PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

FACULDADE DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA - PPGEE

MODELO DE MINERAÇÃO DE DADOS PARA CLASSIFICAÇÃO DE CLIENTES EM TELECOMUNICAÇÕES

RAFAEL JORDAN PETERMANN

Porto Alegre.

Outubro, 2006

RAFAEL JORDAN PETERMANN

MODELO DE MINERAÇÃO DE DADOS PARA CLASSIFICAÇÃO DE CLIENTES EM TELECOMUNICAÇÕES

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre, pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

Orientador: Prof. Dr. Fabian Luis Vargas

Porto Alegre
2006

Dedico este trabalho a minha noiva Roberta,
aos meus pais Cláudio e Sônia e a minha irmã Juliana.

AGRADECIMENTOS

Agradeço a Deus pela proteção, pela saúde e inspiração.

Ao Prof. Dr. Fabian Luis Vargas, por acreditar neste trabalho e pela orientação na elaboração desta dissertação.

Aos meus amigos e familiares, pelo suporte e apoio constantes.

Ao meu amigo Rodolfo, pela parceria nestes meses de mestrado.

Aos meus colegas de trabalho na Brasil Telecom, pela troca de conhecimentos, pela convivência e contribuições para a elaboração deste trabalho.

A equipe da secretaria do PPGEE, pelo suporte e atenção.

A Brasil Telecom S.A. pela concessão do auxílio financeiro e pela disponibilização de informações e ferramentas para a execução deste trabalho.

“Mais alto sobe aquele que ajuda o outro a subir”.
(George Adams)

RESUMO

O objetivo desta dissertação é desenvolver um modelo completo de mineração de dados no ambiente de uma operadora de telecomunicações, com foco na retenção de clientes usuários do STFC (Serviço Telefônico Fixo Comutado). Atualmente, a manutenção da base de clientes é ponto crucial para a atuação das operadoras de telecomunicações no país. Com o surgimento de novas tecnologias de comunicação e com a popularização de acessos de banda larga e do SMP (Serviço Móvel Pessoal), as taxas de cancelamentos dos acessos de STFC exigem das operadoras que oferecem o serviço um processo consistente visando à retenção da sua planta física instalada e da receita gerada. Através da construção de um modelo de mineração de dados, formou-se um sistema visando à predição de eventos e a classificação de clientes. O evento a ser previsto é o *churn* (cancelamento do serviço), com base na utilização de algoritmos classificadores aplicados sobre uma base de dados real, contendo informações cadastrais, de relacionamento com o fornecedor, de consumo e faturamento. A formação do modelo de mineração de dados envolveu as etapas de análise do problema (*churn*), avaliação e entendimento dos dados, pré-processamento e classificação. Como algoritmos classificadores (utilizados na predição), foram estudados e utilizados três métodos: Redes Neurais RBF (*Radial Basis Function*), Árvores de Decisão e Classificadores Bayesianos. Os resultados obtidos validam o modelo desenvolvido, permitindo a sua utilização e aperfeiçoamento no ambiente de uma operadora de telecomunicações ou ainda como um modelo genérico de mineração de dados, passível de aplicação em diferentes segmentos envolvendo o problema da retenção e fidelização de clientes.

Palavras-chave: Mineração de Dados. Telecomunicações. Redes Neurais Artificiais. Árvores de Decisão. Classificadores Bayesianos. *Churn*. *CRM - Customer Relationship Management*

ABSTRACT

The aim of this work is to develop a complete data mining model in a telecommunication company, focusing on client/user of the STFC (Commuted Fixed Telephone Service). Currently, the management of the database containing client information is an essential point in the success of telecommunication companies in the country. With the raise of new communication technologies and with the easy access to broadband and mobile services, there is an increasing number of clients canceling the STFC service. This forces the STFC companies to keep a consistent service process aiming to use well their physical installations and to keep the income. With the development of a data mining model, we generated a system aiming to predict events and to classify clients. The predicted event is called chum (service cancellation) and it is based on classificatory algorithms applied to a real database, containing records such as general information, client relationship and billing. The development of the data mining model was constituted of the following: problem analysis (chum), evaluation and understanding of the data, pre-processing and classification. We studied and used the following three classification algorithms methods for the prediction: Neural Networks RFB (Radial Basis Function), Decision Trees and Bayesian Classifiers. The results obtained validate the model developed by us, allowing the use and improvement in the telecommunication companies. The model can also be used as a generic data mining model, with possible applications in diverse fields related to keeping clients loyalty.

Keywords: Data Mining. Telecommunications. Artificial Neural Networks. Decision Trees. Bayesian Classifiers. Churn. CRM - Customer Relationship Management.

SUMÁRIO

I- OBJETIVOS

II- FUNDAMENTAÇÃO TEÓRICA

III- IMPLEMENTAÇÃO E RESULTADOS

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

1.1 Motivação

Considerando-se o cenário atual de telecomunicações no Brasil, observa-se um forte movimento de convergência em tecnologias, serviços e aplicações – acompanhando a tendência que se verifica no cenário global. Há alguns anos, quando se falava em telecomunicações pensava-se em comunicação de voz. Hoje, fala-se em comunicação de voz, dados e multimídia, com convergência entre operações fixa e móvel.

Atualmente e de forma cada vez mais acentuada, o nível de competitividade de uma organização é ditado pela sua capacidade de inovar em resposta às necessidades do mercado e às investidas da concorrência. O domínio tecnológico é um dos fatores críticos deste processo, fazendo com que tecnologia e informações sejam consideradas um ativo importante para a empresa.

De fato, os investimentos em inovação e capacitação tecnológica são de fundamental importância para uma operadora de telecomunicações garantir e expandir sua posição no mercado. Os investimentos nesta área estão estreitamente vinculados às decisões estratégicas das empresas e, nos mais diferentes países, recebem apoio institucional e financeiro do poder público.

No entanto, a sustentação para a operação de qualquer companhia ou organização é a sua base de clientes. Assim, uma visão unificada sobre estes componentes (capacitação tecnológica, informação e base de usuários) permite a formação de um diferencial competitivo e a possibilidade do oferecimento de um relacionamento vantajoso para ambos (clientes e fornecedor de serviços – no caso de uma operadora de telecomunicações).

Segundo Gensch et al.[1] (1990 citados por NOGUEIRA, 2004), a formatação de um modelo de inteligência sobre uma base de dados de um sistema de relacionamento, permite, sob a ótica do cliente, a possibilidade de identificação de serviços mais adequados às suas necessidades, pagando menos e recebendo mais. Para a empresa, percebe-se a possibilidade de vender mais a clientes que potencialmente poderiam estar com perfis

[1] GENSCH, Denis; AVERSA, Nicola; MOORE, Steven.. **A Choice-Modeling Market Information System That Enabled ABB Eletric to Expand Its Market**. Interfaces Magazine, January-February, 1990.

diferentes de uso. Surgem as oportunidades de retenção maior da sua base de clientes, a criação de laços mais fortes e duradouros, assim como as receitas que dessa relação advirem.

Ao longo dos anos muitos pressupostos e práticas de negócios antes tomadas como ideais foram caindo por terra, dando lugar a novos conceitos utilizados mais e mais a cada dia. O mundo e os mercados em ritmo de rápida mudança obrigam as empresas a evoluírem sempre, se desejarem manter sua prosperidade. O foco deixou de estar no produto e passou para o mercado e o cliente. Deixou de ser doméstico para ser global. Produtos padronizados deram lugar a produtos personalizados, o marketing de massa se transformou em marketing direcionado, caminhando para um marketing individualizado para cada cliente (KOTLER, 1999).

Investir em clientes não lucrativos, oferecendo-lhes subsídios que não trarão retornos posteriores é um exemplo intuitivo da perda de rentabilidade "em todos os prazos". Por outro lado, garantir a permanência de um cliente que signifique uma receita segura e constante, pode justificar a abdicação de parte da lucratividade imediata, ao ofertar-lhe um serviço mais adequado ao seu uso, uma isenção ou desconto (NOGUEIRA, 2004).

Isso se explica pela constante evolução dos custos de aquisição de novos clientes (crescentes), contrapostos à receita média geradas por eles (decrescentes). Aquele cliente que deixou a base de usuários e que estava gerando receitas para a empresa, tem um substituto que custa cada vez mais para ser "adquirido" e, provavelmente, gerará menos receita. Isso se explica pelo fato de que as parcelas menos privilegiadas da sociedade (concentradas nas classes *C*, *D* e *E*) são as classes "entrantes" no segmento de usuários de telefonia (NOGUEIRA, 2004).

A conquista de novos clientes é associada, na maioria dos serviços, à oferta de subsídios, seja em descontos tarifários ou outras promoções. Essa prática tem redundado na redução dos lucros das operadoras. Por isso, verifica-se, atualmente, grande empenho na fidelização dos clientes, o que pode ser traduzido nos grandes investimentos que estão sendo feitos em qualidade, *call-centers* e sistemas de atendimento ao cliente (BNDES, 2000).

Além disso, o aumento da diversidade de produtos e serviços vem impondo às empresas de telecomunicações a necessidade crescente da segmentação de seus produtos e

mercados a fim de se manterem e/ou crescerem num cenário cada vez mais dinâmico e competitivo, no qual os produtos apresentam pouca distinção e ciclos de vida acelerados.

A competição e a concorrência entre operadoras, inexistente em anos atrás devido ao monopólio estatal, ocorre agora em vários ambientes, sendo apoiada por uma tecnologia da informação em crescente eficiência e eficácia de resultados. Estes avanços tecnológicos, conjuntamente com a imprevisibilidade dos eventos em curso nos ambientes empresariais, têm imposto às organizações a necessidade de se reorganizarem para o estabelecimento de novas formas de se fazer negócio e atender os seus clientes.

Enquanto as margens caem, a demanda por produtos de qualidade aumenta, os preços e tarifas tornam-se menores e as exigências, por parte dos clientes, por serviços que agreguem valor aos produtos crescem.

Segundo Bretzke (2000), a nova forma de ação estratégica e mercadológica tem-se processado decisivamente em dois pólos:

1- na tecnologia da informação;

2- na estratégia de comercialização dos produtos e serviços.

Assim, este trabalho propõe-se a definição de um modelo unificando descoberta de conhecimento sobre a base de dados de um sistema de relacionamento com clientes (ou *CRM – Customer Relationship Management*) em uma abordagem preditiva, focada na retenção de clientes usuários do serviço de telefonia fixa da operadora Brasil Telecom, em especial na sua unidade do Rio Grande do Sul. O foco na base de usuários do serviço de telefonia fixa da Brasil Telecom justifica-se devido à:

- extensa base de usuários, concentrando mais de 10 milhões de linhas fixas instaladas no Brasil (conforme ilustrado na Tabela 1.1);
- concentração da receita em serviços da operadora, provenientes do tráfego de voz gerado e pela assinatura básica residencial;
- necessidade de atenção ao problema do abandono de clientes (ou cancelamento dos serviços) no ambiente da empresa.

Em termos mercadológicos, a base de usuários de serviços de telefonia fixa também representa o público alvo para a operadora ampliar a base de acessos banda larga e/ou o número de usuários do Serviço Móvel Pessoal (SMP ou celular). O formato de operação convergente, onde a operadora possui licenças para comercializar acessos fixos, móveis ou banda larga permite a definição de estratégias visando à transição e migração gradual dos serviços de telefonia fixa tradicionais para novas tecnologias.

A telefonia fixa está a caminho de se integrar ao protocolo da Internet – IP, o que também está ocorrendo com a telefonia móvel, desenhando uma nova configuração do setor de telecomunicações, incluindo novos agentes e formas de regulação.

Esta fusão permite a introdução de uma série de processos que agregam valor aos serviços, podendo tornar realidade de mercado tecnologias como a Wi-Fi (*Wireless Broadband*), que faz convergir comunicação sem fio com a banda larga. Este tipo de conexão torna possível prever incrementos no acesso ao do Protocolo Internet (IP), impulsionando ainda mais alternativas ao antigo modelo de telefonia fixa.

No entanto, a base de usuários de telefonia fixa no Brasil apresenta ainda um cenário de estabilidade, conforme exibido na Tabela 1.1. Assim, tendo em vista o novo modelo de operadoras convergentes, a base de usuários de telefonia fixa ainda é responsável pela parcela mais significativa da receita operacional das empresas detentoras de licenças para prestação do serviço de telefonia fixa.

Ano	Acessos Instalados	Acessos em Serviço
2005	50,3	39,6
2004	50	39,6
2003	49,8	39,2
2002	49,2	38,8
2001	47,8	37,4
2000	38.3	30,9
1999	27.8	25
1998	22.1	20
1997	18,8	17

Tabela 1.1: evolução de acessos de telefonia fixa no Brasil (em milhões)

Fonte: Anatel, 2006

A tabela 1.2 demonstra o cenário evolutivo recente, considerando os totais de acessos por operadoras. Observa-se, de modo geral, a estabilização do número de acessos ou pequeno incremento em alguns casos.

	dez/05	jan/06	fev/06	mar/06
Telemar	17.029.361	17.036.812	17.029.179	17.015.627
Brasil Telecom	10.815.169	10.818.980	10.819.728	10.813.351
Telefonica	13.240.978	13.241.664	13.242.599	13.244.488
CTBC	861.038	861.322	862.449	859.529
Sercomtel	163.110	163.118	163.133	163.470
Embratel	1.524	1.524	1.524	1.524
Total Concessionárias	**42.111.180**	**42.123.420**	**42.118.612**	**42.097.989**

Tabela 1.2: evolução de acessos de telefonia por operadoras
Fonte: Anatel, 2006

A migração desta base de usuários para novos serviços, no entanto, já é premissa válida no atual cenário. A substituição dos terminais fixos por acessos móveis é prática comum, ocorrendo em paralelo com a utilização de serviços de voz sobre IP (principalmente devido a pulverização de acessos de banda larga). No caso de operadoras convergentes, ocorre uma alteração no perfil de consumo, e um ajuste na composição da receita – desde que o usuário não migre para serviços da concorrência, cancelando o serviço.

Este trabalho busca tratar o problema dos cancelamentos ou solicitações de retiradas de serviço em terminais fixos da operadora, através da elaboração de um modelo que envolve a formação de um banco de dados baseado no sistema de CRM da Brasil Telecom S.A. e a aplicação de técnicas de mineração de dados e descoberta de conhecimento. No entanto, deve-se notar que a intenção e motivação principal é a de manutenção do cliente (e não apenas do terminal fixo em questão). Para tanto, ao longo deste trabalho, efetua-se um estudo do cenário atual de telecomunicações no Brasil, focado na questão da problemática da retenção de clientes e uma revisão das principais técnicas de descoberta de conhecimento e mineração de dados, visando à estruturação completa de um modelo de retenção de clientes.

1.2 Objetivos

1.2.1 Objetivo principal

O objetivo principal deste trabalho é construir um modelo de mineração de dados (*Data Mining*) funcional para uma base de informações de um sistema de CRM (*Customer Relationship Management* – ou Gestão de Relacionamento com o Cliente) de uma operadora de telecomunicações. Tal modelo visa à retenção de clientes usuários do serviço STFC (Serviço Telefônico Fixo Comutado), através de um enfoque preditivo sobre potenciais usuários dispostos a solicitarem o cancelamento do serviço. O sistema deve mostrar a importância da disponibilidade de uma ferramenta de mineração de dados para subsidiar as diferentes análises que servem de apoio à tomada de decisões.

A utilização de um modelo de mineração de dados com enfoque preditivo sobre a perda de clientes ou a substituição de operadora pode produzir resultados satisfatórios, como descritos por Ferreira (2005), em seu estudo e modelo implementados sobre uma base de usuários de telefonia celular.

Segundo Ferreira (2005), é essencial unir profundo conhecimento do negócio ao uso sábio e judicioso de métodos e algoritmos computacionais, através de um sistema que englobe e leve em consideração todas as peculiaridades do problema a ser solucionado. Só assim o já mencionado conhecimento será realmente encontrado em meio às grandes quantidades de dados disponíveis e, conseqüentemente, desta forma será gerada inteligência real sobre o negócio, servindo de base no processo decisório.

Um sistema de CRM é responsável pelo armazenamento das diversas interações entre provedor de serviços e seus clientes. Por esta natureza, as bases de dados utilizadas compõem uma fonte rica em informações e bastante fértil para o processo de mineração de dados e descoberta de conhecimento (ou *KD – Knowledge Discovery*). Tais bases englobam conceitos como marketing individualizado, customização de ofertas e automatização e direcionamento da força de vendas. Um CRM de qualidade requer o entendimento profundo de quem são os clientes e suas necessidades e anseios, agindo sobre eles de forma pró-ativa. Significa reconhecer que clientes estão insatisfeitos e tomar alguma atitude antes que eles

abandonem a empresa em busca de um concorrente ou de outra alternativa de prestação de serviço.

1.2.2 Objetivos secundários

Este trabalho visa também:

- estudar as principais técnicas e ferramentas aplicáveis em um modelo de mineração de dados com enfoque preditivo, e que sirvam de suporte à definição de sistemas de apoio à decisão em um ambiente de uma operadora de telecomunicações;
- demonstrar as vantagens e os problemas encontrados em um modelo baseado em mineração de dados, considerando projetos referentes a predição, especialmente na retenção de clientes.

Este modelo prevê a definição de uma base de dados, onde serão utilizadas e testadas ferramentas de mineração de dados, visando à predição de eventos condizentes com uma situação iminente de perda do cliente, ou seja, a solicitação da retirada do serviço. Assim, também é considerado objetivo deste trabalho a busca de respostas para as seguintes questões, levantadas por Nogueira (2004):

- em que clientes investir?
- quais estarão propensos a trocar de operadora ou cancelar o serviço?
- quais oferecem um custo de manutenção maior do que a receita gerada?
- quais poderiam estar em um patamar superior de consumo?
- quais os que podem estar com serviços inadequados?

Sob a ótica deste cenário, evidencia-se a importância de sistemas e modelos focados no relacionamento e contato direto com a base de clientes, que é o pilar principal de sustentação das operações em uma operadora de telecomunicações. Assim, a proposta deste

trabalho é projetar um modelo de mineração de dados plenamente funcional, projetado sobre a base de informações de um sistema de CRM da operadora.

1.3 Justificativa

As contribuições potenciais deste trabalho são as seguintes:

- servir de fonte para consultas sobre a descoberta do conhecimento em bases de dados, visto que o assunto é relativamente novo e não existe grande disponibilidade de bibliografia, principalmente em língua portuguesa;

- o desenvolvimento de uma pesquisa envolvendo melhores práticas de áreas diversas, como marketing, relacionamento com cliente, engenharia de dados e descoberta de conhecimento oferece perspectivas para o desenvolvimento de novos projetos semelhantes, envolvendo questões multidisciplinares;

- a utilização de ferramentas diversas e algoritmos robustos de mineração de dados com resultados práticos sobre um estudo de caso real pode servir como fator motivacional para a implementação de novos modelos e projetos de pesquisa no ambiente industrial ou acadêmico;

- para a empresa de telecomunicações que fornece dados para as amostras utilizadas, serão fornecidos resultados e padrões de informações descobertos. Estes, por sua vez, poderão ser utilizados em ações práticas, favorecendo o processo de tomada de decisão, direcionando estratégias e processos. Além disso, o conhecimento oculto a ser e descoberto nos padrões minerados pode ser utilizado como vantagem competitiva e aperfeiçoamento das ações de relacionamento com clientes;

- em termos de inovação, principalmente considerando-se o novo modelo de operadoras de telefonia convergentes no país (fornecedoras de um conjunto completo de serviços – voz fixa e móvel, dados e internet), esta

dissertação apresenta um estudo de caso aplicado sobre um cenário ainda em formação e caracterização. A literatura e as pesquisas encontradas referenciando o tratamento do problema do *churn* em telecomunicações aplicam-se basicamente à telefonia celular e a constante troca de operadoras da base de clientes. Assim, a formatação de um modelo de previsão de churn sobre STFC (Serviço Telefônico Fixo Comutado) em uma operadora convergente em termos de oferecimento de serviço, é distinta e diferenciada das publicações e referências disponíveis.

1.4 Metodologia

Este trabalho caracteriza-se, sob termos de metodologia científica, em apresentar um estudo de caso, descritivo e avaliativo, visto que apresenta a realidade do tema em questão, sem o objetivo de modificá-la. No entanto, propõe-se a elaboração de um modelo que permite a análise estruturada e a avaliação detalhada das variáveis que formam a realidade do estudo, favorecendo a posterior tomada de decisão e formação de ações.

Em relação à metodologia de mineração de dados a ser empregada, trata-se de um processo de avaliação das principais técnicas de descoberta de conhecimento em bases de dados, em um modelo de Classificação / Predição de uma classe de usuários de serviços da operadora de telecomunicações Brasil Telecom S.A.

As etapas envolvidas na elaboração deste trabalho são descritas em função da metodologia de padrões abertos para mineração de dados, proposta em CRISP (2000).

O modelo de processo CRISP-DM (*Cross Industry Standard Process for Data Mining*), apresentado em CRISP (2000), foi desenvolvido como um projeto neutro de indústria e ferramentas para a implementação de mineração de dados. A proposta deste modelo é a definição de um processo aplicável aos diferentes segmentos da indústria, permitindo maior agilidade, redução de custos e facilidade de gerenciamento no processo de mineração de dados.

CRISP-DM define um processo de mineração de dados não linear. Nesse modelo, o ciclo de vida do projeto de mineração de dados consiste em seis fases. A seqüência dessas

fases não é rigorosa, depende do resultado de cada fase ou de qual tarefa particular de uma fase precisa ser executada na próxima fase. A Figura 1.1 ilustra a metodologia.

As setas indicam as dependências mais importantes e freqüentes entre as fases. O círculo externo na figura simboliza a natureza cíclica da mineração de dados. Um processo de mineração continua mesmo após a descoberta de uma solução. Os projetos seguintes de mineração se beneficiarão das experiências anteriores.

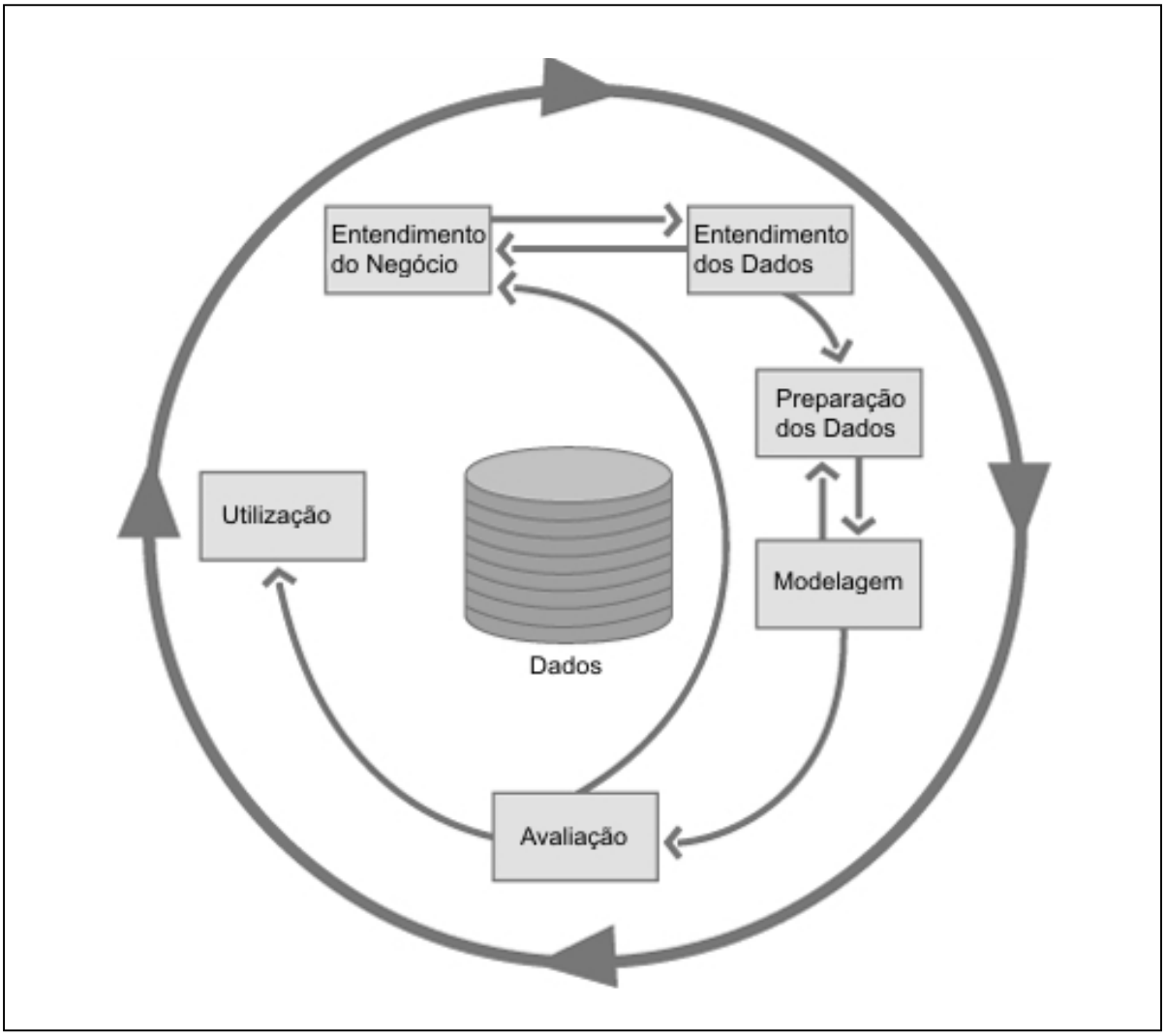

Figura 1.1: metodologia CRISP-DM para mineração de dados

Fonte: Crisp (2000)

Descreve-se a seguir, resumidamente, cada uma das etapas da metodologia CRISP-DM, visto que os conceitos são bastante semelhantes e de acordo com os modelos

propostos na literatura, principalmente por Fayyad et al. (1996), Cabena et al. (1997), Cios (2000), Han et al. (2001) e Klösgen et al. (2002).

- **Entendimento do Negócio** (*Business Understanding*): visa a assimilação dos objetivos do projeto e dos requisitos sob o ponto de vista do negócio. Com base no conhecimento adquirido, o problema de mineração de dados é definido e então um plano preliminar é projetado para atingir os objetivos;

- **Entendimento dos dados** (*Data Understanding*): inicia com uma coleção de dados e procede com atividades que visam buscar familiaridade com os dados, identificar problemas de qualidade dos dados, descobrir os primeiros discernimentos nos dados ou detectar sub-conjuntos interessante para formar hipóteses sobre a informação oculta;

- **Preparação dos dados** (*Data Preparation*): cobre todas as atividades de construção do conjunto de dados (*data set*) final. As tarefas de preparação de dados são, provavelmente, desempenhadas várias vezes e não em qualquer ordem prescrita. Essas tarefas incluem seleção de tabelas, registros e atributos, bem como transformação e limpeza dos dados para as ferramentas e algoritmos de mineração;

- **Modelagem** (*Modelling*): várias técnicas de modelagem são selecionadas e aplicadas, com ajuste de parâmetros para valores ótimos. Geralmente existem várias técnicas para o mesmo tipo de problema de mineração de dados. Algumas técnicas possuem requisitos específicos para a formação de dados, portanto, retornar à fase de preparação de dados é freqüentemente necessário;

- **Avaliação** (*Evaluation*): o modelo construído na fase anterior é avaliado e são revistos os passos executados na sua construção, visando o atendimento dos requisitos definidos na primeira fase. O principal objetivo é determinar se existe alguma questão de negócio importante que não foi suficientemente considerada;

- **Utilização ou aplicação** (*Deployment*): após o modelo ser construído e avaliado, a fase de utilização pode ser a simples geração de dados apresentáveis na forma de relatórios de suporte à decisão ou então pode se submeter o modelo a um processo repetitivo de ajustes e evolução.

1.5 Estrutura da Dissertação

Esta dissertação apresenta um modelo completo de mineração de dados, com enfoque preditivo na classificação de clientes da Brasil Telecom S.A. Apresenta-se uma revisão bibliográfica contemplando todas as etapas de um processo de descoberta de conhecimento em bases de dados, aplicadas na construção de um modelo e de um Estudo de Caso.

O Capítulo 2 apresenta um estudo sobre o cenário atual do mercado de telecomunicações no país, abordando aspectos da regulamentação e da estrutura de concessão de serviços. Descreve-se o problema do *churn* (perda de clientes em telecomunicações), com suas causas e conseqüências. O Capítulo apresenta brevemente a Brasil Telecom S.A., operadora que fornece os conjuntos de dados utilizados no Estudo de Caso, ressaltando sua área de atuação e principais informações referentes aos serviços oferecidos.

O Capítulo 3 apresenta uma revisão sobre o processo de Descoberta de Conhecimento em Bases de Dados, citando as principais ferramentas de suporte e os desafios inerentes ao método.

As principais técnicas de Descoberta de Padrões são apresentadas no Capítulo 4, com destaque para os métodos de Classificação e Predição (foco deste trabalho).

Os três algoritmos de Predição utilizados no Estudo de Caso são apresentados no Capítulo 5, que descreve as características das técnicas de Redes Neurais (principalmente as redes RBF – *Radial Basis Function*), de Árvores de Decisão e Classificadores *Bayesianos*.

O Capítulo 6 apresenta a estrutura do modelo de predição desenvolvido, ilustrando a formação dos principais conjuntos de dados utilizados e descrevendo suas características e limitações.

O Estudo de Caso deste trabalho é apresentado no Capítulo 7, descrevendo as etapas do processo de mineração de dados aplicado sobre o modelo construído. Apresentam-se os resultados da aplicação de cada um dos algoritmos de classificação e predição selecionados, com uma análise comparativa de desempenho.

2 CONTEXTUALIZAÇÃO

Neste capítulo são apresentados os principais aspectos que compõe o cenário de atuação desta dissertação, descrevendo a estrutura atual do mercado de telecomunicações no país e informações referenciais sobre a Brasil Telecom S.A., principalmente no que se refere ao seu modelo de atuação. Também descreve-se o problema do *churn* e a atenção necessária com a retenção e fidelização de clientes em uma operadora de telecomunicações.

2.1 O Cenário Atual de Telecomunicações no Brasil

A demanda do mercado por serviços cada vez melhores e mais eficientes tem levado a uma reestruturação do setor de telecomunicações em todo o mundo. Tal reestruturação ocorre tanto no âmbito institucional e regulatório como no conjunto de serviços oferecidos ao mercado.

No Brasil, o modelo adotado para a reestruturação institucional foi o da quebra do monopólio e privatização das empresas de serviço de telefonia. O modelo brasileiro visava alcançar objetivos sociais e econômicos e almejava também fomentar a assimilação e incorporação tecnológica por parte do setor de telecomunicações, fazendo com que o Brasil ingressasse com maior rapidez na era da informação (KICKINGER et al., 2001).

Segundo o Relatório de Perspectivas para Ampliação e Modernização do Setor de Telecomunicações 2000 (ANATEL, 2000), o Brasil dispõe de amplo e tecnologicamente avançado leque de serviços de telecomunicações, aspecto qualitativo que levou o país à condição de destaque que hoje ocupa no cenário internacional. Visto o mesmo cenário pelo aspecto quantitativo, verifica-se o rápido crescimento da resposta à demanda nas telefonias fixa e móvel.

O marco inicial deste ciclo de renovação no Brasil é a Lei Geral de Telecomunicações - LGT, promulgada em 1997. Ela define os elementos do novo modelo das telecomunicações brasileiras, onde a operação é transferida aos agentes privados, o marco regulatório é estável e existe independência da agência nacional reguladora. Estes elementos

são necessários para criar um ambiente institucional adequado ao desenvolvimento e capaz de estimular a transformação da estrutura monopolista em outra, de caráter competitivo (SIAS, 2005).

Na prática, a reestruturação do Sistema Telebrás se deu através de três etapas (KICKINGER et al., 2001):

- a cisão de cada operadora do sistema em operadoras de telefonia fixa e de telefonia celular;
- a divisão do território nacional em nove regiões de telefonia celular de banda A, idênticas às adotadas para a banda B, exceto no Estado de São Paulo, onde foi definida apenas uma região para a banda A;
- a divisão do território nacional em três regiões de telefonia fixa local - mantida a Embratel como concessionária de telefonia de longa distância para todo o País. A Tabela 2.1 ilustra a composição das três regiões de concessão definidas pela ANATEL.

	Setores	Concessionárias	Empresas Espelho
Região I	1,2,4 a 17	Telemar	Vésper (Embratel)
	3	CTBC	
Região II	18,19,21,23,24, 26 a 30	Brasil Telecom	GVT
	20	Sercomtel	
	22 e 25	CTBC	
Região III	31, 32 e 34	Telefonica	Vésper SP (Embratel)
	33	CTBC	

Tabela 2.1: concessionárias e "espelhos" por região

Com a reestruturação, houve a preocupação em se criar competição entre empresas prestadoras de serviços de telecomunicações. A reestruturação do setor de telecomunicações brasileiro foi pensada com o intuito de gerar a concorrência, ao mesmo tempo em que permitia o controle e acompanhamento do setor pelas autoridades brasileiras. A concorrência na telefonia móvel foi possibilitada pela criação da banda B, que é formada por empresas-espelho (vencedoras da licitação), concorrentes diretas das empresas concessionárias ("ex-estatais") que operavam antes da reestruturação do setor de telecomunicações (KICKINGER et al., 2001).

Antes da reestruturação do setor e, portanto, antes da privatização, os serviços de telefonia fixa, móvel e de longa distância eram providos pela *holding* estatal de telecomunicações – Sistema Telebrás. Uma vantagem da reestruturação foi o fato de possibilitar concessões de serviços em sua maioria ainda não explorados pela iniciativa privada, que apresentavam elevada atratividade econômica, como é o caso do serviço móvel celular (KICKINGER et al., 2001).

Quanto à telefonia móvel, o Serviço Móvel Celular (SMC), implantado no Brasil em 1990, até 1997, só era explorado por empresas do Sistema Telebrás e quatro outras independentes. Em 1994, antes da quebra do monopólio estatal, existiam no País cerca de 800 mil linhas de celulares. No final de 1997, foram licitadas concessões para a banda B, tendo sido previsto um prazo para que essas operadoras implantassem as suas redes antes da privatização da banda *A*.

A licitação prévia da banda *B* garantiu uma vantagem competitiva para essas operadoras, uma vez que as da banda A estavam com suas plantas desatualizadas tecnologicamente e com restrições a novos investimentos. Em julho de 1998, as empresas de telefonia celular da banda A foram privatizadas. Nessa época, o número de aparelhos em uso chegava a 5,6 milhões. Um ano depois, já havia 10,9 milhões de celulares operando e, no final de 1999, eram 15 milhões, Em junho de 2006, o total de acessos móveis no país superava os 90 milhões (KICKINGER et al., 2001).

A expansão do número de telefones celulares – seguida por um aumento dos serviços ofertados aos usuários e pela queda nos preços – ocorreu, em parte, devido à regulamentação do serviço pré-pago, em 1998. No ano de 2001, no intuito de atualizar o serviço e de introduzir maior competição na telefonia celular, a Anatel licitou licenças para a

operação do Serviço Móvel Pessoal (SMP), considerado sucessor do Serviço Móvel Celular (SMC), abrangendo as operações das bandas *D* e *E*. Foram então criadas novas regiões de prestação de serviço, idênticas às adotadas para a telefonia fixa (KICKINGER et al., 2001).

2.2 O Novo Modelo de Serviço Telefônico Fixo Comutado

Aprimorados e ampliados em relação aos contratos de concessão do Serviço Telefônico Fixo Comutado (STFC) de 1998, os contratos prorrogados em dezembro de 2005 pela Anatel valem a partir de 1º de janeiro de 2006, têm vigência de 20 anos e prevêem a possibilidade de revisão a cada cinco anos, com vistas a novos condicionamentos e re-estudo das metas de universalização e de qualidade. Foram elaborados seguindo as diretrizes de política pública para as telecomunicações definidas pelo Decreto 4.733/2003, acrescidas dos preceitos do Decreto 5.581/2005.

Em 36 meses de trabalhos, o processo de elaboração dos contratos de concessão e dos documentos a eles relacionados seguiu, com rigor, todos os preceitos legais, regulamentares e contratuais, evidenciando a estabilidade regulatória que caracteriza o modelo brasileiro de telecomunicações (KICKINGER et al., 2001).

Neste processo, a minuta de cada documento foi submetida a consultas e a audiências públicas, dando oportunidade para que os diversos segmentos da sociedade apresentassem suas contribuições acerca das regras que nortearão a prestação da telefonia fixa a partir de 2006. Além disso, a Agência registrou valiosas contribuições feitas pela Câmara dos Deputados, pelo Senado Federal, por órgãos de defesa do consumidor, pelo Ministério Público, e por entidades de classe e de representantes dos usuários.

Entre as inovações da telefonia fixa que passarão a vigorar em 2006, destacam-se pontos como:

- faturamento por minutos de utilização do serviço em substituição à medição por pulso aleatório;
- detalhamento não oneroso da conta telefônica (1ª via);
- a criação de conselhos de usuários;

- direito de o assinante receber comunicação prévia, por escrito, da inclusão de seu nome em cadastros de inadimplentes;
- Acesso Individual Classe Especial (AICE);
- regras de acessibilidade para possibilitar o acesso e a utilização do serviço de telefonia fixa por pessoas portadoras de deficiências.

2.3 Retenção de Clientes em Telecomunicações

2.3.1 Definição de *Churn*

Churn consiste no ato de um cliente abandonar uma empresa em favor de uma concorrente, terminando a sua relação com a empresa antiga (na totalidade ou em algum serviço específico) e iniciando uma nova relação com uma outra (MATTISON, 2001).

A origem do termo muito provavelmente se encontra no sentido do verbo "*to churn*" na língua inglesa, que significa "mexer, agitar violentamente". O fenômeno do *churn* na indústria de telefonia causa exatamente o que o verbo quer dizer: uma grande "agitação" de clientes no mercado, onde a troca de fornecedor de serviço ou a substituição por uma nova tecnologia leva as operadoras a produzirem novas formas de manter seus clientes no seu negócio, ao mesmo tempo em que buscam a aquisição de clientes da concorrência (FERREIRA, 2005).

Na informática, *churn* é utilizado, por alguns autores, para expressar a renovação acelerada dos produtos ou a conhecida obsolescência programada. Mas o *churn*, objeto deste estudo, que trata da perda de clientes sofrida por uma empresa para a concorrência, ou seja, é uma medida da infidelidade dos clientes. Este é o conceito que está mais em pauta entre as empresas de telecomunicação ou em qualquer outra empresa de serviços. No Brasil, com a desregulamentação que as telecomunicações sofreram, este é um dos setores que mais vive este fenômeno (CISTER, 2005).

Outros setores da economia também têm que administrar o *churn*. Os bancos e as administradoras de cartão de crédito são dois exemplos conhecidos. Para os consumidores,

esta vasta gama de opções significa maior liberdade de escolha. Portanto, seja na telefonia fixa, móvel, comunicação de dados ou internet, a facilidade de mudança de fornecedor é uma premissa relevante. Conseqüentemente, os fornecedores tratarão de traçar estratégias específicas, de acordo com o comportamento dos consumidores e terão que ir mais longe, atingindo níveis maiores de segmentação e diferenciação, para alcançar seu objetivo de fidelidade. Aplicações de *business intelligence* e CRM são estratégias para tais indústrias (CISTER, 2005).

Sob a ótica dos consumidores, isso significa um incremento na qualidade dos serviços prestados, a possibilidade de comparar propostas, receber suporte e assistência técnica mais qualificados vitória. Hoje o cliente pode comparar propostas, exigir um tratamento de qualidade e ter um suporte e assistência técnica cada vez melhores.

