13% dos dados disponíveis e se forem 400 variáveis com um mesmo α, tem-se apenas 2% de informação. Se os dados faltantes aumentarem para 5% $(\alpha = 0{,}05)$, tem-se menos de 1% dos dados disponíveis, com 100 variáveis de entrada.

Trabalhando com uma regressão com dados faltantes pode-se substituir primeiro os valores em falta, por palpites. Isso é chamado de imputação, uma abordagem natural é a de ajustar um modelo com os valores não-*missing* para prever os que faltam. Árvores podem ser a melhor ferramenta de modelagem para este fim, por causa de sua tolerância à falta de dados, a sua aceitação de diferentes tipos de dados e sua robustez nas suposições sobre as distribuições das variáveis de entrada. Para cada entrada da regressão, construir uma árvore que use as outras variáveis de entrada para prever o dado faltante. Ou seja, se X, Y e Z representam as variáveis de entradas (*input*), cria-se, então, uma árvore para prever X em função de Y e Z, outra árvore para prever Y em função de X e Z, e outra para prever Z dado X e Y.

3.2.1.5 Interpretação do modelo

Árvores são, por vezes, usadas para ajudar a compreender os resultados de outros modelos, um exemplo ocorre em pesquisa de mercado. Uma empresa pode oferecer muitos produtos e diferentes clientes estão interessados em produtos diferentes. Uma tarefa de pesquisa de mercado é segregar os potenciais clientes em segmentos homogêneos e em seguida, atribuir campanhas de *marketing* para esses segmentos. Normalmente, nenhuma informação está disponível sobre a resposta dos clientes e assim nenhuma variável *target* existe.

Segmentação é baseada em similaridades entre as variáveis de entrada. As pessoas diferem um pouco em suas opções de compra dependendo da sua demografia: idade, situação familiar e onde vivem. Informações demográficas são relativamente fáceis de se obter, e os dados faltantes, muitas vezes, podem ser imputados utilizando informações do censo.

Após os segmentos serem construídos, a idade média, renda e outras estatísticas estão disponíveis para cada um deles. No entanto, essas estatísticas demográficas não são muito sugestivas de quais produtos o segmento está interessado. O próximo passo, então, é selecionar uma amostra de cada segmento e perguntar às pessoas sobre seu estilo de vida e preferências de produtos. Por fim, combina-se as amostras de todos os segmentos em um único conjunto de dados e cria-se uma árvore usando a perguntas da pesquisa como variaveis de entrada e o número do segmento como a variável *target*. Usando apenas alguns segmentos com um número igual de pessoas em cada um aumenta a chance de se obter uma árvore útil. A idéia é que a árvore caracterize alguns segmentos pelo tipo de roupas, carros, ou hobbies que sugerem quais produtos cada segmento de pessoas gostaria de comprar.

3.2.1.6 Modelagem preditiva

Como listado anteriormente, a árvore pode ajudar a superar alguns obstáculos na modelagem preditiva, em cada exemplo a árvore ajuda a preparar os dados ou interpretar os resultados de um outro modelo preditivo. No entanto, muitos autores compartilham a idéia comum de que as árvores por si só são eficazes modelos preditivos (MORGAN; SONQUIST, 1963; KASS, 1980; BREIMAN et al.,1984; QUINLAN, 1979). Cada autor pode descrever estudos em que as árvores foram usadas para predição.

Árvores não substituem outras técnicas de modelagem. Trata-se apenas de mais uma técnica disponível para análise, que pode ser usada para vários objetivos.

3.2.2 Como construir uma árvore de decisão

Para que uma árvore seja construída com sucesso é necessário que os dados sejam divididos utilizando o método do particionamento recursivo. Existem diversas formas de divisão e de seleção de qual variável será usada em cada regra. Disserta-se a seguir os pontos mais importantes para o estudo em questão. Utiliza-se como premissa o fato da variável *target* ser uma variável binária e descreve-se os métodos possíveis.

3.2.2.1 Como uma regra é criada usando uma divisão binária

Na divisão binária, dois galhos são criados em cada nó. Quando uma variável intervalar é utilizada para particionar as observações em dois grupos, um valor específico dessa variável pode ser escolhido. Por exemplo, imagine a variável investimento (valor investido no último mês), um possível valor para a quebra poderia ser R$4.000,00. As observações com investimento menor que “valor da quebra” (R$4.000,00) são armazernados no galho esquerdo e as observações com investimento maior ou igual ao “valor da quebra” serão armazenados no galho direito. No caso de múltiplas divisões, mais de dois galhos são criados a partir de um nó. Por exemplo, a variável investimento poderia ser dividida como R$2.000,00 - R$4.000,00, R$4.000,01 - R$6.000,00, R$6.000,01 - R$8.000,00, etc.

Com o propósito de dividir qualquer segmento ou sub-segmento do conjunto de dados em um nó, necessita-se calcular algum valor que mensure qual seria a melhor divisão, dado todas as variáveis de entrada, mais o possível “valor de quebra” de cada uma delas. A idéia é localizar o melhor valor de quebra dentro de uma variável e comparar esse valor com todos os outros valores de quebra das outras variáveis *input*. O método de cálculo desse “valor” que mensura qual o melhor valor de quebra pode ser feito de diversas formas.

O processo de seleção da melhor separação consiste em duas etapas. No primeiro passo, o melhor valor de separação para cada entrada é determinado. Na segunda etapa, a melhor variável *input* dentre todas as variáveis de entrada é selecionada por meio da comparação do valor da melhor divisão de cada variável com o valor da melhor divisão das outras variáveis e seleciona-se a variável *input*

cujo valor de separação produz o maior valor. Este processo pode ser ilustrado pelo seguinte exemplo:

Suponha-se que existam 50 variáveis explicativas em um determinado estudo, representadas por $X_1, X_2, \dots, X_{50}$. O algoritmo da árvore começa com a variável X_1 e examina todas as candidatas divisões na forma $X_1 < C$, em que C é um valor de separação que está entre o mínimo e o máximo dos valores de X_1. Todas as observações que tiverem $X_1 < C$ irão para o nó filho da esquerda e todas as observações em que $X_1 \geq C$ irão para o nó filho da direita. O algoritmo percorre todos os possíveis valores de divisão na mesma variável de entrada e seleciona o melhor valor de divisão. Imagine que para a variável X_1 o melhor valor de separação seja C_1. Esse mesmo processo é repetido para X_2 e também para $X_3, X_4, \dots, X_{50}$ até definirem-se os melhores valores de divisão como sendo $C_2, C_3, \dots, C_{50}$. Tendo encontrado o melhor valor de separação para cada variável de entrada, o algoritmo compara esses valores para encontrar a variável de entrada cujo melhor valor de separação oferece a melhor repartição dentre todas as variáveis testadas. Suponha que C_{21} é o melhor valor de divisão para a variável X_{21} e suponha que X_{21} é escolhida como a melhor variável para realizar a divisão do nó. Por conseguinte, o nó é particionado usando a variável X_{21} de entrada. Todos os registros com $X_{21} < C_{21}$ são enviados para o nó filho esquerdo e todos os registros com $X_{21} \geq C_{21}$ são enviados para o nó filho direito. Este processo é repetido para cada nó. Variáveis diferentes podem ser selecionadas em nós diferentes.

3.2.2.2 Mensurar a importância de uma divisão quando a variável resposta é binária

O valor que representa a importância da separação pode ser mensurado de diversas formas e é terminado pelo analista responsável. Quando a variável resposta é binária ou categórica com mais de 2 níveis, existem duas maneiras de mensurar a importância da divisão: pelo grau de separação alcançado na divisão, ou pela redução da impureza atingida na separação. Normalmente o grau de separação é medido pelo valor-p do teste Qui-Quadrado de Pearson e a redução de impurezas é medido pela redução da entropia ou pela redução do índice

de Gini. Já quando a variável resposta é contínua, essa importância pode ser mensurada pelo teste F, que testa cada grau de separação para os nós filhos.

3.2.2.2.1 Grau de separação

Todas as separações bidirecionais dividem um nó pai em dois nós filhos. *Logworth* é uma medida de como esses nós filhos diferem um do outro. Quanto maior for a diferença entre os dois nós filhos e quanto maior o grau de separação alcançado pela divisão, melhor a divisão é considerada.

Imagine uma situação em que a variável resposta seja uma variável binária, sendo 1 o indivíduo respondente e 0 o não-respondente e a variável investimento seja uma variável explicativa. Cada linha do conjunto de dados representa uma observação (ou indivíduo). A Tabela 1, a baixo, mostra uma vista parcial do conjunto de dados, que estão expostos ordenados pela variável investimentos.

Tabela 1 - Demonstração de uma base de dados com variável resposta binária

Indivíduos (Obervações)	Resposta	Invenstimento (R$)
1	0	2000
2	0	3000
...	...	...
278	1	10000
...	...	...
10.000	1	200000

Os dados mostrados na Tabela 1 podem ser divididos em diferentes valores da variável investimento. Em cada valor de separação, uma tabela de contingência 2x2 pode ser construída, como mostrado na Tabela 2 (exemplo de uma divisão). As colunas representam os dois nós filhos que resultarão da divisã, e as linhas representam o comportamento da variável resposta.

Tabela 2 - Tabela de Contigência quando a divisão é realizada em R$2.000 da variável investimento

	$Invenstimento < R\$2.000$	$Invenstimento \geq R\$2.000$	Total
Respondente (1)	n_{11}	n_{12}	$n_{1\cdot}$
Não-Respondente (0)	n_{01}	n_{02}	$n_{0\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	n

Para avaliar o grau de separação alcançado por uma divisão, é necessário calcular o valor da estatística qui-quadrado (χ^2) e testar a hipótese nula de que a proporção de respondentes entre aqueles com investimentos menores que $R\$2.000$ não é diferente daqueles com investimento maior ou igual a $R\$2.000$. Isto pode ser escrito como:

$$H_0: P_1 = P_2, em\ que\ \hat{P}_1 = \frac{n_{11}}{n_{\cdot 1}}, \hat{P}_2 = \frac{n_{12}}{n_{\cdot 2}}, \hat{P} = \frac{n_{1\cdot}}{n}$$

Sob a hipótese nula, o valor esperado de cada casela é exposto na Tabela 3.

Tabela 3 - Tabela de Contigência quando a divisão é realizada R$2.000 da variável investimento, sob a hipótese nula

	$Invenstimento < R\$2.000$	$Invenstimento \geq R\$2.000$
Respondente (1)	$E_{11} = \hat{P} \times n_{\cdot 1}$	$E_{12} = \hat{P} \times n_{\cdot 2}$
Não-Respondente (0)	$E_{01} = (1 - \hat{P}) \times n_{\cdot 1}$	$E_{02} = (1 - \hat{P}) \times n_{\cdot 2}$

A estatística qui-quadrado é calculada da seguinte forma:

$$\chi^2 = \sum_{1=0}^{1} \sum_{j=1}^{2} \frac{(n_{ij} - E_{ij})^2}{E_{ij}}$$

O *valor-p* de χ^2 é encontrado resolvendo a equação $P(\chi^2 > \chi^2 calculado\ |hipótese\ nula\ verdadeira) = valor - p$. O *logworth* é simplesmente calculado como $Logworth = -log_{10}(valor - p)$. Quanto maior for o *logworth* (e, por conseguinte, quanto menor for o valor-p), melhor será a separação.

Imagine que este primeiro *logworth* calculado a partir da primeira divisão é chamado $logworth_1$. Outra separação é feita no próximo nível do rendimento (por exemplo $R\$3.000$), outra tabela de contingência é feita, e o *logworth* é calculado da mesma maneira. O nome desse novo cálculo é $logworth_2$. Se existem 100 valores distintos para a variável investimento no conjunto de dados, 99 tabelas de contingência serão criadas, e o *logworth* calculado para cada uma. O valor calculado para o *logworth* de cada tabela de contingência são $logworth_1, logworth_2, \ldots logworth_{99}$. A divisão que resulta no maior *logworth* é selecionada.

Suponha que o melhor valor de divisão de investimento é de $R\$15.000$, com o *logworth* de $20{,}5$. Agora considere a próxima variável, Idade. Se há 67 valores distintos de idade nos dados, 66 divisões serão consideradas. Considerando a melhor divisão de Idade como 35, com o *logworth* de $10{,}2$. Se a idade e o investimento são as únicas variáveis explicativas no conjunto de dados, então a variável investimento é selecionada para dividir o nó porque tem o maior valor *logworth*. Assim, o conjunto de dados será dividido em $R\$15.000$ de investimento. Essa divisão pode ser chamada de a melhor das melhores possíveis divisões.

Se houver 200 variáveis explicativas no conjunto de dados, o processo de encontrar a melhor divisão será realizada 199 vezes (uma para cada variável de entrada) e repetido isso para cada nó dividido. Cada variável de entrada deve ser examinada e a melhor divisão encontrada é aquela com o maior *logworth*. Esta será escolhida como a melhor das melhores divisões.

3.2.2.2.2 Redução da impureza como medida para mensurar a importância de uma quebra

Impureza de um nó é o grau de heterogeneidade no que diz respeito à composição dos níveis da variável resposta. Se nó v é dividido em nós filhos a e b, e se ω_a e ω_b são as proporções de registos nos nós a e b, então, a diminuição da impureza é $i(v) - \omega_a\, i(a) - \omega_b i(b)$, em que $i(v)$ é o índice de impureza de nó v, e $i(a)$ e $i(b)$ são os índices de impureza dos nós filho a e b, respectivamente.

Para dividir o nó v em dois nós filhos a e b baseado no valor divisão da variável de entrada X_1, o algoritmo da árvore examina todos os candidatos que se dividem da forma $X_1 < C_j$ e $X_1 \geq C_j$, em que C_j é um número real entre o valor mínimo e máximo da variável X_1. Os registros que têm $X_1 < C_j$ irão para o nó filho esquerdo e os registros em que $X_1 \geq C_j$ irão para a direita. Suponha que há 100 candidatos a divisão na variável X_1. Os valores candidatos são $C_j,\ j = 1{,}2{,}\ldots{,}100$. O algoritmo compara a redução de impurezas sobre estes 100 divisores e seleciona o que atingiu maior redução como o valor para a melhor divisão.

3.2.2.2.2.1 Índice de impureza GINI

Se p_1 é a proporção de respondentes em um nó, e p_0 é a proporção de não-respondentes, o índice de impureza Gini para aquele nó é definido como $i(p) = 1 - p_1^2 - p_0^2$. Se dois registros são escolhidos de forma aleatória (com reposição) a partir de um nó, a probabilidade de que ambos sejam respondentes é p_1^2, enquanto que a probabilidade de que ambos sejam não-respondentes é p_0^2, e a probabilidade de que eles sejam ou ambos respondentes ou ambos não-respondentes é $p_1^2 + p_0^2$. Assim, $1 - p_1^2 - p_0^2$ pode ser interpretado como a probabilidade de que qualquer um dos dois elementos escolhidos ao acaso (com reposição) são diferentes. Para variáveis respostas binárias, o índice de Gini simplifica para $2p_1(1 - p_1)$. Um nó puro tem um índice Gini igual a zero. Tal índice pode atingir o valor máximo de $\frac{1}{2} = 0{,}5$ quando ambas as classes são igualmente representadas.

3.2.2.2.2.2 Entropia

A entropia é uma outra medida de impureza do nó. É definida como $i(p) = -\sum_{i=0}^{1} p_i log_2(p_i)$ para variáveis respostas binárias. Um nó que tem uma entropia maior do que a de outro nó é mais heterogêneo e portanto, menos puro. A raridade de um evento é medido como $-log_2(p_i)$. Se um evento é raro, isso significa que a probabilidade de resposta de sua ocorrência, em um nó, é baixa. Suponha que a probabilidade de ser respondente em um nó é $0{,}005$. Em seguida, a raridade da resposta é $-log_2(0{,}005) = 7{,}644$. Este é um evento raro. A probabilidade de ser não-respondente é inversamente proporcional $0{,}995$; daí a raridade de não-respondentes é $-log_2(0{,}995) = 0{,}0072$. Um nó que tem uma resposta rara de $0{,}005$ é menos impuro do que um nó que tem proporções iguais de respondentes e não-respondentes. Assim, $-log_2(p_i)$ é grande, quando a raridade é alta e pequeno quando a raridade do evento é baixa. A entropia deste nó é definida por:

$$i(p) = -\sum_{i=0}^{1} p_i log_2(p_i) = -[0{,}005 \times log_2(0{,}005) + 0{,}995 \times log_2(0{,}995)] = 0{,}0454$$

Considere um outro nó em que a probabilidade de respondentes seja igual a probabilidade de não-respondentes ($0{,}5$). A entropia deste nó será:

$$i(p) = -\sum_{i=0}^{1} p_i log_2(p_i) = -[0{,}5 \times log_2(0{,}5) + 0{,}5 \times log_2(0{,}5)] = 1$$

O nó que é predominantemente de não-respondentes (com uma proporção de $0{,}995$) tem um valor de entropia de $0{,}0454$. Um nó com distribuição igual de respondentes e não-respondentes tem entropia igual a 1. Um nó que possui todos os respondentes ou todos os não-respondentes tem entropia a zero. Assim, a entropia varia entre 0 e 1, em que 0 indica a pureza máxima e 1 a impureza máxima.

3.2.2.3 Mensurar a importância de uma divisão quando a variável resposta é categórica

Se a variável resposta é categórica com mais de duas categorias (níveis), os procedimentos são os mesmos. As estatísticas de qui-quadrado serão calculadas a partir de tabelas de contingência $r \times b$, em que b é o número de nós filhos a serem criados com base em uma certa entrada e r é o número de níveis da variável *target* (categorias). Os valores-p são calculados a partir da distribuição de qui-quadrado com grau de liberdade igual a $(r-1)\,(b-1)$. O índice de Gini e de Entropia também podem ser aplicados neste caso, eles estão simplesmente prorrogados por mais de dois níveis da variável alvo.

3.2.2.4 Ajustes para o valor-p quando as variáveis explicativas têm diferentes níveis

Quando se compara as divisões de diferentes variáveis de entrada, os valores-p devem ser ajustados para levar em conta o fato de que nem todas as variáveis de entrada têm o mesmo número de níveis. Em geral, algumas entradas são binárias, algumas são ordinais, algumas são nominais e outras são intervalares.

Por exemplo, uma variável como compra ou não compra sorvete, chamada de compra. Para esta variável (compra), apenas uma divisão é avaliada,

apenas uma tabela de contingência é considerada, e apenas um teste é realizado. Uma variável explicativa como Idade pode assumir qualquer valor inteiro maior que 0. Suponha que existam 67 possíveis valores de Idade no conjunto de dados, 66 tabelas de contingência serão construídas e portanto, 66 testes qui-quadrado são calculados. Em outras palavras, sessenta e seis testes são realizados sobre esta entrada para selecionar a melhor separação.

Suponha que a $i-ésima$ divisão da variável Idade tenha um $p-valor = \alpha_i$, o que significa que $P(\chi^2 > \chi^2 calculado\,|hipótese\ nula\ verdadeira) = \alpha_i$. Em outras palavras, a probabilidade de encontrar um qui-quadrado maior do que o χ^2 calculado, de forma aleatória, é α_i, sob a hipótese nula. A probabilidade de que, a partir dos 66 testes qui-quadrado calculados sobre a variável Idade, pelo menos, um dos testes produz uma decisão falsa positiva (em que se rejeita a hipótese nula, dada que ela é verdadeira) é:

$$1 - \prod_{i=1}^{66}(1 - \alpha_i).$$

Esta taxa de erro do experimento é muito maior do que a taxa de erro individual de α_i. Por exemplo, se a taxa de erro indivídual (α_i) em cada teste é de $0{,}05$, em seguida, a taxa de erro do experimento é $1 - 0{,}95^{\,66} = 0{,}9661$. Isto significa que quando você tem múltiplas comparações χ^2 (uma para cada possível divisão), o valor-p subestima o risco de rejeitar a hipótese nula quando ela é verdadeira. Claramente, quanto mais possíveis divisões a variável tem, menos preciso os valores-p serão.

3.2.2.4.1 Ajuste de Bonferroni

Ao comparar a melhor divisão da variável Idade com a melhor divisão da variável compra, os *logworth´s* precisam ser ajustados para o número de divisões, ou testes, em cada variável. Neste caso da variável compra, há apenas um teste e portanto, não é necessário ajuste. Mas, no caso da variável Idade, a melhor separação é escolhida a partir de um conjunto de 66 divisões. Portanto, $log_{10}(66)$ é subtraído do *logworth* da melhor separação. Em geral, se uma entrada tem *m*

possíveis divisões, então $log_{10}(m)$ é subtraído do *logworth* de cada divisão da variável de entrada. Esse ajuste é chamado de ajuste de Bonferroni.

3.2.2.4.2 Ajuste de Profundidade

Pode-se chamar o ajuste baseado no número de divisões antecedentes como ajuste de profundidade, porque o ajuste depende da profundidade da árvore na qual a separação é feita. A profundidade é baseada no número de ramos criados anteriormente ao nó em questão.

O valor-p calculado é multiplicado por um multiplicador de profundidade, com base na profundidade da árvore no nó em questão, para chegar ao valor-p ajustado à profundidade da divisão. Por exemplo, suponha que, antes do nó em questão havia quatro divisões (quatro divisões foram realizadas a partir do nó raiz até o nó atual) e que cada divisão envolveu dois ramos (usando divisão binária). Neste caso, o multiplicador de profundidade é $2 \times 2 \times 2 \times 2$. Em geral, o multiplicador de profundidade para divisões binárias 2^d, em que d é a profundidade, ou seja, o número de ramos, a partir do nó raiz até o nó atual.

O valor-p calculado é ajustado por meio da multiplicação pelo multiplicador de profundidade. Isto significa que a uma profundidade de 4, se o valor-p calculado é $0{,}04$, o valor-p ajustado à profundidade será $0{,}04x16 = 0{,}64$. Sem o ajuste de profundidade, a separação teria sido considerada estatisticamente significativa. Mas após o ajuste, a separação não é estatisticamente significativa.

O ajuste de profundidade também pode ser interpretado como divisão do limiar do valor-p pelo multiplicador de profundidade. Se o limiar do valor-p especificado pelo nível de significância é $0{,}05$, então o valor ajustado será $0{,}05/16 = 0{,}003125$. Qualquer divisão com valor-p acima de $0{,}003125$ será rejeitada. Em geral, se α é o nível de significância especificado, então qualquer separação, que tem um valor-p acima de um $\alpha/multiplicador\ de\ profundidade$ é rejeitada.

O efeito do ajuste de profundidade é o de aumentar o valor do limiar do *logworth* por $log_{10}(2^d) = d\ log_{10}(2)$. Assim, quanto mais profunda for a árvore, mais a norma se torna rigorosa para aceitar uma divisão significativa. Isto leva à rejeição de mais divisões do que teria sido rejeitadas sem o ajuste de profundidade. Assim, o

ajuste de profundidade pode também, limitar o tamanho da árvore, aceitando menos divisões.

3.2.3 Controlar o crescimento da árvore: regras de parada

Regras de parada são aplicadas durante a fase de desenvolvimento da árvore para decidir se o particionamento recursivo foi realizado suficientemente. Existem algumas maneiras utilizadas para impedir o crescimento desnecessário da árvore. A seguir descrevem-se algumas alternativas:

Controlar o crescimento das árvores por meio do nível de significância. Pode-se controlar o tamanho inicial da árvore, definindo-se um limite para o valor-p. Por exemplo, definindo-se um nível de significância como 0,05, o *logworth* será limitado em $-\log_{10}(0{,}05)$ ou 1,30. Se, em qualquer nó, nenhuma das variáveis de entrada tem uma divisão com *logworth* superior ou igual ao limiar, então o nó não é particionado. Diminuindo o limiar do valor-p, aumenta-se o grau em que os dois nós filhos podem variar, a fim de considerar uma separação dos dados mais significativa. Assim, o crescimento da árvore pode ser controlado.

Controlar o crescimento das árvores por meio do ajuste de profundidade. Como mencionado anteriormente, o ajuste de profundidade ajusta o valor-p conforme o número de ramos anteriores ao nó. Em particular, se α é o nível de significância especificado, então, qualquer separação que tenha um valor-p acima de $\alpha/(\text{multiplicador de profundidade})$ será rejeitado. Assim, quanto mais profunda, mais rigorosa tornam-se as regras para aceitar uma divisão como significativa. Isto leva à rejeição de mais divisões do que sem o ajuste, resultando em menos partições.

Controlar o crescimento das árvores por meio do tamanho da folha. Pode-se controlar o crescimento da árvore, definindo-se um tamanho para a folha. Por exemplo, definindo o tamanho da folha como 100, isto significa que, se uma divisão resulta em uma folha com menos de 100 registros, essa divisão não deverá ser executada. Assim, o crescimento pára no nó atual.

Controlar o crescimento das árvores por meio do tamanho do nó a ser dividido. Por exemplo, se o tamanho do nó deve ser de 300 registros, isto significa

que, se um nó tem menos de 300 registros, então ele não deve ser considerado para a separação.

Controlar o crescimento das árvores por meio da profundidade máxima. Isso determina o número máximo de gerações de nós. O nó raiz é nó da geração, ou seja, zero e os filhos do nó raiz são os nós da primeira geração etc. Pode-se, então, controlar o crescimento da árvore especificando o número de gerações desejadas.

3.2.4 Poda: a seleção da árvore do tamanho certo

Após criar a maior árvore possível (árvore máxima) sob as regras de paradas estipuladas, necessita-se podar a árvore no tamanho correto. A idéia é começar com a árvore máxima e eliminar uma divisão em cada etapa. Por exemplo, se a árvore máxima tem F folhas e remove-se uma divisão em determinado ponto, encontra-se uma sub-árvore com $F-1$ folhas. Removendo-se outra divisão em outro ponto, encontra-se outra sub-árvore com $F-1$ folhas. Assim, pode-se encontrar n sub-árvores com tamanho $F-1$. Então, seleciona-se dentre todas as sub-árvores com $F-1$ folhas a melhor delas, a partir de algum critério de seleção, que serão descritos abaixo. Em seguida, remove-se outra divisão da sub-árvore com $F-1$ folhas e encontra-se, então, outra sub-árvore com $F-2$ folhas e, assim por diante, até encontrar uma árvore com uma única folha. No final deste processo, haverá uma sequência de árvores de tamanhos $F, F-1, F-2, F-3, \ldots, 1$. E para cada uma delas obtém-se a métrica, conforme o critério de seleção estipulado, a fim de se chegar na melhor árvore.

Alguns critérios para a seleção do modelo final incluem: minimização de custos, minimização da taxa de erro (misclassification), minimização do erro quadrado médio, ou maximização do *Lift*. No caso de uma variável *target* contínua, a minimização do erro quadrado médio é o critério mais utilizado. Outro critério possível consiste em comparar o lucro das sub-árvores em cada passo. Todos os cálculos realizados nas sub-árvores são realizados usando a base da dados de validação.

Imagine um estudo em que a variável resposta seja binária, contendo respostas 0 ou 1. Sendo classificado como 1 o indivíduo de interesse, entende-se que misclassification é uma taxa de erro encontrada a partir de um modelo. É uma

métrica utilizada em modelos com resposta categórica, em que estuda-se a taxa de erro no caso do modelo ter classificado um indivíduo como 1, quando na verdade ele é 0 ou então quando o modelo classifica-o como 0 quando na verdade ele é 1. A utilização deste critério para seleção do melhor modelo tem como objetivo minimizar o erro de classificação.

O erro quadrado médio é o quadrado da diferença entre o valor predito e o valor real. É a métrica mais apropriada para variáveis resposta contínuas. Já o *Lift* é utilizado para modelos com resposta categórica, como os que possuem um alvo binário. O *lift* é calculado como a divisão entre a taxa de resposta observada (proporção de registros classificados como 1) no topo de $n\%$ das observações da base de validação e a taxa de resposta global (proporção de respondentes 1 na base toda) nos dados de validação. O *ranking* é criado a partir da probabilidade predita (probabilidade do registro ser classificado como 1 na variável *target*) de resposta para cada registro no conjunto de dados de validação.

Parte-se agora para um exemplo real. A ilustração a seguir mostra o passo-a-passo da poda de uma árvore. A árvore máxima (Figura 6) foi construída utilizando os dados de treinamento com 10309 registros. As regras de partição foram seguidas e os nós foram classificados utilizando a base de treinamento.

Os dados de validação utilizados para a poda consistem em 8937 registros. As definições dos nós e a classificação deles são as mesmas das desenvolvidas com a base de treinamento, porém os registros em cada nó são construídos a partir da base de validação.

A Figura 6 mostra a árvore desenvolvida a partir dos dados de treinamento. O diagrama de árvore fornece: a identificação do nó, a identificação da folha, o número de respondentes no nó, o número de não-respondentes, o número total de registros em cada nó; proporção de respondentes (probabilidade posterior de resposta), proporção de não-respondentes (probabilidade posterior de não-resposta) e o rótulo da decisão em que as folhas são classificadas.

Quando a variável resposta é binária, as probabilidades posteriores são a proporção de respondentes e a proporção dos não-respondentes em cada nó. Em modelagem, essas probabilidades posteriores são utilizadas como predições das

probabilidades. A todos os registros em uma folha são atribuídos a mesma probabilidade predita de resposta.

A árvore consiste na criação de regras em cada folha. Começando a partir do nó raiz e indo para baixo para um nó terminal, pode-se ler a regra de cada folha de uma árvore. Estas regras são expressas por intervalos nas variáveis de entrada. As variáreis de entrada selecionadas pelo algoritmo de árvore neste exemplo fictício são: investimento, sexo e idade.

As regras dos nós folha são:

Folha 1: se o valor de investimento for menor que R$15.000 e se o sexo for Feminino, então, todos os integrantes deste nó folha serão classificados como respondentes (1).

Folha 2: se o valor de investimento for menor que R$15.000 e se o sexo for Masculino, então, todos os integrantes deste nó folha serão classificados como não-respondentes (0).

Folha 3: se o valor de investimento for maior ou igual a R$15.000 e se a Idade for menor que 35, então, todos os integrantes deste nó folha serão classificados como não-respondentes (0).

Folha 4: se o valor de investimento for maior ou igual a R$15.000 e se a Idade for maior ou igual a 35, então, todos os integrantes deste nó folha serão classificados como respondentes (1).

Neste exemplo, usam-se apenas as probabilidades para decidir se o nó será respondente ou não-respondente. Especificando uma matriz de custos, por exemplo, pode-se mudar a decisão inserindo essa nova informação, buscando minimizá-lo.

