

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CTC - CENTRO TECNOLÓGICO
INE – DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

**DESCOBERTA DE CONHECIMENTO EM
SISTEMAS GERENCIADORES DE BANCOS DE
DADOS: mineração de dados**

ROYQUENER REUTER

FLORIANÓPOLIS, NOVEMBRO DE 2004.

ROYQUENER REUTER

**DESCOBERTA DE CONHECIMENTO EM SISTEMAS
GERENCIADORES DE BANCOS DE DADOS:
mineração de dados**

Trabalho de Conclusão de Curso, apresentado ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Sistemas de Informação, orientado pelo Professor Dr. Paulo José Ogliari.

FLORIANÓPOLIS, NOVEMBRO DE 2004.

ROYQUENER REUTER

**DESCOBERTA DE CONHECIMENTO EM SISTEMAS
GERENCIADORES DE BANCOS DE DADOS: mineração de dados**

Trabalho de Conclusão de Curso apresentado ao Departamento de Informática e Estatística da Universidade Federal de Santa Catarina para obtenção do título de Bacharel em Sistemas de Informação.

BANCA EXAMINADORA

Professor Dr. Paulo José Ogliari
Departamento de Informática e Estatística – UFSC
Presidente

Professor Dr. Dalton Francisco de Andrade
Departamento de Informática e Estatística – UFSC

Professor Dr. Pedro Alberto Barbeta
Departamento de Informática e Estatística – UFSC

Professor Dr. José Leomar Todesco
Departamento de Informática e Estatística – UFSC

FLORIANÓPOLIS, NOVEMBRO DE 2004.

AGRADECIMENTOS

Aos Profissionais da Softway Contact Center, que cada qual com sua singularidade possibilitaram o êxito deste trabalho, trabalhar com vocês possibilitou um grande aprendizado. Um abraço especial ao Topázio, Paulo Muller e Silvio Borges que sempre estiveram presentes nos momentos decisivos de minha vida acadêmica e profissional.

Aos Amigos e Colegas: "No início, unidos apenas por um objetivo comum. Recuados e desconfiados, aos poucos a convivência foi nos aproximando. Sempre colegas, soubemos conviver e nos respeitar. Lutamos, sobrevivemos, crescemos. Acima de tudo como seres humanos. E, por tudo, a saudade há de ficar." A todo o grupo muito obrigado, sucessos para cada um de vocês,

Aos amigos, Rodrigo, Wagner e Eduardo por tudo que vivenciamos no decorrer do curso, pela dedicação e paciência, justamente por acreditar na possibilidade de realizar um sonho, muito obrigado. E obrigado pelas muitas horas dedicadas pelo bem e sucesso do grupo.

Aos professores: "Ser professor não é apenas lecionar. Ensinar não é apenas transmitir o conteúdo programático. Ser professor é, ser orientador e amigo, guia e companheiro, é caminhar com o aluno passo a passo. É transmitir a este os segredos da caminhada. Ser professor é ser exemplo de dedicação, de doação, de dignidade pessoal e de amor. O agradecimento sincero aos professores e amigos, aos somente professores, e àqueles que, com seus problemas e dores humanas, não foram amigos e nem professores, mas que também passaram por nós. Meu respeito, e afeto".

A DEUS, que me deu a vida e que se fez presente nesta caminhada, mesmo nos momentos de pouca fé, fazendo-me concretizar este sonho.

Aos meus pais, ainda que quase sempre ausentes obrigado por oportunizar a vida. Aos meus irmãos Royciner e Bernadete por mostrar-me que na vida, não importando o quão grande seja a distância, sempre haverá um jeito de se reencontrar, perdoar e abraçar novamente.

Aos meus avós, Lúcia e Hercílio pela dedicação, tempo e energia gastos para tornar-me um ser humano melhor. Através de seu amor, da sua educação e dos seus conselhos foi possível estar aqui.

Àqueles, que mesmo por alguns instantes, contribuíram de forma decisiva, deixando lembranças e lições de vida, mas infelizmente não podem estar presentes senão nos meus pensamentos e sonhos.

Aos meus familiares, em especial à minha sogra Arlete por estar sempre presente e disposta a ajudar de forma simples e espontânea. Às minhas cunhadas Giulia e Deise, ao meu cunhado Arion, e às minhas sobrinhas Darian e Adne.

E finalmente a você Elaine, minha amiga e companheira. Sempre presente e disposta a me ajudar nos momentos mais difíceis. Por acreditar em meu potencial e fazer-me enxergar que os defeitos e problemas serão sempre uma oportunidade de mudança e crescimento. É quase impossível definir em palavras meus sentimentos e minha gratidão por estar ao meu lado lutando e incentivando na busca de sonhos pessoais e comuns.

Graças a estas pessoas, foi possível realizar um sonho...

REUTER, Royquener. **DESCOBERTA DE CONHECIMENTO EM SISTEMAS GERENCIADORES DE BANCOS DE DADOS:** mineração de dados. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação). Universidade Federal de Santa Catarina, Florianópolis, 2004.

RESUMO

Data Mining se refere à extração ou “mineração” de conhecimento de grandes quantidades de dados. Considerando que os Bancos de Dados podem alcançar tamanhos na ordem de terabytes e dentro desta massa de dados podem estar ocultas muitas informações de importância estratégica, são essenciais buscá-las para manter uma margem competitiva em cada fase do ciclo de vida de um cliente. Entre os muitos segmentos de vendas e relacionamento com o cliente, o setor de telemarketing tem se destacado devido à sua rápida expansão. O objetivo geral é estudar e aplicar os conceitos de DM no setor, visando elaborar uma base de conhecimento para a utilização de ferramentas de mineração integradas a SGBD's (*Sistemas Gerenciadores de Bancos de Dados*). Apresentar-se-á o termo DCBD (Descoberta de Conhecimento em Banco de Dados), seus processos de descoberta de conhecimento em banco de dados através do padrão CRISP-DM (Cross-Industry Standard Process for Data Mining), além da análise de algumas das relações do DM com outras tecnologias de armazenamento e tratamento de informações. Serão abordadas as principais tarefas de DM para a busca de relacionamentos e padrões, principais técnicas estatísticas e de aprendizado de máquina. Será apresentada a especificação de padrão industrial para a construção de provedores de DM (MS OLE-DM for Data Mining) e através de aplicação prática, os resultados alcançados. Por fim, conclui-se o trabalho com as reflexões e considerações finais sobre a questão apresentada.

Palavras-chaves: Data Mining, Sistemas Gerenciadores de Banco de Dados, Knowledge Discovery in Database, Árvores de Decisão.

REUTER, Royquener. **KNOWLEDGE DISCOVERY IN DATABASE MANAGEMENT SYSTEMS**: data mining.

ABSTRACT

Data Mining refers to the extraction of knowledge from large quantities of data. Considering that databases might grow to sizes of the order of terabytes, and that inside this data many strategic information might be hidden, it is essential to seek them in order to maintain a competitive level in each phase of a client's life cycle. Among many sales and customer relationship segments, the telemarketing business has outstanced due to its quick expansion. The overall goal is to study and apply the concepts of DM in this segment, creating a knowledge base for the use of data mining tools integrated to the DBMS. The DBDM term will be presented, along with its knowledge discovery processes in databases through the CRISP-DM standard, and the analysis of some of the DM relationships with other information storage and handling technologies. The main data mining tasks used in the search of relationships and patterns, and the main statistics and machine learning techniques will be approached. Also the specification of the industrial standard for the construction of data mining providers (MS OLE-DM for Data Mining) will be presented - and, through the use of an application, the achieved results. Finally, the work will be concluded with the thoughts and final considerations regarding the studied matter.

Keywords: data mining, database management systems, database knowledge discovery, decision trees.

LISTA DE ABREVIATURAS

Sigla	Significado	Descrição / Comentário
ABT	Associação Brasileira de Telemarketing	Entidade sem fins lucrativos que congrega empresas que fornecem equipamentos, prestam serviços ou utilizam, de diversas formas, o Telemarketing.
API	Application Programming Interface	Interface de Programação entre Aplicativos.
COM	Component Object Model	
CRISP-DM	Cross-Industry Standard Process for Data Mining	Padrão inter-industrial para o processamento de tarefas de mineração de dados.
CRM	Customer Relationship Management	Gerenciamento do relacionamento com o cliente.
DCBD	Descoberta de Conhecimento em Base de Dados	O mesmo que KDD.
DM	Data Mining	Mineração de Dados
DMM	Data Mining Model	Modelo de Mineração de Dados.
DSO	Decision Support Objects	Objetos de Suporte à Decisão
DTS	Data Transformation Services	Serviços de Transformação de Dados
DW	Data Warehouse	Depósito de dados.
IA	Inteligência Artificial	
KDD	Knowledge Discovery Database	Descoberta de Conhecimento em Banco de Dados
MSDT	Microsoft Decision Tree	Árvore de Decisão da Microsoft
ODBC	Open DataBase Connectivity	
OLAM	On-Line Analytical Mining	Mineração Analítica em Tempo Real.
OLAP	OnLine Analytical	Processamento Analítico em Tempo

	Processing	Real
OLE DB	Object Linking and Embedding Database	Conjunto de Interfaces que permite a integração de recursos de banco de dados entre aplicativos.
OLE DB DM	Object Linking and Embedding Database for Data Mining	Conjunto de Interfaces que permite a integração de recursos de mineração entre aplicativos.
SGBD	Sistema Gerenciador de Bancos de Dados	
SQL	Structured Query Language	Linguagem estrutura de consultas.
UDF	User-Defined Funtion	Função definida pelo usuário.
UDF	User-Defined Functions)	Funções definidas pelo usuário
WEB	Teia	Abreviação de WWW
WWW	World Wide Web	Em termos gerais, é a interface gráfica da Internet.

LISTA DE FIGURAS

Figura 1.1 – Processos de KDD	33
Figura 1.2 – Fases do modelo de referência do CRISP-DM	34
Figura 2.1 – Processo de estimar a acuracidade com o método holdout	48
Figura 2.2 – Árvore de Decisão que representa o cliente que comprará ou não um computador	53
Figura 2.3 – Algoritmo para geração de uma árvore de decisão	54
Figura 2.4 – Árvore de decisão que representam o atributo estudante	59
Figura 4.1 – Visão geral das funcionalidades de DM inseridas no SQL Server 2000	82
Figura 4.2 – Hierarquia de objetos DSO	85
Figura 4.3 Diagrama de Relacionamentos	89
Figura 4.4 – Visualização do DMM através de componentes gráficos	94
Figura 4.5 – Visualização do DMM através de XML	95
Figura 4.6 – Exemplo de consulta para consumo do DMM	98

LISTA DE LISTAGENS

Listagem 3.1 – Exemplo de criação de um DMM, usando a notação proposta pelo padrão OLE DB DM	72
Listagem 3.2 – Exemplo de operação de inserção de um DMM, usando a notação proposta pelo padrão OLE DB DM.....	74
Listagem 3.3 – Exemplo de uma junção com um DMM, usando a notação proposta pelo padrão OLE DB DM.....	75
Listagem 4.1 – Criação do DMM.....	93
Listagem 4.2 – Povoamento do DMM.....	93
Listagem 4.3 – Consulta de junção com a base de validação.....	96

LISTA DE TABELAS

Tabela 2.1 – Registros de possíveis compradores	59
Tabela 2.2 – Entropia total do atributo “estudante?”	59
Tabela 2.3 – Entropia total do atributo idade	60
Tabela 2.4 – Entropia total do atributo classe de crédito	60
Tabela 2.5 – Entropia total do atributo rendimento	60
Tabela 2.6 – Entropia total na raiz principal	60
Tabela 4.1 – Variáveis selecionadas	88
Tabela 4.2 – Parâmetros utilizados para o treinamento	92

SUMÁRIO

INTRODUÇÃO	15
CAPÍTULO 1 DCBD – DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS	22
1.1 A Integração de DM e Bancos de Dados Relacionais	24
1.2 Relações entre Data Mining e outras Tecnologias	26
1.2.1 Data Mining e Data Warehouse	26
1.2.2 Data Mining e OLAP (<i>OnLine Analytical Processing</i>)	27
1.2.3 DM, Inteligência Artificial e Estatística	30
1.3 Processos de DCBD	31
CAPITULO 2 MINERAÇÃO DE DADOS	37
2.1 As Tarefas de DCBD	37
2.1.1 Tarefas ou Mineração Descritiva	38
2.1.1.1 Caracterização e Comparação	38
2.1.1.2 Associação	39
2.1.1.3 Segmentação e Cluster Analysis	41
2.1.2 Tarefas Preditivas	44
2.1.2.1 Classificação	45
2.1.2.2 Regressão	46
2.1.3 Comparando métodos de classificação	47
2.1.4 Acuracidade do Classificador	48
2.2 Árvores de Decisão	51
2.2.1 Indução de Árvores de Decisão	53
2.2.1.1 Seleção de Atributos	56
2.2.1.2 Poda da Árvore de Decisão	61
2.2.1.3 Algoritmo C4.5	63
CAPÍTULO 3 ESPECIFICAÇÃO DE PROVEDORES DE RECURSOS DE DM	65
3.1 Tecnologia OLE DB DM	66
3.1.1. Motivações	66
3.1.2 Propostas	67
3.1.3 Filosofia Básica do OLE DB DM	68
3.1.4 Componentes Básicos do OLE DB DM	69
3.1.4.1. Dados como Casos	70
3.1.5 Criar e Definir modelos de DM	71
3.1.6 Operações na modelagem de dados	73
3.1.7 Considerações finais sobre OLE DB DM	76

CAPITULO 4 ESTUDO DE CASO	78
4.1 Análise do negócio	78
4.2 Recursos tecnológicos utilizados	80
4.2.1 Microsoft® SQL Server™ 2000 Analysis Services	81
4.2.1.1 Classificador MSDT (<i>Microsoft Decision Tree</i>)	82
4.2.2 <i>Decision Support Objects</i>	83
4.2.3 Linguagem de desenvolvimento C#	85
4.3 A base de dados	87
4.3.1 Seleção dos atributos	87
4.3.2 Preparação dos dados	90
4.3.3 Amostragem	90
4.4 Desenvolvimento da aplicação	91
4.5 Aplicação da técnica de árvore de decisão	92
4.5.1 Construção	92
4.5.2 Validação	95
4.5.3 Consumo do DMM	97
 CONCLUSÃO	 99

REFERÊNCIAS BIBLIOGRÁFICAS

ANEXOS

APÊNDICE

INTRODUÇÃO

De forma simples, Data Mining, se refere à extração ou “mineração” de conhecimento de grandes quantidades de dados. Há muitos outros termos conduzindo de uma forma similar ou para diferentes propósitos na mineração de dados. Assim como: “minerando conhecimento de banco de dados”, “extraíndo conhecimento”, “dados/padrões de análise” e “escavando dados”. (HAN e KAMBER, 1999).

Devido ao aumento da competição por lucro no mercado, a mineração de dados tem se tornado uma prática essencial para manter uma margem competitiva em cada fase do ciclo de vida de um cliente. Historicamente, uma forma de minerar informações de dados é conhecida como “escavar dados”. Isto é, um procedimento abaixo de padrões para uma boa investigação. Implica basicamente que um analista ou pesquisador procure informações através dos dados sem nenhuma hipótese ou técnica específica predeterminada. Recentemente, esta prática tem se tornado mais aceitável, principalmente porque é uma metodologia que eventualmente pode conduzir para a descoberta de algumas informações valiosas. Se através do acaso se descobre uma informação útil que possa incrementar o lucro ou a receita, o criador deste processo rapidamente ganha aceitação e respeito dentro da organização. (RUD, 2001)

Os Bancos de Dados podem alcançar tamanhos na ordem de terabytes e dentro desta massa de dados podem estar ocultas muitas informações de importância estratégica. A recente resposta é minerar informações tendo como principais objetivos o incremento de renda e redução de custos. As organizações estão usando a mineração de dados para localizar clientes, para reconfigurar seus

produtos oferecidos, para aumentar vendas e para minimizar perdas por erros ou fraudes. (TWO CROWS CORPORATION, 1999)

Mas ao o que se deveu a evolução da mineração de dados em bancos de dados? A resposta está na crescente capacidade tanto de gerar e coletar dados nas últimas décadas. As principais causas que contribuíram para este crescimento foram: o uso dos códigos de barras nos produtos, a informatização dos negócios, ciências e regime de transação e gestão, e avanços em ferramentas de coleta de dados, para instrumentação, manufatura e vendas on-line. Em adição, o uso popular da World Wide Web (www) como um sistema de informação global tem nos inundado com uma grande quantidade de dados e informações. Esta explosão fez crescer a necessidade em gerar e armazenar dados, de novas técnicas e ferramentas de automação que podem nos assistir em transformar a grande quantidade de dados em informação útil e conhecimento. (TWO CROWS CORPORATION, 1999)

Estamos cada vez mais dependentes da criação, administração e distribuição de recursos de informação. As empresas estão se expandindo para vender seus produtos e serviços, realizando parcerias e competindo cada vez mais com concorrentes por clientes. Em um ambiente competitivo, tem ser tornado cada vez mais difícil conquistar novos clientes e sustentar os diferenciais no atendimento destes. Descobrir algo realmente novo, que possa significar um diferencial competitivo, requer investimento e tempo. Os diferenciais obtidos, em muitos casos, são rapidamente copiados pela concorrência. Ter acesso rápido a dados e poder transformar esses dados em informações que auxiliam no processo decisivo, é fator determinante para a reversão desse quadro.

Até poucos anos atrás, os gerentes e diretores de uma empresa podiam descobrir o que estava acontecendo consultando demonstrativos contábeis, caminhando por corredores ou conversando com seus colaboradores. Hoje, as informações necessárias estão ocultas em algum lugar no incomensurável número de dados depositados por nossos sistemas. Elas estão nos milhares de linhas das tabelas de nossos bancos de dados.

Torna-se necessária a implementação de mecanismos que possam oferecer acesso às informações estratégicas, de forma a melhorar a produtividade através da busca de novas visões do negócio; aumentar a velocidade e flexibilidade na obtenção das informações e colocar as informações na ponta para quem tem que decidir.

Entre os diversos segmentos de vendas e relacionamento com o cliente, o setor de telemarketing tem se destacado no cenário mundial devido à sua rápida expansão. O telemarketing se divide em dois segmentos principais: o receptivo e o ativo. O receptivo permite aos clientes encomendarem produtos ou façam indagações e/ou sugestões sobre o serviço de atendimento. O telemarketing ativo ocorre quando a empresa liga para um cliente atual ou potencial com uma oferta, um anúncio ou uma solicitação de pagamento.

Hoje o telemarketing atua por meio de multicanais, como a Internet – vem crescendo acima das expectativas e, conseqüentemente, gerando mais empregos. De 465 mil pessoas trabalhando na área em 2002, em todo o Brasil, o número foi para 500 mil em 2003, o que equivale a um crescimento de 7,5%. “O setor de tele-serviços, que engloba as atividades de telemarketing, centrais de atendimento, *help desk*, dentre outras atividades por telefone, vem crescendo no Brasil intensamente

nos últimos cinco anos”, conforme Topázio Silveira Neto, presidente da ABT. (apud COLOGNA, 2004)

A que se deve o sucesso do telemarketing? De acordo com o autor, isso “Decorre, principalmente, da expansão da base de telefonia no País, além de uma maior conscientização e preocupação das empresas no sentido de atender e abrir canais de comunicação com seus consumidores”, para ele, “este movimento, de crescimento do mercado e geração de novas atividades, fez com que o número de empregos no setor tenha crescido entre 1997 e 2003 mais de 235%”. Para 2004, segundo o mesmo, está previsto um crescimento de 10% no número de empregos, gerando mais de 50 mil empregos diretos neste ano.

De acordo com pesquisas da ABT, 60% destas vagas estão disponíveis no Estado de São Paulo que, assim como o Rio de Janeiro, concentra a maioria das grandes empresas. Apesar destas constatações, este cenário vem mudando. Ocorre em função de incentivos fiscais por parte do governo e outros fatores, como abundância de mão-de-obra qualificada, infra-estrutura e acesso à educação, o que vem sendo oferecido por outras cidades. Segundo Topázio apud Colagna, 2004, “[...] Hoje já existem centrais de atendimento espalhadas por todo o Brasil, de Manaus (AM) a Novo Hamburgo (RS)”.

Entre os segmentos de mercado em que o telemarketing atua, o mercado de cartões de crédito e serviços associados tem uma posição de destaque. Trata-se de 38 milhões de cartões vendidos e com um enorme potencial considerando que mais de 80% das pessoas com mais de 16 anos não possuem cartão de crédito.

Este trabalho terá como preocupação estudar e aplicar os recursos e benefícios do DM para o aperfeiçoamento deste segmento de telemarketing nas suas principais preocupações como uma empresa de *call center*: procurar meios de

automatizar processos que possam melhorar o contato com o cliente e buscar respostas para o impacto social conseqüente deste rápido crescimento. Espera-se também melhorar a produtividade das campanhas e conseqüente diminuição dos custos relacionados.

Esta preocupação se deve porque o universo dos clientes atuais e potenciais é muito concorrido. Adicional a esta situação se identifica nos últimos anos uma crescente perda de rentabilidade na aquisição de cada novo cliente, provocada pelo aumento dos custos, principalmente de telefonia e de pessoal. Ao entender esse mercado, é necessário ter a preocupação em oferecer soluções que melhorem a competitividade e a rentabilidade.

Está claro que não basta receber uma campanha de telemarketing e simplesmente executá-la no *call center*. É necessário criar modelos que permitam um melhor aproveitamento de cada campanha em cada região, da característica e do perfil de cada consumidor, da oferta de cada produto e da situação de telefonia onde envolvem custos e produtividade.

A experiência no setor mostra que os resultados veêm da observação de pequenos detalhes, cujos percentuais são cada dia menores. Ou seja, em cada cadastro a variação de telefones errados de mais 1%, ou menos 1%, pode significar o sucesso ou o fracasso de uma campanha. De outra forma, a velocidade para identificar e reagir a essa pequena variação, também contribui muito para o seu sucesso.

A importância do DM tem que ser vista como a descoberta de conhecimento através de fatos e dados e não por suposições. Nem sempre poderá ser possível automatizar processos com base no Data Mining mas, no entanto, será possível a

descoberta da informação à luz de preceitos técnicos e científicos e o mais importante: sedimentar este conhecimento de forma organizada.

Trata-se portanto, de mais um mecanismo para conhecer intimamente as singularidades do negócio de telemarketing, vindo a complementar as tecnologias de informação já existentes.

A presente monografia tem como **objetivo geral**, o estudo e aplicação dos conceitos de DM no setor de telemarketing, com foco em uma operação ativa de venda de cartões de crédito, visando elaborar uma base de conhecimento para a utilização de ferramentas de mineração integradas a SGBD's (*Sistemas Gerenciadores de Bancos de Dados*) e suas técnicas de mineração de dados.

O desenvolvimento deste estudo buscou atender o seguinte **objetivo específico**:

- Através de recursos tecnológicos disponíveis no mercado, desenvolver uma aplicação visando construir um modelo preditivo para melhorar a performance de contatos com os clientes.

A exposição deste trabalho foi organizada em quatro capítulos. No capítulo 1 apresentar-se-á o termo DCBD (*Descoberta de Conhecimento em Banco de Dados*), seus processos de descoberta de conhecimento em banco de dados através do padrão CRISP-DM (*Cross-Industry Standard Process for Data Mining*), além da análise de algumas das relações do DM com outras tecnologias de armazenamento e tratamento de informações.

No capítulo 2, serão abordadas as tarefas de DM para a busca de relacionamentos e padrões. Também serão apresentadas as principais técnicas estatísticas e de aprendizado de máquina relacionadas aos objetivos propostos.

No capítulo 3, será apresentada uma especificação de padrão industrial para a construção de provedores de DM (MS OLE-DM for Data Mining) para a realização dos objetivos propostos, com as características e motivações para o seu desenvolvimento.

No capítulo 4, através de aplicação prática dos conceitos apresentados nos capítulos anteriores, serão apresentados os resultados alcançados.

Por fim, conclui-se o trabalho com as reflexões e considerações finais sobre a questão apresentada.

CAPÍTULO 1 DCBD – DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

A mineração de dados é tratada por muitas pessoas como um sinônimo de outro termo popularmente usado, KDD (Knowledge Discovery in Database) ou DCBD (Descoberta de Conhecimento em Base de Dados). Alternativamente algumas outras visões de mineração de dados podem ser consideradas simplesmente um passo do processo de DCBD. (HAN e KAMBER, 1999)

O termo Data Mining é comumente usado por estatísticos, analistas de dados e pela comunidade MIS (Management Information Services); enquanto pesquisadores de inteligência artificial utilizam KDD. (AMARAL, 2001)

Neste trabalho será mais comumente utilizado o termo Data Mining enquanto referir-se ao processo de minerar dados e KDD ou DCBD ao se referir ao projeto como um todo.

Resumidamente, o primeiro e simples passo para a mineração de dados é a sua descrição – sumarizar atributos estatísticos (assim como medir e indicar desvios), examiná-los visualmente usando mapas e gráficos e procurar por potenciais e significantes ligações entre variáveis (assim como valores que muitas vezes ocorrem juntamente). No entanto, somente dados descritos não podem prover um plano de ação, é necessário construir um modelo preditivo baseado em padrões determinados por resultados já descobertos e então testar esse modelo de resultados nos exemplos originais.

O passo final é verificar o modelo de forma empírica. Por exemplo, para uma base de dados de clientes que já têm respostas para uma oferta particular, já existe

embutido um modelo de predição nas quais as possibilidades são prováveis para responder à mesma oferta. Seria possível confiar nesta predição?

Pode-se dizer, com o objetivo de simplificar o entendimento, que o DM é uma ferramenta e não está em seus propósitos ficar monitorando o banco de dados prevendo o que vai acontecer e enviar um e-mail para chamar a atenção ao encontrar um padrão interessante. Tampouco eliminará a necessidade de conhecer o negócio, de entender os dados ou de entender métodos analíticos. Seu objetivo é somente ajudar a analisar o negócio procurando padrões e relacionamentos entre os dados – não apresenta valores de padrões. Os padrões descobertos por DM devem ser verificados no mundo real. (TWO CROWS CORPORATION, 1999)

Os relacionamentos preditivos encontrados não são necessariamente causas de uma ação ou comportamento. Por exemplo, o DM pode determinar que homens com certa renda anual que assinam certas revistas são prováveis compradores de um produto. Enquanto este padrão for aproveitado, não é possível assumir que alguns destes fatores são as causas para a compra do produto. Para assegurar resultados significantes, é vital entender os dados. O DM não descobre soluções automaticamente sem um direcionamento, particularmente se tiver um objetivo turvo.

