CENTRO UNIVERSITÁRIO FEEVALE

CARLOS EDUARDO DOS SANTOS

DATA MINING APLICADO A CAMPANHAS DE MARKETING

Novo Hamburgo, dezembro de 2008.

CARLOS EDUARDO DOS SANTOS

DATA MINING APLICADO A CAMPANHAS DE MARKETING

Centro Universitário Feevale
Instituto de Ciências Exatas e Tecnológicas
Curso de Sistemas de Informação
Trabalho de Conclusão de Curso

Professor orientador: Edvar Bergmann Araujo

Novo Hamburgo, dezembro de 2008.

AGRADECIMENTOS

Gostaria de agradecer a todos os que, de alguma maneira, contribuíram para a realização desse trabalho de conclusão, em especial: á Deus que está me proporcionando esta oportunidade, aos meus pais que me proporcionaram ensinamentos preciosos e me incentivaram a estudar, a minha esposa, aos amigos e ao meu orientador que me auxiliou no que foi preciso para execução desse trabalho.

RESUMO

O mercado vive um momento em que as empresas possuem grandes volumes de dados armazenados em seus discos rígidos. No entanto, poucas estão valendo-se disso para ampliar sua capacidade competitiva através da aplicação de técnicas computacionais para obter conhecimento. A tecnologia de *data mining* utiliza como matéria prima a informação e, a partir desta, produz conhecimento. Desta forma, auxilia os responsáveis pela tomada de decisões a efetuarem isso de forma mais assertiva, com o objetivo de obter o retorno do investimento de forma satisfatória, trazendo assim um diferencial competitivo. Sendo assim, esse trabalho tem o objetivo de estudar e aplicar as técnicas e algoritmos de *data mining*. Serão analisados os resultados a fim de indicar o algoritmo que demonstrar resultados satisfatórios, quando aplicados em uma base de clientes do setor de varejo calçadista, visando à elaboração de campanhas de *marketing* mais eficazes.

Palavras-chave: *Data Mining*. Campanhas de *Marketing*. Descoberta de Conhecimento. Varejo Calçadista.

ABSTRACT

The market experiences a time where companies have a large volume of data stored on their hard drives. However, few companies are taking advantage of that scenario by applying computational techniques in order to become more competitive and gain knowledge. Data Mining technology uses as its main raw material information and, from that point, produces knowledge. This way, it helps the responsible personnel to take the decision with accuracy, focusing on a satisfactory return of investment, thereby bringing a competitive differential. Therefore, this work aims to study and apply the techniques and algorithms for data mining. The results are analyzed and the algorithms that demonstrate satisfactory results are pointed out, when applied to a market of retail footwear industry, aiming the development of more effective marketing campaigns.

Keywords: Data Mining. Marketing campaigns. Discovery of Knowledge. Retail footwear.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

BI	*Bussines Intelligence*
DM	*Data Mining*
DW	*Data Warehouse*
KDD	*Knowledge Database Discovery*
MD	Mineração de Dados
MER	Modelo Entidade Relacionamento
OLAP	*On-Line Analytic Processing*
SAD	Sistemas de Apoio à Decisão
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	*Structured Query Language*
TI	Tecnologia de Informação

SUMÁRIO

INTRODUÇÃO

Devido ao crescente volume de dados gerados a cada dia pelas organizações, somado ao aumento constante da competitividade do mercado, surge a necessidade da utilização de ferramentas capazes de gerar conhecimento a partir dos dados armazenados. Considerando um cenário de mercado globalizado, onde a concorrência é mais acirrada, as empresas necessitam reduzir cada vez mais suas margens de lucros para se manter competitivas. Para tanto, torna-se essencial utilizar as informações de forma mais ativa.

Com o intuito de auxiliar as empresas nessa exploração de dados, alguns conceitos e ferramentas para organizar as informações se fazem necessárias. Destacam-se o *Data Warehouse* (DW), *Data Mart* (DM), *Bussines Intelligence* (BI) e ferramentas OLAP que formam os pilares estratégicos dos Sistemas de Apoio a Decisão (SAD)[1]. Existe também o processo de Descoberta do Conhecimento em Bancos de Dados (*Knowledge Discovery in Database* - KDD). Esses recursos têm aumentado o seu grau de maturidade e também sua participação no orçamento dos gestores de TI, devido a extração de padrões e conhecimento da base de dados ser uma das áreas do negócio, onde é possível ter vantagens competitivas de forma tangível.