Para empresas que possuem seus negócios baseados em assinatura, *churn* é um tema recorrente e preocupante. O esforço das operadoras de telefonia (fixa ou móvel) é sempre no sentido de reduzir ou eliminar o *churn*. A grande questão é evitar que o usuário tome essa decisão e não apenas desenvolver políticas de retenção paliativas.

2.3.2 Causas do *Churn*

Segundo Cister (2005), existem três tipos de *churn*: involuntário, voluntário e inevitável.

- Involuntário - quando o usuário deixa de pagar pelo serviço e tem seu fornecimento cancelado, por exemplo. Os motivos pelos quais o cliente deixa de pagar podem ser os mais diversos, como desemprego, falta de capital suficiente, entre outros;

- Voluntário - quando o cliente decide mudar de fornecedor, seduzido por campanhas de marketing e/ou promoções;

- Inevitável - quando o usuário muda-se para uma localidade não atendida pelo fornecedor, por exemplo.

Até há pouco tempo atrás, a cultura das empresas era “conquistar o cliente a qualquer preço”. Atualmente, a cultura é “reter o cliente” e, claro, conquistar o bom cliente. Conquistar um cliente do tipo alta-rotatividade, com propensão compulsiva ao *churn* pode não ser um bom negócio; em alguns casos, é até preferível deixá-lo com a concorrência.

É provado que o custo para se manter um cliente é muito menor que o de se conquistar um novo cliente. Para se evitar o *churn*, empregam-se ferramentas de mineração de dados e estatística multivariada. Estas ferramentas permitem que se analise o banco de dados com informações do perfil histórico de cada usuário e que se determine quais clientes são leais, quais são propensos ao *churn* e quais são realmente de alto valor para a empresa (CISTER, 2005).

Com base nessas informações, a empresa toma atitudes não só reativas, em relação aos clientes que desistiram do serviço, mas, principalmente, ações pró-ativas, ao identificar os bons clientes e selecionar planos especiais para garantir sua fidelização, evitando, com isso, a evasão dos clientes que agregam altos valores para a empresa (CISTER, 2005).

Não se pode assumir automaticamente que uma taxa de *churn* alto é ruim, bem como uma taxa de *churn* baixa é boa. Tudo depende do contexto mercadológico envolvido. Em mercados altamente competitivos ou em transição, quando se deseja manter uma participação elevada na publicidade, o *churn* alto pode ser o custo de tal estratégia (CISTER, 2005).

Em um cenário onde a própria operadora oferece novos serviços ao usuário, incentivando a migração de tecnologia, o *churn* pode ser um fator controlado sob o contexto da manutenção do cliente – assim, ao invés de permitir que o usuário migre para a concorrência, a própria operadora oferece um novo serviço (por exemplo, caso o cliente esteja determinado a cancelar seu terminal residencial fixo em função da substituição por um acesso móvel, é interesse da operadora o oferecimento deste serviço).

O baixo *churn* também deve ser considerado no contexto estratégico. É lucrativo, mas pode significar pouca agressividade de marketing, revelando-se (ou não) prejudicial em longo prazo, já que tende a inibir a entrada de novos consumidores para cobrir os cancelamentos naturais ao longo do tempo.

Predizer que é provável que clientes saiam e persuadir os "*churners*" a permanecerem é empreendimento extremamente difícil para a maioria das companhias de telecomunicações. Os volumes de dados necessários são enormes e, freqüentemente, os *data marts* foram segmentados para que cada setor possa ver seu cliente de uma maneira, tornando a consolidação dos dados difícil. E o que falta para muitas organizações se adequarem a essa nova postura do consumidor é, simplesmente, a perícia para suportar a mineração complexa dos dados e as tarefas de análise preditiva, que são essenciais no *churn* (CISTER, 2005).

2.4 A Brasil Telecom S.A.

A Brasil Telecom S.A. surgiu do programa brasileiro de privatização em telecomunicações. Como concessionária do STFC (Serviço Telefônico Fixo Comutado), atua na região II, conforme definido no Plano Geral de Outorgas – PGO, (ANATEL, 1998), constituindo-se pela aquisição e fusão de dez operadoras de telecomunicações, oferecendo serviços nos estados do Rio Grande do Sul, Santa Catarina, Paraná, Mato Grosso, Mato Grosso do Sul, Goiás, Distrito Federal, Acre, Tocantins e Rondônia.

Suas operações atendem a 24% da população brasileira (aproximadamente 41 milhões de habitantes), 25% do Produto Interno Bruto do País, - PIB (aproximadamente R$ 280 bilhões) e 33% do território nacional (aproximadamente 2,8 milhões de quilômetros quadrados). Em sua região de atuação existem, ainda, quatro áreas metropolitanas com população acima de um milhão de habitantes. Além disso, a região de atuação da Brasil Telecom faz fronteira com Peru, Bolívia, Paraguai, Argentina e Uruguai, o que facilitaria a expansão de seus serviços para o Mercosul (SIAS, 2005).

Focada originalmente no STFC em sua concessão original, a empresa também vem buscando um posicionamento estratégico em novas tecnologias convergentes, ao definir estratégias para ocupação imediata da demanda por serviços de Comunicação de Dados (principalmente em grandes clientes corporativos) e serviços de banda larga e Internet para usuários residenciais (SIAS, 2005).

Com a concessão de licença para a operação de telefonia móvel celular (ou SMP – Serviço Móvel Pessoal), a Brasil Telecom passou a integrar um seleto grupo de operadoras de telecomunicações no mundo capazes de oferecer um conjunto completo de serviços (fixo –

com Longa Distância Nacional e Internacional, móvel, dados e Internet). Hoje, já são mais de 10,8 milhões de linhas fixas (STFC) e mais de 2,8 acessos móveis celular (SMP) em toda a área de atuação da Brasil Telecom, ilustrada na Figura 2.1.

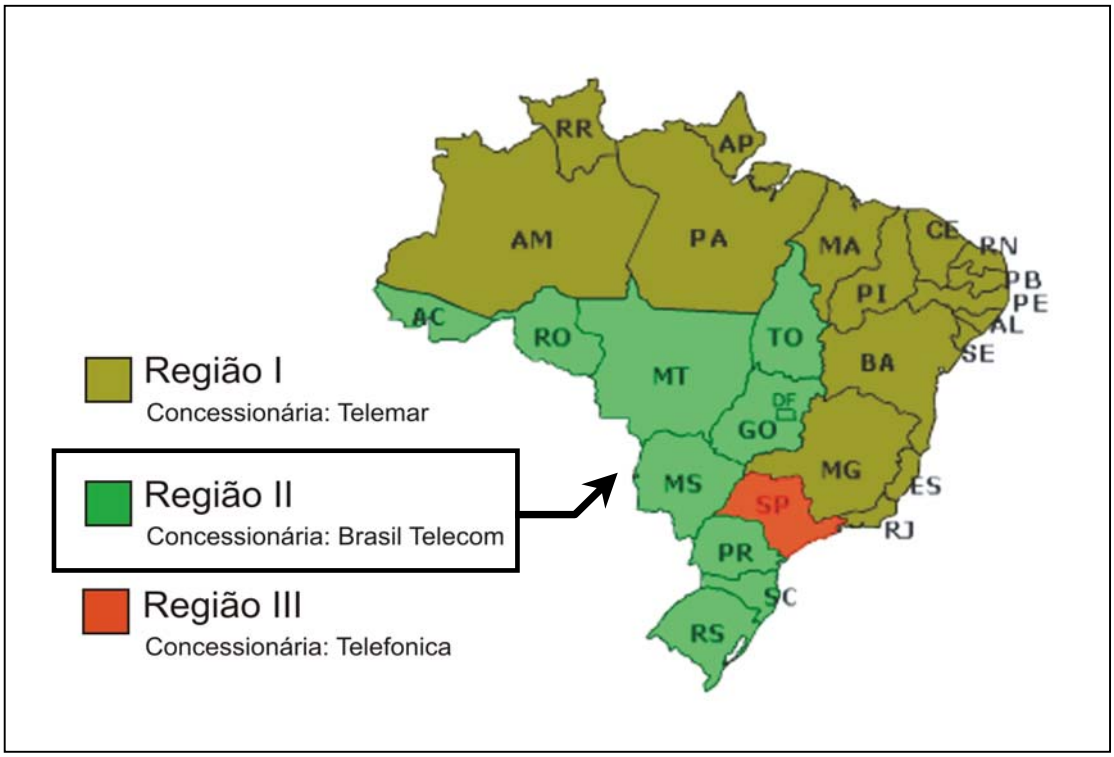

Figura 2.1: região de atuação da Brasil Telecom

2.5 Resumo

Este capítulo apresentou um breve resumo sobre o atual cenário de telecomunicações no país, passando pela estrutura regulatória e destacando o novo modelo definido pela Anatel para o STFC, que é o serviço focado nesta dissertação. Descreveu-se o problema do *churn* em telecomunicações, buscando sua definição e suas principais causas. Evidenciou-se a necessidade de formação de modelos de combate e controle do *churn*, em função da premissa da manutenção da base de clientes ser vital para as operadoras. Apresentou-se também a estrutura da Brasil Telecom S.A. (operadora que fornece as bases de dados utilizadas no estudo de caso). Foram descritos o modelo de atuação, as áreas de concessão e os principais números em quantidades de acesso fixo e móvel da operadora.

3 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

Neste capítulo apresentam-se os principais conceitos referentes ao processo de mineração de dados e descoberta de conhecimento, buscando o entendimento das principais técnicas e suas formas de aplicação no problema em questão.

3.1 Introdução à Mineração de Dados

Mineração de dados é o termo que se popularizou para denominar o processo de descoberta de conhecimento em bases de dados. Trata-se da utilização de ferramentas computacionais a fim de descobrir informações valiosas, potencialmente úteis, descritas na forma de padrões, a partir dos volumes de dados que estão sendo coletados e armazenados pelas organizações atualmente. A obtenção desses conhecimentos implícitos tem sido útil, sobretudo, para as empresas conhecerem melhor seu público-alvo e tomarem decisões mais acertadas ao objetivarem aumentar a competitividade (FAYYAD et al., 1996).

De acordo com Fayyad et al. (1996), o conceito de descoberta de conhecimento em bases de dados pode ser resumido como o processo não-trivial de identificar padrões novos, válidos, potencialmente úteis e, principalmente, compreensíveis em meio às observações presentes em uma base de dados.

Sob a ótica de John (1997) mineração de dados, ou *data mining*, é o processo de análise de conjuntos de dados que têm por objetivo a descoberta de padrões interessantes e que possam representar informações úteis. Um padrão pode ser definido como sendo uma afirmação sobre uma distribuição probabilística. Esses padrões podem ser expressos principalmente na forma de regras, fórmulas e funções, entre outras.

Sobre a definição do termo *data mining*, John (1997) destaca seus diferentes usos nas comunidades acadêmicas e no mercado. Segundo o autor, pesquisadores tipicamente referiam-se ao processo completo envolvendo mineração de dados como Descoberta de Conhecimento em Bases de Dados (do termo em inglês *KDD – Knowledge Discovery in Databases*), também citado por Fayyad et al. (1996). No entanto, no mercado e na indústria

popularizou-se a utilização do termo *data mining*, como referência ao processo completo. Para John (1997), esta disparidade no uso do termo está sendo amenizada com a adoção do termo "mineração de dados" também na comunidade acadêmica como referência ao conjunto de ações de descoberta de conhecimento em bases de dados (e será desta forma que o termo "mineração de dados" será utilizado nesta dissertação).

O interesse crescente no tema mineração de dados deve-se sobretudo ao fato de as empresas e organizações estarem coletando e armazenando grandes quantidades de dados como conseqüência da redução dos preços de meios de armazenamento e computadores e do aumento da capacidade de ambos (FAYYAD et al., 1996).

A popularização na utilização de armazéns de dados, ou *data warehousing* (grandes bancos de dados criados para análise e suporte à decisão), tende a aumentar ainda mais a quantidade de informações disponível. Os métodos tradicionais de análise de dados, como planilhas e consultas, não são apropriados para tais volumes de dados, pois podem criar relatórios informativos sobre os dados, mas não conseguem analisar o conteúdo desses com a finalidade de obter conhecimentos importantes (FAYYAD et al., 1996).

Almeida (1995) afirma que o fato de uma empresa dispor de certas informações possibilita-lhe aumentar o valor agregado de seu produto ou reduzir seus custos em relação àquelas que não possuem o mesmo tipo de informação. Assim, as informações e o conhecimento compõem um recurso estratégico essencial para o sucesso da adaptação da empresa em um ambiente de concorrência.

Oliveira (1997) diz que toda empresa tem informações que proporcionam a sustentação para as suas decisões, entretanto apenas algumas conseguem otimizar o seu processo decisório e aquelas que estão neste estágio evolutivo seguramente possuem vantagem empresarial.

As ferramentas de mineração de dados, por definição, devem trabalhar com grandes bases de dados e retornar, como resultado, conhecimento novo e relevante; porém, o retorno gerado por este tipo de ferramenta deve ser criteriosamente avaliado, principalmente devido às inúmeras relações e equações geradas, o que pode tornar impossível o processamento dos dados.

Outra promessa em relação a esta tecnologia de informação diz respeito à forma como elas exploram as inter-relações entre os dados. Segundo Figueira (1998), as diversas ferramentas de análise disponíveis dispõem de um método baseado na verificação, isto é, o usuário constrói hipóteses sobre inter-relações específicas e então verifica ou refuta estas hipóteses, através do sistema. Esse modelo torna-se dependente da intuição e habilidade do analista em propor hipóteses interessantes, em manipular a complexidade do espaço de atributos, e em refinar a análise, baseado nos resultados de consultas potencialmente complexas ao banco de dados. Já o processo de mineração de dados, para o autor, ficaria responsável pela geração de hipóteses, garantindo mais rapidez e resultados.

Segundo Carvalho (2002), para ocorrer aprendizado sobre uma base de dados, uma série de informações de diferentes formatos e fontes precisa ser organizada de maneira consistente na grande memória empresarial (*Data Warehouse*). Após isto, métodos de análise estatística e de Inteligência Artificial precisam ser aplicados sobre esses dados e novas e úteis relações à empresa devem ser descobertas, ou seja, os dados devem ser minerados (*data mining*). Sobre o enfoque do autor, a mineração de dados consiste em descobrir relações entre produtos, classificar consumidores, prever vendas, localizar áreas geográficas potencialmente lucrativas, prever ocorrências ou inferir necessidades.

Por isso diversas ferramentas têm sido usadas para examinar os dados que possuem, no entanto, a maioria dos analistas tem reconhecido que existem padrões, relacionamentos e regras ocultas nestes dados que não podem ser encontrados utilizando estes métodos tradicionais. Para Gonçalves (2001) a resposta é usar softwares de mineração de dados que utilizam algoritmos matemáticos avançados para examinar grandes volumes de dados detalhados.

A necessidade de transformar a grande quantidade de dados armazenados em informações significativas é óbvia, entretanto, a sua análise ainda é demorada, dispendiosa, pouco automatizada e sujeita a erros, mal entendidos e falta de precisão (Gonçalves, 2001). A automatização dos processos de análise de dados, com a utilização de softwares ligados diretamente à massa de informações, tornou-se uma necessidade (Figueira, 1998).

O mercado de mineração de dados tem crescido consideravelmente nos últimos anos. No entanto, Fayyad et al. (1996), chama a atenção para o fato de que existem poucas ferramentas de mineração de dados bem desenvolvidas; o autor salienta que a maioria delas

não foi testada em uma variedade de ambientes, que a maioria não é robusta quanto à falta de dados e ao surgimento de erros, e que não está claro o quanto elas podem ser utilizadas por outras pessoas – que não sejam os seus desenvolvedores.

Desde então, as ferramentas têm apresentado evolução quanto à robustez e versatilidade, mas ainda exigem conhecimento especialista e validação. Daí a importância do uso de ferramentas disponíveis no mercado sobre bases de dados reais, testando o modelo e a aplicação com informações verdadeiras, provenientes de sistemas em produção e com histórico de dados armazenados.

3.2 Fundamentação da Mineração de Dados

3.2.1 Estatística

Mineração de Dados (ou *Data Mining – DM*) descende fundamentalmente de três linhagens. A mais antiga delas é a estatística clássica. Sem a estatística não seria possível termos o *DM*, uma vez que é a base da maioria das tecnologias a partir das quais o *DM* é construído.

A Estatística Clássica envolve conceitos, como distribuição probabilística, média, mediana, moda, variância, covariância, assimetria, curtose, correlação, análise de resíduos, análise de conjuntos e intervalos de confiança, todos usados para estudar dados e os relacionamentos entre eles (JOHNSON, 1998). Estes são os elementos fundamentais onde avançadas análises estatísticas se apóiam. No núcleo das atuais ferramentas e técnicas de *DM*, a análise estatística clássica desempenha um papel fundamental (CISTER, 2005).

3.2.2 Inteligência Artificial

A segunda linhagem do *DM* é a Inteligência Artificial (IA). Essa disciplina, que é construída a partir dos fundamentos da heurística, em oposto à estatística, tenta imitar a maneira como o homem pensa na resolução dos problemas estatísticos.

De acordo com Cister (2005), em função dessa abordagem, ela requer um impressionante poder de processamento, que era impraticável até os anos 80, quando os computadores começaram a oferecer um bom poder de processamento a preços mais acessíveis.

A IA desenvolveu algumas aplicações para o alto escalão do governo / cientistas americanos, sendo que os altos preços não permitiram que ela ficasse ao alcance de todos. As notáveis exceções foram, certamente, alguns conceitos de IA adotados por alguns produtos de ponta, como módulos de otimização de consultas para SGBD - Sistemas de Gerenciamento de Banco de Dados (COUTINHO, 2003).

Todavia, o custo / benefício das atuais aplicações torna a metodologia acessível para uma grande maioria das empresas, principalmente no Brasil. Atualmente, pequenas e médias brasileiras já dispõem de softwares de IA para descobrimento de conhecimento, principalmente no que tange ao descobrimento do perfil de clientes inseridos em base de dados ou sistemas de relacionamento (CISTER, 2005).

3.2.3 Aprendizado de Máquina

A terceira linhagem da mineração de dados é o chamado Aprendizado de Máquina - AM (ou *Machine Learning* - *ML*), que pode ser melhor descrito como a junção de melhores práticas de estatística e Inteligência Artificial.

Enquanto a IA não se transformava em sucesso comercial, suas técnicas foram sendo largamente cooptadas pelo Aprendizado de Máquina, que foi capaz de se valer das sempre crescentes taxas de preço / performance oferecidas pelos computadores nos anos 80 e 90, conseguindo ampliar a quantidade de aplicações devido às suas combinações entre heurística e análise estatística.

O Aprendizado de Máquina baseia-se no intuito de fazer com que os programas de computador “aprendam” com os dados que eles estudam, tal que esses programas tomem decisões diferentes baseadas nas características dos dados estudados, usando a estatística para os conceitos fundamentais e adicionando mais heurística avançada da IA e algoritmos para alcançar os seus objetivos.

De muitas formas, a mineração de dados é fundamentalmente a adaptação das técnicas de Aprendizado de Máquina para as aplicações de negócios. Desse modo, podemos descrevê-lo como a união dos históricos e dos recentes desenvolvimentos em estatística, em IA e Aprendizado de Máquina. Essas técnicas são usadas juntas para avaliar os dados e encontrar tendências e padrões. Atualmente, a mineração de dados tem experimentado uma crescente aceitação nas ciências e nos negócios, que precisam analisar grandes volumes de dados e encontrar tendências que não poderiam ser avaliadas de outra forma (COUTINHO, 2003).

3.3 O Ciclo da Descoberta de Conhecimento em Bases de Dados

Segundo Ferreira (2005), o objetivo último da descoberta do conhecimento em bases de dados não é o de simplesmente encontrar padrões e relações em meio à imensa quantidade de informação disponível em bases de dados, mas sim a extração de conhecimento inteligível e imediatamente utilizável para o apoio às decisões.

A origem diversa dos dados que serão utilizados, coletados em diferentes instantes de tempo em lugares distintos, cria um esforço inicial de consolidação e agrupamento de toda a informação que irá servir de base para o processo. A compreensão do negócio e do ambiente no qual os dados estão inseridos é crítica para o entendimento dos mesmos.

Em função da diversidade e heterogeneidade dos dados, esforços de pré-processamento e limpeza dos mesmos são cruciais na geração de dados que possam vir a ser trabalhados em busca de conhecimento útil. É essencial que seja realizada a investigação de inconsistências e problemas devido a diferenças de escalas, assim como o tratamento de valores fora da normalidade (*outliers*) e observações errôneas (FERREIRA, 2005).

Realizadas essas tarefas iniciais, que tornam os dados tratáveis e homogêneos, a mineração dos dados pode ser iniciada, na busca por padrões e relações que façam sentido e sejam úteis para o problema a ser resolvido ou objetivo a ser alcançado. Finalmente, a interpretação, compreensão e aplicação dos resultados encontrados é o passo que torna o conhecimento adquirido através de bases de dados um real insumo para o apoio às decisões (FERREIRA, 2005).

Existem diferentes formas para definir um alvo de mineração de dados e atividades de descoberta de conhecimento, dependendo basicamente da generalização do problema e das expectativas relacionadas ao seu tratamento. Segundo Cios (2000), pode-se distinguir diferentes níveis de mineração de dados:

- **não-direcionado ou mineração de dados pura**: o cenário mais genérico encontrado. O pesquisador normalmente formula seu problema de forma bastante genérica, como "busque algo interessante em meus dados". Não existem restrições no sistema, mas também não existem indicações do que o pesquisador pode esperar e que tipo de descoberta pode ser retornada. Existe uma expectativa para o encontro de padrões inesperados mas também ocorre o risco de reprodução de padrões já conhecidos, que não trarão nenhuma informação nova ao processo;

- **mineração de dados direcionada:** neste modelo, o ponto focal torna-se mais específico. Como exemplo de aplicação, o pesquisador pode ter o interesse em descobrir perfis de consumo em um grupo particular de clientes ("*caracterize clientes compradores de itens de maior valor*");

- **teste de hipóteses e refinamento:** neste caso, a pesquisa torna-se ainda mais específica, provendo algumas hipóteses e aguardando a validação por parte do sistema e o refinamento das descobertas, ou, em caso de insucesso, modificando as hipóteses e desenvolvendo outras versões do conjunto de hipóteses. Um exemplo deste tipo de mineração do ponto de vista do pesquisador pode ser descrito assim: "*existe uma correlação positiva entre as vendas de computadores pessoais e as vendas de câmeras de vídeo no último mês, certo?*".

Segundo Klösgen et al. (2002), a busca por conhecimento em dados massivos, com o uso de diferentes espaços de hipóteses, é a fase central e necessária intrínseca ao processo. Diferentes métodos foram desenvolvidos para o gerenciamento de muitas tarefas, mas a inferência de hipóteses e verificação é apenas uma parte de todo o processo de *KDD* (e, em alguns casos, a mais curta das tarefas envolvidas). O processo ou ciclo completo é composto de várias fases, como descrito a seguir.

De forma similar (com alterações em terminologias ou descrições de etapas), Fayyad et al. (1996), Cabena et al. (1997), Cios (2000) e Klösgen et al. (2002) definem o ciclo de Descoberta de Conhecimento em Bases de Dados (ou ciclo de KDD). A estrutura de formação do ciclo de *KDD* proposta por Fayyad et al. (1996) é ilustrada na Figura 3.1.

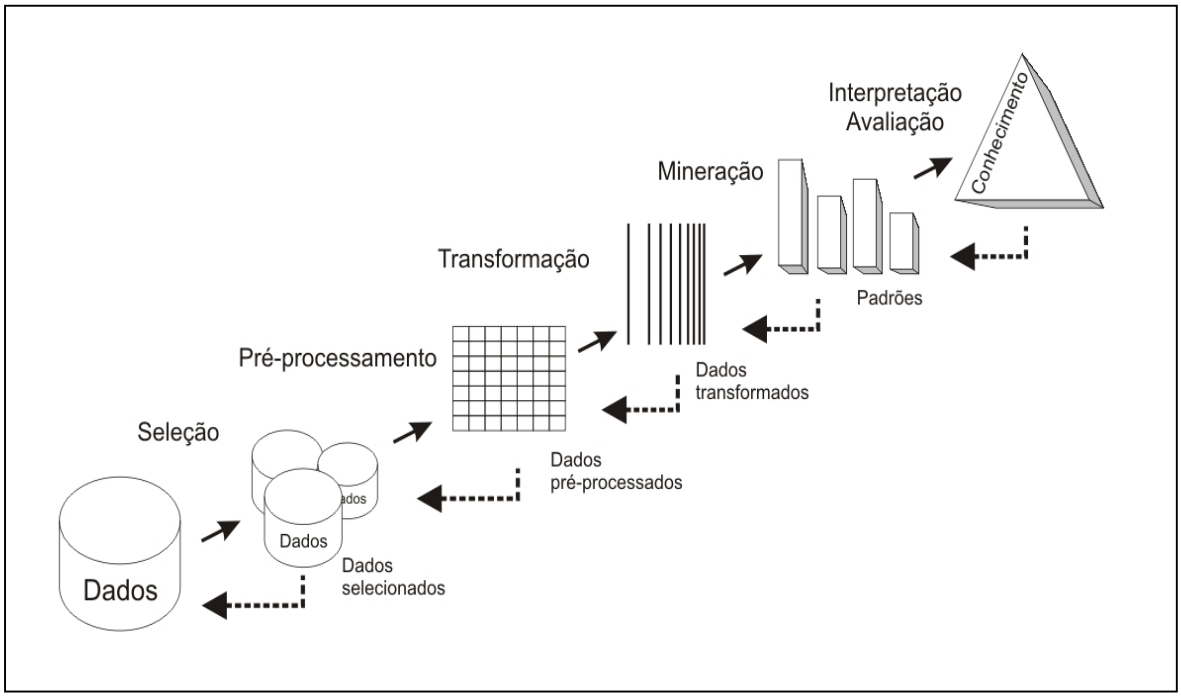

Figura 3.1: ciclo de Descoberta de Conhecimento em Bases de Dados
Fonte: Fayyad et al. (1996)

A estrutura deste trabalho utilizará uma composição mista para a revisão sobre o ciclo de *KDD*, referenciando basicamente a proposta de Fayyad et al. (1966), mas agregando itens do modelo de Klösgen et al. (2002), ilustrado na Tabela 3.1 - que se diferencia do modelo proposto por Fayyad et al. (1996) mais na apresentação e organização, do que no conteúdo e método.

1	Definição e análise do problema
2	Entendimento e preparação dos dados
3	Configuração da busca por conhecimento
4	Busca de conhecimento
5	Refinamento das descobertas
6	Aplicação do conhecimento na resolução do problema
7	Depuração e avaliação dos resultados

Tabela 3.1: ciclo completo de KDD
Fonte: Klösgen et al. (2002, p. 10)

3.3.1 Definição e análise do problema

Entre as diversas aplicações possíveis para o processo de Descoberta de Conhecimento em Bases de Dados, os casos relacionados ao *marketing* (principalmente com o foco em predição e análise de comportamentos de clientes) pertencem à linha de frente de aplicações de sucesso em *KDD*. Interesses típicos de negócios são perfis de clientes e produtos (quais clientes desejam determinado produto), predição de lealdade de clientes e retenção (que é o objetivo principal deste trabalho) e avaliação da efetividade de campanhas de venda ou campanhas de *marketing* direcionadas (KLÖSGEN et al., 2002).

A forte pressão dos concorrentes, a saturação dos mercados e a maturidade dos produtos e serviços ocasionam uma transição da competição por qualidade para a competição por informação, onde o conhecimento compreensivo e detalhado do comportamento dos consumidores e da concorrência é crucial. Mas o simples conhecimento que leva à predição e à classificação também é útil.

Na área financeira, o processo de *KDD* é utilizado, por exemplo, para a predição de retorno sobre um *portfolio* de investimento, determinar o potencial de crédito de um cliente, detectar fraudes em cartões de crédito ou lavagem de dinheiro.

A identificação e o entendimento de comportamento dinâmico é outra aplicação comum em diversas áreas, como a identificação de falhas em redes de telecomunicações, a análise de interatividade de usuários de um *web site*, mineração de texto e áudio em e-mails e chamadas de telefone. Como exemplos de problemas científicos, pode-se citar a exploração de estruturas de dados complexas e a análise de grandes conjuntos de seqüências de DNA e outras informações genéticas (KLÖSGEN et al., 2002).

3.3.2 Entendimento dos dados e pré-processamento

Métodos de *KDD* podem prover conhecimento utilizável na resolução de problemas apenas quando este conhecimento puder ser gerado pelos dados disponíveis. Duas questões precisam ser respondidas positivamente (KLÖSGEN et al., 2002):

- Dados relevantes ao problema estão disponíveis?

- A generalização dos dados pode ser aplicada em questões objetivas?

Respondendo a primeira questão, para cada tabela de dados e para cada atributo deve-se decidir quais atributos são mais relevantes. Se for desejável a predição para o atributo *C*, este dado necessita constar no conjunto, como também um número potencial de atributos de predição cujos valores são conhecidos antes que o processo de predição de *C* seja realizado. Deve-se encontrar uma maneira de agrupar dados de diferentes fontes ou tabelas em um conjunto único, devido à preponderância que as ferramentas de descoberta apresentam quando aplicadas a tabelas únicas (KLÖSGEN et al., 2002).

O modelo deve garantir que o volume de dados tenha alcance sobre a variedade de resultados das variáveis relevantes. Em particular, os dados devem estar disponíveis para contrastar diversas generalizações com apropriados grupos de controle. Se for considerada a efetividade de algum tratamento para uma necessidade particular, então se necessita dados em elementos tratados e não tratados.

Uma vez que o problema foi definido, dados relevantes precisam ser coletados. Na maioria dos casos, este conjunto de informações é extraído de um banco de dados operacional, ou de um *Data Warehouse* que foi originalmente criado com o intuito de atender a diversas demandas analíticas. Então, as informações são extraídas de uma base de dados relacional e armazenada em um formato que possa ser acessado e interpretado por algoritmos de mineração. Neste caso, o processo de extração e formação manual desta base é apenas um contorno sobre a inabilidade que as ferramentas atuais de mineração de dados possuem sobre um acesso direto sobre o banco de dados (JOHN, 1997).

Em alguns casos, a melhor estratégia é a formação de uma base de dados completamente nova, mas isto pode ter um custo proibitivo associado. Na retenção de clientes, por exemplo, a companhia provavelmente não terá uma única base de dados, conforme ilustrado na Tabela 3.2.

Nome	Endereço	CEP	Tempo de contrato	Serviços	...	Retido?
Jones	51 Rua Elm, 51	94305	3,25	A		S
Davis	Rua Thayer, 14	82138	0,75	ACD		N
⋮						

Tabela 3.2: fragmento de exemplo de uma base de dados de retenção de clientes
Fonte: John (1997)

Tomando como exemplo o fragmento ilustrado na Tabela 3.2, pode-se considerar que os dados representam uma base de clientes de uma operadora de telefonia, e que o objetivo da mineração de dados é prevenir a perda de clientes. Na indústria, o problema da troca (por vezes repetida) de fornecedor de serviço é conhecida como *churn*. Ao invés da disponibilidade direta de informações como a Tabela 3.2, a companhia provavelmente possuiu diversos conjuntos de tabelas com dados relevantes distribuídos (JOHN, 1997).

Cada divisão da companhia coleta dados necessários para as suas atividades, mas o resultado é que não existe uma fonte única para ser utilizada na extração de todas as informações relevantes sobre um cliente. Surge a necessidade da construção de uma nova base de dados, para atender especialmente o problema em questão.

No exemplo da Tabela 3.2, para cada cliente tem-se o nome, o endereço, o CEP, seu tempo de contrato, os serviços assinados e uma variável indicando se ele permaneceu ou não como usuário da companhia. O nome, endereço e o CEP estão, provavelmente, armazenados em uma tabela do departamento de cadastro da empresa. Dados geográficos costumam ser importantes, porque pessoas com estágios de vida (estado civil, idade) e estilos de vida (faixa salarial, padrões de gasto) tendem a residir próximos uns aos outros (JOHN, 1997). Bases de dados adicionais de mapeamento de endereço / CEP para coordenadas geográficas também poderiam ser utilizadas no modelo, referenciando indicadores demográficos.

O tempo de contrato é uma informação que pode ser extraída de uma base de dados do sistema de faturamento. Pode-se identificar, como informações adicionais, se o cliente costuma pagar as faturas em dia, principalmente nos últimos meses de referência. A mesma base pode ser utilizada para extrair informações como: se o cliente possui algum plano de desconto, a quantidade de vezes em que foi aberto algum chamado de atendimento ou registro de reclamação.

O procedimento descrito de extração e coleção dos dados é apenas ilustrativo. Nesta etapa, não foram definidos quais os atributos podem ser adquiridos em dados reais em um projeto de mineração de dados (e se tais atributos podem ser encontrados em algum conjunto de informações disponível). No cenário descrito, poderia utilizar-se uma variável que indicasse o pagamento das contas no prazo, especialmente nos últimos meses. Assume-se que o pagamento de uma conta em atraso pode significar uma dificuldade financeira, um

descontentamento do cliente com o serviço ou ambos. Existem formas diferentes para a inserção deste tipo de informação na base do modelo. Uma das alternativas é fixar o número de meses (12, por exemplo) e criar o mesmo número de atributos, cada um deles contendo a quantidade em dias (de antecedência ou de atraso) que o cliente efetuou o pagamento da conta. Ou, em uma segunda alternativa, sumarizar em apenas um atributo, a quantidade de vezes em que a fatura foi paga em atraso. Assim, assume-se que o comportamento de pagamento mais recente pode indicar a opção de permanência do cliente. No entanto, estas alternativas e possibilidades são consideradas e resolvidas no processo de formação do modelo, normalmente operacionalizado por um especialista (com conhecimento do negócio) e por um analista de mineração de dados (JOHN, 1997).

O pré-processamento de dados em um processo de *KDD* é freqüentemente tido como sendo uma fase que envolve uma grande quantidade de conhecimento de domínio. Normalmente, dados coletados diretamente de bancos de dados são de má qualidade, ou seja, possuem informações incorretas e imprecisas, além de uma grande quantidade de valores desconhecidos. Embora muitos dos algoritmos utilizados na fase de mineração de dados tenham sido projetados para manipular dados em tais situações, pode-se esperar que esses algoritmos gerem resultados mais precisos caso a maioria dos problemas presentes nos dados tenha sido removida ou corrigida (BATISTA, 2003).

Segundo Ferreira (2005), o pré-processamento visa detectar e remover anomalias presentes nos dados com o objetivo de aumentar e melhorar a sua qualidade. Tipicamente, o processo não pode ser executado sem o envolvimento de um perito no negócio ao qual correspondem os dados, uma vez que a detecção e correção de anomalias requerem conhecimento especializado.

O pré-processamento dos dados envolve uma verificação da consistência das informações, a correção de possíveis erros e o preenchimento ou a eliminação de valores nulos e redundantes. Nessa fase são identificados e removidos os dados duplicados e corrompidos. A execução dessa fase corrige a base de dados eliminando consultas desnecessárias que seriam executadas pelos modelos e que afetariam o desempenho (FERREIRA, 2005).

Um exemplo comum na limpeza de dados (que ocorre no pré-processamento) é a procura por valores absurdos que não deveriam existir na base simplesmente por serem

impossíveis na prática. Boas ilustrações disso são bases de dados que possuem idades ou tempos de contrato com clientes. Por vezes encontram-se clientes que possuem mais de 120 anos de vida, ou até mesmo clientes com menos de 2 anos de idade. Da mesma forma, encontrar consumidores que possuem um relacionamento de 400 anos com a empresa não é tão incomum assim. Esses valores são oriundos provavelmente de erros de digitação ou de preenchimento de cadastros. No esforço para limpeza e consistência dos dados, tais campos, mesmo sendo raros, devem ser preenchidos com valores possíveis, utilizando-se, por exemplo, médias ou medianas da variável. Outra opção seria a eliminação do registro que contém tal valor. A filosofia por trás dessas ações é evitar que tal valor atrapalhe a compreensão dos dados pelos modelos, levando-os a tomar conclusões errôneas (FERREIRA, 2005).

Se um dado é relevante ao modelo, sua preparação é normalmente categorizada em limpeza dos dados, seleção do modelo e redução de dados. É um processo iterativo que inclui laços de retorno e, em passos subseqüentes pode gerar informações sobre o domínio e requerendo preparação adicional dos dados (KLÖSGEN et al., 2002).

Segundo Batista (2003), de uma forma geral, pré-processamento de dados é um processo semi-automático. Por esta natureza, entende-se que essa fase depende da capacidade do analista de dados em identificar os problemas presentes nos dados e utilizar os métodos mais apropriados para solucionar cada um dos problemas.

Batista (2003) também classifica tarefas realizadas por métodos empregados na fase de pré-processamento em dois grupos:

a) tarefas fortemente dependentes de conhecimento de domínio: essas tarefas somente podem ser efetivamente realizadas com o uso de conhecimento específico ao domínio. Um método automático pode ser empregado para realizar uma tarefa fortemente dependente de conhecimento de domínio, entretanto, esse método depende de que um conhecimento específico seja fornecido. Como exemplo, cita-se as verificações de integridade de dados. Por exemplo, em uma aplicação para concessão de crédito, um atributo crédito, o qual informa o valor emprestado, não pode assumir valores negativos. Ou ainda, caso existam informações a respeito do bem adquirido por meio de um empréstimo, esse atributo pode possuir faixas de valores

permitidas, as quais dependem do bem adquirido. Com o uso de um conjunto de regras dependentes de domínio é possível verificar a integridade dos atributos presentes em um conjunto de dados;

b) tarefas fracamente dependentes de conhecimento de domínio: essas tarefas podem ser realizadas por métodos que extraem dos próprios dados as informações necessárias para tratar o problema de pré-processamento de dados. Se por um lado essas tarefas ainda dependem de conhecimento de domínio, pois é necessário selecionar o método correto para tratar o problema, por outro lado, essas tarefas podem ser realizadas por métodos que são mais automáticos do que aqueles utilizados em tarefas que dependem fortemente de conhecimento de domínio. São exemplos de tarefas fracamente dependentes de domínio o tratamento de valores desconhecidos (ou ausentes) e a identificação de valores extremos (também denominados *outliers*).

Apresenta-se a seguir as principais tarefas de pré-processamento de dados fortemente dependentes de domínio. Tais tarefas serão aplicadas no modelo a ser implementado no estudo de caso deste trabalho, visto que a base de dados em questão será modelada com informações provenientes de fontes diversas e sujeita a erros e inconsistências típicas de um conjunto de dados desta natureza.

3.3.2.1 Identificação de inconsistências

Inconsistências podem ocorrem quando dados diferentes são representados pelo mesmo rótulo, ou quando o mesmo dado é representado por rótulos diferentes. Um exemplo de inconsistência ocorre quando um atributo assume diferentes valores, os quais representam, na verdade, uma mesma informação. Por exemplo, um atributo “nome_empresa”, que armazena nomes de empresas, assume os valores PUC, Puc-RS, Pontifícia Universidade Católica, etc, sendo que todos esses valores representam uma mesma instituição (BATISTA, 2003).

3.3.2.2 Identificação de poluição

Existem diversas fontes de poluição de dados. De certa forma, pode-se entender por poluição a presença de dados distorcidos, os quais não representam os valores verdadeiros.

Segundo Batista (2003), uma possível fonte de poluição de dados é a tentativa, por parte dos usuários do sistema que coletam os dados, de utilizar esse sistema além da sua funcionalidade original. Por exemplo, o caso de uma empresa de cartão de crédito cujo banco de dados possuía um campo "*gender*" para armazenar o sexo de seus clientes. Entretanto, alguns registros assumiam o valor "B" para esse atributo, o qual, posteriormente, descobriu-se que correspondia à informação "*Business*". Originalmente, o sistema tinha sido projetado somente para cadastrar cartões para pessoas físicas, porém, quando cartões para empresas foram permitidos, não havia um campo específico para indicar que o cadastrado era uma empresa. Essa informação foi então armazenada no campo "*gender*".

