CENTRO UNIVERSITÁRIO FEEVALE

INGO JOST

MINERAÇÃO DE DADOS PARA ADOÇÃO DE PRÁTICAS DE MARKETING EM AMBIENTE ACADÊMICO

Novo Hamburgo, novembro de 2008.

INGO JOST

MINERAÇÃO DE DADOS PARA ADOÇÃO DE PRÁTICAS DE MARKETING EM AMBIENTE ACADÊMICO

Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Ciência da Computação
Trabalho de Conclusão de Curso

Professor Orientador: Juliano Varella de Carvalho

Novo Hamburgo, novembro de 2008.

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial:

Aos amigos que sempre pude contar, à minha família, à minha namorada Marilaine, minha gratidão, pelo apoio emocional. .

Agradeço também ao professor e amigo Juliano pela orientação e dedicação.

RESUMO

Este trabalho apresenta o projeto que visa utilizar a área de Data Mining, disseminada nos mais diversos segmentos, utilizada por empresas que buscam descobrir o conhecimento que se encontra oculto em suas próprias bases de dados a fim de possibilitar um diferencial a seus concorrentes, criando, desta forma, uma aplicação destinada a uma instituição de ensino superior. A base de dados desta instituição é composta por informações de vestibulandos, como município de residência, renda familiar, sexo, idade, entre outros. Dentre as diversas técnicas de Data Mining, a que será utilizada é a de Classificação, que possui diferentes algoritmos, em que os dados serão tratados e após a aplicação desses, serão classificados em perfis pré-definidos. O objetivo é encontrar relação entre os perfis de vestibulandos com a demanda por cursos. Assim, espera-se que, com o conhecimento adquirido, seja possível ao departamento de Marketing adotar práticas direcionadas a áreas ou cursos, com o objetivo de atrair mais alunos à instituição, o que seria um diferencial em um mercado cada dia mais disputado.

Palavras-chave: *Data Mining*. Técnicas de Classificação. Marketing de precisão. Mapeamento de Perfis

ABSTRACT

This work presents the project that will use Data Mining, spread on many segments, used for companies that aim to know the discovery that has been hidden in their databases to differ from their rivals, to build a application to high school institution. Its database is composed for students that have made vestibular's information, like address, familiar rent, gender, age and others. Among the Data Mining's techniques, the classification technique that will be used. Its has some algorithms. The data will be prepared to begin of application these algorithms and it will be classified into pre-defined profiles. The goal is to find relationship between the profiles and demand for courses. Thus, it is expected that with the knowledge, it is possible for the department of Marketing adopt practices directed to areas or courses, aiming to attract more students to the institution, which would be a gap in the market each day over disputed

Key words: Data Mining. Classification Technique. Precision Marketing. Discovery Profiles

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

BD	Banco de Dados
BI	Business Inteligence
CART	Classification and Regression Tree
CHAID	CHi-squared Automatic Interaction Detector
CIM	Customer Interaction Management
CRM	Customer Relationship Management
GUI	Graphical User Interface
IA	Inteligência Artificial
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
KDD	Knowledge Database Discovery
OSBI	Open Source Business Intelligence
RNA	Redes Neurais Artificiais
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
TDIDT	Top-Down Induction of Decision Trees
TI	Tecnologia em Informação
XML	eXtensible Markup Language

SUMÁRIO

INTRODUÇÃO

As corporações têm buscado, no decorrer das últimas décadas, digitalizar os seus dados e adquirir ou desenvolver sistemas de informação que manipulam e alimentam essas bases. No entanto, este gerenciamento das informações é utilizado normalmente apenas para a gestão e otimização de processos, não sendo aproveitado para uma análise mais detalhada, capaz de propiciar novas oportunidades às instituições.

Com esse objetivo, surge a área de Descoberta de Conhecimento em Bases de Dados, cujo nome é originado do inglês *Knowledge Discovey in Databases* (KDD). É um processo composto pelas seguintes etapas operacionais:

- Pré-processamento: etapa em que os dados recebidos passam por uma preparação para a próxima etapa. Esta preparação consiste na seleção, excluindo os dados irrelevantes e os que possuem inconsistências e transformação destes, sendo realizadas conversões e adaptações.

- Mineração de Dados: quando é realizada a busca propriamente dita pelo conhecimento. Existem diversos algoritmos que implementam diferentes técnicas para a obtenção do conhecimento, destacando-se Associação, Classificação e Agrupamento.

- Pós-processamento: aproveitamento das informações adquiridas, sendo realizada a interpretação e avaliação da importância do conhecimento descoberto, se houver.

A Mineração de Dados é utilizada nas mais diversas áreas, desde a descoberta de pesos de atributos em um sistema de Raciocínio Baseado em Casos, conforme Silveira (2003), até, de acordo com Neves (2003), a definição de padrões em pacientes de Diabetes.

Independente do foco, são utilizadas as técnicas já citadas, inclusive os mesmos algoritmos. Dentre estas técnicas, destacam-se:

- Agrupamento: procura separar em grupos dados similares
- Classificação: consiste em classificar os registros em categorias (classes) pré-definidas
- Associação: busca associações entre atributos em diferentes transações

Existem diversas ferramentas que implementam as técnicas de KDD, destacando-se Sipina, desenvolvida pela Universidade de Lyon e Weka, sendo ambas utilizadas neste projeto. Através das técnicas e tecnologias apresentadas, tem-se como motivação deste trabalho a criação de uma ferramenta que extraia conhecimento da base de dados de um centro universitário. Esta instituição, que conta com milhares de alunos, procura uma solução para melhor distribuir seus esforços em relação ao Marketing, área que vem sendo bastante explorada por aplicações de *Data Mining*, como verificado em *KDNuggets*[1], portal referência em KDD.

Dentre os exemplos na área, destacam-se processos de relacionamento de clientes, conforme Almeida (2002), ou de mapeamento de perfis, como a empresa *Sigma*[2] que possui uma ferramenta com diversos canais de comunicação para que as empresas conheçam as características de seus consumidores e o *Talisma Knowledgebase Software*[3], que permite direcionar o contato da empresa com o cliente, a partir de transações já realizadas, evitando gastos com comunicação.

Este projeto manipulará informações pessoais de alunos, procurando peculiaridades comuns em, por exemplo: escolha de curso, município de residência, idade, escolas freqüentadas. Serão mapeados perfis de vestibulandos para que se encontre associações entre as características dos alunos e a escolha por cursos, possibilitando ao departamento de Marketing adotar práticas direcionadas para institutos, como o de Ciências Exatas, ou especificamente para cursos.

---

1 http:// www.kdnuggets.com/
2 http://www.sigmamarketing.com/
3 http://www.talisma.com/tal_products/knowledgebase.aspx

Fica a total critério da equipe de Marketing a forma de aproveitamento do conhecimento adquirido, juntamente com as práticas, que podem ser desde o direcionamento de propagandas até o agendamento de visitas de apresentação conforme a escola ou região de maior procura pelo curso (ou menor).

1 DESCOBERTA DE CONHECIMENTO

A área de Descoberta de Conhecimento em Bases de Dados, cujo nome é originado do inglês *Knowledge Discovey in Databases* (KDD), busca através de dados já cadastrados, aplicar determinadas técnicas que extraem conhecimento. Este conhecimento pode ser utilizado na tomada de decisões ou posicionamento estratégico das instituições.

Dada a grande distinção entre as técnicas, identifica-se na Descoberta de Conhecimento, de acordo com (FREITAS, 1998) uma grande interdisciplinaridade, envolvendo pelo menos três grandes áreas: Estatística, Inteligência Artificial e Banco de Dados. O processo de KDD é composto por diversas etapas, sendo ilustradas na Figura 1.1

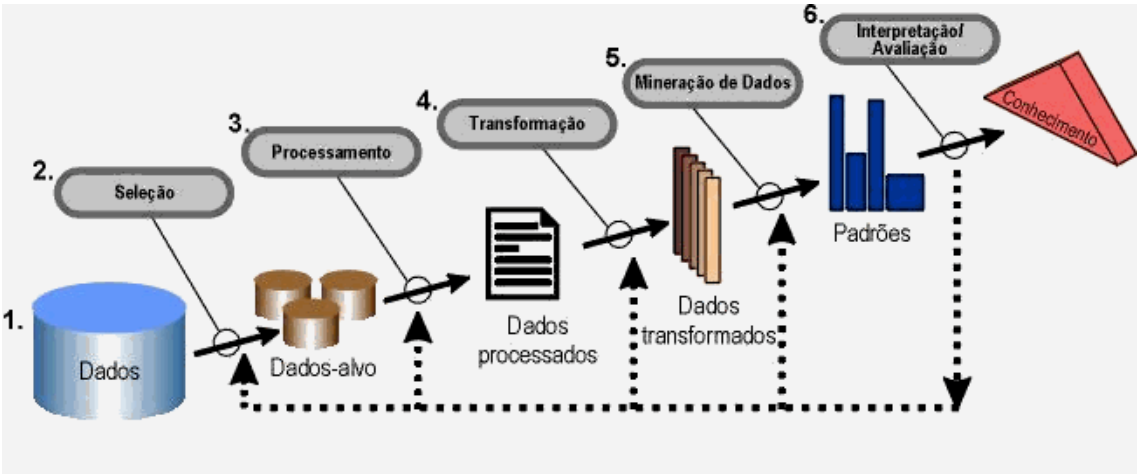


Figura 1.1 – Etapas do processo de KDD
Fonte: ROMANI.

Segundo (GOLDSCHMIDT, 2005), a Descoberta do Conhecimento é composta pelas etapas operacionais: Pré Processamento, Mineração de Dados e Pós-Processamento.

1.1 Pré Processamento

Dentre as etapas da Figura 1.1, Seleção, Processamento e Transformação se enquadram na fase de Pré-Processamento. A seleção consiste na filtragem dos dados, excluindo aqueles irrelevantes, por exemplo, no mapeamento de perfis de clientes, provavelmente nome e CPF são atributos irrelevantes. .

O processamento consiste em executar operações como as de limpeza de dados, evitando inconsistências decorrentes de erros de cadastro (salário ou idade com valores negativos) e tratamento de campos com grande ocorrência de valores nulos, por serem dados não disponíveis ou não informados. É necessário avaliar se devem ser desconsiderados ou receber um valor padrão.