Outro ponto importante: ainda que uma boa ferramenta de mineração de dados proteja o usuário de complicadas técnicas estatísticas, requer que haja domínio sobre a ferramenta e o algoritmo na qual elas são baseadas. (TWO CROWS CORPORATION, 1999)

A escolha da configuração da ferramenta de DM e a escolha de sua otimização podem afetar na precisão e velocidade do modelo. O DM também não pode substituir analistas de negócio ou administradores; somente entregar novas

ferramentas para melhorar o desempenho do trabalho. (TWO CROWS CORPORATION, 1999)

1.1 A Integração de DM e Bancos de Dados Relacionais

O progresso da pesquisa em DM tem vindo da possibilidade de implementar várias operações de mineração de forma eficiente em grandes bancos de dados. Enquanto isto é seguramente uma importante contribuição, não podemos deixar de olhar os objetivos finais do DM – isto é, permitir que aplicações de banco de dados construam modelos de DM (por exemplo: árvores de decisão e classificação, modelos de regressão, segmentação) e usem estes modelos para realizar tarefas preditivas e analíticas e possam também compartilhar estes mesmos modelos com outras aplicações. Tal integração deve ser uma pré-condição para que o DM tenha sucesso em banco de dados. (NETZ *et al*, 2000)

Reconhecendo o fato acima, é obvio que um aspecto chave para integração com sistemas de banco de dados que é preciso ser observado é como tratar modelos de DM como objetos de primeira classe¹ nos SGBD's. Infelizmente, em qualquer aspecto, o DM ainda continua sendo uma “ilha” de análises que é pobremente integrada com sistemas de banco de dados.

Lembrando que um modelo de DM é obtido via aplicação de um algoritmo de DM em um conjunto de treinamento. Desta forma, mesmo que um modelo possa ser derivado usando uma aplicação SQL (*Structured Query Language*) que implemente um algoritmo de treinamento, o SGBD é completamente inconsciente da semântica

¹ Refere-se a objetos de primeira classe, quando podemos representá-los na forma de ocorrências com seus atributos e inseridos em tabelas.

do modelo de DM porque estes não podem ser representados explicitamente em um banco de dados.

Em seguida, para efetivamente representar modelos de mineração em SGBD's, precisamos capturar a criação da mineração de dados usando algoritmos arbitrários de mineração, navegar por estes modelos (examinando suas estruturas ou seu conteúdo), e aplicar um modelo selecionado num conjunto de dados para analisar tarefas como uma previsão. Além disto, para a coluna que é o resultado da previsão, são necessárias informações suficientes para que outras ferramentas de análise possam interpretar as propriedades da previsão, como acuracidade, por exemplo. (NETZ *et al*, 2000)

Os SGBD's entendem e suportam somente relações como objetos de primeira classe e assim se queremos representar um modelo de DM em um banco de dados, este modelo precisa ser visto como a estrutura de uma tabela (ou um conjunto de linhas). Numa observação superficial, um modelo pode ser observado como um gráfico, com uma interpretação complexa de sua estrutura, como por exemplo, uma árvore de decisão.

Os passos chaves do ciclo de vida de um modelo estão em criar e alimentar um modelo via um algoritmo em cima de um conjunto de dados de treinamento, e ser capaz de predizer valores de conjunto de dados. Se for possível capturar estes passos usando um SQL que os represente, então está assegurado que desenvolvedores de banco de dados serão capazes para transferir funcionalidades deste paradigma para o desenvolvimento de aplicações. (NETZ *et al*, 2000)

1.2 Relações entre Data Mining e outras Tecnologias

1.2.1 Data Mining e Data Warehouse

Inicialmente, o DW (*Data Warehouse*) foi usado principalmente para gerar relatórios e responder consultas predefinidas. Progressivamente, começou a ser usado para analisar dados sumarizados e detalhados, onde os resultados eram apresentados na forma de relatórios e gráficos. Mais tarde, foi usado com fins estratégicos, como melhorar análises multidimensionais e sofisticar operações de “*slice-and-dice*”. Finalmente, o DW pôde ser empregado para a descoberta de conhecimento e decisões estratégicas usando ferramentas de mineração. Neste contexto, as ferramentas de DW podem ser categorizadas dentro de ferramentas de acesso e de busca de informações, ferramentas de geração de relatórios em BD, ferramentas de análise de dados e ferramentas de mineração. (HAN e KAMBER, 1999)

Freqüentemente os dados a serem minerados são primeiramente extraídos de um DW empresarial para um banco de DM ou Data Mart. Os dados de um DW podem não ser usado exclusivamente pelo DM, podem ser compartilhados também por OLAP e outras aplicações. De qualquer forma, por razões pragmáticas, aplicações de DM devem assumir o DW como um ‘backend’ relacional (CHAUDHURI, 1998). Há um real benefício se o dado já existir em um DW uma vez que dados operacionais ou de transação, não estão no formato mais adequado para o processo de mineração. Os problemas de limpeza dos dados para um *Data Warehouse* e para a mineração são muito similares. Se os dados já existirem limpos num DW, é muito mais provável que não precise limpá-lo para ser minerado. Além disto, já estarão

resolvidos muitos dos problemas de consolidação e não será necessário gastar tempo na manutenção do processo (TWO CROWS CORPORATION, 1999)

A análise do DM tende a ser da base para o topo, e as melhores técnicas têm seu desenvolvimento orientado para grandes volumes de dados. Isso é importante no contexto do DW, visto que usualmente se deseja usar o máximo possível dos dados coletados. (SINGH, 2001)

O banco de DM não precisa necessariamente ser uma parte física do DW, desde que o servidor de DW possa suportar a demanda adicional de recursos do DM. Por último, o DW não é um requerimento para minerar dados. (TWO CROWS CORPORATION, 1999)

1.2.2 Data Mining e OLAP (*OnLine Analytical Processing*)

A premissa de Sistemas de Suporte à Decisão é explorar dados empresariais para conquistar vantagem competitiva. O processo de decidir o que coletar e como limpar tais dados não é um assunto trivial. De qualquer modo, mesmo que um DW tenha sido construído, isto muitas vezes é difícil analisar e assimilar os dados. A tecnologia OLAP resolve um importante passo do problema por permitir a visualização de dados de forma multidimensional - como uma gigantesca planilha eletrônica - com ferramentas visuais sofisticadas para visualizar e consultar dados. (CHAUDHURI, 1998)

Uma das questões mais comuns sobre o processamento de dados é saber qual a diferença entre DM e OLAP. Num primeiro momento o que se pode dizer é que são extremamente diferentes e que podem se complementar.

O OLAP é uma ferramenta de sumarização e agregação de dados que ajuda a simplificar a análise fazendo parte de um espectro de ferramentas de suporte à decisão. Já o DM permite a descoberta automatizada de padrões implícitos e de conhecimentos ocultos em grandes quantidades de dados. (HAN e KAMBER, 1999)

O OLAP é usado para responder como certas coisas podem ser verdadeiras. O usuário formula uma hipótese sobre um relacionamento e o verifica em uma série de consultas frente ao dado. Por exemplo, um analista talvez queira determinar quais fatores são determinantes para um empréstimo padrão. Inicialmente, pressupondo que pessoas com baixa renda são más pagadoras o analista verifica esta afirmação frente aos dados para aprovar ou desaprovar esta suposição. Se a hipótese não pode ser sustentada pelos dados, o analista talvez verifique se a dívida é um determinante para o risco. Se o dado também não apóia esta suposição, então, talvez seja necessário tentar a dívida e a renda juntas como um melhor prognóstico para o risco de crédito. Em outras palavras, o OLAP gera uma série de análises para padrões hipotéticos e relacionamentos usando consultas contra o banco de dados para confirmar ou desaprovar. *“OLAP é essencialmente um processo dedutivo”* (TWO CROWS CORPORATION, 1999)

Mas, o que poderá acontecer quando o número de variáveis a serem analisadas é uma dúzia ou até mesmo centenas? Tornar-se-á muito mais difícil e consumirá muito mais tempo para encontrar boas hipóteses e analisar o banco de dados com OLAP.

O DM é diferente do OLAP porque o DM verifica particularmente hipóteses padrões, usando os dados para descobrir padrões semelhantes. *“É essencialmente um processo indutivo”*. Por exemplo, suponha que um analista queira identificar os fatores de risco de empréstimos usando ferramentas de DM. A ferramenta talvez

descubra que as pessoas com dívidas altas e renda baixa têm risco de crédito, mas talvez descubra também um padrão de análise que o analista não pensou em procurar, como por exemplo, que a idade é também determinante para avaliar o risco. (TWO CROWS CORPORATION, 1999)

Neste ponto DM e OLAP podem ser complementares. Antes de agir com um padrão, o analista precisa saber que implicações financeiras podem existir se usar um padrão descoberto para administrar o empréstimo. A ferramenta OLAP pode ajudar o analista a responder a esta qualidade de questão. Além disto, OLAP é complementar à primeira fase do processo de descoberta de informação porque pode ajudar na exploração dos dados, a focalizar a atenção nas variáveis mais importantes, a identificar exceções ou encontrar interações. Isto é importante porque quanto melhor o entendimento dos dados, mais efetivo é o processo de descoberta.

Segundo Han e Kamber (1999), muitas pesquisas têm sido feitas para minerar dados de várias plataformas, incluindo bancos relacionais, bancos transacionais, bancos de textos, bancos de séries temporais, etc. Entre os muitos paradigmas e arquiteturas de sistemas de DM, o desenvolvimento do OLAM (*On-Line Analytical Mining*) o qual integra as pesquisas de OLAP com DM e mineração de conhecimento em bancos de dados multidimensionais, é particularmente importante pelas seguintes razões:

- Visa à qualidade dos dados em DW: a maioria das ferramentas de DM precisa trabalhar com dados consistentes, integrados e limpos, o qual requer um custo elevado. Um DW construído para ser um repositório de dados pré-processados é uma valiosa fonte de dados de alta qualidade tanto para OLAP como para DM. É preciso lembrar também que o DM também pode contribuir para a transformação e limpeza dos dados.

- Construir uma arquitetura de DW se preocupando com outras tecnologias: de acesso, integração, consolidação e transformação de dados de múltiplos e heterogêneos bancos de dados, conexões OLEDB (*Object Linking and Embedding Database*) / ODBC² (*Open DataBase Connectivity*), acesso WEB³, relatórios e ferramentas de OLAP.
- OLAP baseado na exploração e análise de dados, onde o objetivo será prover para o DM diferentes conjuntos de dados em diferentes níveis de abstração em um cubo de dados intermediário em cima dos resultados intermediários do DM.

Por último, conforme expõe Carvalho (2001) não se pode confundir OLAP com Data Mining. Sistemas OLAP emitem respostas para perguntas do tipo “que dados se encaixam neste padrão?” enquanto o DM responde à pergunta “que padrões existem nestes dados?”.

1.2.3 DM, Inteligência Artificial e Estatística

O DM tira vantagem dos avanços dos campos da IA (*Inteligência Artificial*) e estatísticas. Ambas disciplinas têm sido trabalhadas em problemas de reconhecimento de padrões e classificação. Estas comunidades de pesquisa têm feito grandes contribuições para o entendimento e aplicação de redes neurais e árvores de decisão no processo de DM.

² Uma especificação projetada pela Microsoft® para permitir que aplicações para Windows® acessem dados sem considerar os diversos formatos dos arquivos de dados e simplificar o acesso de forma que o usuário não tenha necessidade de um alto grau de conhecimento técnico para poder ter acesso a diferentes bancos de dados.

³ Abreviação de WWW (World Wide Web)

O DM não tem trocado técnicas estatísticas tradicionais. O desenvolvimento de muitas técnicas estatísticas tem sido, até recentemente, baseadas em teorias distintas e métodos estatísticos que têm trabalhado com um modesto número de dados a ser analisado. A diferença está no incremento do poder dos computadores que aliado ao seu baixo custo e com a necessidade de analisar enormes quantidades de dados, tem permitido o desenvolvimento de novas técnicas baseadas na exploração por “força bruta” de possíveis soluções. Outra diferença básica entre DM e estatística está no fato de que as técnicas de DM tendem a ser muito mais robustas para minerar massas de dados geralmente confusas e para usuários menos experientes.

O ponto chave do DM na aplicação da IA e técnicas estatísticas para problemas comuns ao negócio é que possam ser usadas de maneira que façam estas técnicas eficazes tanto para profissionais experientes no negócio como também para profissionais treinados em estatística. O DM é uma ferramenta para incrementar a produtividade de pessoas treinadas para construir modelos preditivos. (TWO CROWS CORPORATION, 1999)

1.3 Processos de DCBD

Conforme já foi definido na introdução deste trabalho, DM se refere à extração ou mineração de conhecimento de grandes quantidades de dados. De forma alternativa, outras visões de mineração de dados podem ser simplesmente um dos passos essenciais do processo de descobrir conhecimento em banco de dados

(HAN e KAMBER, 1999). Na Figura 1.1, a DCBD é descrita como um processo que consiste de uma seqüência interativa de passos:

- ✓ Consolidação dos Dados: os dados resultantes podem ser armazenados em um DW:
 - Limpeza: remover ruídos e dados irrelevantes.
 - Integração: os dados de diversas fontes são integrados;
- ✓ Seleção dos Dados: os dados relevantes para a tarefa de análise são retirados de um banco de dados ou DW;
- ✓ Transformação dos Dados: os dados são transformados ou consolidados em uma forma apropriada para ser minerado.
- ✓ Mineração dos Dados: um processo essencial onde métodos inteligentes são aplicados para extrair padrões.
- ✓ Avaliação do Padrão: para identificar e avaliar padrões verdadeiros utilizando métricas de validação;
- ✓ Apresentação e geração do conhecimento;

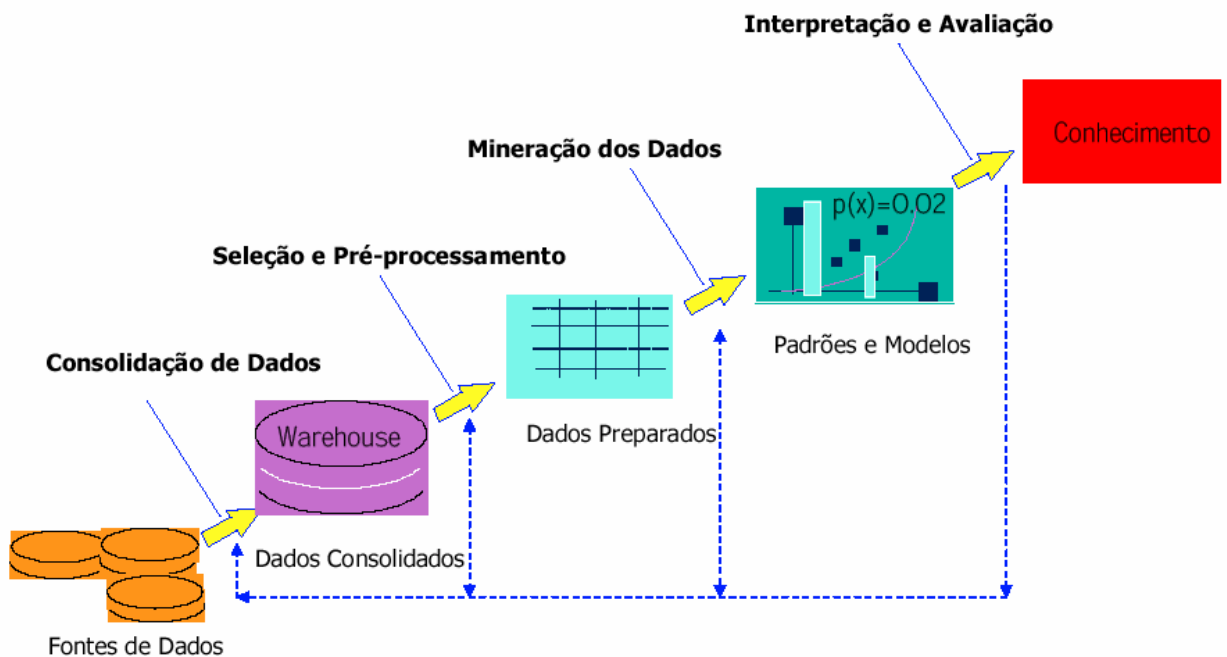


Figura 1.1 – Processos de KDD (HAN e KAMBER, 1999).

Mais recentemente, objetivando a geração de modelos com maior qualidade e utilizando-se de padronização de conceitos e técnicas na busca de informações para a tomada de decisões, foi formulado o padrão CRISP-DM que fornece um conjunto de metodologias, práticas e definições das atividades que envolvem o processo de DCBD. Este padrão, apesar de estar ligado a profissionais de empresas especializadas em DM e DW (DaimlerChrysler, SPSS e NCR), não se restringe a uma ferramenta ou tecnologia específica. Sua origem ocorreu em 1996.

Foi proposta uma metodologia que pudesse auxiliar os administradores e responsáveis nos processos de planejar e executar o DM, englobando desde a especificação do processo até a apresentação dos resultados obtidos.

Atualmente tem sido largamente utilizado e seu sucesso se deve principalmente por se basear na prática de profissionais que trabalham diretamente em projetos de DCBD.

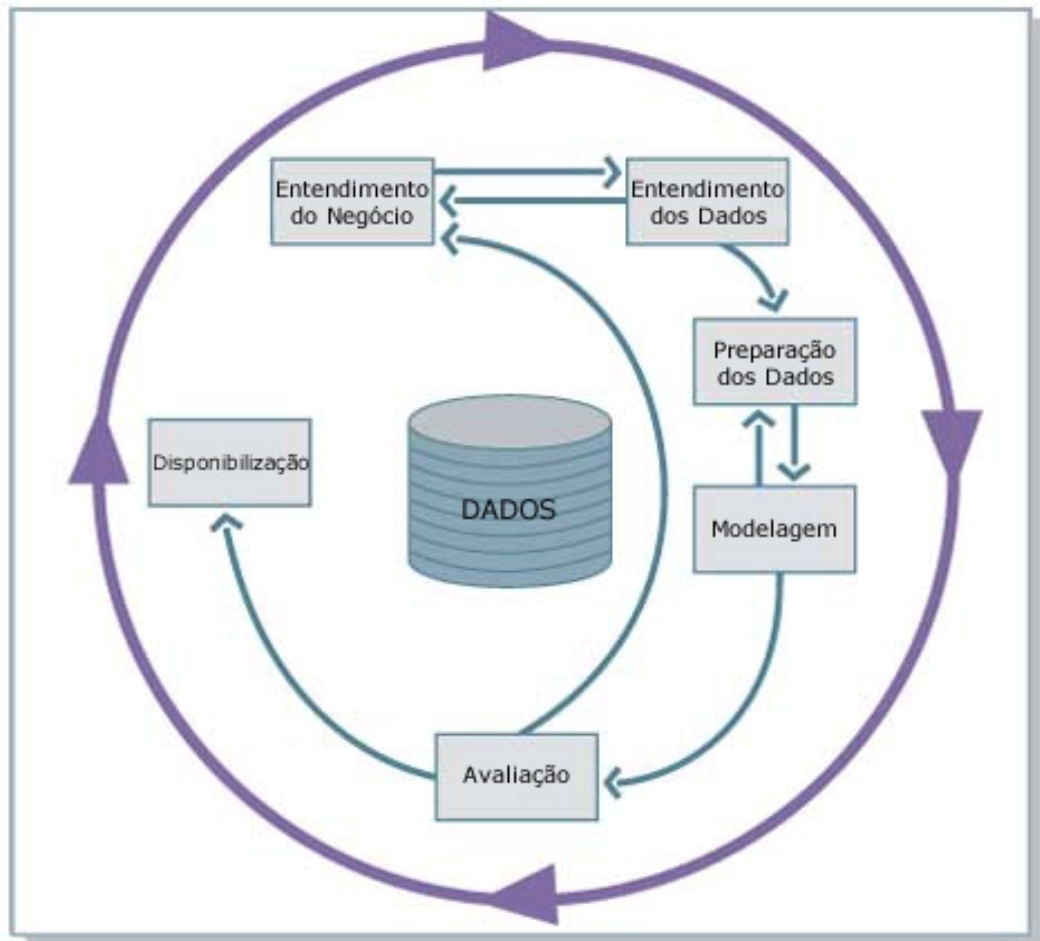


Figura 1.2 – Fases do modelo de referência do CRISP-DM.

O padrão CRISP-DM, conforme Figura 1.2, está fundamentado em seis fases distintas:

- **Entendimento do negócio:** foca no entendimento do negócio, analisar o que o cliente realmente precisa. Foca nos objetivos do projeto e requerimentos para a perspectiva do negócio. Este conhecimento é convertido em um problema de DM e um plano preliminar é projetado para alcançar os objetivos. Muitas vezes existem objetivos competindo e obstáculos que precisam ser corretamente balanceados. Esta fase se preocupa em descobrir os fatores que podem influenciar no resultado do

projeto. Negligenciar esta etapa pode gerar a possibilidade de um enorme esforço que produzirá respostas certas para questões erradas.

- **Entendimento dos dados:** objetiva, com uma coleção inicial de dados, procedimentos que buscam trazer familiaridade com o dado, identificar problemas de qualidade, ter as primeiras percepções dos dados e detectar subconjuntos que possam ser interessantes para formular hipóteses sobre informações ocultas;
- **Preparação dos dados:** a fase de preparação cobre todas as atividades para a construção do conjunto de dados finais (dados que podem ser alimentados em ferramentas de modelagem) dos dados brutos iniciais. As tarefas de preparação de dados são para serem executadas diversas vezes e não em uma ordem fixa. As tarefas incluem tabelas, registros e seleção de atributos assim como transformação e limpeza para as ferramentas de modelagem.
- **Modelagem:** enquanto possivelmente já foi selecionada uma ferramenta para compreender o negócio, esta tarefa se refere à seleção e aplicação de uma ou mais técnicas específicas e os parâmetros destas técnicas são ajustados para otimização. Exemplos: árvore de decisão construída com C4.5 ou rede neural com *backpropagation*. Se mais de uma técnica for aplicada, esta tarefa deve ser para cada técnica separadamente. Como algumas têm requerimentos específicos na transformação do dado, será preciso retornar diversas vezes para a fase de preparação dos dados.
- **Avaliação:** neste estágio do projeto, deverá haver um modelo construído (ou vários) que precisam apresentar alta qualidade para uma perspectiva de análise dos dados. Antes de proceder para a construção final do

modelo, é importante aprofundar e rever os passos executados para sua construção para ter certeza de que os objetivos do negócio serão alcançados.

- **Implantação:** a criação do modelo não é geralmente o fim do projeto. Mesmo que o propósito do modelo seja incrementar o conhecimento sobre os dados, o conhecimento adquirido precisa ser organizado e apresentado em um meio em que possa ser usado. Muitas vezes envolve em aplicar os modelos dentro da organização na construção de processos. Por exemplo em uma personalização em tempo real de Web Pages ou em um ranqueamento de banco de dados de marketing. De qualquer forma, dependendo dos requerimentos, a fase de implantação pode ser simples como gerar um relatório ou uma implementação complexa de um processo de DM que possa ser repetido. Em muitos casos o cliente, não o analista dos dados, é quem continuará os passos do desenvolvimento.

CAPITULO 2 MINERAÇÃO DE DADOS

Neste capítulo, serão apresentadas as principais tarefas e técnicas de DM. De acordo com os objetivos propostos na introdução desta monografia, será abordada com maior ênfase as tarefas de classificação. Quanto às técnicas, será apresentada em detalhes a técnica de Árvores de Decisão, por fazer parte da aplicação escolhida para o cumprimento dos objetivos propostos.

2.1 As Tarefas de DCBD

As tarefas de DCBD podem ser definidas como um conjunto de técnicas ou algoritmos que representam um tipo de conhecimento desejado do banco de dados sendo que cada tarefa vai requerer algoritmos diferentes para a extração do conhecimento. Em geral as tarefas de DM podem ser classificadas em duas categorias: **descritivas** e **preditivas** (HAN e KAMBER, 1999). Tarefas descritivas caracterizam as propriedades gerais dos dados nos bancos de dados. Tarefas preditivas atuam inferenciando nos dados correntes para fazer predições.

2.1.1 Tarefas ou Mineração Descritiva

Segundo Han e Kamber (1999), antes de construir modelos preditivos, é preciso entender o dado. Desta forma é preciso descrever conjuntos de dados em uma forma concisa e resumida que apresente as propriedades dos dados. Geralmente, SGBD's provêm ferramentas para que os usuários possam pesquisar os dados. As ferramentas de extração utilizam muitas vezes linguagens de consulta e pesquisa como SQL. Estas ferramentas podem ser usadas para encontrarem dados ou gerar uma lista todas as transações registradas. Utilizar as funções de agregação (média, soma, etc.) destas linguagens de consulta pode contribuir de forma significativa para o processo de conhecer os dados.

2.1.1.1 Caracterização e Comparação

Segundo Han e Kamber (1999), nesta tarefa o objetivo principal é caracterizar e comparar o dado. Caracterizar consiste em resumir de forma concisa uma determinada coleção de dados e comparar consiste em comparar duas ou mais coleções de dados. O conceito de descrição envolve tanto caracterização quanto comparação:

- **Caracterização:** é uma sumarização das características gerais ou traços do objeto alvo. Podemos dizer também que, dada possibilidade de que um grande número de dados pode estar armazenado, seria útil poder descrever análises de forma concisa e sucinta em vários níveis de abstração; facilitaria examinar o comportamento geral dos dados. Os

conceitos de múltiplas dimensões e sumarização de dados em vários níveis são similares à análise multidimensional de DW, mas diferem por terem um escopo menor de acordo com os objetivos definidos no projeto de DM. Os produtos resultantes da caracterização dos dados podem ser apresentados em várias formas, como gráficos, cubos de dados multidimensionais e tabelas multidimensionais (incluindo “*crosstabs*”). Os resultados podem também se apresentar em forma de regras.

- **Comparação ou Discriminação:** é uma comparação das características gerais do conjunto de dados do objetivo alvo com um ou mais conjuntos com características opostas, como, por exemplo, comparar as características de um determinado produto que teve suas vendas incrementadas em 10% com aqueles produtos que tiveram perda de 30% no mesmo período. Os produtos resultantes desta análise são os mesmos da caracterização.

2.1.1.2 Associação

A mineração de regras de associação é uma abordagem descritiva para a exploração dos dados que pode ajudar na identificação de relacionamentos entre valores ou transações em um banco de dados. As duas abordagens mais comuns são: a descoberta de associações e a descoberta de seqüências. A descoberta de associação procura regras sobre itens que ocorrem juntos em um mesmo evento assim como uma transação de venda. A descoberta de seqüência é muito similar,

mas se trata de uma associação relacionada com o tempo. (TWO CROWS CORPORATION, 1999)

Um exemplo típico de regra de associação é análise da cesta de mercado (*market basket analysis*). Este processo analisa os hábitos de compra dos clientes para procurar associações entre os diferentes itens que os clientes costumam colocar em suas cestas de compras. A descoberta de associações neste contexto pode ajudar a desenvolver estratégias de marketing. (HAN e KAMBER, 1999)

Associações podem ser descritas como $A \rightarrow B$, onde A é chamado de antecedente ou lado esquerdo e B é chamado de conseqüente ou lado direito. A e B são conjuntos de itens e a intersecção entre eles é um conjunto vazio. É relativamente fácil determinar a proporção de transações que um item em particular ou conjunto de item: basta contá-los. (TWO CROWS CORPORATION, 1999)

A freqüência com que uma associação em particular pode ocorrer no banco de dados é chamada de *fator de suporte*. Se existem 15 transações de um item de um total de 1000 transações, então indica que o suporte à associação é de 1,5%. Um nível baixo de suporte pode indicar que a associação em particular não é muito importante ou pode indicar a presença de dados ruins. (TWO CROWS CORPORATION, 1999)

Para descobrir regras significantes, é preciso olhar a *freqüência relativa* de ocorrências do item e suas combinações. Dada a ocorrência do item A (o antecedente), como B (o conseqüente) ocorre? Isto é, qual a condição para haver uma possibilidade de B ocorrer, dado A? Outro termo para esta condição é o *fator de confiança*. A confiança é calculada por uma razão: (freqüência de A e B) / (freqüência de A). (TWO CROWS CORPORATION, 1999)

2.1.1.3 Segmentação e Cluster Analysis

O objetivo da análise de segmentação é agrupar os dados de forma que os grupos formados sejam muito diferentes entre si mas que os membros destes grupos sejam semelhantes. Diferentemente da classificação, não é possível saber quais características os conjuntos de dados segmentados terão quando o processo for iniciado, ou como serão os atributos dos dados após serem segmentados. Em consequência disto, será preciso que alguém que conheça o negócio interprete os segmentos gerados. (TWO CROWS CORPORATION, 1999)

É importante salientar que não se pode confundir análise de segmentação com *clustering*. Segmentação se refere ao problema geral de identificar grupos que tenham características comuns. *Clustering* é somente uma maneira de segmentar os dados em grupos que não estão previamente definidos, enquanto que classificação é uma maneira que segmenta os dados determinando grupos que já são conhecidos. (TWO CROWS CORPORATION, 1999)

Segundo Han e Kamber (1999), o processo de agrupar um conjunto físico ou abstrato de objetos em classes de objetos similares é chamado de *clustering*. Um *cluster* é uma coleção de objetos que são similares entre si e não similares aos objetos de outros *clusters*.