> A análise de grandes quantidades de dados pelo homem é inviável sem o auxílio de ferramentas computacionais apropriadas. Portanto, torna-se imprescindível o desenvolvimento de ferramentas que auxiliem o homem, de forma automática e inteligente, na tarefa de analisar, interpretar e relacionar esses dados para que se possa desenvolver e selecionar estratégias de ação em cada contexto de aplicação. (GOLDSCHMIDT, 2005, p.1).

[1] O Sistema de Apoio a Decisão, segundo Sprague (1991), é um sistema de informação que apóia qualquer processo de tomada de decisão em áreas de planejamento estratégico, controle gerencial e controle operacional.

Segundo Elmasri (2005), o Processo de Descoberta de Conhecimento é composto por seis fases, sendo elas: seleção dos dados, limpeza, enriquecimento, transformação ou codificação, *data mining* e construção de relatórios e apresentação da informação descoberta. Durante a etapa de Mineração de Dados, é realizada a busca efetiva por conhecimentos úteis no contexto da aplicação de KDD (GOLDSCHMIDT, 2005).

Mineração de Dados é definida como: "O processo não-trivial de identificar válidos, novos, potencialmente utilizáveis e por último, padrões compreensíveis dentro dos dados." (tradução nossa) (Fayyad, 1996, p. 40 - 41). Isto é obtido através da aplicação de determinados algoritmos que são utilizados para descoberta de padrões nos dados armazenados em uma base.

Vários autores, dentre eles Elmasri (2005) e Goldschmidt (2005), tem apontado *data mining* como uma das tecnologias mais promissoras para o futuro próximo. Os autores citam cinco técnicas de descoberta de conhecimento durante o *data mining*:

- **Regras de Associação** – contempla a busca por itens que ocorram na mesma freqüência em transações de banco de dados. Pode servir de exemplo, a descoberta sobre a melhor aproximação de produtos nas gôndolas de supermercados;
- **Hierarquias de classificação** – o objetivo é um mapeamento de um conjunto de registros através da criação de hierarquia de classes em forma de rótulos categóricos predefinidos. A divisão de clientes de uma financeira em faixas de crédito pode servir como exemplo;
- **Padrões seqüenciais** – é uma extensão da regra de associação com a finalidade de descobrir padrões através de várias transações ocorridas durante um período de tempo. Na regra de associação os padrões a serem conhecidos referem-se a cada transação, denominados padrões intratransações. Na descoberta de seqüências, um número variado de transações é analisado em um período de tempo denominado por padrões intertransações;
- **Padrões com séries temporais** – algumas similaridades podem ser encontradas em uma série temporal de uma seqüência de dados através de movimentos de tendências cíclicas, sazonais e irregulares ou randômicos como vendas diárias ou fechamento de valores de ações que podem servir como exemplo;

- ***Clustering*** **(agrupamento)** – dado pelo particionamento dos registros que compartilhem propriedades comuns entre os elementos do *cluster* que os distinguem de outros *clusters* tendo por objetivo aumentar a similaridade interna do *cluster* e diminuir a similaridade externa;

As técnicas citadas podem apresentar resultados diferentes. Portanto, a partir da aplicabilidade dessas técnicas, é necessário efetuar uma análise para verificar qual se mostra mais eficaz, considerando o tipo de tarefa específica a qual se deseja da Descoberta de Conhecimento.

As tecnologias de *Data Mining* têm sido aplicadas em uma grande variedade de contextos empresariais. No contexto do *marketing*[2], a aplicação de técnicas de *Data Mining* permite descobrir o padrão de comportamento do público consumidor e definir estratégias de *marketing* mais eficazes, padrão esse, que não pode ser obtidos através das ferramentas convencionais de inferência[3] de dados.

Segundo Mackenna (2002, p.28), a tecnologia da informação é agora um componente tão essencial para reagir às mudanças do mercado e satisfazer os clientes que os executivos de muitas áreas da organização estão tomando decisões sobre *marketing* e executando plano de *marketing*. Um exemplo bastante conhecido foi a aplicação feita por uma grande rede de supermercados americana, onde foi descoberto um universo de compradores de fralda que também compravam cerveja nas vésperas de finais de semana em que jogos eram transmitidos na televisão. Esse conhecimento foi utilizado na aproximação das gôndolas desses dois produtos, aumentando assim a venda dos mesmos.