Um segundo motivo que pode gerar poluição nos dados é a resistência humana em entrar com os dados corretamente. Enquanto que campos em um banco de dados podem ser incluídos para capturar informações valiosas, esses campos podem ser deixados em branco, incompletos ou simplesmente com informações incorretas (BATISTA, 2003).

3.3.2.3 Verificação de integridade

Analisar a integridade dos dados freqüentemente envolve uma analise das relações permitidas entre os atributos. Por exemplo, um empregado pode possuir vários carros, entretanto, um mesmo empregado não pode possuir mais de um número funcional em um dado sistema. Dessa forma, é possível analisar os atributos por meio de faixa de valores válidos.

Um caso especial de verificação de integridade de dados é a identificação de casos extremos. Casos extremos são casos em que a combinação dos valores é válida, pois os atributos estão dentro de faixas de valores aceitáveis, entretanto, a combinação dos valores dos atributos é muito improvável. A identificação de casos extremos pode ser considerada

uma tarefa fracamente dependente de domínio, pois a probabilidade das combinações de valores de atributos pode ser feita a partir dos dados disponíveis.

3.3.2.4 Identificação de atributos duplicados e redundantes

Redundância ocorre quando uma informação essencialmente idêntica é armazenada em diversos atributos. Como exemplo, cita-se o caso de uma mesma tabela possuir atributos como preço por unidade, unidades compradas e preço total da compra (que é uma informação que pode ser calculada em tempo real, se necessário, evitando o armazenamento direto do atributo). O maior dano causado pela redundância para a maioria dos algoritmos utilizados na fase de mineração de dados é um aumento no tempo de processamento (BATISTA, 2003).

Entretanto, alguns métodos são especialmente sensíveis ao número de atributos, e variáveis redundantes podem comprometer seus desempenhos. Se o problema de coletar atributos redundantes não for solucionado durante a fase de coleta de dados, existe a possibilidade de utilizar métodos de pré-processamento de dados, conhecidos como métodos de seleção de atributos, para tentar identificar e remover os atributos redundantes.

Klösgen et al. (2002) exemplificam outra situação de atributos duplicados ou redundantes. Segundo o autor, se uma aplicação de mineração de dados tem como objetivo a predição baseada em padrões, então é essencial que todas as informações de determinado cliente, por exemplo, estejam associadas ao mesmo indivíduo na base de dados. Entretando, erros de entrada de dados, problemas cadastrais e outras ocorrências, podem resultar na duplicação de dados para um mesmo indivíduo. Steve Jones, Steven Jones e Stephen Jones podem ser a mesma pessoa, mas seus dados podem estar distribuídos em três registros diferentes, fazendo com que a aplicação perca padrões importantes.

3.3.2.5 Valores padrão (defaults)

A maioria dos sistemas gerenciadores de banco de dados permite valores *defaults* ou padrão para alguns atributos. Esses valores podem causar algumas confusões, especialmente se o analista de dados não está informado a respeito. Um valor *default* pode

estar ligado condicionalmente a outros atributos, o que pode criar padrões significantes à primeira vista. Na realidade, tais valores *defaults* condicionais simplesmente representam falta de informações, em vez de informações relevantes. Um exemplo é a área médica, na qual os valores de um atributo, como por exemplo, período de gravidez, está condicionalmente ligado a valores de outros atributos, como por exemplo, sexo. Valores *defaults* podem ser especialmente perigosos quando o usuário está interessado em uma análise preditiva (BATISTA, 2003).

As tarefas de pré-processamento que são incluídas na classe descrita como fracamente dependente do domínio de aplicação, podem ser tipicamente solucionadas por métodos que extraem do próprio conjunto de dados as informações necessárias para tratar o problema. A seguir, descrevem-se as principais tarefas deste domínio.

3.3.2.6 Tratamento de valores desconhecidos

Um problema comum em pré-processamento de dados é o tratamento de valores desconhecidos. Muitas técnicas têm sido aplicadas, sendo algumas delas bastante simples, como a substituição dos valores desconhecidos pela média ou moda do atributo. Entretanto, outras técnicas mais elaboradas podem ser implementadas e avaliadas experimentalmente. Por exemplo, pode-se substituir os valores desconhecidos por valores preditos utilizando um algoritmo de aprendizado.

O tratamento de valores desconhecidos deve ser tratado com atenção, especialmente em casos envolvendo predição. Normalmente, os algoritmos e ferramentas de predição não demonstram resultados efetivos sobre bases com valores desconhecidos (WEISS et al., 1998).

A utilização de técnicas simples, como por exemplo a substituição de todos os valores desconhecidos por uma única variável global, pode trazer resultados indesejados, desvirtuando o processo de predição – neste caso, um valor desconhecido será interpretado em um fator positivo que não poderá ser justificado (WEISS et al., 1998).

3.3.2.7 Tratamento de conjuntos de dados com classes desbalanceadas

Conjuntos de dados com classes desbalanceadas são aqueles que possuem uma grande diferença entre o número de exemplos pertencentes a cada valor de um atributo classe qualitativo. A maioria dos algoritmos tem dificuldades em criar um modelo que classifique com precisão os exemplos da classe minoritária. Uma forma de solucionar esse problema é procurar por uma distribuição da classe que forneça um desempenho aceitável de classificação para a classe minoritária (BATISTA, 2003).

Ferreira (2005) cita o caso prático de um conjunto de dados com classes desbalanceadas. Em seu estudo referente a uma base de telefonia celular, ele cita uma base de dados que pode possuir uma variável que denota se um cliente deixou ou não a empresa. Essa variável, em geral, possui algo em torno de 98% dos clientes como os que continuam na empresa e somente 2% dos clientes como os que terminaram sua relação com a operadora. Segundo Ferreira (2005), o que acontece no momento da construção de qualquer modelo envolvendo uma variável deste tipo é que, dada essa distribuição desequilibrada entre as classes, o modelo terminará enxergando somente uma das classes, sendo incapaz de distinguir a classe de menor número de registros. Isso acontece porque o modelo reconhece que, se sua resposta for sempre dizer que todas as observações pertencem à classe com maior número de registros, ele acertará 98% dos padrões.

Para evitar esse problema e facilitar a distinção de classes, Ferreira (2005) utiliza o procedimento conhecido como *oversampling*. Através do *oversampling* cria-se uma nova base de dados para a modelagem, selecionando-se aleatoriamente um maior número de registros pertencentes à classe rara e um menor número de ocorrências da classe comum, desta forma ajustando a proporção entre as classes. No caso de variáveis binárias, por exemplo, em geral se deseja que a classe rara corresponda a algo entre 10% e 40% da base para a modelagem. Frequências entre 20% e 30% são desejáveis para melhores resultados.

Infelizmente, o processo de *oversampling* possui limitações. Dado que só existe um pequeno número de observações da classe rara na base de dados, não é possível criar uma base de qualquer tamanho para a análise, mesmo que a base de dados original seja imensa. Por exemplo, em uma base de 100.000 registros, se somente 2% pertencem a uma classe, isso significa que só estão disponíveis 2000 amostras desta certa classe. Sendo assim, é impossível construir uma base para modelagem com, por exemplo, 50.000 registros e uma freqüência

maior do que 4% para a classe rara. É necessário, para atingir as proporções entre 10% e 40% mencionadas, que a base de dados seja bastante diminuída no que diz respeito às observações da classe comum (FERREIRA, 2005).

3.3.2.8 Seleção de atributos

Seleção de atributos é um problema importante em *KDD*. Consiste em encontrar um subconjunto de atributos no qual o algoritmo de aprendizado utilizado em mineração de dados irá se concentrar.

Posteriormente, é realizada uma combinação do conhecimento adquirido pelos algoritmos de Aprendizado de Máquina (AM) usando essas amostras, tendo esta metodologia mostrado-se promissora (BARANAUSKAS, 1998).

Algumas das razões para a aplicação de métodos de seleção de varáveis - ou atributos (BARANAUSKAS, 1998):

a) muitos algoritmos de mineração de dados não funcionam bem com uma grande quantidade de atributos, dessa forma, a seleção de atributos pode melhorar o desempenho do modelo;

b) com um número menor de atributos, o conhecimento induzido por algoritmos de mineração de dados é, freqüentemente, mais compreensível;

c) alguns domínios possuem um alto custo de coletar dados, nesses casos, métodos de seleção de atributos podem diminuir o custo da aplicação.

Os algoritmos de AM comumente utilizam *datasets* (ou conjunto de dados) contendo poucos exemplos ($n<20.000$) com número restrito de atributos ($a<30$). Quando se trabalha com uma dimensão restrita é possível usar um algoritmo de seleção de atributos que simplesmente procura pelas possíveis combinações, selecionando aqueles que melhoram a taxa de classificação do algoritmo.

Segundo Dietterich (1997), mesmo considerando o fato que os algoritmos de AM possam ser utilizados com muitos atributos, sabe-se que o desempenho dos indutores e redes neurais artificiais é prejudicado quando existem muitos atributos irrelevantes. Além disso,

estatisticamente, exemplos com muitos atributos e ruídos fornecem pouca informação (BARANAUSKAS, 1998).

Existem diferentes abordagens propostas para selecionar um subconjunto de atributos. De uma forma geral, pode-se dividir as abordagens mais utilizadas em pré-processamento de dados em três grupos (BARANAUSKAS, 1998):

a) Embutida: a abordagem embutida consiste na seleção de atributos realizada como parte do processo de criação do modelo;

b) Filtro: a abordagem filtro consiste em aplicar um método de seleção de atributos anterior à aplicação do algoritmo de mineração de dados ou aprendizado de máquina, geralmente analisando características do conjunto de exemplos que podem levar a selecionar alguns atributos e excluir outros;

c) *Wrappers*: a abordagem *wrapper* consiste em selecionar um subconjunto de atributos e medir a precisão do classificador induzido sobre esse subconjunto de atributos. É realizada uma busca pelo subconjunto que gera o classificador com menor erro. Essa busca avalia cada subconjunto candidato, até que o critério de parada, relacionado com a precisão do classificador induzido, seja satisfeito.

3.3.2.9 Construção de atributos

Os atributos podem ser considerados inadequados para a tarefa de aprendizado quando são fracamente ou indiretamente relevantes, condicionalmente relevantes ou medidos de modo inapropriado (BARANAUSKAS, 1998).

Se os atributos utilizados para a descrição do conjunto de dados são inadequados, os algoritmos de aprendizado utilizados em mineração de dados provavelmente criarão classificadores imprecisos ou excessivamente complexos. Atributos fracamente, indiretamente ou condicionalmente relevantes podem ser individualmente inadequados, entretanto, esses atributos podem ser convenientemente combinados gerando novos atributos que podem mostrar-se altamente representativos para a descrição de um conceito. O processo de construção de novos atributos é conhecido como construção de atributos ou indução construtiva (BLOEDORN et al., 1998).

Assim, construção de atributos é o processo de composição de atributos ditos primitivos, produzindo-se novos atributos possivelmente relevantes para a descrição de um conceito. De uma forma bastante ampla, o processo de indução construtiva pode ser dividido em duas abordagens: a automática e a guiada pelo usuário. A indução construtiva automática consiste em um processo de construção de atributos guiada automaticamente pelo método de construção. Geralmente, os atributos construídos são avaliados em relação aos dados, e podem ser descartados ou integrados ao conjunto de dados. A indução construtiva guiada pelo usuário utiliza o conhecimento do usuário ou do especialista no domínio para guiar a composição dos atributos (BATISTA, 2003).

Sob a ótica da natureza das bases de dados utilizadas neste trabalho, a indução construtiva será utilizada para a definição do *dataset* final. Exemplificando esta técnica na prática e no modelo a ser definido neste trabalho, pode-se considerar a variável "Fat_medio" (faturamento médio, considerando o histórico de contas do cliente nos últimos seis meses), que é construída de acordo com um conjunto anterior de atributos.

3.3.3 Transformação dos dados

O principal objetivo desta fase é transformar a representação dos dados a fim de superar quaisquer limitações existentes nos algoritmos que serão empregados para a extração de padrões. De uma forma geral, a decisão de quais transformações são necessárias depende do algoritmo que será utilizado na fase de mineração de dados.

Em geral uma transformação nos dados envolve a aplicação de alguma fórmula matemática ao conteúdo de uma variável, buscando obter os dados em uma forma mais apropriada para a posterior modelagem, maximizando a informação, satisfazendo premissas de modelos ou simplesmente prevenindo erros (FERREIRA, 2005).

Além disso, o processo visa também atender à restrições impostas pelos algoritmos de mineração de dados. Determinadas ferramentas podem ser aplicadas apenas a conjuntos de dados com atributos nominais, enquanto outros algoritmos conseguem inferir e descobrir padrões apenas sobre variáveis numéricas, por exemplo.

A seguir, descrevem-se as técnicas de transformação de dados mais comuns.

3.3.3.1 Normalização

Consiste em transformar os valores dos atributos de seus intervalos originais para um intervalo específico, como, por exemplo, [-1, 1] ou [0, 1]. Esse tipo de transformação é especialmente valioso para os métodos que calculam distancias entre atributos. Por exemplo, um método como o "k-vizinhos mais próximos" tende a dar mais importância para os atributos que possuem um intervalo maior de valores (BATISTA, 2003).

Segundo Ferreira (2005), a normalização (ou padronização) dos dados é realizada com o objetivo de homogeneizar a variabilidade dos atributos de uma base de dados, criando um intervalo de amplitude similar onde todas as variáveis irão residir. Em geral a normalização é necessária no caso de variáveis com unidades diferentes ou dispersões muito heterogêneas.

Métodos como redes neurais são reconhecidamente melhores treinadas quando os valores dos atributos são pequenos (WEISS et al., 1998). Entretanto, normalização não é de grande utilidade para a maioria dos métodos que induzem representações simbólicas, tais como árvores de decisão e regras de decisão, uma vez que a normalização tende a diminuir a compreensibilidade do modelo gerado por tais algoritmos (BATISTA, 2003).

Ferreira (2005) cita como formas comuns de normalização:

- Normalização pelo desvio padrão: $y = \frac{x - \mu}{\sigma}$ *Equação 2.1*

- Normalização pela faixa de variação: $y = \frac{x - \min}{\max - \min}$ *Equação 2.2*

Nas Equações 2.1 e 2.2, *y* representa o novo valor normalizado; *x*, o valor atual; μ e σ, a média e o desvio padrão da variável; e *max* e *min*, os valores de máximo e mínimo, respectivamente.

3.3.3.2 Discretização de atributos quantitativos

Segundo Batista (2003), muitos algoritmos possuem a limitação de trabalhar somente com atributos qualitativos. Entretanto, muitos conjuntos de dados possuem atributos quantitativos, e para que esses algoritmos possam ser aplicados é necessário utilizar algum método que transforma um atributo quantitativo em um atributo qualitativo, ou seja, em faixas de valores. Diversos métodos de discretização de atributos foram propostos pela comunidade científica. Detalhes sobre os principais métodos de discretização podem ser encontrados em Dougherty (2005), que realiza uma descrição geral destes algoritmos, dividindo-os em grupos de técnicas supervisionadas e não-supervisionadas.

3.3.3.3 Transformação de atributos qualitativos em quantitativos

Alguns algoritmos não são capazes de manipular atributos qualitativos. Dessa forma, é necessário converter os atributos qualitativos em atributos quantitativos. Existem diversas abordagens para realizar essa transformação dependendo das características e limitações de cada algoritmo. De uma forma geral, atributos qualitativos sem ordem inerente, tal como verde, amarelo e vermelho, podem ser mapeados arbitrariamente para números. Entretanto, esse mapeamento acaba por criar uma ordem nos valores do atributo que não é real. Atributos qualitativos com ordem, tal como pequeno, médio e grande, podem ser mapeados para valores numéricos de forma a manter a ordem dos valores, por exemplo pequeno = 1, médio = 2 e grande = 3 (BATISTA, 2003).

3.3.3.4 Atributos de tipos de dado complexos

A maioria dos algoritmos utilizados para extrair padrões não consegue trabalhar com tipos de dado mais complexos. Por exemplo, atributos do tipo data e hora não são normalmente analisados pela maioria dos algoritmos utilizados na fase de mineração de dados. Dessa forma, é necessário converter esses atributos para algum outro tipo de dado com o qual esses algoritmos possam trabalhar.

No caso específico dos tipos de dado data/hora, a escolha mais simples é pela conversão para o tipo inteiro. Isso pode ser feito calculando-se a diferença em dias, meses, ou qualquer outra unidade de tempo, entre os valores das datas do atributo em questão e uma data fixa. Por exemplo, um atributo data de nascimento pode ser convertido para idade calculando-se a diferença em anos entre os valores do atributo data de nascimento e a data atual (BATISTA, 2003).

Um exemplo prático da utilização desta técnica neste trabalho é o tratamento efetuado em um atributo ("dt_instalacao") que determina o tempo em que o cliente é usuário do serviço. Este atributo é extraído do banco de dados em sua forma original como uma variável em formato texto (exemplo – "07101998"). Nesta forma, o atributo não produz informação significativa aos algoritmos de predição. Assim, o tratamento inicial para esta variável é uma conversão de tipo, para um atributo reconhecível no formato data (ou *datetime*, para o banco de dados em questão). Assim, a próxima formatação deste atributo será em um formato como "07/10/1998". Uma segunda transformação no atributo envolve um cálculo de datas e a formação (ou construção indutiva) de uma nova variável – assim, forma-se a coluna "tempo_instalacao", que conterá um atributo numérico, definindo a quantidade de meses que o cliente é usuário do serviço.

Comparativamente, o novo atributo "tempo_instalacao" possuirá o valor numérico 92 (quantidade de meses em que o cliente possui o serviço – atributo que possui representatividade para um algoritmo de mineração), enquanto a variável original "dt_instalacao" apresentava a seqüência de caracteres "07101998".

3.3.3.5 Redução de dados

As vantagens teóricas da utilização de grandes conjuntos de dados para treinamento e teste são claras, mas na prática, o conjunto pode tornar-se excessivamente grande. As dimensões podem exceder a capacidade de uma ferramenta de predição, ou o tempo de processamento e produção de resultados pode oferecer um custo proibitivo (WEISS et al., 1998).

A técnica de redução de dados tem significativa importância neste trabalho. Considerando a metodologia de predição sobre a base de dados de clientes da Brasil Telecom

no estado do Rio Grande do Sul, o conjunto inicial de dados é de um tamanho não adequado para a formação de um modelo para estudo de caso, visto que a quantidade de registros prejudicaria a construção do Estudo de Caso e da análise de resultados. O tamanho excessivo da base original elevaria o custo de processamento, tornando-o proibitivo para o prosseguimento do trabalho.

Após a validação do modelo, torna-se mais fácil a aplicação das técnicas sobre o conjunto total, visto que a técnica foi testada, os resultados foram avaliados e pode-se medir e prever o custo de processamento sobre determinada base.

A Figura 3.3 ilustra a proposta de Weiss et al. (1998) para um modelo de redução de dados.

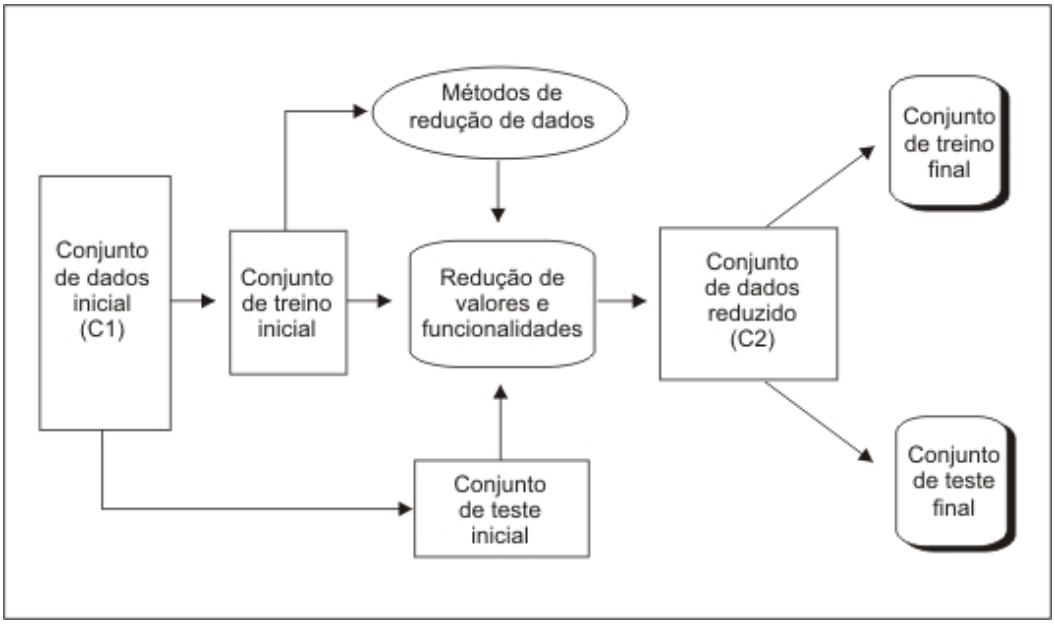

Figura 3.2: modelo de redução de dados

Fonte: Weiss et al. (1998)

Dado um conjunto (C1) de dados inicial, os dados são reduzidos em quantidade de valores (ou de atributos / funcionalidades), e um novo conjunto é produzido (C2). Enquanto as dimensões do conjunto inicial enquadrarem-se em limites aceitáveis, não é necessário a utilização de técnicas de redução de dados.

Uma vez que o conjunto inicial de dados tiver sido reduzido para um novo *dataset* (C2), os dados podem ser oficialmente divididos em casos de treino e de testes. Os casos de teste são cruciais para a avaliação dos resultados. As próximas duas fases da mineração preditiva são tarefas iterativas. Escolhas e interpretações não podem ser efetuadas sem estimativas apuradas da performance da predição (WEISS et al., 1998).

O tema principal quando se trata de simplificação dos dados é a redução da dimensão. No entanto, surge um fator importante – o descarte de dados não pode sacrificar a qualidade dos resultados.

Em Weiss et al. (1998) pode-se encontrar uma definição detalhada das principais técnicas utilizadas para um processo de redução de dados com foco em mineração de dados preditiva.

Uma abordagem de redução de dados com base em amostragem (ou *sampling*) é proposta em Klösgen (2002). Conforme o autor, o uso de rotinas e métodos de amostragem, características de populações podem ser estimadas de uma forma eficiente, com mínima distorção. Por exemplo, uma amostra aleatória simples de 1500 registros de uma população de milhões de pessoas pode estimar o percentual de opinião sobre determinado tema, com margem de erro de 3%.

Klösgen (2002) descreve os tipos principais de técnicas de amostragem:

- **amostragem simples aleatória:** abordagem direta – em um determinado conjunto de dados, todas as amostras do tamanho requerido possuem a mesma chance de serem selecionadas. Mesmo que seja possível obter-se um conjunto bastante atípico, probabilisticamente quanto maior o tamanho da amostra, mais representativo será o conjunto;

- **amostragem aleatória estratificada:** divide-se a população ou o conjunto de dados total em grupos, e então utiliza-se a amostragem simples aleatória em cada grupo. Forma-se então a população ou amostragem final, agrupando os sub-conjuntos;

- **amostragem em *Cluster*:** esta técnica é particularmente útil nos casos em que os elementos de um conjunto de dados normalmente forma um

Cluster, ou conjunto (como clientes em uma mesma localidade, empregados de uma companhia ou pacientes em hospitais);

- **amostragem sistemática:** quando indivíduos em uma população são numerados de alguma forma, a amostragem sistemática é uma opção. A técnica envolve a escolha aleatória de um membro da população para aqueles numerados entre 1 e k, e depois incluindo cada k^o membro na amostra. Devido a esta técnica não ser totalmente randômica, amostras sistemáticas podem não se tornar boas representações da população, o que exige cuidado no uso;

- **amostragem em duas fases:** esta técnica pode ser utilizada quando deseja-se organizar a amostra com base em valores de uma ou mais variáveis, mas desconhece-se a variação ou distribuição destas variáveis na população. Por exemplo, supõe-se que em uma base de dados uma determinada correlação pode variar em função da faixa etária; pode-se efetuar uma pesquisa para buscar qual grupo (faixa etária) é mais representativo na base e, após, em uma segunda etapa, auxiliar no processo decisório de utilização de uma amostragem simples aleatória ou estratificada.

3.4 Data Warehouse

Bancos de dados são de vital importância para as organizações. Mas a análise e compreensão das informações armazenadas sempre foi um grande desafio para pesquisadores e administradores. Isso porque, geralmente, as grandes empresas detêm grande volume de dados, em diversos sistemas diferentes não centralizados (CISTER, 2005).

A abordagem tradicional de armazenamento de dados é encontrada normalmente em grandes bases nas organizações, utilizadas para transações normais diárias em sistemas cliente / servidor. Estas bases de dados são conhecidas como *bases operacionais*; em muitos casos, estes bancos de dados são projetados apenas para armazenamento, não oferecendo recursos de inteligência sobre a informação armazenada (ADRIAANS, 1996).

Conforme Adriaans (1996), um segundo tipo de bases de dados encontrado atualmente nas organizações é o *Data Warehouse*. Este modelo é desenhando para prover suporte à decisões estratégicas, e é construído sobre as bases operacionais. A característica básica de um *Data Warehouse* (ou *DW*) é que o mesmo contém grande quantidade de dados armazenados – o que pode significar bilhões de registros.

Existem regras específicas que governam a estrutura básica de um *DW* (ADRIAANS, 1996):

- dependência de tempo: contém informações armazenadas ao longo do tempo, o que relaciona a informação com a época em que foi inserida. Este é um dos aspectos mais importantes de um *DW* e é diretamente relacionado com mineração de dados, porque permite que a informação seja vasculhada de acordo com por períodos de tempo;

- não-volátil: dados em um *DW* não são atualizados mas apenas utilizados para consultas. Isto significa que informações são sempre carregadas de outras fontes, como bases de dados operacionais – assim, um *DW* conterá sempre dados históricos. Em geral, ao serem encaminhados para um *Data Warehouse*, dados operacionais são limpos e transformados em uma primeira instância, principalmente para garantir que dados de diferentes fontes e formatos passem então a possuir as mesmas definições e obedeçam às mesmas regras;

- orientação ao conteúdo: a organização de um *DW* é voltada para termos de informações de negócios, e é agrupada de acordo com indicadores como clientes, produtos, relatórios de vendas ou campanhas de *marketing*;

- integração: o conteúdo armazenado em um *DW* é definido de forma que as informações seja válidas para outras aplicações na organização, incluindo outras fontes de dados.

O *DW* dispõe de habilidade para extrair, tratar e agregar dados de múltiplos sistemas operacionais em *data marts* ou *Data Warehouses* separados. Armazenam dados

freqüentemente em formato de cubo (OLAP) multidimensional, permitindo rápida agregação de dados e detalhamento das análises (*drilldown*). Disponibilizam visualizações informativas, pesquisando, reportando e modelando capacidades que vão além dos padrões de bancos de dados relacionais convencionais. O fator temporal também é um diferencial dos modelos baseados em *Data Warehouses*, visto que o modelo permite a geração de dados integrados e históricos, auxiliando no processo de tomada de decisões (CISTER, 2005).

3.5 OLAP

O método tradicional de análise de dados em uma *Data Warehouse* consiste em execução de consultas pré-definidas, normalmente utilizando SQL (*Structured Query Language*) para a geração de relatórios. Recentemente, uma nova classe de análise, denominada de *Online Analytical Processing* (OLAP) tem sido referenciada como uma forma mais natural e eficiente de visualização e operacionalização de dados armazenados em um *Data Warehouse* (KLÖSGEN et al., 2002).

A ênfase principal da metodologia OLAP é a utilização de uma visão multidimensional dos dados. Aplicações típicas de OLAP são relatórios de negócios, marketing, relatórios gerenciais, *business performance management* (BPM), *budgeting* e previsão, relatórios financeiros e áreas similares.

Bases de dados mais adequadas para OLAP empregam um modelo de base de dados dimensional, que permite consultas analíticas complexas ou *ad hoc*, com um tempo de execução pequeno.

Um software OLAP trabalha capturando uma réplica da fonte de dados e reestruturando-a em um formato de cubo. As consultas são então feitas sobre esse cubo. O cubo é criado a partir de um esquema estrela (*ou star schema*) de tabelas. No centro está a tabela de fatos (*fact table*) que lista os fatos principais de que consiste a pesquisa. Várias tabelas dimensionais estão ligadas às tabelas de fatos. Estas tabelas indicam como as agregações de dados relacionais podem ser analisadas. O número de agregações possíveis é determinado por todas as maneiras possíveis em que os dados originais podem ser conectados hierarquicamente.

Por exemplo, um conjunto de clientes pode ser agrupado por cidade, por distrito ou por país; com 50 cidades, 8 distritos e 2 países, há três níveis hierárquicos com 60 membros. Esses clientes podem ser estudados em relação a produtos; se há 250 produtos com 20 categorias, três famílias e três departamentos, então há 276 membros de produto. Com apenas duas dimensões, localização geográfica e produto, há 16.560 agregações possíveis. À medida que os dados considerados aumentam, o número de agregações pode facilmente chegar às dezenas de milhões ou mais.

O cálculo de agregações e a base de dados combinada fazem um cubo OLAP, que pode potencialmente conter todas as respostas para cada consulta que pode ser respondida com os dados. Devido ao potencial número de agregações para ser calculado, freqüentemente apenas um número predeterminado é completamente calculado enquanto o restante é resolvido sob demanda (KLÖSGEN et al., 2002).

3.6 Data Mart

A formação e configuração de um *Data Warehouse* não é uma tarefa trivial, especialmente se a intenção é fornecer suporte ao empreendimento da organização como um todo. Assim, recentemente as organizações têm optado pela formação de *data marts*, que são conjuntos de dados mais acessíveis e significativamente menores em comparação a um *Data Warehouse* completo (HAN et al., 2001).

Assim, a formação de *data marts* é uma maneira eficiente para o início da formação de um *Data Warehouse* – implicando que um *DW* pode também ser definido como um conjunto de *data marts*. Para as organizações que já possuem um *Data Warehouse* operacional, os *data marts* são ferramentas úteis para processamento especializado – por exemplo, em função da necessidade de um departamento ou de uma análise individual de dados.

3.7 Inteligência de Negócio (*Business Intelligence*)

O termo *business intelligence* (ou inteligência de negócio) é utilizado de forma global como referência aos processos, técnicas e ferramentas que fornecem suporte às decisões de negócio baseadas em tecnologia de informação. A abordagem pode variar de uma simples planilha até um avançado sistema de inteligência competitiva. Mineração de dados é um importante novo componente no processo de inteligência de negócio (CABENA et al., 1997).

A Figura 3.4 ilustra o posicionamento lógico das diversas metodologias de suporte à inteligência de negócio.

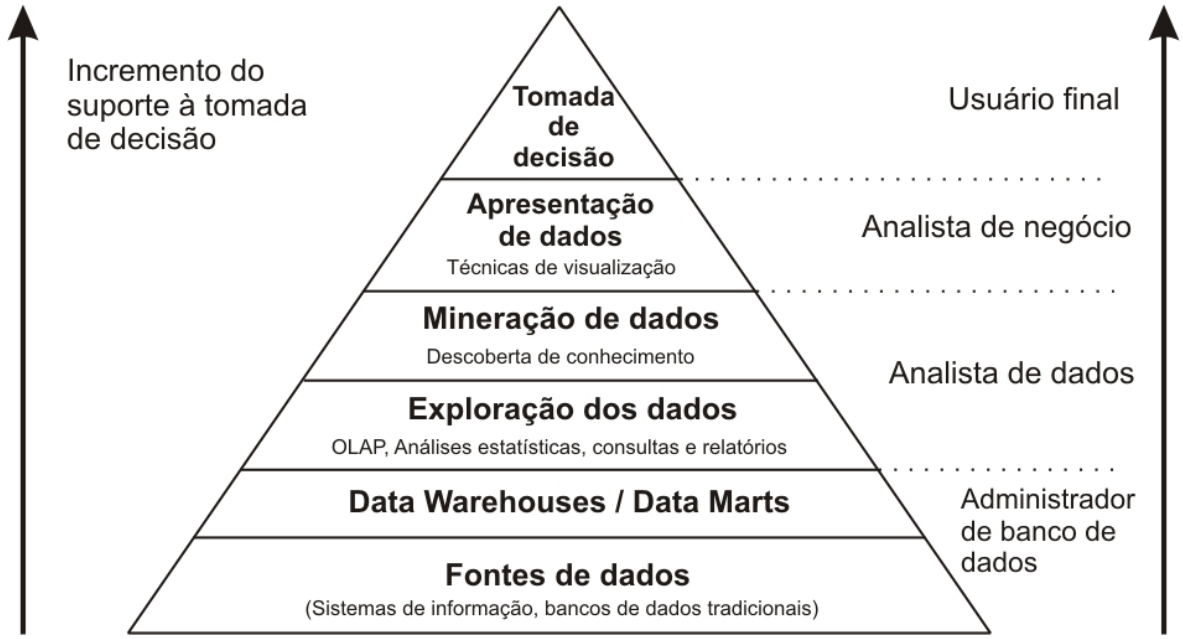

Figura 3.3: Data Mining e Business Intelligence

Fonte: Cabena et al. (1997)

Em geral, o valor da informação para o suporte à decisão cresce da base para o topo da pirâmide. Uma decisão baseada em dados das camadas inferiores, onde tipicamente existem milhões de registros, irá afetar tipicamente apenas uma transação individual. Já uma decisão baseada nos dados sumarizados e tratados das camadas superiores permite uma

tomada de decisão com potencial de subsidiar alguma ação de um departamento ou mesmo um redirecionamento na organização (CABENA et al., 1997).

3.8 CRM

Embora mundialmente utilizado, o termo CRM nunca foi formalmente definido. Assim, muitos fornecedores, aproveitando o movimento do mercado nessa direção, chamam suas aplicações de CRM (as mesmas que já existem há algum tempo). Pode-se dizer que CRM é a infra-estrutura para implementar-se a filosofia *one-to-one* (um a um) de relacionamento com os clientes. CRM também pode ser definido como uma estratégia de negócio voltada ao entendimento e antecipação das necessidades dos clientes atuais e potenciais de uma empresa (CISTER, 2005).

Do ponto de vista tecnológico, CRM envolve capturar os dados do cliente ao longo de toda a organização, consolidar todos os dados capturados interna e externamente em um banco de dados central, analisar os dados consolidados, distribuir os resultados dessa análise aos vários pontos de contato com o cliente e usar essa informação ao interagir com o cliente através de qualquer ponto de contato com a empresa.

Os sistemas tradicionais geralmente são concebidos ao redor de processos ou de produtos onde o cliente é meramente um mal necessário. Nos sistemas e processos que são concebidos à luz do CRM, o cliente é o centro, e todos os relatórios e consultas têm o cliente como porta de entrada.

Existe uma tendência clara da migração do foco em produtos para o foco em clientes. Cada vez mais as empresas se organizam em função dos vários tipos de clientes que possuem. Essa organização permite a diferenciação dos clientes, primeiramente pelas necessidades dos vários tipos de cliente e depois, por suas necessidades individuais. As estratégias corporativas alteram-se de uma visão de marketing focado em produto para um contexto mais personalizado, centrado em estratégias e processos com foco no consumidor. Neste contexto, a utilização de um modelo baseado em CRM tornou-se importante para o atingimento de um diferencial competitivo (KANTARDZIC et al., 2005).

Em um ambiente de CRM podem distinguir-se projetos de aquisição e fidelização de clientes. Aquisição exige o estabelecimento de um canal de comunicação com entidades sendo prospectadas e pertencentes a determinados grupos pré-definidos, com a intenção de convertê-las, gradualmente, em clientes ou consumidores. Projetos buscando a fidelização ou a retenção de clientes centralizam seus processos visando a manutenção do bom atendimento e da satisfação dos consumidores (KANTARDZIC et al., 2005).

A melhor forma de se testemunhar essa mudança é visitando os *sites* das empresas na *web*. Verifica-se que os *sites* estão mudando seus menus da orientação a produtos para a categorização por tipo de cliente e/ou necessidade. Isso facilita, de certa forma, o acesso às informações e ofertas pertinentes às necessidades dos clientes, além de facilitar o aprendizado da empresa a respeito das necessidades de seus clientes (CISTER, 2005).

Do ponto de vista dos sistemas, CRM é a integração dos módulos de automação de vendas, gerência de vendas, telemarketing e televendas, serviço de atendimento e suporte ao cliente (SAC), automação de marketing, ferramentas para informações gerenciais, *web* e comércio eletrônico.

Outro aspecto importante em CRM é a integração de todas essas aplicações com os sistemas de ERP (*Enterprise Resource Planning* – ou Planejamento de Recursos Empresariais) ou com os sistemas transacionais. O crescimento e a qualidade da receita são os objetivos das organizações e, por isso, a metodologia de CRM deve ser incorporada à visão da organização.

Conclui-se, então, que CRM é relacionado com a captura, processamento, análise e distribuição de dados, com a total preocupação com o cliente, permitindo que a empresa venda mais seus produtos ou ofereça produtos mais eficientes. Talvez o dado mais revelador seja a expectativa de alta integração entre as áreas de negócios (*marketing*, vendas, etc.) com a área de tecnologia de informação (TI). Essas áreas, no passado, eram quase que opostas, com divergência de propósitos. Hoje, nas empresas mais competitivas, cada vez mais se verifica a preocupação da área de TI em ser uma ferramenta para a realização dos objetivos de negócio da empresa.

Cister (2005) define padrões diferenciados para um sistema de CRM:

- **CRM Analítico:** sua função é determinar quais são os clientes que devem ser tratados de forma personalizada e quais os que devem ser deslocados para níveis de prioridade inferior;

- **CRM Operacional:** foco da maioria das empresas. Sistemas, como automatização da força de vendas, centros de atendimento a clientes (call centers), sites de comércio eletrônico e sistemas automatizados de pedido. O objetivo é racionalizar e otimizar processos da organização. Quando bem implementadas essas iniciativas podem trazer agilidade ao atendimento, o que pode, em última análise, traduzir-se em benefício para o cliente. Para melhor ilustração, pode-se usar o exemplo dos call – centers. Tais serviços são orientados ao tempo de cada ligação, quantidade de ligações não atendidas, chamadas por agente, chamadas específicas, entre outros. Porém, falta à grande maioria das empresas adicionar a tal serviço o conceito da mais-valia, direcionando a clientes mais preciosos tratamento diverso do oferecido a outrem menos valioso, claro, sem deixar de oferecer o serviço adequado a este;

- **CRM Colaborativo:** é a aplicação da tecnologia de informação que permite a automação e a integração entre todos os pontos de contato do cliente com a empresa. Esses pontos de contato devem estar preparados para interagir com o cliente e disseminar as informações levantadas para os sistemas do CRM Operacional.