Como um ramo da estatística, a análise de cluster tem sido estudada extensivamente por muitos anos, concentrando-se principalmente na análise baseada em medidas de distâncias. Ferramentas de *Cluster Analysis* são baseadas em métodos como *k-means*, *k-medoids* e vários outros que têm sido também

construídos e disponibilizados em pacotes ou sistemas de softwares de análise estatística, como SAS e SPSS.

Em aprendizado de máquina, o *Cluster Analysis* muitas vezes se refere ao aprendizado não supervisionado. Diferentemente da classificação, a clusterização não conta com classes predefinidas e conjuntos de treinamento. Por esta razão, é uma forma de aprendizado por observação, particularmente chamado de aprendizado por exemplo.

Em DM, muitos métodos têm sido estudados para tornar o *Cluster Analysis* eficiente em grandes bases de dados. A clusterização é um campo de pesquisa desafiador onde aplicações em potencial propõem requerimentos especiais. Para o DM são consideradas necessidades essenciais para a utilização de *Cluster Analysis* em grandes bases de dados:

- **Escalabilidade:** muitos algoritmos trabalham bem com pequenas quantidades de dados. Agrupar uma amostra de um conjunto que pode conter milhões de registros pode levar a resultados falsos. Desta forma é preciso desenvolver algoritmos que possam ser altamente escaláveis.
- **Habilidade para trabalhar com diferentes tipos de dados:** muitos algoritmos são construídos para clusterizar intervalos numéricos de dados. Entretanto, algumas aplicações requerem clusterizar outros tipos de dados, como valores binários, nominais, ordinais ou uma mistura destes tipos de dados.
- **Identificar clusters como formas arbitrárias:** como muitos algoritmos se baseiam em medidas de distâncias Euclidianas⁴ ou Manhattan⁵, estes tendem a encontrar clusters esféricos com tamanho e forma similares.

⁴ Medidas euclidianas são aquelas a que estamos acostumados, com três dimensões (altura, largura e profundidade) e onde a distância mais curta entre dois pontos é um caminho reto.

⁵ Métrica de Manhattan, a distância total percorrida entre dois pontos.

Entretanto, um cluster pode ser de muitas outras formas. É importante desenvolver algoritmos que possam detectar clusters com formas arbitrárias.

- **Domínio mínimo sobre como determinar parâmetros de entrada:** muitos algoritmos requerem que os usuários determinem os parâmetros, como o número de clusters necessários. No entanto, os resultados são muitas vezes sensíveis aos parâmetros de entrada. Muitos parâmetros são difíceis de determinar, especialmente para conjuntos de dados que contêm objetos com muitas dimensões. Isto pode sobrecarregar o usuário, mas também pode afetar a qualidade por dificultar o controle do processo.
- **Habilidade para trabalhar com dados ruins:** no mundo real, os bancos de dados contêm valores discrepantes ou vazios, desconhecidos ou errados. Muitos algoritmos são sensíveis para dados assim e podem levar a um resultado com pouca qualidade.
- **Não sensível quanto à ordem de entrada dos registros:** muitos algoritmos são sensíveis quanto à ordem de entrada dos registros. Conjuntos de dados quando apresentados com diferentes ordenações para um algoritmo, podem gerar diferentes clusters.
- **Muitas dimensões:** um banco de dados ou DW pode conter várias dimensões e atributos. Muitos algoritmos são bons para tratar com poucas dimensões, envolvendo somente duas ou três. Os olhos humanos são bons para julgar a qualidade de um cluster para até três dimensões.
- **Clusterização baseado em restrições:** as aplicações precisam trabalhar em cima de vários tipos de restrições.

- **Interpretabilidade e usabilidade:** os usuários esperam que os resultados da clusterização possam ser interpretáveis, compreensíveis e usáveis. É preciso levar em consideração interpretações e aplicações semânticas específicas, estudar como o objetivo de uma aplicação pode influenciar na seleção de métodos de clusterização.

2.1.2 Tarefas Preditivas

Nas tarefas preditivas, são construídos modelos chamados *modelos preditivos*, onde as variáveis ou classes serão chamadas de *variável resposta*, *dependente* ou *alvo*. O valor usado para fazer a predição é chamado de *preditor* ou *variável independente*.

Os modelos preditivos são construídos, ou *treinados*, usando dados para o qual o valor da variável de resposta já é conhecido. Este treinamento é algumas vezes referenciado como *aprendizado supervisionado*, porque os valores calculados ou estimados são comparados com os resultados conhecidos. Em contraste, técnicas de descrição como um cluster, são ditas como *aprendizado não supervisionado* porque não conhecem os resultados para orientar o algoritmo. Podemos identificar duas principais tarefas preditivas: classificação e regressão ou predição, onde a classificação é usada para prever valores discretos ou nominais, enquanto regressão ou predição é usada para prever valores ordinais ou contínuos. Normalmente, se refere à predição de variáveis nominais como classificação e à predição de valores contínuos como predição.

2.1.2.1 Classificação

Problemas de classificação servem para identificar as características que indiquem o grupo para o qual cada caso pertence. Este padrão pode ser usado tanto para entender o dado existente quanto para prever como uma nova instância se comportará. DM cria modelos de classificação para examinar dados já classificados (casos) e através de indução encontrar um padrão existente.

A classificação é um processo de dois passos. No primeiro passo, um modelo é construído descrevendo um predeterminado conjunto de classes ou conceitos. O modelo é construído através da análise dos registros descritos por seus atributos. Cada registro é assumido como pertencente a uma classe pré-definida, como determinado por um de seus atributos, chamado de *atributo alvo*. No contexto da classificação, os registros são também chamados como uma amostra, exemplo ou objetos. Os registros analisados para construir o modelo formam coletivamente o conjunto de dados de treinamento. Os registros individuais utilizados para treinar o modelo são definidos como amostras de treinamento e são randomicamente selecionados para gerar a *população amostral*. Visto que a variável alvo (ou resultado) para cada amostra é também fornecido, este passo é também conhecido como *aprendizado supervisionado*. Tipicamente, o modelo treinado é representado na forma de regras de classificação, árvores de decisão, ou fórmulas matemáticas.

No segundo passo, o modelo é usado para a classificação. Em primeiro lugar, a acuracidade do classificador deve ser estimada. No item 2.1.4 serão descritos os principais métodos para estimar a acuracidade do classificador, sendo que o mais utilizado é o método **holdout**, uma técnica simples que usa um conjunto de teste

para avaliar as amostras. Os elementos deste conjunto de teste são randomicamente selecionados e são independentes do conjunto de treinamento.

A acuracidade do modelo em um determinado conjunto de teste é a porcentagem de amostras de teste que são corretamente classificadas pelo modelo. Para cada amostra de teste, a variável-alvo conhecida é comparada com a classe prevista pelo modelo classificador. Note que se a acuracidade estimada do modelo for estimada com base no conjunto de dados de treinamento, a estimativa pode ser superavaliada simplesmente porque o modelo treinado pode gerar o processo de “**overfitting**”⁶, isto é, talvez possa ter incorporado alguma anomalia particular do dado treinado que não está presente em toda a população da amostra.

Se a acuracidade do modelo é considerada aceitável, o modelo treinado pode ser usado para classificar futuras linhas de dados para o qual a variável-alvo não é conhecida.

2.1.2.2 Regressão

A regressão utiliza valores existentes para prever outro valor. Em casos simples, a regressão usa técnicas estatísticas padrão como regressão linear. Infelizmente, muitos problemas do mundo real não são simples projeções de valores existentes. Para exemplificar, volumes de vendas e preços são muito difíceis de prever porque dependem de muitas interações ou de múltiplas variáveis preditivas. De qualquer modo, técnicas como, por exemplo, regressão logística, árvores de decisão ou redes neurais podem ser usadas para prever valores futuros. Se o valor

⁶ Significa que o modelo consegue diagnosticar corretamente quase todos os casos novos idênticos aos usados para gerá-lo, porém, é impreciso para diagnosticar novos casos desconhecidos.

a ser previsto for uma variável do tipo tempo, poderá ser uma predição de série temporal.

2.1.3 Comparando métodos de classificação

Os métodos de classificação e predição podem ser comparados e avaliados de acordo com os seguintes critérios:

- **Acuracidade preditiva:** refere-se à habilidade do modelo predizer corretamente o nome da classe para valores desconhecidos;
- **Velocidade:** envolve o custo computacional para a geração e uso do modelo;
- **Robustez:** habilidade do modelo de fazer predições com dados de baixa qualidade ou dados com valores faltantes;
- **Escalabilidade:** habilidade do modelo de aprendizado de trabalhar eficientemente com grandes quantidades de dados.
- **Interpretabilidade:** refere-se ao nível de compreensão e percepção que é fornecida pelo modelo.

2.1.4 Acuracidade do Classificador

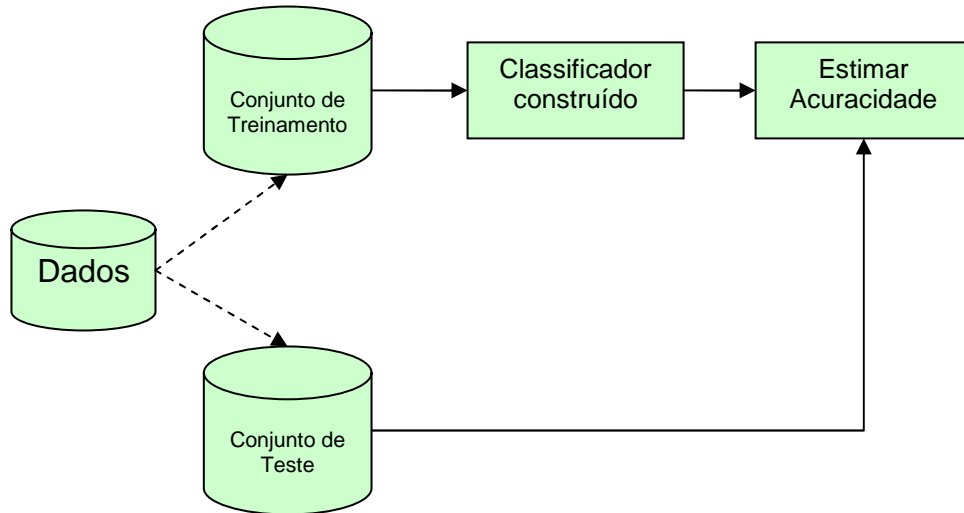


Figura 2.1 – Processo de estimar a acuracidade com o método holdout (HAN e KEMBER, 1999)

O principal objetivo a ser atingido pelos modelos classificadores está na sua capacidade de diagnosticar corretamente casos nunca vistos, sendo que assim podemos definir a sua acuracidade em quão bem ele representa a realidade do problema. Existem várias técnicas de estimar a performance de um modelo treinado, sendo que algumas podem ser melhores que outras. A taxa de erro é a mais utilizada para medir a performance de um classificador e pode ser representada através da equação 2.1:

$$\text{taxa de erro} = \frac{\text{número de erros}}{\text{número de casos}} \quad (2.1)$$

Usar somente dados de treinamento para produzir um classificador e então estimar a sua acuracidade com este mesmo conjunto de dados pode resultar em uma estimativa ilusória ou superotimista devido à superespecialização do algoritmo de aprendizado; pode ocorrer o processo de “overfitting”. Durante o processo de treinamento, existe um esforço muito grande do sistema de aprendizado para “acomodar” os exemplos de treinamento, e tal esforço pode levar a uma extração de regras poucos genéricas para novos casos.

Os métodos holdout, random subsampling e validação cruzada são as técnicas mais comuns para avaliar a acuracidade do classificador, baseadas em partes de amostras randômicas de um determinado conjunto de exemplos:

- **Método Holdout ou Método H (Figura 2.1):** os dados são randomicamente particionados em dois conjuntos independentes, um conjunto de treinamento e um conjunto de teste. Tipicamente, dois terços dos dados são alocados para um conjunto de treinamento e outro terço é alocado para teste. O conjunto de treinamento é usado para produzir o classificador e sua acuracidade é estimada com o conjunto de teste. Este método é o mais simples de estimar o erro de um modelo classificador e normalmente apresenta ótimos resultados, entretanto uma única partição pode gerar resultados imprecisos, principalmente para pequenas e médias amostras de exemplos, tanto para treinamento quanto para teste.
- **Random subsampling** (subamostras randômicas): é uma variação do método *holdout* em que o processo de treinamento e avaliação é repetido várias vezes. O método *random subsampling* soluciona o problema típico do método *holdout*, a escolha de uma partição dos exemplos não é representativa aos conceitos. O problema é contornado pois são criadas

várias partições e o sistema é treinado e testado a cada iteração com uma partição diferente. No caso de uma partição ser pouco representativa, tal problema pode ser amenizado pois a taxa de erro final é calculada através das médias de erro de todas as partições.

- **Cross validation:** também conhecido por *k-fold cross-validation*. Os dados iniciais são particionados em k mutuamente exclusivos conjuntos de dados e com tamanhos aproximados (S_1, S_2, \dots, S_k) . O treinamento e o teste são feitos k vezes. Na interação i , o conjunto S_i é reservado como um conjunto de teste, e os conjuntos restantes são coletivamente usados para treinar o classificador. Isto é, o classificador da primeira interação é treinado com os subconjuntos S_2, \dots, S_k e testado em S_1 ; o classificador da segunda interação é treinado com os subconjuntos S_1, S_3, \dots, S_k e testado com S_2 , e assim por diante. A acuracidade estimada é o número geral de classificações corretas das k interações dividida pelo número total de amostras dos dados iniciais.

O uso de várias técnicas para estimar a acuracidade do classificador incrementa o tempo total de processamento, mas será útil para selecionar o melhor entre os muitos classificadores gerados.

2.2 Árvore de Decisão

As técnicas de DM são grupos de soluções ou algoritmos que são usados para responder aos problemas propostos nas tarefas. Cada tarefa apresenta várias técnicas que podem também ser utilizada para solucionar tarefas diferentes. A escolha destas técnicas precisa levar em consideração o objetivo final da DCBD.

A mineração de dados possui não somente um amplo espectro de aplicações, como também de técnicas, algoritmos e procedimentos. Diversas áreas, apresentadas a seguir, estão envolvidas para o desenvolvimento de algoritmos de DM:

- **Redes Neurais:** é uma técnica computacional que constrói um modelo matemático, emulado por computador, de um sistema neural biológico simplificado, com capacidade de aprendizado, generalização, associação e abstração.(ICA, 1999)
- **Algoritmos Genéticos:** são modelos estocásticos⁷ e probabilísticos de busca e otimização, inspirados na evolução natural e na genética, aplicados a problemas complexos de otimização. Têm sido empregados em DM para as tarefas de classificação e descrição de registros, além da seleção de atributos que melhor caracterizem o objetivo da tarefa de KDD. (ICA, 1999)
- **Métodos Estatísticos:** existem diversos métodos estatísticos, sendo alguns clássicos (regressão linear e múltipla, clusterização, etc.) e outros mais recentes, que assumem a existência de uma variável (atributo) resposta y , e uma coleção de variáveis preditoras x , além da

⁷ O oposto de determinístico. Ao invés de assumir que os seus dados assumem um determinado valor, você assume que esses dados possuem uma determinada distribuição probabilística.

disponibilidade de dados para treinamento. Entre as técnicas estatísticas podemos citar as Redes Bayesianas e Árvores de Decisão. (ICA, 1999)

A ênfase deste trabalho será para a técnica de árvore de decisão. Os motivos que levaram a esta escolha estão fundamentados na escolha da ferramenta de DM integrada a um SGBD comercial (MS SQL Server) e também nos objetivos propostos no capítulo 1 desta monografia.

Uma árvore de decisão é uma estrutura em forma de árvore, onde cada nó interno ou nó-decisão representa um teste sobre o valor de um atributo do registro, cada ramo representa um resultado do teste, cada aresta que sai de um nó-decisão até um de seus nós filhos representa um dos possíveis resultados dos valores dos atributos de cada nó e os nós-folha representam classes ou distribuição de classes (valor do atributo objetivo). O nó superior em uma árvore é o nó-raiz. (HAN e KAMBER, 1999).

Uma árvore de decisão típica é mostrada na Figura 2.2 e representa o conceito da venda de computadores, isto é, prediz quem será ou não um cliente potencial de uma loja de venda de computadores. Os nós internos são representados por retângulos e os nós-folhas são representadas por círculos (HAN e KAMBER, 1999).

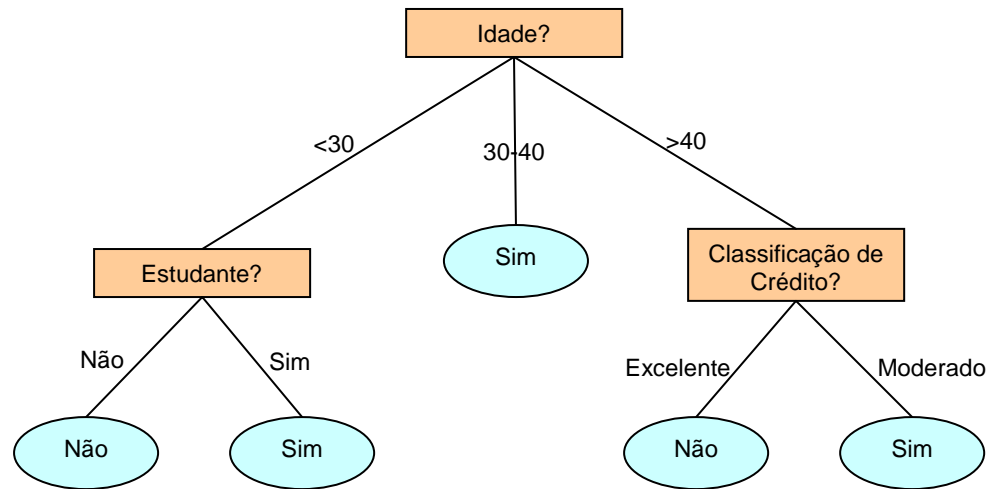


Figura 2.2 – Árvore de Decisão que representa o cliente que comprará ou não um computador

O próximo passo é classificar um exemplo desconhecido: os valores dos atributos são testados frente à árvore de decisão, um caminho é traçado do nó principal para os nós-filhos os quais contém a classe de predição para aquele exemplo. Árvores de decisão são facilmente convertidas para regras de classificação. (HAN e KAMBER, 1999)

2.2.1 Indução de Árvores de Decisão

O processo para construir a estrutura de uma árvore de decisão a partir de dados é conhecido como indução de árvores de decisão. Uma árvore de decisão pode ser induzida selecionando algum atributo inicial, dividindo os casos segundo este atributo em conjuntos disjuntos, e então repetindo este procedimento para todos os nós subsequentes. Estes se tornam terminais e portanto não são mais

divididos quando todos os exemplos deste nó pertencem à mesma classe. (BATISTA, 1997)

O algoritmo básico para indução de árvores de decisão é um algoritmo que constrói árvores de forma recursiva, de cima para baixo, utilizando um algoritmo baseado na aproximação “dividir para conquistar”. (HAN e KAMBER, 1999).

Na Figura 2.3 está sumarizado uma versão do ID3, um algoritmo bastante conhecido.

Entrada: os exemplos para treinamento, representados por atributos com valores discretos; o conjunto de atributos candidatos, lista de atributos.

Saída: Uma árvore de decisão.

Método:

- 1) criar um nó N;
- 2) **if** os exemplos são todos de classes similares, C **then**
- 3) retorna N como um nó folha rotulado com a classe C;
- 4) **if** a lista de atributos está vazia **then**
- 5) retorna N como um nó-folha rotulado com a classe mais comum dos exemplos; // com a maioria da votação
- 6) seleciona atributo-teste, o atributo dentro da *lista de atributos* como o maior ganho de informação;
- 7) rotula o nó-N com o *atributo-teste*;
- 8) **foreach** valor conhecido a_i do *atributo-teste* // *partição de amostras*
- 9) inserir um ramo do nó N para a condição *atributo-teste* = a_i ;
- 10) S_i passa a ser o conjunto de amostras das *amostras* onde o *atributo-teste* = a_i ; // *uma partição*
- 11) **if** S_i está vazio **then**
- 12) inserir uma folha rotulada com a classe mais comum das amostras;
- 13) **else** inserir o nó retornado pelo ID3(S_i , lista de atributos – *atributo-teste*);

Figura 2.3 – Algoritmo para geração de uma árvore de decisão (HAN e KAMBER, 1999)

A estratégia do algoritmo:

- A árvore começa com um simples nó representando os registros das amostras de treinamento (passo 1); sempre que este procedimento for executado, esse nó será o nó corrente.

- Se a as amostras são todas de uma mesma classe, então o nó se torna uma folha e é rotulado com aquela classe (valor do atributo alvo) (passos 2 e 3);
- Se a lista de atributos está vazia, retorna a raiz o valor da classe mais comum no conjunto de amostras (analisa se ainda existem atributos não utilizados) (passo 4).
- De outra forma, o algoritmo usa uma medida baseada na entropia conhecida como *ganho de informação*, como uma heurística⁸, para selecionar o atributo que irá separar da melhor maneira possível às amostras em classes individuais (passo 6). Este atributo se torna o atributo "teste" ou "decisão" do nó (passo 7). Nesta versão do algoritmo ID3, todos os atributos são categóricos, por exemplo, valores discretos. Atributos com valores contínuos precisam ser categorizados.
- Uma divisão é criada para cada valor conhecido do atributo teste (cria-se os *links* do nó raiz com todos os possíveis valores do *atributo-teste* para outros nós), e as amostras são conseqüentemente particionadas em novos conjuntos onde *atributo-teste* = a_i (passos 8-10);
- O algoritmo usa muitos processos recursivos para formar uma árvore de decisão para os exemplos em cada partição. Uma vez que um atributo tenha ocorrido em um nó, ele não será considerado em qualquer nó descendente (passo 13).
- O particionamento recursivo irá parar somente quando qualquer uma das seguintes condições é verdadeira:

⁸ Heurístico é o oposto de algorítmico. Significa que não tem como garantir que o resultado é correto, mas que pode provar alguma propriedade desejável sobre o resultado.

- Todos os exemplos de um dado nó pertencem a uma mesma classe (passos 2 e 3) ou
- Não existem mais atributos nos quais os exemplos podem ser particionados (passo 4). Neste caso, a seleção pela classe mais comum é empregada (passo 5). Isto envolve converter o nó em uma folha e rotular como a classe mais comum das amostras. Alternativamente, a classe de distribuição do nó da amostra pode ser armazenada, ou
- Não existem mais amostras para o ramo *atributo-teste* = a_i (passo 11). Neste caso, a folha é criada com a classe dos exemplos mais representativa (passo 12).

As principais vantagens dos algoritmos baseados em árvores de decisão são: sua eficiência computacional e simplicidade, mas devido ao uso da aproximação “dividir para conquistar”, também possuem desvantagens. Por exemplo, uma condição envolvendo um atributo que será incluído em todas as regras. Essa situação produz regras com informações irrelevantes, além de desperdício de processamento. (ICA, 1999)

2.2.1.1 Seleção de Atributos:

As funções utilizadas para avaliar os nós procuram reduzir o grau de aleatoriedade ou “impureza” do nó corrente. Por exemplo, durante a indução da árvore mostrada na Figura 2.2, gerada a partir dos registros conforme a Tabela 2.1,

o sistema de aprendizado inicia analisando o nó raiz. Neste nó, se considerarmos que temos 5 casos, 3 casos de possíveis compradores e 2 não, vê que se trata de um nó muito impuro e portanto não pode ser terminal. A tarefa é encontrar algum teste que possa dividir este nó raiz de tal forma que se obtenha o melhor resultado. Se for encontrado algum atributo que divida o nó em dois grupos de tal forma que cada um das classes fique em ramos diferentes, então a aleatoriedade ou impureza pode ser reduzida para 0.

A medida de *ganho de informação* é usada para selecionar e testar atributos em cada nó da árvore. O atributo com maior ganho de informação (ou com maior redução entrópica) é escolhido como um atributo teste para o nó corrente. Este atributo minimiza a informação necessária para classificar as amostras em partições resultantes.

A definição do melhor atributo segue o seguinte critério (ICA, 1999) (HAN e KAMBER, 1999):

- a) Seja S todas os registros do conjunto de treinamento, A um atributo, s um registro, v um valor, e c o número de classes (valores distintos da variável alvo), defini-se:

$$S_v = \{s \in S \mid A(s) = v\} \quad (2.2)$$

- b) A equação 2.2, representa o conjunto de todos os registros do conjunto de treinamento que possuem no atributo A , o valor v . Defini-se a entropia de S , como:

$$Entropia(S) = -\sum_{i=1}^c p_i \log_2(p_i) \quad (2.3)$$

Onde p_i é a probabilidade de ocorrência de uma determinada classe.

- c) Desse modo pode-se definir o ganho da escolha de um atributo **A** com respeito a **S** como:

$$Ganho(S, A) = Entropia(S) - \sum_{v \in \text{valores}(A)} (|S_v| / |S|) * Entropia(S_v) \quad (2.4)$$

Podemos ainda utilizar a equação 2.5 para calcular a impureza total de uma divisão:

$$Ti(n) = \sum_k p_k * i(n_k) \quad (2.5)$$

Onde:

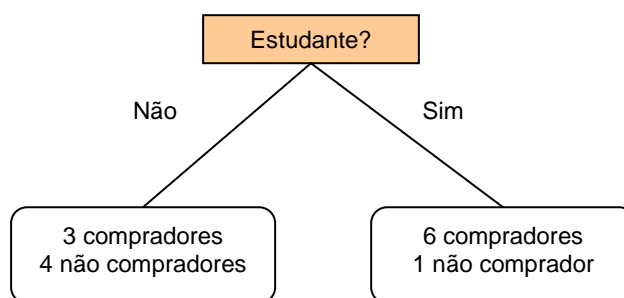
- Ti é impureza total de certa divisão;
- n é o nó que está sendo dividido;
- p_k são as probabilidades de cada ramo criado pela inclusão do atributo;
- $i(n_k)$ são as impurezas dos nós ligados aos novos ramos gerados.