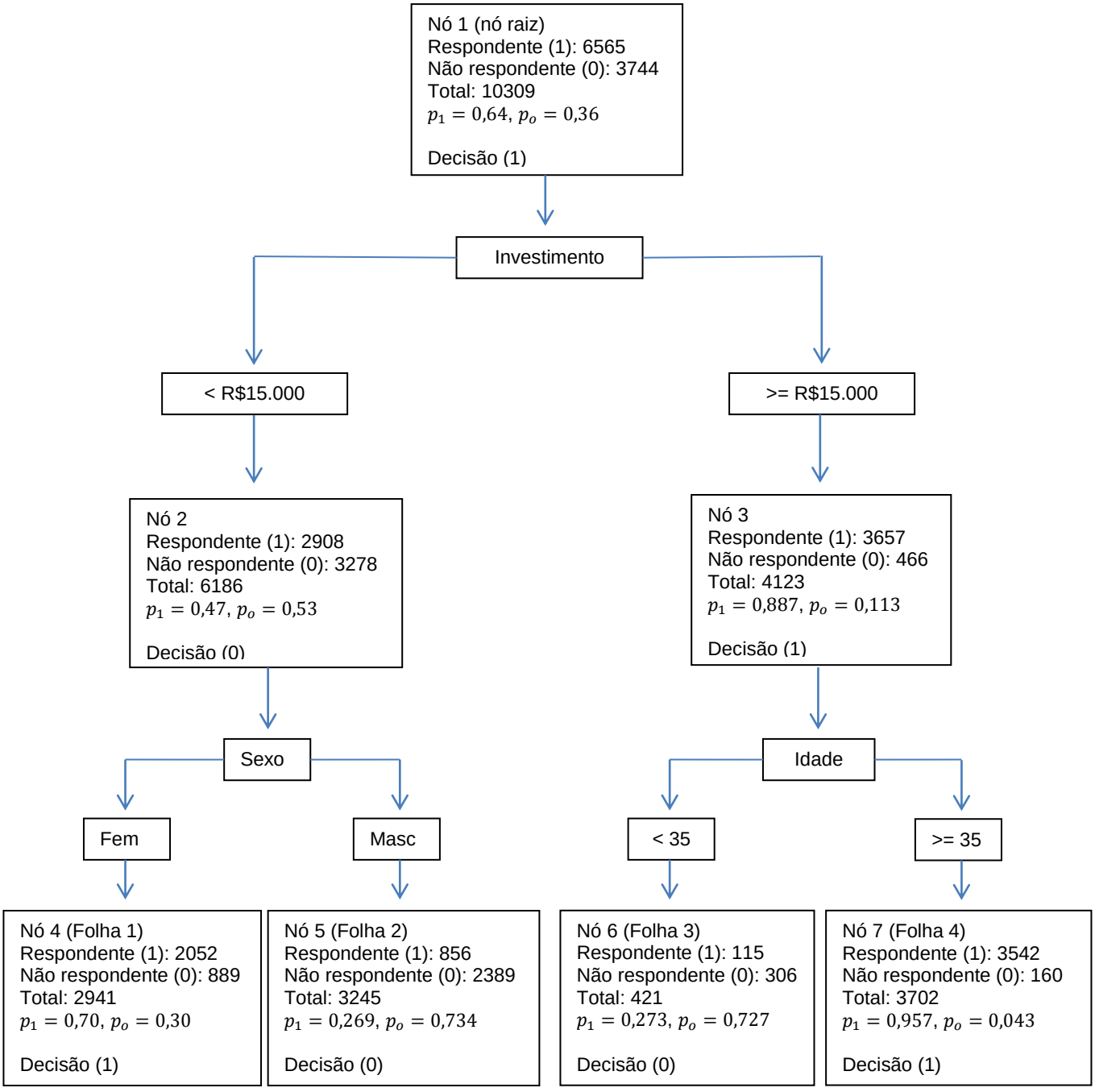

Figura 6 - Exemplo de árvore de decisão construída a partir dos dados de treinamento

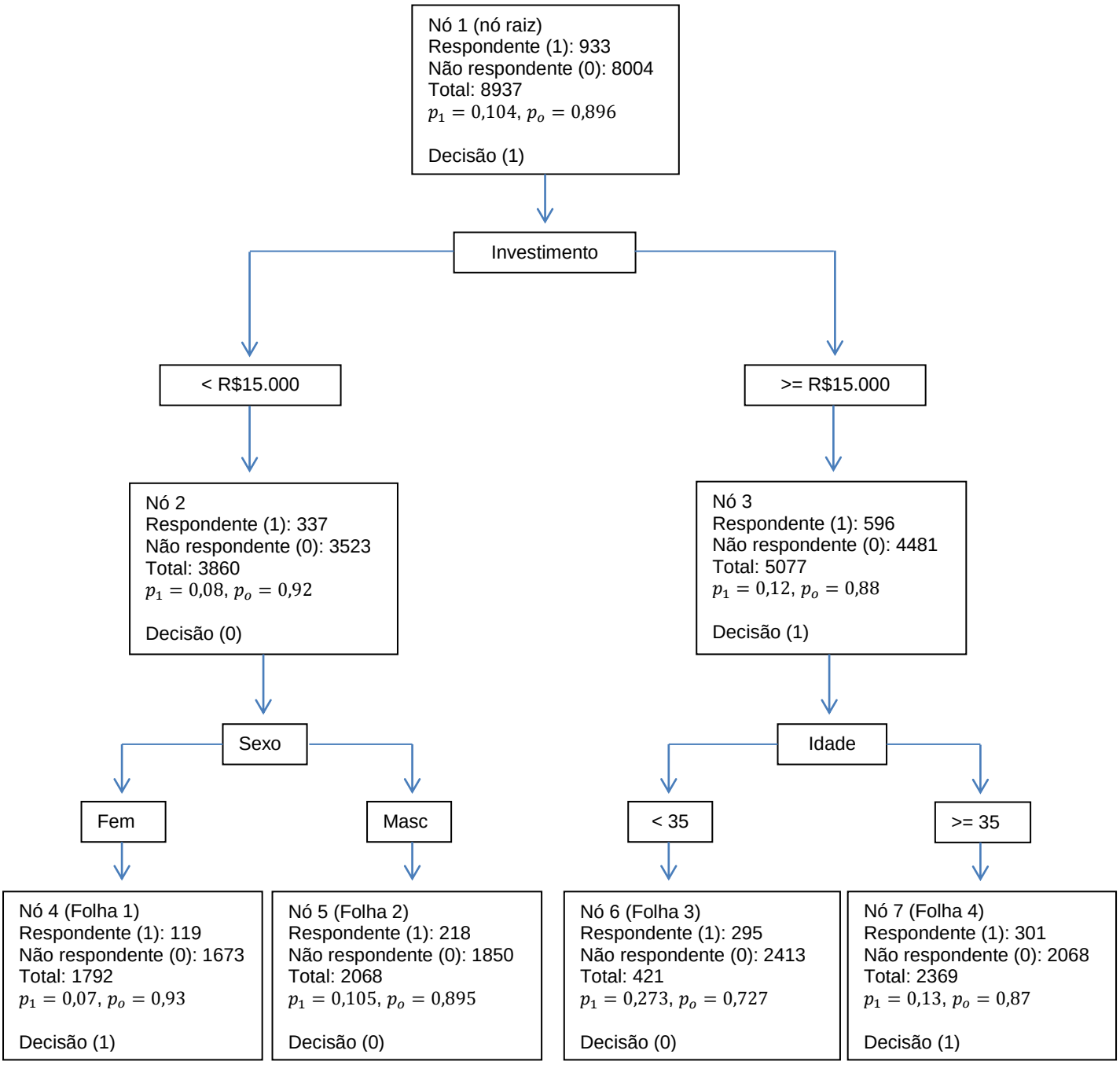

Figura 7 - Exemplo de árvore de decisão construída a partir dos dados de validação

A poda será realizada a partir dos dados de validação. Primeiro, as regras criadas serão utilizadas para dividir os dados de validação em diferentes nós. Uma vez que cada nó já tem atribuído um nível de destino com base nas probabilidades posteriores, pode-se calcular a taxa de erro de cada nó da árvore utilizando o conjunto de dados de validação. A Figura 7 mostra a aplicação da árvore para o conjunto de dados de validação.

Depois de aplicar as regras na base de dados de validação, tem-se uma árvore como a da Figura 3. Comparando a árvore a partir dos dados de validação (Figura 7) com a árvore a partir dos dados de treinamento (Figura 6), observa-se que as decisões em cada nó são exatamente as mesmas em ambos os diagramas. Isso ocorre porque as decisões são baseadas nas probabilidades posteriores geradas durante a criação da árvore, com a base de treinamento. Essas regras e decisões tornam parte do modelo e não mudam quando aplicados a um novo conjunto de dados.

A árvore na Figura 7 é a árvore máxima neste exemplo, com quatro nós folha. No entanto, dentro desta árvore existem várias sub-árvores de diferentes tamanhos. Existem duas sub-árvores com 3 nós folha, uma sub-árvore com 2 nós folhas e uma sub-árvore com apenas 1 nó folha (o nó raiz).

Podando-se os nós 6 e 7, obtém-se a sub-árvore com os nós folhas 3, 4 e 5 (sub_árvore_3_4_5). Podando-se os nós 4 e 5, obtém-se a sub-árvore com os nós folhas 2, 6 e 7 (sub_árvore_2_6_7). Podando-se os nós 4, 5, 6 e 7, tem-se a sub-árvore com 2 folhas (sub_árvore_2_3) e podando-se os nós 2 e 3, tem-se a sub-árvore com apenas 1 nó folha (sub_árvore_1).

Para cada uma das sub-árvore mais a árvore máxima, deve-se calcular a taxa de erro (misclassification) e escolher como melhor modelo, a árvore com menor taxa. O cálculo desta taxa pode ser entendido como uma matriz:

Target	Decisão (1)	Decisão (0)
1	1	0
0	0	1

Nesta matriz, se um respondente está classificado corretamente, então uma unidade de precisão é atingida. Se um não-respondente está corretamente

classificado como não-resposta, em seguida, uma unidade de precisão é adquirida. Caso contrário, não há ganho.

Como dito anteriormente, os nós são classificados como respondentes ou não-respondentes com base nas probabilidades posteriores calculadas a partir do conjunto de dados de treinamento. Na árvore criada (Figura 7) a proporção de respondentes é 10,4% e a proporção de não-respondentes é de 89,6%, no nó raiz. Assim, se o nó raiz é classificado como um nó respondente, a probabilidade predita será 0,104. O erro para esse nó será de 89,6%.

Para a sub-árvore com 3 nós, com os nós folha 4, 5 e 3, a taxa de erro é 0,71, ou seja, (1673+218+4481)/8937, em que 1673 é a quantidade de registros que foram classificados como 1 (decisão do nó 4), quando na verdade eram 0. O valor 218 é referente aos registros classificados incorretamente como 0, quando eram na verdade 1 (nó 5) e 4481 são os registros que foram classificados como 1, quando na verdade eram para ser 0 (nó 3).

Deve-se calcular a taxa de erro para cada sub-árvore listada acima. A Tabela abaixo mostra a taxa de erro para cada sub-árvore:

Sub-árvore	Taxa de erro
sub_árvore_4_5_6_7	0.475999
sub_árvore_2_6_7	0.302115
sub_árvore_3_4_5	0.712991
sub_árvore_2_3	0.539107
sub_árvore_1	0.895603

Como se observa a sub-árvore com 3 folhas, contendo os nós 2, 6 e 7, é a melhor escolha, baseado na minimização da taxa de erro.

3.2.5 Algoritmos Conhecidos

A lista, a seguir, contém os algoritmos mais conhecidos e descreve como eles trabalham. Cada algoritmo foi desenvolvido por uma pessoa ou grupo de pessoas inspiradas em criar algo melhor do que o que já existe. O último tópico são “os algoritmos SAS”. O *software* SAS permite que o usuário misture algumas das melhores idéias dos algoritmos mais conhecidos.

3.2.5.1 ID3

Este algoritmo, apresentado por J. R. Quinlan (QUINLAN, 1986), constitui uma das referências base dos algoritmos atuais de indução de árvores de decisão. Desenvolvido com vista ao tratamento de problemas contendo apenas características discretas, a sua estrutura básica é iterativa. Adotando o critério de maximização da informação para a escolha da característica que serão testadas em cada nó, a sua estrutura é muito simples no que se refere ao tratamento de problemas. Cada característica permite a divisão do conjunto de treino num número de subconjuntos igual à sua cardinalidade (número de diferentes valores possíveis).

O algoritmo ID3 (*Inductive Decision Tree*) segue os seguintes passos para construção de uma árvore de decisão:

1. Começar com todos os exemplos de treino;
2. Escolher o teste (atributo) que melhor divide os exemplos, ou seja agrupar exemplos da mesma classe ou exemplos semelhantes;
3. Para o atributo escolhido, criar um nó filho para cada valor possível do atributo;
4. Transportar os exemplos para cada filho tendo em conta o valor do filho;
5. Repetir o procedimento para cada filho não "puro". Um filho é puro quando cada atributo X tem o mesmo valor em todos os exemplos.

O algoritmo ID3 foi um dos primeiros algoritmos de árvore de decisão, tendo sua elaboração baseada em sistemas de inferência e em conceitos de sistemas de aprendizagem. Logo após foram elaborados diversos algoritmos, sendo os mais conhecidos: C4.5, CART (*Classification and Regression Trees*), CHAID (*Chi Square Automatic Interaction Detection*), entre outros.

3.2.5.2 C4.5

Apresentado no mais recente trabalho de Ross Quinlan (QUINLAN, 1993), este algoritmo visa a geração de árvores de decisão e de regras de classificação permitindo o tratamento de atributos discretos e/ou contínuos. Sendo possível a aquisição, juntamente com o livro citado, de um pacote de *software* sob a

forma de fontes que permite o teste e a avaliação de resultados. Embora o *software* tenha sido desenvolvido para a instalação em sistemas UNIX, foi adaptado para o ambiente MS-Windows utilizando o compilador Borland C++ 3.1 de forma a possuir uma plataforma única de execução dos vários algoritmos. Apesar desta adaptação, dado que a estimação de erro por validação cruzada é efetuada nesta versão do C4.5 (*release* 5), à custa de um ficheiro de comandos do sistema UNIX, a maioria das experiências utilizando esta técnica foram efetuadas neste sistema executando a versão original deste programa.

3.2.5.3 CART

O algoritmo CART - *Classification And Regression Trees* - foi apresentado por quatro estatísticos chamados Leo Breiman, Jerome Friedman, Richard Oslen e Charles Stone em uma de suas publicações (BREIMAN, 1984). Por ser um algoritmo não-paramétrico, uma das suas características principais é a grande capacidade de pesquisa de relações entre os dados, mesmo quando elas não são evidentes, bem como a produção de resultados sob a forma de árvores de decisão de grande simplicidade e legibilidade.

Tal como o seu nome indica, esta é uma metodologia que prevê o tratamento de variáveis dependentes discretas (classificação) ou contínuas (regressão) usando uma mesma tecnologia. O resultado deste algoritmo é sempre uma árvore binária que pode ser percorrida da sua raiz até às folhas respondendo apenas a questões simples do tipo sim/não. A análise é efetuada de forma completamente automática requerendo uma intervenção humana mínima. Segundo os autores, esta técnica permite a obtenção de resultados, em geral, superiores aos obtidos pelas técnicas estatísticas clássicas, sendo superado apenas num restrito número de casos e apenas por algoritmos de complexidade muito superior. No entanto, quando superado, a diferença nos resultados é mínima.

Este algoritmo é um exemplo de um algoritmo de partição binária recursiva. O processo é binário pois os nós efetuam uma partição em dois subconjuntos e recursivo pois é aplicado recursivamente a cada um dos

subconjuntos assim gerados, até que não seja possível ou não seja necessário efetuar mais nenhuma partição.

3.2.5.4 CHAID

CHAID é uma das técnicas para construção de uma árvore de decisão, baseada no teste de significância ajustado (teste de Bonferroni). A técnica foi desenvolvida na África do Sul e foi publicada em 1980 por Gordon V. Kass (KASS, 1980), que tinha completado sua tese de doutorado sobre este tema. CHAID pode ser usado para a predição (de uma maneira semelhante à análise de regressão), bem como, classificação e para a detecção de interação entre as variáveis.

CHi-squared Automatic Interaction Detection, CHAID, é um método exploratório para estudar as relações entre uma variável resposta e um conjunto de variáveis explicativas que podem interagir entre si. O método CHAID permite obter árvores de decisão com múltiplas categorias, ou seja, divisões com mais de duas opções. Para selecionar as variáveis explicativas relevantes para a explicação da variável resposta, o método em questão utiliza o teste do qui-quadrado quando trata-se de uma variável nominal como resposta, utiliza a razão de verossimilhança quando o variável resposta é ordinal e utiliza o teste F da ANOVA quando a variável resposta é quantitativa.

Este método é frequentemente utilizado como uma técnica exploratória e é uma alternativa à regressão linear múltipla e regressão logística, especialmente quando o conjunto de dados não é bem adequado à análise de regressão.

3.2.5.5 Algorítmos SAS

Algoritmos SAS incorporam e estendem a maioria das boas idéias discutidas para o particionamento recursivo. Tanto a variável *target* como as variáveis *input* podem ser nominais, ordinais ou intervalares. O usuário especifica o número máximo de galhos de uma divisão, permitindo assim a obtenção de árvores binárias, árvores espessas ou qualquer que se queira. As quebras podem ser avaliadas como uma redução na impureza (Mínimos Quadrados, índice de Gini ou Entropia), ou como um teste de significância (Qui-Quadrado ou Teste F). Testes de significância permitem ajustes de Bonferroni, como foi feito no CHAID. Valores

faltantes podem, opcionalmente, ser tratado como um valor especial, como no CHAID. Regras *surrogate*, se adequado, atribuiem os casos com valores faltantes a um ramo, como nos algoritmos de Breiman et al. (1984).

Há muitas opções de controle sobre a poda da árvore. Como no CHAID, um limite para o nível de significância pode parar o crescimento das árvores. O usuário tem opções na especificação de uma medida de avaliação. Por exemplo, incluir custos da má classificação.

Os algoritmos de árvore de decisão estão incluídos no SAS Enterprise Miner, que fornece um ambiente de programação visual para modelagem preditiva. Probabilidades a priori, os custos de má classificação, por exemplo, se aplicam a todas as ferramentas de modelagem. A árvore pode incorporar probabilidades antes para o critério de divisão ou apenas usá-los para ajustar as probabilidades posteriores. A árvore pode criar uma variável indicadora para cada folha. Estas variáveis automaticamente entram em outros modelos, tais como modelos de regressão, colocando o nó de interesse após o nó da árvore.

3.3 Rede Neural

Redes Neurais Artificiais (RNA), também conhecida como conexionismo ou sistema de processamento paralelo e distribuído tiveram seu ressurgimento no final da década de 1980, alguns anos após sua primeira aparição em 1943. Essa forma de computação não-algorítmica é caracterizada por sistemas que, em algum nível, relembram a estrutura do cérebro humano. Por não ser baseada em regras, a computação neural se constitui em uma alternativa à computação algorítmica convencional. Grande parte da investigação em RNA foi inspirada e influenciada pelo sistema nervoso do ser humano. A RNA é vista como a aproximação mais promissora para a construção de verdadeiros sistemas inteligentes.

RNA são sistemas paralelos distribuídos compostos por unidades de processamento simples (neurônios artificiais) que calcula determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos essas conexões estão associadas a pesos,

os quais armazenam o conhecimento adquirido pelo modelo e servem para ponderar a entrada recebida por cada neurônio da rede.

Em RNAs o procedimento usual na solução de problema passa inicialmente por uma fase de aprendizagem, em que um conjunto de exemplos é apresentado para a rede, que extrai as características necessárias para representar a informação fornecida. Essas características são utilizadas posteriromente para gerar respostas para o problema.

Sem dúvida, o fato mais atrativo em uma RNA é a capacidade de aprender por meio de exemplos e de generalizar a informação aprendida com o objetivo de encontrar a resposta adequada. Atualmente, os modelos neurais tem tido inúmeras aplicações nas mais diversas áreas, desde as telecomunicações ao mercado imobiliário, das despesas militares ao turismo (SHACHMUROVE, 2002; LAW; PINE, 2004), das relações internacionais (BECK; KING; ZENG, 2000) às questões de política interna (EISINGA; FRANSES; DIJK, 1998). Na área financeira, vários problemas tem sido abordados recorrendo às redes neurais, como a análise de risco de crédito (NEVES; VIEIRA, 2004), a modelagem da inflação (MCNELIS, 2005) e taxas de câmbio (ZHANG et al., 2002), o cálculo do *rating*, a previsão da volatilidade das opções (MCNELIS, 2005) e a previsão da rentabilidade de ações (THAWORNWONG; ENKE, 2004).

3.3.1 O cérebro humano

O cérebro humano é responsável por funções cognitivas básicas, assim como pela execução de funções sensoriomotoras autônomas. Além disso, sua rede de neurônios tem a capacidade de reconhecer padrões e relacioná-los, usar e armazenar conhecimenos por experiência, além de interpretar observações.

Apesar dos grandes avanços científicos, o conhecimento do modo como o cérebro humano funciona está longe de estar completo. No entanto, o comportamento individual dos neurônios biológicos é bem entendido do ponto de vista funcional e é exatamente nesse comportamento conhecido que se baseiam as RNAs.

3.3.2 Os Neurônios

O cérebro humano contém em torno de 10^{11} neurônios, sua célula fundamental. O neurônio é uma celula do sistema nervoso responsável pela condução do impulso nervoso. Cada um desses neurônios processa e se comunica com milhares de outros continuamente e em paralelo. A estrutura individual desses neurônios, a topologia de suas conexões e o comportamento conjunto desses elementos de processamento naturais formam a base para o estudo das RNAs.

Segundo Damásio (1996) os neurônios biológicos são divididos em três seções: um corpo celular; uma fibra principal de saída, o axônio; e fibras de entrada, os dentritos. Cada qual com suas funções específicas, porém complementares.

O corpo celular mede apenas alguns milésimos de milímetros, e os dentritos aprensentam poucos milímetros de comprimento. O axônio, contudo, pode ser mais longo e em geral, tem calibre uniforme. Os dentritos tem por função receber as informações, ou impulsos nervosos, oriundos de outros neurônios e conduzí-las até o corpo celular. Neste, a informação é processada e novos impulsos são gerados. Esses impulsos são transmitidos a outros neurônios, passando através do axônio até os dentritos dos neurônios seguintes. O ponto de contato entre a terminação axônia de um neurônio e o dentrito do outro é chamado de sinapse. São pelas sinapses que os neurônios se unem funcionalmente, formando as redes neurais biológicas. As sinapses funcionam como válvulas e são capazes de controlar a transmissão de impulsos (o fluxo da informação) entre os neurônios na rede neural.

Segundo Kohonen (2001) a ligação entre os axônios possuem um comprimento tal no seu conjunto que se fossem esticados daria para fazer duas viagens de ida e volta da Terra à Lua.

3.3.3 A comunicação entre os Neurônios

Uma rede neural consiste num conjunto de unidades de processamento simples (neurônios) que se comunicam entre si enviando sinais através de um número elevado de conexões. Em termos biológicos, se a informação acumulada no corpo celular de um determinado neurônio atingir certo limite, o neurônio “dispara”, transmitindo um sinal eletroquímico ao neurônio adjacente a ele, através de um

canal emissor, o axônio. A extremidade do axônio é composta por ramificações (as sinapses) que por sua vez estão ligadas à estrutura do neurônio receptor através de outras ramificações, os dentritos. Na Figura 8 pode-se ver o diagrama de um neurônio.

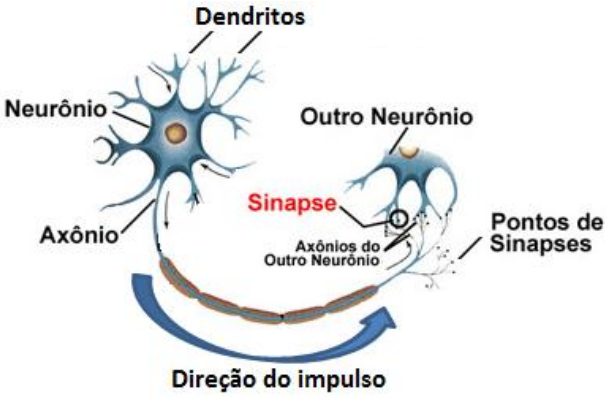

Figura 8 - Diagrama de um neurônio

Um único neurônio pode estar ligado a centenas ou mesmo a dezenas de milhares de neurônios. Num cérebro existem estruturas anatômicas de pequena, média e alta complexidade com diferentes funções, sendo possíveis parcerias.

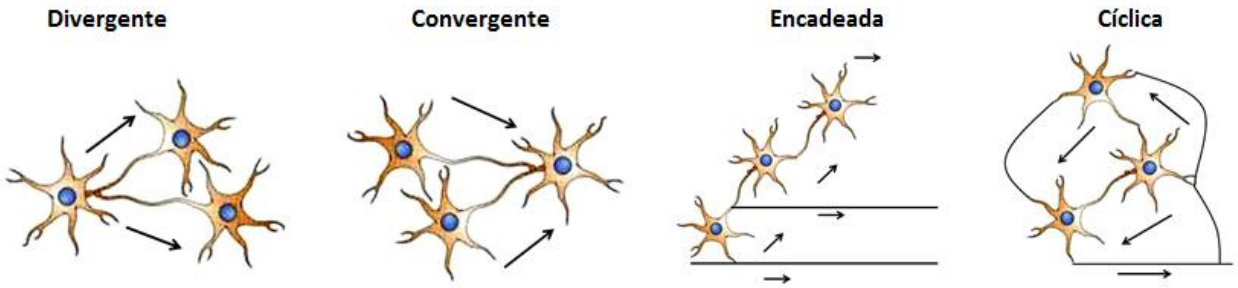

Figura 9 - Os diferentes tipos de conexões

Cortez e Neves (2000) comentam que os neurônios tendem a agrupar-se em camadas, existindo três principais tipos de conexões: divergente, em que o neurônio pode ser ligado a vários neurônios via uma arborização do axônio; convergentes, em que vários neurônios podem ser conectados a um único neurônio; e encadeadas ou cíclicas, as quais podem envolver vários neurônios e formarem cliclos (Figura 9).

3.3.4 O modelo MCP (McCulloch e Pitts)

O primeiro modelo artificial de um neurônio biológico foi fruto do trabalho pioneiro de Warren McCulloch e Walter Pitts, em 1943. McCulloch, psicólogo e neurofisiologista, dedicou sua carreira à tentativa de representar e modelar eventos no sistema nervoso. Pitts, um matemático recém formado, juntou-se a ele em 1942. No trabalho publicado em 1943, “A Logical Calculus of the Ideas Immament in Nervous Activity”, são apresentadas uma discussão sofisticada de redes lógicas de neurônios artificiais (chamados de neurônio MCP devido a McCulloch e Pitts), além de novas idéias sobre máquina de estados finitos, elementos de decisão limiar lineares e representações lógicas de várias formas de comportamento e memória.

O modelo de neurônio artificial proposto por McCulloch e Pitts é uma simplificação do que se sabia na época a respeito do neurônio biológico. Um neurônio biológico pode ser visualizado do ponto de vista funcional: as suas múltiplas entradas recebem ativações excitatórias ou inibitórias dos neurônios anteriores e, caso essa soma das excitações e inibições ultrapasse um determinado limite, o neurônio emite um impulso nervoso. Foi com base nesse comportamento funcional que o modelo MCP foi proposto na década de 1940.

Os neurônios (ou nós) transportam informação entrada (*input*) e passam a outros neurônios através das suas conexões de saída (*output*). Nas redes neurais artificiais estas conexões são designadas por pesos ou ponderações (*weights*). A informação “elétrica” é simulada com valores numéricos específicos armazenados nestes pesos.

A descrição matemática do modelo MCP resultou um modelo com n terminais de entrada (dentritos) que recebem os valores $x_1, x_2, \dots, x_n$ (que representam as ativações dos neurônios anteriores) e apenas um terminal de saída y (representando o axônio). Para representar o comportamento das sinapses, os terminais de entrada dos neurônios tem pesos acoplados $w_1, w_2, \dots, w_n$ cujos valores podem ser positivos ou negativos, dependendo das sinapses correspondentes serem inibitórias ou excitatórias. O efeito de uma sinapse particular i no neurônio pós-sináptico é definido por $w_i x_i$. Os pesos determinam “em que grau” o neurônio deve considerar sinais de disparo que ocorrem naquela conexão.

Como descrito na Figura 10, a informação é enviada para o neurônio com base nos pesos de recepção da camada de entrada (*input*). Este *input* é processado por uma função de combinação que soma os valores $w_i x_i$ recebidos pelo neurônio (soma ponderada). O valor resultante é comparado com um determinado valor limiar (*threshold*) pelas funções de ativação do neurônio. Se a soma obtida excede ao valor limiar, o neurônio será ativado e enviará um *output* pelos seus pesos de envio para todos os neurônios a ele conectados e assim sucessivamente, caso contrário o neurônio será inibido.

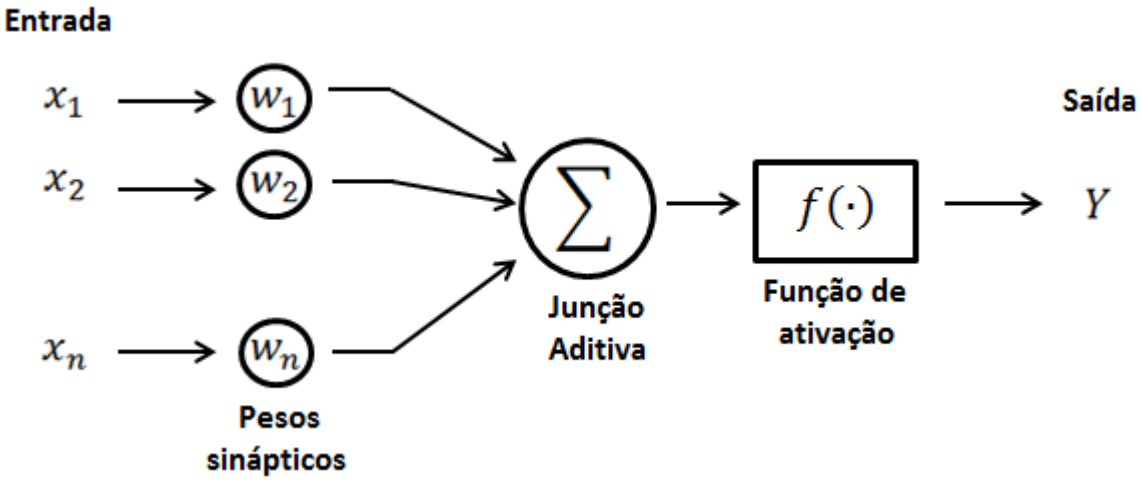

Figura 10 - Neurônio de McCulloch e Pitts, no qual ∑ representa a soma ponderada das entradas e $f(\cdot)$ a função de ativação

No modelo MCP, a ativação do neurônio é obtida por meio da aplicação de uma "função de ativação", que ativa ou não a saída, dependendo do valor da soma ponderada de suas entradas.

3.3.5 Funções de Ativação

A função de ativação é responsável por gerar a saída y do neurônio a partir dos valores dos vetores de peso $\underline{\boldsymbol{w}} = (w_1, w_2, \dots, w_n)^T$ e de entrada $\underline{\boldsymbol{x}} = (x_1, x_2, \dots, x_n)^T$. A função de ativação de um neurônio MCP é definida por

$$f(u) = \begin{cases} 1, \sum_{i=1}^{n} x_i w_i \geq \theta \\ 0, \sum_{i=1}^{n} x_i w_i < \theta \end{cases}$$

e é do tipo degrau deslocada do limiar de ativação θ em relação à origem, ou seja, a saída y será 1 para $\sum_{i=1}^{n} x_i w_i \geq \theta$ e 0 para $\sum_{i=1}^{n} x_i w_i < \theta$.

Existem diversas funções de ativação, entre elas a função degrau (Figura 11), exemplificada para $\theta = 3$. Uma aproximação contínua da função degrau é conhecida como função de ativação sigmoidal (Figura 12) definida por:

$$f(u) = \frac{1}{1 + e^{-\beta u}}$$

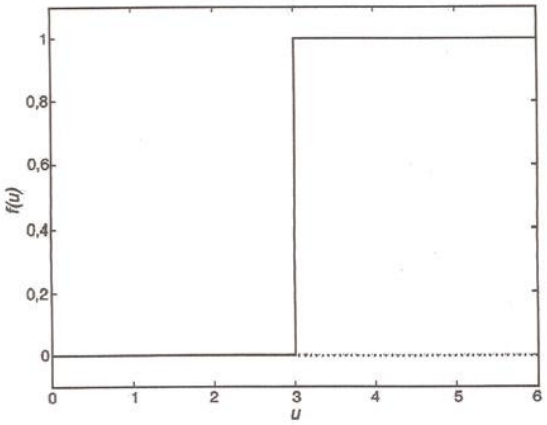

Figura 11 - Função de ativação degrau

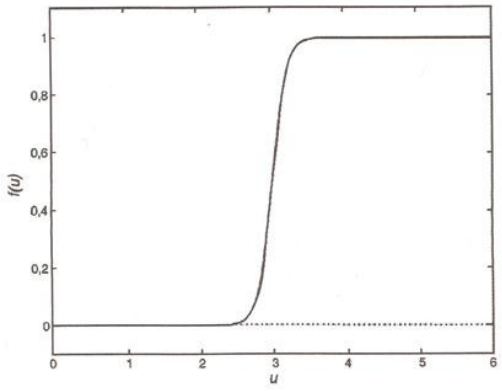

Figura 12 - Função de ativação sigmoidal

Essa função, além de ser diferenciável, possui uma região semi linear que pode ser impotante na aproximação de funções contínuas. Dependendo do tipo de problema a ser abordado, neurônios com função de ativação linear (Figura 13) podem ser utilizados como:

$$f(u) = u$$

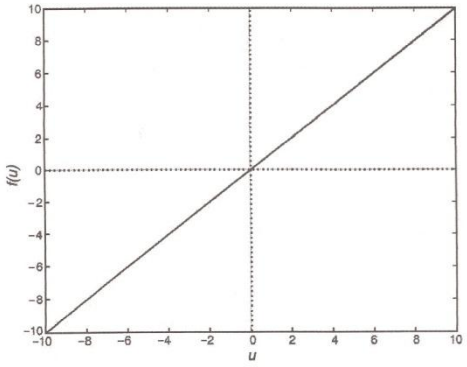

Figura 13 - Função de ativação linear

Já as RNAs do tipo Radial Basis Functions (RBF) utilizam neurônios com funções de ativação radiais, como a gaussiana (Figura 14) definida por:

$$f(u) = e^{-\frac{(u-\mu)^2}{r^2}}$$

em que μ é o centro (ponto médio) e r é o raio de abertura da função.

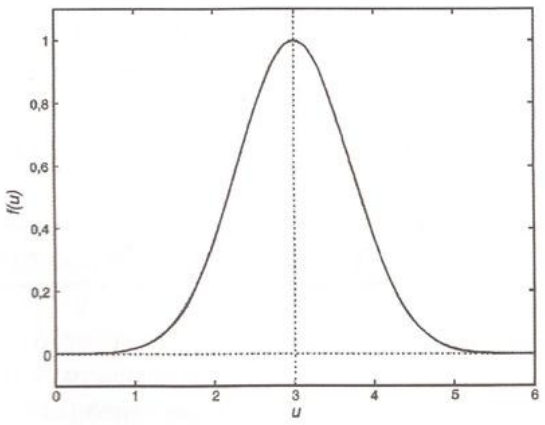

Figura 14 - Função de ativação gaussiana

3.3.6 Principais arquiteturas de RNAs

As redes neurais artificiais diferenciam-s pela sua arquitetura e pela forma como os pesos associados às conexões são ajustados durante o processo de aprendizagem. A arquitetura de uma rede neural restringe o tipo de problema no qual a rede poderá ser utilizada, e é definida pelo número de camadas (camada

única ou múltiplas camadas), pelo número de nós em cada camada, pelo tipo de conexões entre os nós e pela sua topologia (HAYKIN, 1999).