3.9 Desafios no Processo de Mineração de Dados

Em mineração de dados, diversos são os requisitos considerados importantes para a obtenção de sucesso em sua aplicação. Alguns destes itens, tal como os descritos a seguir, ainda se encontram em estágio de pesquisa, e, portanto, representam desafios para sistemas de MD (HAN et al., 2001):

a) Metodologia de mineração e questões de interação do usuário

- mineração de diferentes tipos de conhecimento em banco de dados: deve ser coberto um largo espectro de tarefas de descoberta de conhecimento e análise de dados, incluindo caracterização, discriminação, associação, classificação, agrupamento, análise de tendências e desvios, e análise de similaridade de dados;

- mineração interativa de conhecimento a níveis múltiplos de abstração e incorporação de conhecimento de fundo: permite ao usuário focar a pesquisa por padrões, provendo e refinando as requisições de MD baseadas nos resultados retornados. A informação referente ao domínio estudado deve ser usada para guiar o processo de descoberta e permitir que os padrões descobertos sejam expressos em termos concisos e em níveis diferentes de abstração;

- linguagens de consulta de mineração de dados e mineração de dados *ad hoc*: as linguagens de consulta relacionais, como SQL, permitem a construção de consultas específicas para um determinado fim (*ad hoc*). De forma similar, devem ser desenvolvidas linguagens de consultas para MD e o ideal é que venham a ser integradas em linguagens de consulta de banco de dados ou de *Data Warehouse* e otimizadas para a mineração de dados eficiente e flexível;

- apresentação e visualização dos resultados da MD: o sistema deve apresentar requisitos de expressão do conhecimento descoberto em linguagens de alto nível, representações visuais ou outras formas expressivas que tornem esse conhecimento de fácil entendimento e usável pelos humanos, como por exemplo, árvores, tabelas, regras, gráficos, etc;

- suporte a ruídos ou a dados incompletos: o sistema deve prover métodos de limpeza e análise para o tratamento de ruídos e dados incompletos, os quais comprometem a acurácia dos padrões descobertos. Devem também prover métodos de mineração de *outliers*, para a descoberta e análise de exceções;

- avaliação de padrões: diz respeito ao uso de medidas de interesse para guiar o processo de descoberta e reduzir o espaço de busca.

b) Itens relacionados ao desempenho

- eficiência e escalabilidade dos algoritmos de MD: para a extração de informação de bancos de dados volumosos, os algoritmos de MD devem ser eficientes e escaláveis, isto é, o tempo de execução deve ser predizível e aceitável em grandes bancos de dados. Sob a perspectiva de banco de dados, estes são considerados itens chaves em implementações de sistemas de MD;

- algoritmos de mineração paralelos, distribuídos e incrementais: referentes à divisão de dados em partições, as quais são processadas em paralelo e depois mescladas. Além disso, alguns processos de MD de alto custo geram a necessidade de algoritmos incrementais que incorporem atualizações no banco de dados sem que seja necessária a mineração de todo o banco novamente;

c) Itens relacionados à diversidade de tipos de banco de dados:

- suporte de tipos de dados relacionais e complexos: os sistemas de MD devem suportar não apenas os tipos de dados presentes em bancos de dados relacionais e *Data Warehouses*, mas também dados complexos, tais como hipertexto, multimídia, dados espaciais, temporais ou transacionais;

- mineração de informação de banco de dados heterogêneos e sistemas de informações globais: referente à conexão de múltiplas fontes de dados por meio de redes de computadores, tal como a Internet. A descoberta de conhecimento em fontes estruturadas, semi-estruturadas e não-estruturadas representa grande desafio para MD. Um exemplo disto é o processo de *web mining*, em que um simples sistema de consulta de dados é inviável para a extração de informação importante sobre as ações que ocorrem na *web*;

d) proteção da privacidade e segurança da informação em MD:

- itens importantes devido ao uso popular crescente de ferramentas de MD e redes de telecomunicações e de computadores, gerando a necessidade de métodos que assegurem a proteção da privacidade e segurança da informação enquanto facilitador do acesso e mineração da informação.

Considerando particularmente a formação do modelo proposto neste trabalho, os principais desafios e condições de contorno observadas dizem respeito principalmente à:

- diversidade das bases de dados: utilização de fontes diversas, desde de relatórios formatos a partir do *Data Warehouse*, ou até a necessidade de construção de ferramentas específicas para a formação de conjuntos de dados;

- dificuldades na obtenção do conjunto necessário de variáveis para a elaboração dos *data marts* exigidos pelos algoritmos de classificação;

- paradigma da performance *versus* precisão: em função da natureza das bases de dados envolvidas neste trabalho (grandes conjuntos de informações, contendo cadastros de clientes, dados de consumo de telefonia e faturamento), torna-se necessária a utilização de técnicas de redução de dados e seleção de atributos, a fim de garantir um tempo de execução predizível e aceitável (sem prejudicar a eficácia e validade do modelo).

3.10 Resumo

Neste capítulo foram apresentados os principais conceitos aplicados em um modelo de mineração de dados. Foram descritas as principais definições a respeito do termo "mineração de dados", procurando a diversificação e abrangência através da citação de referências diversas. O capítulo descreveu a fundamentação básica da mineração de dados, citando as raízes teóricas do método, passando pela Estatística, pela Inteligência Artificial e pelo Aprendizado de Máquina.

A descrição do Ciclo de Descoberta de Conhecimento em Bases de Dados é um dos pilares da aplicação de técnicas de mineração de dados no Estudo de Caso desta dissertação. Este capítulo contém a fundamentação teórica para a formatação da base de dados submetida aos algoritmos de classificação e predição.

Em cada etapa que compõe o ciclo de Descoberta de Conhecimento em Bases de Dados, descrevem-se as principais técnicas para a correta avaliação e preparação dos conjuntos de informações que serão, posteriormente, utilizados na etapa de mineração de padrões. O entendimento de técnicas como a identificação de inconsistências e poluição, a

verificação de integridade, o tratamento de classes desbalanceadas e a redução de dados (apresentadas neste capítulo) possibilitou o correto tratamento das bases originadas de diversos sistemas, garantindo a sua aplicabilidade e eficácia na formação do conjunto de informações utilizado no modelo final. Foram apresentadas também as definições sobre sistemas de CRM, além da conceituação das principais ferramentas para manutenção e armazenamento de dados no processo de mineração, como *Data Warehouse* e *data marts.*

4 PRINCIPAIS TÉCNICAS DE DESCOBERTA DE PADRÕES

Funcionalidades da mineração de dados são utilizadas para especificar o tipo de padrões a serem encontrados em um modelo de descoberta de conhecimento. De forma geral, tarefas de mineração de dados podem ser classificadas em duas categorias principais: descritiva e preditiva (HAN et al., 2001).

Tarefas descritivas de mineração de dados caracterizam propriedades gerais dos dados em um modelo. Já as tarefas preditivas efetuam inferência nos dados em análise, visando o delineamento de predição.

Em determinados casos, o pesquisador não conhece os tipos de padrões que podem caracterizar alguma importância em seu conjunto de dados, e pode ser interesse do modelo a pesquisa por diferentes tipos de padrões em paralelo. Assim, é importante a definição de um modelo de mineração capaz de buscar diferentes tipos de padrões para atender as necessidades específicas do modelo.

Segundo Han et al. (2001), os modelos de mineração de dados são capazes de descobrir padrões em diferentes granularidades (ou diferentes níveis de abstração). Além disso, devem permitir ao pesquisador a especificação de instruções ou regras para focar a busca de padrões. Inerente ao processo de descoberta, existe uma medida de precisão ou confiabilidade do modelo, que usualmente é associada em cada padrão descoberto e fornecido pelo modelo.

Diversas funcionalidades são usadas para especificar os tipos de padrões que podem ser encontrados nas tarefas de MD. Essas funcionalidades se referem a técnicas que podem ser usadas individualmente ou em conjunto para a descoberta de padrões.

A seguir, são descritas as funcionalidades de MD e os tipos de padrões que elas podem descobrir, seguindo principalmente a estrutura proposta em Han et al. (2001).

4.1 Descrição de Classe / Conceito - caracterização e discriminação

Em um BD, os dados podem estar associados a classes ou conceitos. Por exemplo, em uma loja de eletrônicos, computadores e impressoras são classes de itens para venda, e grandes e pequenos consumidores são conceitos de clientes. Tais descrições de classes/conceitos são úteis para a sumarização, concisão e precisão de termos e podem ser obtidas por intermédio de:

(a) caracterização de dados, que é dada pela sumarização das características gerais ou atributos de uma classe alvo de dados, por exemplo, o estudo das características de produtos de software cujas vendas cresceram em 10% no ano passado.

A sumarização cria descrições compactas das características de um subconjunto de dados, permitindo a visualização de sua estrutura de dados. Alguns métodos envolvem a derivação de regras gerais, técnicas de visualização para variáveis múltiplas e a descoberta de relações funcionais entre variáveis (FAYYAD et al., 1996).

(b) discriminação de dados, pela comparação dos atributos gerais dos objetos da classe alvo com os atributos gerais de objetos de uma ou de um conjunto de classes comparativas (classes contrastantes), por exemplo, a comparação de atributos gerais de produtos de software cujas vendas cresceram em 10% no ano passado com aqueles cujas vendas decresceram no mínimo 30% durante esse ano.

As descrições discriminativas utilizam medidas comparativas para distinguir as classes alvo e as classes contrastantes, tais como métodos de análise de relevância dimensional e generalização síncrona para a construção de classes no mesmo nível conceitual incluindo apenas as dimensões mais relevantes. Podem ser expressas em forma de regras discriminativas, como por exemplo, 60% dos produtos de software mais vendidos no ano passado custavam menos de R$ 500,00, enquanto que 80% dos produtos menos vendidos nesse ano custavam mais de R$1.000,00.

(c) ou ambas as opções (caracterização e discriminação).

4.2 Análise Associativa

É a descoberta de regras associativas que mostram condições de atributo-valor que ocorrem freqüentemente juntas em um determinado conjunto de dados. São muito utilizadas em cestas de compras ou análise de transações de dados. Por exemplo, no banco de dados relacional da loja de eletrônicos, um sistema de MD encontra regras do tipo:

$$idade(X, \text{“20 ... 29”}) \wedge renda(X, \text{“R\$1000 ... R\$2900”}) \Rightarrow compra(X, \text{“CD player”})$$

$$[suporte = 2\%, confiança = 60\%]$$

onde *X* é uma variável que representa o cliente. A regra indica que, dos clientes estudados, 2% (suporte) têm de 20 a 29 anos de idade e renda de R$ 1.000 a R$ 2.900 reais e compraram CD *player*. Há 60% de probabilidade (confiança) de um cliente desse grupo de idade e renda vir a comprar um CD *player*. Este exemplo se refere a uma regra associativa multidimensional, em que ocorre a associação de mais de um atributo ou predicado, e cada atributo representa uma dimensão, segundo a terminologia usada em bancos de dados multidimensionais. Um outro exemplo é a determinação de quais itens são comprados juntos freqüentemente na mesma transação. Então, uma regra gerada pode ser:

$$contém(T, \text{"computador"}) \Rightarrow contém(T, \text{"software"})$$

$$[suporte = 1\%, confiança = 50\%]$$

em que se uma transação *T* contém computador, há 50% de chance de conter também software e 1% de todas as transações contêm ambos. Aqui, um único atributo ou predicado se repete (contém) e esta regra é dita regra associativa unidimensional. Pode ser escrita na forma "$computador \Rightarrow software\ [1\%, 50\%]$".

4.3 Análise de Agrupamento ou de *Cluster*

Ao contrário da classificação e predição (métodos descritos a seguir), que analisam objetos com classes rotuladas, agrupamento ou *clustering* analisa objetos cujos rótulos de classe são desconhecidos. Em dados de treinamento em que os rótulos de classe não estão presentes, esta técnica pode ser utilizada para gerar esses rótulos.

Os objetos são agrupados sob o princípio da maximização da similaridade intraclasse e minimização da similaridade interclasse. Significa que os agrupamentos de objetos são formados de maneira que objetos dentro de um agrupamento possuem alta similaridade entre si, mas os objetos de agrupamentos diferentes apresentam alta dissimilaridade. Cada agrupamento formado pode ser visto como uma classe de objetos da qual regras podem ser extraídas. Também é usado para facilitar a formação de taxonomia, que diz respeito à organização de observações dentro de uma hierarquia de classes que agrupam eventos similares.

4.4 Análise de *Outlier*

Outliers são objetos de um banco de dados que não acompanham o comportamento ou modelo dos dados. Muitos métodos de MD descartam os *outliers* como ruído ou exceções. Porém, em aplicações, como por exemplo, detecção de fraudes, os eventos raros podem ser bastante interessantes. Podem ser detectados com testes estatísticos que assumem um modelo de distribuição ou probabilístico para os dados, ou utilizam medidas de distância, em que aqueles objetos mais distantes de qualquer um dos agrupamentos são considerados *outliers*. Também existem métodos baseados em desvios, que os identificam examinando as diferenças das principais características de objetos em um grupo.

4.5 Análise de Evolução de Dados

Descreve e modela regularidades ou tendências para objetos cujo comportamento se modifica com o passar do tempo. Apesar de incluir caracterização, discriminação, associação, classificação ou agrupamento de dados relacionados com o tempo, esta análise se caracteriza por incluir a análise de dados de séries temporais, casamento de padrões de seqüência ou periodicidade e análise baseada em similaridade.

4.6 Classificação e Predição

A classificação é o processo de encontrar um conjunto de modelos que descrevem e distinguem classes de dados ou conceitos. Esses modelos são usados para predição de objetos cujas classes são desconhecidas, baseada na análise de um conjunto de dados de treinamento (objetos cujas classes são conhecidas).

Segundo Soares (2005), a tarefa de classificação tem como objetivo encontrar algum tipo de relacionamento entre os atributos preditivos e o atributo objetivo, de modo a obter um conhecimento que possa ser utilizado para prever a classe de um determinado registro que ainda não possui classe definida. O modelo gerado pode ser representado sob a forma de regras de classificação (*se-então*), árvores de decisão, fórmulas matemáticas ou redes neurais.

As tarefas de classificação e predição podem requerer análise de relevância para identificar atributos que não contribuem para esses processos, os quais poderão, portanto, ser excluídos.

Conforme Ferreira (2005), um problema de classificação de padrões se resume a determinar, da forma mais correta possível, a classe de saída à qual o padrão de entrada (registro) pertence. O desenvolvimento de um modelo de classificação caracteriza-se pela definição do grupo de classes, das varáveis relevantes e de um conjunto de treinamento consistindo em exemplos (padrões) pré-classificados.

Grande parte dos problemas de classificação de padrões de interesse real tem duas fases: uma etapa *in sample* ou de treinamento (aprendizado), executada a partir do banco de dados existente, e uma etapa *out of sample* ou de generalização, onde são apresentados dados que não foram utilizados no treinamento. O que se deseja, fundamentalmente, é extrair do banco de dados as características que definem o mapeamento entre entrada e saída, para posteriormente utilizá-las na tomada de decisões em dados novos, ainda não avaliados pelo sistema (FERREIRA, 2005).

Han et al. (2001) descreve o primeiro passo em classificação como a construção de um conjunto pré-determinado de classes e conceitos. O modelo é construído através da análise de tuplas de um banco de dados descritas como atributos. Cada tupla é assumida como

pertencente a determinada classe, como determinado por um dos atributos que formam a base – este atributo é denominado atributo de rótulo de classe (ou *class label attribute*). Neste contexto, as tuplas ou registros também são denominadas *amostras*, *exemplos* ou *objetos*.

Estas amostras selecionadas para o modelo formam o *data mart* (ou *data set*) de treinamento. A Figura 4.1 (a) ilustra este processo. Os registros ou amostras individuais que formam o conjunto de dados de treinamento são denominados *amostras de treino* e são selecionadas aleatoriamente da população total (recomenda-se a utilização de técnicas de redução de dados e amostragem para a seleção destes objetos).

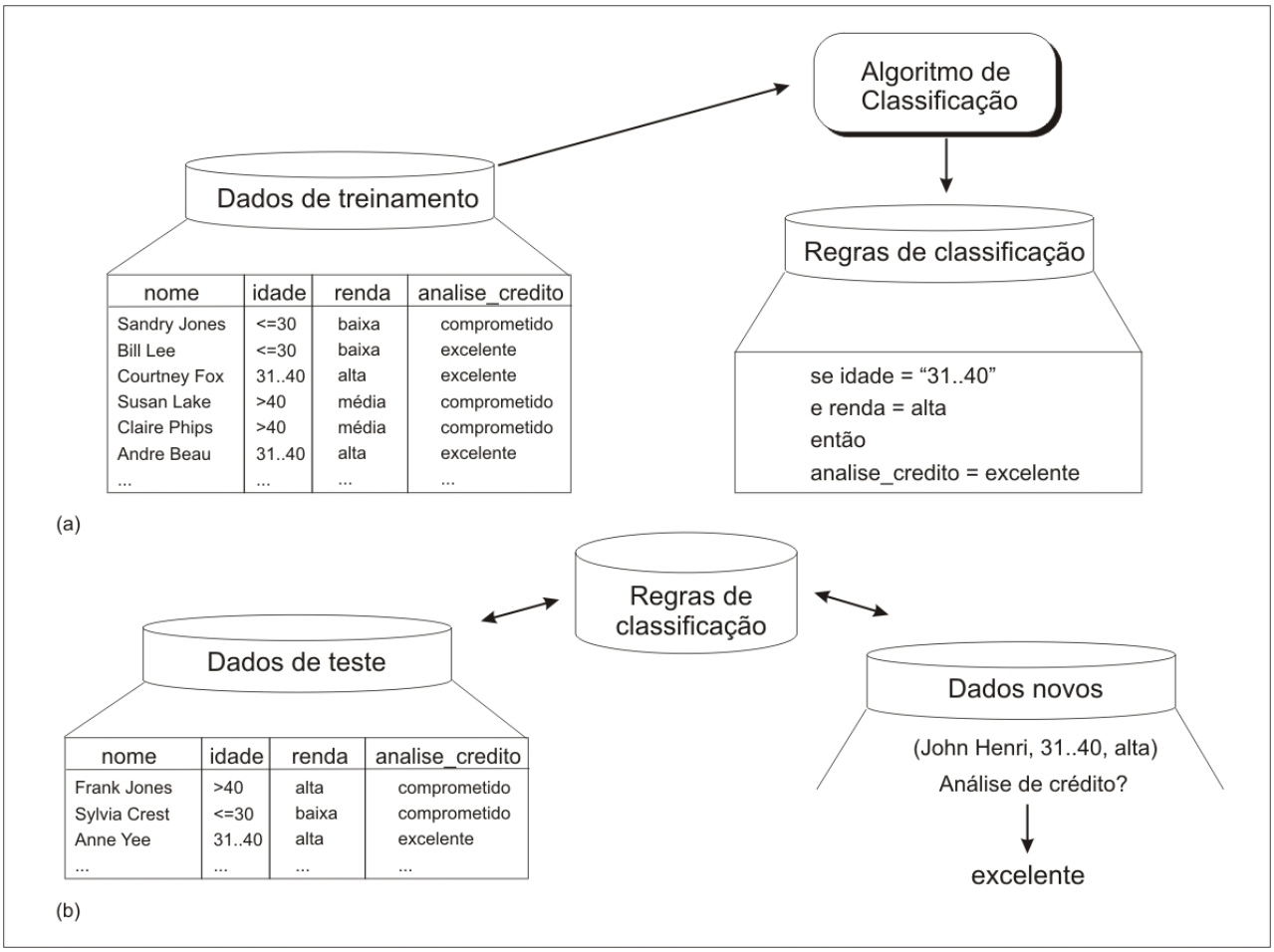

Figura 4.1: processo de classificação de dados
Fonte: Han et al. (2001)

Como o rótulo (ou descrição) de cada atributo é conhecido e fornecido ao modelo, esta etapa de classificação também é denominada **aprendizado supervisionado** (a

aprendizagem do modelo é dita "supervisionada" porque se conhece a classe a que pertence cada um dos objetos do conjunto de treinamento). Tomando como exemplo o conjunto de dados utilizado na formação do modelo proposto neste trabalho, para cada registro (ou objeto) do conjunto de treinamento, formado por diversos atributos referentes a cada cliente (como dados cadastrais, dados de consumo e de faturamento), são conhecidos os registros ou amostras pertencentes à classe alvo da predição (ou seja, a classe *churner* – clientes que efetuaram o cancelamento do serviço junto à operadora).

O aprendizado supervisionado contrasta com o **aprendizado não supervisionado** (ou *clustering*), onde o rótulo dos atributos em cada amostra de treino não é conhecido, e o número ou conjunto de classes a serem aprendidas pode não ser conhecidas previamente.

Este trabalho utiliza a metodologia de classificação e predição para a identificação da classe objetiva, ou seja, clientes potenciais a cancelarem o serviço (também denominados *churners*). Assim, o modelo definido baseia-se na utilização de uma metodologia de classificação / predição, com aprendizado supervisionado.

Em um segundo passo, o modelo é utilizado para classificação (Figura 4.1 – b). Neste momento, estima-se a precisão do modelo (ou classificador). Os principais critérios para avaliar a precisão do modelo são (HAN et al., 2001):

- precisão da predição: refere-se a habilidade do modelo em corretamente predizer a classificação de objetos novos ou ainda não conhecidos;
- velocidade: refere-se aos custos computacionais em geral envolvidos na utilização do modelo;
- robustez: habilidade do modelo em efetuar predições corretas mesmo em um modelo contendo dados com ruído ou valores ausentes;
- escalabilidade: habilidade para construir um modelo eficiente em função da grande quantidade de dados;
- interpretabilidade: refere-se ao nível de informação útil e conhecimento descoberto e provido pelo modelo.

A precisão de um modelo em um determinado conjunto de dados de treinamento é a porcentagem de amostras de teste que são corretamente classificadas pelo modelo. Para cada amostra do conjunto de teste, o valor conhecido da classe (real) é comparado com o valor previsto pelo algoritmo.

Se a precisão do modelo é considerada aceitável, o modelo pode ser utilizado para classificar objetos futuros (onde não se conhece o valor da classe). Considerando este trabalho, esta etapa seria a utilização do modelo para a predição de clientes *churners* no ambiente organizacional (ou a aplicação comercial da técnica), buscando a predição da classe para um conjunto de dados não utilizado na fase de aprendizado.

4.6.1 Definição de Classe

Uma importante questão é o entendimento do conceito de classes nos estudos de classificações, considerando a natureza e o caminho que elas são definidas. Existem três casos bem definidos de classes, descritos em (MICHIE et al., 1994):

- classes como rótulos de diferentes populações: neste tipo de classe não leva em consideração associações de várias populações. Por exemplo, cachorro e gato formam duas diferentes classes ou populações, e como é conhecido, com certeza, ou o animal é um cachorro ou um gato (ou nenhum). Associação de uma classe ou população é determinada por uma autoridade independente, a distribuição para uma classe é independentemente determinada de quaisquer atributos ou variáveis;

- classes resultantes de um problema de predição: esta classe é uma conseqüência do resultado de uma predição através do conhecimento dos atributos. Em termos estatísticos, a classe é uma variável randômica. Um exemplo típico é em aplicações de predições de taxas de juros. Freqüentemente a questão é: as taxas de juros subirão (classe = 1) ou não (classe = 0)?;

- classes determinadas pelos valores espaciais dos seus atributos: pode-se dizer que a classe é uma função dos atributos. Então um item

manufaturado pode ser classificado como falho se alguns atributos estão fora dos limites pré-determinados, e não falhos de outra forma. Existem regras que classificam os valores dos atributos. O problema é criar uma regra que imite a realidade o mais próximo possível. Muitos sistemas de crédito são desse tipo de classe.

No modelo proposto neste trabalho, a classe é definida claramente como resultante de um problema de predição, podendo ser descrita como um atributo pertencente à classe *Churner* (C) ou *Não-Churner* (NC), em função do resultado da aplicação de um algoritmo de classificação.

4.7 Resumo

Este capítulo avaliou os principais modelos de descoberta de padrões utilizados em mineração de dados. São descritas as metodologias de Descrição de Classe / Conceito; a Análise Associativa; a Análise de *Outlier*; a Análise de Evolução dos Dados e a Classificação / Predição.

Em função do escopo deste trabalho, que apresenta a modelagem de um sistema de mineração de dados direcionada à identificação de clientes *churners* em potencial, as técnicas de Predição e Classificação receberam atenção especial. Apresentou-se a estrutura do processo, envolvendo a execução de duas etapas principais: treinamento (ou aprendizado) e generalização (ou teste) do modelo, com base em um conjunto de dados inicial oferecido.

Foram descritos também os cenários que caracterizam determinado modelo como Aprendizado Supervisionado ou Aprendizado Não-Supervisionado. Verificou-se que o sistema desenvolvido nesta dissertação utiliza a metodologia de classificação e predição para identificação de classe objetiva (clientes dispostos a cancelarem o serviço), com Aprendizado Supervisionado.

Apresenta-se também as principais definições para a caracterização de Classe em mineração de dados, ilustrando os casos principais e conceituando a classe alvo do Estudo de Caso apresentado nesta dissertação.

5 FERRAMENTAS DE CLASSIFICAÇÃO E PREDIÇÃO

Em um projeto de mineração de dados, com foco definido em Classificação e Predição, a utilização de diferentes abordagens e algoritmos é fundamental para um processo de avaliação. Como descrito por Han et al. (2001), são avaliados não apenas a precisão do modelo, mas também a velocidade, a robustez, a escalabilidade e a interpretabilidade.

Dependendo do modelo e da estrutura de predição em um cenário de aprendizado supervisionado, alguns algoritmos e ferramentas podem ser mais adequados ao problema em questão. Este capítulo apresenta um estudo exploratório sobre as principais técnicas e algoritmos de Classificação / Predição, em função do núcleo do modelo proposto. Serão avaliados três métodos, citados principalmente por Han et al. (2001), Fayyad et al. (1996) e Klösgen et al. (2002):

- Redes Neurais – particularmente o modelo *RBF Network – Radial Basis Function Network*;
- *Decision Trees* (Árvores de Decisão) – particularmente o algoritmo C4.5 (através do classificador J48, da ferramenta WEKA);
- Classificadores Bayesianos (principalmente o método *Naive Bayes*).

A escolha destes algoritmos como ferramentas de predição a serem utilizadas neste trabalho foi embasada nos seguintes aspectos:

- disponibilidade de uso – tais algoritmos são encontrados no pacote de classificadores da ferramenta WEKA (*Waikato Environment for Knowledge Analysis*), utilizada no estudo de caso deste trabalho;
- performance: em testes preliminares, os algoritmos explorados neste capítulo foram os que apresentaram melhor desempenho sobre o conjunto de dados do modelo;
- custo: avaliado principalmente sob a ótica dos recursos necessários para a execução dos algoritmos. Por exemplo, a utilização de Redes Neurais

MLP´s (*MultiLayer Perceptron*), foi descartada em função do tempo de processamento e dos recursos de hardware exigidos para o modelo em questão.

5.1 Redes Neurais

Segundo Klösgen et al. (2002), Redes Neurais oferecem uma arquitetura computacional robusta e distribuída, composta de significativas funções de aprendizado e capacidade de representação de relacionamentos não-lineares e multi-variáveis.

Conforme Coutinho (2003), essa tecnologia é a que oferece o mais profundo poder de mineração, mas é, também, a mais difícil de se entender. As redes neurais tentam construir representações internas de modelos ou padrões achados nos dados, mas essas representações não são apresentadas para o usuário. Com elas, o processo de descoberta de padrões é tratado pelos programas de *data mining* dentro de um processo "caixa preta".

Estruturalmente, uma rede neural consiste em um número de elementos interconectados (chamados neurônios) organizados em camadas que aprendem pela modificação da conexão que conectam as camadas (LEMOS, 2003).

As Redes Neurais Artificiais utilizam um conjunto de elementos de processamento (ou nós) análogos aos neurônios no cérebro humano. Estes elementos de processamento são interconectados em uma rede que pode identificar padrões nos, ou seja, a rede aprende através da experiência. Esta característica distingue redes neurais de tradicionais programas computacionais, que simplesmente seguem instruções em uma ordem seqüencial fixa (DIN, 1998).

Embora seja verdadeiro que as redes neurais apresentem o mais avançado poder de mineração, muitos analistas de negócio não podem fazer uso delas porque os resultados finais não podem ser explicados. Estruturalmente, uma rede neural consiste em um número de elementos interconectados (chamados neurônios), organizados em camadas que aprendem pela modificação da conexão firmemente conectando as camadas. Geralmente, constroem superfícies equacionais complexas através de interações repetidas, cada hora ajustando os parâmetros que definem a superfície. Depois de muitas repetições, uma superfície pode ser

internamente definida e se aproxima muito dos pontos dentro do grupo de dados (CISTER, 2005).

As redes neurais artificiais consistem em um método para solucionar problemas da área de inteligência artificial, através da construção de um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São técnicas computacionais que apresentam um modelo inspirado na estrutura neural dos organismos inteligentes, que adquirem conhecimento através da experiência (LEMOS, 2003).

Assim como o sistema nervoso é composto de bilhões de células nervosas, a rede neural artificial também seria formada por unidades que nada mais são do que pequenos módulos (ou unidades de processamento ou nós) que simulam o funcionamento de um neurônio. Estes módulos devem funcionar de acordo com os elementos em que foram inspirados, recebendo e retransmitindo informações (LEMOS, 2003).

O neurônio artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico. Assim sendo, os dendritos foram substituídos por entradas, cujas ligações com o corpo celular artificial são realizadas através de elementos chamados de peso (simulando as sinapses). Os estímulos captados pelas entradas são processados pela função de soma, e o limiar de disparo do neurônio biológico foi substituído pela função de transferência (TAFNER, 1998). A Figura 5.1 ilustra a estrutura de um neurônio artificial.

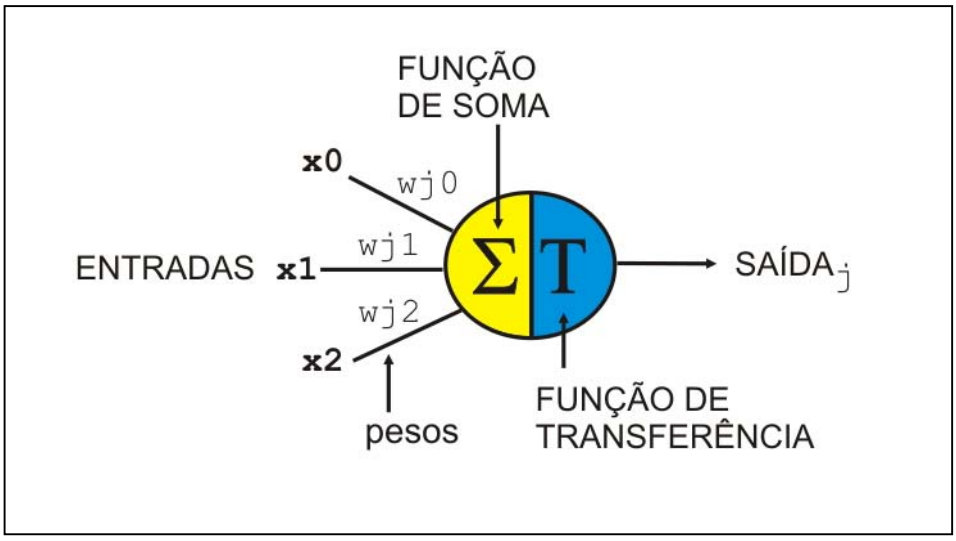

Figura 5.1: estrutura de um neurônio artificial

As funções básicas de cada neurônio são: (a) avaliar valores de entrada, (b) calcular o total para valores de entrada combinados, (c) comparar o total com um valor limiar, (d) determinar o que será a saída.

Segundo Kröse et al. (1996), o vetor *x* que representa um conjunto de "*n*" entradas, é multiplicado por um vetor de pesos, *w*, e o produto, $p = x\,w$, é aplicado aos canais de entrada do neurônio.

A soma de todas as entradas ponderadas é então processada por uma função de ativação, *F(x)*, que vai produzir o sinal de saída *a*, do neurônio:

$$a = F\left(\sum_{i=0}^{n-1} x_i w_i + \theta \right) \quad \textit{Equação 5.1}$$

O parâmetro θ é um valor *threshold* adicionado a soma ponderada, e em alguns casos é omitido, enquanto que em outros é considerado como o valor cujo peso correspondente ao valor de entrada é sempre igual a 1.

Segundo Kröse et al. (1996), o papel de θ, chamado de *bias* ou vício, é aumentar o número de graus de liberdade disponíveis no modelo, permitindo que a rede neural tenha maior capacidade de se ajustar ao conhecimento a ela fornecido.

A função de ativação é importante para o comportamento de uma rede neural porque é ela que define a saída do neurônio artificial e, portanto, o caminho pelo qual a informação é conduzida (KRÖSE et al., 1996).

É através de uma função de ativação que são calculadas as respostas geradas pelas unidades. Existem vários tipos de funções de ativação, sendo que as mais comuns são as descritas a seguir (KRÖSE et al., 1996).

- Função Passo, que produz uma saída binária, e embora seja similar aos neurônios reais, é inadequada para o algoritmo de aprendizagem;
- Função Linear, que elimina a descontinuidade em $x = \theta$;
- Função Sigmoidal, que adiciona alguma não-linearidade.

Enquanto a operação de cada neurônio é razoavelmente simples, procedimentos complexos podem ser criados pela conexão de um conjunto de neurônios. Tipicamente, as entradas dos neurônios são ligadas a uma camada intermediária (ou várias camadas intermediárias) e esta conectada à camada de saída.

Para construir um modelo neural, primeiramente, "adestra-se" a rede em um *dataset* de treinamento e, então, usa-se a rede já treinada para fazer predições. O *dataset* pode ser monitorado durante a fase de treinamento para checar seu progresso.

Cada neurônio, geralmente, tem um conjunto de pesos que determina como avalia-se a combinação dos sinais de entrada. A entrada para um neurônio pode ser positiva ou negativa. O aprendizado se faz pela modificação dos pesos usados pelo neurônio em acordo com a classificação de erros que foi feita pela rede como um todo. As entradas são, geralmente, pesadas e normalizadas para produzir um procedimento suave.

Em uma rede neural artificial as entradas, simulando uma área de captação de estímulos, podem ser conectadas em muitos neurônios, resultando, assim, em uma série de saídas, onde cada neurônio representa uma saída. Essas conexões, em comparação com o sistema biológico, representam o contato dos dendritos com outros neurônios, formando assim as sinapses.

A função da conexão em si é tornar o sinal de saída de um neurônio em um sinal de entrada de outro, ou ainda, orientar o sinal de saída. As diferentes possibilidades de conexões entre as camadas de neurônios podem ter, em geral, *n* números de estruturas diferentes (TAFNER, 1998).

Usualmente, trabalha-se com três camadas, que são classificadas em:

- Camada de Entrada: onde os padrões são apresentados à rede;
- Camadas Intermediárias ou Ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- Camada de Saída: onde o resultado final é concluído e apresentado.

A Figura 5.2 ilustra a formação de uma rede neural artificial.

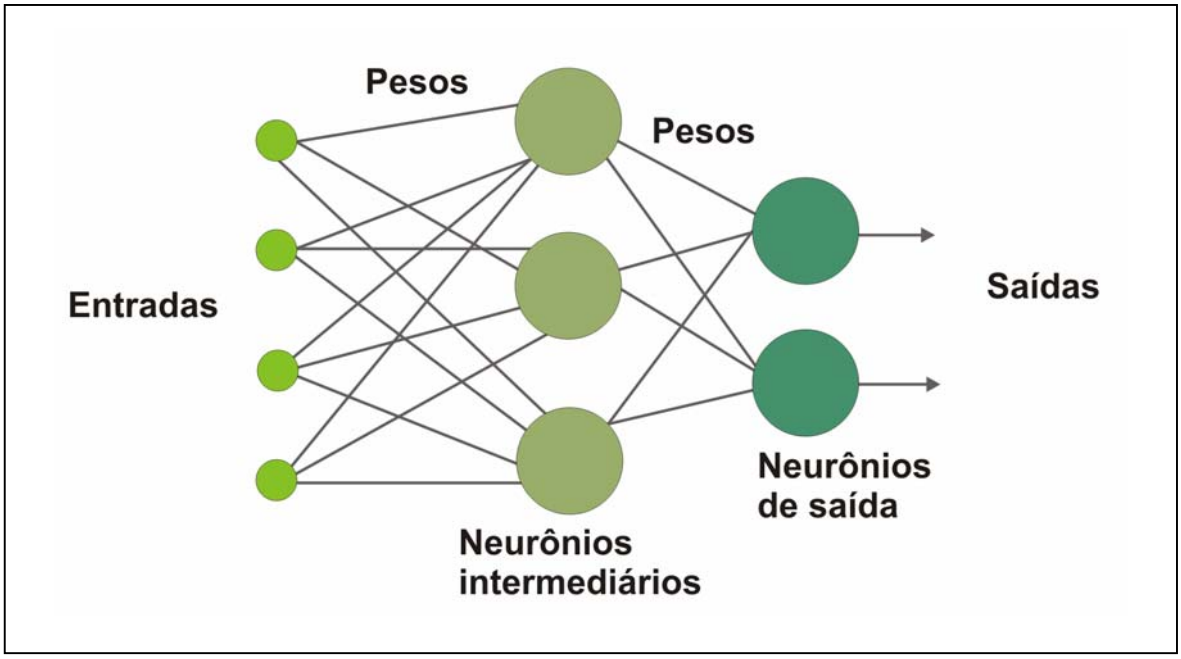

Figura 5.2: formação de uma rede neural artificial

Existe uma significativa quantidade de variantes de uma rede neural, e podem ser modeladas conforme a aplicação. O que faz as redes neurais diferirem entre si são os tipos de conexões e formas de treinamento. Basicamente, os itens que compõem uma rede neural e, portanto, sujeitos a modificações, são os seguintes:

• forma de conexões entre camadas;

• número de camadas intermediárias;

• quantidade de neurônios em cada camada;

• função de transferência;

• algoritmo de aprendizado.

Existem diferentes tipos de redes neurais artificiais e diferentes maneiras de classificá-las. Talvez a mais importante seja quanto à forma de aprendizado ou treinamento, que pode ser supervisionado ou não-supervisionado.

No aprendizado supervisionado são apresentados à rede padrões de entrada e suas saídas (respostas desejadas). Durante este processo, a rede realiza um ajustamento dos pesos das conexões entre os elementos de processamento, segundo uma determinada regra de aprendizagem, até que o erro calculado em função das saídas geradas pela rede alcancem um valor mínimo desejado (MEDEIROS, 1999).

O aprendizado supervisionado (que é o foco deste trabalho) em uma rede neural artificial é ilustrado na Figura 5.3:

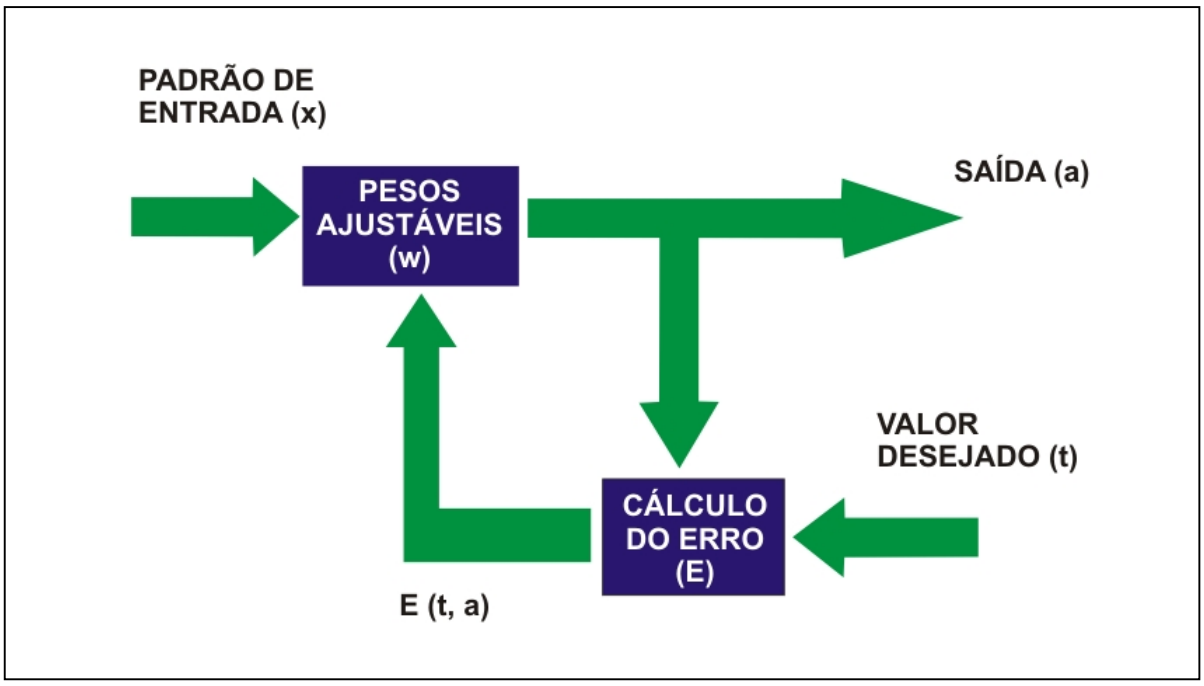

Figura 5.3: aprendizado supervisionado em redes neurais artificiais

No aprendizado não-supervisionado, a rede analisa os conjuntos de dados apresentados e determina algumas propriedades dos conjuntos de dados, "aprendendo" a refletir estas propriedades na sua saída. A rede utiliza padrões, regularidades e correlações para agrupar os conjuntos de dados em classes.