> [...] uso de banco de dados e redes para analisar as tendências do consumidor e da concorrência, realizar várias simulações, compilar informação atual e comunicá-la a todos na empresa que devem adaptar-se e responder de acordo. A rede muda a natureza de quem participa do processo estratégico. Usando a rede, os planos estratégicos podem ser mais abrangentes, monitorados constantemente e adaptados com uma comunicação instantânea. (McKenna, 2002, p.34).

O setor financeiro tem utilizado intensamente esse recurso na análise de concessão de crédito a clientes, análise de *performance* do mercado acionário e também na detecção de

---

[2] **Marketing:** é uma arquitetura integradora que permite o contínuo processo de aprendizado organizacional através do qual a empresa acumula conhecimento por meio da interação contínua com os consumidores e o mercado para aprender, adaptar-se e responder de forma criativa e competitiva. (McKenna, 2002, p.202)

[3] **Inferência: d**eduzir por meio de raciocínio, tirar por conclusão ou conseqüência.

fraudes, buscando por compras no cartão de crédito que divirjam do perfil habitual de compra do proprietário. Na área da saúde, tem sido utilizado em análise dos efeitos colaterais dos medicamentos, na estimativa de sobrevivência de um determinado paciente, considerando o resultado diagnosticado através dos exames.

No setor do varejo calçadista, recursos computacionais como a mineração de dados ainda são pouco utilizados. Na maioria dos casos os investimentos são direcionados para melhorar a estrutura da área de vendas, ficando esse tipo de investimento em segundo plano. Outro fator é o desconhecimento dos gestores sobre esses recursos e qual o retorno financeiro real que pode trazer através da aplicação da mineração dos dados, produzindo campanhas de *marketing* com menores investimentos e maior retorno. Segundo Mackenna (1999, p. 53) a tecnologia de computadores projetada especificamente para automatizar as funções de *marketing* e vendas sem dúvida melhora os lucros. Pode gerar conhecimento para reorganizar o *layout* das lojas, dispondo seus produtos em um melhor formato, com base na análise do padrão de consumo identificado nas informações dos clientes.

Com base nos conceitos citados, o objetivo deste trabalho é identificar os resultados obtidos pelas técnicas e algoritmos de mineração, buscando o que possua a maior aderência a uma base de clientes do setor de varejo calçadista, através da utilização da ferramenta Weka que implementa essas técnicas. Dessa maneira, espera-se gerar conhecimento que permita o desenvolvimento de campanhas de *marketing* mais eficazes, aumentando o retorno financeiro e diminuindo o custo destas, aproveitando de forma eficiente as informações disponíveis.

Este trabalho está estruturado em 6 capítulos. O capítulo 1 discorre sobre o conceito e as etapas da descoberta de conhecimento em banco de dados. O capítulo 2 concentra-se na explanação das técnicas e algoritmos utilizados na etapa de *Data Mining*. O capítulo 3 trata sobre a ferramenta de mineração Weka e também dos algoritmos implementados por ela. O capítulo 4 aborda algumas características da empresa do setor de varejo calçadista e também aspectos da modelagem, seleção e pré-processamento dos dados. O capítulo 5 aborda a aplicação do algoritmo de associação apresentando os resultados obtidos. O capítulo 6 discorre sobre o algoritmo de classificação e os métodos de meta aprendizagem *Bagging* e *Boosting* com apresentação dos resultados gerados em cada algoritmo. Completam o documento a conclusão e a bibliografia.

1 DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Devido à crescente evolução da área de tecnologia da informação, os meios de armazenamento de informações têm aumentado muito a sua capacidade e reduzido o seu custo. Paralelo a isso, o volume de dados tem aumentando muito nos últimos anos, o que torna cada vez mais complexa a transformação dessa informação em conhecimento útil para as organizações.

Torna-se inviável ao homem efetuar uma análise dessa grande massa de dados, sem a ajuda de um processo de tratamento dos dados e uma ferramenta computacional que o auxilie nessa árdua tarefa. Com intuito de auxiliar a análise, interpretação e transformação do dado em conhecimento, surgiu uma área de estudos denominada Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Database* - KDD) (GOLDSCHMIDT, 2005).