Independentemente da função de ativação escolhida, neurônios individuais possuem capacidade computacional limitada. No entanto, um conjunto de neurônios artificiais conectados na forma de uma rede neural é capaz de resolver problemas de complexidade elevada. As figuras a seguir mostram algumas configurações possíveis de neurônios artificiais conectados na forma de redes neurais artificiais.

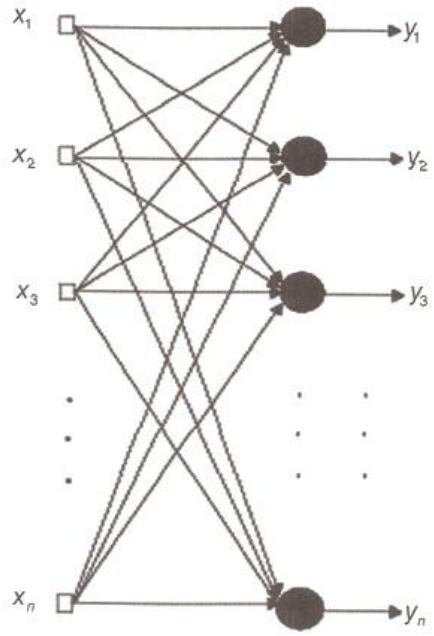

Figura 15 - Rede *feedforward* de uma única camada

A estrutura mais simples é apresentada nas Figuras 15 e 16 que correspondem a redes neurais alimentadas para frente (*feedforward*). Uma RNA *feedforward* pode ser organizada por camadas, porque não existem ciclos, dado que as conexões são sempre unidirecionais (convergentes ou divergentes) não existindo realimentação. Na sua forma mais simples (Figura 15), uma rede é composta por uma camada de entrada, cujos valores de saída são fixados externamente e por uma camada de saída.

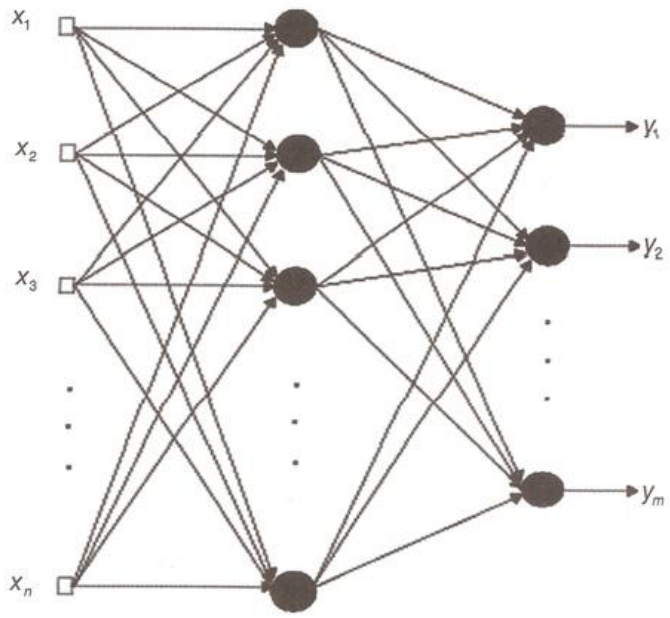

Figura 16 - Rede *feedforward* de duas camadas

É importante ressaltar, que a camada de entrada não é contabilizada como camada num RNA, dado o fato de nesta não se efetuarem qualquer forma de cálculo. A segunda classe de redes *feedforward* distingue-se pelo fato de possuir uma ou mais camadas intermediárias, cujos nós são designados por nós intermediários tendo como função, intervir de forma útil entre a entrada e a saída da rede (Figura 16). Ao se acrescentar camadas intermediárias, aumenta-se a capacidade da rede em modelar funções de maior complexidade, uma particularidade bastante útil, quando o número de nós na camada de entrada é elevado. Por outro lado, este aumento nas camadas intermediárias pode vir a atrapalhar no tempo de aprendizagem, visto que este tempo aumenta de forma exponencial.

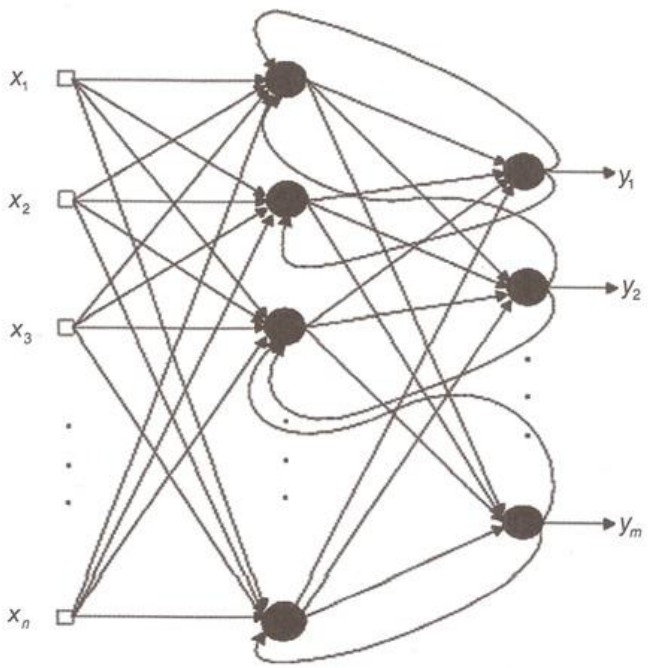

Figura 17 - Rede com recorrência entre saídas e camada intermediária

As RNAs apresentadas nas Figuras 15 e 16 são consideradas estáticas, já que não possuem recorrência em sua estrutura: as suas saídas em um determinado instante dependem apenas das entradas atuais. Já as estruturas das Figuras 17 e 18 possuem conexões recorrentes entre neurônios de um mesmo nível ou entre neurônios de saída e de camadas anteriores. Na Figura 17, a saída depende não somente das entradas, mas também do seu valor atual. Essa estrutura de RNA é utilizada na resolução de problemas que envolvam processamento temporal, como em previsão de eventos futuros. Já a estrutura da Figura 18 possui um único nível de neurônios, em que a saída de cada um deles está conectada às entradas de todos os outros. A rede não possui entradas externas e sua operação se dá em função da dinâmica de mudança de estados dos neurônios, que operam de forma auto-associativa.

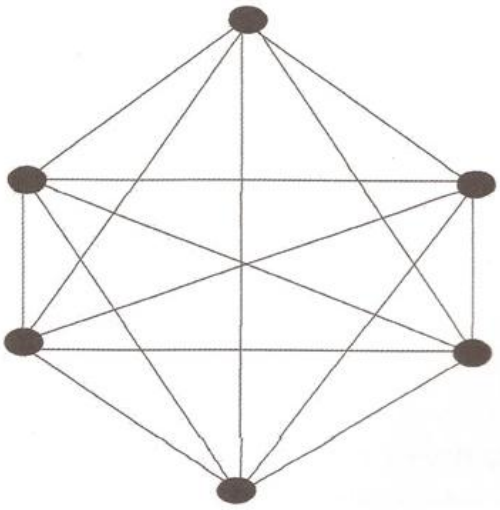

Figura 18 - Rede com recorrência auto-associativa

3.3.7 Aprendizado

Como já mencionado, uma das propriedades mais importantes de uma rede neural artificial é a capacidade de aprender a partir da interação com o meio ambiente e fazer inferências do que aprenderam.

A utilização de redes neurais, independente do problema, passa primeiramente pela fase de aprendizagem que ocorre quando a rede neural consegue extrair padrões de informação no subconjunto de treino, criando assim uma representação própria. Segundo Braga, Carvalho e Ludemir (2000), a etapa de aprendizagem consiste num processo interativo de ajuste dos parâmentros da rede, os pesos das conexões entre as unidades de processamento, que guardam, ao final do processo, o conhecimento que a rede adquiriu do ambiente em que se encontra a operar.

Para Haykin (1999), a aprendizagem é um processo pela qual os parâmetros de uma rede neural são ajustados por meio de um processo de estímulo do meio ambiente no qual a rede está inserida, sendo o tipo de aprendizagem determinado pela maneira como ocorrem os ajustamentos nos parâmetros. Sendo assim, o objetivo do treino/aprendizagem consiste em atribuir valores apropriados aos pesos sinápticos de modo a produzir o conjunto de saídas desejadas ou ao menos consistentes com um intervalo de erro estabelecido. Desta forma, o processo

de aprendizagem consiste na busca de um espaço de pesos pela aplicação de alguma regra que defina esta aprendizagem.

É importante ressaltar que o conceito de aprendizado está relacionado com a melhoria do desempenho da rede segundo algum critério pré-estabelecido. O erro quadrático médio da resposta de rede em relação ao conjunto de dados fornecido pelo ambiente, por exemplo, é utilizado como critério de desempenho dos algoritmos de correção dos erros. Assim, quando estes algoritmos são utilizados no treinamento de RNAs, espera-se que o erro diminua à medida que o aprendizado prossiga.

De uma forma genérica, o valor do vetor de pesos $w(t+1)$ no instante $t+1$ pode ser escrito como:

$$w(t+1) = w(t) + \Delta w(t)$$

em que $w(t)$ e $w(t+1)$ representam os valores dos pesos nos instantes t e $t+1$, respectivamente, e $\Delta w(t)$ é o ajuste aplicado aos pesos.

Os algoritmos de aprendizado diferem, basicamente, na forma como $\Delta w(t)$ é calculado. Há vários algoritmos diferentes para treinamento de redes neurais, podendo os mesmos ser agrupados em dois paradigmas principais: aprendizado supervisionado e aprendizado não-supervisionado.

3.3.7.1 Aprendizado supervisionado

Aprendizado supervisionado implica a existência de um supervisor, ou professor externo, o qual é responsável por estimular as entradas da rede por meio de padrões de entrada e observar a saída calculada pela mesma, comparando-a com a saída desejada. Como a resposta da rede é função dos valores atuais do conjunto de pesos, estes são ajustados de forma a aproximar a saída da rede da saída desejada. A Figura 19 ilustra uma representação esquemática do aprendizado supervisionado. Para cada padrão de entrada, a rede tem sua saída corrente comparada com a saída desejada pelo supervisor, que fornece informações sobre a direção de ajustes dos pesos.

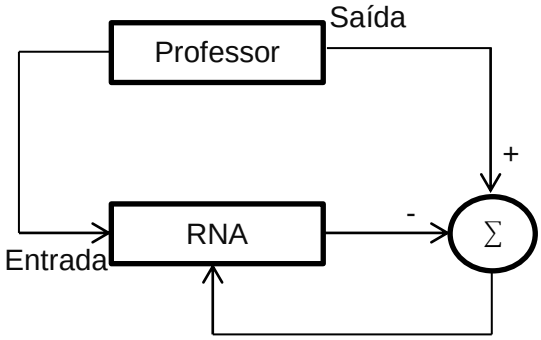

Figura 19 - Aprendizado supervisionado

Este “professor” pode ser um humano, que especifica a classe correta para cada padrão de entrada, ou um sistema físico cujo comportamento se pretende modelar. A cada interação efetuada a rede neural compara a resposta desejada com o valor de saída da rede, originando um erro. O erro resultante é utilizado para ajustar os pesos da rede. A soma dos erros quadráticos de todas as saídas é normalmente utilizada como medida de desempenho da rede. Uma das vantagens da aprendizagem supervisionada é a de que o seu modelo é bem definido, apontando-se como principais críticas e artificialismo, a limitação do modelo de aprendizagem e a necessidade de professor (REED; MARKS ll, 1999).

O aprendizado supervisionado pode ser implementado basicamente de duas formas: off-line ou on-line. Para treinamento off-line, os dados do conjunto de treinamento não mudam, e uma vez obtida uma solução para a rede, esta deve permanecer fixa. Caso novos dados sejam adicionados, um novo treinamento, envolvendo também os dados anteriores, deve ser realizado para se evitar interferência no treinamento anterior. Por sua vez, no aprendizado on-line o conjunto de dados muda continuamente e a rede deve estar em um contínuo processo de adaptação.

3.3.7.2 Correção de erros

O caso mais comum de aprendizado supervisionado é o aprendizado por correção de erros, em que se procura minimizar o erro da resposta atual da rede em relação à saída desejada. A expressão genérica para o erro $e(t)$ no instante de tempo t pode ser escrita como:

$$e(t) = \gamma_d(t) - \gamma(t)$$

em que $\gamma_d(t)$ é a saída desejada e $\gamma(t)$ é a resposta atual calculada pela rede. A forma genérica para atualização dos pesos por correção dos erros é definida por:

$$w_i(t+1) = w_i(t) + \eta e(t) x_i(t)$$

em que $w_i(t)$ corresponde ao peso de entrada i, η é a taxa de aprendizado, $e(t)$ é uma medida de erro e $x_i(t)$ a entrada i do neurônio.

A obtenção das equações de ajuste envolve a minimização da soma dos erros quadráticos das saídas, como:

$$\varepsilon^2 = 1/2 \sum_{i=1}^{p} \left(\gamma_d^i - \gamma\right)^2$$

em que p é o número de exemplos de treinamento, γ_d^i é a saída desejada para o vetor de entrada x_i e γ é a saída corrente da rede para o vetor x_i.

Portanto o conjunto de dados formado pelos pares de entradas e saídas (x_i, γ_d^i) define a superfície de erro. Para cada valor possível de w, a soma dos erros quadráticos do conjunto de dados é calculada, e um vetor ε^2 é obtido. A superfície formada por todos os valores de ε^2 resulta na superfície de erro para o conjunto de dados. O valor de w que minimiza ε^2 correponde à solução de erro mínimo, ou mínimo global, para o conjunto de dados atual. Dependendo do tipo de unidade de processamento utilizado para construir a rede, a superfície de erro pode assumir formas diferentes:

1. No caso da rede ser formada inteiramente por unidades de processamento lineares, a superfície de erro é definida por uma função quadrática dos pesos da rede, podendo a mesma possuir um único mínimo.
2. Para o caso da rede ser formada por unidades de processamento não-lineares, a superfície de erro poderá ter uma forma irregular e vários mínimos locais, além do mínimo global.

Em ambas as situações, o objetivo do aprendizado por correção de erros é, a partir de um ponto arbitrário da superfície de erro, mover-se na direção do mínimo global. Na primeira situação só existe um mínimo global, já que se trata de uma superfície de erro quadrática, que pode ser facilmente atingido. Na segunda

situação, nem sempre o mínimo global é alcançado, já que as saídas não-lineares geram superfícies de erros irregulares, podendo levar a rede a se estabilizar em um mínimo local indesejado. Apesar disso, existem técnicas de treinamento que levam a rede a se aproximar do mínimo global. Não obstante, nem sempre o mínimo global corresponde à solução com a melhor resposta da rede para dados não pertencentes ao conjunto de treinamento.

3.3.7.3 Aprendizado por reforço

O aprendizado por reforço se caracteriza por um processo de tentativa e erro que visa a maximizar o índice de desempenho escalar chamado de sinal de reforço. Enquano no aprendizado supervisionado o supervisor externo fornece informações para a atualização dos pesos baseado em um critério de desempenho como o erro, no aprendizador por reforço o crítico externo procura maximizar o reforço das ações boas executadas pela rede.

Na Figura 20 essa idéia fica exposta claramente, podendo-se observar que a função do crítico é semelhante a do supervisor (professor) no aprendizado supervisionado. Segundo Sutton, o aprendizado por reforço ocorre quando uma ação tomada pelo sistema de aprendizado é seguida de estados satisfatórios, então a tendência do sistema de produzir essa ação particular é reforçada. Se não for seguida de estado satisfatório, a tendência do sistema de produzir essa ação é enfraquecida.

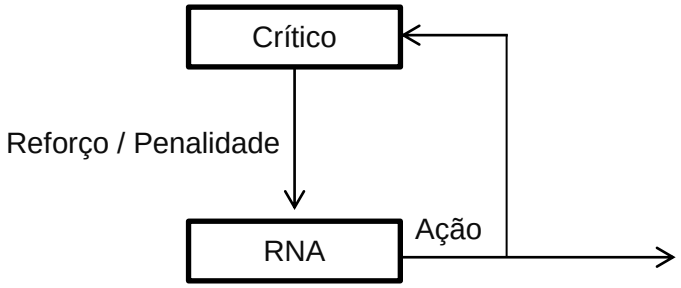

Figura 20 - Aprendizado por reforço

O aprendizado por reforço se aplica principalmente a problemas de aprendizado envolvendo tarefas de controle nas quais é permitdo à rede errar durante o processo de interação com o sistema a ser controlado.

3.3.7.4 Aprendizado não supervisionado

Um dos incovenientes do treino supervisionado é a necessidade de “professor”. Dado que não se sabe a priori o número nem as classes envolvidas, surge-se a necessidade de uma aprendizagem e classificação não supervisionada. Neste esquema de treinamento somente os padrões de entrada estão disponíveis para a rede, ao contrário do aprendizado supervisionado, cujo conjunto de treinamento possui pares de entrada e saída. Durante o processo de aprendizado os padrões de entrada são apresentados continuamente à rede e a existência de regularidades nesses dados faz com que o aprendizado seja possível. Regularidade e redundância nas entradas são características essenciais para haver aprendizado não-supervisionado.

Se uma rede tem a habilidade de descobrir cluster com similaridade de padrões sem supervisão, isto é, sem possuir informação sobre a variável *target*, por qualquer que seja o processo utilizado, diz-se que a rede, além de não ser supervisionada, possui capacidade de auto-organização (GURNEY, 1997). Neste tipo de aprendizado não existe a figura do supervisor externo, sendo o ajuste dos pesos feito independentemente de qualquer critério de desempenho da resposta da rede, por meio de um mecanismo local às sinapses.

3.3.8 Perceptron

O modelo perceptron de uma única camada, ou perceptron simples, proposto por Rosenblatt (ROSENBLATT, 1962) era composto por uma estrutura de rede, tendo como unidades básicas neurônios MCP, e por uma regra de aprendizado. Alguns anos mais tarde, Rosenblatt demonstrou o teorema de convergência do perceptron, que mostra que o neurônio MCP treinado com o algoritmo de aprendizado do perceptron sempre converge caso o problema em questão seja linearmente separável (ROSENBLATT, 1962).

A topologia original descrita por Resenblatt era composta por unidades de entrada (retina), por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pelas unidades de resposta. Embora essa topologia original possua três níveis, ela é conhecida como perceptron de uma

única camada, já que somente o nível de saída (unidades de resposta) apresenta propriedades adaptativas.

3.3.8.1 O algorítmo de aprendizado do Perceptron

Uma RNA é composta por um conjunto de neurônios com capacidade de processamento local, uma topologia de conexão que define a forma como estes neurônios estão conectados e uma regra de aprendizado.

Durante o processo de aprendizado o que se deseja obter no instante n é o valor do incremento $\Delta \boldsymbol{w}(n)$ a ser aplicado ao vetor de pesos $\boldsymbol{w}(n)$ de tal forma que o seu valor atualizado $\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \Delta \boldsymbol{w}(n)$ esteja mais próximo da solução desejada do que $\boldsymbol{w}(n)$. Sendo assim, os algoritmos de aprendizado de RNA visam o desenvolvimento de técnicas para a obtenção do valor de $\Delta \boldsymbol{w}(n)$ mais apropriado para a obtenção da solução do problema.

Considerando um neurônio arbitrário da camada de resposta de um perceptron e seus vetores de entrada $\boldsymbol{x}'$ e de pesos $\boldsymbol{w}'$, sua ativação é definida por $\sum_i w'_i x'_i = \boldsymbol{w}'.\boldsymbol{x}'$, em que $\boldsymbol{w}'.\boldsymbol{x}'$ representa o produto interno entre $\boldsymbol{w}'$ e $\boldsymbol{x}'$. Consequentemente, a condição crítica de disparo do neurônio é $\boldsymbol{w}'.\boldsymbol{x}' = \theta$ ou $\boldsymbol{w}'.\boldsymbol{x}' - \theta = 0$, o que é equivalente a se adicionar um peso w_0 com o valor $-\theta$ às entradas do neurônio e conectá-lo a uma entrada com valor fixo $x_0 = 1$. A nova condição crítica de disparo para os vetores aumentados passa então a ser $\boldsymbol{w}.\boldsymbol{x} = 0$, em que $\boldsymbol{w} = \{-0, w_1, w_2, w_3, \dots, w_n\}^T$ e $\boldsymbol{x} = \{1, x_1, x_2, x_3, \dots, x_n\}^T$.

Considere agora o par de treinamento $\{x, \gamma_d\}$ para um neurônio arbitrário da rede em que $\boldsymbol{x}$ é o seu vetor de entrada e γ_d a saída desejada para um neurônio arbitrário da rede, rede em resposta ao vetor de entrada $\boldsymbol{x}$ será chamada simplesmente de γ, podendo-se então definir o erro devido à saída atual como sendo $e = \gamma_d - \gamma$. Para o caso do percepetron, tem-se sempre que $\gamma \in \{0,1\}$ e $\gamma_d \in \{0,1\}$, podendo, portando haver apenas duas situações possíveis para as quais o erro de saída é diferente de 0, conforme mostrado na Tabela a seguir.

Tabela 4 - Possíveis situações para o erro

γ_d (saída desejada)	γ (saída atual)	e (erro)
0	0	0
1	0	1
0	1	-1
1	1	0

Para duas situações possíveis ($\gamma_d = 1$ e $\gamma = 0$ ou $\gamma_d = 0$ e $\gamma = 1$), chegou-se à mesma expressão para a regra de atualização dos pesos, que pode então ser escrita como a equação geral para a atualização dos pesos de um neurônio de um perceptron simples: $\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \eta e \boldsymbol{x}(n)$, em que a constante η é uma medida de rapidez com que o vetor de pesos será atualizado, sendo comumente chamada de taxa de aprendizado. De acordo com o Teorema da Convergência (ROSENBLATT, 1958), a atualização dos pesos leva sempre a uma solução caso as classes em questão sejam linearmente separáveis.

3.3.8.2 Implementação do algorítmo de aprendizado do Perceptron

O algoritmo de aprendizado do perceptron sempre chega, em um tempo finito, a uma solução para o problema de separação de duas classes linearmente separáveis (ROSENBLATT, 1958). De maneira geral, o algoritmo de aprendizado de um perceptron pode ser descrito como:

1. Inicialize η;
2. Inicialize o vetor de pesos $\boldsymbol{w}$ com valores aleatórios;
3. Aplique a regra de atualização dos pesos $\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \eta e \boldsymbol{x}(n)$ para todos os p pares (x^i, γ_d^i) do conjunto de treinamento $\Gamma = \{(x^i, \gamma_d^i)\}_{i=1}^{p}$;
4. Repita o passo anterior até que $e = 0$ para todos os p elementos de Γ.

3.3.8.3 Considerações sobre o aprendizado do Perceptron

Sabe-se que independentemente do valor de η, haverá convergência em um tempo finito, caso as classe sejam linearmente separáveis; no entanto, esse tempo pode ser proibitivo em situações reais. Um valor muito pequeno de η pode levar a um tempo de convergência muito alto, equanto um valor muito alto pode levar a instabilidade no treinamento. O melhor ajuste para o valor de η dependerá do problema, não havendo uma recomendação geral para todos os casos.

Uma outra consideração é com relação aos valores iniciais atribuídos aos elementos do vetor de pesos. Uma regra geral é iniciá-los com valores amostrados em uma ditribuição uniforme definida no intervalo $[-a, a]$, em que a é um valor positivo próximo de zero, como 0,5, por exemplo. A recomendação de se iniciar os pesos com valores pequenos, próximos a zero, faz-se necessária para evitar saturação forte do neurônio MCP, o que resultaria em dificuldades para convergência do algoritmo. Valores iniciais grandes para os pesos resultariam em um valor igualmente grande para a soma ponderada das entradas, o que levaria a uma resposta da função de ativação muito distante do limiar, resultando na necessidade de muitos passos de treinamento para alterar o estado de saída do neurônio.

3.3.9 Redes Perceptron de Múltiplas Camadas (MLP)

As redes de uma única camada têm a limitação de resolver apenas problemas com características lineares. Sabe-se, no entanto, que as não-linearidades são inerentes à maioria as situações e problemas reais, sendo necessárias, portanto, a utilização de estruturas com características não-lineares para a resolução de problemas de maior complexidade.

As não-linearidades são incorporadas a modelos neurais por meio das funções de ativação (não-lineares) de cada neurônio da rede e da composição da sua estrutura em camadas sucessivas. Assim, a reposta da camada mais externa da rede corresponde à composição das respostas dos neurônios das camadas anteriores. À rede neural de múltiplas camadas compostas por neurônios com

funções de ativação sigmoidais nas camadas intermediárias dá-se o nome de Perceptron de Múltipas Camadas (MLPs – Multilayer Perceptron).

Os perceptrons de múltiplas camadas são uma importante classe de redes neurais artificiais, eles consistem em um conjunto de unidades sensoriais, que constituem a camada de entrada; as camadas ocultas e as de saída, formadas por nós computacionais. Um perceptron de múltiplas camadas tem três características distintas:

a) O modelo de cada neurônio da rede inclui uma função não-linear chamada função de ativação. É importante ressaltar que essa não-linearidade deve ser suave, isto é, diferenciável em qualquer ponto. Uma forma que é normalmente utilizada e que satisfaz essas exigências é uma não-linearidade sigmoidal (como função de ativação descrita anteriormente).
b) A rede contém uma ou mais camadas intermediárias, ou ocultas, que não são parte da entrada nem da saída da mesma. Os neurônios ocultos capacitam a rede a aprender tarefas complexas extraindo progressivamente as característcas mais sinificativas dos padrões (vetores) de entrada.
c) A rede possui um alto grau de conectividade, determinado pelas sinápses da rede.

É por meio da combinação destas características, em conjunto com a habilidade de aprender da experiência por treinamento, que o perceptron de múltiplas camadas deriva seu poder computacional.

O treinamento de redes de uma única camada por meio de aprendizado supervisionado e correção de erros é realizado por meio da aplicação do ajuste $\Delta \boldsymbol{w} = \eta e \boldsymbol{x}$ ao vetor de pesos $\boldsymbol{w}$. Para redes de uma única camada, o erro e é obtido diretamente por meio da diferença entre a saída desejada e saída corrente da rede. No entanto, para redes de múltiplas camadas esse procedimento pode ser aplicado somente para a camada de saída, já que não existem saídas desejadas definidas para as camadas intermediárias. Assim, o problema passa a ser então como calcular ou estimar o erro das camadas intermediárias.

A solução para esse problema de treinamento de MLPs surgiu em meados da década de 1980 com a descrição do algoritmo de retropropagação de erros, ou *back-propagation*. O princípio do algoritmo é, utilizando-se o gradiente descendente, estimar o erro das camadas intermediárias por meio de uma estimativa de efeito que estas causam no erro da camada de saída. Assim, o erro de saída da rede é calculado e este é retroalimentado para as camadas intermediárias, possibilitando o ajuste dos pesos proporcionalmente aos valores das conexões entre camadas. A utilização do gradiente descendente requer o uso de funções de ativação contínuas e diferenciáveis, assim, funções de ativação do tipo degrau utilizadas no perceptron simples, por exemplo, não poderão ser utilizadas. Funções sigmoidais serão utilizadas para prover uma aproximação da função degrau.

O papel das múltiplas camadas em uma rede *feedforward*, como a rede MLP, é transformar, sucessivamente, o problema descrito pelo conjunto de dados no espaço de entrada em uma representação tratável para a camada de saída da rede. Por exemplo, um problema não-linearmente separável, resolvido por uma rede de duas camadas, é transformado em um problema linearmente separável pela camada intermediária, criando uma nova disposição interna à rede para os dados de entrada. A partir dessa nova disposição, linearmente separável, a camada de saída pode resolver o problema descrito no espaço de entrada.

3.3.9.1 A arquitetura de uma rede Perceptron de Múltiplas Camadas (MLP)

Redes MLP apresentam um poder computacional maior do que aquele apresentado pelas redes de uma única camada. Redes com duas camadas intermediárias podem implementar qualquer função, seja ela linearmente separável ou não (CYBENKO, 1989). A qualidade da aproximação obtida dependerá da complexidade da rede, ou seja, do número de neurônios utilizados nas camadas intermediárias. A Figura 16, mostrada anteriormente, apresenta uma rede MLP típica com uma camada intermediária.

O comportamento de uma rede MLP, como a da Figura 16, pode ser descrita por meio de duas transformações sucessivas, sendo uma delas $H(\boldsymbol{x}; \boldsymbol{w}_H)$, relativa à camada intermediária, e a outra $Y(H(\boldsymbol{x}; \boldsymbol{w}_H);\ \boldsymbol{w}_s)$, relativa à camada de

saída, em que $\boldsymbol{w}_H$ e $\boldsymbol{w}_s$ correspondem, respectivamente, aos vetores de pesos das camadas escondida e de saída.

3.3.9.2 Número de camadas

Para uma rede com pelo menos duas camadas intermediárias, pode-se dizer que o seguinte processamento occorre em cada uma das camadas:

1. *Primeira camada intermediária*: cada neurônio contribui com retas para a formação da superfície no espaço de entrada;
2. *Segunda camada intermediária*: cada neurônio combina as retas descritas pelos neurônios da camada anterior conectados a ele, formando regiões convexas, em que o número de lados é definido pelo número de unidades a ele conectadas.
3. *Camada de saídia*: cada neurônio forma regiões que são combinações das regiões convexas definidas pelos neurônios a ele conectadas da camada anterior. Os neurônios definem, dessa maneira, regiões com formatos diversos.

A idéia é que a rede responda de acordo com as características presentes nos dados de entrada e não exatamente igual aos dados de entrada. Por exemplo, o princípio de Ockham diz que deve-se preferir modelos simples a modelos complexos e esta preferência deverá aplicar-se até que os modelos se adequem aos dados. Igualmente, Chorão (2005) diz que apesar de várias práticas para determinar a dimensão da camada intermediária, na maioria dos casos continua ser a "tentantiva e erro" a melhor regra a seguir.

Uma rede MLP com uma camada intermediária é suficiente para aproximar qualquer função contínua e em problemas mais complexos pode-se utilizar duas camadas. Independentemente da complexidade do problema, duas camadas são suficientes para que a rede possa aproximar o problema. A utilização de um grande número de camadas escondidas não é recomendada. Cada vez que o erro médio durante o treinamento é utilizado para atualizar os pesos das sinápses da camada imediatamente anterior, ele se torna menos útil ou preciso. A única camada que tem uma noção precisa de erro cometido pela rede é a camda de saída. A última

camada escondida recebe uma estimativa sobre o erro. A penúltima camada escondida recebe uma estimativa da estimativa, e assim por diante.

3.3.9.3 Número de neurônios

Em relação ao número de neurônios nas camadas escondidas, este é geralmente definido empiricamente. Deve-se ter cuidado para não utilizar nem unidades demais, o que pode levar a rede a memorizar os dados de treino (*overfitting*), ao invés de extrair as caracaterísticas gerais que permitirão a generelização, nem um número muito pequeno, que pode forçar a rede a gastar tempo em excesso tentando encontrar uma representação ótima. Devido a estas dificuldades é recomendado manter o número de neurônios escondidos baixo, mas não tão baixo quanto o estritamente necessário. Existem várias propostas de como determinar a quantidade adequada de neurônios nas camadas escondidas de uma rede neural. São as mais utilizadas:

1. O número de neurônios deverá estar compreendido entre o número de variáveis de *input* e o número de *output* (BLUM, 1992).
2. O número de neurônios deverá ser menor que a metade do número de variáveis da primeira camada (SWINGLER, 1996).
3. O número de neurônios deverá ser igual ao número de dimensões (componentes principais) necessárias para explicar 70 a 90% da variabilidade dos dados de entrada (BOGER; GUTERMAN, 1997).

3.3.9.4 Treinamento de Redes MLP

O algoritmo de treinamento de redes MLP mais popular é o *back-propagation* que, por ser supervisionado, utiliza pares de entrada e saída para, por meio de um mecanismo de correção de erros, ajustar os pesos da rede. O treinamento ocorre em duas fases, em que cada fase percorre a rede em um sentido. Essas duas fases são chamadas de fase *forward* e fase *backward*. A fase *forward* é utilizada para definir a saída da rede para um dado padrão de entrada. A fase *backward* utiliza a saída desejada e a saída fornecida pela rede para atualizar os pesos de suas conexões.