As propriedades que a rede "aprende" sobre os dados pode variar em função do tipo de arquitetura utilizada e da forma de aprendizagem. Por exemplo, Mapa Auto-Organizável de Kohonen, Redes de Hopfield e Memória associativa Bidirecional, são alguns métodos de aprendizado não-supervisionado (MEDEIROS, 1999).

5.1.1 Redes Neurais: Radial Basis Function (RBF)

As redes RBF são modelos de Redes Neurais Artificiais inspirados pelas respostas "localmente sintonizáveis" de alguns neurônios biológicos. Estas células, encontradas em muitas partes dos sistemas nervosos biológicos, respondem a características selecionadas de algumas regiões finitas do espaço dos sinais de entrada (HASSOUN, 1995).

A rede neural RBF apresenta uma estrutura em três camadas em modelo *feedforward* (ou seja, sem conexões recorrentes entre os neurônios). Esta arquitetura é a mais indicada para tarefas de classificação e predição, porque representa melhor os mapeamentos estáticos de entrada e saída; modelos que utilizam arquiteturas recorrentes (ou seja, redes neurais que possuem conexões de retorno ou *feedback*) são mais complexas matematicamente e mais apropriadas para problemas dinâmicos, como análises de séries temporais ou aplicações em controle (KLÖSGEN et al., 2002).

A arquitetura utiliza uma função de transferência linear para as unidades de saída e uma função não-linear de transferência (normalmente Gaussiana) para as unidades ocultas. A camada de entrada consiste de *n* unidades conectadas por conexões com pesos para a camada oculta. Em geral, uma rede RBF pode ser descrita como uma construção global de aproximações para funções utilizando combinações de funções de base centralizadas em vetores de pesos (JAGOTA, 1998).

Segundo Fernandes et al. (1999), as redes de funções radiais de base (redes RBF) têm significativa posição dentro do domínio das redes neurais artificiais. A principal razão para esse resultado é a simplicidade do processo de treinamento e a eficiência computacional. A estrutura da rede RBF é do tipo múltiplas camadas, o método de treinamento é do tipo *feedforward* e o treinamento pode ser supervisionado (método aplicado neste trabalho), ou híbrido (combinando um método não-supervisionado com um supervisionado).

A estrutura básica de uma rede RBF é apresentada na Figura 5.4. A primeira camada é a conexão do modelo como o meio. A segunda camada (ou camada escondida) realiza uma transformação não-linear do espaço vetorial de entrada para um espaço vetorial interno que geralmente tem uma dimensão maior (FERNANDES et al., 1999).

A última camada, a camada de saída, transforma o espaço vetorial interno em uma saída, através de um processo linear. Os neurônios da camada escondida são funções radiais de base (FERNANDES et al., 1999).

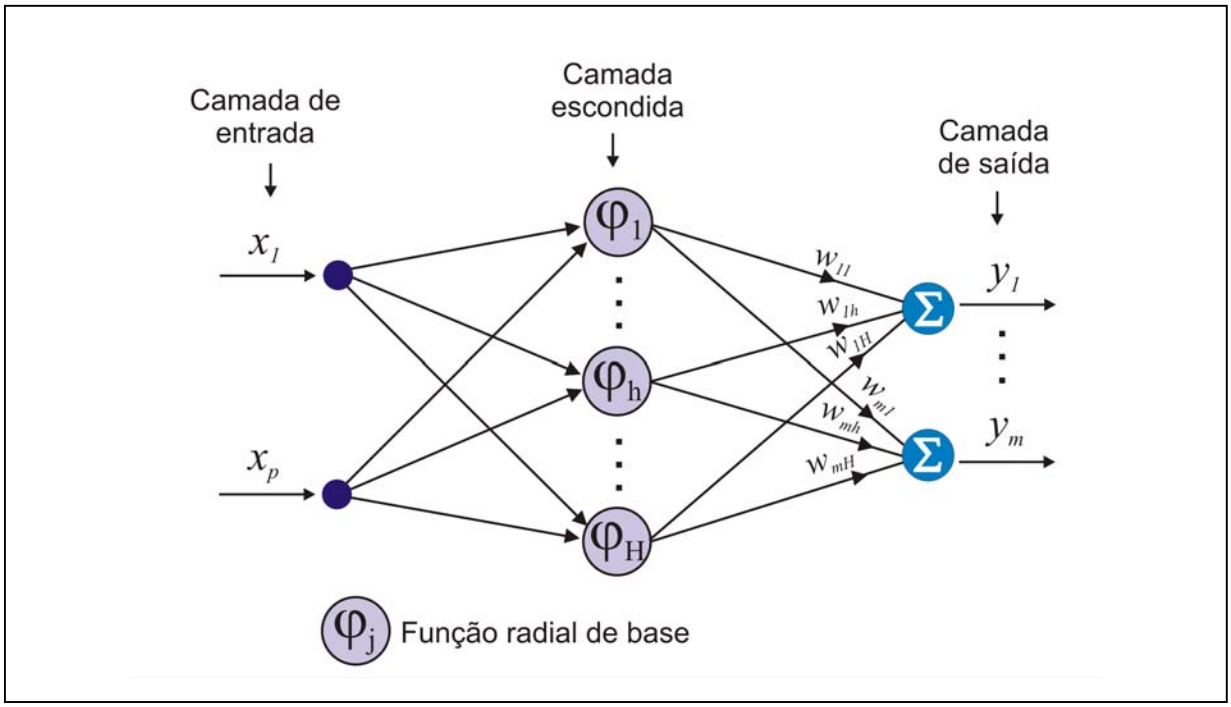

Figura 5.4: arquitetura de uma rede RBF

Fonte: Fernandes et al. (1999).

As funções radiais de base produzem uma resposta significativa, diferente de zero, somente quando o padrão de entrada está dentro de uma região pequena localizada no espaço de entrada. Cada função requer um centro e um parâmetro escalar. A função que é mais utilizada com função de radial de base é a função de Gauss (FERNANDES et al., 1999).

$$\varphi(v) = \exp\left(-\frac{1}{2\sigma^2} \|v - c\|^2 \right) \quad \textit{Equação 5.2}$$

Assim, uma componente y_j do vetor de saída y da rede RBF é caracterizada como:

$$y_j = F_k(x) = \sum_{h=1}^{II} w_{kh}\varphi_h(x_s), \quad \begin{matrix} j=1,2,...,m. \\ s=1,2,...,N. \end{matrix} \quad \textit{Equação 5.3}$$

Substituindo a Equação 5.2 na Equação 5.3:

$$y_j = F_j(x) = \sum_{h=1}^{II} w_{jh} \exp\left(-\frac{1}{2\sigma_h^2} \left\| x_s - ch \right\|^2 \right) \quad \textit{Equação 5.4}$$

Onde w_{jh} é o peso sináptico entre o neurônio h da camada escondida com o neurônio j da camada de saída; x_s é o s – *ésimo* vetor de entrada de um conjunto de treinamento X; e c_h é o vetor de centro relativo ao neurônio h da camada escondida, definido por:

$$c_h = \left[c_{h1}, c_{h2,}, \ldots, c_{hj}, \ldots, c_{hp} \right]^T \quad \textit{Equação 5.5}$$

5.1.2 Comparativo entre Redes RBF e Redes MLP

Normalmente, a utilização de arquiteturas de redes neurais para o tratamento de problemas de classificação sugere duas abordagens: redes RBF (descritas anteriormente) e redes MLP (*Multilayer Perceptron*). Apresenta-se seguir um resumo da estrutura das redes MLP´s e um comparativo entre as duas metodologias, justificando a escolha da rede RBF para a aplicação no estudo de caso deste trabalho.

Uma estrutura MLP é uma rede neural que consiste de uma camada de entrada, uma ou múltiplas camadas escondidas, e uma camada de neurônios de saída. As unidades da camada de entrada não executam nenhuma função; elas apenas repassam os seus valores. Os demais neurônios da rede são simples unidades de processamento combinando (múltiplas) entrada(s) para uma única saída. MLP´s são extensões do modelo simples Perceptron, que consiste de apenas uma camada de entrada e uma de saída, sem camadas escondidas. A introdução de unidades em camada oculta ampliou o modelo Perceptron, que não atendia a certas classes de problemas (KLÖSGEN et al., 2002).

Formalmente, a *Multilayer Perceptron* utiliza uma função f: $R^n \rightarrow R^m$, onde n é o número de unidades de entrada e m é o número de unidades de saída. A saída de uma MLP é calculada pela propagação da entrada de camada em camada da rede (em uma arquitetura *feedforward*). Durante este processo cada neurônio calcula sua nova ativação e saída até que a

saída da rede seja alcançada. Para isso, cada neurônio *j* das camadas ocultas e de saída calcula uma saída o_j e sua ativação atual a_j baseada na entrada da camada anterior através do uso de uma função de entrada net_j ; uma função de ativação A_j(x): $a_j = A_j(net_j)$ e uma função de saída O_j(x): $o_j = O_j(net_j)$. A função de entrada net_j simplesmente efetua o somatório dos produtos da saída o_i de cada unidade da camada precedente e do peso w_{ij} pelo qual é conectado ao neurônio *j*:

$$net_j = \sum_i w_{ij} o_i \quad \textit{Equação 5.6}$$

A função de ativação deve ser diferençável, visto que esta propriedade é requerida para o processo de aprendizado. Freqüentemente, a Função Logística é utilizada:

$$f(x) = \frac{1}{1 + e^{-\beta x'}}, \beta > 0 \quad \textit{Equação 5.7}$$

A Função Logística aproxima uma função *threshold* (limiar) e aproxima o valor *0* para $x \rightarrow -\infty$ e *1* para $x \rightarrow +\infty$. O parâmetro β determina a atenuação da função sigmóide. Uma transformação linear é utilizada como a saída da função, geralmente a função identidade. Neste caso, a saída da rede é restrita ao intervalo [0, 1] (KLÖSGEN et al., 2002).

O método de aprendizado utilizado em redes MLP´s é baseado no algoritmo *backpropagation*. Após a propagação de determinado padrão através da rede, o padrão de saída é comparado com um dado padrão alvo e o erro de cada unidade de saída é calculado. Este erro é propagado para trás – ou seja, para a camada de entrada – através da rede. Com base neste sinal de erro, as unidades escondidas podem determinar seu próprio erro. Finalmente, os erros das unidades são utilizados para modificar seus pesos (KLÖSGEN et al., 2002).

Comparativamente, redes MLP e RBF possuem características semelhantes (ambas são consideradas funções aproximadoras), mas diferem em arquitetura e metodologia de classificação.

As unidades escondidas em redes MLP´s dependem de somas ponderadas das entradas, transformadas por funções de ativação monotônicas (BISHOP, 1995). Uma função de ativação comumente aplicada às unidades escondidas de redes MLP´s é a função sigmoidal

que é não-linear e continuamente diferençável (exemplos: Função Logística e a Função Tangente Hiperbólica).

Já nas redes RBF, a ativação de uma unidade escondida é determinada por uma função não-linear da distância entre o vetor de entrada e um vetor de referência. As unidades escondidas de uma rede RBF possuem funções de ativação que são localizadas e apresentam base radial sobre seu domínio (BISHOP, 1995).

Uma MLP forma uma representação distribuída no espaço de valores de ativação para as unidades escondidas, pois para um vetor de entrada, muitas unidades escondidas contribuirão para a determinação do valor de saída (MLP´s tendem a resultar aproximações globais). A "interferência" e o "acoplamento cruzado" entre as unidades escondidas levam a resultados (processo de treinamento da rede) que são muito não-lineares, resultando em problemas de mínimos locais ou em regiões quase planas na função de erro, fatores estes que podem levar a uma convergência muito lenta (SILVA, 2003).

RBF´s possuem funções de base localizadas que formam uma representação no espaço de unidades escondidas que é local em relação ao espaço de entrada porque, para um vetor de entrada, tipicamente apenas algumas unidades escondidas apresentarão ativações significantes. Por isso, as RBF´s tendem a produzir aproximações locais (HAYKIN, 1998).

MLP´s têm muitas camadas de pesos e um complexo padrão de conectividade, de modo que nem todos os possíveis pesos em uma dada camada podem estar presentes. E uma variedade de diferentes funções de ativação podem ser utilizadas na mesma rede (BISHOP, 1995).

Uma RBF tem uma arquitetura simples, consistindo de duas camadas de pesos, em que a primeira contém os parâmetros das funções de base radial, e a segunda forma contém combinações lineares das ativações das funções de base radial para gerar a saída (BISHOP, 1995).

Os parâmetros de uma MLP são usualmente determinados ao mesmo tempo (estratégia global de treinamento, envolvendo treinamento supervisionado). Este tipo de treinamento apresenta um alto custo computacional, pela necessidade de retro-propagação do erro, o que faz as MLP´s terem um aprendizado muito lento. Porém o desempenho de generalização é bom (BISHOP, 1995).

Enquanto que uma RBF é treinada em 2 estágios, com as funções de base radial sendo determinadas primeiramente por técnicas não-supervisionadas, usando para tal os dados de entrada e a segunda camada (pesos) sendo após determinada por métodos lineares supervisionados, de rápida convergência (BISHOP, 1995).

A diferente estratégia de treinamento e a conseqüente diferença de velocidade de treinamento entre as duas redes faz com que as MLP´s sejam menos adequadas do que as RBF´s em operações dinâmicas, que envolvam aprendizado continuado (predição de séries temporais e aplicações on-line) (HAYKIN, 2001).

No contexto de aproximação de funções, sob idênticas condições do ambiente no qual estão inseridas, de uma forma geral pode-se afirmar que (BISHOP, 1995):

· erro final atingido por uma RBF é menor que o de uma MLP;

· a convergência de uma RBF pode chegar a uma ordem de grandeza mais rápida do que a convergência de uma MLP;

· a capacidade de generalização da MLP é, em geral, superior a capacidade de generalização da RBF.

A ferramenta WEKA (utilizada no estudo de caso desenvolvido neste trabalho) oferece classes para a aplicação de ambas as redes (RBF e MLP). Em testes preliminares na ferramenta, verificou-se que:

- a rede RBF obteve uma aproximação melhor (menor taxa de erro) em menos épocas de execução (convergência mais rápida);

- a implementação do algoritmo da rede MLP na ferramenta apresenta alto custo de execução, exigindo maior processamento e memória (pelo menos 1 GB para o conjunto de dados utilizado no estudo de caso). Além disso, a convergência mais lenta e a taxa de erro superior, determinaram a escolha da rede RBF como arquitetura de rede neural a ser utilizada no estudo de caso.

5.2 Árvores de decisão

Amplamente utilizadas em algoritmos de classificação, as árvores de decisão são representações simples do conhecimento e, um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados (GARCIA, 2000).

As árvores de decisão consistem de nodos que representam os atributos, de arcos, provenientes destes nodos e que recebem os valores possíveis para estes atributos, e de nodos folha, que representam as diferentes classes de um conjunto de treinamento (HOLSHEIMER, 1994).

Uma árvore de decisão tem a função de particionar recursivamente um conjunto de treinamento, até que cada subconjunto obtido deste particionamento contenha casos de uma única classe. Para atingir esta meta, a técnica de árvores de decisão examina e compara a distribuição de classes durante a construção da árvore. Os resultados obtidos após a construção de uma árvore de decisão são dados organizados de maneira compacta, que são utilizados para classificar novos casos (HOLSHEIMER, 1994). A Figura 5.5 apresenta um exemplo de árvore de decisão.

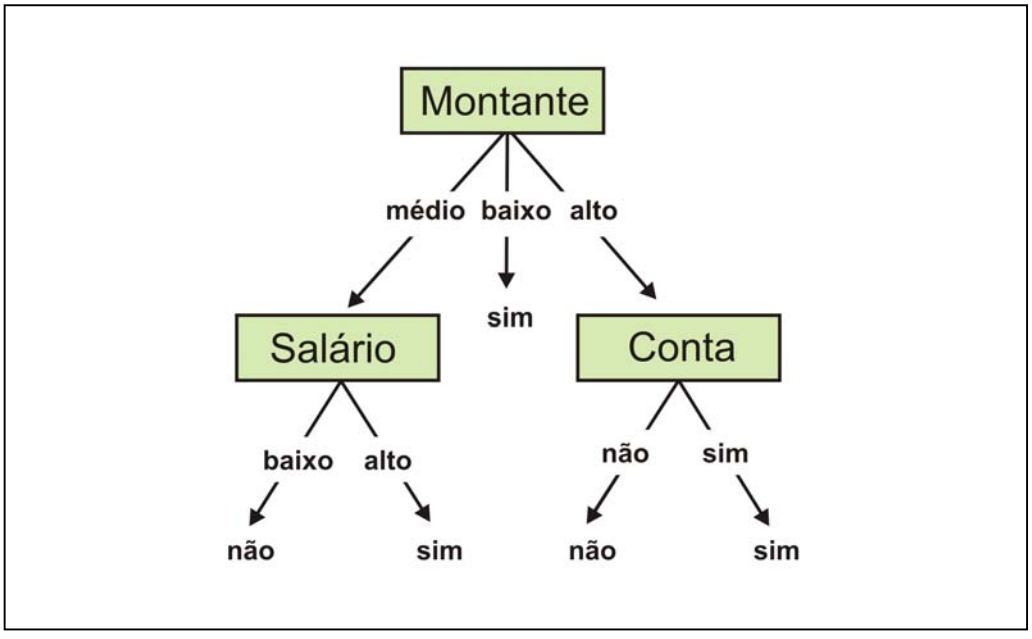

Figura 5.5: exemplo de árvore de decisão

Fonte: Garcia (2000).

No exemplo da Figura 5.5, são trabalhados objetos que relatam as condições propícias de uma pessoa receber ou não um empréstimo. É considerada a probabilidade do montante do empréstimo ser médio, baixo ou alto. Alguns objetos são exemplos positivos de uma classe sim, ou seja, os requisitos exigidos a uma pessoa, por um banco, são satisfatórios à concessão de um empréstimo, e outros são negativos, onde os requisitos exigidos não são satisfatórios à concessão de um empréstimo. Classificação, neste caso, é a construção de uma estrutura de árvore, que pode ser usada para classificar corretamente todos os objetos do conjunto (BRAZDIL, 1999).

A partir de uma árvore de decisão é possível derivar regras. As regras são escritas considerando o trajeto do nodo raiz até uma folha da árvore. Estes dois métodos são geralmente utilizados em conjunto. Devido ao fato das árvores de decisão tenderem a crescer muito, de acordo com algumas aplicações, elas são muitas vezes substituídas pelas regras. Isto acontece em virtude das regras poderem ser facilmente modularizadas. Uma regra pode ser compreendida sem que haja a necessidade de se referenciar outras regras (INGARGIOLA, 1996).

Com base na árvore de decisão apresentada na Figura 5.5, pode-se exemplificar a derivação de regras. Dois exemplos de regras obtidas a partir desta árvore são mostrados a seguir:

Se montante = médio e salário = baixo

então classe = não

Se montante = médio e salário = alto

então classe = sim

Existem diferentes algoritmos de classificação que elaboram árvores de decisão. Não existe uma regra que determine parâmetros de performance para a definição do melhor algoritmo e, como no caso das redes neurais, cada algoritmo pode apresentar desempenho satisfatório em função do problema em análise.

A formação de uma árvore de decisão segue os seguintes passos (BRAGA et al., 2004):

1) associar a partição do nó-raiz ao espaço de objetos;

2) verificar se o nó atual é um nó folha checando se pelo menos um dos seguintes quesitos é verdadeiro:

- todos os objetos contidos na partição do nó atual são da mesma classe;
- todos os atributos de objetos já foram utilizados no teste de algum nó no caminho deste até a raiz;
- a quantidade de objetos na partição do nó atual é inferior ao limite estabelecido (o limite mínimo é 1).
- no caso do nó atual ser uma folha, encerrar a exploração deste.

3) dividir a partição do nó atual segundo um atributo que não foi utilizado em nenhum outro teste sobre atributo no caminho entre o nó atual e o nó raiz;

4) aplicar recursivamente o passo 2 e 3 do algoritmo para cada nó filho do nó atual.

Após a construção de uma árvore de decisão é importante avaliá-la. Esta avaliação é realizada através da utilização de dados que não tenham sido usados no treinamento. Esta estratégia permite estimar como a árvore generaliza os dados e se adapta a novas situações, podendo, também, se estimar a proporção de erros e acertos ocorridos na construção da árvore (BRAZDIL, 1999).

Existem questões a serem superadas em qualquer algoritmo de construção de Árvores de Decisão para que esta seja ótima em quesitos como altura, eficiência de classificação, tempo de construção, entre outros. Alguns destes, que ainda hoje são tema de pesquisa, são listados a seguir (BRAGA et al., 2004):

· escolha da melhor partição para um nó ¾ em geral, por escolha do atributo.

· estratégias para limitação no crescimento da árvore.

· tratamento de valores desconhecidos no conjunto objetos para treino e para teste.

· partições baseadas em características discretas contínuas.

O algoritmo ID3 (*Inductive Decision Tree*) foi um dos primeiros algoritmos de árvore de decisão, tendo sua elaboração baseada em sistemas de inferência e em conceitos de sistemas de aprendizagem. Proposto por Quinlan (1986), é aplicável para conjuntos de objetos com atributos discretos ou discretizados. Este algoritmo possui uma implementação simples e um bom desempenho, o que levou a ser este um dos mais populares.

Sua estrutura é a mesma do algoritmo básico apresentado anteriormente; a inovação reside no critério de seleção de atributos para o particionamento, que é o Critério da Entropia. Como características principais do algoritmo ID3 citam-se (BRAGA et al., 2004):

· espaço de busca de hipóteses (árvores) é completo; ou seja, não há o risco de a melhor árvore (*target function*) não estar neste espaço.

· ID3 não realiza *backtracking* na busca pela melhor árvore (ou seja, uma vez escolhido um atributo de teste em um nível particular da árvore, ele nunca retrocede a este nível para reconsiderar esta escolha). Por isso, há o risco da solução encontrada corresponder a uma solução ótima local;

· ID3 usa todas as instâncias do conjunto de treino em cada passo da busca devido à seleção de atributos baseada em medidas estatísticas. Com isso, este algoritmo é menos sensível à presença de instâncias erroneamente classificadas ou com atributos sem valores.

Após a popularização do algoritmo ID3, foram elaborados diversos algoritmos, sendo os mais conhecidos: C4.5, CART (*Classification and Regression Trees*), CHAID (*Chi Square Automatic Interaction Detection*), entre outros (GARCIA, 2000).

O algoritmo C4.5 é uma extensão do ID3, com alguns aperfeiçoamentos, dos quais pode-se citar principalmente: redução em erros de poda na árvore; implementação de regras de verificação após poda; capacidade de trabalhar com atributos contínuos; capacidade de trabalhar com atributos de valores ausentes e aperfeiçoamento da eficiência computacional.

Uma implementação do algoritmo C4.5 é implementada no classificador J48, disponibilizado pela ferramenta WEKA – e será utilizado no estudo de caso como ferramenta de classificação utilizando a metodologia de árvores de decisão.

5.3 Classificadores Bayesianos

Thomas Bayes sugeriu uma regra (regra de Bayes, publicada em 1763) possibilitando que a probabilidade de um evento possa ser dada com base no conhecimento humano, ou seja, em eventos nos quais não se pode medir a freqüência com que ocorrem - a probabilidade pode ser dada com base no conhecimento que um especialista tem sobre o mesmo (HRUSCHKA et al., 1997).

A estatística *bayesiana* passou a ser aplicada em sistemas de Inteligência Artificial (IA) no início dos anos 60. Naquela época, o formalismo da utilização de probabilidades condicionais ainda não estava bem definido (MELLO, 2002).

Hruschka et al. (1997) define a probabilidade *bayesiana* como uma teoria consistente e que permite a representação de conhecimentos certos e incertos sobre condições de incerteza via distribuição conjunta de probabilidades. Tal distribuição conjunta pode ser representada pelo produto de distribuições condicionadas, como, por exemplo:

$$P(X1,X2,X3,X4,X5,X6)=P(X6|X5)\ P(X5|X2,X3)\ P(X2|X1)\ P(X4|X1)\ P(X3|X1)\ P(X1)$$

Uma variável é condicionada a uma ou mais variáveis numa relação causal. Uma distribuição pode ser representada por um grafo orientado. No grafo, cada nó representa uma variável do modelo, e os arcos ligam as variáveis que estão em relação direta causa / efeito. Por exemplo, se houvesse uma pesquisa com pessoas que passam pela rua, que perguntasse: que dia é hoje? Sem dúvida, a resposta da maioria seria a mesma. Isso porque a maioria das pessoas segue um mesmo calendário e isso causa o fato das respostas serem aproximadamente iguais. Sendo assim, caso deseja-se saber qual será a resposta do próximo entrevistado, não é necessário observar todas as respostas anteriores; basta observar a causa, ou seja, o fato do calendário ser utilizado pela maioria da população. A esta estrutura gráfica, com a quantificação de crença nas variáveis e seus relacionamentos, dá-se o nome de Redes *Bayesianas* (RB) ou Redes Causais (MELLO, 2002).

Uma rede *bayesiana* é um gráfico direcionado acíclico (ou DAG - *Directed Acyclic Graph*) com uma distribuição de probabilidades condicionais para cada nó. Cada nó *x* representando uma variável de domínio, e cada arco representando uma dependência

condicional entre os nós (CHENG et al., 1999). No aprendizado em redes *bayesianas*, usa-se o nó para representar atributos dos conjuntos de dados, como representado na Figura 5.6.

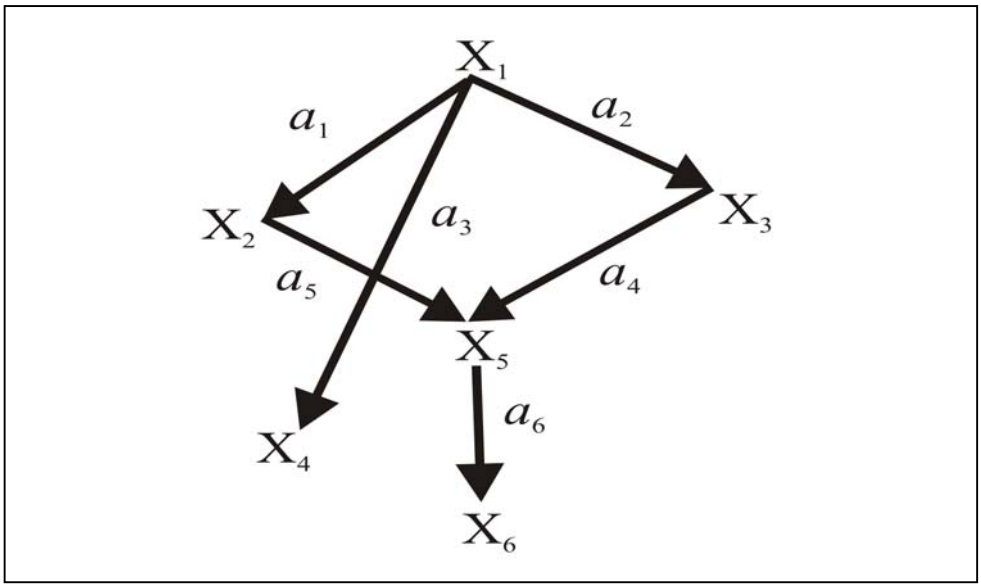

Figura 5.6: grafo da Rede Bayesiana para a distribuição P(X1,X2,X3,X4,X5,X6)
Fonte: Mello (2000).

Para verificar se um grafo *DAG* é uma rede *bayesiana*, precisa-se verificar se cada variável *x* do grafo deve ser condicionalmente independente de todos os nós que não são seus descendentes exceto seus pais (PERL, 1988). Esta condição permite reduzir consideravelmente o esforço computacional, porque existe uma explosão combinatória no cálculo da distribuição conjunta das probabilidades.

Para reduzir o esforço computacional, basta explorar as distribuições das relações entre as variáveis do problema (PERL, 1988). Graças a esta distribuição foram desenvolvidos vários algoritmos de propagação de crença em RB, os quais permitiram, sobre uma rede, propagar o conhecimento de novos fatos. Propagação de crenças corresponde em estabelecer um procedimento que, explorando a conectividade da rede, permita a determinação das distribuições de probabilidades das variáveis, objetivo do diagnóstico, condicionadas aos valores das variáveis que representam evidências. Assim, o conhecimento incerto pode ser atualizado de forma simples e clara, mantendo a consistência e a confiabilidade.

As redes *bayesianas* podem ser visualizadas de duas maneiras:

- como uma estrutura que codifica um grupo de relacionamentos de independência condicional entre os nós, conforme um conceito chamado

de *d-serapação* (quando uma RB implica mais independência condicional do que geralmente aquelas independências envolvidas com os pais de um nó.). A idéia é que uma estrutura de RB possa ser instruída pelo mecanismo de aprendizagem com a independência condicional entre os nós da rede. Usando alguns testes estatísticos, pode-se encontrar a relação de independência condicional entre os atributos e usar o relacionamento como forma para construção de uma RB;

- como uma estrutura que codifica a união distributiva dos atributos. Então uma RB pode ser usada como um classificador que dá a posterior probabilidade distributiva do nó de classificação, dados os valores de outros atributos.

Com a RB pode-se representar problemas do mundo real em que existam relações de causa e conseqüência entre as variáveis. Isso motiva o uso desses tipos de redes porque os seres humanos têm uma certa necessidade de representar o conhecimento utilizando fatos em forma de relacionamentos causais (HRUSCHKA et al., 1997).

As RB's também possuem a possibilidade de realizar o aprendizado a partir dos dados. Nesse aprendizado, oferece-se uma amostra e o sistema, através de um algoritmo, gera uma estrutura que melhor se adapta aos dados do problema (HECKERMAN, 1995). Existem vários algoritmos e métodos para o aprendizado de RB's a partir dos dados, sendo que cada um se adapta melhor a uma determinada classe de problema (MELLO, 2002).

5.3.1 O Classificador *Naive Bayes*

As redes *bayesianas* têm sido usadas em processos de classificação há muitos anos. Quando sua estrutura apresentar uma classe como nó pai de todos os outros nós e nenhuma outra conexão é permitida, torna-se ideal para os processos de classificação. Esta estrutura é comumente chamada de redes *Naive Bayes*, que é um caso especial de redes probabilísticas ou redes *bayesianas* (MELLO, 2002).

O princípio básico de classificadores *bayesianos* está fundamentado na teoria da probabilidade *bayesiana* (DUDA et al., 2000). Os classificadores *bayesianos* são capazes de encontrar regras que respondem a perguntas do tipo:

- qual a probabilidade de se jogar tênis dado que o dia está ensolarado, com temperatura quente, umidade alta e vento fraco? Em termos probabilísticos, essa pergunta equivale a P(JOGAR TÊNIS = Sim | [Ensolarado, Quente, Alta, Fraco]);

- qual a probabilidade de NÃO se jogar tênis dado que o dia está ensolarado, com temperatura quente, umidade alta e vento fraco? Em termos probabilísticos essa pergunta equivale a P(JOGAR TÊNIS = Não | [Ensolarado, Quente, Alta, Fraco]).

Na representação de gráfico acíclico de uma rede *bayesiana*, cada nó representa um atributo (interpretado como uma variável randômica), que é usado para descrever um domínio de interesse, e cada ligação representa uma dependência entre os atributos. O gráfico mostra uma particular junção de probabilidade distributiva, onde cada nó da rede representa uma probabilidade condicional distributiva entre os valores dos atributos (MELLO, 2000).

Uma rede *bayesiana* como classificador *Naive Bayes* apresenta o gráfico em forma de estrela, no qual o centro da estrela é a classe que será classificada. Como pode ser observado na Figura 5.7, os atributos formam as pontas da estrela (A_1 a A_n). A única conexão possível é cada atributo com a classe (C_i) . Nenhuma outra conexão é permitida na rede *Naive Bayes* (MELLO, 2000).

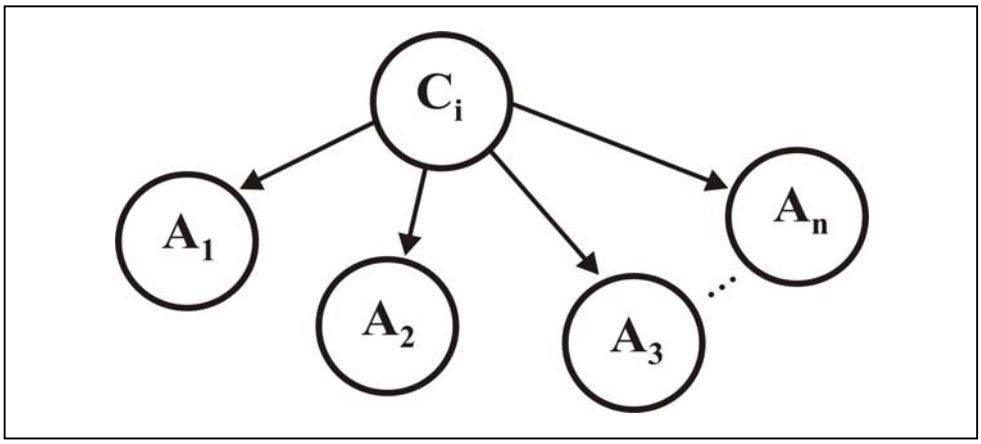

Figura 5.7: estrutura da rede Naive Bayes em estrela

Fonte: Mello (2000).

A rede *Naive Bayes* tem sido usada por pesquisadores, em classificações há muitos anos, por apresentar características vantajosas sobre outros tipos de classificadores, tais como:

- facilidade na construção de seu algoritmo: pela simplicidade do seu algoritmo, estimulou pesquisadores a aplicar este método em muitas ferramentas (CHEN et al., 1997);

- o processo de classificação é muito eficiente quando os atributos são independentes entre si: em situações onde os atributos não são correlacionados, o classificador *Naive Bayes* sobressai surpreendentemente sobre muitos sofisticados classificadores. Esta característica é rara na prática de aprendizagem. Isso ocorre porque a rede *Naive Bayes* apresenta uma limitação nas ligações entre os nós (conforme Figura 5.7);

- é rápido na aprendizagem e predição: seu tempo de aprendizagem e de predição é linear independentemente do número de exemplos. Pesquisadores têm desenvolvido trabalhos comparando e ressaltando o bom desempenho do classificador *Naive Bayes* em relação a outros modelos de classificação complexos - um comparativo é descrito por Friedman et al. (1997).

É interessante citar que existem fatos que podem gerar problemas nas predições do classificador *Naive Bayes*, formando algumas limitações para o algoritmo, como por exemplo (MELLO, 2000):

- trabalhar com valores com casas decimais. Os erros causados por arredondamentos podem causar variações na predição;

- os exemplos devem apresentar atributos com independência condicional, caso contrário, o *Naive Bayes* se torna inviável;

Apesar desses fatores, o classificador *Naive Bayes* ainda se torna eficiente para utilização em aplicações que envolvem predição. Sua facilidade de implementação e seu desempenho colocam o algoritmo como um dos mais citados classificadores nas pesquisas na área de Inteligência Artificial (MELLO, 2000).

5.4 Resumo

Apresentou-se neste capítulo um estudo sobre três algoritmos utilizados no processo de Classificação e Predição. Em função de suas características e adaptabilidade ao modelo desenvolvido nesta dissertação, foram avaliadas as seguintes técnicas: Redes Neurais (particularmente as redes RBF – *Radial Basis Function*); Árvores de Decisão (principalmente o algoritmo C4.5 ou J48, na sua implementação na ferramenta WEKA) e Classificadores Bayesianos (com foco no algoritmo *Naive Bayes*).

A escolha destas técnicas (entre outros algoritmos disponíveis para a descoberta de padrões no modelo proposto) deu-se em função da sua disponibilidade (pois são encontrados no pacote de classes Java da ferramenta WEKA, que foi utilizada no Estudo de Caso); da performance dos algoritmos (ponto essencial de avaliação, considerando-se a natureza e o tamanho das bases de dados em estudo) e do custo (principalmente sob a ótica dos recursos necessários para a execução das ferramentas).

A abordagem exploratória em cada uma das técnicas teve foco principal nas potencialidades de cada algoritmo sobre o problema em questão. A revisão sobre a fundamentação teórica e matemática foi importante para a compreensão dos principais parâmetros e variáveis de cada ferramenta (essencial na busca do melhor conjunto de dados de configuração de uso de cada algoritmo).

A utilização de três algoritmos diferenciados tem como objetivo a comparação entre as técnicas, buscando a melhor ferramenta a ser aplicada em casos reais futuros derivados do Estudo de Caso proposto.

6 MODELAGEM DO SISTEMA

O modelo definido neste trabalho busca a extração de padrões e descoberta de conhecimento sobre a base do sistema de CRM da operadora, buscando a Classificação e Predição sobre uma classe de usuários, denominada *churners* (clientes dispostos a efetuarem o cancelamento do serviço, ou seja, seu terminal de voz residencial).

A estrutura básica do modelo é formada pela metodologia do ciclo de descoberta de conhecimento em bases de dados, proposto principalmente por Fayyad et al. (1996), Cabena et al. (1997), Cios (2000), Han et al. (2001) e Klösgen et al. (2002).

No entanto, considera-se a proposta da metodologia CRISP-DM (CRISP, 2000), no que tange à flexibilização na seqüência das fases. Assim, dependendo do resultado de cada etapa ou tarefa, pode-se retornar a uma (ou mais) tarefa (s) anterior (es) de modo iterativo até que se atinjam os resultados esperados.

O modelo preditivo de *churn* desenvolvido é uma ferramenta com embasamento matemático/estatístico, construído com base em um grupo de variáveis, que, por conhecimento prévio do negócio ou estudo específico, demonstraram ser relevantes para a etapa de classificação. De posse dos dados classificados, a operadora pode desenhar campanhas de retenção focadas em uma base restrita de clientes, exigindo investimentos menores na ação.

A formatação deste modelo busca um equilíbrio entre o conjunto de dados ideal e o custo associado à sua consolidação. Assim, mesmo que o conhecimento de um especialista indique que determinada variável tenha importância na predição do *churn* em telecomunicações, este parâmetro só poderá ser inserido no modelo caso esteja disponível para cada instância da base de dados de treinamento e se os sistemas que geram os *data marts* possam disponibilizar tal informação com um custo aceitável. A natureza diversificada e heterogênea dos sistemas (principalmente em uma operadora de telecomunicações) implica em dificuldades e custos excessivos para a obtenção do conjunto de dados ideal. Para compensar lacunas de variáveis, busca-se a otimização das informações disponíveis, através de etapas consistentes de pré-processamento e transformação dos dados.

6.1 Estrutura principal

A Figura 6.1 ilustra o modelo implementado.