De acordo com Goldschmidt (2005), o termo KDD foi formalizado em 1989 em referência ao amplo conceito de procurar conhecimento a partir de bases de dados. O processo de KDD é interativo e iterativo e composto por etapas seqüenciais (FAYYAD, 1996). Complementando o autor, Goldschmidt (2005) acrescenta que é interativo por indicar a necessidade de envolvimento do homem controlando o processo e iterativo por sugerir a possibilidade de repetições total ou parcial do processo buscando conhecimento satisfatório por meio de refinamentos sucessivos.

Dessa forma o processo de KDD sugere uma estrutura cooperativa entre o homem e a máquina. Humanos definem seus objetivos a fim de identificar novos conhecimentos para auxiliar em sua área de atuação e, por sua vez, computadores processam grandes volumes de dados com o intuito de encontrar padrões válidos, visando satisfazer os objetivos traçados anteriormente pelo humano.

1.1 Etapas do Processo de KDD

Na busca de obter conhecimento em uma base de dados em seu estado bruto, uma das principais tarefas é encontrar o entendimento do domínio da aplicação e ter os objetivos finais definidos de forma clara. Torna-se fundamental que o analista de KDD elabore um modelo de boa qualidade para que o usuário visualize o relacionamento existente entre os atributos, a fim de garantir a qualidade no resultado apresentado ao final do processo.

O processo de KDD é constituído por um conjunto de etapas (figura 1.1), que serão apresentadas a seguir.

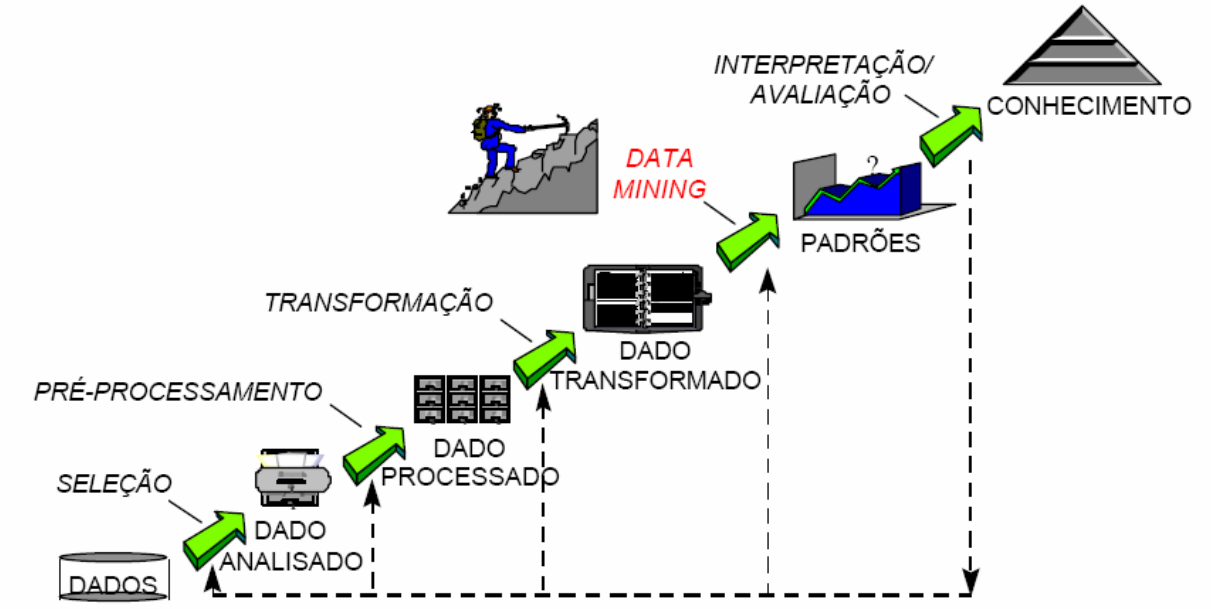

Figura 1.1 – Etapas do processo de KDD
Fonte: FAYYAD et al., 1996

1.1.1 Seleção de Dados

Essencialmente, a etapa de seleção de dados tem o objetivo de identificar os dados que estão armazenados nas mais diversas bases de dados, avaliando a sua importância dentro do domínio e objetivos definidos anteriormente. Esses dados, na maioria das vezes, estão armazenados em bases transacionais, sofrendo alterações constantemente. Goldschmidt

(2005) recomenda que seja feita uma cópia dos dados a fim de que o processo de KDD não interfira nas rotinas operacionais eventualmente relacionadas à base de dados.