Em relação à transformação, conforme Mongiovi (1998), destaca-se a conversão de dados (para aqueles dados com o mesmo significado, porém em formatos diferentes, por exemplo, o campo *sexo* sendo representado ora por números 1 e 2, ora por caractere 'M' e 'F'), criar categorias para variáveis contínuas (categorias de faixas etárias, ao invés de idade), converter variáveis nominais em numéricas, usar escalas de redução / ampliação (normatizar valores com unidades de medida diferentes) e criar novas variáveis. Estes processos são necessários em maior intensidade quando os dados são oriundos de diferentes bases ou até de diferentes formatos, como banco de dados relacional e documentos em *eXtensible Markup Language* (XML).

Além disso, a transformação dos dados é necessária para que estes fiquem no formato de entrada dos algoritmos de *Data Mining,* como por exemplo, utilizar consultas *Structured Query Language* (SQL) e a partir do resultado executar procedimentos que criam arquivos neste formato. A ferramenta *Weka,* que será apresentada posteriormente, tem como entrada de dados, arquivos no formato .ARFF (*Attribute-Relation File Format*).

```
@relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal}
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no
```

Figura 1.2: Exemplo de Arquivo .arff
Fonte: SANTOS, 2005

1.2 Mineração de Dados

Etapa mais importante da Descoberta de Conhecimento (GOLDSCHMIDT, 2005). Sua relevância é tamanha, que o termo Mineração de Dados confunde-se com KDD. É nesta etapa que se realiza a busca propriamente dita pelo conhecimento, extraindo padrões dos dados. Existem diversos algoritmos que implementam diferentes técnicas para a obtenção do conhecimento. No entanto, somente a utilização das técnicas não é suficiente para atingir os resultados esperados. Para utilização de KDD, é necessário o envolvimento de um especialista na área desde o momento da identificação do problema, passando por todas as etapas descritas, juntamente com especialistas do domínio da aplicação.

Entre as principais técnicas de Mineração de Dados, destacam-se: Associação, Classificação e Agrupamento.

1.2.1 Associação

Procura por associações entre os atributos de um conjunto de dados, conforme Frank e Witten (2005). O exemplo mais comum para ilustrar esta técnica é o de transações de compras: neste caso obtém-se regras de associação com a ocorrência de produtos em diferentes transações. Por exemplo, um mercado pode auferir que 80% dos compradores de

pão, também compram leite. A partir destes conhecimentos descobertos, diversas ações podem ser tomadas, como por exemplo, deixar esses produtos próximos (ou distantes), promover a venda conjunta dos produtos, reduzir o preço de um deles e aumentar o preço do outro, dentre outras medidas mercadológicas cabíveis.

Os algoritmos que descobrem regras de associação devem gerá-las de forma a atender parâmetros que são informados pelo usuário: suporte e confiança (AGRAWAL; SRIKANT, 1994). Suporte é o percentual de incidência da regra no conjunto de transações e confiança indica a validade da regra (GONÇALVES, 2008). Isto pode ser constatado, por exemplo, em um conjunto de transações de compra de produtos, com Suporte 60 e Confiança 30, para que a regra *{pão}→{café}* seja verdadeira, é necessário que em pelo menos 60% das transações os dois produtos estejam presentes e que no mínimo 30% das transações que tenham comprado *pão*, tenham adquirido *café* também.

As regras de associações podem ser ainda: "multidimensionais", em que atributos de diferentes tipos aparecem na regra, como o exemplo a seguir:

(Sexo = 'M') ^ (20<Idade<30) → {cerveja}

Esta regra indica que homens com idade entre 20 e 30 anos compram cerveja.

As regras também podem ser "híbridas", em que atributos de mesma dimensão aparecem várias vezes na regra:

(Sexo = 'M') ^ (20<Idade<30) ^ {café} → {leite}

No caso acima verifica-se que homens com idade entre 20 e 30 anos, que compram café, também adquirem leite.

Por fim, também existem as regras negativas, auferindo que o consumidor que compra os produtos *A* e *B,* não compra o produto *C*, por exemplo.

Há casos em que as associações podem ser descobertas a partir de generalizações, ou seja, os dados estarem em forma de hierarquia, como por exemplo, *cereal* é uma generalização de *arroz* e *feijão*, assim como *roupa* é de *camisa* e *calça*. Alguns algoritmos têm suporte para mineração de dados multi-nível. O algoritmo mais utilizado para a técnica de associação é o *Apriori* e suas variantes (AGRAWAL; SRIKANT, 1994).

Para ilustração da técnica de exemplo, será usada Tabela 1.1.

Tabela 1.1 – Conjunto de Transações

Transação	Café	Leite	Manteiga
1	1	1	0
2	0	1	1
3	1	0	0
4	1	0	0
5	1	1	0
6	1	1	1
7	1	1	1
8	0	1	0
9	1	1	1
10	0	1	0

Fonte: Do autor

Os algoritmos de associação partem do total de transações formando os possíveis conjuntos a estarem presentes nas regras. Seguindo o exemplo da Tabela 1.1, o algoritmo identificaria os seguintes conjuntos com dois elementos: {café, leite}, {leite, manteiga} e {café, manteiga}. Não são levados em consideração conjuntos com um elemento porque não se conseguiria elaborar uma regra.

A seguir, o algoritmo verifica se os conjuntos atendem ao suporte, no caso 40%. O conjunto {café, leite} ocorreu nas transações 1, 5, 6, 7 e 8, representando 50% das transações. O conjunto {leite, manteiga} ocorreu em 40 % dos casos (transações 2, 6, 7 e 9), o conjunto {café, manteiga} ocorreu nas transações 6, 7 e 8 (30 %). Sendo assim, apenas o primeiro e segundo conjuntos atendem ao suporte, sendo então verificado o fator confiança. Como o conjunto {café, manteiga} não atende ao suporte, conseqüentemente o algoritmo não identifica o conjunto {café, leite, manteiga}, por este possuir o subconjunto {café, manteiga}.

Dados os conjuntos {café, leite} e {leite, manteiga}, o algoritmo testará a confiança das seguintes possíveis regras: {café} → {leite}, {leite} → {café}, {leite} → {manteiga} e {manteiga} → {leite}. O teste consiste em verificar em quantas das ocorrências de café ocorre leite, das ocorrências que ocorrem leite ocorrem café e assim por diante. Das transações em que há café (1, 3, 4, 5, 6, 7, 9), em cinco ocorrem leite (1, 5, 6, 7 e 9) representando $5/7 \approx 71{,}4$ %. Seguindo o mesmo procedimento, as demais regras apresentam a confiança 62,5%, 50% e 100% (em todas as ocorrências de manteiga, há simultaneamente leite). Infere-se então, dada a Tabela 1, suporte de 40% e confiança 70%, as regras {café} → {leite} e {manteiga} → {leite}.

1.2.2 Agrupamento

Técnica que procura separar em grupos, registros com características as mais homogêneas possíveis, ou seja, que possuam propriedades comuns. Desta forma, a técnica de agrupamento também busca a maior distinção entre os grupos definidos (GOLDSCHMIDT, PASSOS, 2005).

Conforme Grégio (2007), os algoritmos de agrupamento varrem os dados, identificando grupos e associando os dados de entrada a estes. A implementação desses algoritmos pode ser na técnica de hierarquia, em que a base de dados é dividida em subconjuntos menores, até que os dados se encontrem em um único grupo (nodo raiz).

A outra técnica de agrupamento é o particionamento, em que o número de grupos é pré-definido, após estes serem criados os dados são agrupados de acordo com a similaridade em relação aos grupos. Entre os algoritmos de agrupamento, destacam-se o *K-Modes e K-Means*. Este algoritmo traça aleatoriamente valores a serem adotados como o centro dos *clusters* (grupos), calculando a distância de cada registro aos centróides, o associando ao grupo com menor distância.

1.2.3 Classificação

Técnica que consiste em classificar os registros em categorias (classes) pré-definidas, possibilitando, quando da inserção de novos registros, que estes já sejam classificados automaticamente (GOLDSCHMIDT, PASSOS, 2005). É necessária a construção de um modelo, a partir de um *conjunto de treinamento*, que é composto por registros utilizados como exemplo, para a posterior classificação das tuplas da base de dados. O modelo criado para a classificação pode ser representado por regras de classificação, árvores de decisão (CARVALHO, 2000), fórmula matemática ou redes neurais artificiais (RNA).

Segundo (FREITAS, 2000) a principal diferença entre a técnica de classificação e a de associação está no nível sintático (regras de classificação possuem somente um atributo na sua conclusão, enquanto as de associação permitem vários). Já em relação ao agrupamento, a principal diferença é a pré-definição das classes, enquanto os grupos são gerados em tempo de execução. A técnica de classificação será detalhada no Capítulo 2.

1.3 Pós-Processamento

Fase em que ocorre o aproveitamento das informações adquiridas, sendo realizada a interpretação e avaliação da importância do conhecimento descoberto, se houver. É verificada entre o especialista em KDD e o especialista da área da aplicação, a necessidade de repetir o ciclo de etapas.

Na etapa de pós-processamento, são tratados os resultados gerados pelos algoritmos de KDD, que podem estar em formatos não tão claros para o usuário, como informações de natureza estatística, conforme (CARVALHO, 2003).

1.4 Ferramentas

Existem diversas ferramentas *freeware* que permitem a utilização das técnicas de KDD, como RapidMiner4, Pentaho5 ,Weka6 (ferramenta open-source que pode ser acoplada a outras, como o próprio Pentaho) e Sipina7, especializada em árvores de classificação.

1.4.1 Weka

O pacote Weka (*Waikato Environment for Knowledge Analysis*) é composto por bibliotecas que implementam diversos algoritmos das técnicas de KDD. Inerente ao *software*, está a interface gráfica que permite a escolha da forma de apresentação dos resultados (gráficos, árvores). Não obstante, podem ser desenvolvidas aplicações que utilizam apenas a camada dos algoritmos de Mineração de Dados (WITTEN, FRANK, 2000).

Como visto na Figura 1.2 o formato de entrada da ferramenta Weka é arquivo .ARFF, sendo este composto de Relação, Atributos e Dados (SANTOS, 2005). Na ilustração, *relation* indica o nome da relação, já *attribute* indica um atributo com os seus possíveis valores entre "{}". Após a descrição dos atributos, há o conjunto dos dados (*data*), em que a cada linha possui os valores associados a cada atributo separados por ",".