No exemplo de indução da Figura 2.2 construída conforme a Tabela 2.1, deve-se selecionar algum atributo para ser o nó raiz da árvore, segundo uma função de avaliação. Para este exemplo será utilizada a função de entropia. O nó raiz inicialmente possui 14 casos, com 9 casos na classe “comprador” e 5 na classe “não comprador”. O nó raiz possui exemplos de ambas as classes e se deve procurar por algum atributo que divida os exemplos em ramos com o mínimo de impureza possível.

Tabela 2.1 – Registros de possíveis compradores

Caso	<i>Idade</i>	<i>rendimento</i>	<i>Estudante</i>	<i>Classificação de crédito</i>	Classe Comprador?
1	<30	Alto	Não	Moderado	Não
2	<30	Alto	Não	Excelente	Não
3	30-40	Alto	Não	Moderado	Sim
4	>40	Médio	Não	Moderado	Sim
5	>40	Baixo	Sim	Moderado	Sim
6	>40	Baixo	Sim	Excelente	Não
7	30-40	Baixo	Sim	Excelente	Sim
8	<30	Médio	Não	Moderado	Não
9	<30	Baixo	Sim	Moderado	Sim
10	>40	Médio	Sim	Moderado	Sim
11	<30	Médio	Sim	Excelente	Sim
12	30-40	Médio	Não	Excelente	Sim
13	30-40	Alto	Sim	Moderado	Sim
14	>40	Médio	Não	Excelente	Não

Assim, ao avaliarmos os atributos que forneçam a maior redução de impurezas, temos *idade*, *rendimento*, “*estudante?*” e *classificação de crédito*. Se considerarmos primeiramente “*estudante?*” a situação seria conforme a Figura 2.4.

Figura 2.4 – Árvore de Decisão que representa o atributo *estudante*

A impureza total para a divisão através do atributo “*estudante?*” aplicando a equação 2.5 seria:

Tabela 2.2 – Entropia total do atributo “*estudante?*”

Atributo Estudante?	Classe alvo		Total	Entropia da folha
	Sim	Não		
Sim	6	1	7	0,5917
Não	3	4	7	0,9852
			14	

Desta forma para “estudante?”,

$$Ti(\text{raiz}) = 7/14 * 0,9852 + 7/14 * 0,5917 = \mathbf{0,7885};$$

Aplicando aos demais atributos:

Tabela 2.3 – Entropia total do atributo idade

Atributo	Classe alvo		Total	Entropia da folha
	Sim	Não		
Idade				
<30	2	3	5	0,9710
30-40	4	0	4	0
>40	3	2	5	0,9710
TOTAL	EQUAÇÃO (2.5)		14	0,6936

Tabela 2.4 – Entropia total do atributo classe de crédito

Atributo	Classe alvo		Total	Entropia da folha
	Sim	Não		
Classe de Crédito				
Moderado	6	2	8	0,8113
Excelente	3	3	6	1
TOTAL	EQUAÇÃO (2.5)		14	0,8922

Tabela 2.5 – Entropia total do atributo rendimento

Atributo	Classe alvo		Total	Entropia da folha
	Sim	Não		
Rendimento				
Alto	2	2	4	1
Médio	4	2	6	0,9183
Baixo	3	1	4	0,8113
TOTAL	EQUAÇÃO (2.5)		14	0,9111

Tabela 2.6 – Entropia total na raiz principal

	casos	Total	- (casos/ Total)	log₂	- (casos/ Total)*log₂
Sim	5	14	(0,3571)	(1,4854)	0,5305
Não	9	14	(0,6429)	(0,6374)	0,4098
			EQUAÇÃO 2.3	Total	0,9403

Conforme foi observado nas tabelas acima, considerando que a entropia total na raiz da árvore seja dada pela Tabela 2.6, podemos afirmar que o ganho de cada atributo para o nó raiz foi: ganho (Idade) = 0,246, ganho ("estudante?") = 0,151, ganho (Rendimento) = 0,029, ganho (Classe de crédito) = 0,048. Assim, idade tem o maior ganho de informação entre todos os atributos, e será selecionada como o atributo teste.

Através da seleção de atributos que fornecem a divisão de amostras com menor impureza ou com maior ganho de informação, é possível induzir uma árvore mais simples e, provavelmente, mais preditiva na classificação. Entretanto, muitas vezes esta abordagem não é suficiente e melhores resultados podem ser alcançados se ramos considerados "fracos", ou seja que contribuem muito pouco no processo de classificação, fossem cortados da árvore. Este processo é chamado de *poda de árvores de decisão*.

2.2.1.2 Poda da Árvore de Decisão

Após a construção de uma árvore de decisão, muitos de seus ramos poderão refletir anomalias existentes nos dados de treinamento devido a ruídos. Os métodos de poda resolvem este problema chamado de *overfitting*. Métodos estatísticos são normalmente usados para remover de forma confiável estes ramos, resultando assim em uma classificação mais rápida e no aprimoramento da habilidade de classificar corretamente dados de teste.

Existem duas maneiras mais comuns de realizar a poda de uma árvore:

- Na *pré-poda*, uma árvore é “*podada*” durante a sua construção através de paradas, como, por exemplo, decidindo em não expandir um nó de uma árvore. Nesta parada o nó se torna uma folha e pode conter a classe mais freqüente entre o subconjunto de amostras ou a probabilidade de distribuição das amostras. Na construção de uma árvore, medidas como significância estatística, qui-quadrado e ganho de informação (Equação 2.4), podem ser usadas para assegurar a qualidade de uma divisão. Pode-se empregar um valor pré-especificado chamado de *threshold*⁹ onde, no momento de uma divisão, verifica se continuará a expandir o nó. Há muitas dificuldades nesta abordagem, principalmente na definição do *threshold*. Valores altos resultam em uma simplificação em demasia da árvore, enquanto valores baixos resultam em pouca simplificação.
- O método de *pós-poda* remove os ramos de uma árvore de decisão após ela ter sido totalmente construída. O procedimento pode ser descrito da seguinte forma: para cada nó não-folha da árvore o algoritmo calcula a taxa de erro esperada que irá ocorrer se a sub-árvore do nó for podada. Em seguida, é calculado o erro esperado se o nó não for podado através das taxas de erro para cada ramo, combinado de acordo com a proporção de observações ao longo das ramificações. Se podando o nó pai um grande erro é esperado, então a sub-árvore continua. De outra forma, a sub-árvore será podada. Após consecutivas podas, é realizado um teste independente para estimar a acuracidade de cada árvore. A Árvore de decisão que minimiza a estimativa de erro é selecionada.

⁹ Diz-se de um valor limiar, fronteira.

Alternativamente, em comparação com podas baseadas na expectativa de taxa de erros, pode-se podar uma árvore com base no número de bits necessários para que sejam escritas em código. O “*best pruned tree*” é um método que minimiza o número de bits codificados, e consiste em adotar o princípio do *Tamanho de Descrição Mínima* (MDL - Minimum Description Length) no qual se baseia no conceito de que a solução mais simples é a melhor escolha.

2.2.1.3 Algoritmo C4.5

É a versão aprimorada do ID3 que introduziu várias propriedades novas, sendo as principais:

- a) O processo de pós-poda é introduzido no algoritmo: após a árvore de decisão ter sido construída, o C4.5 examina recursivamente cada sub-árvore para determinar se substituindo a mesma por uma folha ou um ramo será benéfico para o desempenho da árvore.
- b) Aceita valores contínuos: são definidos os *thresholds* para os testes de atributos contínuos (os limites superior e inferior para cada teste são calculados). Digamos que o atributo C_i tem uma faixa contínua de valores. São examinados os valores para este atributo em um conjunto de treinamento. Digamos que todos são, em ordem crescente:

$$C_i = \{A_1, A_2, \dots, A_m\}$$

Então para cada valor A_j , $j=1,2,..m$, são particionados aqueles registros em C_i que tem o valor menor ou igual a A_j , e aqueles que têm valor maior que A_j . Para cada uma destas partições, são computados os ganhos e escolhida a partição que maximiza o ganho. Este método envolve um considerável custo computacional.

- c) Tratamento de dados com valores nulos, faltantes ou desconhecidos;
- d) Melhora da eficiência computacional.
- e) Derivação de regras.

CAPÍTULO 3 ESPECIFICAÇÃO DE PROVEDORES DE RECURSOS DE DM

Segundo Curotto (2003), vários projetos foram desenvolvidos para a integração da tecnologia de DM com SGBD's, podendo destacar:

- Projeto Quest: projeto desenvolvido pelo centro de pesquisas da IBM®;
- Projeto dbMiner: iniciado por um grupo de desenvolvedores da Universidade Simon Fraser de Burnaby, British Columbia, Canadá que desenvolveu uma série de trabalhos de integração de técnicas de DM com OLAP. Após várias contribuições, este projeto evoluiu em 2002 para um produto comercial;
- Tecnologia OLE DB DM: com a colaboração de vários pesquisadores foi, elaborado e publicado, em Julho de 2000, a especificação da tecnologia OLE DB DM, cuja especificação define um padrão industrial para DM tal que diferentes algoritmos de DM implementados por diversos desenvolvedores possam ser facilmente embutidos em aplicativos de usuários, especificando a API (*Application Programming Interface*) entre consumidores de DM (aplicativos que utilizam recursos de DM) e provedores de recursos de DM (pacotes de *software* que fornecem algoritmos de DM). Em setembro de 2000, o MSSQL 2000, foi lançado com componente importante: o módulo de Serviços de Análise. Incluso, o primeiro provedor de recursos de DM, com dois algoritmos implementados: uma para problemas de classificação através de árvores de decisão e outro para problemas de clusterização.

3.1 Tecnologia OLE DB DM

Há um grande interesse comercial em minerar informações de DW, mas construir aplicações de DM para bancos de dados relacionais não é uma tarefa fácil e requer um trabalho significativo. Neste sentido, foi desenvolvido a API “OLE DB for Data Mining” (OLE DB DM) que é uma API padrão para o desenvolvimento de recursos de DM, que possibilita a portabilidade destes provedores que podem aproveitar os recursos dos SGBD’s. (NETZ *et al*, 2001)

Curotto (2003) salienta que apesar de ter sido desenvolvido pela Microsoft®, este padrão de API é totalmente independente de qualquer fabricante, de qualquer *software* ou de qualquer provedor de *software* e do ponto de vista do pesquisador esta tecnologia é altamente promissora, já que possibilita com um mínimo de esforço, portar ou desenvolver algoritmos de DM e *interfaces* de visualização de resultados em linguagens não proprietária, para que se torne um provedor de recursos de DM pronto para uso integrado com um SGBD’s.

3.1.1. Motivações

- a. É comum trabalhar com dados armazenados em arquivos, o que requer um ambiente separado de SGBD’s. Os dados ou amostras são tirados de suas bases e uma série de linguagens é usada para prepará-las. O problema é gerenciar estes dados onde podem ocorrer problemas como consistência com os dados já existentes nas bases (de um DW, por exemplo);

- b. Outro problema gerado ao trabalhar com aplicações isoladas é responder a questões do tipo: como o modelo gerado é armazenado, mantido e atualizado? Como utilizá-lo em outros conjuntos de dados e como são visualizados?
- c. Desenvolvedores de software de aplicações empresariais típicas dificilmente são experientes em estatística e reconhecimento de padrões. Portanto, desenvolver aplicações de DM é caro e exige tempo.
- d. Assegurar que modelos e operações de DM ganhem o status de “*objetos de primeira classe*” no objetivo final no ambiente de desenvolvimento de bancos de dados.

3.1.2 Propostas

- a. Amenizar os problemas de desenvolvimento de modelos e facilitar a preparação de dados por trabalhar diretamente em dados relacionais.
- b. Permitir que desenvolvedores de aplicações participem na construção de soluções de DM. Possibilitando desenvolver soluções integradas que são críticas para o crescimento da tecnologia de DM no espaço empresarial. O DM precisa ser visto como um componente que agrega valor junto com as tradicionais técnicas de suporte à decisão como o SQL tradicional e o ambiente de consultas OLAP.
- c. Possibilitar que desenvolvedores possam se sentir confortáveis uma vez que é comum usar bancos de dados com ferramentas de API baseadas

em SQL, OLE DB ¹⁰ e outros padrões conhecidos de protocolos. Desta forma é importante construir em cima da arquitetura OLE DB, uma API uniforme que consiga se tornar popular não somente para acesso de sistemas relacionais mas também a outras fontes de dados que possam serem vistas como um “conjunto de tabelas” também.

- d. Esta arquitetura não pode ser especializada em nenhum modelo específico de mineração mas sim estruturada para atender a todos os modelos de mineração conhecidos.

3.1.3 Filosofia Básica do OLE DB DM

NETZ *et al* (2001), descreveram a filosofia básica e as decisões de projeto que culminaram com a especificação OLE DB DM. O desafio seria de uma especificação capaz de suportar operações básicas de DM sem introduzir muitas mudanças no modelo de programação e no ambiente que um desenvolvedor de bancos de dados está acostumado a usar. Fundamentalmente, são necessárias operações que suportem modelos de DM. Para isto se formulou quatro operações fundamentais que devem ser suportadas por um provedor de recursos de DM:

- **Definir** um modelo de DM, identificando por exemplo o conjunto de atributos de dados a serem preditos, o conjunto de atributos de dados a serem utilizados para predição e o algoritmo utilizado para construir o modelo de DM;

¹⁰ OLE DB consiste de uma especificação orientada a objeto para que um conjunto de dados acesse interfaces construídas para depósitos de dados orientados a registros.

- **Popular**¹¹ um modelo de DM utilizando o algoritmo especificado com os dados de treinamento;
- **Predizer** os atributos para novos dados utilizando um modelo de DM que foi treinado;
- **Expor o modelo** de DM para aplicativos de visualização, de geração de relatórios e de outras tarefas do processo KDD tais como interpretação e avaliação de resultados.

3.1.4 Componentes Básicos do OLE DB DM

Existem somente dois conceitos além da definição tradicional de OLE DB descrita no Anexo B: casos e modelos.

- Os dados de entrada representam um “conjunto de casos” ou *casesets*. De forma estrutural, um conjunto de casos não é diferente de uma tabela;
- Um modelo de DM ou DMM (*Data mining model*) é tratado como um tipo especial de tabela:
 - Um *caseset* é associado a um DMM; meta-informações adicionais são inseridas enquanto o DMM é definido ou criado;
 - Quando os dados, na forma de casos, são inseridos no DMM, um algoritmo de DM é processado e o resultado abstraído é salvo. Uma vez que um DMM está povoado, pode ser usado para predição ou seu conteúdo apresentado; As operações fundamentais do DMM incluem

¹¹ Refere-se ao processo de apresentar os dados ao modelo criado.

CREATE, INSERT INTO, PREDICTION JOIN, SELECT, DELETE FROM, e DROP.

3.1.4.1. Dados como Casos

Para assegurar que um DMM tenha acuracidade e significância, os algoritmos de DM requerem que todas as informações relacionadas a uma entidade estejam consolidadas antes que o algoritmo seja invocado. Normalmente os algoritmos de DM vêm a origem como uma simples tabela que representa informações consolidadas, onde cada linha representa uma instância de uma entidade. Se toda a informação relacionada a uma instância de uma entidade estiver em um conjunto de linhas, haverá dois importantes benefícios: primeiro, facilidade na utilização de algoritmos de DM tradicionais e segundo, aumento da escalabilidade por eliminar a necessidade de constantes *bookkeeping*¹².

Nos bancos de dados relacionais, como os dados estão frequentemente normalizados, a informação relacionada a uma entidade está espalhada por várias tabelas. Um passo chave para o DM é poder coletar a informação relacionada a uma entidade em um simples conjunto de linhas.

Esta coleção de dados relativos a uma simples entidade é chamada de caso e o conjunto de todos os casos relevantes é chamado *caseset* (conjunto de dados).

Um ponto importante a destacar, é o conceito de tabelas aninhadas (tabelas aninhadas como colunas). Apesar de serem familiares no universo de bancos de dados relacionais, não são adotadas universalmente em sistemas comerciais. Deve

¹² Diz-se da necessidade do DMM controlar as informações de entrada.

ser enfatizado que um conjunto de dados hierárquicos é uma definição lógica, não sendo necessário, portanto, subsistemas de armazenamento para suportar tabelas aninhadas. Os casos são somente instanciados como *rowsets* antes do processo de treinamento / predição pelo DMM.

Assim, muitas tabelas físicas podem ser usadas para gerar diferentes *casesets* para diferentes análises. Se por exemplo, escolhermos construir modelos sobre produtos, cada produto se torna um simples caso e os clientes que o compraram precisarão ser representados como colunas do caso.

3.1.5 Criar e Definir modelos de DM

No OLE DB DM, um DMM é tratado como um “objeto de primeira classe”, assim como uma tabela. Nesta operação o foco é a definição (criação) de DMM, onde se descreve as colunas dos dados, com meta-informações e os relacionamentos entre as colunas (se assume que um *dataset* é representado como tabelas aninhadas) e outras operações que DMM talvez suportem.

É preciso especificar:

- O nome do modelo;
- O algoritmo e parâmetros usados para a construção do modelo;
- O algoritmo para predição usando o modelo;
- As colunas do *caseset* que serão usadas e os relacionamentos entre estas colunas;
- Identificação das colunas que serão usadas como colunas de origem e a coluna que será “preenchida” pelo DMM (“coluna de predição”);

Utilizando o exemplo do capítulo 2 utilizado por Han e Kamber (1999), pode-se ilustrar (Listagem 3.1) a sintaxe para a criação do DMM, onde identifica as colunas de origem utilizadas, a coluna a ser predita e o algoritmo a ser utilizado. No anexo A, são apresentadas as especificações de tipos de colunas e atributos para a criação de DMM.

```
CREATE MINING MODEL [nome do modelo]  
(  
  [registro] LONG KEY,  
  [Idade] TEXT DISCRETE,  
  [Rendimentos] TEXT DISCRETE,  
  [Estudante] TEXT DISCRETE,  
  [ClassCredito] TEXT DISCRETE,  
  [Comprador] TEXT DISCRETE PREDICT  
)  
USING algoritmo usado pelo DMM
```

Listagem 3.1 – Exemplo de criação de um DMM, usando a notação proposta pelo padrão OLE DB DM.

Um atributo, contínuo ou discreto, pode ter uma distribuição associada. Estas distribuições são usadas como dicas para o DMM e podem especificar um conhecimento prévio sobre o dado. Desta forma, um atributo contínuo pode ser normal (Gaussiano), log normal ou uniforme. Um atributo discreto pode ser binomial, multinomial ou Poisson. Outras informações podem ser incluídas: NOT_NULL indica que um atributo nunca poderá ter um valor nulo; para o modelo somente quer dizer que a informação de interesse não está no valor de um atributo, mas no fato de o valor estar presente. Por consequência um atributo pode ser modelado como binário, onde a informação significativa será se o valor do atributo é conhecido ou não.

Atributos ou colunas tipo tabelas podem ser colunas de entrada, colunas de saída ou ambos. O provedor de DM constrói um DMM capaz de prever ou explicar valores de colunas de saída baseados nos valores das colunas de entrada.

Uma predição pode ser expressa por um histograma. Um histograma provê múltiplos possíveis valores de predição, cada um acompanhado por uma probabilidade e outras estatísticas. Quando uma informação do histograma é solicitada, cada predição talvez tenha uma coleção de possíveis valores que constituem um histograma. Desta forma é possível, devido à necessidade de extrair somente uma porção de informações preditivas, que se extraia somente a melhor estimativa, as três melhores estimativas ou as estimativas com a probabilidade maior que 55%. Nenhum DMM pode suportar todas as possíveis requisições. De qualquer modo, é necessário definir tudo o que pode ser extraído.

OLE DB DM define um conjunto de funções de transformações padronizadas sobre colunas de saída. O mecanismo básico é a flexibilidade para extrair valores de saída através de noções familiares como UDF (*user-defined functions*) usadas em OLAP. Cada provedor envia um conjunto de funções que pode ser solicitada em consultas de predição. Algumas UDF's são valores escalares assim como probabilidade. Outras têm tabelas como valores, assim como histogramas e então retornam tabelas aninhadas quando invocados.

3.1.6 Operações na modelagem de dados

a. Povoando um DMM: **INSERT**

Uma vez que o modelo está definido, o próximo passo é popular o modelo através de um *caseset* que satisfaça a especificação da criação do DMM declarado. Em OLE DB DM, é usado INSERT para instanciar o DMM. Ao contrário de uma tabela convencional, a inserção não resulta na adição de linhas de um *rowset*. Particularmente a inserção corresponde ao consumo das observações representadas por um caso usando do DMM. A Listagem 3.2 ilustra a sintaxe de povoamento de um DMM.

```

INSERT INTO [nome do modelo]
(
SKIP,
[Idade],
[Rendimento],
[Estudante],
[ClassCredito],
[Comprador]
)
OPENROWSET
(
'SQLOLEDB.1',
'Provider=SQLOLEDB;
Integrated Security=SSPI;
Persist Security Info=False;
Initial Catalog=""
Source=CLC',
'SELECT "Registro", "Idade", "Rendimento", "Estudante", "ClassCredito",
"Comprador" FROM "tabela de origem"')

```

Listagem 3.2 – Exemplo de operação de inserção de um DMM, usando a notação proposta pelo padrão OLE DB DM.

b. Usando o modelo para predições: **PREDICTION JOIN**

Após ter sido povoado, a operação básica de obter predições de um novo conjunto de dados usando um DMM, é através de um “*prediction join*” entre o modelo e o novo conjunto de dados. Certamente, o conjunto de dados precisa estar compatível com o esquema do DMM. O modelo não contém detalhes dos dados, assim a semântica da junção de predição é diferente daquelas de uma junção entre tabelas. A Listagem 3.3 ilustra a sintaxe da junção com um DMM.

```

SELECT FLATTENED
[T1].[Registro],
[T1].[Idade],
[T1].[Rendimento],
[T1].[Estudante],
[T1].[ClassCredito],
[T1].[Comprador],
[nome do modelo].[Comprador] as [Valor da Predicao]
FROM [nome do modelo] PREDICTION JOIN OPENROWSET ('SQLOLEDB.1',
'Provider=SQLOLEDB.1;
Integrated Security=SSPI;
Persist Security Info=False;
Initial Catalog= [nome do modelo];
Data Source=CLC',
'SELECT "Registro", "Idade", "Rendimento", "Estudante",
" ClassCredito ", "Comprador" FROM "[nome do modelo]"
ORDER BY "Registro") AS [T1] ON
[nome do modelo].[Registro] = [T1].[Registro] AND
[nome do modelo].[Idade] = [T1].[Idade] AND
[nome do modelo].[Rendimento] = [T1].[Rendimento] AND
[nome do modelo].[Estudante] = [T1].[Student] AND
[nome do modelo].[ClassCredito] = [T1].[ ClassCredito] AND
[nome do modelo].[Comprador] = [T1].[Comprador]

```

Listagem 3.3 – Exemplo de uma junção com um DMM, usando a notação proposta pelo padrão OLE DB DM.

c. Navegando pelo conteúdo do DMM: **SELECT**

A melhor e mais popular maneira de apresentar o conteúdo de um DMM é visualizando-o através de um gráfico (uma árvore de decisão sendo representada figurativamente na estrutura de uma árvore, um cluster na representado por um cilindro, etc.). O conteúdo de um DMM é o conjunto de regras, fórmulas, classificações, distribuições, nós e outra informação qualquer que tenha sido derivada de um conjunto específico de dados usando técnicas de DM. Desta forma o tipo de conteúdo varia de acordo com a técnica de DM especificada usada para criar o DMM. O conteúdo do DMM de uma classificação através de uma árvore de decisão será diferente de uma segmentação, assim como ambos serão diferentes de um DMM de regressão múltipla.

Atualmente, visando à portabilidade e o intercâmbio de informações o armazenamento é realizado através de uma especificação chamada de PMML (*Predictive Model Markup Language*)¹³. O PMML especifica um formato de persistência para DMM. De fato, na visualização dos métodos do DMM, se utiliza PMML inspirado em *strings* XML na exposição do conteúdo de um DMM.

3.1.7 Considerações finais sobre OLE DB DM

O foco do desenvolvimento do padrão OLE DB DM não foi na descoberta de novos algoritmos de KDD e sim na integração destes algoritmos com banco de dados relacionais. O objetivo final do projeto foi desenvolver uma API tendo como

¹³ Disponível em: <http://www.dmg.org/index.html>. Acesso em 10/10/2004 às 10:00

base a familiaridade do SQL e de conexão com banco de dados assim como ODBC e OLE DB. (MICROSOFT 1)

Segundo NETZ *et al* (2001), Projetos de pesquisa assim como o Quest e o DBMiner provêm interfaces para aplicação e interfaces para usuários que suportem DM e permitam o acesso a dados em DW. Entretanto, tais ferramentas não provêm recursos para trabalhar com modelos arbitrários e integração das interfaces de aplicação com SQL como também com APIs de acesso a dados relacionais (ODBC, OLE DB).

A metodologia CRISP-DM apresentada no capítulo 2, tem sido utilizada como um padrão para os processos de DM. Esta iniciativa é complementar ao OLE DB DM, mas provê somente um conjunto de metodologias, melhores práticas e a tentativa de definir as várias atividades envolvidas no processo de KDD.

CAPITULO 4 ESTUDO DE CASO

4.1 Análise do negócio

Através da compreensão e análise das principais necessidades do negócio, conforme especificado na introdução deste trabalho, foi possível compreender alguns problemas que interferem diretamente na DCBD do setor de telemarketing ativo:

- Devido à necessidade de se trabalhar com *mailings*¹⁴ fornecidos geralmente por clientes, onde normalmente se exige que todos os nomes sejam trabalhados, não faz sentido tentar classificá-los em possíveis compradores ou não. Isto se deve principalmente ao fato que, para cada possível cliente (*prospect*), independente de fatores como renda, idade, etc., normalmente há um produto adequado ao seu perfil. Outra razão importante está no fato destes *mailings* normalmente já terem sido filtrados pelos seus fornecedores de acordo com o objeto de venda.
- Normalmente os clientes de empresas de telemarketing ao adquirirem seus *prospects*, pagam pelos mesmos. Pode-se afirmar que o preço destes registros está ligado diretamente à sua qualidade. *Prospects* com informações sócio-econômicas custam muito mais caro que aqueles que possuem somente nome e telefone.

¹⁴ Refere-se ao conjunto de nomes de possíveis compradores, matéria-prima do telemarketing ativo.

Devido às situações descritas acima, a principal necessidade de quem trabalha com vendas de forma ativa, isto é, ligando para o *prospect*, se torna em encontrá-lo. Neste sentido, pode-se avaliar também que o sucesso da venda estará centralizado principalmente em: primeiro encontrar e em seguida na qualidade da abordagem deste *prospect*.

Desta forma, pode-se dizer que é preciso construir modelos que possam atender estas necessidades, além de outras inerentes à estrutura de um *call center*:

- Robustez: devido à quantidade de consumidores destes modelos (milhares de acessos simultâneos);
- Portabilidade: possa atender a diferentes aplicações consumidoras.
- Fácil manutenção;

De fato, têm-se ainda inúmeras necessidades para serem atendidas, não somente devido à heterogeneidade de produtos que podem ser ofertados como também ao grande número de segmentos que podem ser atendidos pelo *telemarketing* ativo, como por exemplo, venda de assinaturas de revistas e jornais, seguros, etc.