Caso os dados estejam armazenados em um *DataWarehouse,* o mesmo autor, sugere que seja utilizada essa estrutura no processo de KDD. Nos demais casos, é comum utilizar a congregação dos dados em uma única tabela. Dessa forma, percebe-se que o processo de KDD independe da disponibilidade ou não de *DataWarehouses*.

1.1.2 Pré-processamento

A etapa de pré-processamento inicia-se após a seleção dos dados. Consiste na limpeza dos dados, através de um pré-processamento, visando adequá-los para a etapa de mineração de dados. Isso se faz através da integração de dados heterogêneos e da eliminação de ruídos e de erros (SCHNEIDER, 2007).

De acordo com Goldschmidt (2005), é importante perceber que a qualidade dos dados possui grande influência na qualidade dos modelos de conhecimento a serem abstraídos a partir destes dados. Quanto pior a qualidade dos dados informados ao processo de KDD, pior será a qualidade dos modelos de conhecimento gerados (GIGO- *Garbage in, Garbage out*). Fayyad (1996) foi um dos primeiros a apontar os objetivos desta fase, indicando que: "Pré-processamento inclui a remoção de ruídos, caso isso seja necessário, deve-se recolher as informações necessárias para o modelo ou explicar o ruído, decidindo sobre as estratégias para o tratamento de dados ausentes nos campos, e apresentar relatório de seqüência das informações e conhecer as alterações" (tradução nossa) (p. 42).

1.1.3 Transformação

Nessa etapa será feita a codificação necessária nos dados, para atender as necessidades específicas dos algoritmos de mineração de dados (GOLDSCHMIDT, 2005). Elmasri (2005) complementa, indicando que o enriquecimento incrementa os dados com fontes adicionais de informações a fim de fornecer mais elementos para o processo de descoberta de conhecimento.

A etapa de transformação é onde os dados de entrada serão recebidos do pré-processamento, já com uma formatação diferente de quando não haviam sido pré-processados.

A transformação será exclusivamente para que os dados já formatados sejam organizados de maneira que a ferramenta e/ou técnica escolhida possa realizar a mineração nos dados.

Cada ferramenta de mineração de dados e/ou técnica pode ter uma maneira especial de receber os dados. As etapas de seleção, pré-processamento e transformação formam a preparação dos dados em um processo de KDD (GONCHOROSKY, 2007).

1.1.4 *Data Mining*

Vários autores, dentre eles Fayyad (1996) e Elmasri (2005) referem-se a *Data Mining* como a principal etapa do processo de KDD, pois será nessa etapa onde ocorrerá a busca efetiva por novos conhecimentos, através da aplicação de técnicas e algoritmos sobre os dados resultantes das etapas anteriores do KDD. No próximo capítulo, esse assunto será discutido de forma mais ampla e detalhada.

1.1.5 Apresentação da Informação Descoberta

Esta última etapa é uma das mais importantes. Nesse momento, o analista de KDD, juntamente com o especialista do domínio, analisam e interpretam os padrões gerados pela etapa de *Data Mining* a fim de identificar qual dos padrões constitui uma nova descoberta.

Aqui o analista poderá identificar a necessidade ou não de reiniciar qualquer um dos passos para mais iterações. Após avaliar os padrões descobertos e identificar sua relevância para a corporação, então é o momento de consolidar o conhecimento gerado, incorporando o mesmo dentro de outros sistemas, documentar ou utilizá-los, auxiliando a tomada de decisão humana (FAYYAD, 1996).

1.2 Considerações Finais

É importante conceituar o processo de descoberta de conhecimento em banco de dados, bem como suas etapas, para visualizar de forma clara a interação entre o homem e os procedimentos computacionais e sua importância na obtenção de sucesso ao término do processo. A maioria dos autores pesquisados define a etapa de mineração de dados, como uma

das mais importantes do processo de KDD, pelo fato de que a descoberta de novos padrões ocorre nessa etapa.

Considerando a importância da etapa de *Data Mining* e sendo ela o foco desse estudo, essa etapa será explicada de maneira mais ampla no próximo capítulo. Cabe ainda salientar que existem outras tendências de pesquisa associadas a mineração de dados como: mineração de textos (*Text Mining*), mineração multimídia e ainda *Web Mining* que refere-se a garimpagem de dados na *Web*.