---

4 http://www.rapidminer.com
5 http://www.pentaho.com
6 http:// www.cs.waikato.ac.nz/ml/weka/
7 http:// eric.univ-lyon2.fr/~ricco/sipina.html

O Weka oferece um grande conjunto de algoritmos a serem aplicados nesses arquivos, como demonstra a Figura 1.3, possibilitando a visualização dos resultados em diferentes formatos, como árvores de decisão e gráficos (Figura 1.4).

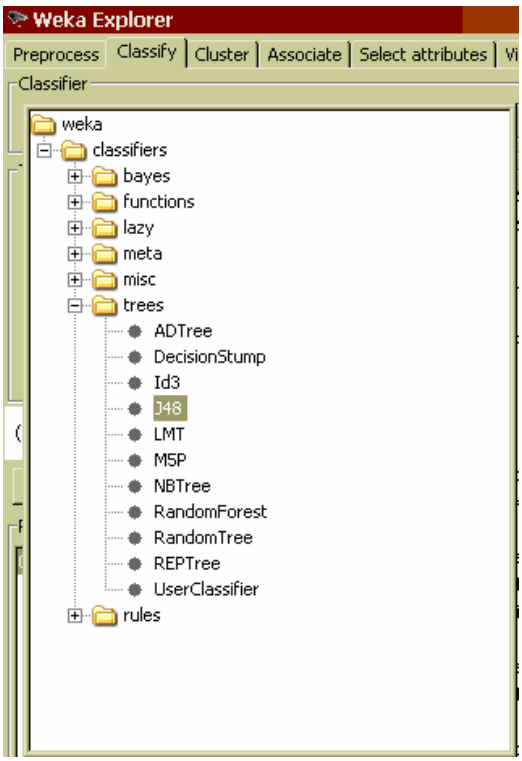


Figura 1.3: Opções do Weka
Fonte: Do autor

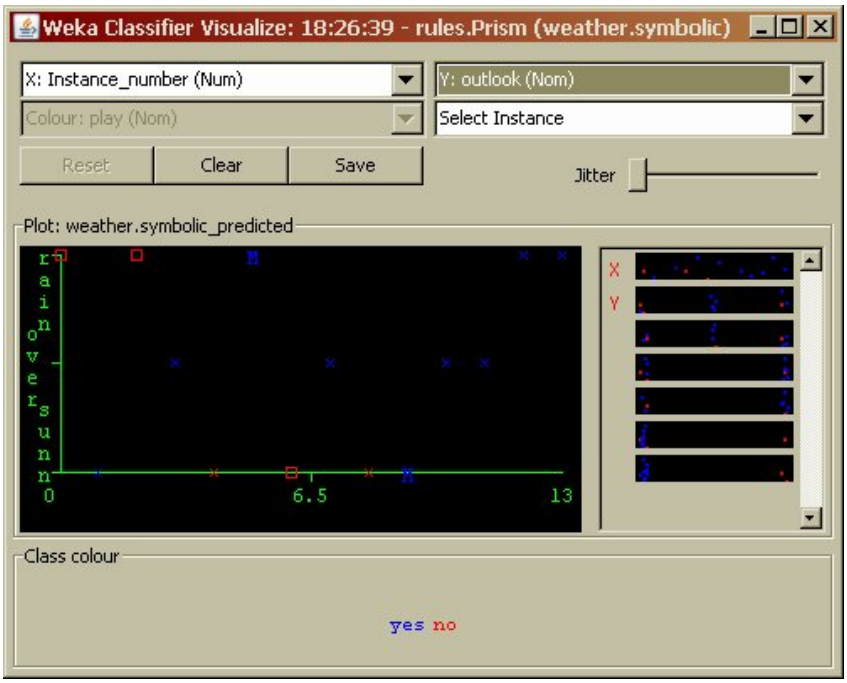
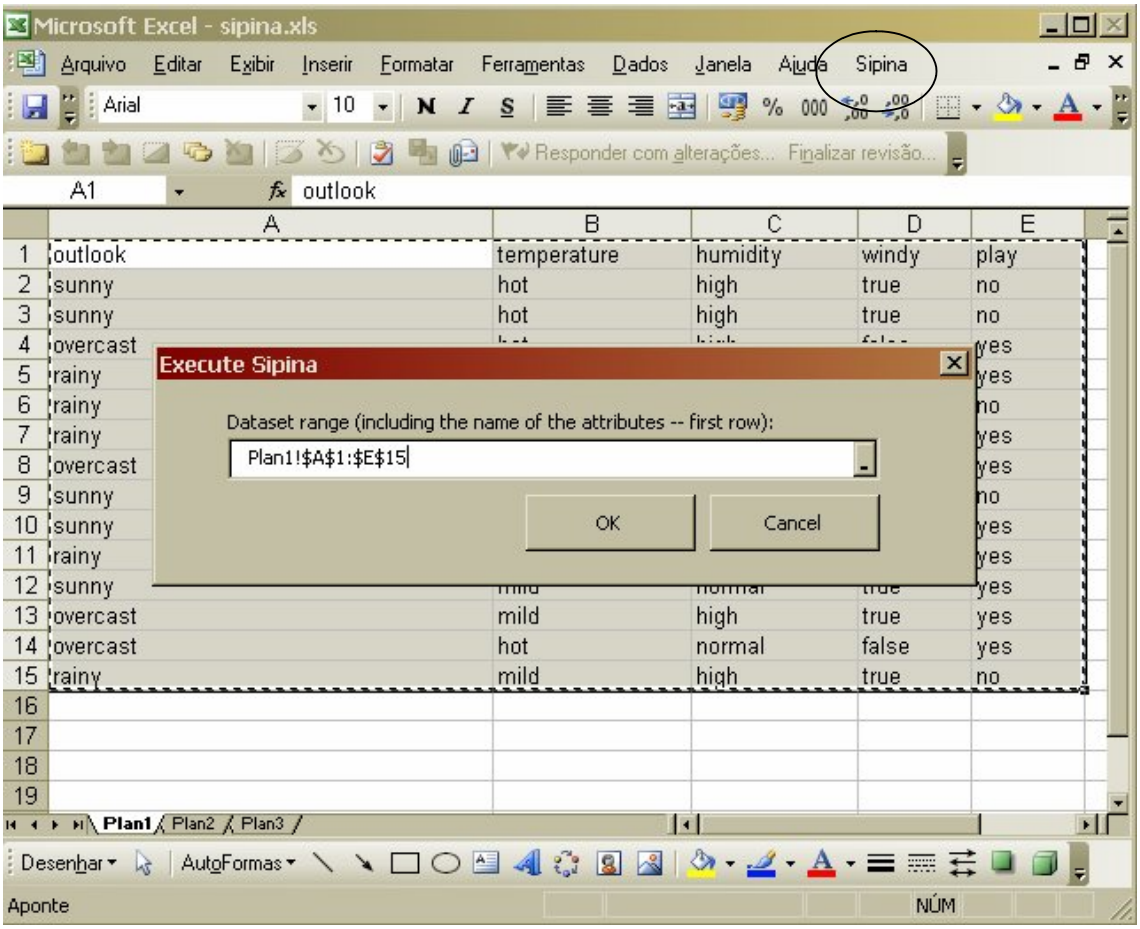


Figura 1.4: Gráfico com as ocorrências
Fonte: Do autor

1.4.2 Sipina

Ferramenta desenvolvida na Universidade de *Lyon*, é especializada na técnica de classificação, possuindo algoritmos próprios. Juntamente com a ferramenta, é disponibilizado um Suplemento para o Microsoft Excel[8]. Realizada a instalação, dados editados em planilhas eletrônicas nesta ferramenta podem servir de entrada de dados para o Sipina, bastando selecionar a opção *Execute Sipina*, no menu Sipina, informando o intervalo das células, conforme a Figura 1.5.



	A	B	C	D	E
1	outlook	temperature	humidity	windy	play
2	sunny	hot	high	true	no
3	sunny	hot	high	true	no
4	overcast				yes
5	rainy				yes
6	rainy				no
7	rainy				yes
8	overcast				yes
9	sunny				no
10	sunny				yes
11	rainy				yes
12	sunny				yes
13	overcast	mild	high	true	yes
14	overcast	hot	normal	false	yes
15	rainy	mild	high	true	no
16					
17					
18					
19					

Figura 1.5: Chamada da ferramenta Sipina a partir do Microsoft Excel
Fonte: Do autor

A ferramenta possui a implementação de diversos algoritmos, conforme na Figura 1.6.

[8] http://office.microsoft.com/pt-br/excel/FX100487621046.aspx

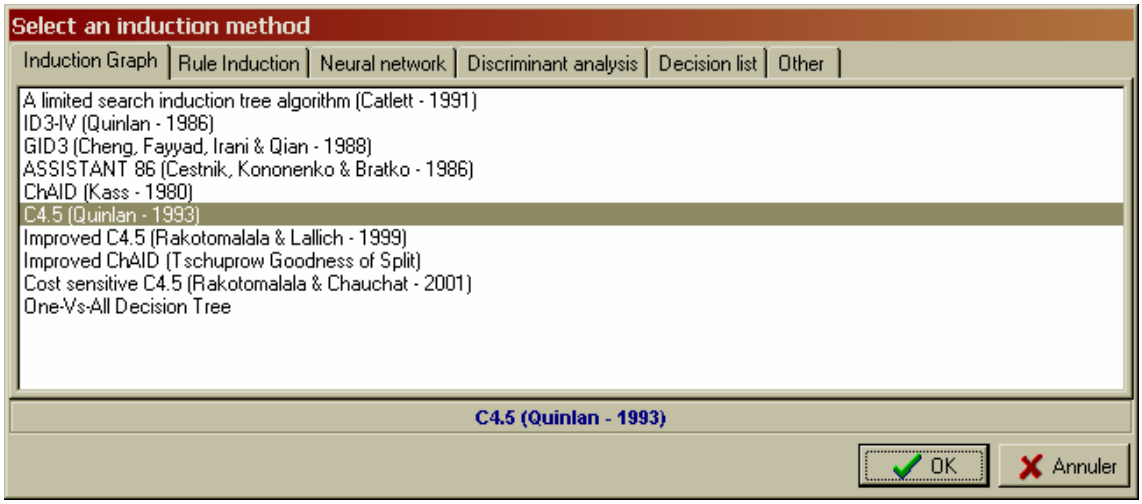


Figura 1.6: Opções de técnicas / algoritmos da ferramenta Sipina
Fonte: Do autor

Diferentemente da ferramenta Weka, para a geração de árvores de decisão é necessária a escolha do atributo que acabará como classe, como verificado na Figura 1.7:

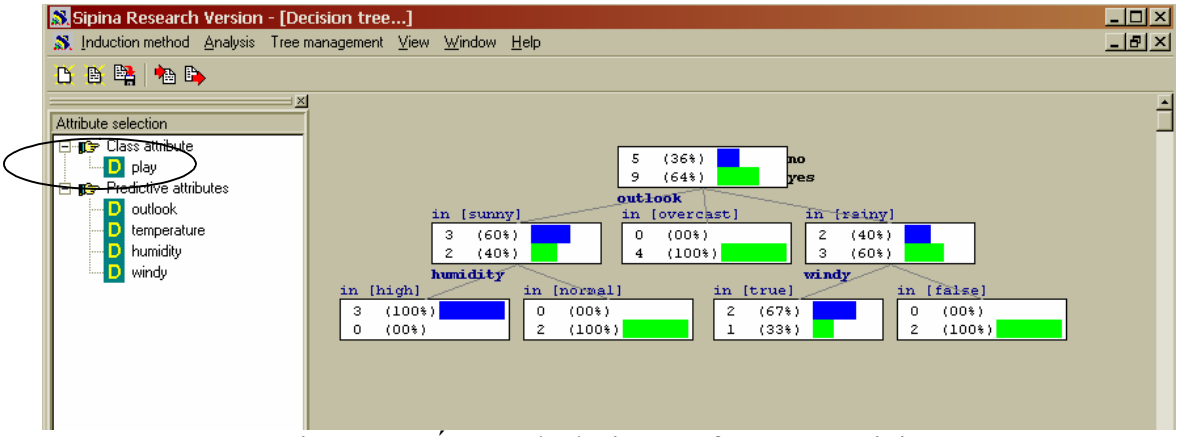


Figura 1.7: Árvore de decisão na ferramenta Sipina
Fonte: Do autor

1.4.3 RapidMiner

Ferramenta *Open-Source* criada na Alemanha, anteriormente chamada *Yale.* Conforme (RapidMiner, 2008), possui interface gráfica ao usuário (GUI) e *scripts* baseados em XML, tornando esta uma *Integrated Development Environment* (IDE) e um interpretador para KDD. É desenvolvida sob a plataforma Java, o que facilita integração com outras aplicações sob esta arquitetura.

Um exemplo desta integração, é o RapidMiner possuir incorporado toda a biblioteca *Weka.* Além disso, pode ser integrada com bibliotecas *Java Database Connectivity* (JDBC),

possibilitando a conexão diretamente ao banco de dados, para aplicação dos algoritmos de Mineração de Dados. Na Figura 1.8 é exibido uma aplicação do RapidMiner.

1.4.4 Pentaho

Pentaho BI é uma *Open Source Business Intelligence* (OSBI), com módulos de *Data Mining*, Análise, *Dashboards* (painel de indicadores) e Relatório.. O módulo de *Data Mining* é a biblioteca *Weka*, que foi incorporada ao ambiente da OSBI. As figuras 1.9 e 1.10 mostram *screensshots* do Pentaho BI.

A ferramenta possui versão para *download* gratuito e versão *enterprise* proprietária, tendo como casos de sucesso de sua utilização empresas de grande porte, como Sun[9] e Mozilla[10].

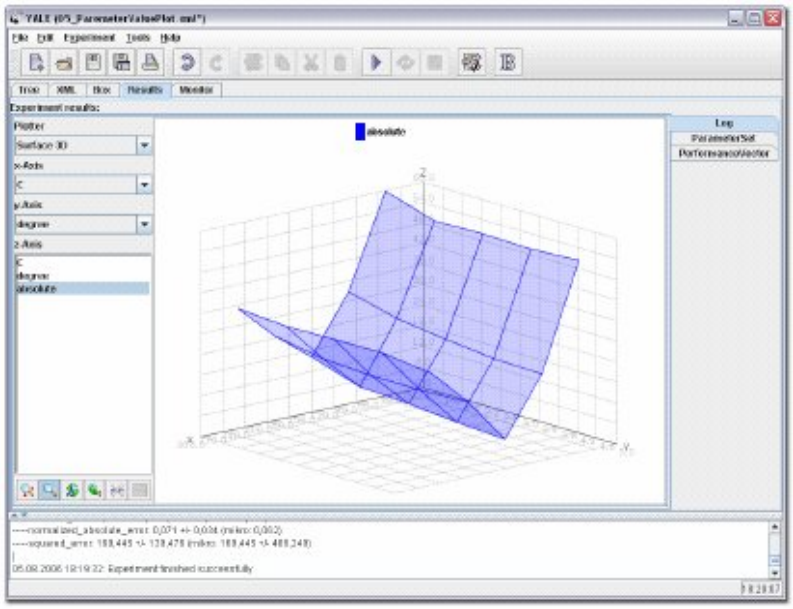


Figura 1.8: *Screenshot* da ferramenta RapidMiner
Fonte: RAPIDMINER, 2008

[9] http://java.sun.com
[10] http://www.mozilla.org

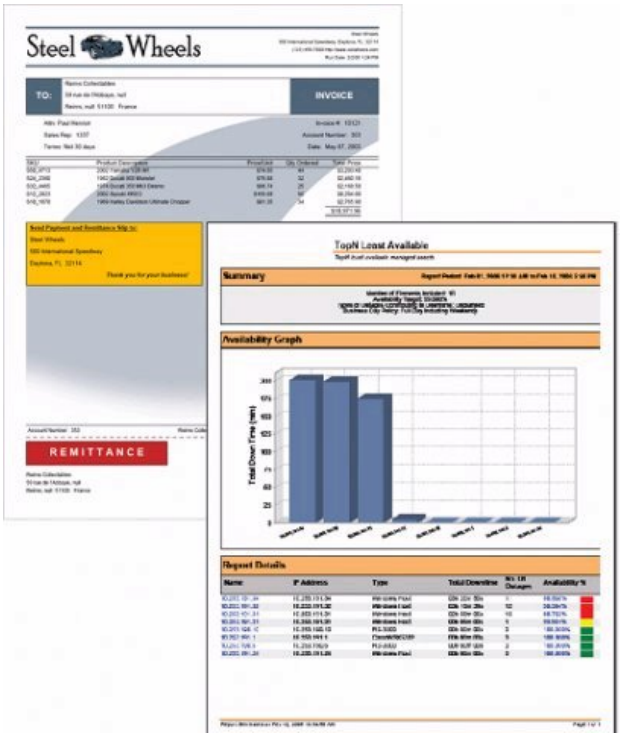


Figura 1.9: Módulo Relatório do Pentaho
Fonte: PENTAHO, 2008

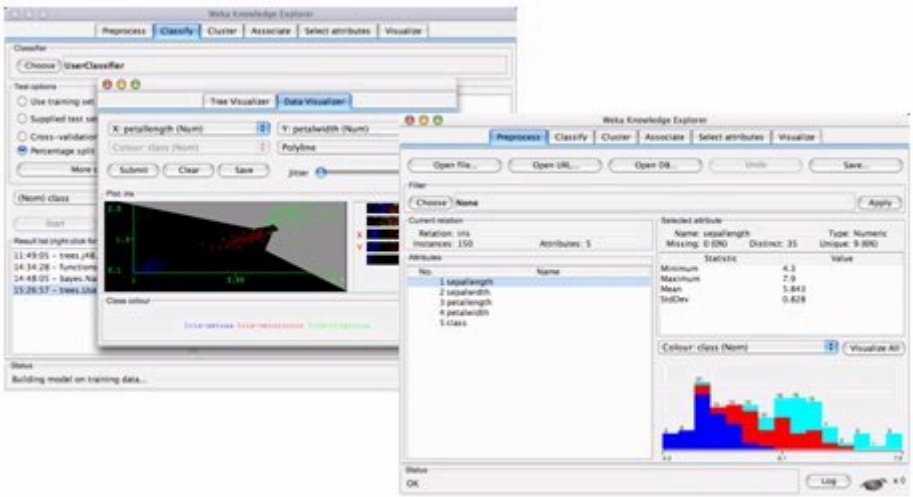


Figura 1.10: Módulo *Data Mining* do Pentaho
Fonte: PENTAHO, 2008

Neste capítulo foi abordada a área de Descoberta de Conhecimento, suas etapas e diferentes técnicas. Além disso, foram apresentadas algumas das ferramentas que implementam essas técnicas. No próximo capítulo, será detalhada a técnica de classificação e alguns algoritmos que a utilizam. Esta foi escolhida por trabalhar com classes pré-definidas, o que melhor se adéqua a aplicação de mapeamento de perfis a ser desenvolvida.

2 TÉCNICA DE CLASSIFICAÇÃO

Dentre os algoritmos de classificação, encontram-se diferentes métodos para a indução de conhecimento, destacando-se árvores de decisão, árvores de decisão com regras, RNA, regras a partir de IF <*condição*> THEN <*classe*>, como o algoritmo PRISM (VASCONCELOS, 2002) e método estatístico, como classificadores bayesianos (MCCALLUM, 1998).

2.1 Redes Neurais Artificiais

Área da Inteligência Artificial (IA) assim batizada por ser inspirada no funcionamento do cérebro humano, com as redes de células nervosas e o próprio comportamento. Segundo Carvalho (2000), as RNA são formadas por diversas redes em que existem unidades de processamento chamadas neurônios artificiais. O modelo de um neurônio artificial pode ser visto na Figura 2.1.