Segundo Beale e Jackson (1990), a grande dificuldade do perceptron de múltiplas camadas consiste no cálculo dos pesos nas camadas intermediárias de uma forma eficiente e que minimize o erro na saída. Quanto mais camadas intermediárias existirem, mais difícil será o cálculo dos erros. O algoritmo *back-propagation* é um algoritmo em que a aprendizagem dá-se por meio de um processamento interativo dos exemplos de treino, comparando as previsões da rede para cada um dos exemplos de treino com os verdadeiros valores. A minimização do erro no algoritmo *back-propagation* é obtida pela execução do gradiente decrescente na superfície de erros do espaço de pesos, em que a altura para qualquer ponto no espaço de pesos correponde à medida de erro. Para cada exemplo de treino, os pesos são modificados de forma a minimizar o erro quadrático médio entre as previsões da rede e os verdadeiros resultados. Estas modificações são feitas no sentido contrário da camada de *output* para a camada de *input*. O erro é apurado na camada de *output* e "retro-propagado" para a camada de *input*, ou seja, uma vez apurado o erro segue-se um processo de "apuramento das responsabilidades" tentando corrigir os pesos que mais contribuíram para esse erro.

É possível identificar duas fases distintas no processo de aprendizagem do algoritmo em questão. A primeira fase é responsável pelo processo de treino e consiste em enviar um sinal funcional que vai da camada de *input* até a de *output*, isto é, processamento para frente, onde um vetor de entrada é fornecido aos neurônios de entrada, propagando-se para frente, camada a camada. Finalmente é produzido um conjunto de saída como resposta da rede. Durante a fase de propagação os pesos sinápticos da rede são todos fixos.

Na segunda fase do treino é enviado um sinal do erro, no sentido inverso, isto é, do *output* para a camada de *input* – denominado de retropropagação. Durante a fase de retropropagação, os pesos sinápticos são todos ajustados de acordo com uma regra de correção do erro. Especificamente esta fase apresenta a validação da fase anterior, ou seja, verifica-se se o *output* produzido foi satisfatório, por meio da comparação das saídas geradas pela rede com a resposta desejada para produzir um sinal de erro. Este sinal de erro é também retropropagado por meio da rede, em sentido contrário das conexões sinápticas – daí o nome de retropropagação do erro.

Para facilitar a compreensão do algoritmo, apresenta-se uma descrição resumida dos passos mais importantes do algoritmo. A fase *forward* (a primeira fase) envolve os seguintes passos:

1. O vetor de entrada $\boldsymbol{x}$ é apresentado às entradas da rede, e as saídas dos neurônios da primeira camada escondida C_1 são calculadas.
2. As saídas da camada escondida C_1 proverão as entradas da camada seguinte C_2. As saídas da camada C_2 são calculadas. O processo se repete até que se chegue à camada de saída C_k.
3. As saídas produzidas pelos neurônios da camada de saída são então comparadas às saídas desejadas γ_d para aquele vetor de entrada $\boldsymbol{x}$ e o erro correspondente $\gamma_d - \gamma$ é calculado.

Conforme pode ser visto nos passos descritos para a fase *forward*, o seu objetivo é obter o erro de saída após a propagação do sinal por todas as camadas da rede. A fase backward, por sua vez, envolve as etapas:

1. O erro da camada de saída C_k é utilizado para ajustar diretamente os seus pesos, utilizando-se para isso o gradiente descendente do erro.
2. Os erros dos neurônios da camada de saída C_k são propagados para a camada anterior C_{k-1}, utilizando-se para isso os pesos das conexões entre as camadas, que serão multiplicados pelos erros correspondentes. Assim, tem-se um valor de erro estimado para cada neurônio da camada escondida que representa uma medida de influência de cada neurônio na camada C_{k-1} no erro de saída da camada C_k.
3. Os erros calculados para o neurônio da camada C_{k-1} são então utilizados para ajustar os seus pesos pelo gradiente descendente, analogamente ao procedimento utilizado para a camada C_k.
4. O processo se repete até que os pesos da camada C_1 sejam ajustados, concluindo-se assim o ajuste dos pesos de toda a rede para o veto de entrada $\boldsymbol{x}$ e sua saída desejada γ_d.

A Figura 21 mostra um esquema de rede MLP com duas camadas. Nesta figura pode-se entender melhor o raciocínio do back-propagation, junto com as deduções a seguir.

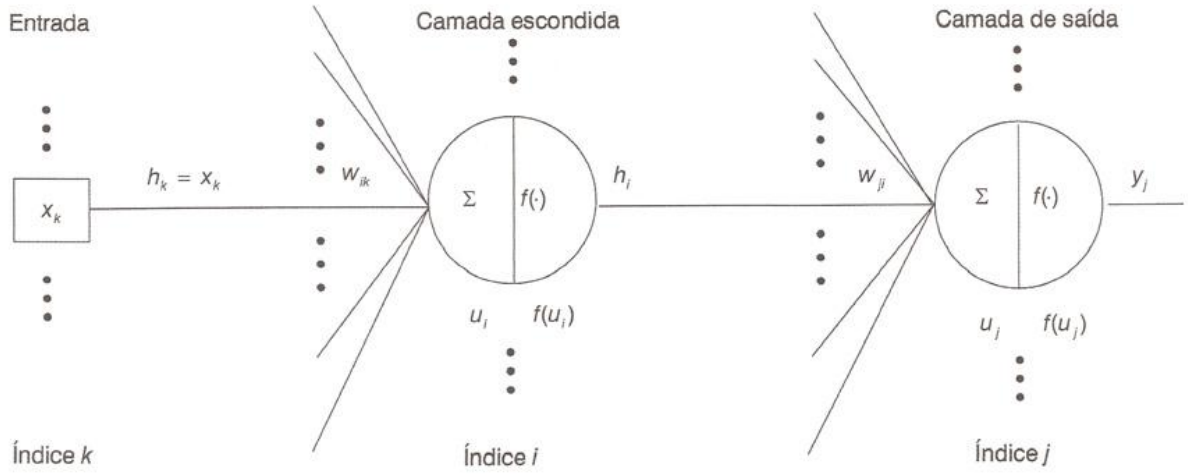

Figura 21 - Esquema da rede MLP e os índices associados

Um neurônio j possui uma saída linear u_j, correspondente à soma ponderada de suas entradas e uma saída, normalmente não-linear, γ_j obtida após a aplicação da função de ativação sobre u_j, ou seja, $\gamma_j = f(u_j)$. Para diferenciar as respostas dos neurônios das camadas de saída e escondidas, estes últimos terão suas saídas referenciadas como $h(u_j)$ para um neurônio i qualquer.

O erro de um neurônio de saída j na iteração n é definido por $e_j(n) = \gamma_d^j(n) - \gamma_j(n)$, sendo a soma dos erros quadráticos de todos os neurônios de saída na iteração n definida por:

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

Como a saída linear do neurônio j da camada de saída é definida por $u_j(n) = \sum_i h_i(n)w_{ji}(n)$, sendo o índice i referente à camada escondida, pode-se reescrever o erro do neurônio j como $e_j(n) = \gamma_d^j(n) - f(u_j(n))$. Assim a soma dos erros quadráticos de todos os neurônios de saída na iteração n pode ser reescrita como:

$$\varepsilon(n) = \frac{1}{2}\sum_j \left(\gamma_d^j(n) - f(u_j(n))\right)^2$$

3.3.9.5 Camada de saída

A idéia é ajustar o vetor de pesos em direção contrária ao gradiente do erro. Assim, as derivadas parciais de ε em relação a cada um dos pesos da camada de saída serão inicialmente obtidas. Para o neurônio j, a derivada parcial de em relação ao peso w_{ji} que o conecta ao neurônio i da camada escondida pode ser obtida por:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = \frac{1}{2}\frac{\partial}{\partial w_{ji}}\left(\gamma_d^j(n) - f(u_j(n))\right)^2$$

Pela regra da cadeia, vê-se:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = \frac{1}{2}2\left[\left(\gamma_d^j(n) - f\left(u_j(n)\right)\right)\right]\frac{\partial}{\partial w_{ji}}\left[\left(\gamma_d^j(n) - f\left(u_j(n)\right)\right)\right]$$

Sabendo que $e_j(n) = \gamma_d^j(n) - f(u_j(n))$, como dito anteriormente, tem-se:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = \left[e_j(n)\right]\frac{\partial}{\partial w_{ji}}\left[\left(\gamma_d^j(n) - f\left(u_j(n)\right)\right)\right]$$

Novamente, pela regra da cadeia, chega-se:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = \left[e_j(n)\right]\frac{\partial}{\partial w_{ji}}\left[\left(\gamma_d^j(n) - f\left(u_j(n)\right)\right)\right]\frac{\partial}{\partial w_{ji}}u_j(n)$$

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = \left[e_j(n)\right]\frac{\partial}{\partial w_{ji}}\left[-f\left(u_j(n)\right)\right]\frac{\partial}{\partial w_{ji}}u_j(n)$$

Como dito anteriormente $u_j(n) = \sum_k h_k(n)w_{jk}(n)$, ou seja, somente o termo em que $k = i$ não terá derivada nula, tem-se que $^{\partial}/_{\partial w_{ji}}\left(u_j(n)\right) = {}^{\partial}/_{\partial w_{ji}}\left(\sum_k h_k(n)w_{jk}(n)\right) = h_i(n)$. Já a derivada da $f\left(u_j(n)\right)$ pode ser representada simplesmente por $f'\left(u_j(n)\right)$, correspondente à derivada da função de ativação do neurônio j em relação ao valor de u_j no instante n. Assim, obtem-se finalmente a equação para o ajuste dos pesos do neurônio j qualquer da camada de saída:

$$\frac{\partial \varepsilon(n)}{\partial w_{ji}} = -e_j(n)\, f'\left(u_j(n)\right)\, h_i(n)$$

3.3.9.6 Camada escondida

Considere que k se refere a uma entrada da rede de duas camadas. Assim, a derivada parcial do erro de saída em relação ao peso pode ser obtida a partir da equação:

$$\frac{\partial \varepsilon(n)}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \frac{1}{2} \sum_j \left(\gamma_d^j(n) - f\left(\sum\nolimits_i h_i(n) w_{ji}(n) \right) \right)^2$$

em que o somatório ocorre sobre todo os neurônios j de saída e pode ser reescrito como:

$$\frac{\partial \varepsilon(n)}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \frac{1}{2} \left(e_1^2(n) + e_2^2(n) + e_3^2(n) + \cdots + e_m^2(n) \right)$$

em que m é o número de neurônios na camada de saída.

Tratando cada termo separadamente, de maneira geral, tem-se:

$$\frac{\partial}{\partial w_{ik}} e_j^2(n) = \frac{\partial}{\partial w_{ik}} \left(\gamma_d^j(n) - f\left(u_j(n)\right) \right)^2$$

Similarmente ao que foi feito anteriormente:

$$\frac{\partial}{\partial w_{ik}} e_j^2(n) = -2\, e_j(n) f'\left(u_j(n)\right) \frac{\partial}{\partial w_{ik}} u_j(n)$$

Como $u_j(n)$ corresponde ao somatório das contribuições ponderadas dos neurônios i conectados a j, a derivada $\frac{\partial}{\partial w_{ik}} u_j(n)$ pode ser obtida por $\frac{\partial}{\partial w_{ik}} \left(\sum_i h_i(n) w_{ji}(n) \right)$. Como somente o neurônio i da camada escondida tem o peso w_{ik} como entrada, a derivada do somatório se reduz simplesmente a:

$$\frac{\partial}{\partial w_{ik}} u_j(n) = \frac{\partial}{\partial w_{ik}} \left(h_i(n) w_{ji}(n) \right)$$

$$= w_{ji}(n) \frac{\partial}{\partial w_{ik}} \left(h_i(n) \right)$$

Pela regra da cadeia, sabe-se que:

$$\frac{\partial}{\partial w_{ik}} u_j(n) = w_{ji}(n)\ h_i\ '\big(u_i(n)\big) \frac{\partial}{\partial w_{ik}} \big(u_i(n)\big)$$

Como $u_j(n)$ corresponde a soma ponderada das entradas conectadas ao neurônio j, a derivada $\frac{\partial}{\partial w_{ik}}\big(u_i(n)\big)$ se reduz somente a $x_k(n)$, já que todos os termos do somatório serão constantes exceto $w_{ik}x_k(n)$, o que nos leva a:

$$\frac{\partial}{\partial w_{ik}} \Big(u_j(n)\Big) = w_{ji}(n)\ h_i\ '\big(u_i(n)\big)x_k(n)$$

Sabendo-se disso,

$$\frac{\partial}{\partial w_{ik}} e_j^2(n) = -2\ e_j(n) f\ '\Big(u_j(n)\Big) \frac{\partial}{\partial w_{ik}} u_j(n)$$

pode ser escrito por

$$\frac{\partial}{\partial w_{ik}} e_j^2(n) = -2\ e_j(n) f\ '\Big(u_j(n)\Big) w_{ji}(n)\ h_i\ '\big(u_i(n)\big)x_k(n)$$

e com isso,

$$\frac{\partial \varepsilon(n)}{\partial w_{ik}} = \frac{\partial}{\partial w_{ik}} \frac{1}{2} \big(e_1^2(n) + \ e_2^2(n) + e_3^2(n) + \cdots + e_m^2(n)\big)$$

será

$$\begin{aligned}\frac{\partial \varepsilon(n)}{\partial w_{ik}} = \frac{1}{2}\big(&-2\ e_1(n) f\ '\big(u_1(n)\big) w_{1i}(n)\ h_i\ '\big(u_i(n)\big)x_k(n) \\ &- 2\ e_2(n) f\ '\big(u_2(n)\big) w_{2i}(n)\ h_i\ '\big(u_i(n)\big)x_k(n)\ +\ \ldots \\ &- 2\ e_m(n) f\ '\big(u_m(n)\big) w_{mi}(n)\ h_i\ '\big(u_i(n)\big)x_k(n)\ \big)\end{aligned}$$

e então:

$$\frac{\partial \varepsilon(n)}{\partial w_{ik}} = \frac{1}{2}(-2\ ) h_i\ '\big(u_i(n)\big)x_k(n) \sum_j e_j(n)\ f\ '\Big(u_j(n)\Big)\ w_{ji}(n)$$

Como o ajuste dos pesos deve ser feito na direção contrária ao gradiente, tem-se que $\Delta w_{ik}(n) \propto -\nabla \varepsilon$. Assim, a equação a seguir apresenta o ajuste

a ser aplicado ao peso arbitrário w_{ik}, que conecta a entrada k ao neurônio i da camada escondida.

$$\Delta w_{ik}(n) = \eta h_i{}'\big(u_i(n)\big) x_k(n) \sum_j e_j(n)\, f'\Big(u_j(n)\Big)\, w_{ji}(n)$$

em que η, como já dito anteriormente, é uma constante de proporcionalidade correspondente à taxa de aprendizado.

Na equação anterior, o termo $\eta h_i{}'\big(u_i(n)\big)$ corresponde à derivada da função de ativação do neurônio i da camada escondida. O seu argumento $u_i(n)$ corresponde a soma ponderada das suas entradas. O termo $\sum_j e_j(n)\, f'\Big(u_j(n)\Big)\, w_{ji}(n)$ corresponde a uma medida de erro do neurônio i da camada escondida. Como o somatório é feito em j, correspondendo aos neurônios da camada de saída, tem-se aqui a soma ponderada de todos os erros dos neurônios de saída pelos pesos que os conectam ao neurônio i da camada escondida. Por meio dessa soma ponderada dos erros da camada de saída, os erros calculados com base no conjunto de treinamento voltam para trás para permitir o ajuste dos neurônios da camada escondida. Esse termo dá o nome ao algoritmo como sendo *error back-propagation*, ou retropropagação de erros.

4 MATERIAL E MÉTODOS

Para a aplicação das técnicas estudas, utilizou-se um conjunto de dados bancários. O objetivo do estudo é encontrar os clientes mais propensos a adiquirem o CDC (Crédito Direto ao Consumidor), com o objetivo final de criar uma campanha de *marketing* ofertando tal produto. O retorno esperado com o uso de modelagem é acertar o público de clientes que receberão o *maillling*, obtendo o maior retorno possível (adesão do cliente).

Segundo Gouveia (2007), CDC é uma modalidade de crédito para aquisição de bens duráveis e serviços. É fornecido por bancos, financeiras e estabelecimentos comerciais que vendem produtos financiáveis via CDC.

O CDC tem prazo variável entre 3 e 48 meses, podendo chegar a 84 meses, quando o bem durável é um automóvel. O prazo para quitação da dívida varia em função do valor e tipo do bem, da capacidade de pagamento do comprador e das condições da economia. Normalmente, o pagamento é em prestações mensais. Geralmente os juros são pré-fixados, mas para prazos maiores que 12 meses pode haver algum reajuste pela TR ou pelo IGP-M (FINANCENTER, 2012).

Os juros são menores até mesmo que o crédito pessoal, mas isso só é possível por que o agente financiador pede garantias. Quando possível, o próprio bem adquirido é dado em garantia. Isso se chama alienação fiduciária. Ou seja, trata-se de um financiamento destinado a aquisição de bens duráveis e serviços, como por exemplo: veículos, eletrodomésticos, eletroeletrônicos, equipamentos profissionais, materiais de construção, vestuário, outros bens não perecíveis - e serviços - assistência técnica, manutenção etc.

O CDC pode ser obtido no estabelecimento vendedor que mantém convênio com uma ou várias instituições financeiras - banco ou financeira. Também há os casos em que o próprio estabelecimento "banca" o financiamento e posteriormente, negocia estes créditos com uma instituição financeira, gerando o CDC-I; nesta modalidade a loja assume o risco de pagamento pelo comprador - chamada Interveniência. O pagamento pode ser realizado por meio de boleto bancário ou *carnet* pagável na loja. O seguro do bem é exigido no caso de veículos. Há outros seguros, como vida e perda de emprego, que poderão ser exigidos.

Normalmente, o preço do seguro é incluído no valor do financiamento. O valor do IOF também é normalmente financiado e a falta de pagamento permite ao vendedor retomar o bem financiado (FINANCENTER, 2012).

O objetivo do banco em questão é saber para quais clientes ofertar esta modalidade de crédito. Utilizando as informações dos clientes que já pertencem ao conjunto de clientes do banco, o objetivo é construir um modelo que forneça a probabilidade de aquisição do financiamento, para novos clientes.

4.1 Descrição do conjunto de dados

O conjunto de dados foi disponibilizado na internet, em uma competição realizada pelo GUSAS (2011) e refere-se a clientes de um banco que adiquiriram ou não o plano de financiamento CDC no mês de agosto de 2011. Dentre os 10 mil clientes, mil adiquiriram a modalidade de crédito.

Para predizer a probabilidade de adesão do CDC, foram disponibilizados uma série de informações sobre os 10 mil clientes. As variáveis cedidas são tanto nominais, ordinais ou contínuas. Na Tabela 5 encontra-se listada o nome da variável mais a descrição da mesma.

Tabela 5 - Caracterização das variáveis em estudo

(continua)

Variável	Descrição
ID	Identificação do cliente
VL_TOTAL_CDB_T0	Valor total em CDB (Certificado de Depósito Bancário)
VL_LIMITE_IMPLANTADO_SM	Valor do saldo médio de limite implantado
VL_LIMITE_UTILIZADO_SM	Valor do saldo médio de limite utilizado
QT_CHEQUE_COMPENSADO	Quantidade de cheques compensados
SEXO	Sexo do cliente (H- homem, M-Mulher)
VL_TOTL_REND	Valor total da renda do cliente
IDADE	Idade do cliente
QTD_ACESSOS_ATM_MES	Quantidade de acessos ao ATM (Automatic Teller Machine, mais conhecido como caixa eletrônico)
QTD_ACESSOS_IB_MES	Quantidade de acessos ao IB (Internet Banking)
QTD_ACESSOS_TMK_MES	Quantidade de acessos ao TMK (Telemarketing)
QT_CDC_LEAS	Quantidade de CDC
VL_SALD_ATIV	Valor do saldo ativo (crédito tomado no banco)
VL_SALD_PASS	Valor do saldo passivo (investimentos bancários do cliente)
VL_SALD_POUP	Valor do saldo de poupança
VL_SALD_PRVD_PRIV	Valor do saldo de previdência
VL_TRANS_INTERNACIONAL	Valor das transações internacionais
VL_TRANS_NACIONAL	Valor das transações nacionais
QT_TRANS_INTERNACIONAL	Quantidade de transações internacionais
QT_TRANS_NACIONAL	Quantidade de transações nacionais
QT_COMPRA_VISA	Quantidade de compras realizadas com Visa
VL_LIMITE_DISPONIVEL_CART_CRED	Valor do limite disponível no cartão de crédito
VL_LIMITE_UTILIZADO_CART_CRED	Valor do saldo médio de limite utilizado no cartão de crédito
VL_LIMITE_IMPLANTADO_CART_CRED	Valor do limite implantado no cartão de crédito
VL_SALDO_DEVEDOR_TOTAL	Valor do saldo devedor total no banco
RENDA_MENSAL	Renda Mensal
AVENC_TOTAL_SCR_CP	Valor total a vencer de crédito pessoal tomado no mercado (incluindo o próprio banco)
VENCD_TOTAL_SCR_CP	Total vencido de crédito pessoal no mercado

Tabela 5 - Caracterização das variáveis em estudo

(conclusão)

Variável	Descrição
AVENC_TOTAL_SCR_CONSIG	Valor total a vencer de consignado tomado no mercado (incluindo o próprio banco)
VENCD_TOTAL_SCR_CONSIG	Total vencido de consignado no mercado
PERFIL_HIST	Perfil do cliente dentro do banco (Investidor ou Tomador)
QTD_DEB_AUTOMATICO	Quantidade de débitos automáticos
VL_DEB_AUTOMATICO	Valor de débitos automáticos
SG_UF	Sigla da unidade da federação em que o cliente abriu conta
QTCLI_SEGUROS_12	Quantidade de seguros que o cliente possui
QTDE_PRODUTOS_PF_12	Quantidade de produtos pessoa física
VL_SM_CAPTACAO_12	Valor do saldo médio de captação no último mês
VL_SM_CRED_PESSOAL_12	Valor do saldo médio de crédito pessoal no último mês
VL_TARIFA_COBRADA_12	Valor médio da tarifa cobrada do cliente no último mês
TOT_SEG_AUTO	Total de meses com seguro auto (de 1 a 9 meses)
MBB_3M	Margem Bruta
SALDO_DISPONIVEL_3M	Saldo do cliente disponível (média trimestral)
VL_TOTAL_INVESTIMENTO_T0	Valor total em Investimentos
FLAG_RESPOSTA	Adquiriu CDCem Ago/11 (1 - Adquiriu, 0 - Não adquiriu)
RESTRICAO_FINANCEIRA	Cliente com restrição financeira (1 - possui, 0 - não possui)
RISCO	Nível de risco de crédito do cliente
ESTADO_CIVIL	Estado civil do cliente
ESCOLARIDADE	Escolaridade do cliente
TEM_PRE_APROV_CDC	Posse de pré-aprovado para CDC (1 - possui; 0 - não possui)
SEGMENTO	Segmento criado pelo banco, que classifica o cliente entre Clássico, Especial e Supremo.

4.2 Sistema computacional SAS

Para a realização deste trabalho foi utilizado o sistema computacional SAS, de domínio privado, existindo a necessidade de licença para utilização do mesmo. SAS é um *softwar*e criado na década de 60, por Jim Goodnight e mais quatro colegas. Atualmente é o *software* mais utilizado no mercado de trabalho, por

garantir as análises realizadas e pela habilidade na manipulação de grandes bases de dados. É uma marca que sempre está presente entre os melhores *softwares*, nas pesquisas realizadas na área de TI (Tecnologia da Informação).

O SAS é uma empresa que está no mercado a mais de 30 anos e no decorrer deste tempo foi aperfeiçoando suas tecnologias e com isso, aumentando seu número de *softwares*. Atualmente existe um *software* para cada perfil de usuários, o que facilitou na escolha do melhor *software* para tal estudo. Todo o trabalho foi realizado utilizando o SAS Enterpise Guide para análises simples e manipulação das bases de dados e o SAS Enterprise Miner para a modelagem.

O SAS Enterprise Miner auxilia no processo de mineração de dados para criar modelos preditivos e descritivos altamente precisos, com base em análises de grandes quantidades de dados de toda uma empresa. É uma ferramenta de fácil manipulação e de capacidades integrada para criar e compartilhar conhecimentos que podem ser usados para melhor tomar decisões. As organizações, com visão de futuro, usam o *software* SAS Enterprise Miner para detectar fraudes, minimizar riscos, prever demandas e aumentar as taxas de resposta para campanhas de marketing.

O SAS Enterprise Miner apoia todo o processo de mineração de dados com um amplo conjunto de recursos. Independentemente da preferência ou nível de habilidade do usuário, o SAS fornece um *software* flexível, que aborda os problemas complexos. No Apêndice D há uma breve descrição sobre o *software* mais um guia introdutório.

O SAS Enterprise Miner inclui um grande benefício que é a auto-documentação. Todos os modelos são criados num fluxo que permite ao desenvolvedor saber o passo a passo do estudo. Essa vantagem diminui o tempo de desenvolvimento de modelo de data mining para os estatísticos ou desenvolvedores.

O *software* permite que os usuários de negócios gerem automaticamente modelos preditivos e ajam sobre eles de forma rápida e eficaz. Resultados analíticos podem ser compreendidos facilmente, o que possibilita a obtenção de conhecimentos necessários para uma melhor tomada de decisão.

O SAS Enterprise Miner permite melhorar a precisão das previsões e compartilhar informações confiáveis a fim de melhorar a qualidade das decisões. Modelos com melhor desempenho melhoram a estabilidade e precisão das previsões, que podem ser verificadas facilmente pelo modelo de avaliação visual e métricas de validação. Previsão de resultados e avaliação estatística de modelos construídos com diferentes abordagens podem ser exibidas lado a lado para facilitar a comparação. Os diagramas resultantes servem como auto-documento de modelos que podem ser facilmente atualizado ou aplicados a novos problemas, sem ter que iniciar tudo novamente. Além disso, o perfil de modelo fornece uma compreensão de como as variáveis preditoras contribuem para o resultado que está sendo modelado.

Facilitar a implantação do modelo e o processo de *scoragem* (processo de aplicação de um modelo para novos dados - é o resultado final de muitos empreendimentos de mineração de dados). SAS Enterprise Miner automatiza o processo tedioso de *scoragem* e fornece o código completo de scoragem para todas as fases de desenvolvimento do modelo no SAS, C, Java e PMML. O código de *scoragem* pode ser implantado em tempo real ou em lotes dentro de ambientes SAS, na Web ou diretamente nos bancos de dados relacionais. O resultado é uma execução mais rápida dos resultados da mineração de dados.

5 RESULTADOS

Para qualquer análise de dados e/ou Data Mining é necessário conhecer as variáveis, seus casos possíveis e distribuições. Inicialmente é essencial realizar uma análise descritiva dos dados. Análise univariada, análise bivariada e análises de correlação para evitar problemas de multicolinearidade.

Nesta análise dispõe-se de 51 variáveis, das quais uma é a variável objetivo (FLAG_RESPOSTA - binária) e as demais são variáveis explicativas, sendo elas binárias, nominais, ordinais e intervalares.

A partir da análise univariada pode-se eliminar algumas variáveis como as que não tem informação suficiente, por exemplo, a variável VL_TOTAL_CDB_T0 com 95% dos dados faltantes (Apêndice B). Já a análise bivariada mostra, por exemplo para a variável SEXO, qual é o número de clientes do sexo masculino que adiquiriam ou não o CDC (Crédito Direto ao Consumidor) e também para o sexo feminino. Todas as variáveis foram analisadas e todas que tiveram alguma categorização ou agrupamento foram renomeadas como “nome_antigo_A”, em que “A” representa algum agrupamento (Apêndice B).

A análise de correlação foi realizada e dentre as variáveis altamente correlacionadas manteve-se apenas as mais importantes. Os dados faltantes, como mencionado no decorrer da dissertação, podem reduzir bruscamente o número de dados válidos para a análise de regressão logística, por exemplo. Para as variáveis com este problema utilizou-se o método de árvore de decisão para inserir valores nos dados sem informação. Neste método os valores faltantes são estimados como se fossem a variável resposta e o restante das variáveis são utilizadas como explicativas. Esta técnica de imputação pode ser mais precisa do que usar simplesmente uma média ou mediana da variável em questão. As análises descritivas das variáveis imputadas e transformadas estão disponíveis no Apêndice B.

Para a modelagem dos clientes que adquirem CDC, partionou-se a base de dados em 70% para a base de treinamento (onde o modelo será construído) e 30% para a base de validação (onde será medido o desempenho do modelo).

Estimou-se um modelo logito binário com as variáveis já descritas anteriormente. Na Tabela 6 têm-se os coeficientes de regressão, as estatísticas de Wald e respectivos intervalos de confiança para cada um dos parâmetros que foi selecionado a partir do método *Stepwise*.

Tabela 6 – Resultado do modelo selecionado a partir do método *Stepwise*

Variável	Domínio	GL	Coeficientes	Erro Padrão	Wald	Sig	IC 95%	
Intercept		1	-1,6078	0,5639	8,13	0,0044	-2,713	-0,5025
IMP_IDADE_A	1 MENOR OU IGUAL A 25 ANOS	1	1,2603	0,175	51,86	<,0001	0,9173	1,6033
IMP_IDADE_A	2 ENTRE 26 E 35 ANOS	1	1,0505	0,1293	66,02	<,0001	0,7971	1,3039
IMP_IDADE_A	3 ENTRE 36 E 50 ANOS	1	0,6586	0,1266	27,05	<,0001	0,4104	0,9068
IMP_QTDE_PRODUTOS_PF_12_A	DE 1 A 5	1	-0,412	0,1079	14,58	0,0001	-0,6235	-0,2006
IMP_QTD_ACESSOS_ATM_MES_A	DE 1 A 5	1	-0,2208	0,1026	4,63	0,0314	-0,422	-0,0197
IMP_QT_CDC_LEAS	0	1	-1,5716	0,1948	65,08	<,0001	-1,9534	-1,1898
IMP_QT_CHEQUE_COMPENSADO_A	0	1	-0,9149	0,1369	44,65	<,0001	-1,1832	-0,6466
IMP_QT_CHEQUE_COMPENSADO_A	DE 1 A 5	1	-0,3616	0,1303	7,7	0,0055	-0,617	-0,1062
IMP_RENDA_MENSAL_A	1 MENOS QUE 500 REAIS	1	-1,3083	0,2381	30,19	<,0001	-1,775	-0,8416
IMP_RENDA_MENSAL_A	2 ENTRE 500 E 1500 REAIS	1	-0,7146	0,1614	19,59	<,0001	-1,031	-0,3982
IMP_RENDA_MENSAL_A	3 ENTRE 1500 E 3000 REAIS	1	-0,2905	0,1328	4,78	0,0287	-0,5508	-0,0302
IMP_RISCO	ALTO	1	-2,168	0,7463	8,44	0,0037	-3,6307	-0,7053
IMP_RISCO	BAIXO	1	0,7617	0,1686	20,41	<,0001	0,4313	1,0921
IMP_SG_UF_A	OUTRAS	1	-0,3735	0,1721	4,71	0,03	-0,7108	-0,0362
IMP_SG_UF_A	SUDESTE	1	-0,5381	0,1325	16,49	<,0001	-0,7979	-0,2784
LOG_IMP_VL_SALD_ATIV		1	0,0369	0,0119	9,67	0,0019	0,0137	0,0602
LOG_IMP_VL_SALD_PRVD_PRIV		1	-0,0503	0,0212	5,61	0,0179	-0,0919	-0,00868
LOG_IMP_VL_TRANS_NACIONAL		1	0,0369	0,0158	5,48	0,0192	0,00601	0,0678
RESTRICAO_FINANCEIRA	0	1	1,4898	0,4936	9,11	0,0025	0,5223	2,4573
SEGMENTO	CLÁSSICO	1	-1,5112	0,2103	51,63	<,0001	-1,9235	-1,099
SEGMENTO	ESPECIAL	1	-0,3157	0,1215	6,75	0,0094	-0,5537	-0,0776
SEXO	H	1	0,3837	0,0963	15,88	<,0001	0,195	0,5724

Pela razão de chance, Tabela 7, conclui-se, por exemplo, que clientes sem nenhuma restrição financeira são 4,436 vezes mais propensos a adiquirirem

CDC do que os clientes com alguma restrição. Já cliente com idade menor ou igual a 25 anos são 3,526 vezes mais propensos a adquirirem CDC que cliente mais velhos que 51 anos. Clientes entre 26 e 35 anos são 2,859 vezes mais propensos que os cliente com idade maior que 51 anos, e assim por diante.