2 DATA MINING

2.1 Conceito

Data Mining é definido por estudiosos como: "um passo do processo de KDD, que consiste na aplicação de algoritmos de descoberta de dados que, sob certas limitações de eficiência computacional aceitáveis, produzem uma enumeração particular de padrões sobre estes dados." (tradução nossa) (Fayyad, 1996, p.41). Também é definida por Cabena (1997) como o "processo de extrair previamente informação não conhecida, válidas e úteis de grandes bases de dados, utilizando a informação para tomada de decisões no mundo dos negócios" (tradução nossa).

> "Para entender o conceito de *Data Mining*, é importante analisar a tradução literal do verbo minerar. O verbo normalmente refere-se à mineração, ato de extrair recursos preciosos escondidos na Terra. Associando isso com a palavra dados, sugere uma pesquisa profunda para extrair informações adicionais e até então desconhecidas entre a massa de dados disponível". [tradução nossa] (GIUDICI, 2003, p.1).

Alguns autores, dentre eles Elmasri (2005) e Goldschmidt (2005), citam que muitas vezes a mineração de dados (MD) e a descoberta de conhecimento em bases de dados (KDD) são confundidas e conceituadas de forma indistinta, como se fossem sinônimos. Mas na verdade, a mineração é apenas uma etapa no KDD. Pois, é no momento da garimpagem que ocorre a busca efetiva por conhecimento e padrões até então desconhecidos pelo humano.

A mineração de dados é uma área multidisciplinar, relacionando as áreas de estatística, aprendizado de máquina, inteligência artificial/redes neurais e bancos de dados, conforme ilustrado na figura 2.1. Segundo Schneider (2007), apesar da mineração de dados e estatística serem tecnologias distintas, a estatística pode ser considerada a base das tecnologias criadas para mineração de dados. Isto por fazer uso de conceitos como

distribuição normal, variância, análise de regressão, desvio simples, análise de conjuntos, análises de discriminantes e intervalos de confiança para que estes sejam utilizados nas pesquisas, análises e descobertas de relacionamentos entre os dados. Já em relação a aprendizado de máquina, que está voltado para a otimização de um agente, a mineração de dados preocupa-se em buscar conhecimento compreensível em grandes conjuntos de dados.

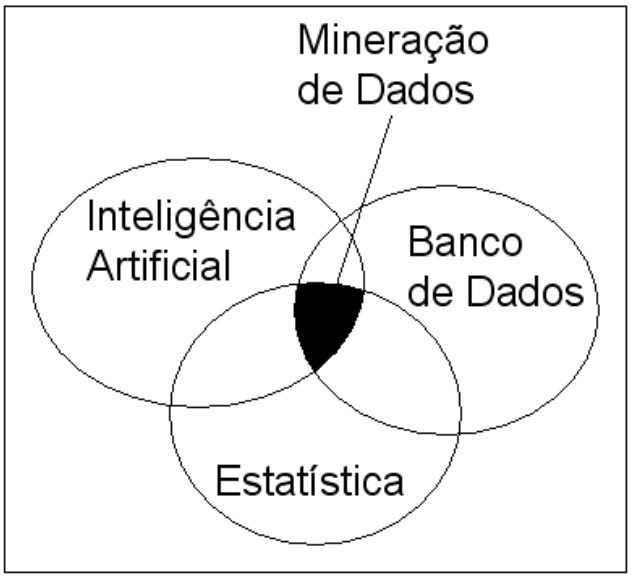

Figura 2.1 – *Data mining* e sua interdisciplinaridade
Fonte: Adaptado de CARVALHO, 1999

Segundo Elmasri (2005), existe uma evolução do dado para informação e para o conhecimento à medida que o processamento evolui. O autor ainda classifica o conhecimento em indutivo e dedutivo. O conhecimento dedutivo deduz novas informações a partir da aplicação de regras lógicas predefinidas de dedução sobre os dados existentes. Já o conhecimento indutivo, é onde o *Data Mining* apóia-se para descobrir novas regras e padrões a partir dos dados fornecidos.

O autor também apresenta cinco técnicas comumente realizadas para descrever o conhecimento descoberto durante o *Data Mining,* conforme apresentado nas próximas seções.

2.2 Regras de associação

Essa técnica tem a intenção de identificar associação entre itens que ocorram na mesma freqüência em uma mesma transação de banco de dados. Inúmeras aplicações dessa

técnica podem ser citadas como exemplo: campanhas de marketing direcionadas, supermercados, planejamento de promoções de vendas e controle de estoque (SCHNEIDER, 2007).