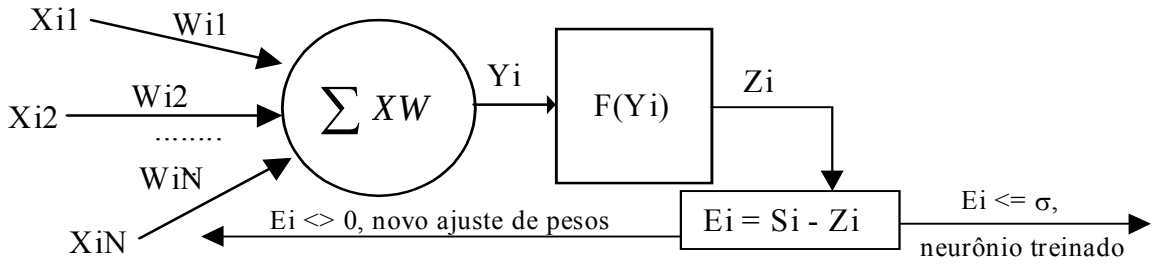


Figura 2.1 – Modelo de um neurônio artificial
Fonte: Carvalho (2000)

Conforme Goldschmidt (2005) as redes neurais artificiais possuem as seguintes características semelhantes ao cérebro humano:

- Busca paralela e endereçamento pelo conteúdo: o conhecimento fica distribuído pelas redes, não existindo endereços de memória.

- Aprendizado por experiência: busca a identificação de padrões através de repetidas apresentações dos dados às redes.

- Generalização: as RNAs conseguem generalizar a partir de exemplos anteriores, facilitando a manipulação de dados com impurezas.

- Associação: as RNAs possuem a capacidade de identificar relações entre padrões de natureza distinta.

- Abstração: é a possibilidade de abstrair a essência de um conjunto de dados de entrada.

- Robustez e Degradação Gradual: a perda de neurônios artificiais não significa prejuízo no desempenho, pois as informações estão distribuídas pela rede.

Já segundo Azevedo (1999), o aprendizado das RNAs pode ser:

Por independência de quem aprende: o aprendizado ocorre por memorização, contato, exemplos, analogia, exploração e descoberta;

Por retroação do mundo: o aprendizado pode ser supervisionado ou não-supervisionado, de acordo com a presença ou ausência de realimentação explícita, onde são assinalados erros ou acertos.

Por finalidade do aprendizado: o aprendizado pode ser por um auto-associador, em que a rede memoriza exemplos, conseguindo reproduzi-los caso sejam apresentados deteriorados posteriormente; hetero-associador, em que os exemplos são apresentados aos pares, e o segundo elemento pode ser reproduzido mesmo que o primeiro seja alterado; e detector de regularidades, em que são identificados padrões pela própria RNA, não sendo estes definidos anteriormente.

As RNA possuem, segundo Barreto (2002), diferentes topologias: diretas, com ciclos e simétricas.

2.1.1 Redes Diretas

É o tipo de rede mais utilizado, haja vista os métodos de aprendizado serem dos mais difundidos e fáceis de usar. Normalmente é utilizado em camadas, em que neurônios que

recebem estímulos são chamados de camada de entrada e os que têm sua saída sendo o final da rede camada de saída.

Na Figura 2.2, tem-se um exemplo de uma rede em camada. Note-se que as redes diretas não possuem ciclo.

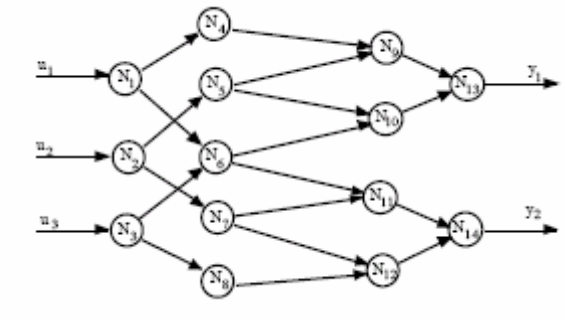


Figura 2.2: Exemplo de RNA direta
Fonte: Barreto(2000)

2.1.2 Redes Com Ciclo

São redes em que o grafo de conectividade possui pelo menos um ciclo. Quando há ocorrência de neurônios dinâmicos, são chamadas de redes recorrentes. Como os neurônios completam ciclos, podem realimentar outros neurônios.

Das redes com ciclo, destacam-se as redes propostas por Hopfield (1984) e as redes bi-direcionais (Kosko, 1988), que podem ser usadas por sistemas especialistas em um de seus principais paradigmas: treinamento com exemplos de uma rede direta e representação do conhecimento de modo localizado pelo uso de rede com ciclos. (AZEVEDO, 1999)

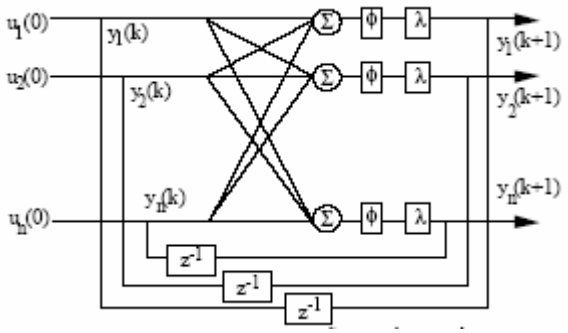


Figura 2.3: Exemplo de RNA com ciclo e neurônios dinâmicos
Fonte: Azevedo (1999)

Na Figura 2.3 é exibido um exemplo de redes com ciclo. Existe ainda um tipo específico de redes com ciclo, as redes simétricas, em que a matriz de conectividade é simétrica. (BARRETO, 2002).

2.2 Árvores de decisão

São uma forma simples de representação das regras, composta de nodos, ligações e folhas, significando, respectivamente, os atributos, seus possíveis valores e as diferentes classes. Na Figura 2.1, é exibida uma árvore de decisão, gerada a partir da aplicação do algoritmo J48 sobre os dados da Figura 1.2.

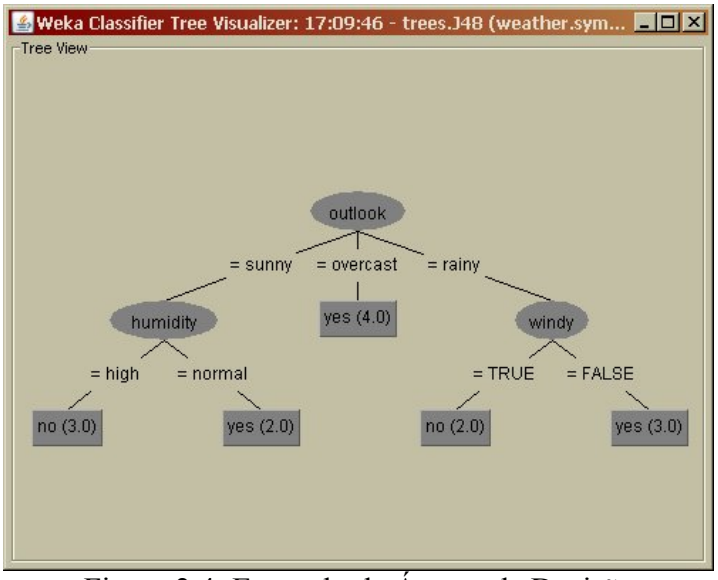


Figura 2.4: Exemplo de Árvore de Decisão
Fonte: SANTOS, 2005

As árvores de decisão são de fácil entendimento, sendo possível a interpretação inclusive pelos usuários, mesmo quando da representação de grandes bases de dados. No entanto, os algoritmos família TDIDT (*Top-Down Induction of Decision Trees*) não são totalmente eficazes, pois existem regras que não são possíveis de se representar. Esse problema é decorrente, muitas vezes, do fato de o nodo raiz obrigatoriamente constar nas regras, sendo chamado de problema sintático, conforme (MONGIOVI, 1998).

Conforme (GOLDSCHMIDT, 2005), os algoritmos dividem o conjunto de treinamento em duas ou mais partes, sendo um processo recursivo repetido até que todos os itens do conjunto pertençam a uma classe. O algoritmo baseado em árvore de decisão mais utilizado é o C4.5 (Quinlan, 1993), que tem sua origem no ID3 (QUINLAN, 1979).

2.2.1 ID3

O algoritmo ID3, cujo nome significa *Iterative Dichotomizer Tree*, inicialmente seleciona um atributo para o nodo raiz, gerando ligações para todos os diferentes valores; se todos os sob sobre um nodo pertencem a uma mesma classe, o nodo passa a ser uma folha que recebe o nome da classe; enquanto existem nodos sem classe, o nodo recebe um atributo ainda não utilizado pela árvore com ligações criadas para todos os valores. A escolha dos atributos a serem utilizados pela árvore se dá a partir de informações de entropia e ganho de informação.

O valor da entropia corresponde à impureza do atributo, a falta de homogeneidade, sendo calculada para cada atributo. O ganho de informação é a variação da impureza. O que possuir menor valor de entropia ou maior ganho de informação é escolhido o nó raiz da árvore (ASCENSO, 2004). O cálculo de entropia é dado na Figura 2.2, em que cada "p" corresponde ao número de instâncias sobre o de exemplos.

$$\text{Entropia}(S) = -(p1 * \log_2 p1 + p2 * \log_2 p2 + \ldots + pn * \log_2 pn)$$

Figura 2.2: Fórmula de cálculo de entropia
Fonte: OSÓRIO, 2001

Para ilustrar o algoritmo, serão tabulados os dados do arquivo da Figura 1.2 na Tabela 2.1.

Tabela 2.1: Conjunto para exemplo do ID3

	tempo	temperatura	umidade	vento	jogar
1	ensolarado	quente	alta	não	não
2	ensolarado	quente	alta	sim	não
3	nublado	quente	alta	não	sim
4	chuvoso	amena	alta	não	sim
5	chuvoso	fria	normal	não	sim
6	chuvoso	fria	normal	sim	não
7	nublado	fria	normal	sim	sim
8	ensolarado	amena	alta	não	não
9	ensolarado	fria	normal	não	sim
10	chuvoso	amena	normal	não	sim
11	ensolarado	amena	normal	sim	sim
12	nublado	amena	alta	sim	sim
13	nublado	quente	normal	não	sim
14	chuvoso	amena	alta	sim	não

Fonte: Do Autor

A seguir são realizados os cálculos de entropia para os atributos do conjunto: tempo, temperatura, umidade e vento (jogar é atributo de classe, não sendo calculada sua entropia). Inicialmente são calculadas as entropias dos possíveis valores do atributo.

No cálculo abaixo, “2/9” corresponde ao número de ocorrências de *ensolarado* (duas ocorrências) nas transações em que *jogar = sim* (nove transações). De forma análoga, há três ocorrências de *ensolarado* nas transações em que *jogar = não* (cinco transações).