As considerações e preocupações expostas orientam para a definição dos objetivos definidos no presente trabalho. No entanto, para efeitos práticos a ênfase desta monografia será na apresentação dos recursos disponíveis para solução dos objetivos atendendo as necessidades descritas acima.

4.2 Recursos tecnológicos utilizados

Um ponto chave do sucesso de DM em bases de dados de telemarketing, trata-se da exploração e análise de forma automática ou semi-automática em grandes bases de dados. Este aspecto se torna imprescindível devido a dois fatores:

- Pela grande quantidade de dados que uma operação pode gerar. Por exemplo: historicamente um operador é capaz de gerar em média num único dia cerca de 200 registros de ligações. Considerando uma operação de telemarketing média com 100 posições de atendimento e portanto 200 operadores (2 turnos), chega-se ao volume de 40.000 registros diários. Em um mês o volume pode chegar a 1.040.000 registros.
- É preciso que os *prospects* estejam disponíveis de forma mais rápida possível e considerando que sejam consumidos mais de 1 milhão de nomes mensalmente, não será possível sempre tratá-los (inserir atributos de predição) para depois disponibilizá-los. Este processo precisa ser *on-line*.

Outro ponto importante a destacar é a cultura empresarial, onde cada gerente responsável por sua conta possui autonomia para definir suas prioridades e investimentos. Neste contexto, será preciso que cada um conheça seus DMM para que possam avaliá-los de acordo com outros fatores alheios ao processo de DMM como, por exemplo, fatores locais de uma determinada campanha não apresentados na forma de dados: atributos que não existam na sua base mas que são conhecidos na forma de dicas por exemplo.

Desta forma, torna-se necessária a implementação de uma aplicação voltada às características e necessidades locais. A estratégia passa a ser o

desenvolvimento de uma interface entre o analista de DM, o avaliador e consumidor dos DMMs.

No próximo tópico serão apresentadas as principais tecnologias utilizadas para o desenvolvimento da solução.

4.2.1 Microsoft® SQL Server™ 2000 Analysis Services

Coincidindo com o lançamento da especificação do OLE DB for Data Mining 1.0, o SGBD Microsoft® SQL Server™ 2000 integrou funcionalidades de DM junto com bancos relacionais e OLAP. O Analysis Services, componente do SQL Server 2000 se apresenta como um provedor para a mineração de dados baseado na especificação OLE DB DM. Esse provedor inclui dois algoritmos de DM: árvores de decisão e cluster Microsoft. A Figura 4.1 apresenta uma visão geral dos componentes presentes na arquitetura Microsoft para DM.

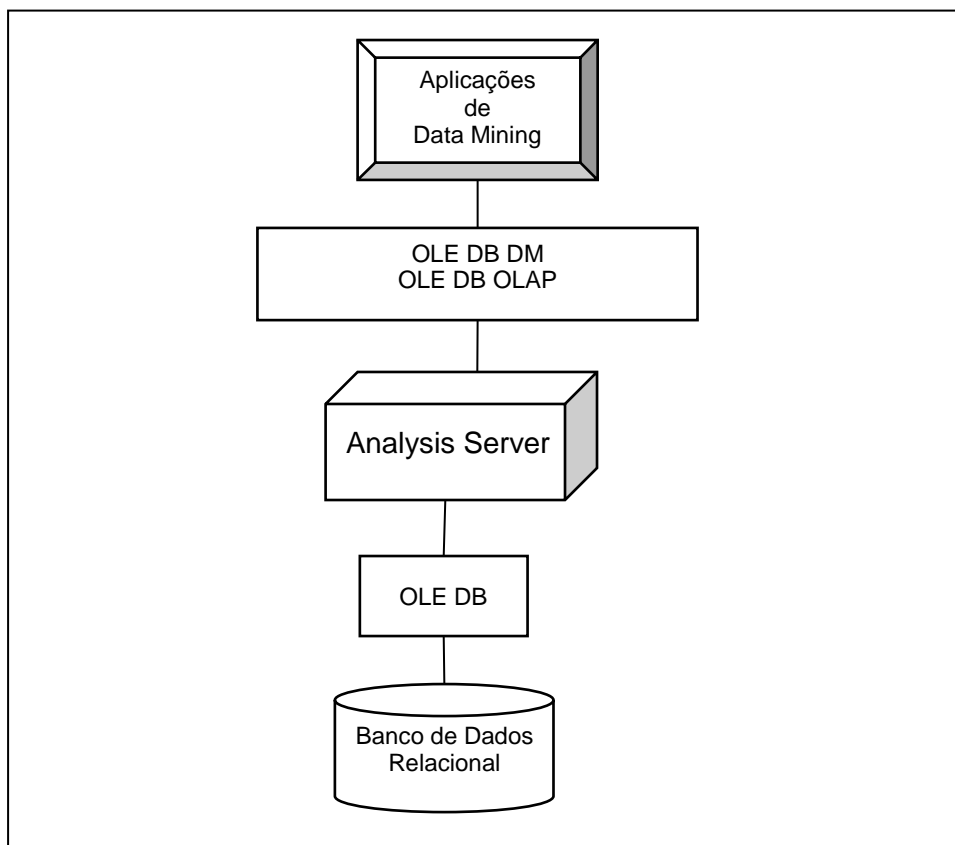


Figura 4.1 – Visão geral das funcionalidades de DM inseridas no SQL Server 2000

4.2.1.1 Classificador MSDT (*Microsoft Decision Tree*)

Segundo Chaudhuri (1999), além de ser a técnica mais popular para a modelagem de prognóstico, a escolha de implementação de árvores de decisão no Analysis Services se deveu ao fato de serem amplamente estudadas em estatística, reconhecimento de padrões e aprendizado de máquina e por poderem ser examinadas e interpretadas facilmente.

Há muitas variações de algoritmos que constroem árvores de decisão e que usam diferentes métodos de divisão: formas de árvore, técnicas de remoção, etc. Por padrão, a árvore de decisão Microsoft é uma árvore de classificação

probabilística muito parecida com o C4.5, mas ao invés de usar a entropia como critério de divisão, usa uma pontuação Bayesiana. Os algoritmos que fazem parte da árvore de decisão Microsoft utilizados para controlar o crescimento de uma árvore são:

- Entropia: baseado no ganho da entropia do classificador (apresentado em 2.2.1.1 Seleção de Atributos);
- Ortogonal: baseado na ortogonalidade da distribuição de estados no classificador. Este método produz somente divisões binárias, resultando em árvores de grande profundidade;
- Bayesiano com K2: baseado no escore Bayesiano com K2 a priori;
- Bayesiano Dirichlet Equivalente com Uniforme a priori: método padrão descrito por Chickering *et al* (1994) apud Curotto (2003).

4.2.2 *Decision Support Objects*

Conforme Microsoft (2), o Microsoft® SQL Server™ 2000 Analysis Services provê funcionalidades de DM e OLAP. Por ter sido desenvolvido para ser flexível e extensível, possibilita adicionar serviços e pacotes de terceiros, como por exemplo, provedores de algoritmos de DM para estender suas funcionalidades. Para o acesso de uma forma simples a estas funcionalidades, a biblioteca DSO (*Decision Support Objects*) fornece uma modelagem hierárquica de objetos para serem usados com ambientes de desenvolvimento que suporte objetos e interfaces COM (*Component Object Model*).

Essas classes e interfaces, quando usadas em conjunto, formam um modelo de objetos que correspondem à estrutura interna de objetos gerenciada pelo Analysis Services o que permite o gerenciamento de forma programada.

Conceitualmente, DSO utiliza um grupo de objetos arranjados de forma hierárquica para definir o elemento base de armazenamento de dados do Analysis Services. Estes elementos bases são *databases*, *data sources*, *dimensions*, *cubes*, *data mining models* e *roles*. O DSO mantém esses elementos básicos em uma estrutura hierárquica onde cada elemento contém outros elementos em uma estrutura de árvore, sendo que o objeto Server, é a raiz dessa árvore. A Figura 4.2 apresenta o diagrama da hierarquia de objetos do DSO.

Uma seqüência comum de operações para uma aplicação de DM que utiliza DSO é:

- Conectar a um servidor Analysis Services;
- Criar um objeto *database*.
- Adicionar um *data source* que indica a origem dos dados;
- Criar um DMM especificando seus parâmetros e a origem dos dados;
- Criar as colunas do DMM com suas propriedades;
- Processar o DMM.

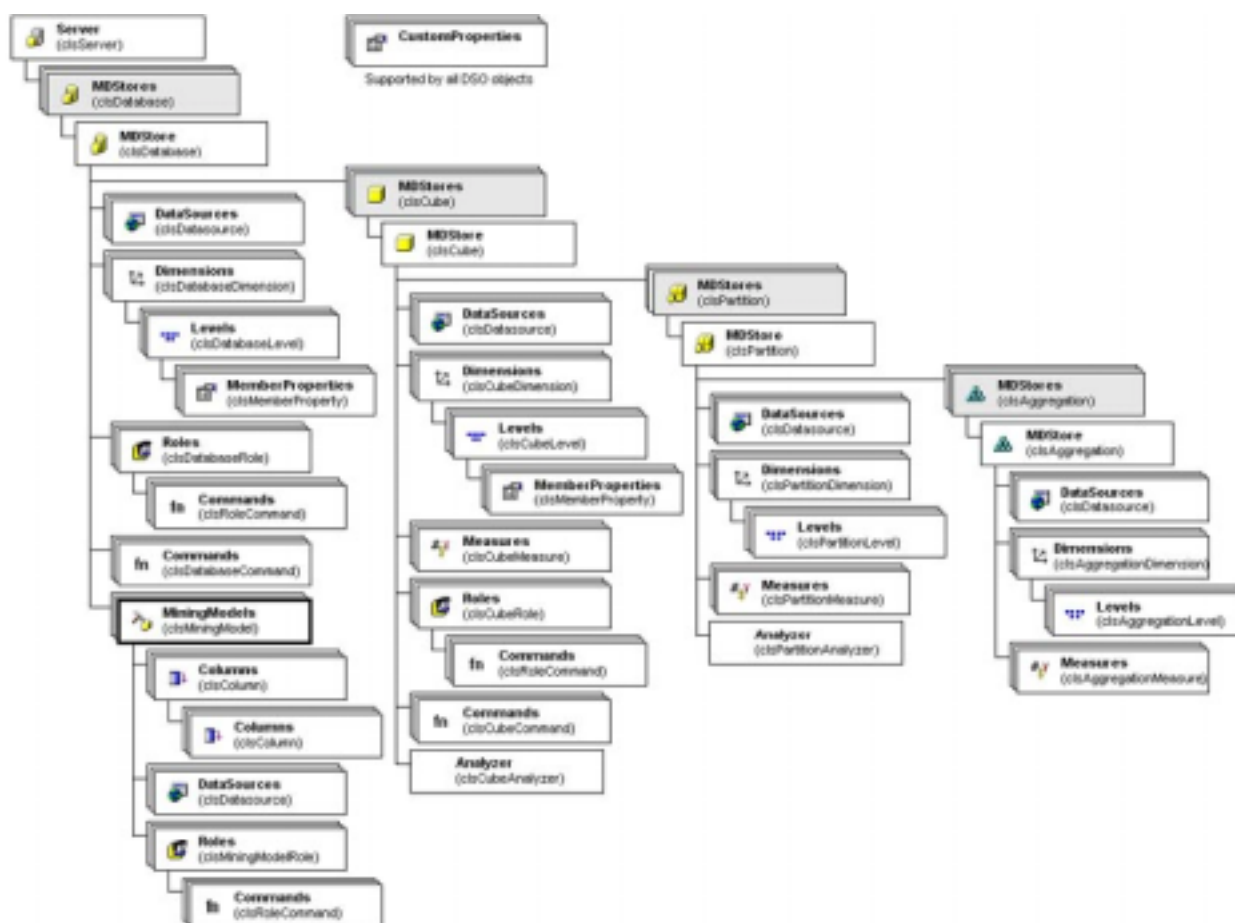


Figura 4.2 – Hierarquia de objetos DSO (MICROSOFT 2)

4.2.3 Linguagem de desenvolvimento C#

Segundo Sant'ana (2001) a linguagem C# foi criada pela Microsoft em conjunto com a arquitetura .NET Framework¹⁵, considerado como uma linguagem referência pelas seguintes razões:

- Foi construída para funcionar na nova plataforma, sem preocupações de compatibilidade com código já existente;

¹⁵ É uma plataforma que permite o desenvolvimento para WEB.

- A grande maioria das classes do .NET Framework e até mesmo o compilador JScript foram desenvolvidos em C#.

O C# inclui idéias de várias linguagens de programação, mas são patentes a influência do Pascal, do Delphi e o Java. Existem também claras influências do C++ e Smalltalk. Assim como o Java, o C# usa como base a sintaxe do C++. Isto significa que elementos como declaração de variáveis, métodos e estruturas de controle (if, loops) são muito semelhantes ao C++. As principais características C#:

- Modelo de orientação a objetos baseado em herança simples de classes com um ancestral comum;
- Herança múltipla de “interfaces”;
- Gerenciamento de memória automático com “coletor de lixo”;
- Tipagem forte;
- Rodam em um “ambiente gerenciado”, no qual a segurança e integridade das operações efetuadas pelos programas podem ser garantidas;
- Amplo suporte a “reflections”, um recurso também conhecido como “informação de tipos em tempo de execução”;
- Os programas na arquitetura .NET são sempre compilados.
- O C# tem enumerações, mais ou menos como versões mais recentes do C++ ou o próprio Pascal.
- Existe passagem de parâmetros por referência, na verdade de duas formas: “*ref*” significa a passagem por referência tradicional; “*out*” significa uma referência apenas “de saída”.
- Sobrecarga de operadores, algo muito útil nos “cálculos científicos”, por permitir tratar números complexos, vetores e matrizes com a notação dos operadores aritméticos tradicionais como “+” e “*”.

- Operadores de conversão, para converter valores de um tipo para outro. No C# existem tanto operadores de conversão implícitos, mais ou menos como no C++, como explícitos, que exigem o operador de “cast”. Ao contrário do C++, o construtor que aceita um único argumento não é usado automaticamente como função de conversão.
- Unificação do sistema de tipos. Todos os valores podem ser atribuídos a uma variável do tipo object em um processo chamado “boxing”.
- Tipo “decimal” para representar valores monetários, pouco sujeito a erros de arredondamento e representação.

4.3 A base de dados

4.3.1 Seleção dos atributos

A fonte dos dados para a mineração é uma base relacional, mais especificamente um repositório de DW e, portanto, tiveram um tratamento antes de serem disponibilizadas. No entanto os dados se encontram espalhados em diversas tabelas. Com o objetivo de simplificar o processo de seleção de dados, se torna necessário utilizar uma instrução SQL para agrupar estas variáveis em uma única tabela, facilitando o acesso do DMM pelo Analysis Services.

A Figura 4.3 representa os relacionamentos existentes entre as tabelas originais. A Tabela 4.1 apresenta as variáveis selecionadas e a sua descrição. Os registros representam os casos de clientes contatados, suas principais características sócio-econômicas e o produto oferecido. Com base nestas

informações será criado um DMM preditivo, utilizando árvore de decisão, que possa explicar possíveis relações com o horário do contato, conforme análise de negócio descrita no item 4.1.

Tabela 4.1 Variáveis selecionadas

VARIÁVEL	TIPO NO MODELO DE DMM	DESCRIÇÃO
IDENTIFICADOR	CHAVE	Identificador do <i>prospect</i>
DIA_DA_SEMANA	TEXTO DISCRETO	Dia da semana em que ocorreu
DE_GENERO	TEXTO DISCRETO	Gênero do <i>prospect</i>
DE_ESTADO_CIVIL	TEXTO DISCRETO	Estado civil do <i>prospect</i>
IDADE_VENDEDOR	INTEIRO CONTÍNUO	Idade do vendedor que efetuou o contato
DDD	INTEIRO DISCRETO	Identificação do DDD discado
DE_REGIAO	TEXTO DISCRETO	Região (específica do cliente)
DE_PROFISSAO	TEXTO DISCRETO	Profissão do <i>prospect</i>
DE_VINCULO	TEXTO DISCRETO	Vínculo empregatício do <i>prospect</i>
DE_FAIXA_RENDA	TEXTO DISCRETO	Faixa de renda do <i>prospect</i>
UF_RES	TEXTO DISCRETO	Estado de residência do <i>prospect</i>
CIDADE_RES	TEXTO DISCRETO	Cidade de residência
UF_NASC_VENDEDOR	TEXTO DISCRETO	Estado de origem do vendedor
GENERO_VENDEDOR	TEXTO DISCRETO	Gênero do vendedor
GRAU_INSTRUCAO_VENDEDOR	TEXTO DISCRETO	Grau de instrução do vendedor
ESTADO_CIVIL_VENDEDOR	TEXTO DISCRETO	Estado civil do vendedor
VL_IDADE	INTEIRO CONTÍNUO	Idade do <i>prospect</i>
NUM_DEPENDENTES	INTEIRO CONTÍNUO	Número de dependentes do <i>prospect</i>
SALARIO	REAL CONTÍNUO	Renda aproximada do <i>prospect</i>

ID_HORA_CONTATO	INTEIRO DISCRETO (TARGET)	Hora identificada do contato
VENDA_ATIVA	DISCRETO IGNORADO	Se a venda foi confirmada.

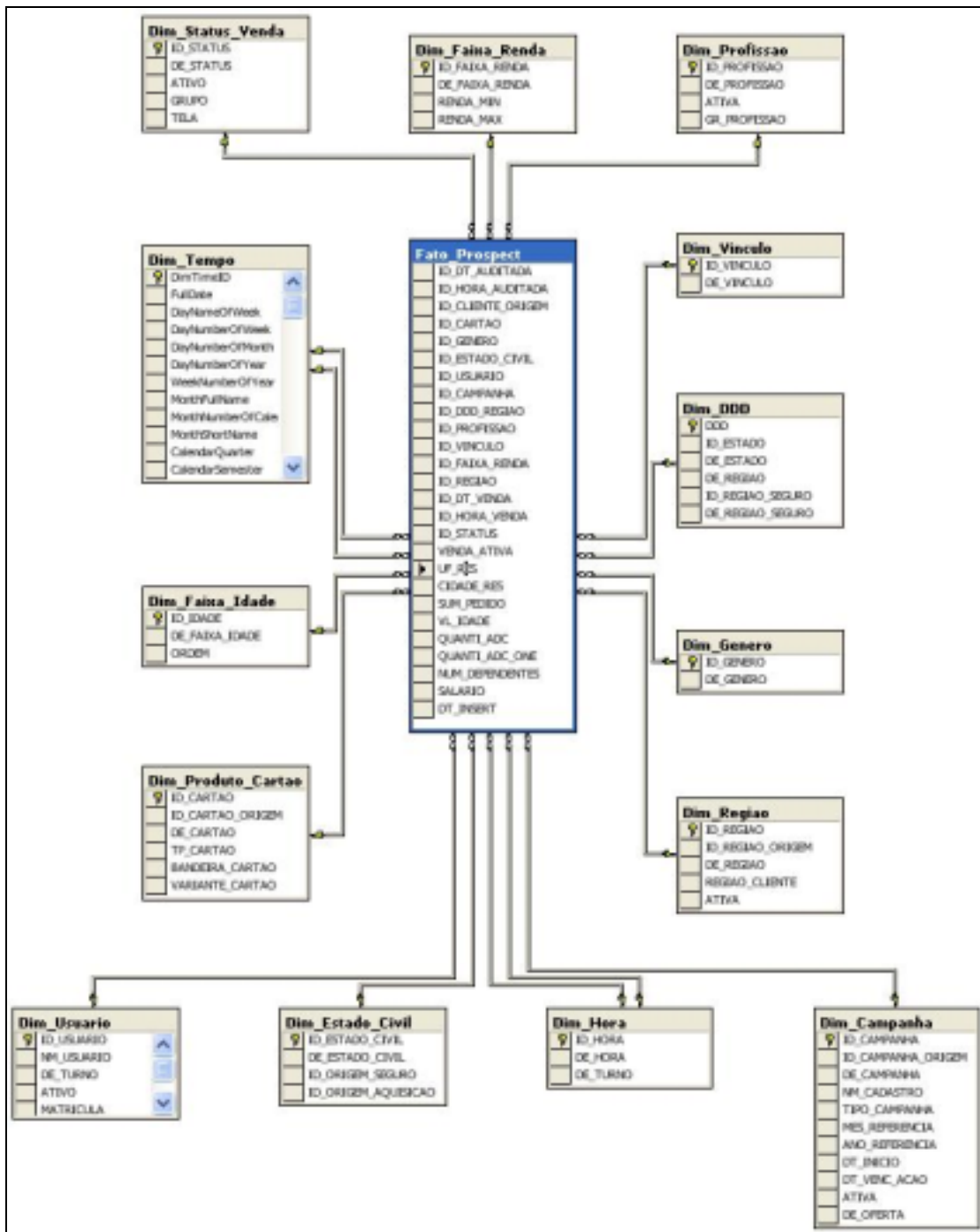


Figura 4.3 - Diagrama de Relacionamentos

4.3.2 Preparação dos dados

Por se originarem de um DW, tarefas comuns como pré-processamento, limpeza e transformação não foram necessários. No entanto, é necessária uma avaliação cuidadosa das transformações efetuadas devido à necessidade de se utilizar o DMM para predições em aplicações customizadas que exijam o uso do DMM em tempo-real. É necessário que estas transformações possam ser reversíveis, ou que possuam um dicionário que permita estas reversões. Isto é necessário para que seja possível fazer junções preditivas através dos valores dos atributos.

4.3.3 Amostragem

Selecionou-se cerca de 75.000 registros, separando 50.000 para a fase de treinamento e o restante para a fase de avaliação. Para o processo de seleção destes casos, foram utilizados os seguintes critérios:

1. Separou-se 300.000 registros do DW no período de 6 meses, atribuindo a cada registro um valor randômico, gerado a partir de uma função específica do SGBD.
2. Ordenando os dados de forma ascendente, separou-se 75.000 registros.
3. Novamente utilizando o atributo randômico, os registros foram separados em dois grupos, sendo o primeiro de treinamento com cerca de $2/3$ dos registros e o restante, ou $1/3$, para avaliação do modelo.

4.4 Desenvolvimento da aplicação

Utilizando-se das tecnologias apresentadas no item 4.2, desenvolveu-se uma *interface* experimental para permitir que objetos do provedor de DMM (Analysis Services) fossem instanciados em uma aplicação cliente, permitindo a customização do acesso, criação, manipulação e visualização dos DMMs.

As principais motivações para esta personalização foram fundamentas nas necessidades características do *telemarketing*, e na possibilidade de integrar outros processos de DM como:

- Importação, tratamento e transformação de dados outras fontes como planilhas eletrônicas, arquivos texto, etc. pelo usuário cliente;
- Integração com aplicações distribuídas para avaliação de modelos e consultas;
- Análises descritivas visuais de dados como média, desvio padrão, etc. utilizando instruções SQL.

Os benefícios imediatos desta estratégia podem ser citados:

- Não haver necessidade de instalações adicionais do Analysis Manager;
- Democratização de conhecimento pela necessidade de dividir as ações e;
- Conseqüente valorização da tecnologia.

O Apêndice 1 apresenta o código da principal classe de acesso ao provedor de DMM.

4.5 Aplicação da técnica de árvore de decisão

4.5.1 Construção

Conforme especificado no Anexo C, a Tabela 4.2 apresenta os parâmetros utilizados para a construção da MSDT. Selecionou-se o algoritmo de ganho de entropia, objeto de estudo desta monografia. A escolha da quantidade mínima de casos selecionada para uma folha, é justificada pela necessidade de inibir um crescimento acentuado, objetivando facilitar sua apresentação. Os demais parâmetros foram escolhidos de acordo com as orientações do anexo C.

Tabela 4.2 – Parâmetros utilizados para o treinamento.

Parâmetro	Valor
COMPLEXITY_PENALTY	0.9
MINIMUM_LEAF_CASES	5000
SCORE_METHOD	1 (Entropia)
SPLIT_METHOD	3 (padrão)

Os comandos utilizados para a criação dos modelos de DM são apresentados na Listagem 4.1, os comandos para povoamento na Listagem 4.2.

```
CREATE MINING MODEL [DM_CARTAO]
([Identificador] LONG KEY , [Dia Da Semana] TEXT DISCRETE , [De Genero] TEXT DISCRETE
, [De Estado Civil] TEXT DISCRETE , [Idade Vendedor] LONG CONTINUOUS , [Ddd] LONG
DISCRETE , [De Regiao] TEXT DISCRETE , [De Profissao] TEXT DISCRETE , [De Vinculo]
TEXT DISCRETE , [De Faixa Renda] TEXT DISCRETE , [Uf Res] TEXT DISCRETE , [Cidade
Res] TEXT DISCRETE , [Uf Nasc Vendedor] TEXT DISCRETE , [Genero Vendedor] TEXT
DISCRETE , [Grau Instrucao Vendedor] TEXT DISCRETE , [Estado Civil Vendedor] TEXT
```

```
DISCRETE , [VI Idade] LONG CONTINUOUS , [Num Dependentes] TEXT DISCRETE , [Salario]
DOUBLE CONTINUOUS , [Id Hora Venda] LONG DISCRETE PREDICT)
```

```
USING Microsoft_Ddecision_Trees (COMPLEXITY_PENALTY=0.9, MINIMUM_LEAF_CASES=5000,
SCORE_METHOD=1, SPLIT_METHOD=3)
```

Listagem 4.1 – Criação do DMM.

```
INSERT INTO [DM_CARTAO]
```

```
(SKIP, [Dia Da Semana], [De Genero], [De Estado Civil], [Idade Vendedor], [Ddd], [De Regiao], [De
Profissao], [De Vinculo], [De Faixa Renda], [Uf Res], [Cidade Res], [Uf Nasc Vendedor], [Genero
Vendedor], [Grau Instrucao Vendedor], [Estado Civil Vendedor], [VI Idade], [Num Dependentes],
[Salario], [Id Hora Venda])
```

```
OPENROWSET('SQLOLEDB.1', 'Provider=SQLOLEDB.1;Password=*****;Persist Security
Info=True;User ID=SA;Initial Catalog=DM_CARTAO;Data Source=LOCALHOST,
```

```
'SELECT "dbo"."TREINAMENTO"."IDENTIFICADOR" AS "Identificador",
"dbo"."TREINAMENTO"."DIA_DA_SEMANA" AS "Dia Da Semana",
"dbo"."TREINAMENTO"."DE_GENERO" AS "De Genero",
"dbo"."TREINAMENTO"."DE_ESTADO_CIVIL" AS "De Estado Civil",
"dbo"."TREINAMENTO"."IDADE_VENDEDOR" AS "Idade Vendedor", "dbo"."TREINAMENTO"."DDD"
AS "Ddd", "dbo"."TREINAMENTO"."DE_REGIAO" AS "De Regiao",
"dbo"."TREINAMENTO"."DE_PROFISSAO" AS "De Profissao",
"dbo"."TREINAMENTO"."DE_VINCULO" AS "De Vinculo",
"dbo"."TREINAMENTO"."DE_FAIXA_RENDA" AS "De Faixa Renda",
"dbo"."TREINAMENTO"."UF_RES" AS "Uf Res", "dbo"."TREINAMENTO"."CIDADE_RES" AS "Cidade
Res", "dbo"."TREINAMENTO"."UF_NASC_VENDEDOR" AS "Uf Nasc Vendedor",
"dbo"."TREINAMENTO"."GENERO_VENDEDOR" AS "Genero Vendedor",
"dbo"."TREINAMENTO"."GRAU_INSTRUCAO_VENDEDOR" AS "Grau Instrucao Vendedor",
"dbo"."TREINAMENTO"."ESTADO_CIVIL_VENDEDOR" AS "Estado Civil Vendedor",
"dbo"."TREINAMENTO"."VL_IDADE" AS "VI Idade", "dbo"."TREINAMENTO"."NUM_DEPENDENTES"
AS "Num Dependentes", "dbo"."TREINAMENTO"."SALARIO" AS "Salario",
"dbo"."TREINAMENTO"."ID_HORA_VENDA" AS "Id Hora Venda" FROM "dbo"."TREINAMENTO")
```

Listagem 4.2 – Povoamento do DMM.