Tabela 7 – Razão de chance para cada uma das variáveis no modelo de Regressão Logística

Variáveis		**Razão de Chance**
IMP_IDADE_A	1 MENOR OU IGUAL A 25 ANOS vs 4 MAIOR OU IGUAL A 51 ANOS	3,526
IMP_IDADE_A	2 ENTRE 26 E 35 ANOS vs 4 MAIOR OU IGUAL A 51 ANOS	2,859
IMP_IDADE_A	3 ENTRE 36 E 50 ANOS vs 4 MAIOR OU IGUAL A 51 ANOS	1,932
IMP_QTDE_PRODUTOS_PF_12_A	DE 1 A 5 vs MAIS OU IGUAL A 6	0,662
IMP_QTD_ACESSOS_ATM_MES_A	DE 1 A 5 vs MAIS OU IGUAL A 6	0,802
IMP_QT_CDC_LEAS	0 vs 1	0,208
IMP_QT_CHEQUE_COMPENSADO_A	0 vs MAIS OU IGUAL A 6	0,401
IMP_QT_CHEQUE_COMPENSADO_A	DE 1 A 5 vs MAIS OU IGUAL A 6	0,697
IMP_RENDA_MENSAL_A	1 MENOS QUE 500 REAIS vs 4 MAIS QUE 3000 REAIS	0,27
IMP_RENDA_MENSAL_A	2 ENTRE 500 E 1500 REAIS vs 4 MAIS QUE 3000 REAIS	0,489
IMP_RENDA_MENSAL_A	3 ENTRE 1500 E 3000 REAIS vs 4 MAIS QUE 3000 REAIS	0,748
IMP_RISCO	ALTO vs MEDIO	0,114
IMP_RISCO	BAIXO vs MEDIO	2,142
IMP_SG_UF_A	OUTRAS vs SUL	0,688
IMP_SG_UF_A	SUDESTE vs SUL	0,584
LOG_IMP_VL_SALD_ATIV		1,038
LOG_IMP_VL_SALD_PRVD_PRIV		0,951
LOG_IMP_VL_TRANS_NACIONAL		1,038
RESTRICAO_FINANCEIRA	0 vs 1	4,436
SEGMENTO	CLÁSSICO vs SUPREMO	0,221
SEGMENTO	ESPECIAL vs SUPREMO	0,729
SEXO	H vs M	1,468

A partir da matriz de confusão da base de validação exposta na Tabela 8 nota-se que dentre os 72 clientes que foram classificados como que adquirem CDC, 47 foram classificados corretamente (65,27%) e dos 2930 clientes que foram classificados como que não adquirem CDC, 2676 foram classificados corretamente (91,33%). Por outro lado, dentre os 301 clientes que adquirem CDC, apenas 47

foram classificados corretamente (15,61%) e dentre os 2701 clientes que não adquirem CDC, 2676 foram classificados corretamente (99,07%).

Tabela 8 – Matriz de confusão para o modelo de Regressão Logística

	Estimado	
Real	**1**	**0**
1	47	254
0	25	2676

A Figura 22 mostra uma visão da árvore de decisão que utilizou a entropia como critério de divisão, restringindo a profundidade a três níveis, para fins de apresentação (o modelo final contou com uma profundidade de cinco níveis). Observa-se, dentro dos retângulos a porcentagem de clientes que não adquirem CDC (0) e os que adquirem (1), tanto para a base de treinamento como na de validação além da frequência em cada base. Abaixo dos nós ficam as variáveis selecionadas para a divisão até que chegue as folhas, quando as divisões adicionais não trazem mais pureza.

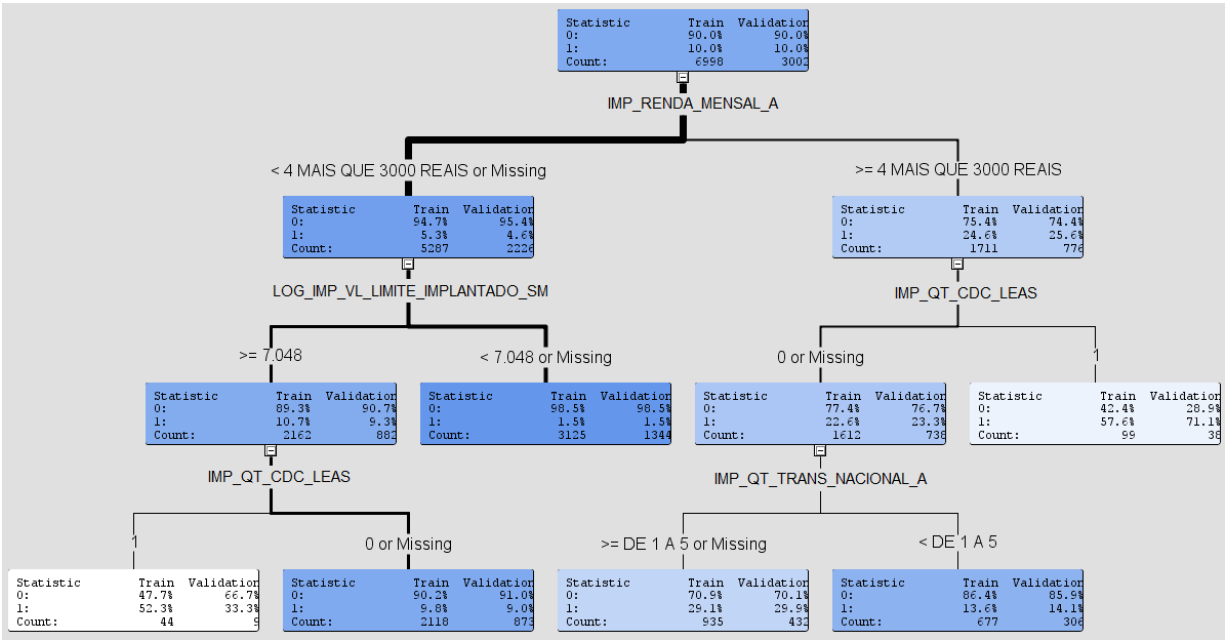

Figura 22 – Ilustração parcial da Árvore de Decisão

Neste modelo são consideradas 9 variáveis importantes para a explicação da variável *target*. A seguir vê-se uma lista (Tabela 9) com as variáveis consideradas importantes, na ordem de importância.

Tabela 9 – Variáveis importantes para o modelo de Árvore de Decisão

Variável	Importância
IMP_RENDA_MENSAL_A	1
IMP_QT_CDC_LEAS	0.61726
LOG_IMP_VL_LIMITE_IMPLANTADO_SM	0.46242
IMP_QT_TRANS_NACIONAL_A	0.43282
LOG_IMP_VL_SALD_ATIV	0.38773
LOG_IMP_VL_TRANS_NACIONAL	0.28886
IMP_QT_CHEQUE_COMPENSADO_A	0.22472
IMP_IDADE_A	0.16863
IMP_TOT_SEG_AUTO_A	0.12289

A Tabela 10 mostra a matriz de confusão da base de validação para a árvore de decisão. Dentre os 47 clientes que foram classificados como que adquirem CDC, 30 foram classificados corretamente (63,82%) e dos 2955 clientes que foram classificados como que não adquirem CDC, 2684 foram classificados corretamente (90,82%). Por outro lado, dentre os 301 clientes que adquirem CDC, apenas 30 foram classificados corretamente (9,96%) e dentre os 2701 clientes que não adquirem CDC, 2684 foram classificados corretamente (99,37%).

Tabela 10 – Matriz de confusão para o modelo de Árvore de Decisão

	Estimado	
Real	**1**	**0**
1	30	271
0	17	2684

Finalmente, no Apêndice C encontram-se as regras em inglês das divisões de cada nó, que mostram como programar as divisões. A sua estrutura começa mostrando as variáveis a serem divididas no nó e seus intervalos, faixas, ou quantidades. No exemplo abaixo, toma-se a variável transformada do valor do limite implantado SM, e verifica-se se é menor do que 7,04. Além disso, a variável agrupada renda mensal deve ser “entre 1500 e 3000 reais”. Caso essas condições sejam satisfeitas, o cliente é alocado ao nó 9, que será considerado como FLAG_RESPOSTA = 0 (não adquire CDC). Como vê-se, para fins de interpretação do resultado, a árvore é bem mais simples de ser compreendida.

Node = 9
----
if Transformed: Imputed VL_LIMITE_IMPLANTADO_SM < 7.04795 or MISSING
AND Imputed RENDA_MENSAL_A = 3 ENTRE 1500 E 3000 REAIS then
Tree Node Identifier = 9
Number of Observations = 108
Predicted: FLAG_RESPOSTA=0 = 0.85
Predicted: FLAG_RESPOSTA=1 = 0.15

A Tabela 11 mostra a alocação dos pesos na rede neural, para algumas variáveis (apenas para fins de apresentação), sendo que em azul estão os pesos positivos, e em vermelho os pesos negativos, sendo H11, H12, e H13 os neurônios da camada escondida (*Hidden Layer*). Esta é uma rede neural com uma camada escondida com três neurônios e função de ativação mlogística.

Tabela 11 – Alocação dos pesos na rede neural (tabela ilustrativa pois contém apenas algumas variáveis)

Origem	Destino	Peso
LOG_IMP_MBB_3M	H11	0.070411
LOG_IMP_SALDO_DISPONIVEL_3M	H11	0.048623
LOG_IMP_VL_TARIFA_COBRADA_12	H11	-0.167993
LOG_IMP_MBB_3M	H12	0.015025
LOG_IMP_SALDO_DISPONIVEL_3M	H12	0.389042
LOG_IMP_VL_TARIFA_COBRADA_12	H12	-0.056576
LOG_IMP_MBB_3M	H13	-0.007322
LOG_IMP_SALDO_DISPONIVEL_3M	H13	-0.244988
LOG_IMP_VL_TARIFA_COBRADA_12	H13	0.064894
RESTRICAO_FINANCEIRA0	H11	0.587923
RESTRICAO_FINANCEIRA0	H12	0.433778
RESTRICAO_FINANCEIRA0	H13	0.296037

Analisando a quantidade de acerto, têm-se a matriz de confusão da base de validação (Tabela 12). Dentre os 110 clientes que foram classificados como que adquirem CDC, 71 foram classificados corretamente (64,54%) e dos 2892 clientes que foram classificados como que não adquirem CDC, 2662 foram classificados corretamente (92,04%). Por outro lado, dentre os 301 clientes que adquirem CDC, apenas 71 foram classificados corretamente (23,58%) e dentre os

2701 clientes que não adquirem CDC, 2662 foram classificados corretamente (98,55%).

Tabela 12 - Matriz de confusão para o modelo de Rede Neural

	Estimado	
Real	**1**	**0**
1	71	230
0	39	2662

Os três modelos: regressão logística, árvore de decisão e rede neural apresentaram a área da curva ROC igual a 0,864, 0,833, 0,86 respectivamente (Figura 23). Pode-se notar que a área da curva ROC para todos os modelos indica uma discriminação excelente (o modelo discrimina de modo excelente os clientes que têm a característica de interesse dos clientes que não têm), porém é visível a partir das matrizes de confusão que os itens de interesse (FLAG_RESPOSTA=1) estão sendo classificados erroneamente, sendo acertivos em apenas 15,61% para regressão logística, 9,96% para a árvore de decisão e 23,58% para a Rede Neural.

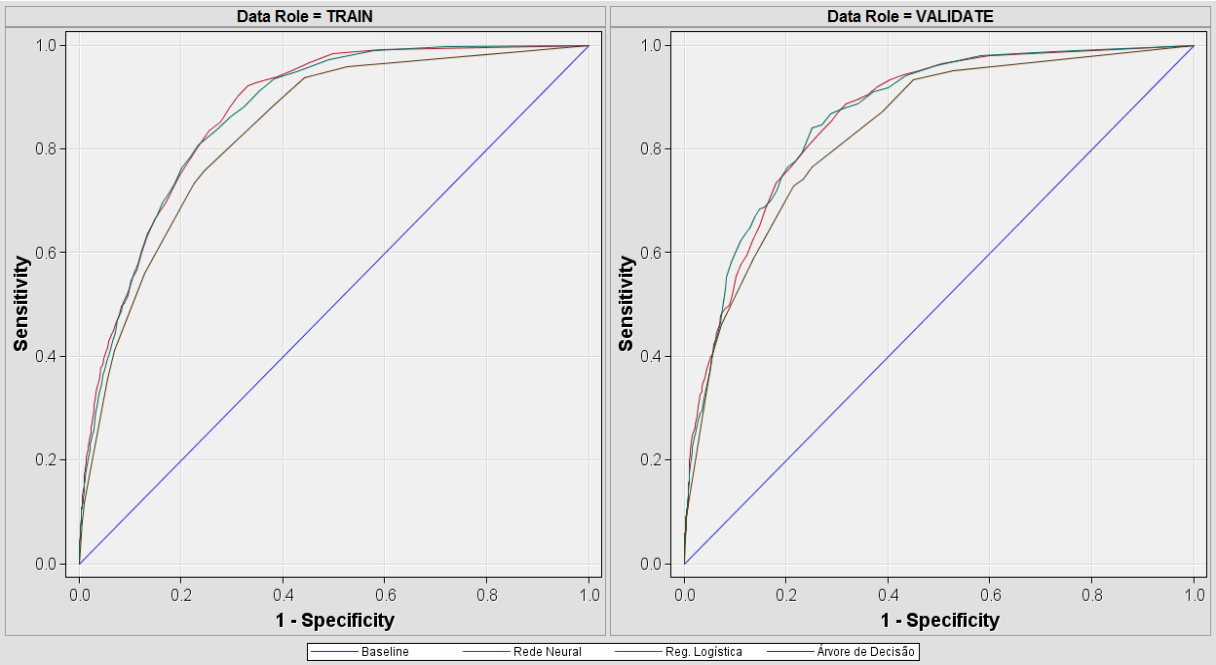

Figura 23 – Gráfico da curva ROC para os três modelos iniciais (Regressão Logística na cor verde, Árvore de decisão na cor marrom e Rede Neural na cor vermelha)

Isso pode ser explicado pela frequência de eventos de interesse comparado aos demais (9000 clientes que não adquirem CDC e apenas 1000 clientes que adquirem). A proporção desbalanceada pode causar um alto valor da área da curva ROC, sem atingir o objetivo principal, dado que percentualmente a quantidade de eventos de interesse não é significativa.

Com o objetivo de suavizar este problema, selecionou-se aleatorimente 1500 clientes que não adquirem CDC e mantêve-se os mil clientes que adquiram. Desta forma a base disponível para o próximo passo será de 2500 clientes, onde 40% adquire CDC e 60% não adquire. Supondo que a proporção real dentro do banco seja esta.

Toda a análise descritva univariada, bivariada, corelações, além das imputações e transformações foram refeitas e os resultados foram mais interessantes. As Tabelas 13, 14 e 15 mostram a matriz de confusão para este novo estudo e como pode-se notar, os modelos foram mais acertivos.

Para Regressão Logística, dentre os 262 clientes que foram classificados como que adquirem CDC, 187 foram classificados corretamente (71,37%) e dos 490 clientes que foram classificados como que não adquirem CDC, 376 foram classificados corretamente (76,73%). Por outro lado, dentre os 301 clientes que adquirem CDC, 187 foram classificados corretamente (62,12%) e dentre os 451 clientes que não adquirem CDC, 376 foram classificados corretamente (83,37%).

Tabela 13 - Matriz de confusão para o modelo de Regressão Logística (2)

	Estimado	
Real	**1**	**0**
1	187	114
0	75	376

Para Árvore de Decisão, dentre os 283 clientes que foram classificados como que adquirem CDC, 194 foram classificados corretamente (68,55%) e dos 469 clientes que foram classificados como que não adquirem CDC, 362 foram classificados corretamente (77,18%). Por outro lado, dentre os 301 clientes que adquirem CDC, 194 foram classificados corretamente (64,45%) e dentre os 451 clientes que não adquirem CDC, 362 foram classificados corretamente (80,26%).

Tabela 14 - Matriz de confusão para o modelo de Árvore de Decisão (2)

	Estimado	
Real	**1**	**0**
1	194	107
0	89	362

Já para Rede Neural, dentre os 269 clientes que foram classificados como que adquirem CDC, 188 foram classificados corretamente (69,88%) e dos 483 clientes que foram classificados como que não adquirem CDC, 370 foram classificados corretamente (76,60%). Por outro lado, dentre os 301 clientes que adquirem CDC, 188 foram classificados corretamente (62,45%) e dentre os 451 clientes que não adquirem CDC, 371 foram classificados corretamente (82,03%).

Tabela 15 - Matriz de confusão para o modelo de Rede Neural (2)

	Estimado	
Real	**1**	**0**
1	188	113
0	81	370

Os três novos modelos: regressão logística (2), árvore de decisão (2) e rede neural (2) apresentaram a área da curva ROC igual a 0,844, 0,814 e 0,831 respectivamente (Figura 24). Pode-se notar que a área da curva ROC para todos os modelos indica uma discriminação excelente (o modelo discrimina de modo excelente os clientes que têm a característica de interesse dos clientes que não têm), com um melhor acerto na variável target de interesse.

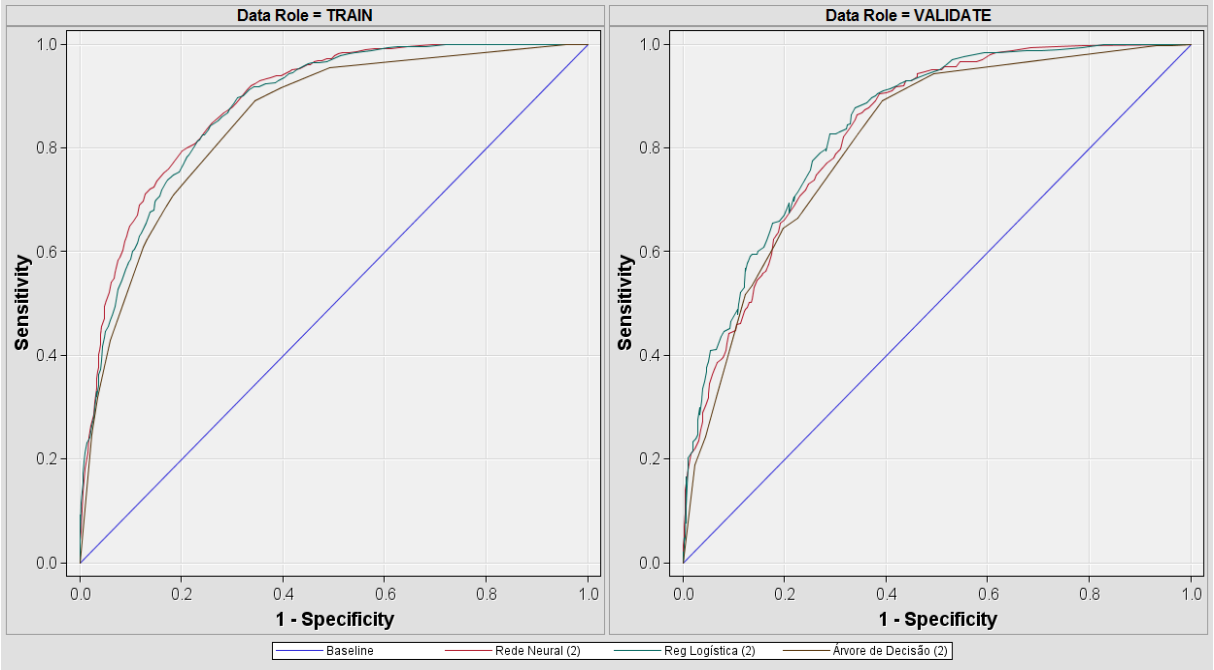

Figura 24 – Gráfico da curva ROC para os três modelos (Regressão Logística (2) na cor verde, Árvore de Decisão (2) na cor marrom e Rede Neural (2) na cor vermelha)

Neste caso, usando a área da curva ROC como parâmetro de decisão, o melhor modelo dentre os 3 desenvolvidos seria o de Regressão Logística. Para este modelo, tem-se na Figura 25 o gráfico do *Lift*. Supondo que o interesse do banco seja ofertar CDC para seus clientes, de uma forma aleatória com 10% da base o retorno seria menor do que se usasse o resultado do modelo. Usando o modelo para selecionar o melhor público a se oferecer CDC, para 10% da base, o acerto do melhor público seria 2,20 vezes melhor.

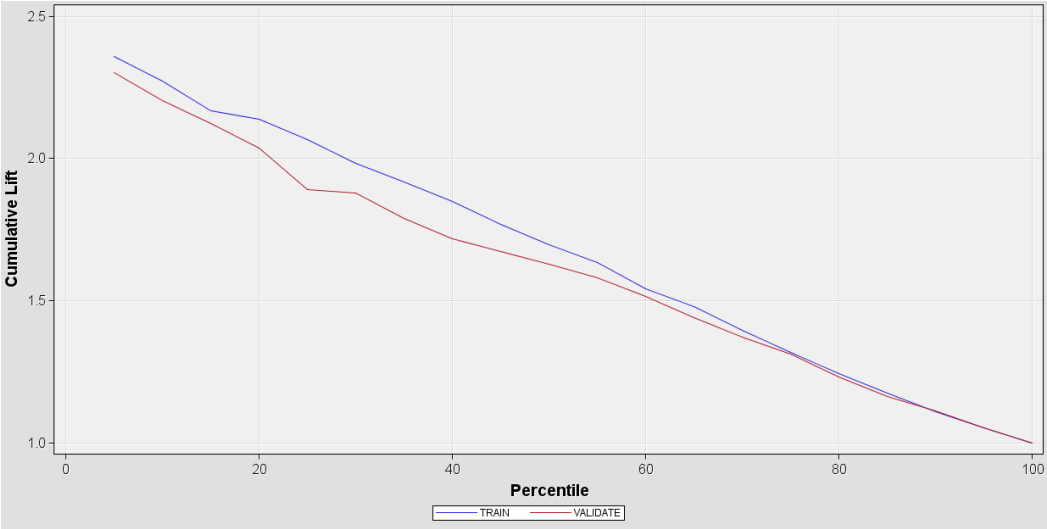

Figura 25 – Gráfico *lift* para o modelo de Regressão Logística (2) onde o azul representa a base de treinamento e o vermelho a base de validação

Já na Figura 26 pode-se ver o comportamento do *lift* para os 3 modelos desenvolvidos. Sendo o décimo percentil da Regressão Logística (2) igual a 2,20, da Árvore de Decisão (2) igual a 2,04, e da Rede Neural (2) igual a 2,13.

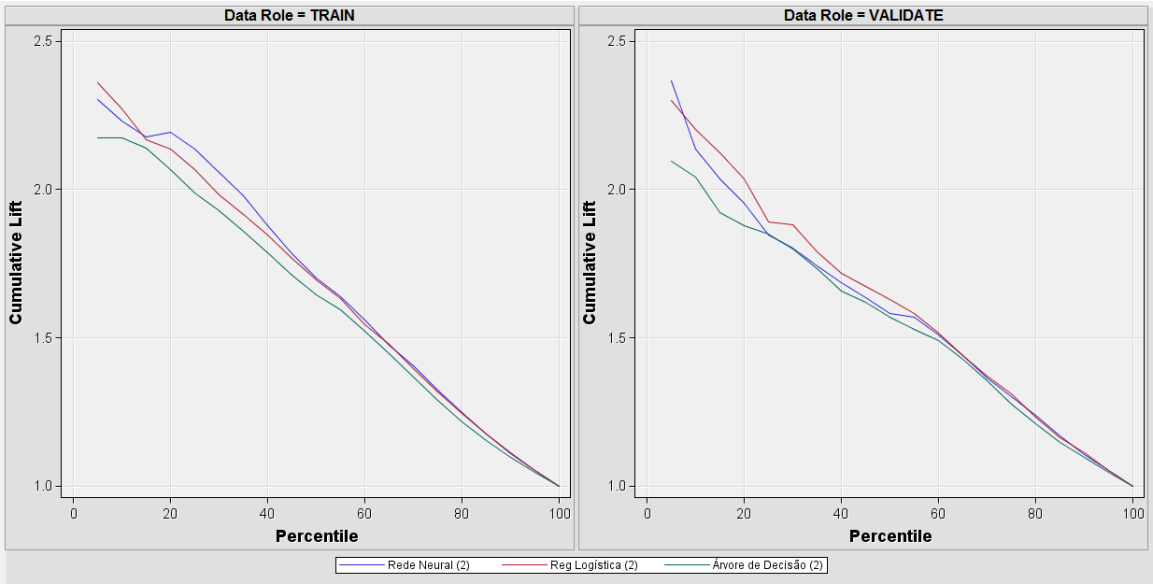

Figura 26 – Gráfico *lift* para os três modelos desenvolvidos (Regressão Logística (2) na cor vermelha, Árvore de Decisão (2) na cor verde e Rede Neural (2) na cor azul)

6 CONCLUSÃO

O objetivo desse trabalho foi dissertar sobre as técnicas de *data mining* mais difundidas: regressão logística, árvore de decisão, e rede neural, além de avaliar se tais técnicas oferecem ganhos financeiros para instituições privadas quando utilizadas corretamente.

Com a aplicação na base de dados de um banco, pôde-se mostrar que os modelos são capazes de oferecer rendimento monetário para as instituições que os usam. O objetivo do banco é encontrar quais são os clientes mais propensos a adquirem o CDC (Crédito Direto ao Consumidor), com o objetivo final de criar uma campanha de *marketing* ofertando tal produto. O retorno esperado com o uso de modelagem, é acertar o público de clientes que receberão o *mailling*, obtendo o maior retorno possível (adesão do cliente).

Supondo que a proporção real de clientes que adquirem CDC seja de 40% e que o interesse do banco seja fazer a campanha de *marketing* para 10% dos clientes, o retorno esperado sem modelo é de 40% dos clientes que receberam a campanha aderindo ao CDC. Por outro lado, se o modelo entregar um *lift* de 1,5 para o primeiro decil, significa que ao estimular esses clientes obter-se-á um retorno 50% superior ao retorno médio.

Logo, os *lift´s* obtidos na modelagem mostram o quanto o emprego do modelo otimiza a lista de seleção de clientes que participarão da campanha. O objetivo do banco é atingir eficientemente a grande base de clientes potenciais. As três técnicas forneceram resultados muito similares e mostraram que a utilização de Data Mining pode ajudar no objetivo do banco. Sendo assim, o critério para a seleção do melhor modelo deve ser a facilidade de implantação e uso. Portanto, pelo que foi visto anteriormente, a árvore de decisão é mais apropriada por apresentar maior facilidade na interpretação dos resultados para o gestor de negócios.

A primeira dificuldade que surge em qualquer tarefa de modelagem diz respeito à elaboração de uma base de dados em condições apropriadas para o estudo. É preciso escolher e preparar um grande volume de dados, sendo necessário observar as condições de preechimento das variáveis e, caso necessário eliminar registos sobre os quais se desconfia da veracidade. A base de dados

utilizada no presente estudo contém algumas variáveis com elevadas porcentagens de *missing*, as quais foram extraídas da análise. A ausência destas variáveis não prejudicou os modelos desenvolvidos, porém poderiam ter enriquecido-os, se significativas.

Sendo assim, é importante ressaltar que o tratamento das informações é de fundamental importância para que o processo de modelagem se desenvolva bem. Modelos bem desenvolvidos são inúteis se as informações para a modelagem não tiverem qualidade. O tratamento da informação deve ser mantida constante dentro de qualquer instituição, para que análises estatísticas tenham qualidade. Dados faltantes devem ser tratados e um sistema de coleta de informação deve ser criado de forma que minimize possíveis erros humanos.

É de interesse realizar posteriormente um estudo detalhado das técnicas de data mining aplicadas a outros tipos de variáveis resposta (nominal ou ordinal), além de outras técnicas também utilizadas em mineração de dados, como clusterização e cesta de produtos.

REFERÊNCIAS

AMEMIYA, T. **Advanced Econometrics***.* 9th ed. Cambridge: Harvard University Press, 1985. 521p.

BASSANEZI, R.C. **Ensino-aprendizagem com modelagem matemática.** São Paulo: Contexto, 2004. 389p.

BEALE, R.; JACKSON, T. **Neural computing:** an introduction. Bristol, UK: IOP, 1990. 240p.

BECK, N.; KING G.; ZENG L. Improving Quantitative Studies of International Conflict: A Conjecture. **American Political Science Review,** Washington, v. 94, n. 1, p 21-35, Mar. 2000.

BERRY, M.J.A.; LINOFF, G.S. **Data mining techniques:** for marketing, sales, and customer relationship management. New York: John Wiley, 2004. 672p.

BLUM, A. **Neural Networks in C++.** New York: Wiley, 1992. 224p.

BOGER, Z.; GUTERMAN H. Knowledge extraction from artificial neural network models. In: IEEE SYSTEMS, MAN, AND CYBERNETICS CONFERENCE, 1997, Florida. **Anais…** Flórida: IEEE, 1997. p: 3030-3035.

BRAGA, A.P.; CARVALHO A.C.P.L.F.; LUDEMIR T.B. **Redes Neurais Artificiais:** Teoria e Aplicações. Rio de Janeiro: LTC Livros Técnicos e Cientificos Editora, 2000. 226p.

BREIMAN, L.; FRIEDMAN, J.H.; OLSHEN, R.A.; STONE, C.J. **Classification and Regression Trees.** Belmont, California: Wadsworth, 1984. 368p.

CHORÃO, L.A.R. **Logit vs Redes Neuronais Artificiais:** Um exemplo aplicado a cartões de crédito. 2005. 156p. Dissertação (Mestrado em Estatística e Gestão de Informação) – Instituto Superior de Estatística e Gestão de Informação, Universidade Nova de Lisboa, Lisboa, 2005.

CORTEZ, P.; NEVES, J. **Redes Neuronais Artificiais.** Braga: Escola de Engenharia Universidade do Minho, 2000. 52p.

CRAMER, J.S. The Origins of Logistic Regression. **Tinbergen Institute Discussion Papers** 02-119/4, Tinbergen Institute, 2002.

CYBENKO, G. Approximation by superpositions of a sigmoid function. **Mathematics of Control, Signals and Systems**, New York, v. 2, p. 303-314, 1989.

DAMÁSIO, A.R. **O Erro de Decartes:** Emoção, Razão e Cérebro Humano. Companhia das Letras, 1996. 336p.

DILLY, R. **Data Mining:** an introduction. Disponível em: < http://www.pcc.qub.ac.uk/tec/courses/datamining/stu_notes/dm_book_1.html>. Acesso em: 16 dez. 2010.

DINIZ, C.A.; LOUZADA-NETO, F. **Data Mining:** uma introdução. São Carlos: Associação Brasileira de Estatística, 2000. 123p.

EISINGA, R.; FRANSES P.; DIJK D. Timing of Vote Decision in First and Second Order Dutch Elections 1978-1995 Evidence from Artificial Neural Networks. **Political Analysis,** Oxford, v. 7, n. 1, p. 117-142, 1998.

FAYYAD, U.M.; PIATETSKI-SHAPIRO, G.; SMYTH, P. The KDD Process for Extracting Useful Knowledge from Volumes of Data. **Communications of the ACM**, New York, v. 39, p.27-34, Nov. 1996.

FAYYAD, U.M.; STOLORZ, P. Data mining and KDD: promise and challenges. **Future Generation Computer Systems**, North-Holland, v.13, p.99-115, Nov. 1997.

FINANCENTER. Seu guia de finanças pessoais. Disponível em: <http://financenter.terra.com.br/Index.cfm/Fuseaction/Secao/Id_Secao/224>. Acesso em: 11 jun. 2012.

GOUVEIA, A. CDC – Crédito Direto ao Consumidor. [18 de outubro, 2007]. Disponível em: <http://endinheirado.wordpress.com/2007/10/18/cdc-credito-direto-ao-consumidor/>. Acesso em: 11 jun. 2012.

GUPTA, S.; HANSSENS, D.; HARDIE, B.; HAHN, W.; KUMAR, V.; LIN, N.; SRIRAM, N.R.S. Modeling Customer Lifetime Value. **Journal of Service Research**, Thousand Oaks, v. 9, n. 2, p. 139-155, Nov. 2006.

GURNEY, K. **An introduction to Neuronal Network.** London: CRC Press, 1997. 234p.

GUSAS. Grupo de Usuários SAS. Disponível em: <http://gusasbrasil.ning.com/>. Acesso em: 13 de out. 2011.

HAIR, J.F.; TATHAM, R.L.; ANDERSON, R.E.; BLACK, W. **Análise Multivariada de Dados.** Tradução de A.S. Sant´Anna; A.C. Neto. 5. ed. Porto Alegre: Bookman, 2005. 593p.

HAN, J.; KAMBER, M. **Data Mining:** Concepts and Techniques. 2nd ed. San Francisco: Elsevier, 2006. 551p.

HAYKIN, S. **Neuronal Networks:** A comprehensive foundation. New Jersey: Prentice Hall, 1999. 842p.

HENLEY, J.A.; MCNEIL B.J. The Meaning and Use of the Area Under the Receiver Operating Characteristics (ROC) Curve. **Radiology**, Oak Brook, p. 29-36, Apr. 1982.

HOSMER, D.W.; LEMESHOW, S. **Applied logistic regression.** 2nd ed. New York: Wiley, 2000. 375p.

ISHIKAWA, N.I. **Uso de tranformações em modelos de regressão logística.** 2007. 92p. Dissertação (Mestrado em Ciências) – Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2007.

KASS, G.V. An Exploratory Technique for Investigating Large Quantities of Categorical Data. **Applied Statistics**, Abingdon, v. 29, n. 2, p. 119-127, 1980.

KOHONEN, T. **Self-Organizing Maps.** 3rd. ed. New York: Information Sciences, 2001. 501p.

LAW, R.; PINE R. Tourism demand forecasting for the tourism industry: a neural network approach. In: ZANG, G.P. **Neural networks in businesses forecasting.** IRM Press, 2004. chap. 6

LEEFLANG, P.S.H.; WITTINK, D.R. Building models for marketing decisions: Past, present and future. **International Journal of Research in Marketing**, Maryland Heights, v. 17, n. 2/3, p. 105-126, Apr. 2000.