Entropia (tempo=ensolarado) = - (2/9) * log2(2/9) – (3/5) * log2(3/5) = 0,924

Quando todas as ocorrências de um valor de atributo correspondem a uma mesma classe, a entropia é 0 (em todas as ocorrências de *nublado*, *jogar = sim*).

Entropia (tempo=nublado) = 0

Entropia (tempo=chuvoso) = - (3/9) * log2(3/9) – (2/5) * log2(2/5) = 1,057

Deve-se multiplicar a entropia de cada valor pela divisão entre sua ocorrência e o total de transações. Este cálculo deve ser realizado para todos os possíveis valores do atributo, sendo a soma destes resultados a entropia do atributo.

No exemplo abaixo, 14 é o número de transações; 5, 4 e 5 são as respectivas ocorrências de *ensolarado*, *nublado* e *chuvoso*, sendo 0,924, 0 e 1,057 suas entropias.

Entropia (tempo) = (5/14) 0,924 + (4/14)0 + (5/14) 1,057= 0,70

Entropia (vento=sim) = - (3/9) * log2(3/9) – (3/5) * log2(3/5) = 0,97

Entropia (vento=não) = - (6/9) * log2(6/9) - (2/5) *log2(2/5) = 0,918

Entropia (vento) = (6/14)0,97 + (8/14)0,918 = 0,94

Entropia (temperatura=fria) = - (3/9) * log2(3/9) – (1/5)* log2(1/5) = 0,992

Entropia (temperatura =amena) = - (4/9) * log2(4/9) – (2/5) * log2(2/5) = 1,048

Entropia (temperatura =quente) = - (2/9) * log2(2/9) – (2/5) * log2(2/5) = 1,01

Entropia (temperatura) = (4/14)0,992 + (6/14)1,048 + (4/14)1,01= 1,021

Entropia (umidade=normal) = - (6/9) * log2(6/9) – (1/5) * log2(1/5)= 0,854

Entropia (umidade=alta) = - (3/9) * log2(3/9) - (4/5) * log2(4/5) = 0,785

Entropia (umidade) = (7/14)0,854 + 7/14(0,785)= 0,822

Percebe-se que o atributo de menor entropia é *tempo*, sendo este o nó raiz da árvore a ser gerada. O algoritmo ID3 não necessariamente produz as melhores regras, sendo criadas em alguns casos, árvores muito grandes. Surgiu então, uma evolução do ID3, o C4.5, pelo mesmo criador, Quinlan, em 1993.

2.2.2 C4.5

O algoritmo C4.5 consegue, conforme Coello (2002), trabalhar com registros com valores desconhecidos para alguns atributos, tratar atributos com valores contínuos, e inferir regras a partir da árvore. Para evitar o problema de tamanho da árvore, o C4.5 trabalha com a poda.

A poda é baseada no grau de incerteza de um atributo, quando, por exemplo, a entropia é maior que um determinado valor. A operação consiste em substituir uma sub-árvore, com grande taxa de erro, por um nodo ou por um ramo e é realizada antes ou depois de a árvore ser elaborada. Assim a árvore é simplificada, o que facilita a inserção de novos nodos e sub-árvores, necessidade que vai surgindo conforme o conjunto de dados cresce. No entanto, existem casos em que a poda é difícil de ser aplicada, como em árvores que não se encontram equilibradas.

Para ilustrar um exemplo de poda, utilizou-se o conjunto de dados da tabela 2.1, inserindo-se mais duas transações, Figura 2.2, com o intuito de tornar a árvore mais complexa.

Tabela 2.2: Novas transações no conjunto

	tempo	temperatura	umidade	vento	Jogar
15	nublado	amena	alta	não	não
16	nublado	fria	alta	sim	sim

Fonte: Do Autor

Utilizando-se da ferramenta Weka, aplicou-se os algoritmos ID3 e J48 (WITTEN; FRANK, 2000) para que esses gerassem suas respectivas árvores. O J48 é a implementação em Java do algoritmo C4.5. Na Figura 2.5 e 2.6 são exibidas as árvores geradas.

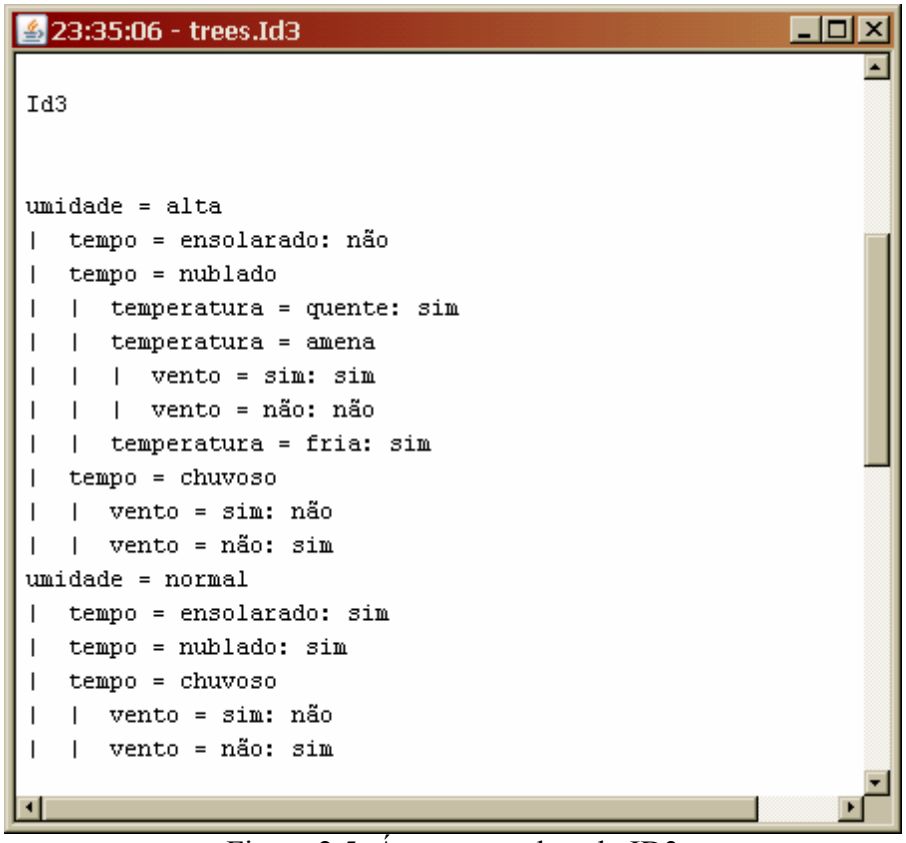


Figura 2.5: Árvore gerada pelo ID3
Fonte: Do autor

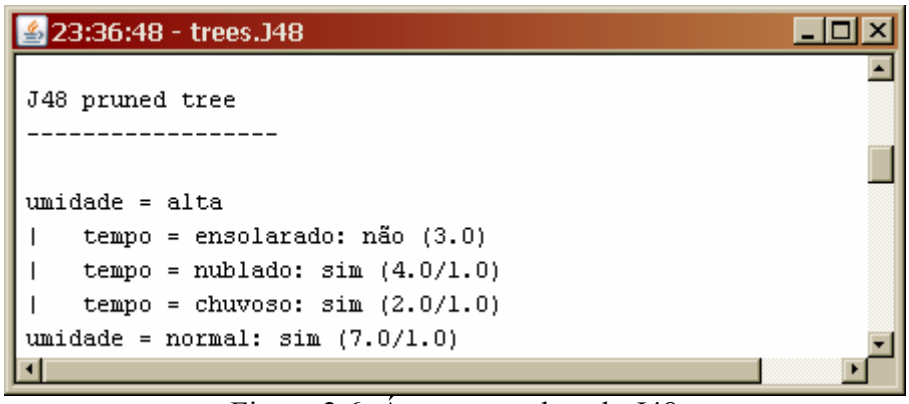


Figura 2.6: Árvore gerada pelo J48
Fonte: Do autor

Ainda com o objetivo de reduzir as árvores, segundo Quinlan (1993), o algoritmo é capaz de agrupar valores de atributos. Conseqüentemente, em uma situação em que um atributo possua *n* valores pertencentes a uma mesma classe, será criado um ramo para o grupo de valores ao invés de *n* ramos. Nos casos em que ocorrem valores desconhecidos para um determinado atributo, o C4.5 trabalha com estatística baseada no conjunto dos dados até então conhecido.

O C4.5 permite ainda gerar regras a partir da árvore de decisão, em que condições irrelevantes são excluídas. Isto permite que as regras se tornem mais simples do que se fossem simplesmente elaboradas a partir de toda a extensão da árvore. Estas regras podem ser no formato convencional (a), com condições combinadas (b) e ainda de dissociação (c), conforme Tabela 2.3:

Tabela 2.3: Formato das regras

	Formato
a	IF <condicao> THEN <classe>
b	IF <condicao1> (and | or)* <condicao2> THEN <classe>
c	IF (NOT)+ <*condicao1*> (and | or)* (NOT)+ <*condicao2*> THEN <*classe*>

Fonte: Do autor

2.3 C&RT

Classification & Regression Trees, também chamado de CART, foi proposto por Breiman (1984). É um algoritmo robusto que possui excelente performance, inclusive quando trabalha com imensa quantidade de dados, sendo um dos mais utilizados algoritmos para construção de árvores de decisão (LEWIS, 2000).

O algoritmo testa todas as possibilidades de divisão para as variáveis, posteriormente realizando a análise para particionar o nodo. Essa partição é feita através de perguntas com respostas binárias (Sim, Não), sendo possível a manipulação de variáveis com valores contínuos ou categorizados.

2.4 CHAID

Outro algoritmo que possui a característica de construir árvores a partir de regressão. É um acrônimo para *CHi-squared Automatic Interaction Detector*. Diferentemente do CART, a partição dos nodos não é binária, acarretando em árvores mais esparsas, o que pode ser uma vantagem para problemas como o de produtos em um supermercado, por exemplo.

O CHAID, segundo Kass (1980), apresenta melhor desempenho para grandes conjuntos de dados. No entanto, apesar deste desempenho, a árvore gerada pelo CHAID é maior que a obtida pelo C4.5. O algoritmo foi executado a partir da ferramenta Sipina, utilizando-se os mesmos dados (Tabelas 2.1 e 2.2).