Após os processos de criação e povoamento do DMM, utilizando-se de componentes gráficos disponíveis (MSN Groups, 2004), possibilitou-se a uma visualização fácil dos padrões do DMM processado conforme Figura 4.4.

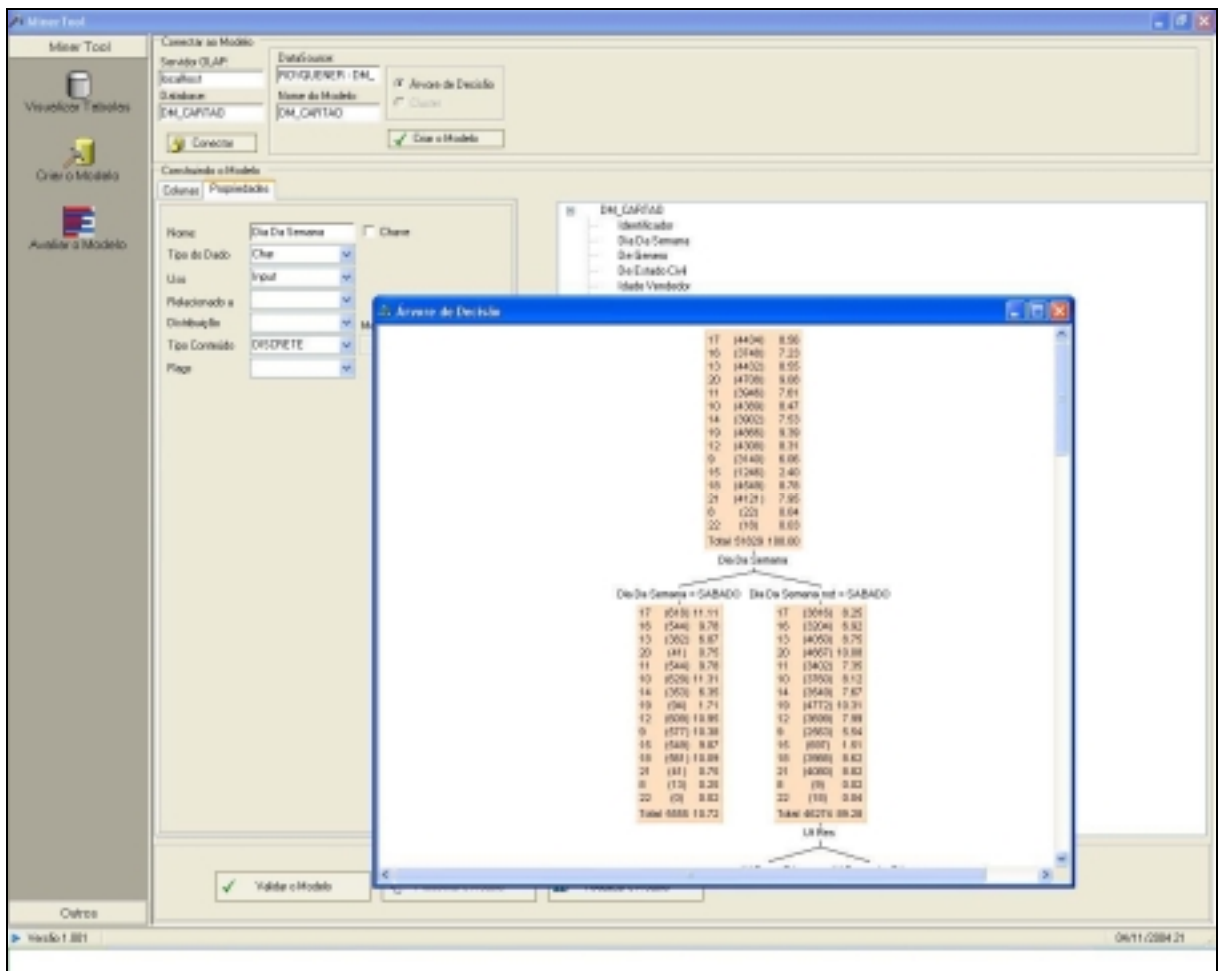


Figura 4.4 – Visualização do DMM através de componentes gráficos.

O provedor Analysis Services disponibiliza, através da *interface* DSO a estrutura do modelo no formato XML segundo padrões PMML, possibilitando a portabilidade do modelo e o desenvolvimento de interfaces gráficas customizadas. A Figura 4.5 apresenta um resumo da estrutura da árvore gerada em XML.

```

<model name="DM_CARTAS" UUID="{446A99CE-62A8-4C46-8783-26528E6F47D3}" creation-time="2884-11-84723:87:42" modified-time="2884-11-84723:88:23">
  <statements>
    <statement type="CREATE" value="CREATE MODEL DM_CARTAS (([Identificador] LONG KEY, [Dia Da Semana] TEXT DISCRETE, [De Genero] TEXT DISCRETE, [De Estado Civil] TEXT DISCRETE, [Idade Vendedor] LONG CONTINUOUS, [Idade] LONG DISCRETE, [De Regiao] TEXT DISCRETE, [De Profissao] TEXT DISCRETE, [De Vinculo] TEXT DISCRETE, [De Faixa Renda] TEXT DISCRETE, [UH Res] TEXT DISCRETE, [[idade Res] TEXT DISCRETE, [UH Anos Vendedor] TEXT DISCRETE, [Genero Vendedor] TEXT DISCRETE, [Grau Instrucao Vendedor] TEXT DISCRETE, [[Estado Civil Vendedor] TEXT DISCRETE, [M Idade] LONG CONTINUOUS, [Vanos Dependentes] TEXT DISCRETE, [Salario] DOUBLE CONTINUOUS, [Id Hora Venda] LONG DISCRETE PREDICT) USING Microsoft_Decision_Trees [COMPLEXITY_PENALTY=0.9, MINIMUM_LEAF_CASES=5000, SCORE_METHOD=1, SPLIT_METHOD=3] );
  </statements>
  <data-dictionary column-ordering="ORIGINAL" column-count="20" score-method="1" split-method="3" minimum-leaf-cases="5000" complexity-penalty="0.9">
    <key name="Identificador" datatype="LONG" isinput="true" column-ordinal="0" />
    <categorical name="Dia Da Semana" isinput="true" datatype="TEXT" column-ordinal="1">
      <category value="SEGUNDA-FEIRA" />
      <category value="TERÇA-FEIRA" />
      <category value="QUARTA-FEIRA" />
      <category value="QUINTA-FEIRA" />
      <category value="SABADO" />
      <category value="DOMINGO" />
    </categorical>
    <categorical name="De Genero" isinput="true" datatype="TEXT" column-ordinal="2">
      <category name="De Estado Civil" isinput="true" datatype="TEXT" column-ordinal="3">
      <continuous name="Idade Vendedor" column-ordinal="4" isinput="true" datatype="LONG">
      <categorical name="Idade" isinput="true" datatype="LONG" column-ordinal="5">
      <categorical name="De Regiao" isinput="true" datatype="TEXT" column-ordinal="6">
      <categorical name="De Profissao" model-as-binary="true" isinput="true" datatype="TEXT" column-ordinal="7">
      <categorical name="De Vinculo" isinput="true" datatype="TEXT" column-ordinal="8">
      <categorical name="De Faixa Renda" isinput="true" datatype="TEXT" column-ordinal="9">
      <categorical name="UH Res" isinput="true" datatype="TEXT" column-ordinal="10">
      <categorical name="Idade Res" model-as-binary="true" isinput="true" datatype="TEXT" column-ordinal="11">
      <categorical name="UH Anos Vendedor" isinput="true" datatype="TEXT" column-ordinal="12">
      <categorical name="Genero Vendedor" isinput="true" datatype="TEXT" column-ordinal="13">
      <categorical name="Grau Instrucao Vendedor" isinput="true" datatype="TEXT" column-ordinal="14">
      <categorical name="Estado Civil Vendedor" isinput="true" datatype="TEXT" column-ordinal="15">
      <continuous name="M Idade" column-ordinal="16" isinput="true" datatype="LONG">
      <categorical name="Vanos Dependentes" isinput="true" datatype="TEXT" column-ordinal="17">
      <continuous name="Salario" column-ordinal="18" isinput="true" datatype="REAL">
      <categorical name="Id Hora Venda" isinput="true" ispredict="true" datatype="LONG" column-ordinal="19">
    </data-dictionary>
    <global-statistics>
      <data-distribution>
        <data-distribution>
          <single-attribute name="De Genero" />
          <state missing="true" support="8" />
          <state value="FEMININO" support="31853" />
          <state value="MASCULINO" support="19719" />
          <state value="NAO INFORMADO" support="253" />
        </data-distribution>
      </data-distribution>
      <data-distribution>
        <data-distribution>
          <single-attribute name="Idade Vendedor" />
          <state missing="true" support="8" />
          <state minimum="13" maximum="56" support="51829" mean="24.3829448611731" standard-deviation="6.3238048639416" />
        </data-distribution>
      </data-distribution>
      <data-distribution>
      <data-distribution>
      <data-distribution>
      <data-distribution>
      <data-distribution>
      <data-distribution>
    </global-statistics>
  </data-dictionary>

```

Figura 4.5 – Visualização do DMM através de XML.

4.5.2 Validação do DMM

Para o processo de validação do DMM, utilizou-se o Método Holdout, onde a taxa de acertos é dada pela fórmula 2.1. Neste processo foi utilizando o utilitário DTS (*Data Transformation Services*) (PETERSON, 2001) integrante do Microsoft® SQL Server™ 2000. Trata-se de uma ferramenta para copiar, mover, consolidar, limpar e validar dados.

Através da consulta da Listagem 4.3, obteve-se uma tabela onde constam os atributos da base de validação agregado ao atributo preditivo gerado pelo DMM.

Observa-se nesta listagem a seleção dos 3 melhores horários de contato, agregado ao valor de suporte e probabilidade.

```

SELECT FLATTENED

  [T1].[Identificador] as Identificador,
  [T1].[Dia Da Semana],
  [T1].[De Genero],
  [T1].[De Estado Civil],
  [T1].[Idade Vendedor],
  [T1].[Ddd],
  [T1].[De Regiao],
  [T1].[De Profissao],
  [T1].[De Vinculo],
  [T1].[De Faixa Renda],
  [T1].[Uf Res],
  [T1].[Cidade Res],
  [T1].[Uf Nasc Vendedor],
  [T1].[Genero Vendedor],
  [T1].[Grau Instrucao Vendedor],
  [T1].[Estado Civil Vendedor],
  [T1].[VI Idade],
  [T1].[Num Dependentes],
  [T1].[Salario],
  [T1].[Id Hora Venda] as Atual,
  [DM_CARTAO].[Id Hora Venda] as Predicao_No,
  (SELECT [Id Hora Venda] as Predicao, $Support as Suporte, $Probability as Certeza FROM
  TopCount(PredictHistogram([DM_CARTAO].[Id Hora Venda]), $Probability, 3)) as PH

FROM
  [DM_CARTAO]
  PREDICTION JOIN
  OPENROWSET
  (
    'SQLOLEDB.1',
    'Provider=SQLOLEDB.1;Password=****;Persist Security Info=True;User ID=sa;Initial
    Catalog=DM_CARTAO;Data Source=LOCALHOST',
    'SELECT "IDENTIFICADOR" AS "Identificador", "DIA_DA_SEMANA" AS "Dia Da Semana",
    "DE_GENERO" AS "De Genero", "DE_ESTADO_CIVIL" AS "De Estado Civil", "IDADE_VENDEDOR"
    AS "Idade Vendedor", "DDD" AS "Ddd", "DE_REGIAO" AS "De Regiao", "DE_PROFISSAO" AS "De
    Profissao", "DE_VINCULO" AS "De Vinculo", "DE_FAIXA_RENDA" AS "De Faixa Renda", "UF_RES"
    AS "Uf Res", "CIDADE_RES" AS "Cidade Res", "UF_NASC_VENDEDOR" AS "Uf Nasc Vendedor",
    "GENERO_VENDEDOR" AS "Genero Vendedor", "GRAU_INSTRUCAO_VENDEDOR" AS "Grau
    Instrucao Vendedor", "ESTADO_CIVIL_VENDEDOR" AS "Estado Civil Vendedor", "VL_IDADE" AS
    "VI Idade", "VL_IDADE" AS "Num Dependentes", "SALARIO" AS "Salario", "ID_HORA_VENDA" AS
    "Id Hora Venda" FROM "VALIDACAO" ORDER BY "IDENTIFICADOR"
  )
  AS [T1]
  ON
  [DM_CARTAO].[Dia Da Semana] = [T1].[Dia Da Semana] AND
  [DM_CARTAO].[De Genero] = [T1].[De Genero] AND
  [DM_CARTAO].[De Estado Civil] = [T1].[De Estado Civil] AND
  [DM_CARTAO].[Idade Vendedor] = [T1].[Idade Vendedor] AND
  [DM_CARTAO].[Ddd] = [T1].[Ddd] AND
  [DM_CARTAO].[De Regiao] = [T1].[De Regiao] AND
  [DM_CARTAO].[De Profissao] = [T1].[De Profissao] AND

```



```

[DM_CARTAO].[De Vinculo] = [T1].[De Vinculo] AND
[DM_CARTAO].[De Faixa Renda] = [T1].[De Faixa Renda] AND
[DM_CARTAO].[Uf Res] = [T1].[Uf Res] AND
[DM_CARTAO].[Cidade Res] = [T1].[Cidade Res] AND
[DM_CARTAO].[Uf Nasc Vendedor] = [T1].[Uf Nasc Vendedor] AND
[DM_CARTAO].[Genero Vendedor] = [T1].[Genero Vendedor] AND
[DM_CARTAO].[Grau Instrucao Vendedor] = [T1].[Grau Instrucao Vendedor] AND
[DM_CARTAO].[Estado Civil Vendedor] = [T1].[Estado Civil Vendedor] AND
[DM_CARTAO].[VI Idade] = [T1].[VI Idade] AND
[DM_CARTAO].[Num Dependentes] = [T1].[Num Dependentes] AND
[DM_CARTAO].[Salario] = [T1].[Salario] AND
[DM_CARTAO].[Id Hora Venda] = [T1].[Id Hora Venda]

```

Listagem 4.3 – Consulta de junção com a base de validação.

O resultado da validação na base de validação resultou em 7.621 acertos utilizando as três maiores possibilidades. Isto representa cerca 30% de acertos, com uma certeza média acumulada de 31,81%. Avaliando-se o acerto considerando somente a maior possibilidade, observou-se 2.546 casos ou 10,62% de acertos.

4.5.3 Consumo do DMM

Um aspecto importante na integração de SGBDs com ferramentas de DM é a possibilidade de consumir os modelos considerando-os como tabelas. Assim, através de *interfaces* de conexão com o provedor de DMM, consumir os modelos através de junções possibilita que aplicações customizadas possam se beneficiar do DM para automatizar processos como exemplificado na Figura 4.6.

The screenshot shows the Alteryx Designer interface. On the left, there's a sidebar with icons for 'Verificar Tabelas', 'Criar o Modelo', and 'Analisar o Modelo'. The main workspace is divided into a 'Modelo' (Data Model) section and a 'Consulta' (Query) section. The Data Model shows a table 'DM_CARTAO' with columns 'M_Hora_Venda' and 'E_estado'. Below it, a 'SELECT' query is visible, starting with 'SELECT RATTENED' and containing various table references and conditions. A 'Consulta' window is open, displaying a table with the following columns: 'Identificador', 'Atual', 'Predicao_Op', 'Predicao', 'Sepsos', and 'Coracao'. The table contains multiple rows of data, such as (2400976, 11, 20, 10, 1821, 0.113621524). At the bottom, there are buttons for 'Executar Query' and 'Mostrar as Saídas', along with a status bar indicating 'Versão 1.001' and '05/11/2004 K'.

Figura 4.6 – Exemplo de consulta para consumo do DMM.

CONCLUSÃO

Através do conteúdo apresentado nesta monografia possibilitou-se a criação de uma ferramenta para a automação de processos de DM no ambiente comercial estudado. Os pontos fortes desta abordagem estão centrados na tecnologia OLE DB DM e nos demais recursos disponíveis para a integração das atividades de DM, demonstrando a viabilidade para uma integração eficiente com SGBDs comerciais.

No entanto para a construção de modelos de alta qualidade, necessita-se agregar práticas de planejamento e construção de modelos através de uma correta especificação desde seu planejamento até a apresentação dos resultados obtidos. Proposta do padrão CRISP-DM.

Um aspecto importante a destacar se trata da integração de diversas áreas de conhecimento para o cumprimento dos objetivos propostos. Áreas de estudo como Estatística, Mineração de Dados e Data Warehouse foram decisivas mas não menos importantes que Banco de Dados, Sistemas Distribuídos e Programação Orientada a Objetos. Para atender às expectativas de um segmento específico de consumidores de DM, desenvolveu-se um produto segundo suas necessidades e características, apresentando-se uma solução inovadora para o segmento de *telemarketing*.

REFERÊNCIAS BIBLIOGRÁFICAS

AMARAL, Fernanda Cristina Naliato do. **Data mining, técnicas e aplicações para o marketing direto**. Editora Berkeley, 2001.

BATISTA, Gustavo Enrique de Almeida Prado Alves. **Um ambiente de avaliação de algoritmos de aprendizado de máquina utilizando exemplos**. São Paulo. Dissertação (Mestrado em Ciências Matemáticas). Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo. São Paulo, 1997.

BERSON, Alex; SMITH, Stephen; THEARLING, Kurt. **Building data mining for CRM**. Ed McGraw-Hill, 1999.

CARVALHO, Luís Alfredo Vidal de; **Data mining: a mineração de dados no marketing, medicina, economia, engenharia e administração**; São Paulo: Érica, 2001.

CHAUDHURI, Surajit. Data mining and database systems: where is the intersection? **Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**. Março, 1998.

CHAUDHURI, Surajit; USAMA Fayyad; BERNHARDT, Jeff. Scalable Classification over SQL Databases. **Proceedings of 15th International Conference on Data Engineering, Sydney, Australia**. 1999.

COLOGNA, Carla di. Telemarketing: setor contribui para geração de empregos. **Carreira & Sucesso – Newsletter**, 212ª edição semanal, 15 janeiro 2004. Disponível em: <http://www.catho.com.br/jcs/inpuer_view.phtml?id=6465>. Acesso em: 05/02/2004 às 10:00h

CRISP-DM 1.0 - Cross-Industry Standart Process dor Data Mining. Step-by-step data mining guide. SPSS 2000. Disponível no endereço: <<http://www.crisp-dm.org/>>. Acesso em: 09/02/2004 às 15:00h.

CUROTTO, Cláudio Luiz. **Integração de recursos de data Mining com gerenciadores de bancos de dados relacionais**. Rio de Janeiro. Tese (Doutorado em Ciências de Engenharia Civil). Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2003;

PETERSON, Timothy. **Microsoft SQL Server 2000 (DTS)**. Tradução Edson Furmankiewicz, Joana Figueiredo. – Rio de Janeiro: Campus, 2001

HAN, Jiawie e KAMBER, Micheline. **Data mining: concepts and techniques**. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, 1999.

HAND, David ; MANNILA , Heikki; SMYTH , Padhraic. **Principles of data mining**; A Bradford Book The MIT Press. Massachusetts Institute of Technology. Cambridge Massachusetts LondonEngland, 2001.

ICA – Laboratório de inteligência Computacional Aplicada. Descoberta de conhecimentos e mineração de dados. Departamento de Engenharia Elétrica, Pontifca Universidade Católica – PUC/ Rio de Janeiro, **1999**.

SANT'ANNA, Mauro. **C#: A nova linguagem da arquitetura .NET**. 2001. Disponível em: http://www.linhadecodigo.com.br/artigos.asp?id_ac=15>. Acesso em: 15/10/2004 às 18:00.

MICROSOFT CORPORATION, **OLE DB for Data Mining Specification Version 1.0**, Microsoft Corporation, Redmond, Washington, USA, 2000. Disponível em: <<<http://www.microsoft.com/downloads/details.aspx?FamilyID=01005f92-dba1-4fa4-8ba0-af6a19d30217&DisplayLang=en>> Acesso em: 10/09/2004 às 23:00h.

MICROSOFT CORPORATION, **Decision Support Objects Architecture**. MSDN Library. Microsoft Corporation, Redmond, Washington, USA, 2000. Disponível em: < http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/prabout_27hh.asp > Acesso em: 08/08/2004 às 17:00h.

NETZ, Amir; CHAUDHURI, Surajit; BERNHARDT, Jeff; FAYYAD, Usama. Integration of data mining and relational databases, **Proceedings of the 26th International Conference on Very Large Databases**, Cairo, Egypt, 2000.

NETZ, Amir; Bernhardt Jeff; CHAUDHURI Surajit; FAYYAD, Usama. Integrating data mining with SQL databases: OLE DB for data mining. **Proceedings of 17th International Conference on Data Engineering**. Heidelberg, Germany, 2001.

PAUL, Seth; GAUTAM, Nitin; BALINT, Raymond. **Preparing and mining data with Microsoft SQL Server 2000 and Analysis Services**. Online Books, Microsoft SQL Server Series. Maio, 2004.

RUD, Olivia Parr. **Data mining cookbook modeling data for marketing, risk, and customer relationship management**. Wiley Computer Publishing, November, 2000

TWO CROWS CORPORATION. **Introduction to data mining and knowlede discovery**. Third Edition, 1999. Disponível em: < <http://www.twocrows.com/intro-dm.pdf>>. Acesso em 20/01/2004 às 12:00h

BIBLIOGRAFIAS CONSULTADAS

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: 05/10/2004 às 09:00h. [INGA96]

CHICKERING, David M., GEIGER, Dan. & HECKERMAN, David.. **Learning Bayesian Networks: the combination of knowledge and statistical data**, Technical Report MSR-TR-94-09, Microsoft Research, Microsoft Corporation, Redmond, Washington, USA. Março, 1994.

O'BRIEN, James A. **Sistemas de informação e as decisões gerenciais na era da internet**. Tradução Cid Knipel Moreira. São Paulo: Saraiva, 2002. [BRIEN02]

PETERSON, Timothy. **Microsoft SQL Server 2000 (DTS)**. Tradução Edson Furmankiewicz, Joana Figueiredo. Rio de Janeiro: Campus, 2001. [DTS01]

SINGH, Harry S.. **Data warehouse – conceitos, tecnologias, implementação e gerenciamento**. São Paulo: Makron Books, 2001.

MSN Groups. Analysis Services : Data Mining. Material utilizado para a implementação. <http://groups.msn.com/AnalysisServicesDataMining/links.msnw?action=view_list&viewtype=1&sortstring=> Acessado em: 15/08/2004 às 18:00.

ANEXOS

ANEXO A

ESPECIFICAÇÃO DOS TIPOS DE COLUNAS E ATRIBUTOS OLE DB DM

As descrições destes tipos de colunas e atributos foram extraídas do artigo Netz *et all* (2001).

A.1 Colunas:

- KEY: a coluna que identifica a linha. Identifica um caso de forma única;
- ATTRIBUTE: um atributo direto de um caso. Este tipo de coluna representa algum valor do caso;
- RELATION: informação usada para classificar atributos, outros relacionamentos ou colunas chaves. Na sintaxe de criação do DMM as relações são identificadas na definição da coluna usando a cláusula RELATED TO para indicar a coluna a ser classificada;
- QUALIFIER: um valor especial associado com um atributo que tem um significado para o provedor. São qualificadores opcionais e se aplicam somente se não há certeza da ligação do dado ou se provêm de uma predição de um DMM de um passo anterior. Na sintaxe de criação do modelo, os modificadores são identificados pela cláusula OF. Tipo de qualificadores:
 - a) PROBABILITY: a probabilidade [0,1] da associação do valor;
 - b) VARIANCE: um número que descreve a variância do valor de um atributo;

- c) SUPPORT: um valor do tipo *float* que representa o peso (fator de replicação) para ser associado a um valor;
- d) PROBABILITY_VARIANCE: a variância associada com o avaliador da probabilidade usada pela PROBABILITY;
- e) ORDER: especifica a ordem de uma coluna;
- f) TABLE: uma tabela aninhada que consiste de um tipo especial de colunas com o tipo TABLE. Para muitas linhas dadas, o valor de uma coluna do tipo TABLE contém o conteúdo completo de uma tabela aninhada associada.

A.2 Atributos:

- DISCRETE: atributo do tipo categórico sem ordenação implícita (p.e., código de área);
- ORDERED: colunas que definem um conjunto de valores ordenados;
- CYCLICAL: um conjunto de valores que tem uma ordenação cíclica. O dia da semana é um bom exemplo, o dia 1 vem após o dia 7;
- DISCRETIZED: o dado que será inserido no DMM é contínuo, mas será transformado e modelado como um número ORDERED pelo provedor. Muitos algoritmos não aceitam atributos contínuos como parâmetro de entrada ou não podem prever valores contínuos;
- SEQUENCE_TIME: uma dimensão de tempo. O formato não é restrito, por exemplo, o número de um período é aceito.

ANEXO B

Tecnologia OLE DB

A tecnologia OLE DB DM é uma extensão da tecnologia OLE DB, a qual será descrita de acordo com o trabalho original de Skonnard (1998) apud Curotto (2003).