LITTLE, J.D.C. Models and Managers: The Concept of a Decision Calculus. **Management Science**, Hanover, v. 50, n. 12, p. 1841-1853, Dec. 2004.

MANNILA, H. Data mining: machine learning, statistics and databases. In: INTERNATIONAL CONFERENCE ON STATISTICS AND SCIENTIFIC DATABASE MANAGEMENT, 1996, Estocolmo. **Anais…** Estocolmo: EIC, 1996. p. 2-9.

MARTINEZ-LOPEZ, F.J.; CASILLAS, J. Marketing Intelligent Systems for consumer behaviour modelling by a descriptive induction approach based on Genetic Fuzzy Systems. **Industrial Marketing Management**, Maryland Heights, v. 38, n. 7, p. 714-731, Oct. 2009.

MCLACHLAN, G. **Discriminant Analysis and Statistical Pattern Recognition.** New York: John Wiley, 1992. 519p.

MCNELIS, P.D. **Neural Networks in Finance:** Gaining Predictive Edge in the Market. Elsevier Academic Press, 2005. 256p.

MONTEGOMERY, D. C.; PECK, E. A.; VINING, G. G. **Introduction to linear regression analysis**. 4th ed. New York: Wiley, 2006. 613p.

MORGAN, J.N.; SONQUIST, J.A. Problems in the Analysis of Survey Data, and a Proposal. **Journal of the Americal Statistical Association**, Alexandria, v. 58, n. 302, p. 415-435, Jun.1963.

NEVES, J.C.; VIEIRA A. Estimating Banruptcy Using Neural Networks Trained with Hidden Layer Learning Vector Quantization. Lisboa: **Working Paper**,Departamento de Gestão, ISEG, UTL., 2004, Departamento de Gestão, ISEG, UTL.

QUINLAN, R.J. Discovering Rules from Large Collections of Examples: A Case Study. In: MICHIE D. **Expert Systems in the Micro Electronic Age.** Edinburgh University Press, 1979. 287p.

QUINLAN, J.R. Induction of Decision Trees. **Machine Learning**, Boston, v. 1, n. 1, p. 81-106. 1986.

QUINLAN, J.R. **C4.5: Programs for Machine Learning**. San Mateo, CA: Morgan Kaufman, 1993. 302p.

REED, R.D.; MARKS II, R.J. **Neuronal Smithing:** Supervised Learning in feedward Artificial Neuronal Network. Cambridge: MIT, 1999. 352p.

RIGBY, D.K.; LEDINGHAM, D. CRM Done Right. **Harvard Business Review**, Cambridge, v. 82, p. 118-129, Nov. 2004.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, Washington, v. 65, n. 6, p. 386-408, Nov. 1958.

ROSENBLATT, F. **Principles of Neurodynamics:** Perceptrons and theory of brain mechanisms. New York: Spartan Books, 1962. 622p.

SARMA, K.S. **Predictive Modeling with SAS Enterprise Miner.** Cary: SAS Press, 2009. 360p.

SHACHMUROVE, Y. **Applying artificial neural networks to business, Economics and finance.** CARESS Working Papers: UCLA Department of Economics, 2002. 43p.

SUMATHI, S.; SIVANANDAM, S.N. **Introduction to data mining and its applications.** Berlin: Springer-Verlag, 2006. 828p.

SWINGLER, K. **Applying neural networks:** a practical guide. London: Academic Press, 1996. 303p.

THAWORNWONG, S.; ENKE D. Forecasting stock returns with artificial neural networks. In: ZANG, G.P. **Neural networks in businesses forecasting**. IRM Press, 2004. chap 3.

ZHANG, Y.; AKKALADEVI, S.; VACHTSEVANOS, G.; LIN T. Granular neural web agents for stock prediction. **Soft Computing**, Belin, v. 6, p. 406 – 413, 2002.

APÊNDICES

APÊNDICE A

Imagine um exemplo onde a variável resposta seja binária (0 ou 1) e que existam 3 variáveis independentes (X_1, X_2 e X_3). A Tabela 16 mostra o conjundo de dados deste exemplo.

Tabela 16 - Conjunto de dados ilustrativo

Observação	**Y**	X_1	X_2	X_3
1	1	Fem	1,70	1
2	0	Fem	1,62	1
3	0	Masc	1,85	0
4	0	Masc	1,80	0
5	0	Masc	1,85	0
6	0	Masc	1,80	0
7	1	Fem	1,70	1
8	1	Fem	1,70	1
9	0	Fem	1,53	1
10	0	Fem	1,62	1

Note que:

- As observações 1, 8 e 7 são iguais: $m_1 = 3$;
- As observações 2 e 10 são iguais: $m_2 = 2$;
- As observações 5 e 3 são iguais: $m_3 = 2$;
- As observações 6 e 4 são iguais: $m_4 = 2$;
- A observação 9 aparece apenas uma vez: $m_5 = 1$;

Assim:

$$J = 5$$

e

$$\sum_{j=}^{J} m_j = (m_1 + m_2 + m_3 + m_4 + m_5) = 10 = n$$

APÊNDICE B

A seguir estão as análises descritivas da base de dado bruta, ou seja, sem nenhuma alteração. Variáveis com final "_A" são variáveis agrupas antes da modelagem e da imputação de dados.

SEXO (Sexo do cliente):

SEXO	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
H	5479	54,79	5479	54,79
M	4521	45,21	10000	100

ESTADO_CIVIL (Estado civil do cliente):

ESTADO_CIVIL	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
	150	1,5	150	1,5
DIVORCIADO	511	5,11	661	6,61
NÃO INFORMADO	3235	32,35	3896	38,96
SOLTEIRO	5698	56,98	9594	95,94
VIÚVO	406	4,06	10000	100

ESCOLARIDADE (Escolaridade do cliente):

ESCOLARIDADE_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
	927	9,27	927	9,27
ENSINO MÉDIO	3575	35,75	4502	45,02
SEM ESCOLARIDADE / ENSINO FUNDAMENTAL	1843	18,43	6345	63,45
SUPERIOR	3655	36,55	10000	100

PERFIL_HIST (Perfil do cliente dentro do banco):

PERFIL_HIST	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
	879	8,79	879	8,79
INVESTIDOR	3954	39,54	4833	48,33
NEUTRO	1364	13,64	6197	61,97
TOMADOR	3803	38,03	10000	100

RESTRICAO_FINANCEIRA (Cliente com restrição financeira (1 - possui, 0 - não possui)):

RESTRICAO_FINANCEIRA	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	8527	85,27	8527	85,27
1	1473	14,73	10000	100

RISCO (Nível de risco de crédito do cliente):

RISCO	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
	35	0,35	35	0,35
ALTO	1185	11,85	1220	12,2
BAIXO	7294	72,94	8514	85,14
MÉDIO	1486	14,86	10000	100

SEGMENTO (Segmento criado pelo banco):

SEGMENTO	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
CLÁSSICO	4040	40,4	4040	40,4
ESPECIAL	3441	34,41	7481	74,81
SUPREMO	2519	25,19	10000	100

SG_UF (Sigla da unidade da federação em que o cliente abriu conta):

SG_UF_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
	346	3,46	346	3,46
OUTRAS	1423	14,23	1769	17,69
SUDESTE	7078	70,78	8847	88,47
SUL	1153	11,53	10000	100

TEM_PRE_APROV_CDC (Posse de pré-aprovado para CDC (1 - possui; 0 - não possui)):

TEM_PRE_APROV_CDC	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	5645	56,45	5645	56,45
1	4355	43,55	10000	100

IDADE (Idade do cliente):

IDADE_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	148	1,48	148	1,48
1 MENOR OU IGUAL A 25 ANOS	1677	16,77	1825	18,25
2 ENTRE 26 E 35 ANOS	2773	27,73	4598	45,98
3 ENTRE 36 E 50 ANOS	2756	27,56	7354	73,54
4 MAIOR OU IGUAL A 51 ANOS	2646	26,46	10000	100

QT_CDC_LEAS (Quantidade de CDC (0 - não tem outro CDC, 1 - tem outro CDC)):

QT_CDC_LEAS	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
.	194	1,94	194	1,94
0	9608	96,08	9802	98,02
1	198	1,98	10000	100

QT_CHEQUE_COMPENSADO (Quantidade de cheques compensados):

QT_CHEQUE_COMPENSADO_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	9	0,09	9	0,09
0	7149	71,49	7158	71,58
DE 1 A 5	2034	20,34	9192	91,92
MAIS OU IGUAL A 6	808	8,08	10000	100

QT_COMPRA_VISA (Quantidade de compras realizadas com Visa):

QT_COMPRA_VISA_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	357	3,57	357	3,57
0	5855	58,55	6212	62,12
DE 1 A 5	1815	18,15	8027	80,27
MAIS OU IGUAL A 6	1973	19,73	10000	100

QT_TRANS_INTERNACIONAL (Quantidade de transações internacionais):

QT_TRANS_INTERNACIONAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	1707	17,07	1707	17,07
0	7959	79,59	9666	96,66
MAIS OU IGUAL A 1	334	3,34	10000	100

QT_TRANS_NACIONAL (Quantidade de transações nacionais):

QT_TRANS_NACIONAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	1707	17,07	1707	17,07
0	4665	46,65	6372	63,72
DE 1 A 5	1573	15,73	7945	79,45
MAIS OU IGUAL A 6	2055	20,55	10000	100

QTCLI_SEGUROS_12(Seguros que o cliente possui (0 - não possui seguro, 1 - possui seguro)):

QTCLI_SEGUROS_12	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
0	5536	55,36	5536	55,36
1	4464	44,64	10000	100

QTD_ACESSOS_ATM_MES (Quantidade de acessos ao ATM (Automatic Teller Machine, mais conhecido como caixa eletrônico)):

QTD_ACESSOS_ATM_MES_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	3475	34,75	3475	34,75
DE 1 A 5	4912	49,12	8387	83,87
MAIS OU IGUAL A 6	1613	16,13	10000	100

QTD_ACESSOS_IB_MES (Quantidade de acessos ao IB (Internet Banking)):

Variável com 74,06% de valores faltantes – excluída da análise.

QTD_ACESSOS_TMK_MES (Quantidade de acessos ao TMK (Telemarketing)):

Variável com 74,94% de valores faltantes – excluída da análise.

QTD_DEB_AUTOMATICO (Quantidade de débitos automáticos):

Variável com 68,80% de valores faltantes – excluída da análise.

QTDE_PRODUTOS_PF_12 (Quantidade de produtos pessoa física):

QTDE_PRODUTOS_PF_12_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	456	4,56	456	4,56
DE 1 A 5	6466	64,66	6922	69,22
MAIS OU IGUAL A 6	3078	30,78	10000	100

TOT_SEG_AUTO (Total de meses com seguro auto (de 1 a 9 meses)):

TOT_SEG_AUTO_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	419	4,19	419	4,19
0	9423	94,23	9842	98,42
MAIS OU IGUAL A 1 MÊS	158	1,58	10000	100

RENDA_MENSAL (Renda mensal do cliente):

RENDA_MENSAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulada
	17	0,17	17	0,17
1 MENOS QUE 500 REAIS	1871	18,71	1888	18,88
2 ENTRE 500 E 1500 REAIS	3843	38,43	5731	57,31
3 ENTRE 1500 E 3000 REAIS	1782	17,82	7513	75,13
4 MAIS QUE 3000 REAIS	2487	24,87	10000	100

Sobre as variáveis contínuas excluiu-se as variáveis com mais de 65% de dados faltantes (sinalizadas em negrito na tabela abaixo).

Tabela 17 – Estatística descritiva para das variáveis contínuas.

Variável	Média	Desvio Padrão	Mínimo	Máximo	N Válido	N Faltante
AVENC_TOTAL_SCR_CONSIG	15835,54	26627,28	0	230882,71	986	9014
AVENC_TOTAL_SCR_CP	14515,64	45611,11	0	1157585,69	1087	8913
MBB_3M	154,1163944	395,4999364	-6147,76	8468,54	9559	441
SALDO_DISPONIVEL_3M	3015,22	14271,79	0	709546,71	9559	441
VENCD_TOTAL_SCR_CONSIG	1011,16	8109,66	0	163222,04	986	9014
VENCD_TOTAL_SCR_CP	279,3943238	1823,18	0	40337,51	1087	8913
VL_DEB_AUTOMATICO	389,3043045	3734,88	0,11	200116,59	3120	6880
VL_LIMITE_DISPONIVEL_CART_CRED	15214,07	38899,53	0	636985,47	5466	4534
VL_LIMITE_IMPLANTADO_CART_CRED	21324,05	48852,77	0	735000	5466	4534
VL_LIMITE_IMPLANTADO_SM	4524,91	7107,06	0	100000	6782	3218
VL_LIMITE_UTILIZADO_CART_CRED	5092,12	16060,57	-131227,53	325246,32	5466	4534
VL_LIMITE_UTILIZADO_SM	-655,9016844	2127,18	-51748,5	0	6768	3232
VL_SALD_ATIV	5057,62	21205,28	0	715036,59	9806	194
VL_SALD_PASS	12697,5	100809,92	0	4883727,79	9806	194
VL_SALD_POUP	2952,61	15349,11	0	685658,24	9806	194
VL_SALD_PRVD_PRIV	1828,45	30984,24	0	1675067,13	9806	194
VL_SALDO_DEVEDOR_TOTAL	6294,94	18055,34	-13072,45	419175,98	5466	4534
VL_SM_CAPTACAO_12	15814,88	112779,65	1	4811773,26	7792	2208
VL_SM_CRED_PESSOAL_12	1555,41	7744,18	0	243912,96	9544	456
VL_TARIFA_COBRADA_12	26,9498767	44,6257627	-114,65	1283,86	5272	4728
VL_TOTAL_CDB_T0	95557,28	311945,89	104,07	4724536,51	407	9593
VL_TOTAL_INVESTIMENTO_T0	23953,85	136505,96	0	4863665,82	3889	6111
VL_TOTL_REND	2661,96	6615,39	0	371476,52	9853	147
VL_TRANS_INTERNACIONAL	77,1608971	1327,85	0	91193,2	8293	1707
VL_TRANS_NACIONAL	414,8774834	1211,92	0	29869,87	8293	1707

Tabela 18 – Percentis das variáveis contínuas.

Variável	5º Percentil	Primeiro Quartil	Mediana	Terceiro Quatil	95º Percentil
AVENC_TOTAL_SCR_CONSIG	0	3938,62	7745,39	16601,41	54496,79
AVENC_TOTAL_SCR_CP	0	1859,07	5233,44	14300,34	51036,93
MBB_3M	0	8,2807667	40,0002	137,0933333	676,2205333
SALDO_DISPONIVEL_3M	0	12,53	135,6666667	975,5366667	13879,46
VENCD_TOTAL_SCR_CONSIG	0	0	0	0	2352,65
VENCD_TOTAL_SCR_CP	0	0	0	0	1042,34
VL_DEB_AUTOMATICO	15	51,165	140,37	358,995	1179,17
VL_LIMITE_DISPONIVEL_CART_CRED	0	590,44	2603,62	11866,7	72786,64
VL_LIMITE_IMPLANTADO_CART_CRED	500	1500	5000	19000	98600
VL_LIMITE_IMPLANTADO_SM	200	750	1850	5200	18500
VL_LIMITE_UTILIZADO_CART_CRED	-328,02	11	853,05	3959,72	22690,04
VL_LIMITE_UTILIZADO_SM	-3255,39	-399,525	-29,36	0	0
VL_SALD_ATIV	0	0	24,675	1943,49	23611,34
VL_SALD_PASS	0	0,97	152,975	1786,12	39868,09
VL_SALD_POUP	0	0	1,645	527,84	13036,97
VL_SALD_PRVD_PRIV	0	0	0	0	0
VL_SALDO_DEVEDOR_TOTAL	0	173,32	1330,57	4927,27	26533,9
VL_SM_CAPTACAO_12	3,55	48,425	289,045	2490,05	54135,31
VL_SM_CRED_PESSOAL_12	0	0	0	0	7448,22
VL_TARIFA_COBRADA_12	2,5	7,05	19	37,5	76
VL_TOTAL_CDB_T0	1029,66	7340,3	29515,95	74633,77	366205,13
VL_TOTAL_INVESTIMENTO_T0	0	0,13	102,72	7005,08	102175,22
VL_TOTL_REND	0	595,22	1200	2931,62	10000
VL_TRANS_INTERNACIONAL	0	0	0	0	0
VL_TRANS_NACIONAL	0	0	0	284,87	2180,75

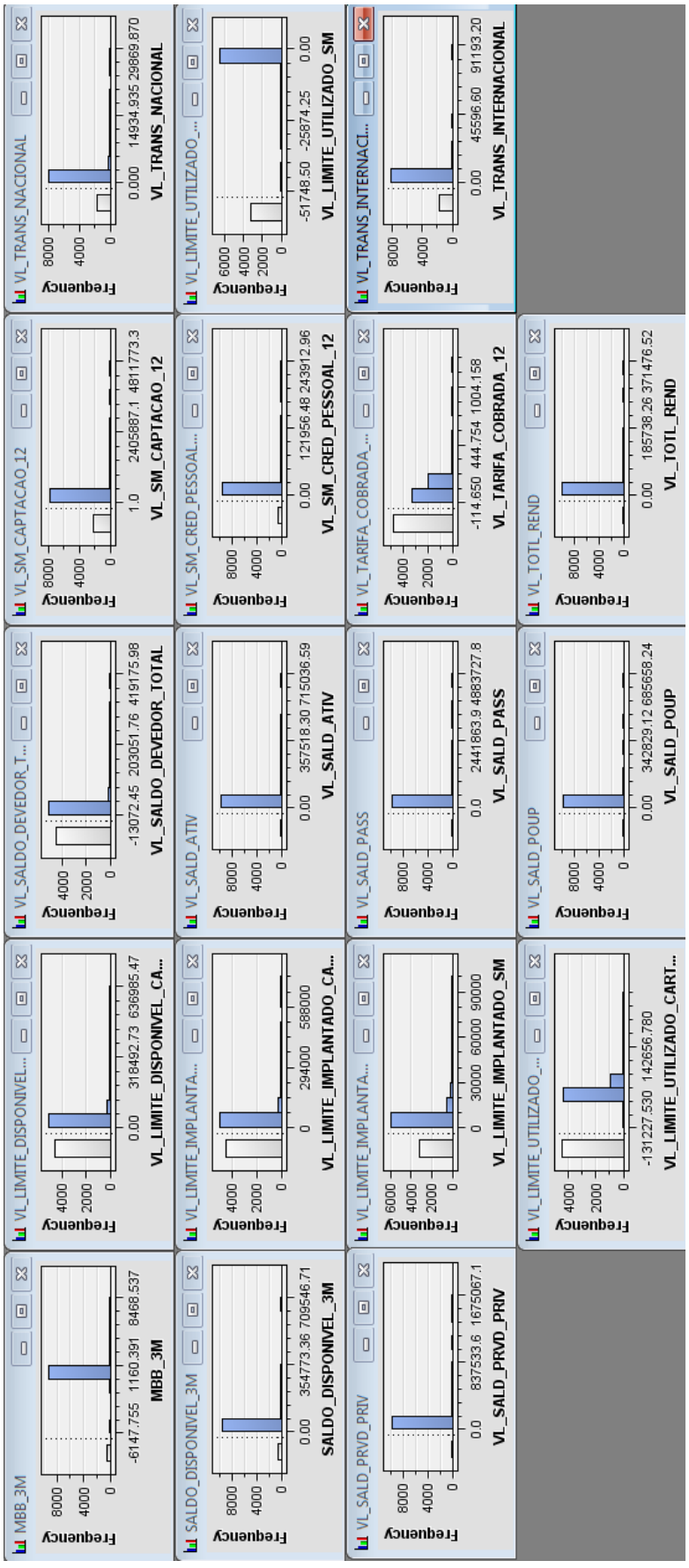

Figura 26 – Histograma para as variáveis contínuas

Após a imputação de valores pelo método de árvore de decisão e após a transformação logarítma das variáveis contínuas, obtevê-se os resultados abaixo. Variáveis com inicial “IMP_” são variáveis que tiveram valores inseridos pelo método de árvore, já as variáveis iniciadas com “LOG_” tiveram o logarítmo aplicado.

IMP_ESTADO_CIVIL (Estado civil do cliente):

IMP_ESTADO_CIVIL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
DIVORCIADO / VIÚVO	1453	14,53	1453	14,53
SOLTEIRO	8547	85,47	10000	100

IMP_ESCOLARIDADE (Escolaridade do cliente):

IMP_ESCOLARIDADE_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
ENSINO MÉDIO	3940	39,4	3940	39,4
SEM ESCOLARIDADE / ENSINO FUNDAMENTAL	2060	20,6	6000	60
SUPERIOR	4000	40	10000	100

IMP_PERFIL_HIST (Perfil do cliente dentro do banco):

IMP_PERFIL_HIST	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
INVESTIDOR	4325	43,25	4325	43,25
NEUTRO	1611	16,11	5936	59,36
TOMADOR	4064	40,64	10000	100

IMP_RISCO (Nível de risco de crédito do cliente):

IMP_RISCO	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
ALTO	1186	11,86	1186	11,86
BAIXO	7324	73,24	8510	85,1
MÉDIO	1490	14,9	10000	100

IMP_SG_UF (Sigla da unidade da federação em que o cliente abriu conta):

IMP_SG_UF_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
OUTRAS	1423	14,23	1423	14,23
SUDESTE	7424	74,24	8847	88,47
SUL	1153	11,53	10000	100

IMP_IDADE (Idade do cliente):

IMP_IDADE_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
1 MENOR OU IGUAL A 25 ANOS	1677	16,77	1677	16,77
2 ENTRE 26 E 35 ANOS	2773	27,73	4450	44,5
3 ENTRE 36 E 50 ANOS	2819	28,19	7269	72,69
4 MAIOR OU IGUAL A 51 ANOS	2731	27,31	10000	100

IMP_QT_CDC_LEAS (Quantidade de CDC (0 - não tem outro CDC, 1 - tem outro CDC)):

IMP_QT_CDC_LEAS	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	9802	98,02	9802	98,02
1	198	1,98	10000	100

IMP_QT_CHEQUE_COMPENSADO (Quantidade de cheques compensados):

IMP_QT_CHEQUE_COMPENSADO_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	7158	71,58	7158	71,58
DE 1 A 5	2034	20,34	9192	91,92
MAIS OU IGUAL A 6	808	8,08	10000	100

IMP_QT_COMPRA_VISA (Quantidade de compras realizadas com Visa):

IMP_QT_COMPRA_VISA_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	6208	62,08	6208	62,08
DE 1 A 5	1815	18,15	8023	80,23
MAIS OU IGUAL A 6	1977	19,77	10000	100

IMP_QT_TRANS_INTERNACIONAL (Quantidade de transações internacionais):

IMP_QT_TRANS_INTERNACIONAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	9666	96,66	9666	96,66
MAIS OU IGUAL A 1	334	3,34	10000	100

IMP_QT_TRANS_NACIONAL (Quantidade de transações nacionais):

IMP_QT_TRANS_NACIONAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	6372	63,72	6372	63,72
DE 1 A 5	1573	15,73	7945	79,45
MAIS OU IGUAL A 6	2055	20,55	10000	100

IMP_QTD_ACESSOS_ATM_MES (Quantidade de acessos ao ATM (Automatic Teller Machine, mais conhecido como caixa eletrônico)):

IMP_QTD_ACESSOS_ATM_MES_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
DE 1 A 5	8321	83,21	8321	83,21
MAIS OU IGUAL A 6	1679	16,79	10000	100

IMP_QTDE_PRODUTOS_PF_12 (Quantidade de produtos pessoa física):

IMP_QTDE_PRODUTOS_PF_12_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
DE 1 A 5	6904	69,04	6904	69,04
MAIS OU IGUAL A 6	3096	30,96	10000	100

IMP_TOT_SEG_AUTO (Total de meses com seguro auto (de 1 a 9 meses)):

IMP_TOT_SEG_AUTO_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
0	9841	98,41	9841	98,41
MAIS OU IGUAL A 1 MÊS	159	1,59	10000	100

IMP_RENDA_MENSAL (Renda mensal do cliente):

IMP_RENDA_MENSAL_A	Frequência	Percentual	Frequência Acumulada	Percentual Acumulado
1 MENOS QUE 500 REAIS	1886	18,86	1886	18,86
2 ENTRE 500 E 1500 REAIS	3844	38,44	5730	57,3
3 ENTRE 1500 E 3000 REAIS	1782	17,82	7512	75,12
4 MAIS QUE 3000 REAIS	2488	24,88	10000	100

Nas variáveis contínuas aplicou-se o logarítmo, como pode-se ver nas distribuições da Figura a seguir, dispostos na mesma ordem do anterior:

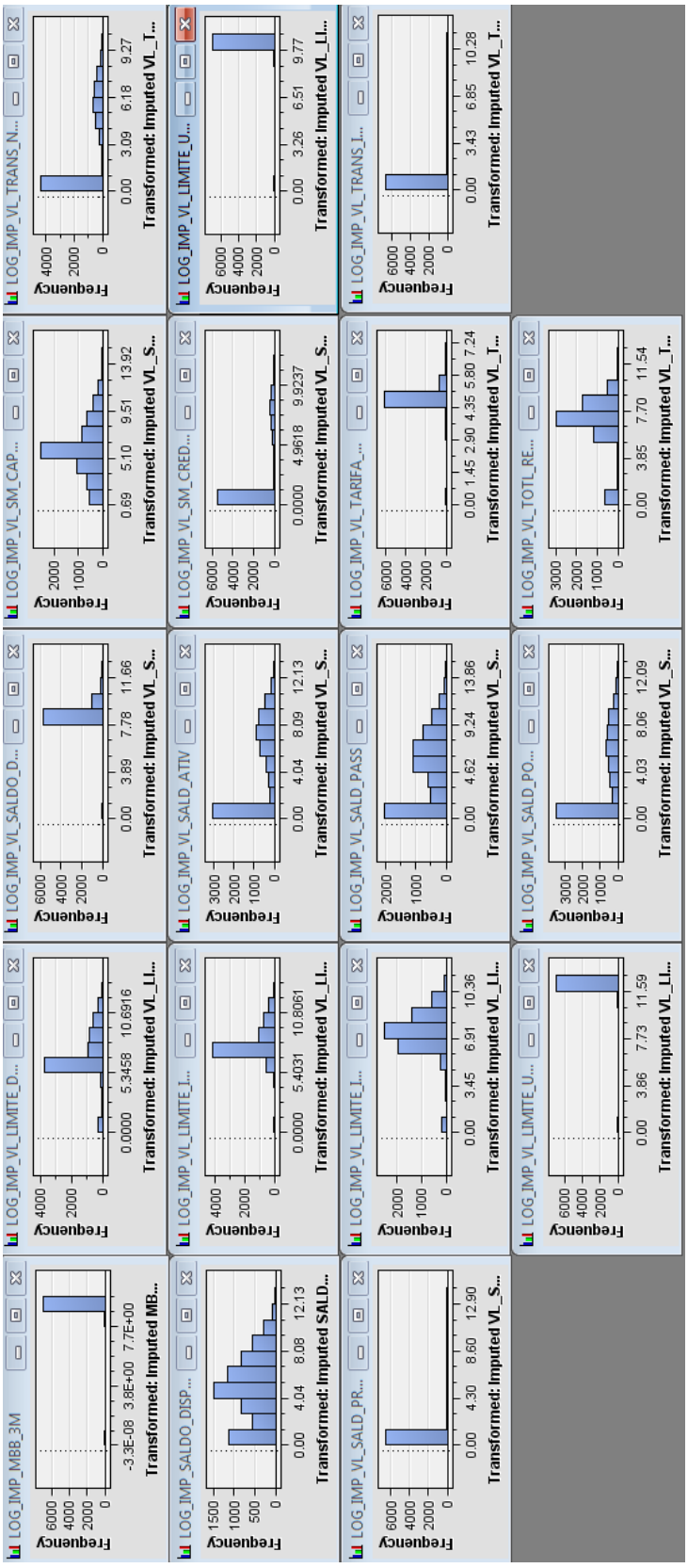

Figura 27 – Histograma para as variáveis contínuas transformadas

APÊNDICE C

A seguir está programada as regras de decisão para o modelo de árvore de decisão exposto no capítulo de resultados.

```
*------------------------------------------------------------*
 Node = 7
*------------------------------------------------------------*
if Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 1
then
 Tree Node Identifier   = 7
 Number of Observations = 99
 Predicted: FLAG_RESPOSTA=0  = 0.42
 Predicted: FLAG_RESPOSTA=1  = 0.58

*------------------------------------------------------------*
 Node = 9
*------------------------------------------------------------*
if Transformed: Imputed VL_LIMITE_IMPLANTADO_SM < 7.04795 or MISSING
AND Imputed RENDA_MENSAL_A >= 3 ENTRE 1500 E 3000 REAIS AND Imputed
RENDA_MENSAL_A <= 3 ENTRE 1500 E 3000 REAIS
then
 Tree Node Identifier   = 9
 Number of Observations = 108
 Predicted: FLAG_RESPOSTA=0  = 0.85
 Predicted: FLAG_RESPOSTA=1  = 0.15

*------------------------------------------------------------*
 Node = 10
*------------------------------------------------------------*
if Transformed: Imputed VL_LIMITE_IMPLANTADO_SM >= 7.04795
AND Imputed RENDA_MENSAL_A <= 3 ENTRE 1500 E 3000 REAIS or MISSING
AND Imputed QT_CDC_LEAS IS ONE OF: 1
then
 Tree Node Identifier   = 10
 Number of Observations = 44
 Predicted: FLAG_RESPOSTA=0  = 0.48
 Predicted: FLAG_RESPOSTA=1  = 0.52

*------------------------------------------------------------*
 Node = 16
*------------------------------------------------------------*
if Transformed: Imputed VL_LIMITE_IMPLANTADO_SM < 7.04795 or MISSING
AND Imputed: TOT_SEG_AUTO_A <= 0 or MISSING
AND Imputed RENDA_MENSAL_A <= 2 ENTRE 500 E 1500 REAIS or MISSING
then
 Tree Node Identifier   = 16
 Number of Observations = 3012
 Predicted: FLAG_RESPOSTA=0  = 0.99
 Predicted: FLAG_RESPOSTA=1  = 0.01



*------------------------------------------------------------*
 Node = 17
*------------------------------------------------------------*
```

if Transformed: Imputed VL_LIMITE_IMPLANTADO_SM < 7.04795 or MISSING
AND Imputed: TOT_SEG_AUTO_A >= MAIS OU IGUAL A 1 MÊS
AND Imputed RENDA_MENSAL_A <= 2 ENTRE 500 E 1500 REAIS or MISSING
then
 Tree Node Identifier = 17
 Number of Observations = 5
 Predicted: FLAG_RESPOSTA=0 = 0.60
 Predicted: FLAG_RESPOSTA=1 = 0.40

--
 Node = 21
--
if Transformed: Imputed VL_TRANS_NACIONAL >= 4.91151
AND Transformed: Imputed VL_LIMITE_IMPLANTADO_SM >= 7.04795
AND Imputed RENDA_MENSAL_A <= 3 ENTRE 1500 E 3000 REAIS or MISSING
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier = 21
 Number of Observations = 630
 Predicted: FLAG_RESPOSTA=0 = 0.83
 Predicted: FLAG_RESPOSTA=1 = 0.17

--
 Node = 22
--
if Imputed: QT_TRANS_NACIONAL_A <= 0
AND Imputed: QT_CHEQUE_COMPENSADO_A <= 0 or MISSING
AND Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier = 22
 Number of Observations = 427
 Predicted: FLAG_RESPOSTA=0 = 0.91
 Predicted: FLAG_RESPOSTA=1 = 0.09

--
 Node = 24
--
if Transformed: Imputed VL_SALD_ATIV < 6.75507 or MISSING
AND Imputed: QT_TRANS_NACIONAL_A >= DE 1 A 5 or MISSING
AND Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier = 24
 Number of Observations = 483
 Predicted: FLAG_RESPOSTA=0 = 0.78
 Predicted: FLAG_RESPOSTA=1 = 0.22

--
 Node = 25
--
if Transformed: Imputed VL_SALD_ATIV >= 6.75507
AND Imputed: QT_TRANS_NACIONAL_A >= DE 1 A 5 or MISSING
AND Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier = 25
 Number of Observations = 452
 Predicted: FLAG_RESPOSTA=0 = 0.63

```
 Predicted: FLAG_RESPOSTA=1  = 0.37

*------------------------------------------------------------*
 Node = 30
*------------------------------------------------------------*
if Transformed: Imputed VL_TRANS_NACIONAL < 4.91151 or MISSING
AND Transformed: Imputed VL_LIMITE_IMPLANTADO_SM >= 7.04795
AND Imputed: IDADE_A <= 3 ENTRE 36 E 50 ANOS or MISSING
AND Imputed RENDA_MENSAL_A <= 3 ENTRE 1500 E 3000 REAIS or MISSING
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier   = 30
 Number of Observations = 951
 Predicted: FLAG_RESPOSTA=0  = 0.91
 Predicted: FLAG_RESPOSTA=1  = 0.09

*------------------------------------------------------------*
 Node = 31
*------------------------------------------------------------*
if Transformed: Imputed VL_TRANS_NACIONAL < 4.91151 or MISSING
AND Transformed: Imputed VL_LIMITE_IMPLANTADO_SM >= 7.04795
AND Imputed: IDADE_A >= 4 MAIOR OU IGUAL A 51 ANOS
AND Imputed RENDA_MENSAL_A <= 3 ENTRE 1500 E 3000 REAIS or MISSING
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier   = 31
 Number of Observations = 537
 Predicted: FLAG_RESPOSTA=0  = 0.97
 Predicted: FLAG_RESPOSTA=1  = 0.03

*------------------------------------------------------------*
 Node = 36
*------------------------------------------------------------*
if Transformed: Imputed VL_SALD_ATIV < 5.59928 or MISSING
AND Imputed: QT_TRANS_NACIONAL_A <= 0
AND Imputed: QT_CHEQUE_COMPENSADO_A >= DE 1 A 5
AND Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier   = 36
 Number of Observations = 128
 Predicted: FLAG_RESPOSTA=0  = 0.88
 Predicted: FLAG_RESPOSTA=1  = 0.13

*------------------------------------------------------------*
 Node = 37
*------------------------------------------------------------*
if Transformed: Imputed VL_SALD_ATIV >= 5.59928
AND Imputed: QT_TRANS_NACIONAL_A <= 0
AND Imputed: QT_CHEQUE_COMPENSADO_A >= DE 1 A 5
AND Imputed RENDA_MENSAL_A >= 4 MAIS QUE 3000 REAIS
AND Imputed QT_CDC_LEAS IS ONE OF: 0 or MISSING
then
 Tree Node Identifier   = 37
 Number of Observations = 122
 Predicted: FLAG_RESPOSTA=0  = 0.69
 Predicted: FLAG_RESPOSTA=1  = 0.31
```

APÊNDICE D

D.1 Conhecendo o SAS *Enterprise Miner*

O SAS *Enterprise Miner* possui uma interface de programação visual que facilita a construção de modelos de *Data Mining* para o processo de descoberta de conhecimento. A ferramenta oferece ricas facilidades para a exploração e manipulação de dados, além de várias técnicas de modelagem e recursos gráficos, para a visualização de dados. As operações são representadas em um diagrama, no qual cada nó (*nodes*) representa um um passo na análise, conforme vê-se na Figura 28.