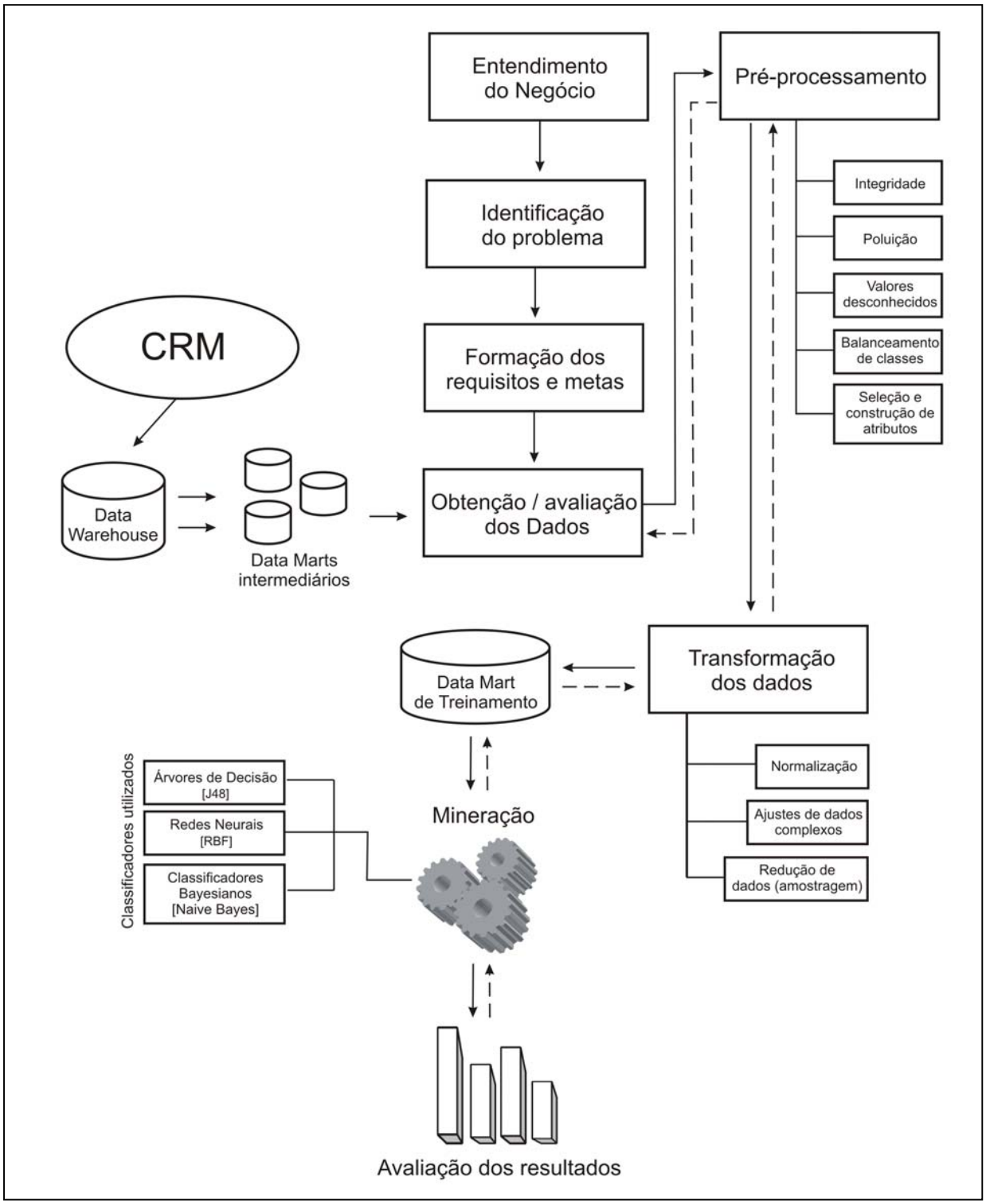

Figura 6.1: estrutura principal do modelo de classificação

As etapas iniciais compreendem o entendimento do negócio e a identificação do problema. No sistema em questão, o problema central é a predição da classe de usuários *churners*. O entendimento do negócio envolve o estudo do atual cenário de telecomunicações no Brasil, das causas e efeitos do *churn* e das tendências no mercado (convergência, novos serviços e aplicações).

A fase de obtenção e avaliação dos dados compreende uma pesquisa (entre as diversas fontes existentes na organização) sobre a disponibilidade, precisão, validade, custo de aquisição e consistência dos dados. Assim, forma-se um conjunto de *data marts* intermediários, que serão tratados na etapa seguinte.

O pré-processamento avalia a integridade das informações disponibilizadas. Nesta etapa, são aplicadas as principais técnicas e algoritmos para tratamento e limpeza dos dados. O balanceamento de classes (*oversampling*) é tarefa essencial nesta fase, devido ao desequilíbrio existente entre objetos das duas classes objeto da predição (*churner ou* fidelizado), pois a ocorrência da classe *churner* é significativamente inferior. Também é objeto de tratamento nesta etapa a seleção de atributos (onde variáveis menos relevantes ao aprendizado dos algoritmos são descartadas) e a construção de atributos (combinação de dados ou cálculo para a definição de uma nova informação).

Na fase de transformação dos dados, são aplicadas técnicas descritas de tratamento sobre dados complexos (como variáveis temporais – ou *datetime*, por exemplo). Esta etapa também é responsável pela redução dos dados (tarefa essencial no modelo, em função da natureza dos dados – bases de informação de tamanhos superiores à capacidade de processamento dos algoritmos de classificação). A seleção correta da amostragem é essencial para a correta elaboração do *data mart* de treinamento final.

Com o conjunto de dados de treino definido, inicia-se a fase de execução de algoritmos de Classificação e Predição. Para esta fase, será utilizado o conjunto de algoritmos e ferramentas disponibilizados na aplicação *open-source* WEKA (ferramenta Java desenvolvida na *University of Waikato*, Nova Zelândia).

Como classificadores serão utilizados os seguintes algoritmos (estudados no capítulo anterior):

- Redes Neurais – particularmente o modelo *RBF Network – Radial Basis Function Network*;

- *Decision Trees* (Árvores de Decisão) – particularmente o algoritmo C4.5 (através do classificador J48, da ferramenta WEKA);

- Classificadores Bayesianos (principalmente o método Naive Bayes).

A fase de avaliação dos resultados determinará a eficácia do modelo através de testes abertos e fechados. A utilização de três técnicas diferentes de predição tem como objetivo a comparação dos resultados e a validação do sistema desenvolvido.

Nesta etapa são efetuados testes com os diversos parâmetros que cada algoritmo possuiu, visando alcançar a configuração mais próxima do resultado ótimo para o conjunto de dados do *data mart* de treinamento.

Como exemplo de parâmetros de configuração disponíveis na ferramenta WEKA nos algoritmos selecionados, pode-se citar:

- no classificador J48 (algoritmo de árvores de decisão), pode-se variar a utilização ou não de particões binárias em atributos nominais na construção das árvores; o fator de confiança (utilizado na poda – valores menores implicam em maior poda) e o número mínimo de instâncias por folha;

- na Rede Neural RBF pode-se variar a semente de geração de *cluster* (que é repassada ao algoritmo *k-means*); o desvio padrão mínimo para os *clusters* e o número de clusters para geração no algoritmo *k-means*;

- no classificador Bayesiano (de estrutura mais simples) pode-se optar pela utilização de um processo de discretização supervisionada, utilizado na conversão de atributos numéricos em nominais.

O desempenho dos métodos é avaliado e comparado utilizando-se a Matriz de Confusão gerada pelos próprios algoritmos. A matriz ilustra de forma clara e simples a taxa de acerto na predição de cada classe construída no sistema.

Como parâmetros de resultados esperados, busca-se atingir valores próximos aos índices citados por Mozer et al. (2000):

Algoritmo utilizado	Predição correta no modelo
Redes Neurais	**68%**
Árvores de Decisão	**60%**

Tabela 6.1: parâmetros de resultados para a predição do modelo
Fonte: Mozer et al. (2000)

6.2 Estrutura e Formação dos Data Marts

A formação de um *data mart* final de treinamento para o modelo envolve:

- avaliação dos principais conjuntos de dados disponíveis;
- integração com o entendimento do negócio;
- redução de dados (amostragem);
- seleção de atributos.

O processo de mineração de dados proposto neste modelo utilizará uma composição de *data marts* para a formação do conjunto final de treinamento. Serão três *data marts* principais, extraídos e consolidados do CRM da organização:

- ***(1) data mart* de cadastro e relacionamento:** contém informações referentes ao cadastro do cliente na organização, como nome, CPF, endereço, número do terminal, localidade, data de início do serviço (que deverá ser convertido em um valor de tempo em meses), produtos e serviços em uso, etc;
- ***(2) data mart* de consumo:** contém informações referentes ao perfil de consumo de telefonia do cliente, em serviços de voz LDN (Longa Distância Nacional), LDI (Longa Distância Internarcional) e Local. São objetos ou atributos deste *data mart*: total de minutos de consumo por CSP

(Código de Seleção de Prestadora) em cada um dos tipos de tráfego de longa distância (Intra-Setor – dentro do estado; Intra-Região – na região 2 definida pela Anatel, ou a área de atuação da operadora e Inter-Região – destino fora da região 2). Os atributos de detalhe do consumo de voz local referem-se principalmente ao total de pulsos consumidos no período;

- ***(3) data mart* de faturamento:** contém informações resumidas e consolidadas sobre a conta ou faturamento do cliente. O *data mart* de faturamento contem três meses de histórico da conta de cada um dos clientes da amostra.

A Figura 6.2 ilustra a formação do conjunto de dados de treinamento, e a sua formação através de três *data marts* intermediários.

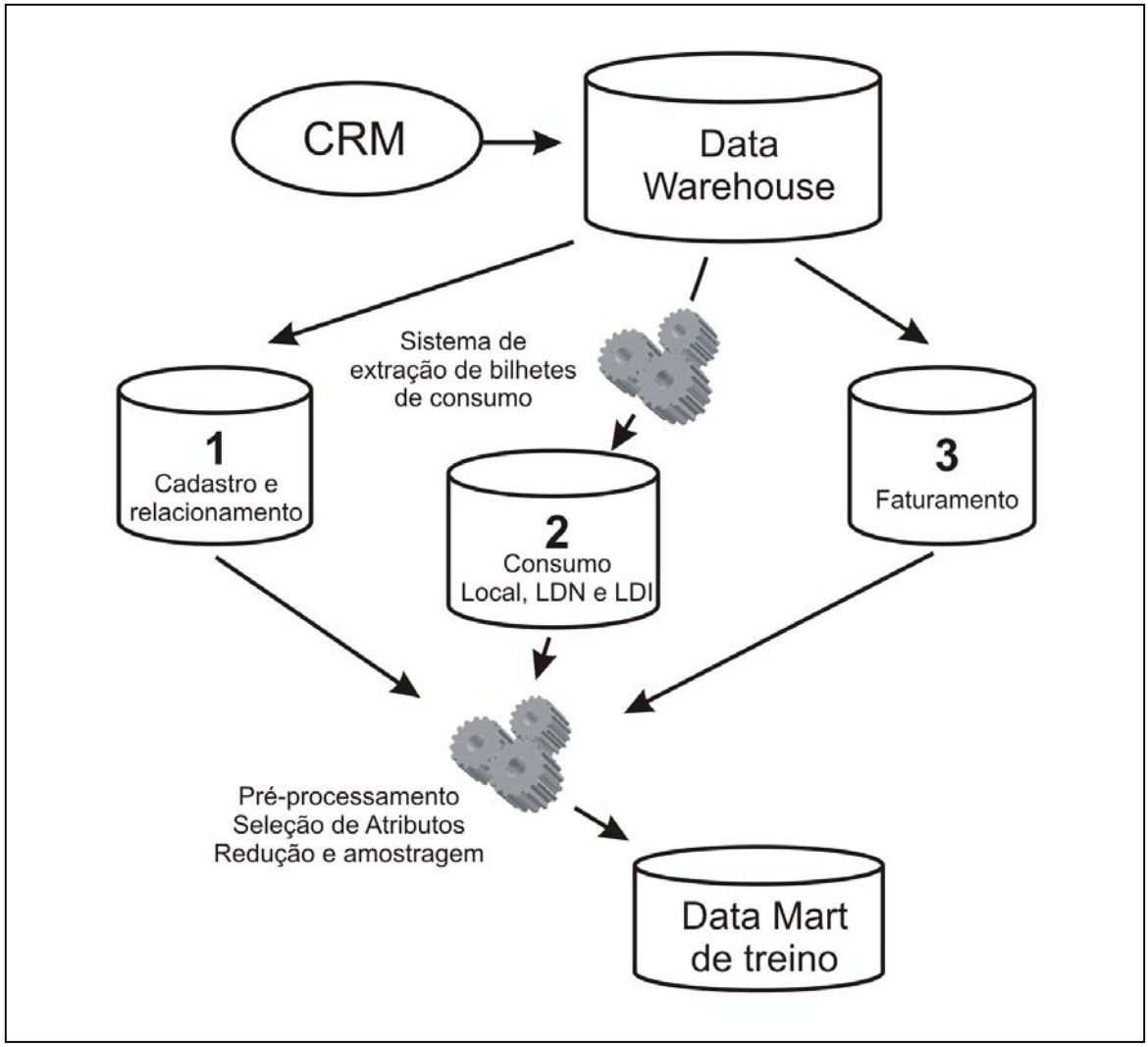

Figura 6.2: formação dos principais data marts do modelo

A extração e consolidação dos *data marts* dá-se, basicamente, através da solicitação de relatórios e consultas aos sistemas da organização, principalmente à módulos de geração de arquivos em lote do *Data Warehouse*, consolidados a partir de informações do sistema de CRM (com exceção do *data mart* de Consumo, que exigiu a elaboração de um sistema de construção de atributos, descrito a seguir). Cada um dos *data marts* intermediários é submetido então ao processo de Pré-processamento, Seleção de Atributos e Redução de dados, formando assim o *data mart* a ser utilizado na etapa de treinamento e classificação. A composição básica de cada um dos *data marts* intermediários é ilustrada nas Tabelas a seguir:

Data Mart (1) - Cadastro e relacionamento

Atributo	Exemplo de conteúdo	Tipo de dados
armario	<NULL>	nvarchar
categoria	11	nvarchar
cgc_cpf	12345678901	nvarchar
cnae	<NULL>	nvarchar
cod_ativ	<NULL>	nvarchar
cod_merc	M03	nvarchar
complemento	AP120	nvarchar
contrato	3278921	nvarchar
ddd	51	nvarchar
desc_merc	MASSA RES D	nvarchar
dt_instalacao	23/11/1986	datetime
estacao	MTZ	nvarchar
ident	F	nvarchar
insc_estadual	<NULL>	nvarchar
lograd_inst	PASTOR BOHN	nvarchar
nomeclie	JOSE ANTONIO DE MORAES	nvarchar
num_faixada	120	nvarchar
servicos	CATBIN0001BLO 0000JUMPER0000	nvarchar
sigla_loc_pta_a	PAE	nvarchar
terminal	5133283XXX	nvarchar
tip_lograd	R	nvarchar
tipo_terminal	0	nvarchar
uso	R	nvarchar

Tabela 6.2: estrutura inicial do Data Mart (1) – Cadastro e Relacionamento

Data Mart (2) - Consumo LDN / LDI / Local

	Atributo	Exemplo de conteúdo	Tipo de dados
Intra-Setor	assin_a	513328XXXX	varchar
	A14	18,00	float
	A15	<NULL>	float
	A21	5,00	float
	A23	<NULL>	float
	A25	<NULL>	float
	A31	<NULL>	float
	A36	<NULL>	float
	A41	<NULL>	float
Intra-Região	R14	27,00	float
	R15	<NULL>	float
	R21	0,68	float
	R23	<NULL>	float
	R25	<NULL>	float
	R31	<NULL>	float
	R36	<NULL>	float
	R41	<NULL>	float
Inter-Região	T14	4,78	float
	T15	<NULL>	float
	T21	<NULL>	float
	T23	<NULL>	float
	T25	<NULL>	float
	T31	<NULL>	float
	T36	<NULL>	float
	T41	<NULL>	float
Internacional	I14	9,00	float
	I15	<NULL>	float
	I21	<NULL>	float
	I23	<NULL>	float
	I25	<NULL>	float
	I31	<NULL>	float
	I36	<NULL>	float
	I41	<NULL>	float
Local	chamadas	29,00	float
	minutos	14,12	float
	pulsos	32,04	float

Tabela 6.3: estrutura inicial do Data Mart (2) – Consumo LDN / LDI / Local

Data Mart (3) - Faturamento

Atributo	Exemplo de conteúdo	Tipo de dados
assin_a	513328XXXX	varchar
Referencia	abr/06	varchar
Fat_liq	172,5	float
Fat_bruto	224,25	float

Tabela 6.4: estrutura inicial do Data Mart (3) – Faturamento

6.2.1 O sistema de tratamento de bilhetes de consumo

A obtenção do *data mart (*2) exigiu a construção de uma ferramenta para obtenção dos dados de consumo de telefonia dos clientes. Tal necessidade foi verificada em função da não disponibilização consolidada de dados desta natureza e no formato adequado pelos sistemas disponíveis (relatórios do CRM, consultas do *Data Warehouse* ou sistemas de produção). Assim, o *data mart* (2) baseia-se na metodologia de construção de novos atributos, citada por Bloedorn et al. (1998).

A importância do *data mart* (2) para o modelo deve-se principalmente ao perfil de informação nele encontrado. Os atributos descrevem basicamente a utilização do serviço pelo cliente. Entende-se que, quando o terminal em questão apresenta tráfego sainte ou originado, o cliente é fiel ao serviço, ou seja, sua necessidade de atendimento está de acordo com o produto utilizado.

Assim, enquanto a formação dos *data marts* (1) e (3) baseou-se principalmente na execução de consultas, extração de relatórios e busca de dados no *Data Warehouse* (principalmente em arquivos de lote, processados e disponibilizados pelo *DW*), a formação do *data mart* (2) tornou-se um desafio significativo para o prosseguimento da elaboração do modelo. Logicamente, a informação de consumo e tráfego de voz (informação base do *data mart* em questão) existe e é processada nos sistemas operacionais da companhia. No entanto, o formato desta informação não era adequado ao modelo, pois o dados existente é disponibilizado sob duas formas: (a) já consolidada e processada pelos sistemas de

faturamento ou, (b) com excessiva quantidade de informações, através do sistema de bilhetagem das centrais telefônicas.

Optou-se então para o desenvolvimento de um sistema que, utilizando as informações disponibilizadas pelo sistema de bilhetagem (controle das chamadas na rede de voz), consolidasse os atributos de cada chamada efetuada e formasse um grande repositório de bilhetes de tráfego. Em função do grande volume de dados gerado (mais de 1GB de dados / dia – ou seja, todo o tráfego de voz de usuários residenciais no estado do Rio Grande do Sul), a formatação deste *data mart* exige uma etapa de pré-processamento específica, onde utiliza-se a ferramenta de *Business Intelligence* desenvolvida pela SAS System. Ao final do mês, todos os arquivos diários de consumo armazenados são consolidados e totalizados, permitindo a formação de atributos únicos e agrupados (exemplo: em apenas um atributo, exibe-se a quantidade de chamadas locais efetuadas). O Anexo 01 exibe a função (*procedure*) principal do sistema de tratamento de bilhetes de consumo.

A quantidade aproximada de chamadas computadas pelo sistema é de aproximadamente 9 milhões em um dia útil, 6 milhões em sábados e 4 milhões aos domingos. A Figura 6.3 ilustra a atuação resumida do sistema implementado para a formação do *data mart* (2):

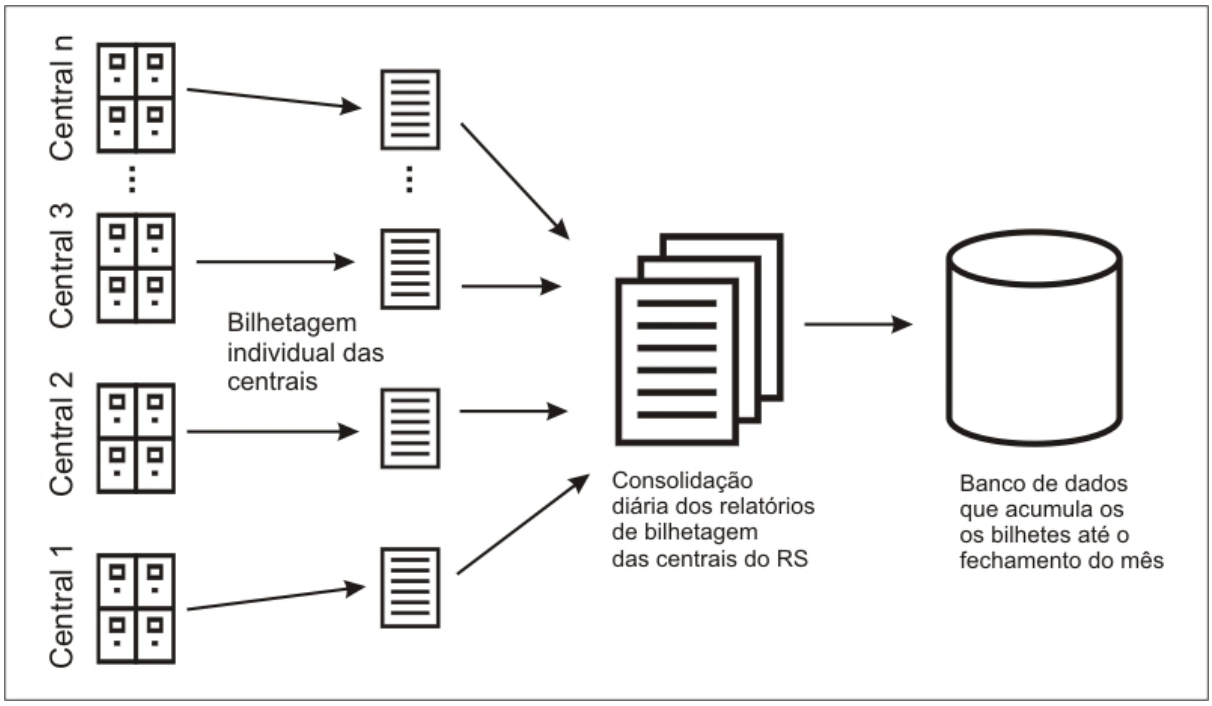

Figura 6.3: o sistema de tratamento de bilhetes de consumo

6.2.1 A formação do data mart de treinamento

A formação do *data mart* de treinamento é basicamente a junção dos três *data marts* intermediários. Na etapa final do processo (na execução do Estudo de Caso), aplica-se uma avaliação para seleção das variáveis que podem ser excluídas, deixando o modelo mais ágil para os algoritmos classificadores. Após este processo, obtém-se o conjunto de dados final a ser aplicado no treinamento. Como complemento, acrescenta a variável (ou classe alvo), ou seja, a indicação de quais clientes tornaram-se *churners* em período decorrente.

Conforme Ferreira (2005), em geral o procedimento utilizado na definição da variável alvo correta se inicia com a escolha de uma janela de previsão de *churn*. A janela de previsão ideal pode variar de uma semana até vários meses, dependendo da conjuntura de mercado e da situação da operadora.

Uma vez definido o tamanho da janela de previsão, é escolhido um momento do histórico de dados que servirá como período de base. Tal período de base deve possuir o mesmo tamanho da janela de previsão (semana, mês, etc). É necessário garantir a existência de pelo menos um período de tempo de mesmo tamanho após o período base, pois será neste primeiro período que será verificado o abandono do cliente, segundo as regras de negócio da empresa.

Assim, o modelo desenvolvido utiliza o esquema para a formação do *data mart* final de treinamento ilustrado na Figura 6.4:

Período: n (mês)									n+2
Data mart (1)				**Data mart (2)**				**Data mart (3)**		**Classe Alvo**
nomeclie	**local**	**terminal**	**...**	**A14**	**A15**	**A21**	**...**	**Med_Fat_liq**	**Med_Fat_bruto**	**churner**
JOSE...	PAE	5133283XXX	...	18,00	<NULL>	5,00	...	172,50	224,25	Não
PAULO...	SMA	5533204xxx	...	0,00	0,00	3,00		51,00	66,30	Sim

Figura 6.4: formação do data mart de treinamento

O atributo da classe alvo é referente ao período *n*+2, pois é o período a ser previsto. A utilização desta estrutura de janela de predição é citada por Ferreira (2005) e por Mozer et al. (2000). Como exemplo, caso o período *n* escolhido seja março / 2006, serão assinalados os atributos da classe alvo referente ao período de maio / 2006 (ou seja, clientes considerados como *churners* pela organização neste mês).

6.3 Resumo

Este capítulo apresentou a estrutura do sistema modelado para a predição do *churn*. Ilustrou-se a fluxo de informações através de um diagrama, consolidando todas as etapas do processo.

O modelo definido abrange as etapas principais citadas na literatura para uma aplicação de mineração de dados, iniciando com a avaliação e entendimento do problema, passando pelas fases de obtenção dos dados e de pré-processamento, finalizando com a aplicação dos algoritmos classificadores e a avaliação do resultado. Para a formação do modelo, utilizou-se principalmente processo de descoberta de conhecimento em bases de dados proposto por Fayyad et al. (1996) e CRISP-DM, principalmente a teoria de processo de mineração de dados não linear, citada em CRISP (2000).

A elaboração dos *data marts* levou em consideração os conjuntos de dados disponíveis nos sistemas da empresa, acessados através do *Data Warehouse*. Foram construídos três *data marts*: 1) cadastro e relacionamento; 2) Consumo LDN / LDI / Local e 3) Faturamento. Em função da dificuldade de obtenção de variáveis, entende-se que o conjunto de dados final formado não é o ideal. Em alguns casos, o custo e o tempo de obtenção de determinados atributos (principalmente variáveis relacionadas ao atendimento do cliente, como quantidade de chamadas ao *call center* ou solicitações de reparo) impediram que fossem inseridas no modelo. Buscou-se o equilíbrio entre custo de obtenção de informações e capacidade de aprendizado do sistema.

A formação do *data mart* 2 (Consumo LDN / LDI / Local) exigiu a formação de um sistema a parte, que consolida a bilhetagem de centrais e gera um extrato de uso de rede por cliente.

7 ESTUDO DE CASO

O modelo definido no capítulo anterior será utilizado como base para a aplicação do Estudo de Caso, a ser efetuado sobre uma base de dados real da operadora de telecomunicações Brasil Telecom S.A.

Serão apresentadas as etapas que formam o ciclo de descoberta de conhecimento em bases de dados, aplicadas sobre os conjuntos de informações adquiridos na construção do modelo do sistema. Descrevem-se as principais técnicas e algoritmos aplicados nos *data marts* intermediários, que permitiram a formação do conjunto de dados final utilizado na fase de treinamento e validação de resultados.

7.1 Objetivos do Estudo de Caso

Como objetivos deste Estudo de Caso listam-se os seguintes:

- validar o modelo de mineração de dados com enfoque em *churn* preditivo, proposto no capítulo anterior;
- descrever as etapas de mineração de dados (principalmente as fases de pré-processamento e transformação), servindo como referencial para a aplicação e evolução do sistema;
- suprimir as limitações da base de informações disponível (impostas pela dificuldade e pelo custo da obtenção do conjunto de dados ideal), através da correta aplicação das técnicas de pré-processamento e transformações de dados;
- atingir uma taxa de predição aceitável para a consolidação do sistema, de acordo com os valores citados em Mozer et al. (2000) – ilustrados em tabela no capítulo anterior, permitindo sua aplicação no ambiente real da empresa.

7.2 Estrutura do Estudo de Caso

A estrutura deste Estudo de Caso utilizará principalmente a metodologia CRISP-DM (CRISP, 2000), em conjunto com a teoria de descoberta de conhecimento em bases de dados, prosposta por Fayyad et al. (1996).

Assim, o Estudo de Caso será conduzido através das etapas descritas a seguir:

- Entendimento do Negócio;
- Compreensão dos Dados e Pré-Processamento;
- Transformação dos Dados;
- Classificação e Predição;
- Avaliação dos Resultados.

A revisão bibliográfica do Capítulo 3 serviu de embasamento teórico para as etapas descritas a seguir, referenciando o detalhamento das técnicas e algoritmos citados.

7.3 Entendimento do Negócio

O problema do *churn* e suas implicações (descritos no Capítulo 2), exige atenção especial das operadoras de telefonia em todo o mundo, independente da sua abrangência, ramo (fixo ou móvel) ou concessão de atuação.

Conforme Cister (2005), a previsão de que, provavelmente, os clientes mudarão de fornecedor ou de tecnologia de serviços e a determinação de incentivos eficazes, do ponto de vista de custo, para persuadi-los a continuar são iniciativas muito difíceis para a maioria das empresas de telecomunicações. Os volumes de dados necessários são elevados e, freqüentemente, difíceis de acessar e consolidar por meio de ferramentas convencionais. Diversas razões levam um assinante a abandonar sua operadora de telecomunicações. Os custos para se conquistar um novo cliente já foram razoavelmente dimensionados por estas empresas. O planejamento de negócios das operadoras locais estão sendo modificados para

adicionar mais um campo de custos: o custo do *churn*. Isto significa, a médio prazo, uma diminuição da margem de lucro operacional das empresas de telefonia local.

Alguns clientes buscam serviços de melhor qualidade, novas tecnologias ou serviços avançados. Uma grande maioria procura as melhores tarifas, pois a tendência é que cada dia haja menor diferenciação quanto à qualidade de serviço entre as operadoras de telecomunicações. Alguns assinantes, simplesmente, mudam de operadoras por dificuldades financeiras (impedidos de quitarem suas faturas - clientes inadimplentes que dão origem ao *churn* involuntário). Outros podem ser seduzidos por campanhas de marketing, promoções e prêmios. Os três últimos tipos de clientes são, efetivamente, os grandes responsáveis pelo *churn* (CISTER, 2005).

Este experimento visa classificar/predizer clientes que possuam inclinação e/ou tendência a evadirem da base de dados da empresa. A população da base de treinamento foi composta de usuários do STFC (Serviço Telefônico Fixo Comutado) da Brasil Telecom S.A. no estado do Rio Grande do Sul.

A definição de classes investigativas do problema é simples e objetiva - duas opções de predição são possíveis para cada usuário ou componente da população do conjunto de dados de treinamento:

- **Classe *Churner*** (identificada como "C" no atributo correspondente da base): usuário que possui tendência a evadir a base de clientes, cancelando o seu terminal residencial, de acordo com a janela de predição do modelo;

- **Classe *Não-Churner*** ou Usuário Fidelizado (identificada como "NC" no atributo correspondente da base): usuário com perfil e padrões identificados como um cliente que deve permanecer na base, mantendo a utilização do serviço.

Finalizado o processo de mineração de dados, a operadora terá meios para desenvolver planos e ações visando a manutenção dos clientes identificados como *churners* potenciais. Entre as ações possíveis, destacam-se, por exemplo:

- abordagem individual, através de mala direta ou contato via *call center*, oferecendo um novo plano ou serviço (ou mesmo uma combinação de produtos, como acesso Banda Larga e celular);
- ação regional, através de planos de mídia ou eventos localizados em áreas geográficas identificadas com alto índice de *churn* potencial.

7.4 Compreensão dos Dados e Pré-Processamento

Esta etapa do processo de descoberta de conhecimento em bases de dados é de significativa importância ao modelo desenvolvido nesta dissertação, em função da natureza da base de informações (oriunda de sistemas diversos e com grande quantidade de registros). Descreve-se a seguir as principais técnicas aplicadas sobre os *data marts* utilizados no modelo.

7.4.1 Limpeza e Validação dos Dados

Os conjuntos de dados do modelo foram extraídos de fontes distintas dos sistemas da empresa, como por exemplo: bases do *Data Warehouse*, relatórios do CRM e aplicações de áreas de negócio (como os atributos de consumo, construídos por um sistema à parte). Devido à natureza heterogênea das informações, foram efetuados tratamentos em alguns atributos específicos, listados a seguir.

- **Identificação de inconsistências:** no campo descrição do mercado (ou "Desc_mercado", que identifica a classe de consumo do usuário), existiam diferenças na representação da mesma informação. Exemplo: para definir a categoria Massa Residencial Ouro, foram encontradas ocorrências de "MASSA RES O" e "MASSA O". Unificou-se a informação para "MASSA RES O";
- **Identificação de poluição e verificação de integridade:** o campo "Desc_merc" também exigiu tratamento quanto à poluição, pois continha

valores inapropriados, como status do CPF – exemplo “CPF Inválido”, em atributo que deveria apenas conter categorização do cliente quanto à sua classe de consumo. Além disso, verificou-se falha de integridade em alguns registros, onde o campo possuía a informação “CORP COMERC” (identificando o usuário como cliente do Mercado Corporativo, ou seja, um CNPJ de uma empresa), em uma base que possui apenas usuários residenciais. Tais valores foram unificados para a moda do atributo (ou seja “MASSA RES P”);

- **Atributos duplicados e redundantes:** tomando como base o campo “Terminal” como identificador único, não foram localizadas amostras duplicadas na base;

- **Valores padrão (*defaults*) e valores desconhecidos:** não verificou-se a ocorrência de valores padrão inseridos na base. Quanto aos valores desconhecidos ou ausentes, a ocorrência maior foi no *data mart* de consumo, onde os registros ausentes (ou *missing*) foram zerados. Exemplo: caso o cliente não tenha efetuado nenhuma ligação para o exterior, o campo que contém o total de minutos para LDI (Longa Distância Internacional) aparece com valor nulo. No pré-processamento, este valor foi zerado, ou seja, o dado correspondente à realidade.

7.4.2 Balanceamento de Classes

O sistema de predição proposto nesta dissertação utiliza apenas duas classes distintas, como descrito anteriormente (classes “C” e “NC”). O modelo de predição de *churn* em telecomunicações caracteriza-se pelo desbalanceamento das classes, ou seja, a quantidade de amostras de usuários da classe “NC” (usuários que não deixarão a base de clientes) é significativamente superior ao número de amostras da classe “C” (potenciais *churners*).

Normalmente, as taxas de *churn* não são divulgadas de forma oficial pelas operadoras de telecomunicações. No entanto, sabe-se que o percentual é significativamente inferior ao mínimo citado pela literatura para que o algoritmo de predição consiga assimilar padrões no treinamento. Segundo Batista (2003), a maioria dos algoritmos tem dificuldades

em criar um modelo que classifique com precisão os exemplos da classe minoritária. Conforme Ferreira (2005), a utilização de freqüências entre 20% e 40% da classe minoritária são desejáveis para melhores resultados.

Esta dissertação utiliza o processo de *oversampling* (descrito no Capítulo 3), reduzindo aleatoriamente a quantidade de registros de amostra pertencentes à classe comum, e injetando na base a quantidade desejada de elementos da classe rara (neste Estudo de Caso, a classe *churner*), ajustando a proporção entre as classes.

O balanceamento de classes é ilustrado na Figura 7.1. O total de registros na base de treinamento foi de 5.045, sendo 2.074 (41%) pertencentes à classe rara (*churner*, ou "C") e 2.971 (59%) pertencentes à classe comum (fidelizado / não-*churner*, ou "NC").

Classe	Qt. de amostras	%
C	2074	41%
NC	2971	59%
	5045	100%

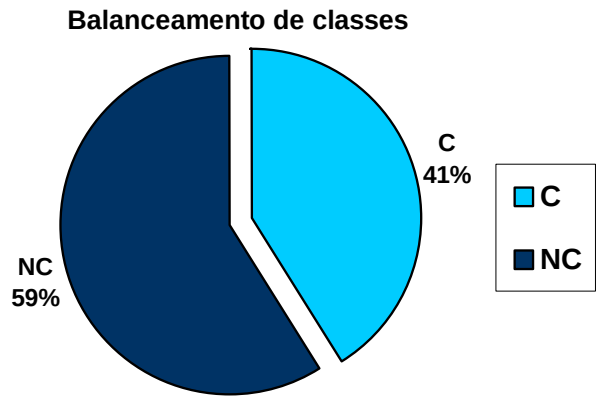

Figura 7.1: balanceamento de classes do modelo

7.4.3 Construção de atributos

Em determinado modelo de mineração de dados, o conjunto de atributos disponível para a tarefa de aprendizado pode ser considerado inadequado ou insuficiente. Atributos fracamente, indiretamente ou condicionalmente relevantes podem ser inadequados quando analisandos isoladamente mas, se utilizados de forma combinada podem gerar novos atributos que podem tornar-se representativos ao modelo. Este conceito, citado por Baranauskas (1998) e por Bloedorn et al. (1998), foi utilizado nesta dissertação com o intuito

de aprimorar o conjunto de dados a ser utilizado pelos algoritmos classificadores. Descreve-se a seguir os atributos construídos com base nos conjuntos de informação originais.

Utilizando-se o atributo "dt_instalacao" do *data mart* (1) – que contém informações de cadastro e relacionamento, construiu-se o atributo "*Tempo_inst*", ou seja, a quantidade de meses em que o cliente é usuário do serviço. Tal informação é mais representativa para os algoritmos classificadores do que uma data simples.

O *data mart* (2) do modelo original – que contém dados de consumo e uso de rede por parte do cliente, foi utilizado para a construção de três novos atributos:

- *Total_LD*: soma do total de minutos de longa distância, incluindo chamadas nacionais e internacionais, cursadas com qualquer CSP (Código de Seleção de Prestadora);
- *Total_LD_14*: soma do total de minutos de longa distância (incluindo chamadas nacionais e internacionais), mas cursadas pelo CSP da Brasil Telecom;
- *Total_LD_Conc*: soma do total de minutos de longa distância (incluindo chamadas nacionais e internacionais), mas cursadas por CSP de outra operadora.

Sobre o *data mart* (3) – que contém dados de Faturamento (ou seja, sobre a conta do cliente), foram construídos os seguintes atributos:

- *Fat_fev*: faturamento líquido do mês de fevereiro de 2006;
- *Fat_mar*: faturamento líquido do mês de março de 2006;
- *Fat_abr*: faturamento líquido do mês de abril de 2006;
- *Evolução_fat*: através da comparação do atributo *Fat_abr* com a média da evolução do faturamento de fevereiro a abril. Caso o faturamento de abril seja maior, o atributo conterá o valor "Crescente". Caso contrário (se o valor for menor que a média do período), conterá o valor "Decrescente".

7.4.4 Seleção de Atributos para o Conjunto de Dados de Treinamento

O processo de seleção de atributos (descrito no Capítulo 3) tem como objetivo principal encontrar o melhor subconjunto a ser utilizado na aplicação dos algoritmos de predição. Considerando-se os três *data marts* ou conjunto de dados originais, são 63 atributos no modelo. Acrescentando-se os 8 novos atributos construídos (conforme descrito na sessão anterior), chega-se a um conjunto de 71 atributos.

Conforme Baranauskas (1998), os algoritmos de mineração de dados não funcionam bem com uma grande quantidade de atributos. Assim, a seleção de atributos pode melhorar o desempenho do modelo. Além disso, um número reduzido de atributos pode tornar o conhecimento induzido pelos algoritmos mais compreensível.