De acordo com Goldschmidt (2005), uma regra de associação é uma implicação da forma $X \Rightarrow Y$, onde X e Y são conjuntos de itens tais que $X \cap Y = \varnothing$. É importante enfatizar, que nessa regra, X é definido como o antecedente, e Y como conseqüente, onde a interseção vazia entre eles garante que não serão extraídas regras óbvias, indicando que um item está associado a ele mesmo.

A seguir estão indicados dois exemplos de regras de associação. A regra (1) indica que a compra de bota pode levar a compra de bolsa e a regra (2) que a compra de bolsa e sapato, pode levar a compra de meia.

(1) Bota → Bolsa

(2) Bolsa ∧ Sapato → Meia

Quadro 2.1 – Relação das vendas de uma loja de calçados

Transação	Bota	Meia	Tênis	Bolsa	Sapato	Sandália	Cinto
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

Fonte: Adaptado de GOLDSCHMIDT, 2005

Uma alternativa para a representação de dados mostrado no quadro 2.1 é denominada de formato *basket* conforme observado no quadro 2.2.

Quadro 2.2 – Formato *Basket* da relação das vendas do quadro 2.1

Transação	Item
1	Meia
1	Bolsa
1	Sapato
2	Bota
2	Tênis

2	Bolsa
2	Sapato
3	Meia
3	Bolsa
3	Sapato
4	Meia
4	Bota
4	Bolsa
4	Sapato
5	Tênis
6	Sapato
7	Bolsa
8	Cinto
9	Sandália
9	Cinto
10	Sandália

Fonte: Adaptado de GOLDSCHMIDT, 2005

Suporte mínimo e confiança mínima são fatores considerados relevantes, para que uma regra de associação seja avaliada. Suporte é calculado através da freqüência que a transação ocorre no banco de dados, com determinado conjunto de itens, e dividido pelo número total de transações. Se o percentual do suporte gerado for baixo, sugere que não existe evidência expressiva que os itens em $X \cup Y$ ocorram juntos, considerando que o conjunto de itens ocorre em uma pequena fração de transações (ELMASRI, 2005).

Suporte = $X \cup Y$ / número total de registros

No exemplo do quadro 2.1 e 2.2, as regras (1) e (2) possuem suporte de 20% e 30% respectivamente.

A confiança da regra é calculada de forma semelhante ao suporte, porém a divisão é feita considerando apenas a freqüência em que X ocorre.

Confiança = $X \cup Y$ / número de registro com X

A medida de confiança tem por objetivo, demonstrar a qualidade da mesma, indicando o quanto a ocorrência do antecedente da regra pode garantir a ocorrência do conseqüente (GOLDSCHMIDT, 2005).

Considerando o exemplo do quadro 2.1 e 2.2, as regras (1) e (2) possuem confiança de 20% e 30% respectivamente.

Como exemplo de algoritmos de regra de associação, pode ser citado o Apriori, por ser um dos mais utilizados e referenciados na literatura dentre outros existentes. Essa técnica

revela a afinidade oculta existente entre produtos que freqüentemente são comprados em uma mesma transação. Com base nisso, é possível identificar características de um potencial comprador, permitindo desenvolver campanhas de *marketing* direcionadas.

2.3 Hierarquias de Classificação

A técnica de classificação tem o objetivo de mapear um conjunto de registros, examinando suas características através de uma função para enquadrá-lo corretamente a um único rótulo categórico predefinido conforme mostra figura 2.2. Segundo Goldschmidt (2005), é uma das técnicas mais importantes e populares de *Data Mining* e pode ser formalizada considerando um par ordenado da forma *(x,f(x))*, onde *x* representa um vetor de entradas e *f(x)* a saída de uma função f, desconhecida, que foi aplicada a *x*.

Uma vez identificada, essa função pode ser aplicada a novos registros com o objetivo de prever em qual rótulo eles se enquadram, sendo denominado pelo autor como algoritmo de aprendizado (GOLDSCHMIDT, 2005).