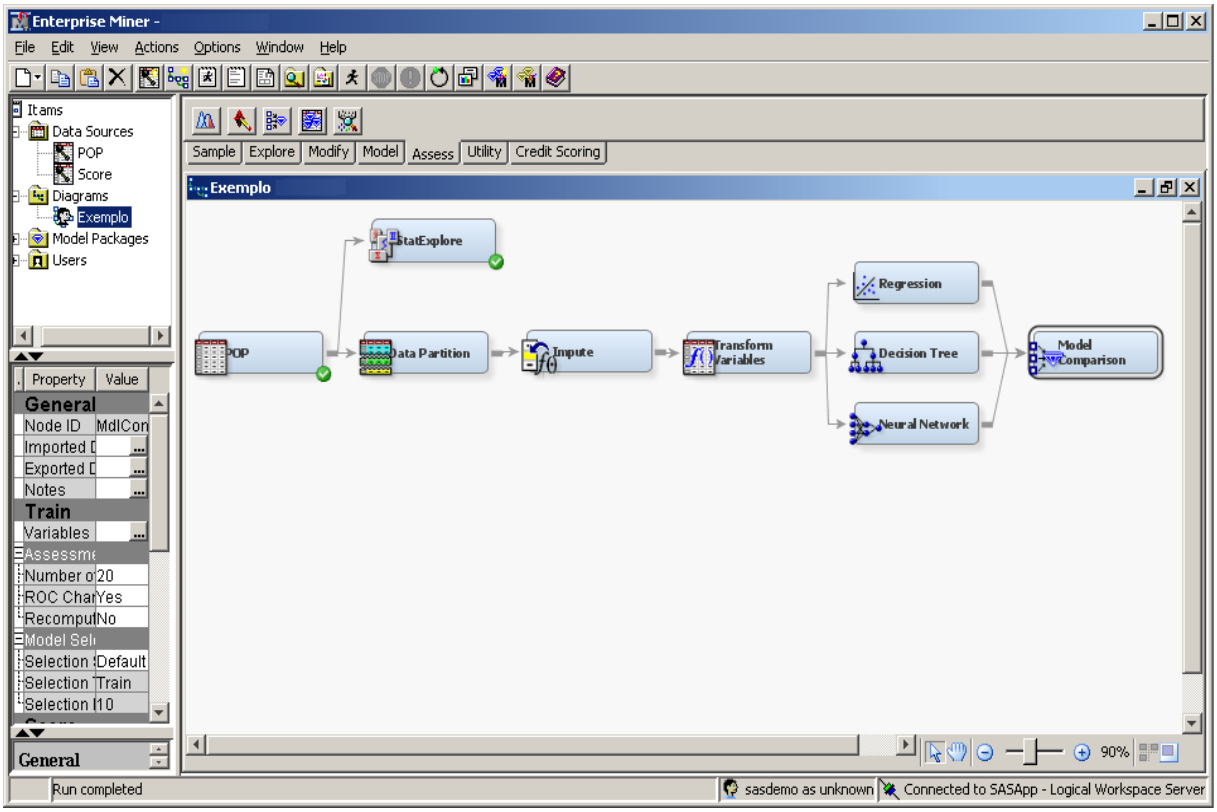

Figura 28 - Interface do SAS *Enterprise Miner*

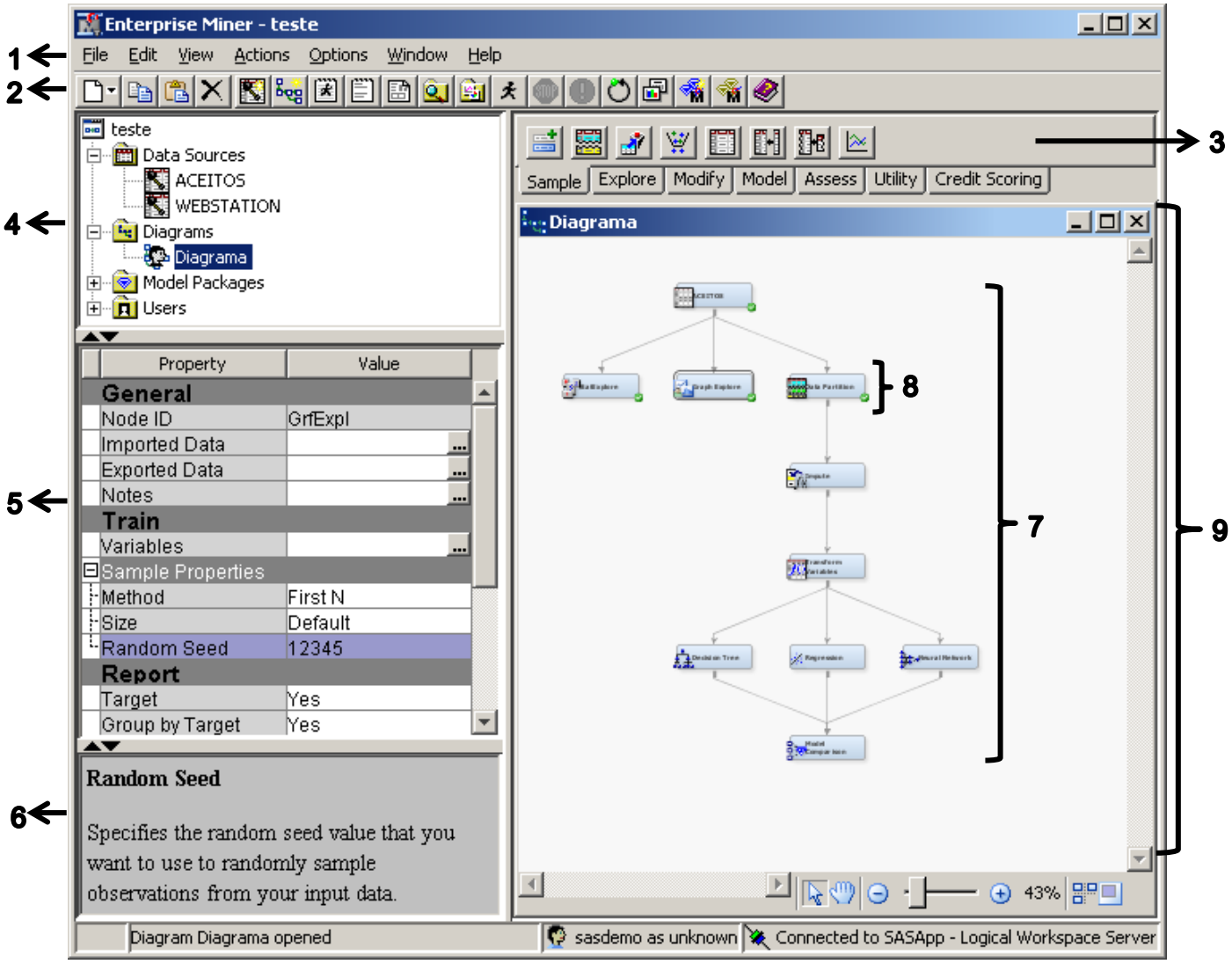

Figura 29 - Interface do SAS *Enterprise Miner*

Já na Figura 29 expõe-se um *tour* pelo software, onde cada número será explicado a seguir.

1. Menu inicial

2. Os botões de atalho permitem desenvolver tarefas rápidamente, como por exemplo, executar um nó.

3. A barra de ferramentas permite acessar as ferramentas, é dividida em abas de acordo com a arquitetura SEMMA, que será explicada a seguir.

4. O painel do Projeto permite visualizar e gerenciar os *data sources*, diagramas, resultados e usuários do projeto.

5. O painel de propriedades permite exibir e editar as configurações dos *data sources*, diagramas, nós, resultados e os usuários.

6. O painel de ajuda exibe uma breve descrição do objeto selecionado no painel de propriedades.

7. A área de trabalho do diagrama contém um ou mais fluxos. Um fluxo começa com um *data source* e sequencialmente aplica-se ferramentas do SAS *Enterprise Miner* (que são chamados de nós dentro do diagrama) para completar o objetivo analítico.

8. Um fluxo contém vários nós. Os nós são ferramentas do SAS *Enterprise Miner*, que são conectados por setas para mostrar a direção do fluxo de informações em uma análise.

9. A área de trabalho do diagrama permite criação de uma sequência gráfica de todos os passos utilizados para análise de dados.

O software SAS *Enterprise Miner* é um produto que contém uma série de ferramentas úteis para suportar todo o processo de *Data Mining*. Tais ferramentas estão organizadas de acordo com o processo SEMMA, ou seja, de acordo com 5 estágios, que serão listados a seguir.

D.2 Principal Processo SAS para Mineração de Dados

A barra de ferramentas do SAS *Enterprise Miner* é organizada de acordo com o processo SAS para mineração de dados, conhecido como SEMMA. A sigla SEMMA - amostrar, explorar, modificar, modelar e avaliar - se refere ao processo principal da mineração de dados. Antes de examinar cada fase da SEMMA é importante salientar que a SEMMA não é uma metodologia de mineração de dados, mas sim uma organização lógica do conjunto de ferramentas do SAS *Enterprise Miner* que realizam tarefas essenciais na mineração de dados.

Enterprise Miner pode ser usado como parte de qualquer metodologia iterativa de mineração de dados adotada. Obviamente que medidas como a formulação do problema de negócio e a montagem da fonte de dados com qualidade são essenciais para o êxito global de qualquer projeto de mineração de dados.

Seguindo esse raciocínio, tem-se que o processo de *Data Mining* pode seguir os passos expostos na Figura 30. Note que o processo SEMMA faz parte do processo, momento em que o SAS *Enterprise Miner* é ativo.

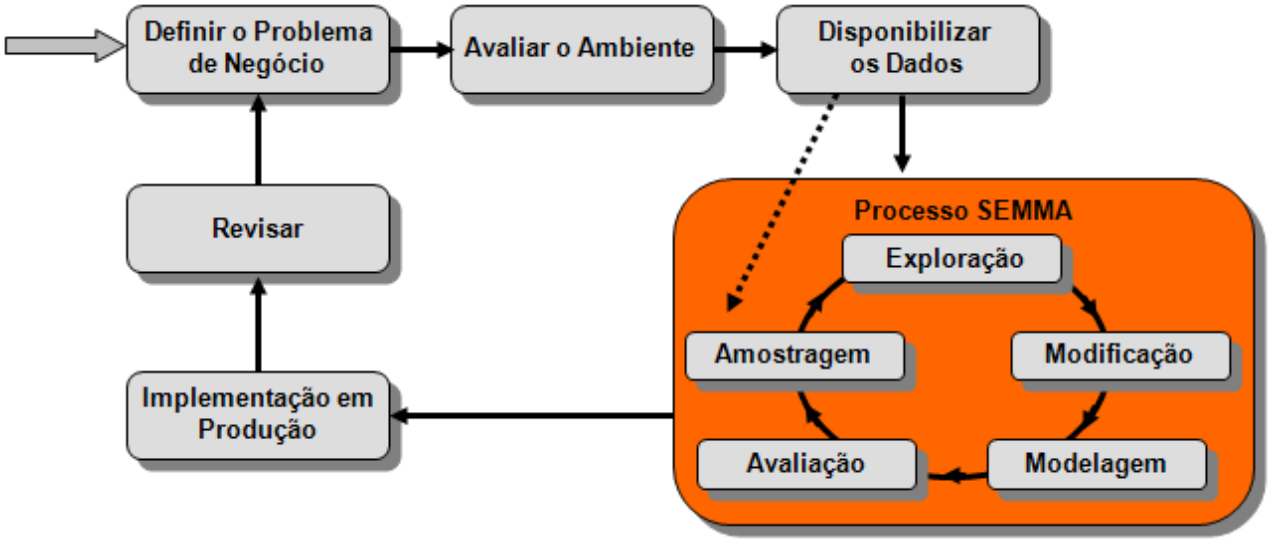

Figura 30 - Principal Processo SAS para Mineração de Dados no SAS *Enterprise Miner*

D.2.1 Arquitetura SEMMA

As etapas do processo SEMMA estão focadas nos aspectos de desenvolvimento do modelo de mineração de dados:

D.2.1.1 *SAMPLE*

Realizar uma amostra (opcional) dos dados, extraindo uma parte de um grande conjunto de dados. Esta amostra deve ser grande o suficiente para conter as informações significativas e também pequena o suficiente para processar, conforme a capcidade do *hardware*. Mineração de uma amostra representativa, em vez de todo o volume de dados reduz o tempo de processamento necessário para obter informações cruciais ao negócio. Se os padrões gerais aparecem nos dados como um todo, estes serão detectáveis em uma amostra representativa. Se um nicho é tão pequeno que não é representado em uma amostra e, ainda assim é tão importante que influencia o todo, ele pode ser descoberto por meio de métodos de síntese. É

importante, também, a criação de conjuntos de dados particionados com o nó de partição de dados:

Treinamento - base utilizada para a montagem do modelo.

Validação - base utilizada para a avaliação e para apontar *overfitting* de modelo.

Teste - base usada para obter uma avaliação honesta de quão bem o modelo generaliza.

D.2.1.2 *EXPLORE*

Explorar os dados a fim de encontrar tendências e/ou anomalias não previstas, para obter conhecimento e idéias. O passo de exploração ajuda a aperfeiçoar o processo de descoberta. Se a exploração visual não revelar tendências claras, pode-se explorar os dados por meio de técnicas estatísticas, incluindo a análise fatorial, análise de correspondência e de cluster. Por exemplo, no processo de mineração de dados para uma campanha de mala direta, o agrupamento pode revelar grupos de clientes com diferentes padrões. Conhecer esses padrões cria oportunidades para *mailings* personalizados ou promoções específicas.

D.2.1.3 *MODIFY*

Modificar os dados, criando, selecionando e transformando as variáveis para o foco do processo de seleção do modelo. Baseado nas descobertas obtidas na fase de exploração, pode ser necessário manipular os dados para incluir informações como o agrupamento de clientes e subgrupos significativos, ou de introduzir novas variáveis. Pode-se também notar a necessidade de tratar *outliers* ou reduzir o número de variáveis, a fim de restringi-las as mais importantes. Mineração de dados é um processo dinâmico, interativo, pode-se atualizar os métodos de mineração de dados ou modelos, quando novas informações estiverem disponíveis.

D.2.1.4 *MODEL*

Modelar os dados a partir de técnicas de modelagem em mineração de dados. No SAS *Enterprise Miner* tem-se, por exemplo: redes neurais, árvore de decisão, modelos logísticos e outros modelos estatísticos - como a análise de séries temporais, raciocínio baseado em memória e de componentes principais. Cada técnica tem seu ponto forte e é apropriado dentro de situações específicas de mineração de dados, dependendo dos dados. Por exemplo: redes neurais são muito boas no ajuste de alta complexidade de relações não lineares.

D.2.1.5 *ASSESS*

Avaliar os dados, avaliar a utilidade e confiabilidade dos resultados do processo de mineração de dados e entender como ele executa. Uma forma comum de avaliar um modelo é aplicá-lo a uma parte do conjunto de dados, ainda não utilizado durante a fase de amostragem. Se o modelo for válido, ele deve trabalhar para esta amostra reservada, bem como para a amostra utilizada para construir o modelo. Da mesma forma, pode-se testar o modelo com os dados conhecidos. Por exemplo, sabe-se que os clientes em um arquivo tinham altas taxas de retenção e o modelo prevê a retenção, pode-se verificar se o modelo seleciona esses clientes com precisão. Além disso, as aplicações práticas do modelo, tais como expedições parciais em uma campanha de mala direta, ajuda a provar sua validade.

Ao avaliar os resultados obtidos em cada etapa do processo SEMMA, pode-se observar novas questões a partir dos resultados anteriores e assim, proceder de volta para a fase de exploração para o refinamento adicional dos dados.

Depois de ter desenvolvido o modelo campeão usando a abordagem SEMMA de mineração, o próximo passo é a implementação do modelo em novos clientes (indivíduos), ou novas bases. A implantação do modelo é o resultado final da mineração de dados. O SAS *Enterprise Miner* automatiza a fase de implantação, fornecendo o código de escoragem em SAS, além do código em C, Java e PMML.

D.3 Arquitetura e configuração do SAS *Enterprise Miner*

O SAS *Enterprise Miner* é organizado em torno de uma arquitetura *client/server*. Isso significa que o SAS *Enterprise Miner* Client é apenas uma parte de um conjunto maior de programas. O SAS *Enterprise Miner* Client é simplesmente uma janela de interface feita em Java. O trabalho de análise é feita por um software conhecido como SAS *Foundation*, que é outro nome para a linguagem e procedimentos SAS. O SAS *Foundation* por sua vez é apoiado por outros softwares conhecidos como Servidor de Metadados SAS. O SAS *Metadata Server* monitora o acesso a dados e informações de arquitetura do sistema.

Existem várias maneiras de configurar o SAS *Enterprise Miner*: Na configuração de estação de trabalho pessoal (*Personal Workstation*), o SAS *Enterprise Miner Client*, SAS *Foundation*, e SAS *Metadata Server* residem em um único computador central. Os componentes se comunicam por meio de uma tecnologia proprietária chamada SAS IOM, como mostra a Figura 31.

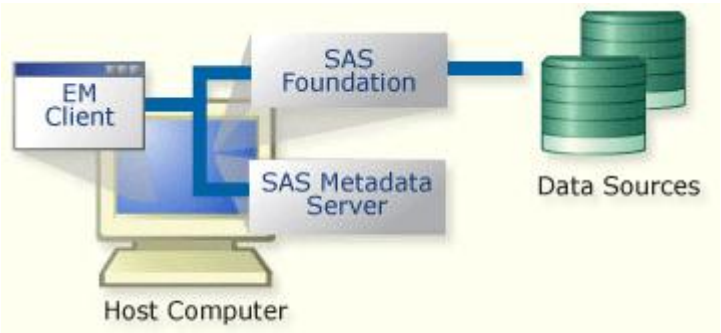

Figura 31 - Interface do SAS *Enterprise Miner*

Na configuração do *Enterprise Client*, a comunicação entre o SAS *Enterprise Miner Client*, o SAS *Foundation Server* e o SAS *Metadata Server* é criado por meio de um componente adicional denominado *Analytics Platform*. Isso permite que vários *Clients* conectem-se a vários servidores SAS *Foundation*. Para executar o SAS *Enterprise Miner* o administrador do sistema SAS deve instalar e configurar esses componentes, geralmente em diversos computadores independentes. Depois que a configuração for estabelecida pouco importa, para o analista, exceto para lembrar que todos os dados são lidos no servidor do SAS *Foundation* e não no PC

físico local. A única coisa que o analista vai ver é a interface do SAS *Enterprise Miner Client*. Na Figura 32 vê-se uma imagem ilustrativa dessa forma de instalação.

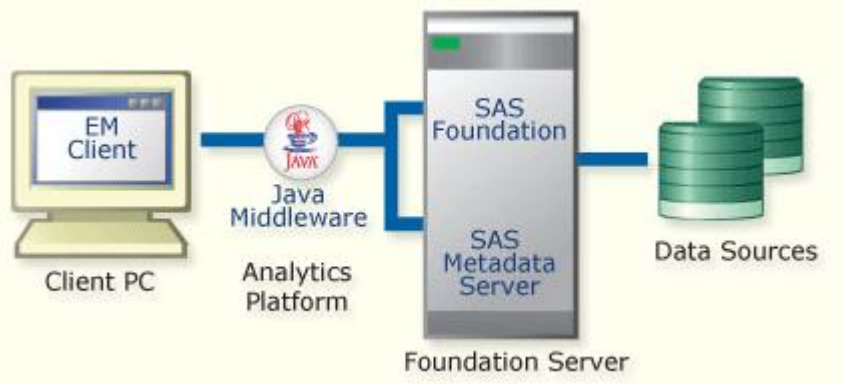

Figura 32 - Interface do SAS *Enterprise Miner*

D.3 Entendendo a Forma de Trabalho do SAS *Enterprise Miner*

No SAS *Enterprise Miner* as análises são organizadas em projetos, diagramas, fluxos e nós. Com auxilio da Figura 33 pode-se entender isso facilmente. O primeiro passo é a criação de um Projeto e é nesse projeto onde serão realizadas todas as análises necessárias. Dentro de um projeto pode-se criar diversos Diagramas, organizando-os da forma necessária.

Dentro de um Diagrama é que se cria os Fluxos. Um Fluxo pode ser composto, por uma base de dados, um particionamento de dados e uma regressão, por exemplo. Cada Fluxo é composto por nós, ou seja, cada passo da análise. Cada nó, como já dito anteriormente, executa uma tarefa.

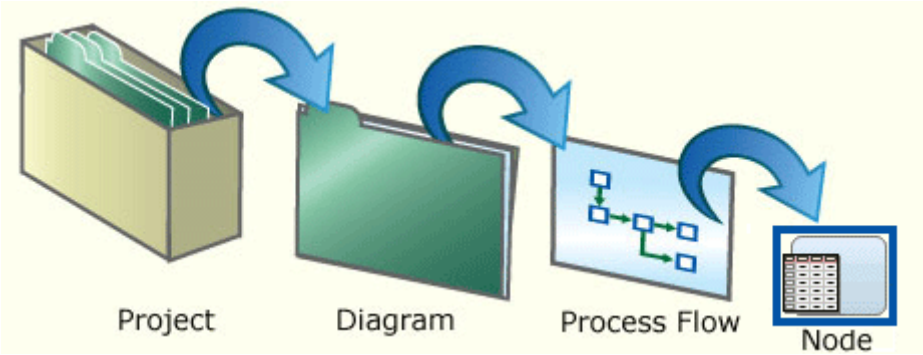

Figura 33 - Forma de organização do SAS *Enterprise Miner*

Por trás desse esquema, existe um espaço físico onde realmente o projeto está salvo. A organização física de um projeto SAS *Enterprise Miner* é mais complicada. Quando um projeto é criado no SAS *Enterprise Miner*, quatro subdiretórios são criados automaticamente dentro do diretório do projeto: *DataSources*, *Reports*, *Workspaces* e *System*. A estrutura do diretório do projeto "teste" (ilustrado na Figura 29) é mostrada na Figura 34.

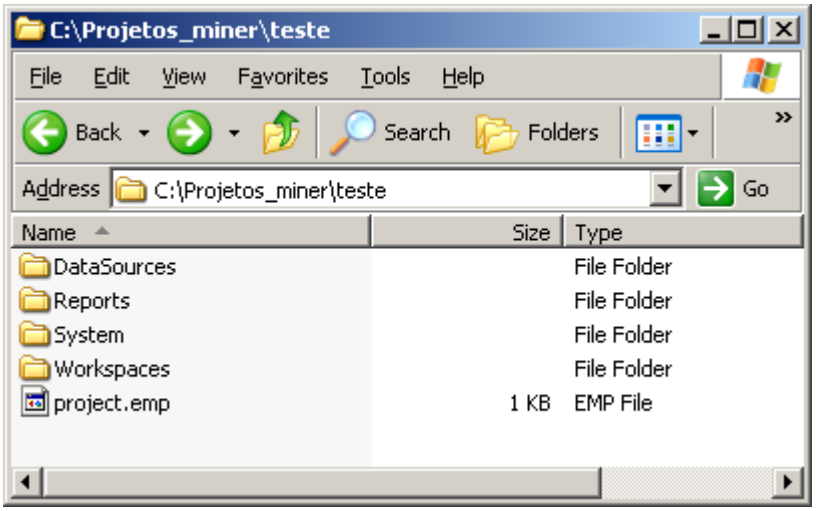

Figura 34 - Forma de organização física do SAS *Enterprise Miner*

Os projetos contêm diagramas, que são o próximo nível da hierarquia da organização do SAS *Enterprise Miner*. Diagramas geralmente dizem respeito a um tema único do projeto. Quando um diagrama é definido, um novo subdiretório é criado no diretório *Workspaces* do projeto correspondente. Cada diagrama é independente e nenhuma informação pode ser passada de um diagrama para o outro. A estrutura do diretório *Workspaces* para o projeto "teste" criado para a Figura 29 é mostrado na Figura 35.

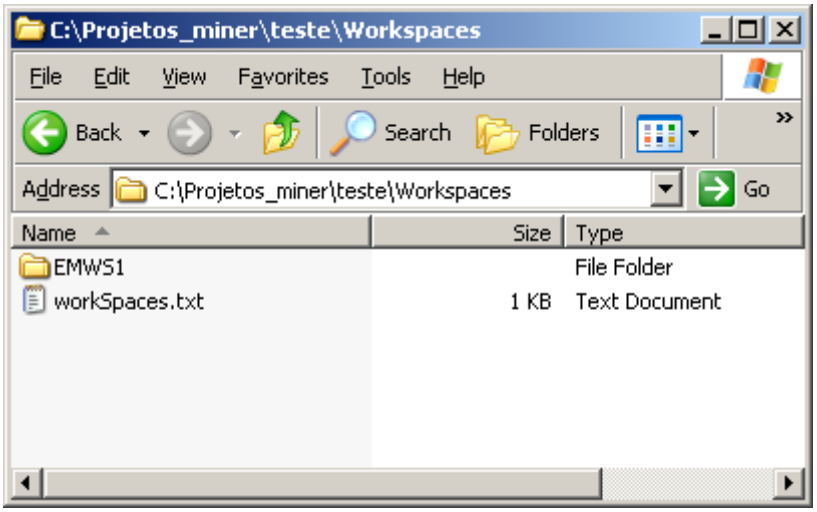

Figura 35 - Estrutura do diretório Workspaces

As análises realizadas no SAS *Enterprise Miner* são desenvolvidas por um fluxo. Um fluxo é uma sequência de nós, conectados por flechas que definem a ordem da análise. A organização do fluxo está contida em um arquivo, EM_DGRAPH, que é armazenado dentro do diretório do diagrama correspondente. Cada nó do diagrama corresponde a um subdiretório separado no diretório desse diagrama. As informações de um fluxo podem ser enviadas para outro, bastando apenas ligá-los pelas flechas. O diretório do diagrama EMWS1 (nome dado para a pasta do diagrama que contém os fluxos e nós) é mostrado a seguir na Figura 36.

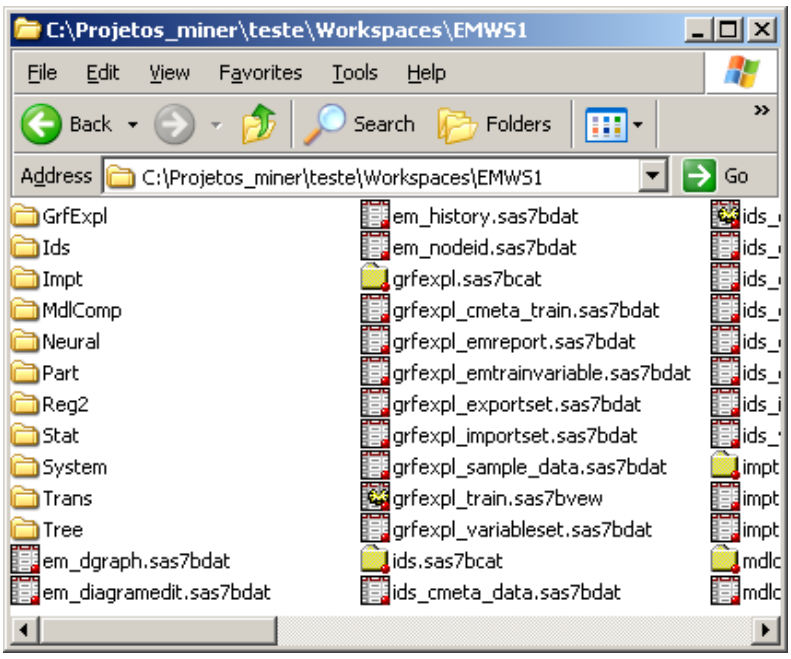

Figura 36 - Estrutura do diretório de um diagrama

Felizmente, a interface do SAS *Enterprise Miner* nos protege dessa complexidade.

D.4 Primeiros Passos

O propósito desta seção é introduzir os passos iniciais a serem dados em qualquer análise de mineração de dados. Como por exemplo, a criação de um projeto, a criação de uma biblioteca e criação de um *Data Source*.

D.4.1 Criação de um Projeto

Após aberto o *Miner* e digitado usuário e senha, o primeiro passo será a criação de um projeto. Como mostrado na Figura 37, clicar-se em *New Project.*

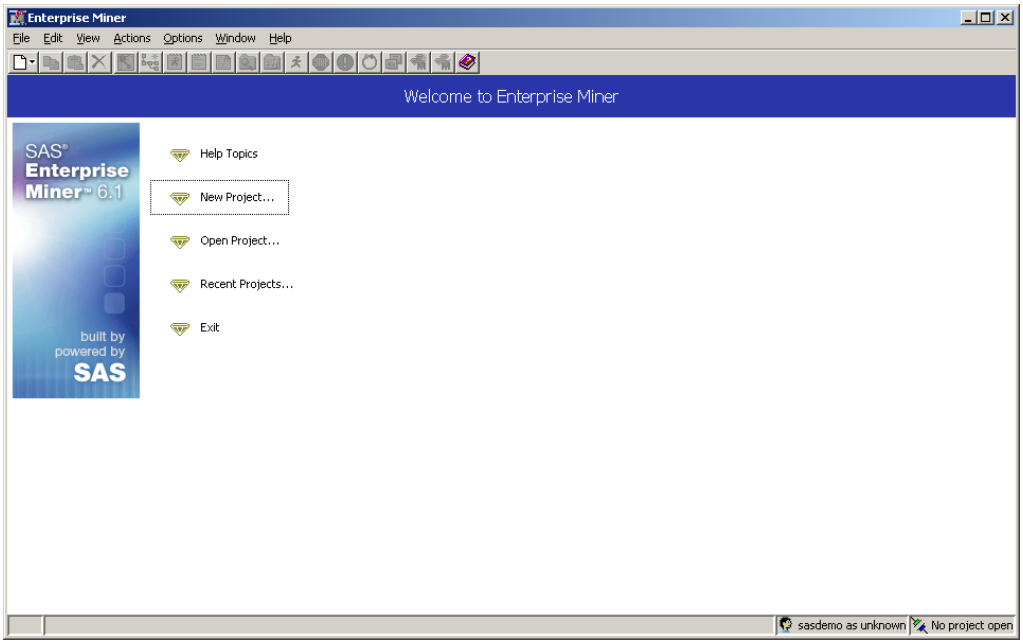

Figura 37 - Inicialização do SAS *Enterprise Miner*

Depois de clicado em *New Project*, segue-se com as solicitações do *Wizard*. A etapa 1 da criação de um projeto é a especificação do SAS *Server* onde salva-se o projeto. Após selecionado, clica-se em "*Avançar*". Na etapa 2 coloca-se um nome para o projeto e especifica-se a pasta, dentro do SAS *Server*, onde o projeto será salvo, como mostra a Figura 38.