Dentre as três abordagens propostas por Baranauskas (1998) citadas no Capítulo 3, será utilizada a metodologia de Filtro para o processo de seleção de atributos. Esta abordagem consiste em aplicar um método de seleção de variáveis anteriormente à aplicação do algoritmo de classificação. O processo de seleção de atributos neste Estudo de Caso utilizará um algoritmo da ferramenta WEKA para seleção do melhor conjunto de atributos para o modelo. A Figura 7.2 ilustra o resultado da seleção de atributos gerada pela ferramenta WEKA:

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 13 Classe):
        Symmetrical Uncertainty Ranking Filter

Ranked attributes:
 0.03128   2 Tempo_inst
 0.01964   1 Desc_mercado
 0.01594  12 Qt_pulsos_local
 0.01007   5 Fat_abr
 0.00994  10 Qt_chamadas_local
 0.00952   3 Fat_fev
 0.00352   8 Total_LD_14
 0.00327  11 Qt_minutos_local
 0.00303   9 Total_LD_Conc
 0.00189   4 Fat_mar
 0.00182   6 Evolucao_fat
 0.0016    7 Total_LD

Selected attributes: 2,1,12,5,10,3,8,11,9,4,6,7 : 12
```

Figura 7.2: seleção de atributos utilizando a ferramenta WEKA

O algoritmo de seleção de atributos selecionado (ilustrado na Figura 7.2) é denominado *SymmetricalUncertAttributeEval*. Este algoritmo baseia-se no conceito da entropia, e é descrito em Yu *et al* (2003). Este conjunto de atributos forma o *data mart* final de treinamento a ser utilizado pelo modelo, ilustrado na Tabela 7.1.

Rankeamento	Atributo	Descrição
1	Tempo_inst	Tempo de uso do serviço (em meses)
2	Desc_mercado	Classe do usuário
3	Qt_pulsos_local	Quantidade de pulsos locais
4	Fat_abr	Faturamento líquido de abr/2006
5	Qt_chamadas_local	Quantidade de chamadas locais
6	Fat_fev	Faturamento líquido de fev/2006
7	Total_LD_14	Total de minutos em chamadas de Longa Distância com CSP da Brasil Telecom
8	Qt_minutos_local	Quantidade de minutos locais
9	Total_LD_Conc	Total de minutos em chamadas de Longa Distância com CSP de outras operadoras
10	Fat_mar	Faturamento líquido de mar/2006
11	Evolucao_fat	Avalia o faturamento de abr/2006 contra a média dos últimos meses
12	Total_LD	Total de minutos em chamadas de Longa Distância com qualquer CSP

Tabela 7.1: atributos selecionados para o Data Mart de Treinamento

7.5 Transformação dos Dados

Como descrito no Capítulo 3, a etapa de transformação dos dados envolve a aplicação de equações matemáticas ao conteúdo de atributos, com o objetivo de obter-se a forma mais adequada de representatividade da informação para determinado algoritmo.

Três etapas principais de transformação de dados foram utilizadas na composição do *data mart* de treinamento deste Estudo de Caso:

- Discretização de atributos;
- Tratamento de tipos de dados complexos;
- Redução de dados.

Descreve-se a seguir as principais tarefas realizadas em cada uma das etapas.

7.5.1 Discretização de Atributos

Apesar de nenhum dos algoritmos classificadores utilizados neste estudo de caso exigir apenas atributos nominais no conjunto de dados de treinamento, optou-se em realizar o processo de discretização dos atributos quantitativos.

Em testes preliminares com os algoritmos classificadores utilizados no Estudo de Caso, verificou-se um ganho de performance de até 5% na utilização de conjuntos de dados totalmente discretizados (nominais). Assim, optou-se pela aplicação do método no *data mart* final de treinamento.

A ferramenta WEKA possui filtros não-supervisionados de transformação de atributos. Para a conversão de atributos numéricos em atributos nominais, utilizou-se o filtro *Discretize*.

A Figura 7.3 exemplifica o conteúdo de um atributo discretizado pelo filtro. O *label* (ou rótulo nominal) dos dados passou a conter um intervalo descritivo, em substituição ao atributo numérico anterior. O atributo "Tempo_inst" contém a quantidade de meses em que o cliente é usuário do serviço (exibido em um atributo nominal representando uma grandeza numérica).

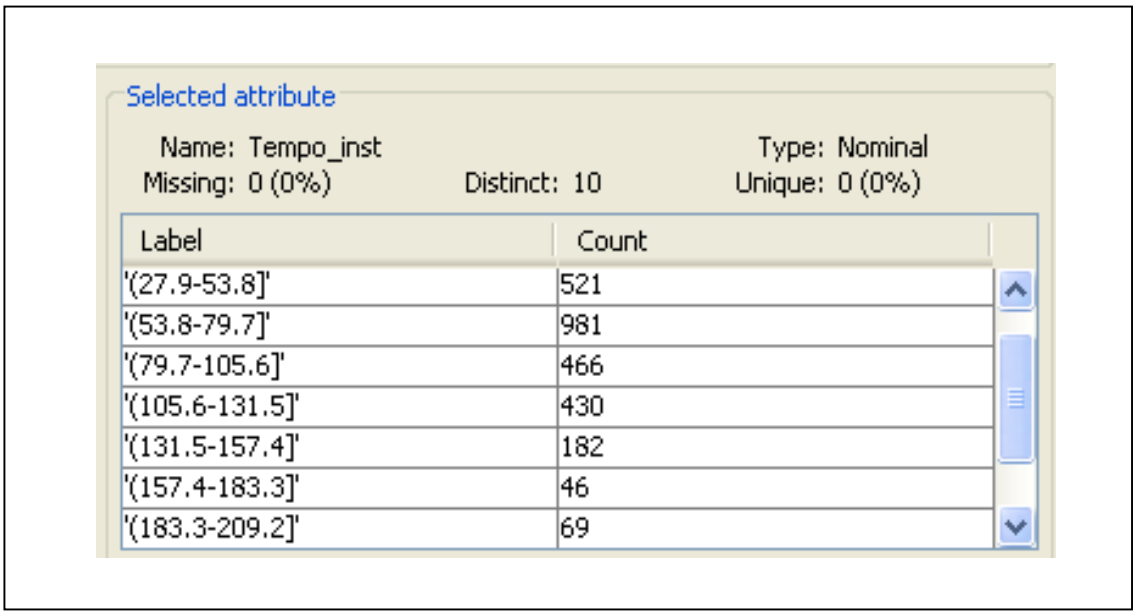

Figura 7.3: exemplo de atributo transformado através do filtro Discretize

7.5.2 Tratamento de Tipos de Dados Complexos

Tipos de dados complexos, como por exemplo data e hora, não são assimilados pela maioria dos algoritmos de extração de padrões. Assim, atributos com esta característica devem ser tratados a fim de otimizarem o desempenho do modelo.

Neste Estudo de Caso, o atributo "Dt_instalacao", encontrado no *data mart* inicial de Cadastro e Relacionamento contém a data em que o terminal telefônico foi ativado. Como o conteúdo do atributo é um tipo de dado complexo (data), optou-se em construir uma nova variável, contando a quantidade de meses em que o cliente possui o serviço (processo citado na etapa de Construção de Atributos).

A relevância do novo atributo resultante do tratamento da variável original pode ser visualizada na Tabela 7.1 – o atributo "Tempo_inst" foi classificado como o mais significativo para a heurística do algoritmo de seleção de variáveis.

O conhecimento especialista também valida esta importância, visto que o atributo representa o tempo em que o cliente utiliza o serviço. Usuários mais antigos tendem a manter o serviço, visando também a manutenção do mesmo número de terminal (que já é de conhecimento de sua rede de contatos). Assim, o atributo representa também, de certa forma, o grau de fidelidade do cliente.

7.5.3 Redução de Dados

O Capítulo 3 desta dissertação cita a importância da utilização da técnica de redução de dados em aplicações de mineração de dados. Este Estudo de Caso baseia-se em um conjunto de dados dos clientes da Brasil Telecom S.A. no estado do RS, o que torna o volume de informações inadequado para um processo de mineração de dados direto.

Assim, este Estudo de Caso utiliza o modelo de redução de dados proposto por Weiss et al. (1998) ilustrado no Capítulo 3, com o objetivo de formar um *data mart* de tamanho (quantidade de registros) adequado em termos de custo e tempo de processamento para a execução dos algoritmos classificadores.

Como método de redução, optou-se por uma junção de duas técnicas propostas por Klösgen (2002): amostragem em *cluster*, seguida de amostragem simples aleatória.

A amostragem em *cluster* pressupõe que os próprios elementos de dados formem subconjuntos. Neste Estudo de Caso, o *cluster* utilizado foi a base de clientes da cidade de Porto Alegre. No entanto, esta base ainda representa um volume de dados inadequado para a aplicação direta dos algoritmos de predição. Utilizou-se então um processo de amostragem simples aleatória sobre o *cluster* inicial.

A formação do *data mart* de treinamento final encerra-se com a aplicação da técnica de balanceamento de classes (ou *oversampling*) já descrito na fase de Compreensão dos Dados e Pré-processamento neste Estudo de Caso.

A Figura 7.4 ilustra a o processo de redução dos dados complementado pelo balanceamento de classes, formando o *data mart* final de treinamento.

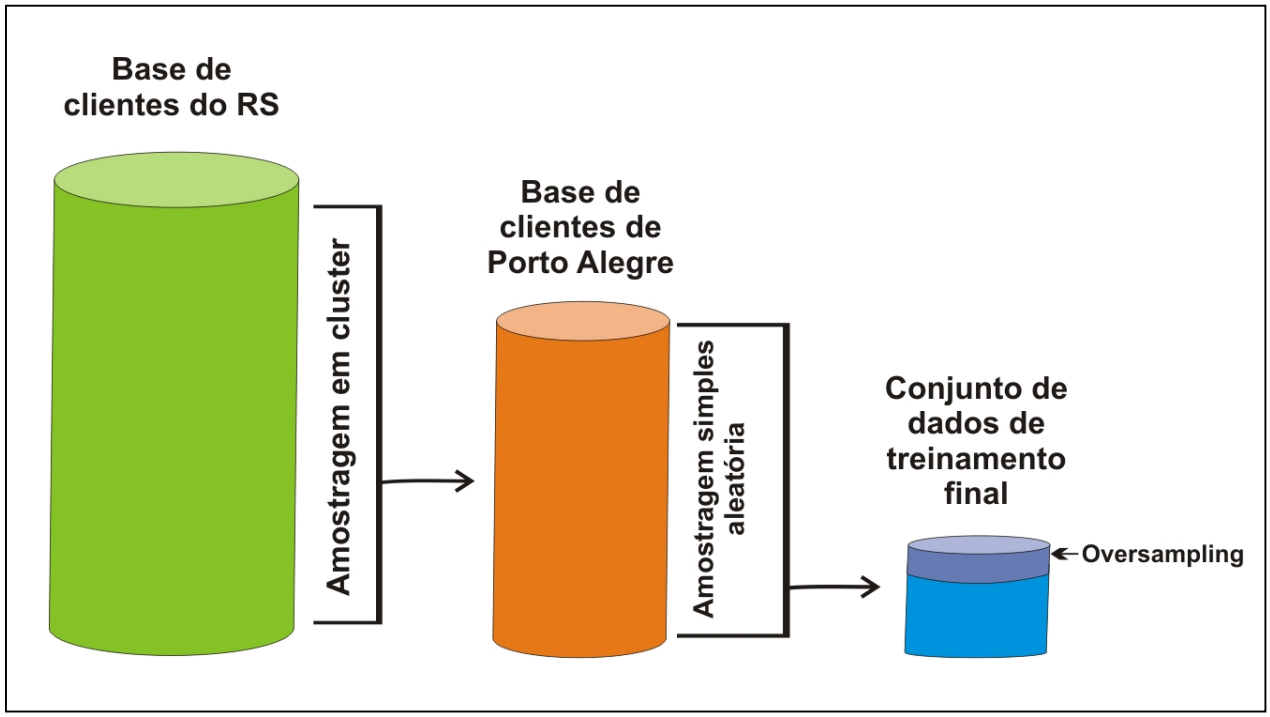

Figura 7.4: aplicação da técnica de redução de dados

A quantidade de registros na base final de treinamento é de 5.045, sendo 2.074 (41%) pertencentes à classe rara (*churner*) e 2.971 (59%) pertencentes à classe comum (fidelizado ou não-*churner*).

7.6 Aplicação dos Algoritmos Classificadores

Para a fase de descoberta de padrões, foram selecionados três algoritmos de técnicas distintas – Redes Neurais RBF, Árvores de Decisão (através do classificador J48) e o Classificador Bayesiano Naive Bayes. O Capítulo 5 descreve as estruturas principais destes algoritmos, demonstrando suas potencialidades e características que os tornam adequados para a aplicação neste Estudo de Caso.

7.6.1 Ferramenta utilizada

A seleção da ferramenta de mineração de dados é parte do processo de construção do sistema proposto nesta dissertação. Como requisitos principais para uma ferramenta de mineração de dados, destacam-se os seguintes (SOARES, 2005):

- Possibilidade de acesso e compreensão de diversas fontes de dados (como arquivos texto, bancos de dados e planilhas);
- Habilidade de processamento de grandes conjuntos de dados;
- Quantidade e variedade de tipos de atributos passíveis de manipulação;
- Capacidade de desempenhar funções diversas referentes ao processo de mineração de dados, como pré-processamento, seleção de atributos e classificação de padrões;
- Custo x benefício;
- Diversidade de algoritmos disponíveis para as tarefas de mineração.

Este Estudo de Caso utiliza o pacote de classes WEKA (*Waikato Environment for Knowledge Analysis*), que compõe um ambiente completo para mineração de dados. Desenvolvida em plataforma Java (o que garante portabilidade e facilidade de adaptação à diversos ambientes e sistemas operacionais), o código da aplicação é de domínio público e a ferramenta possuiu um fórum especializado para pesquisadores, com suporte dos próprios desenvolvedores. O pacote WEKA está disponível em http://www.cs.waikato.ac.nz/ml/weka.

A aplicação acessa bases de dados diretamente através de conexão JDBC. No entanto, recomenda-se a utilização do formato de arquivos próprio da ferramenta, denominado ARFF – um tipo formatado de arquivo texto. Este Estudo de Caso utiliza conjuntos de dados convertidos para ao formato ARFF.

A interface principal da ferramenta é ilustrada na Figura 7.5.

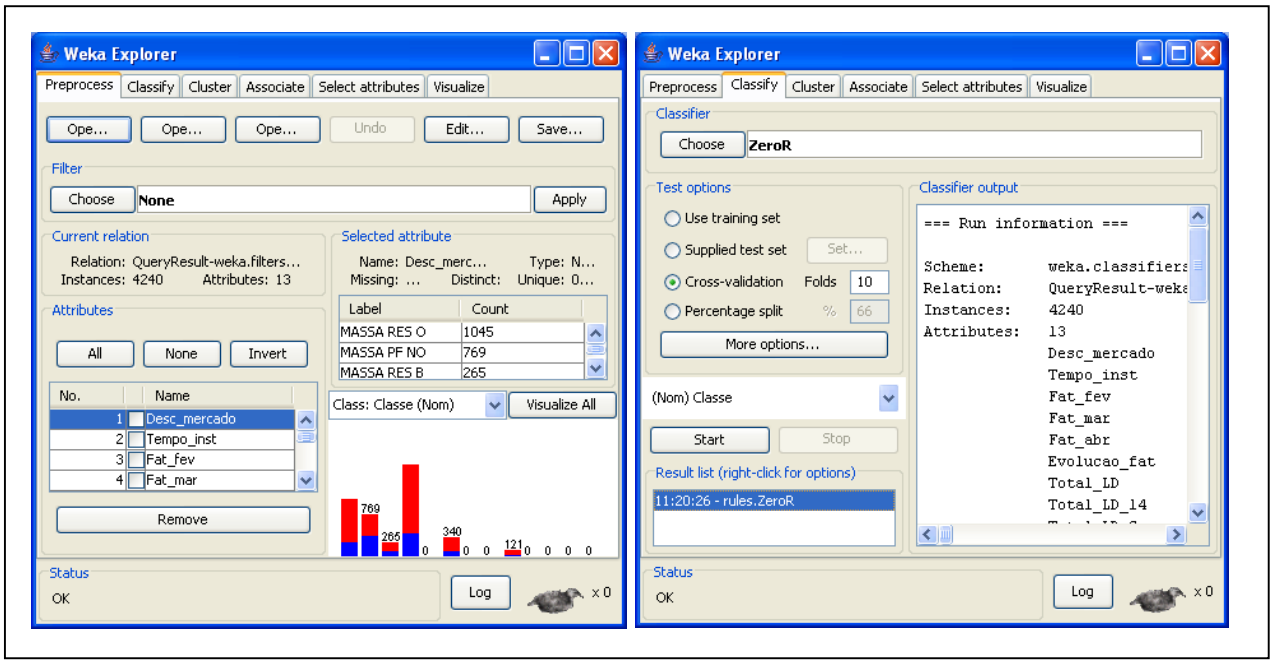

Figura 7.5: telas de pré-processamento e classificação na ferramenta WEKA

7.6.2 Metodologia de Avaliação dos Resultados

Com a finalização das etapas de Pré-Processamento e Transformação dos dados, aplicam-se os algoritmos selecionados ao conjunto de dados de treinamento final. Para avaliar os resultados e taxas de acerto de cada algoritmo, serão utilizadas três metodologias de teste e validação, descritas a seguir (WITTEN et al., 2000):

- **validação utilizando o próprio conjunto de treinamento:** esta metodologia é a mais otimista, visto que utiliza o mesmo conjunto de dados da fase de treinamento para efetuar os testes;

- **validação utilizando divisão percentual do conjunto de treinamento (*Percentage slipt*):** define-se um percentual de utilização do *data mart* para treinamento, sendo que as amostras restantes serão utilizado no teste. Este método apresenta amostras desconhecidas ao modelo (ou seja, diferentes dos dados utilizados na fase de treinamento), produzindo resultados de validação mais eficientes. Os parâmetros utilizados neste Estudo de Caso são os seguintes: 66% para o treinamento e 34% para o teste;

- **validação cruzada:** é um processo estatístico de partição das amostras de dados em subconjuntos onde a análise é efetuada em um conjunto inicial, enquanto outros subconjuntos são retidos para uso subseqüente na validação e testes do modelo. Existem diversas variações para o processo de validação cruzada. Este estudo de caso utilizará o método *k-fold cross-validation* (ou validação cruzada com k-dobras). A técnica consiste em particionar o conjunto de dados original em K subconjuntos (como descrito anteriormente). Um dos K subconjuntos é retido como dados de validação para testar o modelo, enquanto os K-1 subconjuntos são utilizados como dados de treinamento. O processo de validação cruzada é repetido K vezes (são as chamadas "dobras" – *folds*), com cada um dos K subconjuntos utilizados exatamente uma vez como dados de validação. Os K resultados de cada etapa podem ser combinados ou pode-se gerar a média a fim de produzir um único resultado estimado. Este é o modelo de teste mais eficaz entre as metodologias apresentadas. Neste Estudo de Caso, será utilizado um teste de Validação Cruzada em *10* etapas;

A utilização destas três metodologias de teste visa garantir a coerência dos resultados, validando o modelo e o Estudo de Caso desenvolvidos.

7.6.3 Aplicação de Redes Neurais RBF

O primeiro algoritmo testado no processo de Classificação é a Rede Neural RBF (*Radial Basis Function*). A implementação desta metodologia na aplicação WEKA é nomeada

como *RBFNetwork*, e pode ser encontrada na árvore de funções (*functions*) do nó de classificadores da ferramenta.

Esta classe é descrita como uma implementação de uma rede de função Gaussiana radial normalizada. Utiliza o algoritmo de *clusterização k-means*, a fim de prover funções de base e aprendizado com regressão logística (problemas com classes discretas – que é o caso desta dissertação) ou regressão linear (problemas com classes numéricas).

Em cada algoritmo deste Estudo de Caso, foram realizados testes buscando a melhor configuração em cada uma das ferramentas. Os principais parâmetros de configuração da Rede Neural RBF no aplicativo WEKA são os seguintes:

- ***ClusteringSeed*:** "semente" randômica enviada ao algoritmo K-means. Valor utilizado: 1 (padrão);
- ***MaxIts*:** número máximo de iterações executadas pela regressão. Aplicado apenas em problemas com classes discretas (como neste Estudo de Caso). Valor utilizado: 10;
- ***MinStdDev*:** desvio padrão mínimo para os *clusters*. Valor utilizado: 0,1 (padrão);
- ***NumClusters*:** número de *clusters* a serem gerados pelo algoritmo *K-means*. Valor utilizado: 50;
- ***Ridge*:** configura o valor de topo para regressão logística ou linear. Valor utilizado: 1.0E-8 (padrão).

A Figura 7.6 ilustra os resultados da aplicação da Rede Neural RBF:

		Método de Teste					
		Conjunto de treinamento		Divisão % (66%)		Validação Cruzada 10 etapas	
		Qt. Amostras	%	Qt. Amostras	%	Qt. Amostras	%
Classif. Corretas		**3429**	**67,97%**	**1123**	**65,44%**	**3273**	**64,88%**
Classif. Incorretas	✗	1616	32,03%	593	34,56%	1772	35,12%

Figura 7.6: resultados da aplicação de Rede Neural RBF

A Figura 7.7 apresenta as Matrizes de Confusão em cada um dos métodos de teste para a Rede Neural RBF:

Conjunto de treinamento		Divisão % (66%)		Valid. Cruzada 10 etapas	
C	**NC**	**C**	**NC**	**C**	**NC**
1149	925	**336**	346	**1018**	1056
691	**2280**	247	**787**	716	**2255**
3429 acertos		**1123 acertos**		**3273 acertos**	

Figura 7.7: Matrizes de Confusão para a Rede Neural RBF

A Matriz de Confusão exibe a quantidade de acertos e erros de predição para cada uma das classes. Por exemplo, no teste de Validação Cruzada em 10 etapas, a classe "C" (ou *churner*) teve 1.018 instâncias identificadas corretamente e 1.056 identificadas incorretamente. A classe "NC" (ou não-*churner*) teve 2.255 instâncias de acerto, contra 716 de erro. Os resultados da aplicação dos algoritmos de J48 (Árvore de Decisão) e do classificador Naive Bayes serão consolidados e exibidos no mesmo formato.

O relatório detalhado da execução de cada um dos algoritmos classificadores utilizados neste Estudo de Caso (com os resultados completos nos três testes distintos aplicados para cada técnica) pode ser visualizado no Anexo 02 desta dissertação.

7.6.4 Aplicação de Árvores de Decisão

O classificador J48 utilizado neste Estudo de Caso é uma implementação do algoritmo de Árvores de Decisão C4.5 (que, por sua vez, é uma evolução do algoritmo ID3). A classe implementada na aplicação WEKA pode gerar árvores com ou sem poda.

A ferramenta também disponibiliza uma interface gráfica para a visualização e análise da árvore construída (o que torna-se impraticável dependendo da quantidade de nodos e folhas gerados em árvores de grandes dimensões). Os principais parâmetros de configuração do algoritmo são os seguintes:

- ***BinarySplits*:** indica se serão utilizadas divisões binárias em atributos nominais na construção da árvore. Valor utilizado: *false* (padrão).

- ***ConfidenceFactor*:** fator de confiança utilizado na poda da árvore (valores menores geram maior poda). Valor utilizado: 0,4;

- ***MinNumObj*:** mínimo número de instâncias por folha. Valor utilizado: 5;

- ***NumFolds*:** determina a quantidade de dados utilizados para redução de erros de poda. Um conjunto é utilizado para a poda, e os demais para o crescimento da árvore. Valor utilizado: 15;

- ***Unpruned*:** indica se ocorrerá poda na árvore. Valor utilizado: *false* – ou seja, a poda é realizada (padrão).

A Figura 7.8 ilustra os resultados da aplicação do algoritmo J48:

	Método de Teste					
	Conjunto de treinamento		Divisão % (66%)		Validação Cruzada 10 etapas	
	Qt. Amostras	%	Qt. Amostras	%	Qt. Amostras	%
Classif. Corretas	**3423**	**67,85%**	**1137**	**66,26%**	**3296**	**65,33%**
Classif. Incorretas	1622	32,15%	579	33,74%	1749	34,67%

Figura 7.8: resultados da aplicação do algoritmo J48

A Figura 7.9 apresenta as Matrizes de Confusão em cada um dos métodos de teste para o algoritmo J48 (Árvore de Decisão):

Conjunto de treinamento		Divisão % (66%)		Valid. Cruzada 10 etapas	
C	**NC**	**C**	**NC**	**C**	**NC**
1019	1055	**327**	355	**983**	1091
567	**2404**	224	**810**	658	**2313**
3423 acertos		**1137 acertos**		**3296 acertos**	

Figura 7.9: Matrizes de Confusão para o algoritmo J48

A principal vantagem da utilização de Árvores de Decisão sobre outras ferramentas é a sua capacidade de representação dos padrões e conhecimentos descobertos na base de dados. A Figura 7.10 ilustra a árvore gerada neste Estudo de Caso (observação: o Anexo 02 contém a estrutura da árvore em relatório textual, facilitando a visualização e análise).

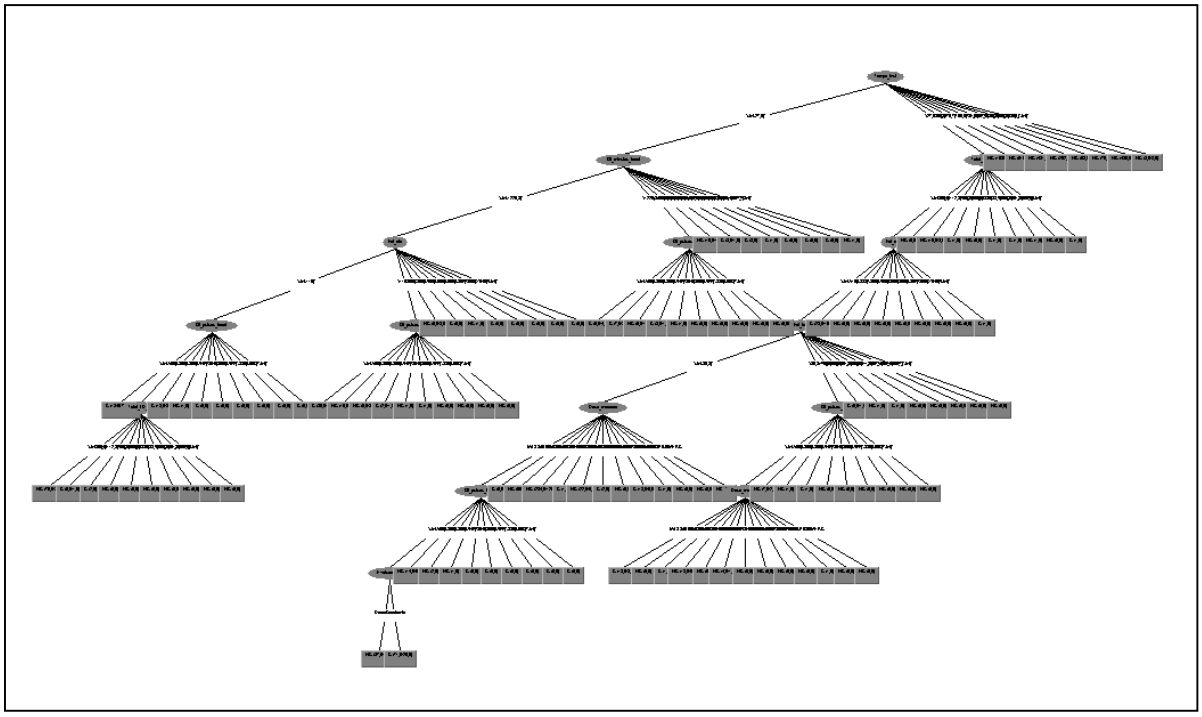

Figura 7.10: estrutura da Árvore de Decisão gerada pelo classificador J48

A árvore gerada pelo classificador J48 é de tamanho 149, com 134 folhas. Através da representação do conhecimento descoberto, o analista ou especialista pode validar o modelo, além de visualizar novos padrões descobertos. Em relação à árvore deste Estudo de Caso, destacam-se as seguintes informações relevantes ao processo de mineração de dados:

- o "topo" selecionado pelo algoritmo foi o atributo "Tempo_inst", que contém a quantidade de meses que o cliente utiliza o terminal residencial fixo. Este mesmo atributo foi selecionado como o mais relevante pelo algoritmo de seleção de atributos (processo descrito na sessão de Pré-processamento), denotando a importância da manutenção de um terminal antigo (principalmente devido a manutenção do mesmo número de telefone) para clientes fidelizados;

- os atributos referentes ao uso de rede (como "Total_LD_14" e "Qt_pulsos_local") também ocuparam posições de direcionamento na árvore, o que denota a importância da avaliação do perfil de consumo dos clientes para a predição do *churn*;

- todas as variáveis destacadas nesta análise do resultado do algoritmo J48 foram atributos construídos na fase de Compreensão dos Dados e Pré-processamento, o que ratifica a necessidade de atenção e concentração de esforços nesta fase da construção de um modelo de mineração de dados.

7.6.5 Aplicação de Classificadores Bayesianos

O classificador Bayesiano utilizado foi o algoritmo Naive Bayes, disponibilizado pela aplicação WEKA. Sua estrutura é mais simples, em comparação com os algoritmos anteriores. Como principais atributos de configuração, destacam-se os seguintes:

- ***UseKernelEstimator***: indica a utilização de um *kernel* de previsão para atributos numéricos, ao invés de uma distribuição normal. Valor utilizado: *false* (padrão);

- ***UseSupervisedDiscretization***: indica a utilização de discretização supervisionada para converter atributos numéricos em nominais em tempo de execução.Valor utilizado: *false* (padrão) – os atributos deste modelo foram discretizados na fase de Transformação dos Dados.

A Figura 7.11 ilustra os resultados da aplicação do classificador Naive Bayes:

	Método de Teste					
	Conjunto de treinamento		Divisão % (66%)		Validação Cruzada 10 etapas	
	Qt. Amostras	%	Qt. Amostras	%	Qt. Amostras	%
Classif. Corretas	**3356**	**66,52%**	**1148**	**66,90%**	**3324**	**65,89%**
Classif. Incorretas	1689	33,48%	568	33,10%	1721	34,11%

Figura 7.11: resultados da aplicação do algoritmo Naive Bayes

A Figura 7.12 apresenta as Matrizes de Confusão em cada um dos métodos de teste para o algoritmo Naive Bayes (Classificador Bayesiano):

Conjunto de treinamento		Divisão % (66%)		Valid. Cruzada 10 etapas	
C	**NC**	**C**	**NC**	**C**	**NC**
996	1078	**317**	365	**1005**	1069
611	**2360**	203	**831**	652	**2319**
3356 acertos		**1148 acertos**		**3324 acertos**	

Figura 7.12: Matrizes de Confusão para o algoritmo Naive Bayes

7.7 Resumo e Avaliação dos Resultados

Este capítulo descreveu a construção do Estudo de Caso sobre o modelo de mineração de dados proposto. Foram detalhadas as etapas que compõe o processo de Descoberta de Conhecimento em Bases de Dados, aplicadas sobre os conjuntos de informações obtidos.

A Figura 7.13 consolida os principais resultados obtidos após a aplicação dos algoritmos de predição:

		Método de Teste					
		Conjunto de treinamento		Divisão % (66%)		Validação Cruzada 10 etapas	
		Qt. Amostras	%	Qt. Amostras	%	Qt. Amostras	%
Rede RBF	**Classif. Corretas**	**3429**	**67,97%**	**1123**	**65,44%**	**3273**	**64,88%**
	Classif. Incorretas	1616	32,03%	593	34,56%	1772	35,12%
Árvore de Decisão (J48)	**Classif. Corretas**	**3423**	**67,85%**	**1137**	**66,26%**	**3296**	**65,33%**
	Classif. Incorretas	1622	32,15%	579	33,74%	1749	34,67%
Classificador Bayesiano (Naive Bayes)	**Classif. Corretas**	**3356**	**66,52%**	**1148**	**66,90%**	**3324**	**65,89%**
	Classif. Incorretas	1689	33,48%	568	33,10%	1721	34,11%

Figura 7.13: resultados obtidos após a aplicação dos algoritmos preditivos

Os percentuais de acerto em cada uma das classes objeto de predição, com base na matriz de confusão são exibidos na Figura 7.14.

		Tipo de teste		
		Conjunto de treinamento	Divisão % (66%)	Valid. Cruzada 10 etapas
Rede RBF	Classe Churner (C)	55,40%	49,27%	49,08%
	Classe Fidelizado (ou Não-Churner - NC)	76,74%	76,11%	75,90%
Árvore de Decisão (J48)	Classe Churner (C)	49,13%	47,95%	47,40%
	Classe Fidelizado (ou Não-Churner - NC)	80,92%	78,34%	77,85%
Classificador Bayesiano (Naive Bayes)	Classe Churner (C)	48,02%	46,48%	48,46%
	Classe Fidelizado (ou Não-Churner - NC)	79,43%	80,37%	78,05%

Figura 7.14: percentuais de acerto da predição em cada classe

Comparativamente, nota-se uma performance muito próxima entre os algoritmos selecionados. Tomando-se como base o processo de Validação Cruzada em 10 etapas, o resultado consolidado de predição foi semelhante nos três algoritmos testados (64,88% na utilização de Redes RBF; 65,33% na utilização de Árvores de Decisão e 65,89% na utilização do algoritmo *Naive Bayes*).

Com base nos indicadores citados por Mozer et al. (2000) como taxas referenciais de predição (aproximadamente 68% na utilização de Redes Neurais e aproximadamente 60% em Árvores de Decisão), verifica-se que os resultados deste Estudo de Caso são aceitáveis e coerentes, validando o modelo desenvolvido.

8 CONCLUSÕES E RECOMENDAÇÕES

Este trabalho apresentou a formação de um modelo completo de mineração de dados (ou de Descoberta de Conhecimento em Bases de Dados) para a classificação de clientes em telecomunicações. Utilizou-se como fonte de estudo bases de informações da Brasil Telcom S.A., visando a predição do evento *churn*, que é o termo utilizado para designar o abandono de clientes em telecomunicações.

Apresentou-se um breve estudo sobre o *churn*, ilustrando suas causas e classificações. O processo completo de mineração de dados foi descrito, citando as principais metodologias e ferramentas. Tendo em vista o problema proposto, foram selecionados três algoritmos classificadores (Redes Neurais RBF, Árvores de Decisão e Classificadores *Bayesianos*) a serem utilizados na fase de predição e classificação de clientes.

O modelo desenvolvido utilizou como embasamentos principais o processo de Descoberta de Conhecimentos em Bases de Dados, proposto por Fayyad et al. (1996) e a metodologia CRISP-DM, definida em CRISP (2000).

A fase de Pré-Processamento e Transformação dos dados concentrou-se em aprimorar o conjunto de dados disponíveis para o modelo. A formação de um dos conjuntos de dados (*data marts*) exigiu a construção de um sistema próprio para a consolidação de informações referentes ao perfil de consumo de serviços de telefonia. As demais variáveis formadas através do conhecimento do negócio tornaram-se fundamentais para o atingimento dos resultados, pois os chamados "atributos construídos" obtiveram destaque de pontuação no algoritmo de seleção de variáveis utilizado no Estudo de Caso.

O Estudo de Caso aplicado sobre o modelo desenvolvido obteve resultados coerentes com os índices propostos na literatura para problemas de predição de *churn*, principalmente em Mozer et al. (2000). O resultado consolidado de classificação correta no modelo ficou entre 64,88% e 65,89%, em performance bastante semelhante nos três algoritmos testados. A técnica de Árvores de Decisão apresenta um diferencial em termos de possibilidade de análise e suporte ao processo de tomada de decisão, visto que o algoritmo gera a estrutura de classificação construída.

Tendo em vista as limitações e dificuldades para a formação dos *data marts*, entende-se que os resultados demonstram viabilidade de aplicação prática do modelo desenvolvido, com potencial de aprimoramento e continuidade. A formação do conjunto de dados final de treinamento considerou a relação entre performance do modelo e custo associado para a formação.

O modelo desenvolvido pode ser estudado e adaptado para outros problemas envolvendo predição e classificação de clientes, principalmente em questões envolvendo o abandono ou migração para outro fornecedor.

Como recomendações e sugestões de trabalhos futuros, sugere-se principalmente:

- aperfeiçoamento do conjunto de dados de treinamento, através da inserção de variáveis complementares, principalmente relacionadas à qualidade do atendimento e experiência do cliente com problemas (como por exemplo solicitações de atendimentos de reparos e contatos com o *call-center*);

- formação do conjunto de treinamento variando a técnica de amostragem utilizada, buscando uma diversidade maior para a descoberta de padrões de comportamento da base de clientes;

- aplicação de outras técnicas de Descoberta de Padrões (complementando o modelo de Classificação), como Análise Associativa ou Análise de Agrupamento (ou de *Cluster*);

- avaliação de variáveis externas para inserção no modelo, como dados estatísticos / sócio-demográficos (neste caso, tomando-se uma amostra com variações na abrangência geográfica) ou informações a respeito do cenário e atuação dos concorrentes;

- variação do conjunto de algoritmos classificadores, buscando possibilidades de otimização dos resultados.

9 REFERÊNCIAS BIBLIOGRÁFICAS

ADRIAANS, Pieter; ZANTINGE, Dolf. **Data Mining**. Harlow, Inglaterra: Addison-Wesley, 1996. 158 p.

ALMEIDA, Fernando C. **Desvendando o uso de redes neurais em problemas de administração de empresas.** RAE, São Paulo, v. 35, n. 1, p. 46-55, Jan./Fev. 1995.

ANATEL. **Plano Geral de Outorgas.** Brasília, 1998. Disponível em: < www.anatel.gov.br/ biblioteca/planos/planogeraloutorgas.pdf >

_____ **Relatório de Acompanhamento das Perspectivas para Ampliação e Modernização do Setor de Telecomunicações - PASTE 2000/2005.** Brasília, 2000. Disponível em: < http://www.anatel.gov.br/Universalizacao/relatorios/paste.htm?Cod=1979>

_____ **Indicadores do Plano Geral de Metas de Universalização.** Disponível em: <http://www.anatel.gov.br/index.asp?link=/Telefonia_Fixa/stfc/indicadores_pgmu/2006/ tabela.htm>. Acesso em: 15/05/2006.

BATISTA, Gustavo E. **Pré-processamento de Dados em Aprendizado de Máquina Supervisionado.** 2003**.** 232 f. Tese (Doutorado em Ciências - Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2003.

BARANAUSKAS, José Augusto; MONARD, Maria Carolina (1998). **Metodologias para a Seleção de Atributos Relevantes.** Disponível em: <http://www.fmrp.usp.br/augusto/ publications/1998-sbia.pdf>. Acesso em 04/06/2006.

BLOEDORN, Eric; MICHALSKI, Ryszard S (1998). **Data-Driven Construtive Induction.** IEEE Intelligent Systems 13 (2), 30–37. Disponível em <http://www-ai.cs.uni-dortmund.de/ dokumente/bloedorn_michalski_98a.pdf>. Acesso em 04/06/2006.

BNDES. As telecomunicações no mundo. **Cadernos de Infra-Estrutura**, v. 14, 2000, 76 p. Disponível em: <http://www.bndes.gov.br/conhecimento/cadernos>. Acesso em: 25/04/2006.

BISHOP, Christopher M. **Neural Networks for Pattern Recognition.** Oxford, Inglaterra: Oxford UK University Press, 1995. 500 p.

BRAGA, Bruno da Rocha; ALMEIDA, José Nogueira; MATTOSO, Fernanda Baião. **DSMiner: Data Mining de Modelos de Detecção de Intrusão.** 2004. Disponível em <http://clusterminer.nacad.ufrj.br/TechReport/RT06.pdf>.

BRAZDIL, Pavel B. **Construção de Modelos de Decisão a partir de Dados.** 1999. Disponível em < http://www.liacc.up.pt/~pbrazdil/Ensino/ML /DecTrees.html>. Acesso em: 20/06/2006.

BRETZKE, Miriam. **Marketing de Relacionamento e competição em tempo real com CRM (Costumer Relationship Management)**. São Paulo: Atlas, 2000. 224 p.

CABENA, Peter et al. **Discovering Data Mining – From Concept to Implementation.** New Jersey, EUA: Prentice Hall PTR, 1997. 195 p.

CARVALHO, Luis Alfredo V. **A Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração.** São Paulo: Érica, 2002. 242 p.

CHEN, Liren; SYCARA, Katia. **WebMate: A Personal Agent for Browsing and Searching.** Pittsburgh, EUA: Carnegie Mellon University, 1997. Disponível em <http://citeseer.ist.psu.edu/cache/papers/cs/9176/http:zSzzSzwww.cs.cmu.eduzSz~softagentszSzpaperszSzaa98-webmate.pdf/chen97webmate.pdf >

CHENG, Jie; GREINER, Russell. **Learning Bayesian Belief Network Classifiers: Algorithms and System.** Edmonton, Canadá: Department of Computing Science, University Alberta, 1999. Disponível em < www.cs.ualberta.ca/~jcheng/Doc/cscsi.pdf >. Acesso em 31/07/2006.

CIOS, Krzysztof; PEDRYCZ, Witold; SWINIARSKI, Roman. **Data Mining Methods for Knowledge Discovery.** Boston, EUA: Kluwer Academic Publishers, 2000. 502 p.

CISTER, Angelo Maia. **Mineração de Dados para a Análise de Atrito em Telefonia Móvel.** 2005**.** 158 f. Tese (Doutorado em Engenharia Civil) - Faculdade de Engenharia Civil, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2005.

COUTINHO, Fernando V. **Data Mining**. Disponível em <http://www.dwbrasil.com.br/html/dmining.html>. Brasília, 2003. Acesso em 17/06/2006.

CRISP. **CRISP-DM - Cross Industry Standard Process for Data Mining, 2000.** Disponível em: <http://www.crisp-dm.org/index.htm>. Acesso em: 19/06/2006.

DIETTERICH, Thomas G.; **Machine Learning Research: Four Current Directions.** Draft of May 23, 1997, Oregon State University. Disponível em < http://scholar.google.com/url?sa=U&q=http://www-cse.uta.edu/~holder/courses/cse6363/lectures/dietterich.ps.gz>. Acesso em: 04/05/2006.

DIN - Departamento de Informática - UEM - Universidade Estadual de Maringá. GSI - Grupo de Sistemas Inteligentes - Mineração de Dados, 1998. Disponível em: <http://www.din.uem.br/ia/mineracao/tecnologia/ferramentas.html> Acesso em: 15/07/2006.

DOUGHERTY, J., KOHAVI, R., SAHAMI, M. **Supervised and Unsupervised Discretization of Continuous Features.** In A. Priedits & S. Russell (Eds.), XII International Conference in Machine Learning. San Francisco, EUA: Morgan Kaufmann Publishers Inc., 2005.

DUDA, Richard. O.; HART Peter. E.; STORK, David. G. **Pattern Classification.** 2. ed. New York, EUA: John Wiley Professio, 2000. 654 p.

FAYYAD, Usama M.; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **Advances in Knowledge Discovery and Data Mining.** Menlo Park, EUA: AAAI Press, 1996. 611 p.

FERNANDES, Marcelo A. C.; NETO, Adrião D. D.; BEZERRA, João B. Aplicação das Redes RBF na Deteção Inteligente de Sinais Digitais. **IV Congresso Brasileiro de Redes Neurais**, São José dos Campos, p. 226-230, 1999.

FERREIRA, Jorge B., **Mineração de Dados na Retenção de Clientes em Telefonia Celular.** 2005**.** 93 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2005.

FIGUEIRA, Rafael. **Mineração de dados e bancos de dados orientados a objetos.** 1998. 96f. Dissertação (Mestrado em Ciência da Computação) - Faculdade de Ciências da Computação, Universidade Federal do Rio de Janeiro, 1998.

FRIEDMAN, Nir; GEIGER, Dan; GOLDSZMIDT, Moises. **Bayesian Network Classifiers.** Boston, EUA: Kluwer Academic, 1997.

GARCIA, Simone C. **O uso de árvores de decisão na descoberta de conhecimento na área da saúde.** 2000. Disponível em < http://www.inf.ufrgs.br/pos/ SemanaAcademica/ Semana2000/SimoneGarcia/>. Acesso em 22/07/2006.

GONÇALVES, Lóren Pinto Ferreira. **Avaliação de Ferramentas de Mineração de Dados como fonte de dados relevantes para a tomada de decisão.** 2001. 104 f. Dissertação (Mestrado em Administração) – Escola de Administração, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

HAN, Jiawei; KAMBER, Micheline. **Data Mining Concepts and Techniques.** San Francisco, EUA: Morgan Kaufmann, 2001. 550 p.

HASSOUN, Mohamad H. **Fundamentals of artificial neural networks.** Cambridge, EUA: MIT Press, 1995. 511 p.

HAYKIN, Simon S. **Neural Networks – A Comprehensive Foundation.** 2. ed. New Jersey, EUA: Prentice Hall, 1998. 842 p.

_____ **Adaptive Filter Theory.** 4. ed. New Jersey, EUA: Prentice Hall, 2001. 936 p.

HECKERMAN, David. **A Tutorial on Learning Bayesian Networks.** Redmond, EUA: Microsoft Corporation, 1995. 58 p. Disponível por FTP anônimo em <ftp://ftp.research.microsoft.com/pub/tr/tr-95-06.pdf>. Acesso em: 10/08/2006.

HOLSHEIMER, Marcel; SIEBES, Arno. **Data Mining: the search for knowledge in databases.** 1994. Disponível por FTP anônimo em ftp.cwi.nl no arquivo /pub/CWIreports/AA/CS-R9406.ps.Z, 1994.

HRUSCHKA, Estevam R.; TEIXEIRA, W. Propagação de Crença em Redes Bayesianas. Brasília: UnB, 1997 (Relatório de Pesquisa CIC/UNB – 02/97).

INGARGIOLA, Giorgio. **Building Classification Models: ID3 and C4.5.** 1996. Disponível em <http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html>. Acesso em: 22/07/2006.

JAGOTA, Arun. **The Radial Basis Function Network.** 1998. Disponível em <http://neuron-ai.tuke.sk/NCS/VOL1/P3_html/node37.html>. Acesso em 18/07/2006.

JOHN, George H. **Enhancements to the Data Mining Process**. Stanford, EUA: Stanford University, 1997. Ph.D. Dissertation.

JOHNSON, Richard. A. **Applied Multivariate Statistical Analysis.** 4. ed. Upper Saddle River, EUA: Prentice Hall, 1998. 816 p.

KANTARDZIC, Mehmed M.; ZURADA, Jozef (Org.). **Next Generation of Data-Mining Applications.** New Jersey, EUA: IEEE Press, 2005. 674 p.

KICKINGER, Flávia C.; PEREIRA, Livia F.; FIGUEIREDO, Renata A. O Modelo de Cinco Forças Aplicado ao Setor de Telefonia Celular no Brasil. **Cadernos Discentes COPPEAD, n. 10**, Rio de Janeiro, 2001. Disponível em <http://www.univercidade.br/HTML/cursos/graduacao/admin/ensino/pdf2003/AnaliseIndustriaKubota_2003.pdf>.Acesso em 03/08/2006.

KLÖSGEN, Willi (Org.). **Handbook of Data Mining and Knowledge Discovery.** New York, EUA: Oxford University Press, 2002. 1026 p.

KOTLER, Philip. **Marketing para o Século XXI: Como criar, conquistar e dominar mercados.** São Paulo: Futura, 1999. 320 p.

KRÖSE, Ben; SMAGT, Patrick. **An Introduction to Neural Networks**. Amsterdam, University of Amsterdam, 1996. Disponível em: <http://www.avaye.com/files/articles/nnintro/nn_intro.pdf>. Acesso em: 10/07/2006.

LEMOS, Eliane P. **Análise de Crédito Bancário com o uso de *Data Mining*: Redes Neurais e Árvores de Decisão.** 2003. 147 f. Dissertação (Mestrado em Ciências) – Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Área de Concentração em Programação Matemática, Universidade Federal do Paraná, Curitiba, 2003.

MATTISON, Rob. **The Telco Churn Management Handbook.** McHenry County, EUA: Lulu Press Inc., 2001. 392 p.

MEDEIROS, José S. **Bancos de Dados Geográficos e Redes Neurais Artificiais: Tecnologias de Apoio à Gestão de Território.** 1999. 236 f. Tese (Doutorado em Geografia Física) – Departamento de Geografia da Faculdade de Ciências Humanas, Universidade de São Paulo, São Paulo, 1999.

MELLO, Luis Cesar. **Um Assistente de Feedback para o Serviço de Filtragem do Software Direto.** 2002. 115 f. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2002.

MICHIE, Donald; SPIEGELHALTER, David J.; TAYLOR, Charles C. (Org.). **Machine Learning, Neural, and Statistical Classification.** New York, EUA: Ellis Horwood, 1994. 289 pp.

MOZER, Michael C.; WOLNIEWICZ, Richard; GRIMES, David B.; JOHNSON, Eric; KAUSHANSKY, Howard. Predicting Subscriber Dissatisfaction and Improving Retention in the Wireless Telecommunications Industry. **IEEE Transactions on Neural Networks - Special issue on Data Mining and Knowledge Representation**. Boulder, 2000. Disponível em < http://www.cs.colorado.edu/~mozer/papers/reprints/churn.pdf >.

NOGUEIRA, Carlos Fernando. **Metodologia de valorização de clientes, utilizando mineração de dados.** 2004. 272 f. Tese (Doutorado em Ciências em Engenharia Civil) – Faculdade de Engenharia, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2004.

OLIVEIRA, Djalma de Pinho Rebouças. **Sistemas de informações gerenciais: estratégicas, táticas operacionais.** 4ª ed. São Paulo: Atlas, 1997. 288 p.

PEARL, Judea. **Probabilistic Reasoning in Intelligent Systems.** San Mateo, EUA: Morgan Kaufman, 1988. 552 p.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL. Biblioteca Central Ir. José Otão. **Orientações para apresentação de citações em documentos segundo NBR 10520.** Disponível em: http://www.pucrs.br/biblioteca/citacoes.htm>. Acesso em: 25/04/2006.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL. Biblioteca Central Ir. José Otão. **Modelo de Referências Elaborado pela Biblioteca Central Irmão José Otão.** Disponível em: <http://www.pucrs.br/biblioteca/modelo.htm>. Acesso em: 25/04/2006.

QUINLAN, J. R. **Induction of decision trees. Machine Learning**. Boston, EUA: Kluwer Academic Publishers, 1986.

SIAS, César C. **O Desempenho dos Atributos de Qualidade em Serviços de Conectividade de Redes: o Caso de uma Operadora de Telecomunicações.** 2005. 124 f. Dissertação (Mestrado Profissionalizante em Engenharia de Produção) – Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.

SILVA, Rafael E. Redes Neurais Artificiais MLP´s (Multi Layer Perceptron) versus RBF´s (Radial Basis Function) em uma aplicação. Porto Alegre, 2003. Disponível em: <http://www.inf.ufrgs.br/procpar/disc/cmp135/trabs/rafael/T2/MLPxRBFrevisado.htm>. Acesso em: 19/07/2006.

SOARES, Silviane L. **Aplicação de técnicas de Mineração de Dados na Gestão de Sistemas de Energia Elétrica.** 2005. 106 f. Dissertação (Mestrado em Engenharia Elétrica) – Programa de Pós-Graduação em Engenharia Elétrica, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2005.

TAFNER, M. A. Redes Neurais Artificiais: Aprendizado e Plasticidade. **Revista Cérebro & Mente**, mar./mai. 1998. Disponível em: <http://www.cerebromente.org.br/n05/ tecnologia/plasticidade2.html >. Acesso em: 25/07/2006.

WEISS, Sholom M.; INDURKHYA, Nitin. **Predictive Data Mining – a practical guide.** San Francisco, EUA: Morgan Kaufmann Publishers Inc., 1998. 230 p.

WITTEN, Ian H.; FRANK Eib. **Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations.** San Francisco, EUA: Morgan Kaufmann Publishers Inc., 2000. 376 p.

YU, Lei; LIU, Huan. **Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution.** In: Proceedings of the Twentieth International Conference on Machine Learning, 856-863, 2003. Disponível em: <http://www.hpl.hp.com/conferences /icml2003/papers/144.pdf>. Acesso em: 30/08/2006.

ANEXO 01

FUNÇÃO (PROCEDURE) PRINCIPAL DO SISTEMA DE TRATAMENTO DE BILHETES DE CONSUMO