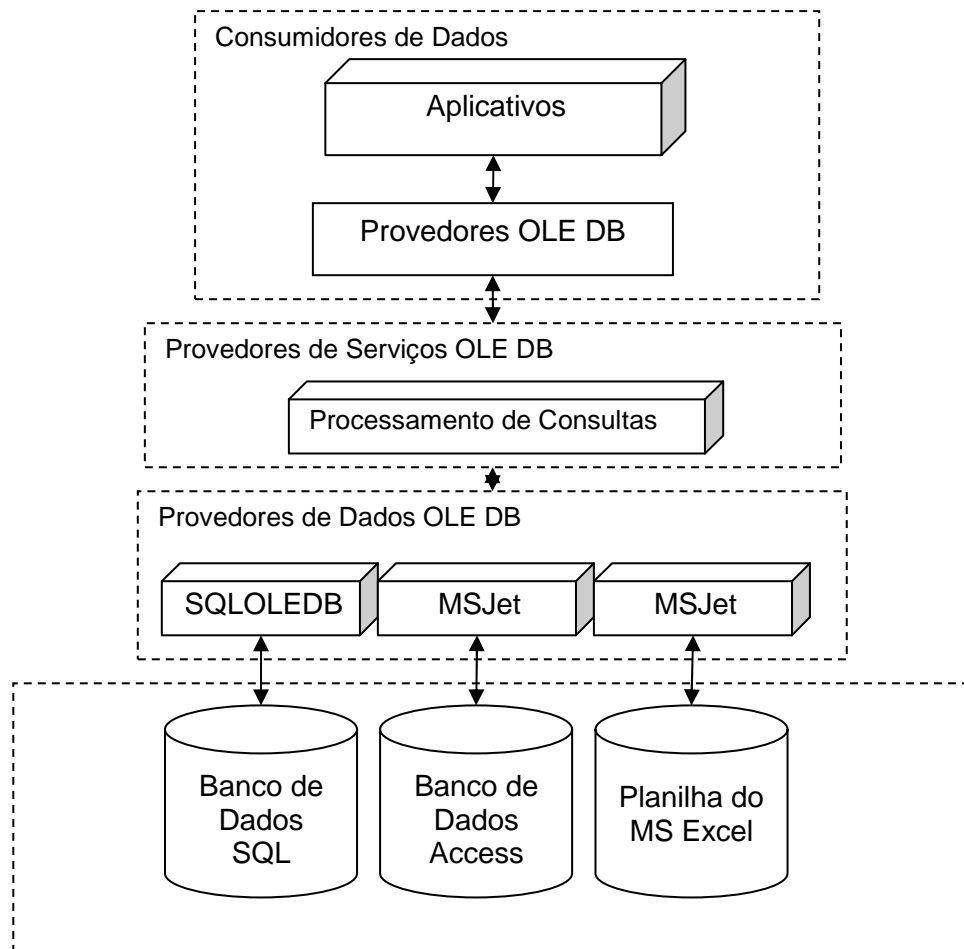


Figura B.1 – Arquitetura OLE DB

OLE DB é baseada na tecnologia OLE, sendo uma especificação para um conjunto de interfaces de acesso de dados projetadas para possibilitar que uma

variedade de armazenadores de dados, de todos os tipos e tamanhos, interajam harmonicamente juntos.

Não existe uma restrição para que os dados sejam manipulados através da linguagem SQL. O único critério imposto por esta tecnologia é que os dados resultantes de um comando devem ser expostos na forma de uma tabela. Assim qualquer dado pode ser manipulado, desde que um simples dado pode ser interpretado como uma tabela de uma linha e uma coluna.

Conforme mostra a Figura B.1, uma característica importante da OLE DB é a sua heterogeneidade, que permite consultas baseadas em dados de diferentes formatos, armazenados por diferentes fontes, tais como servidores SQL, bancos de dados Access® e planilhas Excel®, entre outros.

A arquitetura OLE DB é composta por três componentes: consumidores de dados, provedores de serviços e provedores de dados.

Os consumidores de dados são aplicativos ou sistemas, podendo ser inclusive provedores OLE DB.

Os provedores de dados possuem os dados e expõem os mesmos através de objetos para o mundo exterior. Cada provedor de dados possui características próprias de implementação para manipular os diferentes tipos de dados, porém todos os provedores expõem seus dados em uma única forma tabular através de tabelas virtuais.

Os provedores de serviços são componentes lógicos que encapsulam funcionalidades de SGBD's. Alguns exemplos que podem ser citados são: processadores de consultas, processadores de cursor e gerenciadores de transações.

Os sete componentes básicos do modelo de objetos de um provedor OLE DB são:

- **Enumerator:** enumeradores listam as fontes de dados disponíveis e outros enumeradores. Consumidores de dados que não são customizados para utilizar umas fontes de dados em particular, utilizam enumeradores para pesquisar uma fonte de dados para sua utilização.
- **Data Source:** objetos de fontes de dados contêm os recursos e propriedades para conexões com as fontes de dados, tais como um arquivo simples ou um SGBD. Eles são geradores de sessões.
- **Session:** sessões fornecem um contexto para transações e podem ser implicitamente ou explicitamente transacionadas. Um simples objeto de fonte de dados pode criar sessões múltiplas. Sessões são geradoras de transações, comandos e tabelas virtuais.
- **Transaction:** objetos de transações são utilizados quando transações aninhadas são abortadas ou empenhadas em níveis superiores ao nível mais baixo.
- **Command:** Comandos executam um comando tipo texto, tal como um comando SQL. Se o comando texto especifica uma tabela virtual, tal como um comando SQL SELECT, o comando é um gerador da tabela virtual. Uma simples sessão pode criar múltiplos comandos.
- **Rowset:** tabelas virtuais expõem dados no formato tabular. Um caso especial de tabela virtual é um índice. Tabelas virtuais podem ser criadas por comandos ou por sessões.

- **Error**: objetos de Erros podem ser criados por qualquer interface ou por qualquer objeto OLE DB. Eles contêm informação adicional sobre um erro, incluindo um objeto de erro customizado.

ANEXO C

Parâmetros do Classificador MSDT

As informações constantes neste anexo foram extraídas de Curotto (2003).

C.1 COMPLEXITY_PENALTY

É um número real que pode variar entre 0 e 1 (exclusive), que atua como um restrigente ao crescimento da árvore. A aplicação dele é realizada em cada divisão adicional da árvore. O valor de 0 significa nenhuma penalidade, enquanto que o valor próximo de 1 (já que 1 fica fora do intervalo permitido) significa penalidade máxima e crescimento mínimo da árvore. A aplicação desta penalidade limita a profundidade e a complexidade das árvores de aprendizado, o que evita desajustes. Entretanto o uso de valores elevados de penalidade prejudica a capacidade preditiva do modelo. O efeito deste parâmetro é dependente de cada modelo, assim experimentos e observações devem ser realizados para obter o melhor resultado com cada modelo de DM. O valor padrão é baseado no número de atributos de dados para um dado modelo: para 1 a 9 atributos, o valor é 0,5; para 10 até 99 atributos, o valor é 0,9; para 100 ou mais atributos o valor adotado é 0,99.

C.2 MINIMUM_LEAF_CASES

É um inteiro não negativo que pode variar no intervalo entre 0 e 2.147.483.647. Ele determina o número mínimo de casos de uma folha necessários para gerar uma divisão na árvore. Valores baixos ocasionam maiores divisões na

árvore de decisão, mas pode aumentar a probabilidade de desajuste. Valores altos reduzem o número de divisões na árvore de decisão, mas podem inibir o crescimento da árvore de decisão. O valor padrão é igual a 10.

C.3 SCORE_METHOD

Identifica o algoritmo utilizado para controlar o crescimento de uma árvore de decisão. Este algoritmo seleciona os atributos que constituem a árvore, a ordem no qual estes atributos são usados, o modo pelo qual os valores dos atributos devem ser divididos e o ponto no qual a árvore deve parar o crescimento. Os valores válidos são: 1, 2, 3, 4, que correspondem aos seguintes algoritmos:

1. Entropia;
2. Ortogonal;
3. Bayesiano com K2;
4. Bayesiano Dirichlet Equivalente com Uniforme a priori.

C.4 SPLIT_METHOD

Descreve as várias formas que o algoritmo definido pelo parâmetro SCORE_METHOD deve considerar para dividir valores de atributos. Por exemplo, se um atributo tem 5 potenciais valores, os valores podem ser divididos em galhos binários (valor 3 e valores 1, 2, 4, 5) ou os valores podem ser divididos em 5 galhos separados, ou ainda uma outra combinação qualquer pode ser considerada. O valor 1 para este parâmetro resulta em árvores de decisão que podem ter apenas galhos binários. O valor 2 resulta em árvores de decisão com múltiplos galhos, enquanto

que o valor 3 (padrão) permite que o algoritmo escolha divisões binárias ou múltiplas de acordo com o necessário.

APÊNDICE 1

Código fonte da classe utilizada para consumir o componente DSO.

```
using System;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;
using DSO; //adicionado como referência no Solution...

namespace MinerTool.classe
{
    public class DataMiningModel
    {
        // atributos do objeto DSO
        private DSO.Server dsoServer;
        private DSO.MDStore dsoDB;
        private DSO.MiningModel dsoDMM;
        private DSO.DataSource dsoDS;
        private DSO.Role dsoRegra;

        private string strLQuote, strRQuote;
        private string servidor, database;
        private classe.ConectarSQL umaConexaoSQL; //uma classe de conexão OleDb
        private System.Collections.ArrayList colecaoColunas;
        private System.Collections.ArrayList colecaoFromClause;

        // utilizada para setar o tipo do dado na coluna...
        ADODB.DataTypeEnum umTipo;

        public DataMiningModel(string umServidor, string umDatabase, classe.ConectarSQL
umaConexao)
        {
            // somente o servidor pode ser instanciado...
            // os demais objetos trabalham por referência ...
            this.dsoServer = new DSO.ServerClass();

            this.servidor = umServidor;
            this.database = umDatabase;
            this.umaConexaoSQL = umaConexao;

            this.colecaoColunas = new System.Collections.ArrayList();
            this.colecaoFromClause = new System.Collections.ArrayList();

            // conectando ...
            try
            {
                this.dsoServer.Connect(umServidor);
                this.dsoDB = (MDStore)dsoServer.MDStores.Item(database);
            }
            catch
            {
                throw;
            }
        }

        public bool criarModelo(string umNomeModelo, string umDataSource, string
umaListaDeParametros)
        {
            bool jaExiste = false;

            try
            {
                // iniciando a transacao ...
                this.dsoDB.BeginTrans();

                int qtdeModelos = this.dsoDB.MiningModels.Count;
            }
        }
    }
}
```

```

// procurando se já existe algum modelo com o mesmo nome
int i = 1;
int j = 0;
while (i <= qtdeModelos)
{
    if (((MiningModelClass)this.dsoDB.MiningModels.Item(i)).name == umNomeModelo)
    {
        jaExiste = true;
        j = i; // guardando o indice do modelo que já existe...
    }
    i++;
}

// Se nao existe cria ...
if (jaExiste)
{
    //this.dsoDB.MiningModels.Remove(umNomeModelo);
    this.dsoDMM = (MiningModel)this.dsoDB.MiningModels.Item(j);
}
else
{
    // inserindo um novo modelo...
    this.dsoDMM = (MiningModel)this.dsoDB.MiningModels.AddNew(umNomeModelo,
DSO.SubClassTypes.sbclsRelational);

    // adicionando um datasource...
    this.dsoDMM.DataSources.AddNew(umDataSource, SubClassTypes.sbclsRelational);

    // Setando os parâmetros ...
    string parametros = umaListaDeParametros;
    ((MiningModelClass)this.dsoDMM).set_Parameters(ref parametros);

    // Setando o algoritmo...
    string algoritmo = "Microsoft_Decision_Trees";
    ((MiningModelClass)this.dsoDMM).set_MiningAlgorithm(ref algoritmo);
}

// Obter os caracteres delimitadores do database
this.dsoDS = (DataSource)dsoDMM.DataSources.Item(umDataSource);
this.strLQuote = dsoDS.OpenQuoteChar.ToString();
this.strRQuote = dsoDS.CloseQuoteChar.ToString();

// Encerrando a transacao ...
this.dsoDB.CommitTrans();

// se a conexao SQL estiver aberta, feche-a...
umaConexaoSQL.verificaEstadoConexao();

// setando o datasource da conexao ole para o mesmo da datasource do dso.MDStores ...
umaConexaoSQL.setarStringConexao(((DSO.DataSource)dsoDB.DataSources.Item(umDataSource)).Conn
ectionString);

    return jaExiste;
}
catch
{
    this.dsoDB.Rollback();
    throw;
}
}
public void selecionarTabela(string umaTabela)
{
    // INSERIR O CONCATENACAO DE TABELAS...
    string fromClause = "";
    bool j = false;

    // se nao existir no array, insere ...
    if (colecãoFromClause.IndexOf(umaTabela) < 0)
        coleçãoFromClause.Add(umaTabela);

    for (int i =0; i < coleçãoFromClause.Count; i++)
    {
        if (j == true)//se nao for a primeira vez ...
            fromClause += ", ";
        j = true;
    }
}

```

```

        fromClause +=
strLQuote+"dbo"+strRQuote+"."+strLQuote+colecãoFromClause[i].ToString()+strRQuote;
    }

    // setando a clausula from ...
    ((MiningModelClass)this.dsoDMM).set_FromClause(ref fromClause);
}

public void inserirColuna(string[] listaDeParametros)
{
    // 0 - TABELA DE ORIGEM
    // 1 - NOME DA COLUNA
    // 2 - NOME DA COLUNA NO BD
    // 3 - SE É CHAVE
    // 4 - TIPO DO DADO
    // 5 - USO
    // 6 - DISTRIBUICAO
    // 7 - FLAG
    // 8 - RELACIONADO A
    // 9 - TIPO DE CONTEUDO
    // 10 - METODO DE DISCRETIZACAO
    // 11 - BUCKETS

    DSO.Column dsoColumn;

    try
    {
        // iniciando a transacao...
        this.dsoDMM.Parent.BeginTrans();

        // inserindo nome que será visualizado no modelo (amigável)...
        dsoColumn = (DSO.Column)this.dsoDMM.Columns.AddNew(listaDeParametros[1],
SubClassTypes.sbclsRelational);

        // definindo se a coluna é a chave do caso ...
        bool eChave = Convert.ToBoolean(listaDeParametros[3]);
        ((DSO.ColumnClass)dsoColumn).set_IsKey(ref eChave);

        // definindo o tipo da coluna ...
        switch (listaDeParametros[4])
        {
            case "Char":
                umTipo = ADODB.DataTypeEnum.adChar;
                break;
            case "Integer":
                umTipo = ADODB.DataTypeEnum.adInteger;
                break;
            case "Single":
                umTipo = ADODB.DataTypeEnum.adSingle;
                break;
        }
        ((DSO.ColumnClass)dsoColumn).set_DataType(ref umTipo);

        // Estes parâmetros são somente para colunas não chave...
        if (!(DSO.ColumnClass)dsoColumn).get_IsKey()
        {
            // definindo o uso da coluna ...
            bool entrada = false;
            bool previsao = false;

            switch (listaDeParametros[5])
            {
                case "Input":
                    entrada = true;
                    break;
                case "Input e Predictable":
                    entrada = true;
                    previsao = true;
                    break;
                case "Predictable":
                    previsao = true;
                    break;
                case "":
                    break;
            }
            ((DSO.ColumnClass)dsoColumn).set_IsInput(ref entrada);
            ((DSO.ColumnClass)dsoColumn).set_IsPredictable(ref previsao);
        }
    }
}

```

```

// definindo o tipo de distribuicao ...
string distribuicao = listaDeParametros[6];
((DSO.ColumnClass)dsoColumn).set_Distribution(ref distribuicao);

// definindo os flags ...
string flag = listaDeParametros[7];
((DSO.ColumnClass)dsoColumn).set_ModelingFlags(ref flag);

// definindo a relacao com outras colunas ...
string relacionadoCom = listaDeParametros[8];
((DSO.ColumnClass)dsoColumn).set_RelatedColumn(ref relacionadoCom);

//definindo como ativa ...
bool ativa = false;
((DSO.ColumnClass)dsoColumn).set_IsDisabled(ref ativa);

// definindo o tipo de conteudo ...
string tipoConteudo = listaDeParametros[9];
string definindoConteudo;
if (tipoConteudo == "DISCRETIZED" )
    definindoConteudo = "DISCRETIZED(" +listaDeParametros[10]+ ", " +
listaDeParametros[11]+ ")";
else
    definindoConteudo = tipoConteudo;
((DSO.ColumnClass)dsoColumn).set_ContentType(ref definindoConteudo);
}

//definindo a origem da coluna ...
string fromClause = ((DSO.ColumnClass)dsoColumn).get_FromClause();
string origem =
this.strLQuote+"dbo"+strRQuote+"."+strLQuote+listaDeParametros[0]+strRQuote+"." +
this.strLQuote + listaDeParametros[2]+ this.strRQuote;

((DSO.ColumnClass)dsoColumn).set_SourceColumn(ref origem);

//definindo a clausula join

//atualizando o modelo ...
this.dsoDMM.Update();

//finalizando com sucesso a transacao ...
this.dsoDMM.Parent.CommitTrans();

//salvando num ArrayList a coluna selecionada...
colecacaoColunas.Add(dsoColumn);
}
catch
{
    this.dsoDMM.Parent.Rollback();
    throw;
}
finally
{
    dsoColumn = null;
}
}

public void removerColuna(string umaColuna)
{
    try
    {
        this.dsoDMM.Parent.BeginTrans();

        this.dsoDMM.Columns.Remove(umaColuna);
        this.dsoDMM.Update();

        this.dsoDMM.Parent.CommitTrans();

        for (int i = 0 ;i < colecaoColunas.Count; i ++)
            if (((DSO.ColumnClass)colecacaoColunas[i]).name == umaColuna)
                this.colecaoColunas.RemoveAt(i);
    }
    catch
    {
        throw;
    }
}

```

```

}

public string[] selecionarColuna(string umaColunaSelecionada)
{
    DSO.Column dsoColumn = null;

    try
    {
        for (int i = 0 ;i < colecaoColunas.Count; i ++)
            if (((DSO.ColumnClass)colecaoColunas[i]).name == umaColunaSelecionada)
                dsoColumn = (DSO.Column)colecaoColunas[i];

        string[] listaDeParametros = new string[12];

        listaDeParametros[1] = ((DSO.ColumnClass)dsoColumn).name;
        listaDeParametros[2] = "";
        listaDeParametros[3] = Convert.ToString(((DSO.ColumnClass)dsoColumn).get_IsKey());

        switch (Convert.ToString(((DSO.ColumnClass)dsoColumn).get_DataType()))
        {
            case "adInteger":
                listaDeParametros[4] = "Integer";
                break;
            case "adChar":
                listaDeParametros[4] = "Char";
                break;
            case "adSingle":
                listaDeParametros[4] = "Single";
                break;
        }

        if (((DSO.ColumnClass)dsoColumn).get_IsInput())
            listaDeParametros[5] = "Input";
        if (((DSO.ColumnClass)dsoColumn).get_IsPredictable())
            listaDeParametros[5] = "Predictable";
        if (((DSO.ColumnClass)dsoColumn).get_IsInput() &&
            ((DSO.ColumnClass)dsoColumn).get_IsPredictable())
            listaDeParametros[5] = "Input e Predictable";
        if (!(DSO.ColumnClass)dsoColumn).get_IsInput() &&
            !(DSO.ColumnClass)dsoColumn).get_IsPredictable())
            listaDeParametros[5] = "";

        listaDeParametros[6] = ((DSO.ColumnClass)dsoColumn).get_Distribution();
        listaDeParametros[7] = ((DSO.ColumnClass)dsoColumn).get_ModelingFlags();
        listaDeParametros[8] = ((DSO.ColumnClass)dsoColumn).get_RelatedColumn();

        string metodoDiscretizacao =
            Convert.ToString(((DSO.ColumnClass)dsoColumn).get_ContentType());

        if (metodoDiscretizacao.Length > 11 && metodoDiscretizacao != "")
        {
            int posicaoVirgula = metodoDiscretizacao.IndexOf(",",1,metodoDiscretizacao.Length-1);

            listaDeParametros[9] = "DISCRETIZED";
            listaDeParametros[10] = metodoDiscretizacao.Substring(12,posicaoVirgula-12);
            listaDeParametros[11] = metodoDiscretizacao.Substring(posicaoVirgula+1,2);
        }
        else
        {
            listaDeParametros[9] = metodoDiscretizacao;
            listaDeParametros[10] = "";
            listaDeParametros[11] = "0";
        }

        return listaDeParametros;
    }
    catch
    {
        throw;
    }
    finally
    {
        dsoColumn = null;
    }
}

public string processar()

```

```

{
    try
    {
        this.dsoDMM.Process(ProcessTypes.processFull);

        string status = "Concluido em " + dsoDMM.LastProcessed.ToUniversalTime();
        return status;
    }
    catch
    {
        throw;
    }
}

public void fechar()
{
    this.dsoDB = null;
    this.dsoDMM = null;
    this.dsoDS = null;
    this.dsoRegra = null;

    if (dsoServer.State == DSO.ServerStates.stateConnected)
        dsoServer.CloseServer();

    dsoServer = null;
}

public string[] pesquisarColunas()
{
    try
    {
        int qtdeColunas = this.dsoDMM.Columns.Count;

        string[] colunas = new string[qtdeColunas];

        for (int i = 1 ; i <= qtdeColunas; i++)
        {
            // guardando no array de colunas ...
            colecaoColunas.Add(dsoDMM.Columns.Item(i));
            // inserindo no array de nomes...
            colunas[i-1] = ((DSO.ColumnClass)colecaoColunas[i-1]).name;
        }
        return colunas;
    }
    catch
    {
        throw;
    }
}

public void validarModelo()
{
    try
    {
        this.dsoDMM.ValidateStructure();
    }
    catch
    {
        throw;
    }
}

public string getPMML()
{
    return this.dsoDMM.XML.ToString();
}
}
}

```

ARTIGO

DESCOBERTA DE CONHECIMENTO EM SISTEMAS GERENCIADORES DE BANCOS DE DADOS: mineração de dados

Royquener Reuter
Universidade Federal de Santa Catarina – UFSC
2004, Brasil

INTRODUÇÃO

De forma simples, Data Mining, se refere à extração ou “mineração” de conhecimento de grandes quantidades de dados. Há muitos outros termos conduzindo de uma forma similar ou para diferentes propósitos na mineração de dados. Assim como: “minerando conhecimento de banco de dados”, “extraindo conhecimento”, “dados/padrões de análise” e “escavando dados”. (HAN e KAMBER, 1999).

Este trabalho terá como preocupação estudar e aplicar os recursos e benefícios do DM para o aperfeiçoamento do segmento de telemarketing nas suas principais preocupações como uma empresa de *call center*: procurar meios de automatizar processos que possam melhorar o contato com o cliente e buscar respostas para o impacto social conseqüente deste rápido crescimento. Espera-se também melhorar a produtividade das campanhas e conseqüente diminuição dos custos relacionados.

O **objetivo geral**, é o estudo e aplicação dos conceitos de DM no setor de telemarketing, com foco em uma operação ativa de venda de cartões de crédito, visando elaborar uma base de conhecimento para a utilização de ferramentas de mineração integradas a SGBD's (*Sistemas Gerenciadores de Bancos de Dados*) e suas técnicas de mineração de dados.

O desenvolvimento deste estudo buscou atender o seguinte **objetivo específico**: através de recursos tecnológicos disponíveis no mercado, desenvolver uma aplicação visando construir um modelo

preditivo para melhorar a performance de contatos com os clientes.

1 DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

A mineração de dados é tratada por muitas pessoas como um sinônimo de outro termo popularmente usado, KDD (Knowledge Discovery in Database) ou DCBD (Descoberta de Conhecimento em Base de Dados). Alternativamente algumas outras visões de mineração de dados podem ser consideradas simplesmente um passo do processo de DCBD. (HAN e KAMBER, 1999)

Resumidamente, o primeiro e simples passo para a mineração de dados é a sua descrição – sumarizar atributos estatísticos (assim como medir e indicar desvios), examiná-los visualmente usando mapas e gráficos e procurar por potenciais e significantes ligações entre variáveis (assim como valores que muitas vezes ocorrem juntamente).

O passo final é verificar o modelo de forma empírica. Por exemplo, para uma base de dados de clientes que já têm respostas para uma oferta particular, já existe embutido um modelo de predição nas quais as possibilidades são prováveis para responder à mesma oferta. Seria possível confiar nesta predição?

1.1 A Integração de DM e Bancos de Dados Relacionais

O progresso da pesquisa em DM tem vindo da possibilidade de implementar várias operações de mineração de forma eficiente em grandes bancos de dados. Enquanto isto é seguramente uma

importante contribuição, não podemos deixar de olhar os objetivos finais do DM – isto é, permitir que aplicações de banco de dados construam modelos de DM (por exemplo: árvores de decisão e classificação, modelos de regressão, segmentação) e usem estes modelos para realizar tarefas preditivas e analíticas e possam também compartilhar estes mesmos modelos com outras aplicações. Tal integração deve ser uma pré-condição para que o DM tenha sucesso em banco de dados. (NETZ *et al*, 2000)

Reconhecendo o fato acima, é obvio que um aspecto chave para integração com sistemas de banco de dados que é preciso ser observado é como tratar modelos de DM como objetos de primeira classe¹⁶ nos SGBD's. Infelizmente, em qualquer aspecto, o DM ainda continua sendo uma “ilha” de análises que é pobremente integrada com sistemas de banco de dados.

Lembrando que um modelo de DM é obtido via aplicação de um algoritmo de DM em um conjunto de treinamento. Desta forma, mesmo que um modelo possa ser derivado usando uma aplicação SQL (*Structured Query Language*) que implemente um algoritmo de treinamento, o SGBD é completamente inconsciente da semântica do modelo de DM porque estes não podem ser representados explicitamente em um banco de dados.

Em seguida, para efetivamente representar modelos de mineração em SGBD's, precisamos capturar a criação da mineração de dados usando algoritmos arbitrários de mineração, navegar por estes modelos (examinando suas estruturas ou seu conteúdo), e aplicar um modelo selecionado num conjunto de dados para analisar tarefas como uma predição. Além disto, para a coluna que é o resultado da predição, são necessárias informações suficientes para que outras ferramentas de análise possam interpretar as propriedades da predição, como acuracidade, por exemplo. (NETZ *et al*, 2000)

¹⁶ Refere-se a objetos de primeira classe, quando podemos representá-los na forma de ocorrências com seus atributos e inseridos em tabelas.

1.2 Processos de DCBD

DM se refere à extração ou mineração de conhecimento de grandes quantidades de dados. De forma alternativa, outras visões de mineração de dados podem ser simplesmente um dos passos essenciais do processo de descobrir conhecimento em banco de dados (HAN e KAMBER, 1999). Na Figura 1, a DCBD é descrita como um processo que consiste de uma seqüência interativa de passos:

- ✓ **Consolidação dos Dados:** os dados resultantes podem ser armazenados em um DW;
 - **Limpeza:** remover ruídos e dados irrelevantes.
 - **Integração:** os dados de diversas fontes são integrados;
- ✓ **Seleção dos Dados:** os dados relevantes para a tarefa de análise são retirados de um banco de dados ou DW;
- ✓ **Transformação dos Dados:** os dados são transformados ou consolidados em uma forma apropriada para ser minerado.
- ✓ **Mineração dos Dados:** um processo essencial onde métodos inteligentes são aplicados para extrair padrões.
- ✓ **Avaliação do Padrão:** para identificar e avaliar padrões verdadeiros utilizando métricas de validação;
- ✓ **Apresentação e geração do conhecimento;**

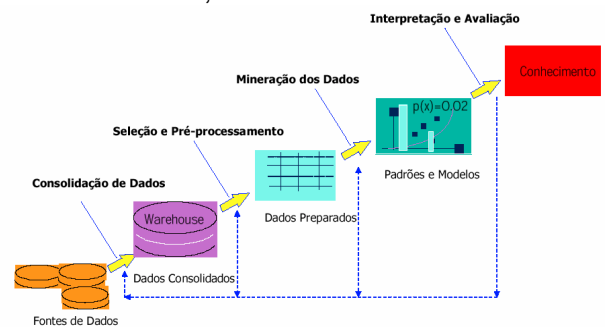


Figura 1 – Processos de KDD (HAN e KAMBER, 1999).