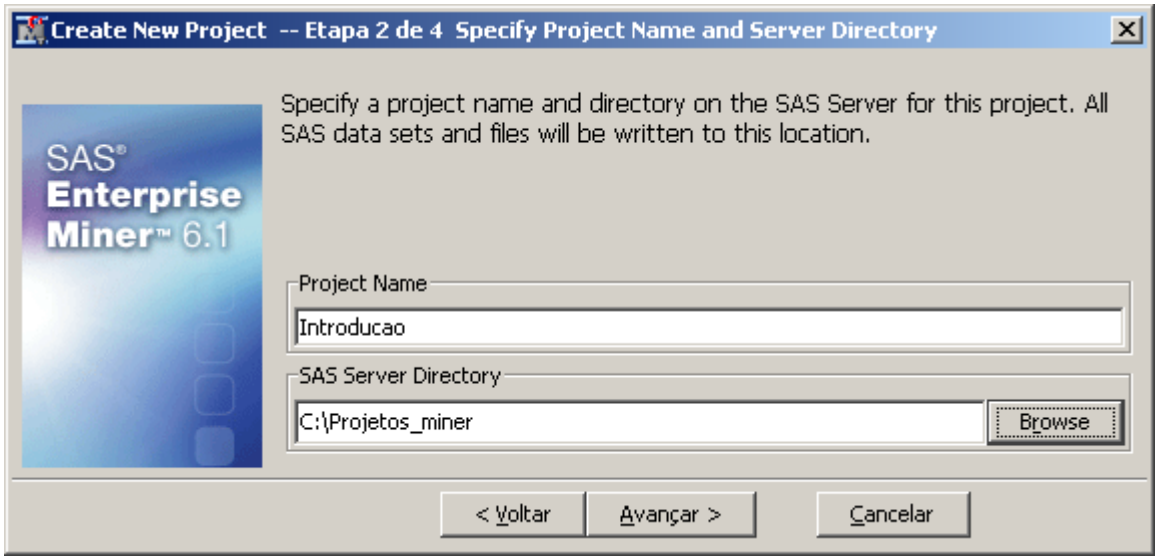

Figura 38 - Etapa 2 na criação de um Projeto

Clicando em *Avançar*, o próximo passo será especificar um folder, como na Figura 39.

Avançar novamente e tem-se a última etapa que é apenas um resumo de todas as informações sobre o novo projeto e então, *Concluir.*

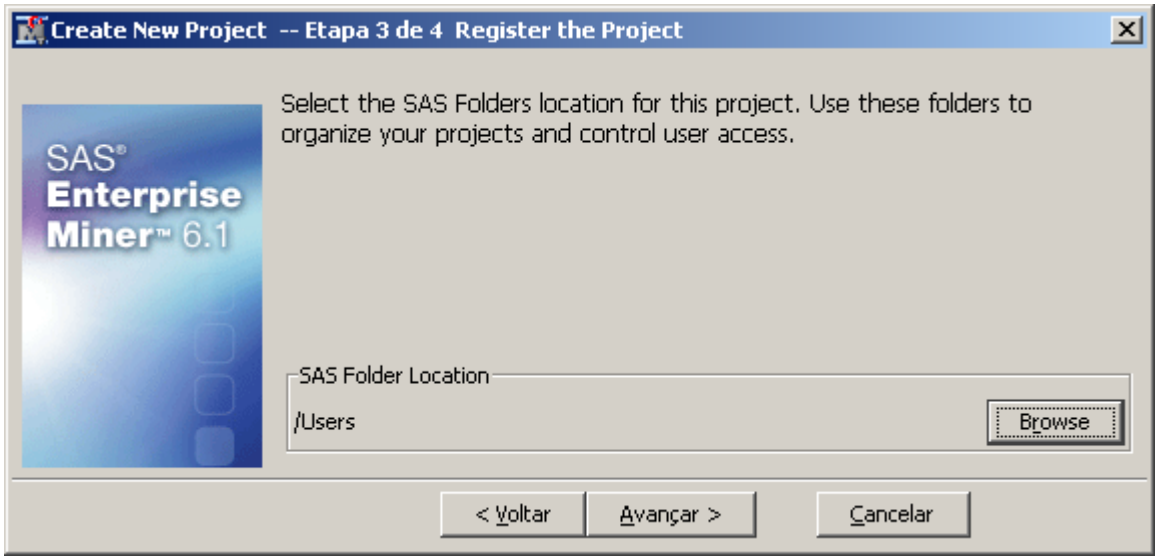

Figura 39 - Etapa 3 na criação de um Projeto

Finalizado a criação do Projeto, tem-se uma tela semelhante a da Figura 40. Observe que nenhuma das funcionalidades está habilitada, pois ainda não existe um diagrama, etapa seguinte à criação de um projeto.

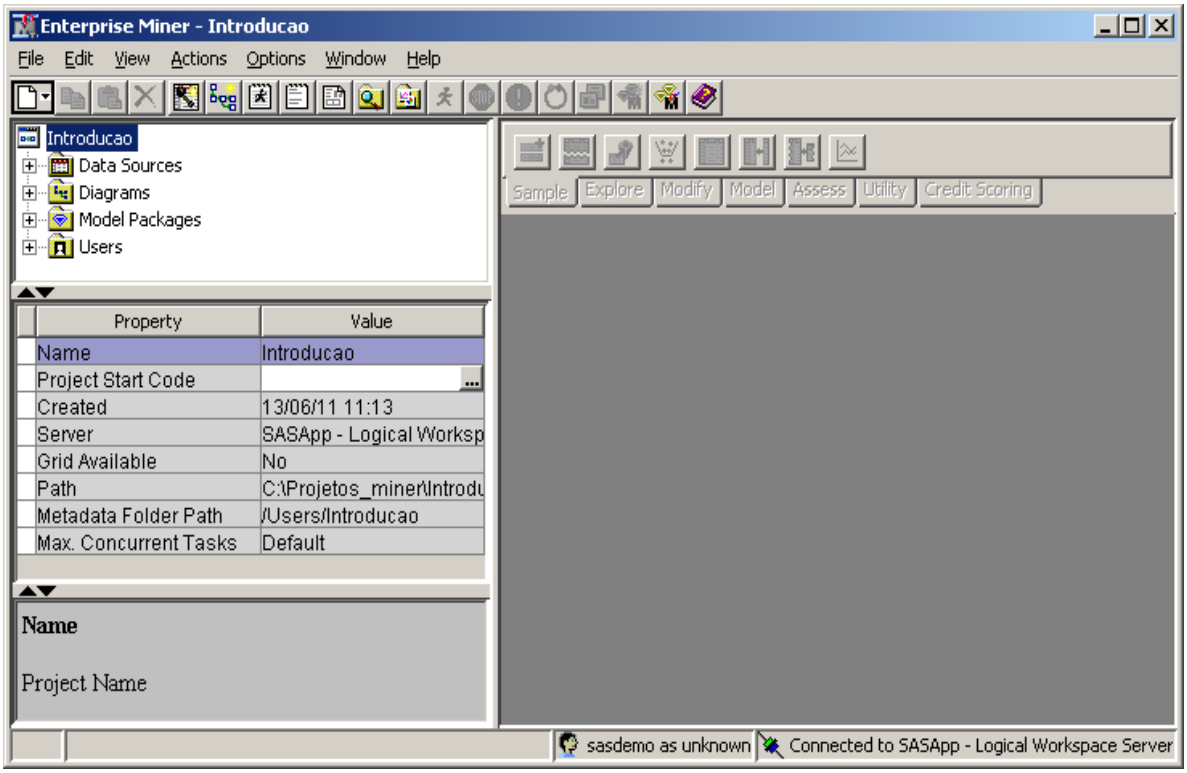

Figura 40 - Visualização do SAS *Enterprise Miner* após a criação de um projeto

D.4.2 Criação de um Diagrama

Sem dúvida esse é o passo mais simples a se realizar dentro de um Projeto. Para isso basta clicar com o botão direito do mouse na palavra *Diagrams* e *Create Diagram*, como mostra a Figura 41.

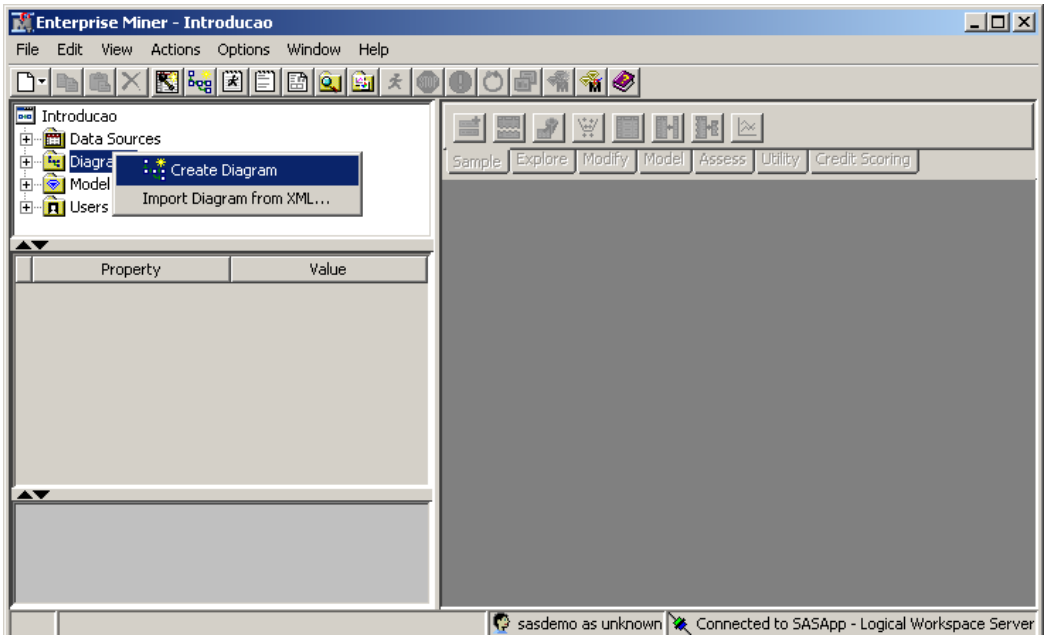

Figura 41 - Indicação para criação de um novo diagrama

Em seguida basta digitar um nome para o diagrama, como na Figura 42 e *OK*.

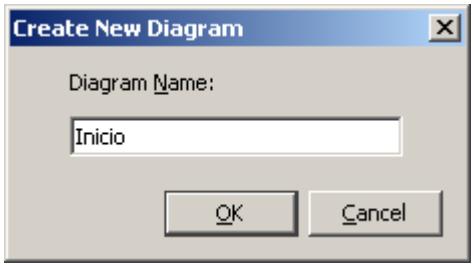

Figura 42 - Criação de um novo diagrama

Com a criação de um diagrama todas as funcionalidades da ferramenta ficam disponíveis para uso (Figura 43). Agora basta criar uma biblioteca e em seguida um *Data Source*.

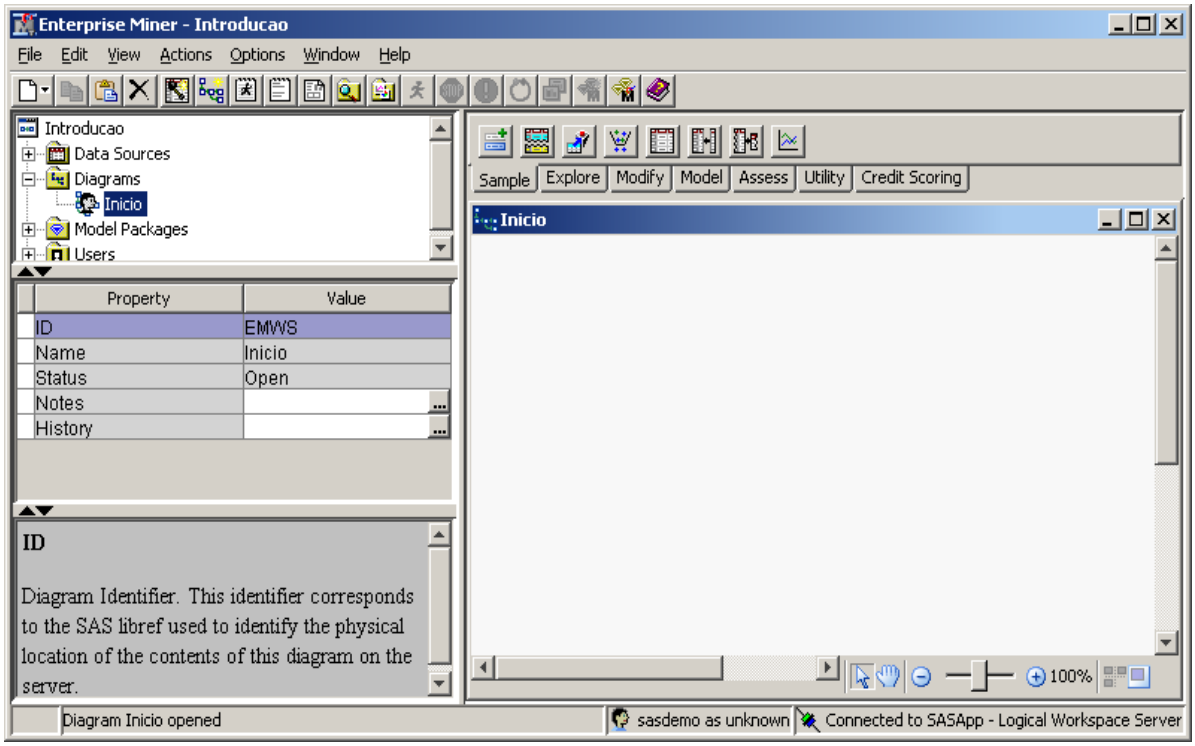

Figura 43 - Visualização do SAS *Enterprise Miner* após a criação de um diagrama

D.4.3 Criação de uma Biblioteca

Para a criação de uma biblioteca precisa-se apenas especificar um caminho, indicando ao SAS onde as bases estão armazenadas. Nesse ponto pode-se fazer uma leitura de bases já em formato SAS (SAS *Data Set*) ou então, por exemplo, num banco de dados (ODBC, Oracle, DB2,...). Para a criação de uma biblioteca dentro do *Miner*, pode-se optar pelo *Wizard* ou então pelo código. Apresenta-se aqui os dois métodos.

D.4.3.1 Opção Wizard

Para criação de uma biblioteca pela função Wizard, deve-se ir em *File*, *New*, *Library*. Na etapa 1 seleciona-se a opção *Create New Library* e *Avançar*.

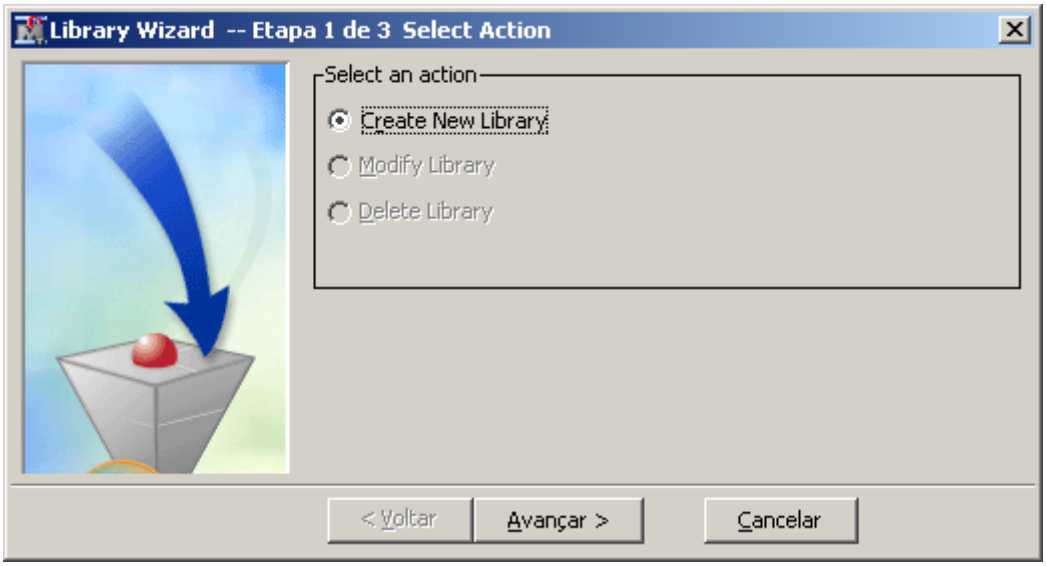

Figura 44 - Etapa 1 para a criação de uma Biblioteca no SAS *Enterprise Miner*

Na etapa 2 nomea-se essa biblioteca, coloca-se o endereço de onde os dados estão armazenados, no campo *Path* (Figura 45). *Avançar* e no próximo passo tem-se o status da criação e as informações sobre a biblioteca e *Concluir* para finalizar a atividade.

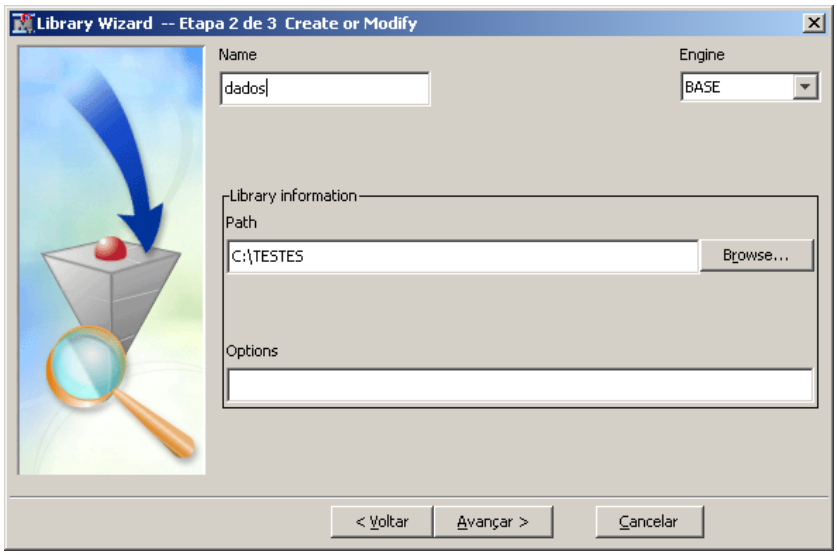

Figura 45 - Etapa 2 para a criação de uma Biblioteca no SAS *Enterprise Miner*

D.4.3.1 Opção Código

Para criação de uma biblioteca via código o procedimento é muito simples. Selecionando o nome do projeto, no campo *Project Start Code* dentro Menu e clica-se na elipse, indicada na Figura 46.

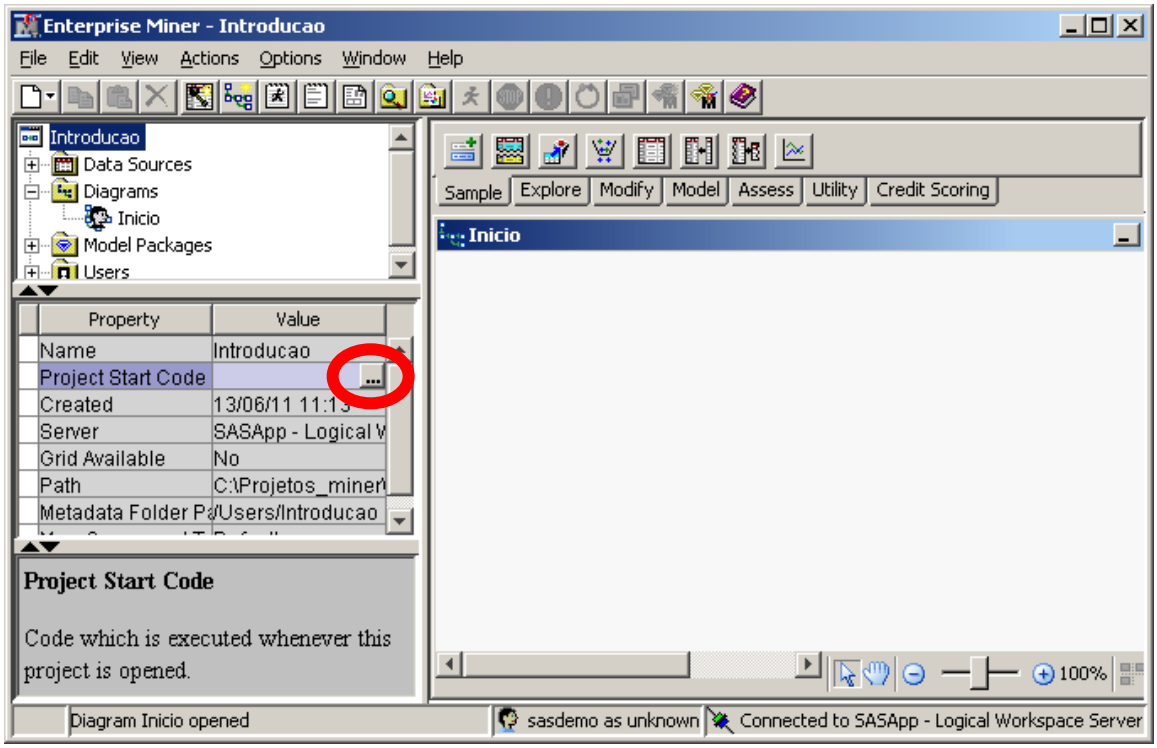

Figura 46 - Indicação do caminho para criação de uma biblioteca via código SAS

O próximo passo será digitar o código com o caminho de onde deverá ser feita a leitura dos dados. O comando é o mesmo usado tanto no SAS *Base* como no SAS *Guide*. A linguagem é exatamente a mesma. Com isso, o código será: *libname dados 'C:\TESTES';* (Figura 47). Para executar o comando, basta clicar em *Run Now* e em seguida verificar a execução, na aba *log*.

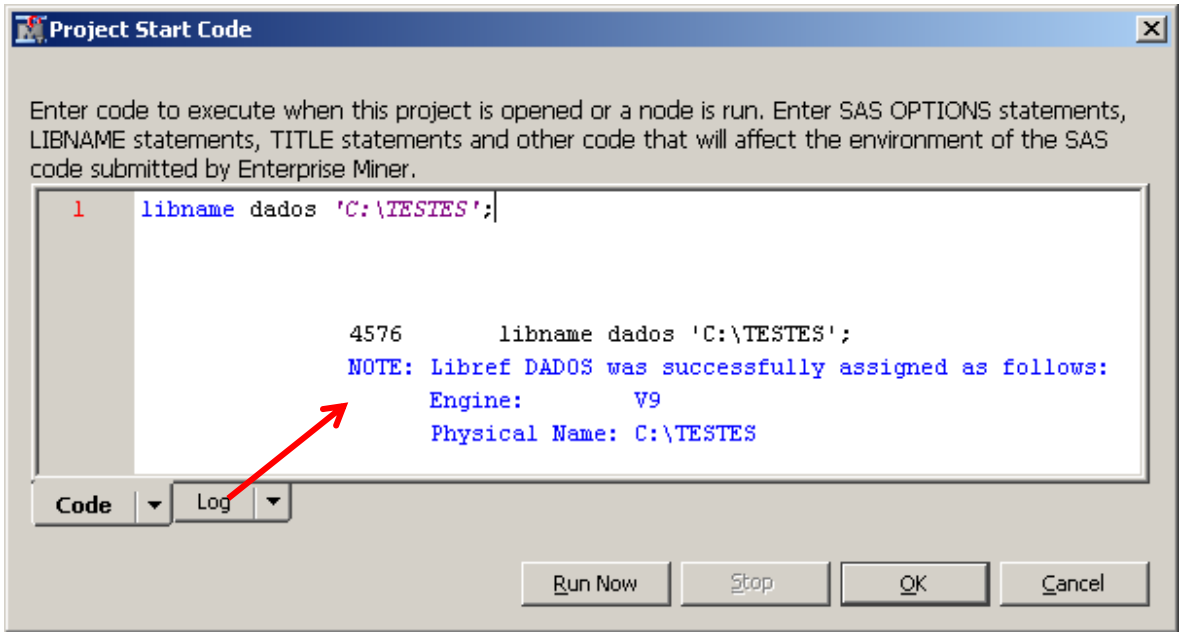

Figura 47 - Código SAS para criação de uma biblioteca, junto com o resultado do Log

Depois de criada a biblioteca o próximo passo é a criação do *Data Source*, ou seja, metadados que informam ao SAS *Enterprise Miner* sobre o nome, a localização da tabela SAS, o SAS código que é usado para definir um caminho da biblioteca, os papéis de cada variável para análise, os níveis de medição e outros atributos que norteiam o processo de mineração de dados.

D.4.4 Criação de um *Data Source*

No software SAS *Enterprise Miner* pode-se inserir tabelas para análise por meio de uma biblioteca e um *Data Source*, ou pelo nó ***File Import***. O mais recomendado é que toda a manipulação e geração de base de dados para a análise seja feita no SAS *Enterprise Guide* e que apenas o desenvolvimento da modelagem seja feita no SAS *Enterprise Miner*. Com isso, nesse material, descreve-se apenas da inserção de uma base de dados que já esteja em formato SAS e no formato exigido pelo modelo.

Como a biblioteca SAS já existe, o caminho para alcance dos dados já está sinalizado no SAS *Enterprise Miner*. O que deve-se fazer é informar ao *Miner* características da base de dados em estudo. Deve-se descrever o papel de cada variável, seus níveis de medição e alguns outros atributos importantes para análise.

Como feito na criação do diagrama, clica-se com o botão direito do mouse na palavra *Data Source* e *Create Data Source*. Na etapa 1 opta-se pela opção SAS *Table*, *Avançar*. Na etapa 2 especifíca-se em qual biblioteca os dados estão armazenados, como na Figura 48 e *Avançar*.

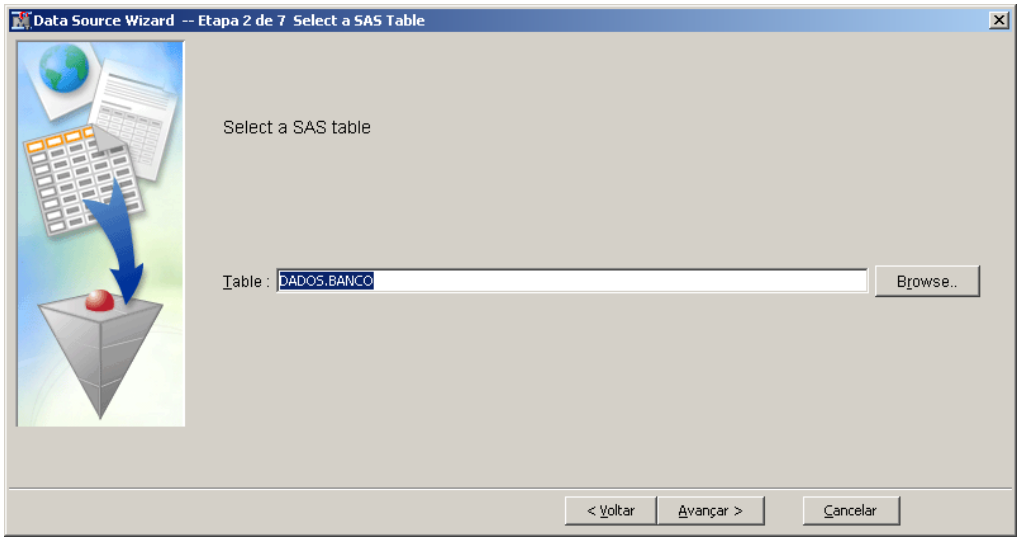

Figura 48 - Etapa 2 para criação de um *Data Source*

Na próxima etapa confere-se as informações sobre o *Data Source* e *Avançar*. No próximo passo especificar-se características de cada variável da base de dados. Existem duas maneiras de se fazer isso, pelo método básico ou pelo método avançado.

No caso do método Básico o SAS *Enterprise Miner* fornece as regras e níveis iniciais com base no tipo e formato das variáveis. Pode ser necessário ajustar estas regras e níveis de medição. Já na opção Avançado, pode-se customizar (botão *Customize...*) como serão as regras de cada variável, como por exemplo (Figura 49) a regra que cada variável com mais de 50% de *missing* será automaticamente marcada como *rejected*, ou então, que uma variável intervalar que tiver menos de 20 números distintos será classificada como Nominal e que uma variável classificatória que tenha mais de 20 níveis será rejeitada. Esses números podem ser alterados conforme a necessidade da análise.

Muitas vezes a opção Avançado já ajuda com as classificações, por isso, muitas vezes é o caminho preferido pelos analistas. Selecionado *Advanced* e *Avançar*, ajusta-se a descrição de cada variável, Figura 49.

A coluna *Role* especifica o papel de cada variável na análise. Por exemplo, uma variável pode ter o papel de ID (identificação), de *input* (variáveis independentes no modelo) ou de *target* (variável dependente).

A coluna *Level* especifica o nível de medição de cada variável. Por exemplo, uma variável pode ser ordinal, nominal, intervalar, ou binária. Todas essas classificações serão usadas nos passos de modelagem, por isso este é um passo muito importante na análise. Cada nó tem uma exigência sobre as variáveis, com isso deve-se ter em mente o tipo de análise que irá realizar. Caso seja necessário mudar algo depois de finalizado o *wizard* do *Data Source*, pode-se fazer alterações na descrição das variáveis direto no nó da base de interesse.

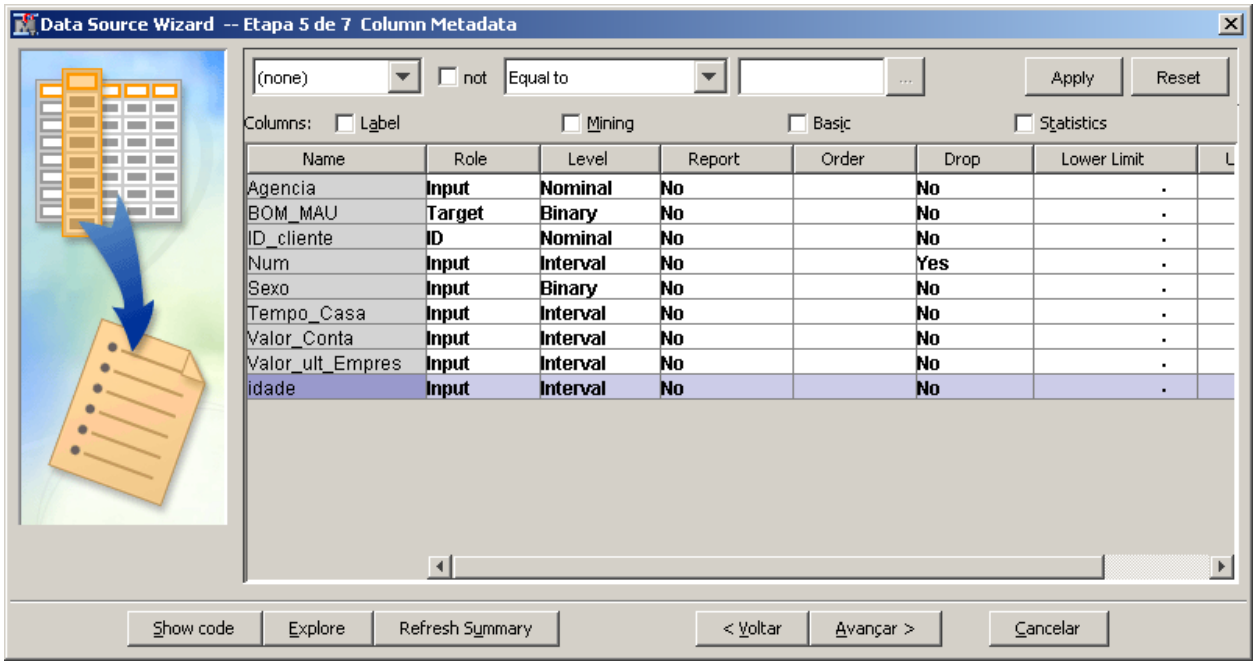

Figura 49 - Etapa 5 para criação de um *Data Source*

A etapa seguinte oferece a opção de criar um modelo baseado no valor de cada decisão (para utilizar essa ferramenta é necessário assegurar que existe uma variável *target* e que o nível desta variável não é intervalar).

Finalmente, o último passo é especificar o papel da tabela SAS na análise. A tabela pode ser: *Raw*, *Train*, *Validation*, *Test*, *Score* ou *Transaction*. Cada ferramenta no *Miner* exige um formato pré-definido das tabelas. Escolhe-se a opção

Raw quando tem-se dados brutos e deles faz-se partições para modelagem e validação. A opção *Train* é usada quando a base será totalmente utilizada para a construção dos modelos, *Validation* quando a base será usada para validação dos modelos e *Test* quando a base será utilizada para testar os modelos. Base *Score* é a base em que aplica-se o modelo selecionado (o nó *Score* exige uma base com essa classificação, caso contrário, não executa) e *Transaction* quando trabalha-se com dados transacionais, por exemplo, para uma análise de Associação.

Um ponto importante que deve ser mencionado é sobre a definição do que é *Data Source*, que não é o mesmo que uma tabela ou dados em formato *Data Set* SAS. *Data Source* é uma definição de metadados que fornece ao SAS *Enterprise Miner* informações sobre um conjunto de dados SAS ou tabela SAS.