```
libname crt 'e:\bd\consumo\0606';
libname crt2 'f:\bd\consumo\0606';
libname crt1 'e:\bd\consumo';
libname crt3 'f:\bd\consumo\0606';

*importa o DDR base SGPI*;
*PROC IMPORT OUT= crt1.DDRnovo
            DATAFILE= "g:\bd\consumo\grandesclientes\tabelas\DDR_SGPI24112005.xls"
            DBMS=EXCEL2000 REPLACE;
     RANGE="Plan1$";
     GETNAMES=YES;
RUN;*

*agrupa os dias pendentes e cria um conj intermediario;

data crt.con0606;
       set crt.con0606 crt.c01 crt.c02 crt.c03 crt.c04 crt.c05 crt.c06 crt.c07
       crt.c08 /*crt.c09 crt.c10 crt.c11 crt.c12 crt.c13 crt.c14 crt.c15
       crt.c16 crt.c17 crt.c18 crt.c19 crt.c20
       crt.c21 crt.c22 crt.c23 crt.c24 crt.c25 crt.c26 crt.c27
       crt.c28 crt.c29 crt.c30 crt.c31 crt.c32 crt.c33 crt.c34 crt.c35 crt.c36
       crt.c37 crt.c38 crt.c39 crt.c40 crt.c41 crt.c42 crt.c43 crt.c44 /*crt.c45
       crt.c46 crt.c47 crt.c48 crt.c49 crt.c50 crt.c51 crt.c52 crt.c53 crt.c54
       crt.c55 crt.c56 crt.c57 crt.c58 crt.c59 crt.c60 crt.c61 crt.c62
       crt.c63 crt.c64 crt.c65 crt.c66 crt.c67 crt.c68 crt.c69 crt.c70
       crt.c71 crt.c72 crt.c73 crt.c74 crt.c75 crt.c76 crt.c77 crt.c78 crt.c79
       crt.c80 crt.c81 crt.c82 crt.c83 crt.c84 crt.c85 crt.c86 /*crt.c87 crt.c88
       crt.c89 crt.c90 crt.c91 crt.c92 crt.c93 crt.c94 crt.c95 crt.c96 crt.c97
       crt.c98 crt.c99 crt.c100 crt.c101 crt.c102 crt.c103 crt.c104 crt.c105
       crt.c106 crt.c107 crt.c108 crt.c109 crt.c110 crt.c111 crt.c112 crt.c113
       crt.c114 crt.c115 crt.c116 crt.c117 crt.c118 crt.c119 crt.c120 crt.c121
       crt.c122 crt.c123 crt.c124 crt.c125 crt.c126 crt.c127 crt.c128 crt.c129*/;
       where substr(dt_chama,3,2) = '06';
       minutos=duracao/60;
run;

proc freq data=crt.con0606;
       table dt_chama;
run;
*/*roda depois de conferir volume diario;
data crt.dia0106 crt.con0606;
       set crt.con0606;
       if dt_chama eq '01062006' then output crt.dia0106;
       if dt_chama ne '01062006' then output crt.con0606;
run;

*/****** retira chamadas duplicadas ***************;
proc sort data=crt.dia0106 nodupkey;
       by assin_a assin_b dt_chama hora_ini min_ini seg_ini;
run;

****** retira chamadas sobrepostas **********;
data crt.dia0106;
       set crt.dia0106;
       length ddd $2. pref $4. mcdu $4.;
       ddd = substr(assin_a,1,2);
       pref = substr(assin_a,3,4);
       mcdu = substr(assin_a,7,4);
       telef = ddd || pref;
run;

proc sort data=crt.dia0106;
       by telef;
run;

proc sort data=crt1.DDRnovo;
       by telef;
run;
```

```
data ddr_1a;
       merge crt.dia0106 crt1.DDRnovo;
       by telef;
       if (ini <= mcdu and mcdu<= fim) then output;
run;

data nova1;
       set ddr_1a;
       v= 'tem';
run;

proc sort data=nova1;
       by assin_a;
run;

proc sort data=crt.dia0106;
       by assin_a;
run;

data crt.dia0106(drop=cliente ini fim cabeceira fabricante_cabeceira
                 central_de_comutacao area_numeracao nr_prefixo
                 faixa_inicial faixa_final ddd pref mcdu telef);
       merge nova1 crt.dia0106;
       by assin_a;
run;

data crt.dia0106;
    set crt.dia0106;
    nova=compress(hora_ini||':'||min_ini||':'||seg_ini);
    inicio=input(nova,time8.);
    fim=inicio + duracao;
    format inicio time8. fim time8.;
    drop nova;
run;

proc sort data=crt.dia0106;
       by assin_a assin_b dt_chama inicio fim;
run;

data crt.dia0106;
   set crt.dia0106;
   atual = duracao;
   anter = lag(duracao);
   ass_a = lag(assin_a);
   ass_b = lag(assin_b);
   data = lag(dt_chama);
   fim_ant = lag(fim);
   if v ne 'tem' then
          do;
      if assin_a = ass_a and assin_b = ass_b and dt_chama = data and
               inicio <= fim_ant
        then do;
                    resto = abs(atual - anter);
              if resto < 19 then delete;
         end;
      end;
       drop resto ass_a ass_b fim_ant data atual anter fim inicio v;
run;

********** cria tipocha1 e tipocha2 ************;
data crt.dia0106;
       set crt.dia0106;
    length tipocha1 $1. tipocha2 $1. tipocha3 $1.;
       tipocha1 = tipo_cha;
       tipocha2 = tipo_cha;
       if (substr(assin_b,3,3) eq '350') or substr(assin_b,1,7) eq '5332779'
          then tipo_cha = 'A';
```

```
    if ((substr(assin_b,1,5) eq '53321') or (substr(assin_b,1,5) eq '53322') or
        (substr(assin_b,1,5) eq '53327') or (substr(assin_b,1,5) eq '53328')) then
        do;
            tipocha1 = 'C';
            tipo_cha = 'T';
        end;

       if (tipo_cha = '3' and substr(assin_b,1,1) not in ('4','5','6') and
        substr (assin_b,3,1) eq '9') then tipocha2 = '4';

       tipocha3 = tipocha2;
       if tipocha2 = 'T' and uf_dest = 'RS' then tipocha3 = 'A';
run;

*********importa arquivo de areas locais*******;
PROC IMPORT OUT= crt1.areas_locais_set04
            DATAFILE=
"g:\Bd\Consumo\GrandesClientes\Tabelas\areas_locais_set04.xls"
            DBMS=EXCEL2000 REPLACE;
     RANGE="Plan1$";
     GETNAMES=YES;
RUN;

*/***** insere a sigla da área local de origem e de destino ********;
data crt1.areas_locais_set04;
       set crt1.areas_locais_set04;
       length sigla_loc2 $4. sigla_area_loc2 $4.;
       if substr(sigla_loc,4,1) eq '' or substr(sigla_area_loc,4,1) eq '' then
       do;
              sigla_loc2 = compress(sigla_loc ||'-');
              sigla_area_loc2 = compress(sigla_area_loc || '-');
       end;
       else
              do;
                     sigla_loc2 = sigla_loc;
                     sigla_area_loc2 = sigla_area_loc;
              end;
run;
proc sql;
       create table crt.dia0106a as
       select dia0106.*, areas_locais_set04.sigla_area_loc as sigla_area_loc_ori,
       areas_locais_set04.grupo1 as grupo1_ori,
       areas_locais_set04.grupo2 as grupo2_ori,
       areas_locais_set04.grupo3 as grupo3_ori
       from crt.dia0106 left join crt1.areas_locais_set04
       on (dia0106.origem = areas_locais_set04.sigla_loc2);
quit;
proc sql;
       create table crt.dia0106 as
       select dia0106a.*, areas_locais_set04.sigla_area_loc as sigla_area_loc_dest,
       areas_locais_set04.grupo1 as grupo1_dest,
       areas_locais_set04.grupo2 as grupo2_dest,
       areas_locais_set04.grupo3 as grupo3_dest
       from crt.dia0106a left join crt1.areas_locais_set04
       on (dia0106a.destino = areas_locais_set04.sigla_loc2);
quit;

*******Compara Grupo**************************;
data crt.dia0106;
       set crt.dia0106;
       length trafego $1;
       if grupo1_ori ne '' then do;
              if (grupo1_ori=grupo1_dest) or (grupo1_ori=grupo2_dest)
              or (grupo1_ori=grupo3_dest) then trafego='L';
       end;
       if grupo2_ori ne '' then do;
              if (grupo2_ori=grupo1_dest) or (grupo2_ori=grupo2_dest)
```

```
              or (grupo2_ori=grupo3_dest) then trafego='L';
       end;
       if grupo3_ori ne '' then do;
              if (grupo3_ori=grupo1_dest)      or (grupo3_ori=grupo2_dest)
              or (grupo3_ori=grupo3_dest) then trafego='L';
       end;
run;


***** cria campo tarifa e insere “P” p/ pulso unico e “M” para multimedido ***;
data crt.dia0106;
       set crt.dia0106;
       length tarifa $1;
       tarifa = gru_hora;
       if ((sigla_area_loc_ori = sigla_area_loc_dest) or (trafego eq 'L')) and
tipocha2 eq 'A'
           and cha800 ne '8' then
              do;
                     if tipo_dia in ('D' 'F') then tarifa = 'P';
                     else if tipo_dia in ('U' 'S') and hora_ini in ('00' '01' '02'
                                   '03' '04' '05') then tarifa = 'P';
                     else if tipo_dia eq 'S' and hora_ini >= '14' then tarifa = 'P';
                     else tarifa = 'M';
              end;
run;
```

ANEXO 02

RELATÓRIOS DE DETALHAMENTO DA EXECUÇÃO DOS ALGORITMOS CLASSIFICADORES

1) Resultados do classificador Rede Neural RBF

1.a) Validação utilizando o próprio conjunto de treinamento

```
Time taken to build model: 5.55 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        3429               67.9683 %
Incorrectly Classified Instances      1616               32.0317 %
Kappa statistic                          0.327
Mean absolute error                      0.4054
Root mean squared error                  0.4546
Relative absolute error                 83.7331 %
Root relative squared error             92.3994 %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.554     0.233      0.624     0.554     0.587    C
  0.767     0.446      0.711     0.767     0.738    NC

=== Confusion Matrix ===

    a    b   <-- classified as
 1149  925 |    a = C
  691 2280 |    b = NC
```

Rede Neural RBF

1.b) Validação utilizando Divisão % (Parâmetro utilizado: 66% de treinamento)

```
Time taken to build model: 5.45 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        1123               65.4429 %
Incorrectly Classified Instances       593               34.5571 %
Kappa statistic                          0.2602
Mean absolute error                      0.4217
Root mean squared error                  0.4787
Relative absolute error                 87.2736 %
Root relative squared error             97.723  %
Total Number of Instances             1716

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.493     0.239      0.576     0.493     0.531    C
  0.761     0.507      0.695     0.761     0.726    NC

=== Confusion Matrix ===

   a   b   <-- classified as
 336 346 |   a = C
 247 787 |   b = NC
```

1) Resultados do classificador Rede Neural RBF

1.c) Validação cruzada em 10 etapas:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3273               64.8761 %
Incorrectly Classified Instances      1772               35.1239 %
Kappa statistic                          0.2562
Mean absolute error                      0.4183
Root mean squared error                  0.4788
Relative absolute error                 86.3939 %
Root relative squared error             97.3173 %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.491     0.241      0.587     0.491     0.535    C
  0.759     0.509      0.681     0.759     0.718    NC

=== Confusion Matrix ===

    a    b   <-- classified as
 1018 1056 |    a = C
  716 2255 |    b = NC
```

2) Resultados do classificador J48 (Árvores de Decisão)

2. a) Validação utilizando o próprio conjunto de treinamento

```
Number of Leaves  :     134

Size of the tree :      149

Time taken to build model: 0.13 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        3423               67.8494 %
Incorrectly Classified Instances      1622               32.1506 %
Kappa statistic                          0.3115
Mean absolute error                      0.4274
Root mean squared error                  0.4623
Relative absolute error                 88.2618 %
Root relative squared error             93.9484 %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.491     0.191      0.642     0.491     0.557    C
  0.809     0.509      0.695     0.809     0.748    NC

=== Confusion Matrix ===

    a    b   <-- classified as
 1019 1055 |    a = C
  567 2404 |    b = NC
```

J48 (Árvores de Decisão)

2.b) Validação utilizando Divisão % (Parâmetro utilizado: 66% de treinamento)

```
Number of Leaves  :     134

Size of the tree :      149

Time taken to build model: 0.11 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        1137               66.2587 %
Incorrectly Classified Instances       579               33.7413 %
Kappa statistic                          0.2717
Mean absolute error                      0.4329
Root mean squared error                  0.4713
Relative absolute error                 89.5924 %
Root relative squared error             96.2189 %
Total Number of Instances             1716

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.479     0.217      0.593     0.479     0.53     C
  0.783     0.521      0.695     0.783     0.737    NC

=== Confusion Matrix ===

   a   b   <-- classified as
 327 355 |   a = C
 224 810 |   b = NC
```

J48 (Árvores de Decisão)

2.c) Validação cruzada em 10 etapas:

```
Number of Leaves  :     134

Size of the tree :      149

Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3296               65.332  %
Incorrectly Classified Instances      1749               34.668  %
Kappa statistic                          0.2607
Mean absolute error                      0.4362
Root mean squared error                  0.4721
Relative absolute error                 90.0884 %
Root relative squared error             95.9466 %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.474     0.221      0.599     0.474     0.529    C
  0.779     0.526      0.679     0.779     0.726    NC

=== Confusion Matrix ===

   a    b   <-- classified as
 983 1091 |    a = C
 658 2313 |    b = NC
```

J48 (Árvores de Decisão)

2.d) Representação da árvore gerada pelo modelo:

```
=== Model information ===
Filename:     Modelo J48 10-fold.model
Scheme:       trees.J48 -C 0.4 -M 5
Relation:      QueryResult-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-
Rfirst-last-weka.filters.unsupervised.attribute.Remove-R1-4,7-8-
weka.filters.unsupervised.attribute.Remove-R3-4
Attributes:   13
              Desc_mercado
              Tempo_inst
              Fat_fev
              Fat_mar
              Fat_abr
              Evolucao_fat
              Total_LD
              Total_LD_14
              Total_LD_Conc
              Qt_chamadas_local
              Qt_minutos_local
              Qt_pulsos_local
              Classe

=== Classifier model ===

J48 pruned tree
------------------

Tempo_inst = '(-inf-27.9]'
|   Qt_minutos_local = '(-inf-1225.3]'
|   |   Fat_abr = '(-inf-116]'
|   |   |   Qt_pulsos_local = '(-inf-103]': C (1345.0/484.0)
|   |   |   Qt_pulsos_local = '(103-206]'
|   |   |   |   Total_LD_14 = '(-inf-56.4]': NC (76.0/35.0)
|   |   |   |   Total_LD_14 = '(56.4-112.8]': C (5.0/1.0)
|   |   |   |   Total_LD_14 = '(112.8-169.2]': C (2.0)
|   |   |   |   Total_LD_14 = '(169.2-225.6]': NC (0.0)
|   |   |   |   Total_LD_14 = '(225.6-282]': NC (0.0)
|   |   |   |   Total_LD_14 = '(282-338.4]': NC (0.0)
|   |   |   |   Total_LD_14 = '(338.4-394.8]': NC (0.0)
|   |   |   |   Total_LD_14 = '(394.8-451.2]': NC (0.0)
|   |   |   |   Total_LD_14 = '(451.2-507.6]': NC (0.0)
|   |   |   |   Total_LD_14 = '(507.6-inf)': NC (0.0)
|   |   |   Qt_pulsos_local = '(206-309]': C (13.0/3.0)
|   |   |   Qt_pulsos_local = '(309-412]': NC (1.0)
|   |   |   Qt_pulsos_local = '(412-515]': C (0.0)
|   |   |   Qt_pulsos_local = '(515-618]': C (0.0)
|   |   |   Qt_pulsos_local = '(618-721]': C (0.0)
|   |   |   Qt_pulsos_local = '(721-824]': C (0.0)
|   |   |   Qt_pulsos_local = '(824-927]': C (0.0)
|   |   |   Qt_pulsos_local = '(927-inf)': C (0.0)
|   |   Fat_abr = '(116-232]'
|   |   |   Qt_pulsos_local = '(-inf-103]': C (35.0/15.0)
|   |   |   Qt_pulsos_local = '(103-206]': NC (15.0/3.0)
|   |   |   Qt_pulsos_local = '(206-309]': NC (9.0/3.0)
|   |   |   Qt_pulsos_local = '(309-412]': C (2.0/1.0)
|   |   |   Qt_pulsos_local = '(412-515]': NC (1.0)
|   |   |   Qt_pulsos_local = '(515-618]': C (1.0)
|   |   |   Qt_pulsos_local = '(618-721]': NC (0.0)
|   |   |   Qt_pulsos_local = '(721-824]': NC (0.0)
|   |   |   Qt_pulsos_local = '(824-927]': NC (0.0)
|   |   |   Qt_pulsos_local = '(927-inf)': NC (0.0)
|   |   Fat_abr = '(232-348]': NC (9.0/3.0)
```

```
|   |   Fat_abr = '(348-464]': C (0.0)
|   |   Fat_abr = '(464-580]': NC (1.0)
|   |   Fat_abr = '(580-696]': C (0.0)
|   |   Fat_abr = '(696-812]': C (0.0)
|   |   Fat_abr = '(812-928]': C (0.0)
|   |   Fat_abr = '(928-1044]': C (0.0)
|   |   Fat_abr = '(1044-inf)': C (0.0)
|   Qt_minutos_local = '(1225.3-2450.6]'
|   |   Qt_pulsos_local = '(-inf-103]': C (9.0/4.0)
|   |   Qt_pulsos_local = '(103-206]': C (7.0/3.0)
|   |   Qt_pulsos_local = '(206-309]': NC (5.0/1.0)
|   |   Qt_pulsos_local = '(309-412]': C (3.0/1.0)
|   |   Qt_pulsos_local = '(412-515]': NC (1.0)
|   |   Qt_pulsos_local = '(515-618]': NC (0.0)
|   |   Qt_pulsos_local = '(618-721]': NC (0.0)
|   |   Qt_pulsos_local = '(721-824]': NC (0.0)
|   |   Qt_pulsos_local = '(824-927]': NC (0.0)
|   |   Qt_pulsos_local = '(927-inf)': NC (0.0)
|   Qt_minutos_local = '(2450.6-3675.9]': NC (10.0/4.0)
|   Qt_minutos_local = '(3675.9-4901.2]': C (3.0/1.0)
|   Qt_minutos_local = '(4901.2-6126.5]': C (3.0)
|   Qt_minutos_local = '(6126.5-7351.8]': C (1.0)
|   Qt_minutos_local = '(7351.8-8577.1]': C (0.0)
|   Qt_minutos_local = '(8577.1-9802.4]': C (0.0)
|   Qt_minutos_local = '(9802.4-11027.7]': C (0.0)
|   Qt_minutos_local = '(11027.7-inf)': NC (1.0)
Tempo_inst = '(27.9-53.8]'
|   Total_LD_14 = '(-inf-56.4]'
|   |   Fat_abr = '(-inf-116]'
|   |   |   Fat_fev = '(-inf-85.3]'
|   |   |   |   Desc_mercado = MASSA RES O
|   |   |   |   |   Qt_pulsos_local = '(-inf-103]'
|   |   |   |   |   |   Evolucao_fat = Decrescente: NC (37.0/16.0)
|   |   |   |   |   |   Evolucao_fat = Crescente: C (71.0/26.0)
|   |   |   |   |   Qt_pulsos_local = '(103-206]': NC (14.0/5.0)
|   |   |   |   |   Qt_pulsos_local = '(206-309]': NC (2.0)
|   |   |   |   |   Qt_pulsos_local = '(309-412]': NC (1.0)
|   |   |   |   |   Qt_pulsos_local = '(412-515]': C (0.0)
|   |   |   |   |   Qt_pulsos_local = '(515-618]': C (0.0)
|   |   |   |   |   Qt_pulsos_local = '(618-721]': C (0.0)
|   |   |   |   |   Qt_pulsos_local = '(721-824]': C (0.0)
|   |   |   |   |   Qt_pulsos_local = '(824-927]': C (0.0)
|   |   |   |   |   Qt_pulsos_local = '(927-inf)': C (0.0)
|   |   |   |   Desc_mercado = MASSA PF NO: C (9.0/3.0)
|   |   |   |   Desc_mercado = MASSA RES B: NC (65.0/20.0)
|   |   |   |   Desc_mercado = MASSA RES P: NC (284.0/129.0)
|   |   |   |   Desc_mercado = PF NOVO INA: C (1.0)
|   |   |   |   Desc_mercado = MASSA SOHO: NC (22.0/9.0)
|   |   |   |   Desc_mercado = EMPR SOHO P: C (2.0)
|   |   |   |   Desc_mercado = CNPJ INVALI: NC (0.0)
|   |   |   |   Desc_mercado = MASSA RES D: C (13.0/6.0)
|   |   |   |   Desc_mercado = EMPR MICRO: C (1.0)
|   |   |   |   Desc_mercado = INATIVO: NC (0.0)
|   |   |   |   Desc_mercado = CNPJ/CPF IN: NC (0.0)
|   |   |   |   Desc_mercado = CORP COMERC: NC (0.0)
|   |   |   Fat_fev = '(85.3-170.6]'
|   |   |   |   Qt_pulsos_local = '(-inf-103]'
|   |   |   |   |   Desc_mercado = MASSA RES O: C (18.0/8.0)
|   |   |   |   |   Desc_mercado = MASSA PF NO: NC (0.0)
|   |   |   |   |   Desc_mercado = MASSA RES B: C (1.0)
|   |   |   |   |   Desc_mercado = MASSA RES P: NC (18.0/6.0)
|   |   |   |   |   Desc_mercado = PF NOVO INA: NC (0.0)
|   |   |   |   |   Desc_mercado = MASSA SOHO: NC (4.0/1.0)
|   |   |   |   |   Desc_mercado = EMPR SOHO P: NC (0.0)
|   |   |   |   |   Desc_mercado = CNPJ INVALI: NC (0.0)
|   |   |   |   |   Desc_mercado = MASSA RES D: NC (0.0)
|   |   |   |   |   Desc_mercado = EMPR MICRO: NC (0.0)
|   |   |   |   |   Desc_mercado = INATIVO: C (1.0)
```

```
|   |   |   |   |   Desc_mercado = CNPJ/CPF IN: NC (0.0)
|   |   |   |   |   Desc_mercado = CORP COMERC: NC (0.0)
|   |   |   |   Qt_pulsos_local = '(103-206]': NC (7.0/2.0)
|   |   |   |   Qt_pulsos_local = '(206-309]': NC (1.0)
|   |   |   |   Qt_pulsos_local = '(309-412]': C (1.0)
|   |   |   |   Qt_pulsos_local = '(412-515]': NC (0.0)
|   |   |   |   Qt_pulsos_local = '(515-618]': NC (0.0)
|   |   |   |   Qt_pulsos_local = '(618-721]': NC (0.0)
|   |   |   |   Qt_pulsos_local = '(721-824]': NC (0.0)
|   |   |   |   Qt_pulsos_local = '(824-927]': NC (0.0)
|   |   |   |   Qt_pulsos_local = '(927-inf)': NC (0.0)
|   |   |   Fat_fev = '(170.6-255.9]': C (5.0/1.0)
|   |   |   Fat_fev = '(255.9-341.2]': NC (1.0)
|   |   |   Fat_fev = '(341.2-426.5]': C (1.0)
|   |   |   Fat_fev = '(426.5-511.8]': NC (0.0)
|   |   |   Fat_fev = '(511.8-597.1]': NC (0.0)
|   |   |   Fat_fev = '(597.1-682.4]': NC (0.0)
|   |   |   Fat_fev = '(682.4-767.7]': NC (0.0)
|   |   |   Fat_fev = '(767.7-inf)': NC (0.0)
|   |   Fat_abr = '(116-232]': C (28.0/10.0)
|   |   Fat_abr = '(232-348]': NC (0.0)
|   |   Fat_abr = '(348-464]': NC (0.0)
|   |   Fat_abr = '(464-580]': NC (0.0)
|   |   Fat_abr = '(580-696]': NC (0.0)
|   |   Fat_abr = '(696-812]': NC (0.0)
|   |   Fat_abr = '(812-928]': NC (0.0)
|   |   Fat_abr = '(928-1044]': NC (0.0)
|   |   Fat_abr = '(1044-inf)': C (1.0)
|   Total_LD_14 = '(56.4-112.8]': NC (9.0/4.0)
|   Total_LD_14 = '(112.8-169.2]': NC (10.0/3.0)
|   Total_LD_14 = '(169.2-225.6]': C (1.0)
|   Total_LD_14 = '(225.6-282]': NC (0.0)
|   Total_LD_14 = '(282-338.4]': C (1.0)
|   Total_LD_14 = '(338.4-394.8]': C (1.0)
|   Total_LD_14 = '(394.8-451.2]': NC (1.0)
|   Total_LD_14 = '(451.2-507.6]': NC (0.0)
|   Total_LD_14 = '(507.6-inf)': C (1.0)
Tempo_inst = '(53.8-79.7]': NC (1089.0/318.0)
Tempo_inst = '(79.7-105.6]': NC (516.0/159.0)
Tempo_inst = '(105.6-131.5]': NC (481.0/155.0)
Tempo_inst = '(131.5-157.4]': NC (202.0/54.0)
Tempo_inst = '(157.4-183.3]': NC (53.0/12.0)
Tempo_inst = '(183.3-209.2]': NC (75.0/11.0)
Tempo_inst = '(209.2-235.1]': NC (430.0/99.0)
Tempo_inst = '(235.1-inf)': NC (8.0/3.0)

Number of Leaves  :       134

Size of the tree :  149
```

3) Resultados do classificador Naive Bayes

3.a) Validação utilizando o próprio conjunto de treinamento

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        3356               66.5213 %
Incorrectly Classified Instances      1689               33.4787 %
Kappa statistic                          0.2842
Mean absolute error                      0.399
Root mean squared error                  0.4762
Relative absolute error                 82.4024 %
Root relative squared error             96.7856 %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.48      0.206      0.62      0.48      0.541    C
  0.794     0.52       0.686     0.794     0.736    NC

=== Confusion Matrix ===

    a    b   <-- classified as
  996 1078 |    a = C
  611 2360 |    b = NC
```

Naive Bayes (Classificador Bayesiano)

3.b) Validação utilizando Divisão % (Parâmetro utilizado: 66% de treinamento)

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances        1148               66.8998 %
Incorrectly Classified Instances       568               33.1002 %
Kappa statistic                          0.2798
Mean absolute error                      0.3983
Root mean squared error                  0.4735
Relative absolute error                 82.4256 %
Root relative squared error             96.6779 %
Total Number of Instances             1716

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.465     0.196       0.61     0.465      0.527    C
  0.804     0.535       0.695    0.804      0.745    NC

=== Confusion Matrix ===

   a   b   <-- classified as
 317 365 |   a = C
 203 831 |   b = NC
```

Naive Bayes (Classificador Bayesiano)

3.c) Validação cruzada em 10 etapas:

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        3324               65.887  %
Incorrectly Classified Instances      1721               34.113  %
Kappa statistic                          0.2734
Mean absolute error                      0.4022
Root mean squared error                  0.4792
Relative absolute error                 83.0614 %
Root relative squared error             97.4    %
Total Number of Instances             5045

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.485     0.219      0.607     0.485     0.539    C
  0.781     0.515      0.684     0.781     0.729    NC

=== Confusion Matrix ===

    a    b   <-- classified as
 1005 1069 |    a = C
  652 2319 |    b = NC
```