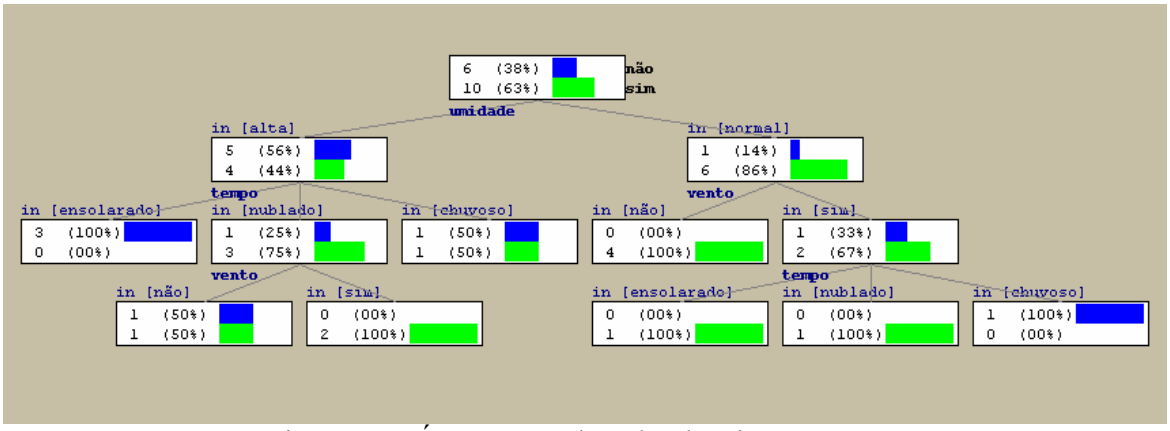


Figura 2.7: Árvore gerada pelo algoritmo CHAID
Fonte: do Autor

O algoritmo recebe como parâmetros *sppliting nodes*, relativo ao número de divisões dos nodos, quanto mais alto, menor a performance e maior a árvore a ser gerada e *merging nodes* em que os nodos são unidos se possuem diferença pouco significativa, sendo este critério relativo ao parâmetro informado.

2.5 Redes Bayesianas

Assim como RNA, é uma área de IA, muito utilizada em problemas para diagnóstico e decisão. As redes bayesianas utilizam o Teorema de *Bayes*, decorrente do Teorema da probabilidade, juntamente com grafos, que representam as relações entre os conjuntos das probabilidades (MELLO, 2007).

Os grafos utilizados nas redes bayesianas são direcionados e acíclicos. Uma ligação de um nodo a outro representa a influência direta entre um e outro. Cada um destes nodos armazena os possíveis estados da variável juntamente com uma tabela de probabilidades que quantificam a influência de outro nodo ligado.

Para a criação desta rede, são passados como entrada para as redes bayesianas dados e informações para que após o processamento, a saída seja a rede propriamente dita. Segundo Guinzani (2006), a aprendizagem consiste inicialmente em gerar a rede que apresenta as relações entre as variáveis, sendo em seguida determinada a representação da distribuição das probabilidades de cada nodo, para assim estimar as probabilidades condicionais com a base de dados, considerando apenas os valores de variáveis relevantes.

As redes bayesianas são amplamente utilizadas na área de diagnósticos médicos. A Figura 2.7 mostra um exemplo de uma rede bayesiana para diagnóstico de doenças cardíacas (SAHESKI, 2005).

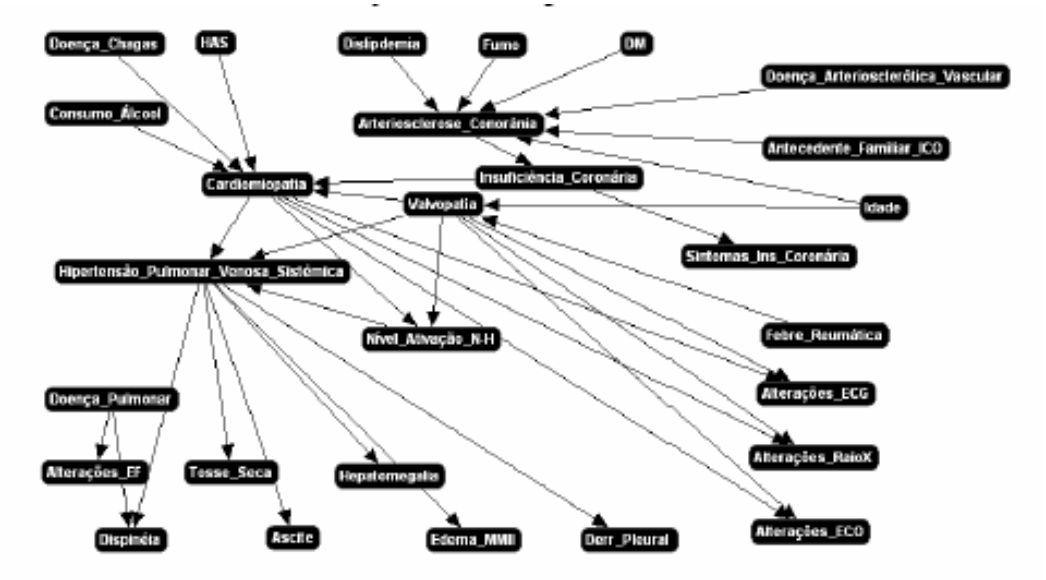


Figura 2.8: Exemplo de Rede Bayesiana
Fonte: SAHESKI (2005)

2.6 Prism

O Prism é um algoritmo de classificação que não pertence à família TDIDT, não possuindo o problema sintático (vide seção 2.2), tendo a garantia de que as regras geradas estão corretas. Esse algoritmo infere regras no formato de IF <*condição*> THEN <*classe*>, em que, conforme (VASCONCELOS, 2002), uma *condição* é um conjunto de termos <atributo_c_t q valor>, onde *atributo_*c_t corresponde a um atributo do conjunto de treinamento; q = {.=., .<., .>., .£., .³., .¹.} e *valor* é um possível valor do atributo. Se a condição for verdadeira, o dado é agrupado à *classe*.

Em comparação com as árvores de decisão, são geradas um número menor de regras, sendo estas ainda menos complexas. No entanto, utilizando o Prism podem ocorrer sobreposição de regras.

2.7 Meta-classificação

Os algoritmos de classificação normalmente trabalham com todos os dados do conjunto de treinamento em memória. No entanto, isto pode ser um grande problema quando se trata de grandes bases de dados. Para o devido tratamento deste problema destacam-se algoritmos incrementais, técnicas de amostragem, processamento paralelo e técnicas de partição e combinação de resultados parciais. Em se adotando esta medida, os dados são divididos em subconjuntos e a cada um destes são aplicados os algoritmos tradicionais e os resultados então são combinados (MONGIOVI, 1998). Os classificadores que aplicam algoritmos aos resultados de outros classificadores são chamados de *meta-classificadores.*

Uma das ferramentas utilizadas neste projeto, Weka, possui meta-classificadores, destacando-se *Bagging* e *Boosting*. De acordo com Oliveira (2006), o método *Bagging* constrói os classificadores a partir de conjuntos independentes de amostras de dados, sendo estes gerados a partir do conjunto de treinamento. São extraídas instâncias conforme estas são repetidas nas amostras.

Ainda segundo Oliveira (2006), no método *Boosting* as instâncias possuem um peso associado. Inicialmente, o peso é o mesmo para todas as instâncias, sendo este alterado conforme as instâncias de treinamento são classificadas incorretamente, baseadas nos classificadores gerados anteriormente.

Neste capítulo foram apresentadas metodologias de utilização da técnica de classificação, como RNAs e árvores de decisão e algoritmos que implementam estas. No Estudo de Caso, será abordado como KDD vem sendo utilizado na área de Marketing e as etapas que se sucederão após o recebimento dos dados da instituição para posterior aplicação dos algoritmos.

3 ESTUDO DE CASO

A crescente procura do mercado de trabalho por empregados cada vez mais qualificados, juntamente com a perspectiva de crescimento profissional, acarretou em uma maior procura por cursos de graduação. Conseqüentemente, acabaram por surgir diversas universidades e centro universitários, tornando-se um setor com acirrada concorrência.

Assim como em qualquer segmento, no de ensino superior também é imprescindível um diferencial em relação aos concorrentes para a atração dos clientes, no caso, os alunos. Buscando esta qualidade, serão aplicadas as técnicas de KDD para que assim identifiquem-se perfis de alunos e relações entre esses perfis e o interesse por cada curso ou área.

3.1 Aplicações na área de Marketing

A identificação de perfis de clientes vem sendo utilizada cada vez mais por aplicações de *Data Mining* com enfoque na área de Marketing, objetivando de estreitar as relações com os consumidores, visando melhor atendimento, fidelização e até a conquista de novos clientes.

Essa área que associa conceitos de Marketing de Relacionamento ao uso de TI é a *Customer Relationship Manager* (CRM), que torna possível às empresas que adotam sistemas da área um serviço personalizado, antecipando as necessidades dos clientes e suprindo-as totalmente. Uma área derivada do CRM é a *Customer Interaction Management* (CIM), em que a empresa norte-americana Talismã possui ferramentas específicas como a *Talisma Knowledgebase Software*, que permite direcionar o contato da empresa com o cliente, a partir de transações já realizadas, evitando gastos com comunicação.

O mapeamento de perfis também é utilizado para análise de crédito, em que características dos clientes são associadas ao histórico de pagamento. As instituições

financeiras utilizam ferramentas específicas para este fim, como o Banco do Brasil, conforme Lemos (2000) possui um *software* desenvolvido internamente chamado Análise de Crédito (ANC).

Um caso de sucesso é o Grupo RBS, que possui a RBS Direct[11], uma empresa específica para o Marketing de precisão, utilizando ferramenta de CRM da Oracle[12] e conta com uma equipe de desenvolvimento, que entre outras atividades, utiliza *Data Mining* para criar perfis de clientes.

Podem ser mapeadas características de consumidores através do acesso a internet, sendo rastreadas as páginas de maior incidência de acessos para os usuários cadastrados de um *site*, por exemplo. Isto acaba caracterizando uma sub-área de KDD, a *Web Mining* (BOULLOSA, 2002).