Mais recentemente, objetivando a geração de modelos com maior qualidade e utilizando-se de padronização de conceitos e técnicas na busca de informações para a tomada de decisões, foi formulado o padrão CRISP-DM que

fornece um conjunto de metodologias, práticas e definições das atividades que envolvem o processo de DCBD.



Figura 2 – Fases do modelo de referência do CRISP-DM.

O padrão CRISP-DM, conforme Figura 2, está fundamentado em seis fases distintas:

- **Entendimento do negócio:** foca no entendimento do negócio, analisar o que o cliente realmente precisa.
- **Entendimento dos dados:** objetiva, com uma coleção inicial de dados, procedimentos que buscam trazer familiaridade com o dado;
- **Preparação dos dados:** a fase de preparação cobre todas as atividades para a construção do conjunto de dados finais dos dados brutos iniciais.
- **Modelagem:** enquanto possivelmente já foi selecionada uma ferramenta para compreender o negócio, esta tarefa se refere à seleção e aplicação de uma ou mais técnicas específicas e os parâmetros destas técnicas são ajustados para otimização.
- **Avaliação:** neste estágio do projeto, deverá haver um modelo construído (ou vários) que precisam apresentar alta qualidade para uma perspectiva de análise dos dados.
- **Implantação:** a criação do modelo não é geralmente o fim do projeto. Mesmo que o propósito do modelo seja incrementar o conhecimento sobre os dados, o conhecimento adquirido

precisa ser organizado e apresentado em um meio em que possa ser usado.

2. MINERAÇÃO DE DADOS

2.1 As Tarefas de DCBD

As tarefas de DCBD podem ser definidas como um conjunto de técnicas ou algoritmos que representam um tipo de conhecimento desejado do banco de dados sendo que cada tarefa vai requerer algoritmos diferentes para a extração do conhecimento. Em geral as tarefas de DM podem ser classificadas em duas categorias: **descritivas** e **preditivas** (HAN e KAMBER, 1999). Tarefas descritivas caracterizam as propriedades gerais dos dados nos bancos de dados. Tarefas preditivas atuam inferenciando nos dados correntes para fazer previsões.

2.1.1 Tarefas ou Mineração Descritiva

- **Caracterização:** é uma sumarização das características gerais ou traços do objeto alvo. Podemos dizer também que, dada possibilidade de que um grande número de dados pode estar armazenado, seria útil poder descrever análises de forma concisa e sucinta em vários níveis de abstração; facilitaria examinar o comportamento geral dos dados.

- **Comparação ou Discriminação:** uma comparação das características gerais do conjunto de dados do objetivo alvo com um ou mais conjuntos com características opostas, como, por exemplo, comparar as características de um determinado produto que teve suas vendas incrementadas em 10% com aqueles produtos que tiveram perda de 30% no mesmo período. Os produtos resultantes desta análise são os mesmos da caracterização.

- **Associação:** a mineração de regras de associação é uma abordagem descritiva para a exploração dos dados que pode ajudar na identificação de relacionamentos entre valores ou transações em um banco de dados. Um exemplo típico de regra de associação é análise da cesta de mercado (*market basket analysis*).

▪ **Segmentação e Cluster Analysis:** o objetivo da análise de segmentação é agrupar os dados de forma que os grupos formados sejam muito diferentes entre si mas que os membros destes grupos sejam semelhantes. Diferentemente da classificação, não é possível saber quais características os conjuntos de dados segmentados terão quando o processo for iniciado, ou como serão os atributos dos dados após serem segmentados. Em consequência disto, será preciso que alguém que conheça o negócio interprete os segmentos gerados. (TWO CROWS CORPORATION, 1999)

2.1.2 Tarefas Preditivas:

▪ **Classificação:** problemas de classificação servem para identificar as características que indiquem o grupo para o qual cada caso pertence. Este padrão pode ser usado tanto para entender o dado existente quanto para prever como uma nova instância se comportará. DM cria modelos de classificação para examinar dados já classificados (casos) e através de indução encontrar um padrão existente.

▪ **Regressão:** a regressão utiliza valores existentes para prever outro valor. Em casos simples, a regressão usa técnicas estatísticas padrão como regressão linear.

2.1.3 Acuracidade do Classificador

O principal objetivo a ser atingido pelos modelos classificadores está na sua capacidade de diagnosticar corretamente casos nunca vistos, sendo que assim podemos definir a sua acuracidade em quanto bem ele representa a realidade do problema. Existem várias técnicas de estimar a performance de um modelo treinado, sendo que algumas podem ser melhores que outras. A taxa de erro é a mais utilizada para medir a performance de um classificador e pode ser representada através da equação 2.1:

$$\text{taxa de erro} = \frac{\text{número de erros}}{\text{número de casos}} \quad (2.1)$$

Os métodos holdout, random subsampling e validação cruzada são as

técnicas mais comuns para avaliar a acuracidade do classificador, baseadas em partes de amostras randômicas de um determinado conjunto de exemplos.

2.2 Técnicas de DM

As técnicas de DM são grupos de soluções ou algoritmos que são usados para responder aos problemas propostos nas tarefas. Cada tarefa apresenta várias técnicas que podem também ser utilizada para solucionar tarefas diferentes. A escolha destas técnicas precisa levar em consideração o objetivo final da DCBD.

A mineração de dados possui não somente um amplo espectro de aplicações, como também de técnicas, algoritmos e procedimentos. Diversas áreas, apresentadas a seguir, estão envolvidas para o desenvolvimento de algoritmos de DM:

▪ **Redes Neurais:** é uma técnica computacional que constrói um modelo matemático, emulado por computador, de um sistema neural biológico simplificado, com capacidade de aprendizado, generalização, associação e abstração. (ICA, 1999)

▪ **Algoritmos Genéticos:** são modelos estocásticos¹⁷ e probabilísticos de busca e otimização, inspirados na evolução natural e na genética, aplicados a problemas complexos de otimização. Têm sido empregados em DM para as tarefas de classificação e descrição de registros, além da seleção de atributos que melhor caracterizem o objetivo da tarefa de KDD. (ICA, 1999)

▪ **Métodos Estatísticos:** existem diversos métodos estatísticos, sendo alguns clássicos (regressão linear e múltipla, clusterização, etc.) e outros mais recentes, que assumem a existência de uma variável (atributo) resposta y, e uma coleção de variáveis preditoras x, além da disponibilidade de dados para treinamento. Entre as técnicas estatísticas podemos citar as

¹⁷ O oposto de determinístico. Ao invés de assumir que os seus dados assumem um determinado valor, você assume que esses dados possuem uma determinada distribuição probabilística.

Redes Bayesianas e Árvores de Decisão. (ICA, 1999)

A ênfase será para a técnica de árvore de decisão. Os motivos estão fundamentados na escolha da ferramenta de DM integrada a um SGBD comercial (MS SQL Server) e também nos objetivos propostos na introdução.

3 ESPECIFICAÇÃO DE PROVEDORES DE RECURSOS DE DM

Segundo Curotto (2003), vários projetos foram desenvolvidos para a integração da tecnologia de DM com SGBD's, podendo destacar:

- Projeto Quest: projeto desenvolvido pelo centro de pesquisas da IBM®;
- Projeto dbMiner: iniciado por um grupo de desenvolvedores da Universidade Simon Fraser de Burnaby, British Columbia, Canadá que desenvolveu uma série de trabalhos de integração de técnicas de DM com OLAP. Após várias contribuições, este projeto evoluiu em 2002 para um produto comercial;
- Tecnologia OLE DB DM: com a colaboração de vários pesquisadores foi elaborado e publicado, em Julho de 2000, a especificação da tecnologia OLE DB DM, cuja especificação define um padrão industrial para DM tal que diferentes algoritmos de DM implementados por diversos desenvolvedores possam ser facilmente embutidos em aplicativos de usuários, especificando a API (*Application Programming Interface*) entre consumidores de DM (aplicativos que utilizam recursos de DM) e provedores de recursos de DM (pacotes de *software* que fornecem algoritmos de DM).

3.1 Tecnologia OLE DB DM

Há um grande interesse comercial em minerar informações de DW, mas construir aplicações de DM para bancos de dados relacionais não é uma tarefa fácil e requer um trabalho significativo. Neste sentido, foi desenvolvido a API "OLE DB for Data Mining" (OLE DB DM) que é uma API padrão para o

desenvolvimento de recursos de DM, que possibilita a portabilidade destes provedores que podem aproveitar os recursos dos SGBD's. (NETZ *ett al*, 2001)

As principais propostas do desenvolvimento do OLE DB DM foram: amenizar os problemas de desenvolvimento de modelos e facilitar a preparação de dados por trabalhar diretamente em dados relacionais. Permitir que desenvolvedores de aplicações participem na construção de soluções de DM. Possibilitando desenvolver soluções integradas que são críticas para o crescimento da tecnologia de DM no espaço empresarial. O DM precisa ser visto como um componente que agrega valor junto com as tradicionais técnicas de suporte à decisão como o SQL tradicional e o ambiente de consultas OLAP. Possibilitar que desenvolvedores possam se sentir confortáveis uma vez que é comum usar bancos de dados com ferramentas de API baseadas em SQL, OLE DB¹⁸ e outros padrões conhecidos de protocolos.

3.1.2 Filosofia Básica do OLE DB DM

NETZ *ett al* (2001), descreveram a filosofia básica e as decisões de projeto que culminaram com a especificação OLE DB DM. Fundamentalmente, são necessárias operações que suportem modelos de DM. Para isto se formulou quatro operações fundamentais que devem ser suportadas por um provedor de recursos de DM:

- **Definir** um modelo de DM, identificando por exemplo o conjunto de atributos de dados a serem preditos, o conjunto de atributos de dados a serem utilizados para predição e o algoritmo utilizado para construir o modelo de DM;
- **Popular**¹⁹ um modelo de DM utilizando o algoritmo especificado com os dados de treinamento;

¹⁸ OLE DB consiste de uma especificação orientada a objeto para que um conjunto de dados acesse interfaces construídas para depósitos de dados orientados a registros.

¹⁹ Refere-se ao processo de apresentar os dados ao modelo criado.

- **Predizer** os atributos para novos dados utilizando um modelo de DM que foi treinado;
- **Expor o modelo** de DM para aplicativos de visualização, de geração de relatórios e de outras tarefas do processo KDD tais como interpretação e avaliação de resultados.

No OLE DB DM, um DMM é tratado como um “objeto de primeira classe”, assim como uma tabela. Nesta operação o foco é a definição (criação) de DMM, onde se descreve as colunas dos dados, com meta-informações e os relacionamentos entre as colunas (se assume que um *dataset* é representado como tabelas aninhadas) e outras operações que DMM talvez suportem.

Utilizando o exemplo utilizado por Han e Kamber (1999), pode-se ilustrar (Figura 3) a sintaxe para a criação do DMM, onde identifica as colunas de origem utilizadas, a coluna a ser predita e o algoritmo a ser utilizado. No anexo A, são apresentadas as especificações de tipos de colunas e atributos para a criação de DMM.

```
CREATE MINING MODEL [nome do modelo]
(
[registro] LONG KEY,
[Idade] TEXT DISCRETE,
[Rendimentos] TEXT DISCRETE,
[Estudante] TEXT DISCRETE,
[ClassCredito] TEXT DISCRETE,
[Comprador] TEXT DISCRETE PREDICT
)
USING algoritmo usado pelo DMM
```

Figura 3 – Exemplo de criação de um DMM, usando a notação proposta pelo padrão OLE DB DM.

Uma vez que o modelo está definido, o próximo passo é popular o modelo através de um *caseset* que satisfaça a especificação da criação do DMM declarado. Em OLE DB DM, é usado INSERT para instanciar o DMM. Particularmente a inserção corresponde ao consumo das observações representadas por um caso usando do DMM. A Figura 4 ilustra a sintaxe de povoamento de um DMM.

```
INSERT INTO [nome do modelo]
(
SKIP,
[Idade],
[Rendimento],
[Estudante],
```

```
[ClassCredito],
[Comprador]
)
OPENROWSET
(
'SQLOLEDB.1',
'Provider=SQLOLEDB;
Integrated Security=SSPI;
Persist Security Info=False;
Initial Catalog='''
Source=CLC',
'SELECT "Registro", "Idade", "Rendimento",
"Estudante", "ClassCredito", "Comprador" FROM
"tabela de origem")
```

Figura 4 – Exemplo de operação de inserção de um DMM, usando a notação proposta pelo padrão OLE DB DM.

Após ter sido povoado, a operação básica de obter predições de um novo conjunto de dados usando um DMM, é através de um “*prediction join*” entre o modelo e o novo conjunto de dados. A Figura 5 ilustra a sintaxe da junção com um DMM.

```
SELECT FLATTENED
[T1].[Registro],
[T1].[Idade],
[T1].[Rendimento],
[T1].[Estudante],
[T1].[ClassCredito],
[T1].[Comprador],
[nome do modelo].[Comprador] as [Valor da
Predicao]
FROM [nome do modelo] PREDICTION JOIN
OPENROWSET ('SQLOLEDB.1',
'Provider=SQLOLEDB.1;
Integrated Security=SSPI;
Persist Security Info=False;
Initial Catalog=[nome do modelo];
Data Source=CLC',
'SELECT "Registro", "Idade", "Rendimento",
"Estudante",
" ClassCredito ", "Comprador" FROM "[nome do
modelo]"
ORDER BY "Registro") AS [T1] ON
[nome do modelo].[Registro] = [T1].[Registro] AND
[nome do modelo].[Idade] = [T1].[Idade] AND
[nome do modelo].[Rendimento] =
[T1].[Rendimento] AND
[nome do modelo].[Estudante] = [T1].[Student]
AND
[nome do modelo].[ClassCredito] = [T1].[
ClassCredito] AND
[nome do modelo].[Comprador] = [T1].[Comprador]
```

Figura 5 – Exemplo de uma junção com um DMM, usando a notação proposta pelo padrão OLE DB DM.

A melhor e mais popular maneira de apresentar o conteúdo de um DMM é visualizando-o através de um gráfico (uma árvore de decisão sendo representada figurativamente na estrutura de uma árvore, um cluster na representado por um cilindro, etc.). Atualmente, visando à portabilidade e o intercâmbio de

informações o armazenamento é realizado através de uma especificação chamada de PMML (*Predictive Model Markup Language*)²⁰. O PMML especifica um formato de persistência para DMM.

4 ESTUDO DE CASO

4.1 Análise do negócio

Através da compreensão e análise das principais necessidades do negócio, conforme especificado na introdução deste trabalho, foi possível compreender alguns problemas que interferem diretamente na DCBD do setor de telemarketing ativo:

- Devido à necessidade de se trabalhar com *mailings*²¹ fornecidos geralmente por clientes, onde normalmente se exige que todos os nomes sejam trabalhados, não faz sentido tentar classificá-los em possíveis compradores ou não. Isto se deve principalmente ao fato que, para cada possível cliente (*prospect*), independente de fatores como renda, idade, etc., normalmente há um produto adequado ao seu perfil. Outra razão importante está no fato destes *mailings* normalmente já terem sido filtrados pelos seus fornecedores de acordo com o objeto de venda.

- Normalmente os clientes de empresas de telemarketing ao adquirirem seus *prospects*, pagam pelos mesmos. Pode-se afirmar que o preço destes registros está ligado diretamente à sua qualidade. *Prospects* com informações sócio-econômicas custam muito mais caro que aqueles que possuem somente nome e telefone.

Devido às situações descritas acima, a principal necessidade de quem trabalha com vendas de forma ativa, isto é, ligando para o *prospect*, se torna em encontrá-lo. Pode-se dizer que é preciso construir modelos que possam atender estas necessidades, além de outras inerentes à estrutura de um *call center*:

- Robustez: devido à quantidade de consumidores destes modelos (milhares de acessos simultâneos);

- Portabilidade: possa atender a diferentes aplicações consumidoras.

- Fácil manutenção;

De fato, têm-se ainda inúmeras necessidades para serem atendidas, não somente devido à heterogeneidade de produtos que podem ser ofertados como também ao grande número de segmentos que podem ser atendidos pelo *telemarketing* ativo, como por exemplo, venda de assinaturas de revistas e jornais, seguros, etc.

4.2 Recursos tecnológicos utilizados

Um ponto chave do sucesso de DM em bases de dados de telemarketing, trata-se da exploração e análise de forma automática ou semi-automática em grandes bases de dados. Este aspecto se torna imprescindível devido a dois fatores:

- Pela grande quantidade de dados que uma operação pode gerar.

- É preciso que os *prospects* estejam disponíveis de forma mais rápida possível. Este processo precisa ser *on-line*.

A estratégia é o desenvolvimento de uma interface entre o analista de DM, o avaliador e consumidor dos DMMs.

4.2.1 Microsoft® SQL Server™ 2000 Analysis Services

Coincidindo com o lançamento da especificação do OLE DB for Data Mining 1.0, o SGBD Microsoft® SQL Server™ 2000 integrou funcionalidades de DM junto com bancos relacionais e OLAP. O Analysis Services, componente do SQL Server 2000 se apresenta como um provedor para a mineração de dados baseado na especificação OLE DB DM. Esse provedor inclui dois algoritmos de DM: árvores de decisão e cluster Microsoft. A Figura 4.1 apresenta uma visão geral dos componentes presentes na arquitetura Microsoft para DM.

4.2.1.1 Classificador MSDT (*Microsoft Decision Tree*)

²⁰ Disponível em: <http://www.dmq.org/index.html>. Acesso em 10/10/2004 às 10:00

²¹ Refere-se ao conjunto de nomes de possíveis compradores, matéria-prima do telemarketing ativo.

Segundo Chaudhuri (1999), além de ser a técnica mais popular para a modelagem de prognóstico, a escolha de implementação de árvores de decisão no Analysis Services se deveu ao fato de serem amplamente estudadas em estatística, reconhecimento de padrões e aprendizado de máquina e por poderem ser examinadas e interpretadas facilmente.

Há muitas variações de algoritmos que constroem árvores de decisão e que usam diferentes métodos de divisão: formas de árvore, técnicas de remoção, etc. Os algoritmos que fazem parte da árvore de decisão Microsoft utilizados para controlar o crescimento de uma árvore são:

- Entropia: baseado no ganho da entropia do classificador
- Ortogonal: baseado na ortogonalidade da distribuição de estados no classificador. Este método produz somente divisões binárias, resultando em árvores de grande profundidade;
- Bayesiano com K2: baseado no escore Bayesiano com K2 a priori;
- Bayesiano Dirichlet Equivalente com Uniforme a priori: método padrão descrito por Chickering *et al* (1994) apud Curotto (2003).

4.2.2 Decision Support Objects

Conforme Microsoft (2), o Microsoft® SQL Server™ 2000 Analysis Services provê funcionalidades de DM e OLAP. Por ter sido desenvolvido para ser flexível e extensível, possibilita adicionar serviços e pacotes de terceiros, como por exemplo, provedores de algoritmos de DM para estender suas funcionalidades. Para o acesso de uma forma simples a estas funcionalidades, a biblioteca DSO (*Decision Support Objects*) fornece uma modelagem hierárquica de objetos para serem usados com ambientes de desenvolvimento que suporte objetos e interfaces COM (*Component Object Model*).

Conceitualmente, DSO utiliza um grupo de objetos arranjados de forma hierárquica para definir o elemento base de armazenamento de dados do Analysis Services. Estes elementos bases são

databases, data sources, dimensions, cubes, data mining models e roles. O DSO mantém esses elementos básicos em uma estrutura hierárquica onde cada elemento contém outros elementos em uma estrutura de árvore, sendo que o objeto Server, é a raiz dessa árvore. A

Uma seqüência comum de operações para uma aplicação de DM que utiliza DSO é:

- Conectar a um servidor Analysis Services;
- Criar um objeto *database*.
- Adicionar um *data source* que indica a origem dos dados;
- Criar um DMM especificando seus parâmetros e a origem dos dados;
- Criar as colunas do DMM com suas propriedades;
- Processar o DMM.

4.4 Desenvolvimento da aplicação

Utilizando-se das tecnologias apresentadas, desenvolveu-se uma *interface* experimental para permitir que objetos do provedor de DMM (Analysis Services) fossem instanciados em uma aplicação cliente, permitindo a customização do acesso, criação, manipulação e visualização dos DMMs.

As principais motivações para esta personalização foram fundamentas nas necessidades características do *telemarketing*, e na possibilidade de integrar outros processos de DM como:

- Importação, tratamento e transformação de dados outras fontes como planilhas eletrônicas, arquivos texto, etc. pelo usuário cliente;
- Integração com aplicações distribuídas para avaliação de modelos e consultas;
- Análises descritivas visuais de dados como média, desvio padrão, etc. utilizando instruções SQL.

Os benefícios imediatos desta estratégia podem ser citados:

- Não haver necessidade de instalações adicionais do Analysis Manager;
- Democratização de conhecimento pela necessidade de dividir as ações e;

▪ Conseqüente valorização da tecnologia.

Após os processos de criação e povoamento do DMM, utilizando-se de componentes gráficos disponíveis (MSN Groups, 2004), possibilitou-se a uma visualização fácil dos padrões do DMM processado conforme Figura 6.

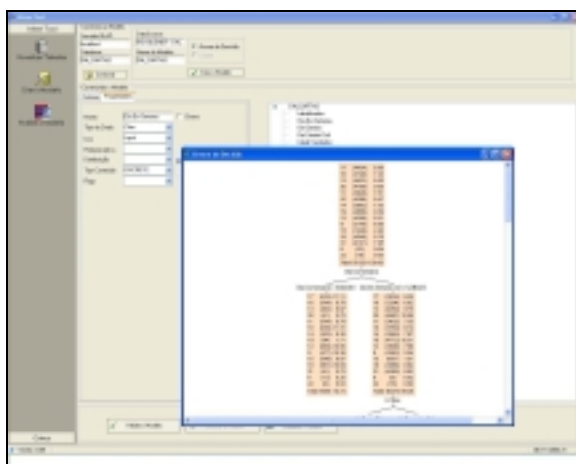


Figura 6 – Visualização do DMM através de componentes gráficos.

Um aspecto importante na integração de SGBDs com ferramentas de DM é a possibilidade de consumir os modelos considerando-os como tabelas. Assim, através de interfaces de conexão com o provedor de DMM, consumir os modelos através de junções possibilita que aplicações customizadas possam se beneficiar do DM para automatizar processos como exemplificado na Figura 7.

CONCLUSÃO

Através do conteúdo apresentado possibilitou-se a criação de uma ferramenta para a automação de processos de DM no ambiente comercial estudado. Os pontos fortes desta abordagem estão centrados na tecnologia OLE DB DM e nos demais recursos disponíveis para a integração das atividades de DM, demonstrando a viabilidade para uma integração eficiente com SGBDs comerciais. No entanto para a construção de modelos de alta qualidade, necessita-se agregar práticas de planejamento e construção de modelos através de uma correta especificação

desde seu planejamento até a apresentação dos resultados obtidos. Proposta do padrão CRISP-DM. Para atender às expectativas de um segmento específico de consumidores de DM, desenvolveu-se um produto segundo suas necessidades e características, apresentando-se uma solução inovadora para o segmento de *telemarketing*.

REFERÊNCIAS BIBLIOGRÁFICAS

- AMARAL, Fernanda Cristina Naliato do. **Data mining, técnicas e aplicações para o marketing direto**. Editora Berkeley, 2001.
- BATISTA, Gustavo Enrique de Almeida Prado Alves. **Um ambiente de avaliação de algoritmos de aprendizado de máquina utilizando exemplos**. São Paulo. Dissertação (Mestrado em Ciências Matemáticas). Instituto de Ciências Matemáticas de São Carlos, Universidade de São Paulo. São Paulo, 1997.
- CARVALHO, Luís Alfredo Vidal de; **Data mining: a mineração de dados no marketing, medicina, economia, engenharia e administração**; São Paulo: Érica, 2001.
- CHAUDHURI, Surajit. Data mining and database systems: where is the intersection? **Bulletin of the IEEE Computer Society Technical Committee on Data Engineering**. Março, 1998.
- CHAUDHURI, Surajit; USAMA Fayyad; BERNHARDT, Jeff. Scalable Classification over SQL Databases. **Proceedings of 15th International Conference on Data Engineering, Sydney, Australia**. 1999.
- CRISP-DM 1.0 - Cross-Industry Standard Process for Data Mining. Step-by-step data mining guide. SPSS 2000. Disponível no endereço: <<http://www.crisp-dm.org/>>. Acesso em: 09/02/2004 às 15:00h.
- CURROTT, Cláudio Luiz. **Integração de recursos de data Mining com gerenciadores de bancos de dados relacionais**. Rio de Janeiro. Tese (Doutorado em Ciências de Engenharia Civil). Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2003;
- PETERSON, Timothy. **Microsoft SQL Server 2000 (DTS)**. Tradução Edson Furmankiewicz, Joana Figueiredo. – Rio de Janeiro: Campus, 2001
- HAN, Jiawie e KAMBER, Micheline. **Data mining: concepts and techniques**. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor Morgan Kaufmann Publishers, 1999.
- ICA – Laboratório de inteligência Computacional Aplicada. Descoberta de conhecimentos e mineração de dados. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica – PUC/ Rio de Janeiro, 1999.
- SANT'ANNA, Mauro. **C#: A nova linguagem da arquitetura .NET**. 2001. Disponível em: <http://www.linhadecodigo.com.br/artigos.asp?id_ac=15>. Acesso em: 15/10/2004 às 18:00.
- MICROSOFT CORPORATION, **OLE DB for Data Mining Specification Version 1.0**, Microsoft Corporation, Redmond, Washington, USA, 2000. Disponível em: <<<http://www.microsoft.com/downloads/details.aspx?FamilyID=01005f92-dba1-4fa4-8ba0-af6a19d30217&DisplayLang=en>>> Acesso em: 10/09/2004 às 23:00h.
- MICROSOFT CORPORATION, **Decision Support Objects Architecture**. MSDN Library.

Microsoft Corporation, Redmond, Washington, USA, 2000.
Disponível em: <
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmpr/prabout_27hh.asp> Acesso em:
08/08/2004 às 17:00h.

NETZ, Amir; CHAUDHURI, Surajit; BERNHARDT, Jeff; FAYYAD, Usama. Integration of data mining and relational databases, **Proceedings of the 26th International Conference on Very Large Databases**, Cairo, Egypt, 2000.

NETZ, Amir; Bernhardt Jeff; CHAUDHURI Surajit; FAYYAD, Usama. Integrating data mining with SQL databases: OLE DB for data mining. **Proceedings of 17th International Conference on Data Engineering**. Heidelberg, Germany, 2001.

PAUL, Seth; GAUTAM, Nitin; BALINT, Raymond. **Preparing and mining data with Microsoft SQL Server 2000 and Analysis Services**. Online Books, Microsoft SQL Server Series. Maio, 2004.

RUD, Olivia Parr. **Data mining cookbook modeling data for marketing, risk, and customer relationship management**. Wiley Computer Publishing, November, 2000

TWO CROWS CORPORATION. **Introduction to data mining and knowledge discovery**. Third Edition, 1999. Disponível em: < <http://www.twocrows.com/intro-dm.pdf>>. Acesso em 20/01/2004 às 12:00h.