3.2 Dados dos vestibulandos

Com o objetivo de encontrar relações entre a demanda dos cursos da instituição e os perfis de vestibulandos a serem traçados, foram selecionados os dados de vestibulares de 2006 a 2008. Nessas bases de dados, encontram-se informações de endereço, nascimento, renda familiar, escola em que o aluno cursou o Ensino Médio entre outras. O formato dos dados recebidos foi de planilha eletrônica.

Serão elaborados procedimentos para alimentar um banco de dados que serão geridos pelo Sistema Gerenciador de Banco de Dados (SGBD) SQL Server Express[13]. A criação deste BD relacional se dará única e exclusivamente pelo fato de facilitar a manipulação dos dados, haja vista o grande número de atributos disponíveis, sendo que estes não serão utilizados, na maior parte dos casos, simultaneamente, por exemplo, ora se manipularão *idade* e *sexo*, ora *município* e tipo de escola freqüentada no Ensino Médio (pública ou particular).

Os alunos poderão estar distribuídos em vários perfis simultaneamente, pois estes podem possuir informações não mutuamente exclusivas, como por exemplo, informações de sexo e idade em um e município de residência e escola de ensino médio em outro. Como esses perfis serão pré-definidos, será utilizada a técnica de classificação.

---

11 http://www.rbsdirect.com.br

12 http://www.oracle.com

13 http://www.microsoft.com/sqlserver/2005/en/us/express.aspx

Algumas rotinas serão implementadas para, a partir de consultas SQL, gerar os arquivos no formato .arff com os atributos a serem utilizados para determinadas classificações. Realizadas as distribuições dos registros nas classes, serão aplicados os algoritmos para identificar as principais relações entre os perfis mapeados e as escolhas pelos cursos. Estes resultados serão apresentando para o departamento de Marketing da instituição, para que este, a seu critério, adote medidas a fim de atrair mais alunos à instituição, que podem ser desde o direcionamento de propagandas até o agendamento de visitas de apresentação conforme a escola ou região de maior procura pelo curso (ou menor).

CONCLUSÃO

A Mineração de Dados, quando bem aplicada, é capaz de propiciar muitos benefícios às corporações, ajudando na tomada de decisões ou na descoberta de conhecimento que pode ser utilizado em posicionamento estratégico. Em determinados segmentos, como no de instituições de ensino superior, em que a concorrência está cada vez mais acirrada, as técnicas de *Data Mining* estão se tornando essenciais.

A área de Marketing se tornou o foco de diversas aplicações de Descoberta de Conhecimento, tanto acadêmica como comercialmente. No entanto, ainda faz-se necessário um estudo das técnicas (e suas possíveis combinações) que facilitem o mapeamento de perfis, um dos principais objetivos de aplicações na área.

Espera-se, com este projeto, que o conhecimento a ser descoberto seja utilizado pelo departamento de Marketing da instituição com o objetivo de atrair os alunos. No entanto, as medidas a serem tomadas e o seu foco (se perfis com alta ou baixa demanda por cursos), ficam a total critério deste setor.

Atingindo as metas, proporcionando bons resultados para a instituição, almeja-se adquirir bases de dados não somente de vestibulandos, mas de alunos desde o início do curso até o final da graduação. Com isto, poderão ser tomadas medidas a fim de evitar que matrículas sejam trancadas ou a própria evasão.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Fernando C. et al. **Data Mining no contexto de Customer Relationship Management.** Caderno de Pesquisas em Administração, São Paulo, v. 12, n. 2, p. 85-97, abril/junho 2005.

AGRAWAL, Rakesh; SRIKANT, Ramakrishnan. **Fast Algorithms for Mining Association Rules, In: 20th International Conference on Very Large Data Bases, 1994**. **Procedings.** San Francisco, 1994. p. 487-499

ASCENSO, João. **Reconhecimento de Padrões.** Escola Superior de Tecnologia – Engenharia Informática. Setúbal, 2004

AZEVEDO, Fernando Mendes de. **Algoritmos Genéticos em Redes Neurais Artificiais.** In: V Escola de Redes Neurais, São José dos Campos, 1999

BARRETO, Jorge M.. **Introdução às Redes Neurais.** Laboratório de Conexionismo e Ciências Cognitivas – UFSC, Florianópolis, 2002

BOULLOSA, José Roberto de Freitas. **Um Ambiente para Mineração de Utilização da Web.** Tese - Universidade Federal do Rio de Janeiro, 2002

BREIMAN, L; FRIEDMAN, J. H.; OLSHEN, R. A. **Classication and regression trees.** Belmont: Wadsworth Statistical Press, 1984

BROWN, Myra; SAHIN, Bilge. **Using Data Mining for Competitive Advantage in Direct Marketing Machine Learning Algorithms - Decision Trees.** Itom 6032, Spring 2002

CARVALHO, Deborah Ribeiro; BUENO, Marcos; NETO, Wilson Alves. **Ferramenta de Pré e Pós-processamento para Data Mining.** In: XII Seminário de Computação. Blumenau, 2003.

CARVALHO, Juliano V. **Reconhecimento de Caracteres Manuscritos Utilizando Regras de Associação.** Campina Grande: 2000. Dissertação (Mestrado), UNCG, 2000.

COELLO, Adán; MANUEL, Juan. **Aprendizado de heurísticas para o escalonamento de sistemas de tempo real.** In: Anais do IV Workshop Brasileiro sobre Sistemas de Tempo-Real, evento integrante do 20o. Simpósio Brasileiro de Redes de Computadores- SBRC'2002. Sociedade Brasileira de Computação (SBC), pp. 3-10, Búzios, RJ, 20 a 24 de maio de 2002.

GOLDSCHMIDT, Ronaldo; PASSOS, Emmanuel. ***Data Mining* :** Um guia prático. Rio de Janeiro: Elsevier, 2005.

FREITAS, Alex A. **Data Mining**, In: XIII Simpósio Brasileiro de Banco de Dados. Maringá/Brasil, 1998.

FREITAS, Alex A. **Understanding the Crucial Differences Between Classification and Discovery of Association Rules** – A Position Paper. SIGKDD Explorations, New York, 2000.

GONÇALVES, Eduardo Corrêa. Data Mining de Regras de Associação. Disponível em < http://www.devmedia.com.br/articles/viewcomp.asp?comp=7065> . Acessado em 27/09/2008

GRÉGIO, André Ricardo Abed. **Aplicação das Técnicas de *Data Mining* para a Análise de *Logs* de Tráfego TCP/IP.** São José dos Campos: 2007. Dissertação (Mestrado), INPE, 2007

HOPFIELD, J. **Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons.** In: Proceedings of the National Academy of Sciences, vol.81, 1984, pp.3088-3092

KASS, G. V. **An Exploratory Technique for Investigating Large Quantities of Categorical Data.** Journal of Applied Statistics, Vol. 29, No. 2 (1980), pp. 119-127.

KOSKO, B. **Bidirectional associative memories.** IEEE Transactions on Systems, Man, and Cybernetics, vol.18, no.1, pp.49-60, 1988.

LEWIS, Roger J. **An Introduction to Classification and Regression Tree (CART) Analysis.** In: Annual Meeting of the Society for Academic Emergency Medicine in San Francisco, 2000.

LEMOS, Eliane Prezepiorski; STEINER, Maria Teresinha Ams; NIEVOLA, Júlio César. **Análise de Crédito Bancário por Meio de Redes Neurais e Árvores de Decisão:** uma aplicação simples de Data Mining. Revista de Administração USP. Ed jul/ago/set 2005.

LOH, Stanley**; Data Mining.** Disponível em <http://atlas.ucpel.tche.br/~loh/dm-ppt.pdf>. Acessado em 02/10/2008.

MCCALLUM, Andrew; NIGAM, Kamal**. A Comparison of Event Models for Naive Bayes Text Classication.** In: AAAI-98 Workshop on Learning for Text Categorization. Stanford, 1998.

MELLO, Márcio Pupin de et al. **Redes Bayesianas no Delineamento de Culturas Agrícolas Usando Informações Contextuais.** In: XXIII Congresso Brasileiro de Cartografia, Rio de Janeiro, 2007

MONGIOVI, Giuseppe. **T.E.I. Data Mining.** Notas de aula. Campina Grande. 1998.

OLIVEIRA, Fernando Luiz et al. **Utilização de Algoritmos Simbólicos para a Identificação do Número de Caroços do Fruto Pequi.** In: II Encontro de Informática do Tocantins, Palmas, 2002

OSÓRIO, Fernando. **Machine Learning: Aprendizado simbólico a partir de exemplos.** Disciplina de Redes Neurais 2001/2, Unisinos, São Leopoldo

QUINLAN, J.R. ***Discovering rules by induction from large collection of examples.*** Expert Sysmtes in the Micro Electronic Age. Edinburgh, UK: Edinburgh University Press, 1979.

QUINLAN, J.R. **C4.5: *Programs for Machine Learning.*** São Francisco: Morgan Kaufmann, 1993.

PENTAHO, Sourceforge. Disponível em <http://sourceforge.net/projects/pentaho/> acessado em 26/10/2008

RAMOS, Túlio Terra. Estudo em mineração de dados aplicado nos Parâmetros de Controle do Processo de Produção de Agentes Tanantes. Trabalho de Conclusão, ULBRA, Canoas, 2004.

RAPIDMINER, Dortmund – Alemanha. **RapidMiner 4.1 User Guide.** Dortmund, 2008.

ROMANI, Daniel David; KNUPPE, Gustavo; SARAIVA, Marco Antônio Barbosa. ***Data Mining.*** Disponível em <http://paginas.terra.com.br/informatica/arruda/Downloads/Artigos/artigo07/index.htm>. Acessado em 24/09/2008

SAHEKI, André Hideaki. **Construção de uma rede Bayesiana aplicada ao diagnóstico de doenças cardíacas.** (Mestrado) - ESC POLITECNICA, Universidade de São Paulo, São Paulo, 2005.

SANTOS, Rafael. **Weka na Munheca:** Um guia para uso do Weka em scripts e integração com aplicações em Java. Apostila Princípios e Aplicações de Mineração de Dados. S.l., 2005.

VASCONCELOS, Benitz; SAMPAIO, Marcus Costa. **Mineração Eficiente de Regras de Classificação com Sistemas de Banco de Dados Objeto-Relacional.** In: XVII Simpósio Brasileiro de Banco de Dados, Gramado, 2002.

WITTEN, Ian H.; FRANK, Eibe. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. San Diego: Morgan Kaufmann Publishers, 2000