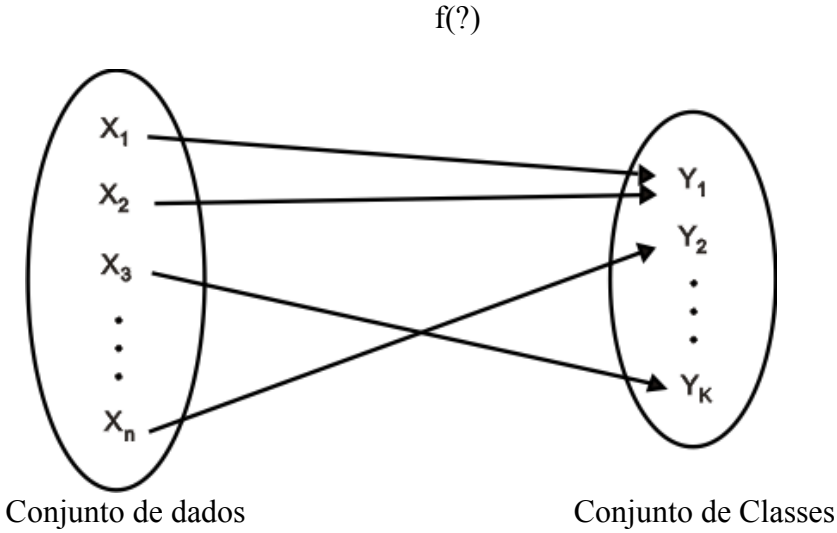

Figura 2.2 – Associações entre registros de dados e classes
Fonte: GOLDSCHMIDT et al., 2005

É definido por Elmasri (2005) como processo de aprendizado supervisionado. Onde o primeiro passo do modelo de aprendizado é efetuado utilizando um treinamento a partir de um conjunto de dados já classificados anteriormente. Cada registro do conjunto de treinamento terá um atributo, definido como rótulo, indicando a que classe ele pertence. O autor ainda cita algumas preocupações pertinentes em relação ao modelo e o algoritmo que o

produz, que são a habilidade do modelo em categorizar corretamente os novos dados, seu custo computacional ligado ao algoritmo e a sua escalabilidade.

Goldschmidt (2005) sugere que um conjunto de hipóteses podem ser alcançadas a partir do algoritmo de aprendizado. Essas hipóteses são chamadas de classificador. Conforme representada geometricamente na figura 2.3, três hipóteses possíveis induzidas a partir do conjunto de exemplos da figura 2.2.

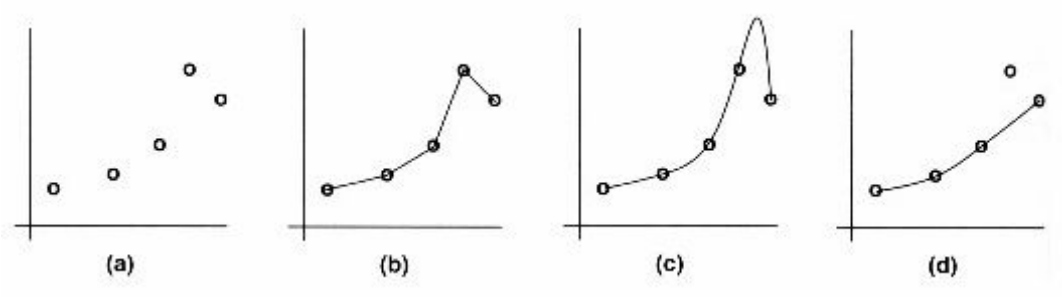

Figura 2.3 – Hipóteses de funções induzidas a partir dos exemplos de entradas e saídas
Fonte: GOLDSCHMIDT et al., 2005

Como exemplo de classificação, é possível considerar os clientes de uma loja de calçados residentes em uma determinada cidade e categorizá-los de acordo com o tipo de calçados comprados por eles anteriormente. Essas categorias podem ser divididas em: esporte, social, casual e infantil. O objetivo dessa classificação é prever em qual dos rótulos o restante dos clientes se enquadram, para desenvolvimento de uma campanha de marketing focada por categoria.

Os algoritmos J48.J48, J48.PART, C4.5, ID3, redes neurais *Back-Propagation*, classificadores Bayesianos, podem ser usados para aplicação na técnica de classificação.

2.4 Padrões Seqüenciais

A descoberta de padrões seqüenciais é uma extensão da regra de associação. Essa técnica tem por objetivo descobrir um modelo através de várias transações ocorridas durante um período de tempo, buscando encontrar por padrões a partir de um conjunto de transações conceituado como, intertransações, tornado evidente uma complexidade maior na busca por padrões. Já a regra de associação citada anteriormente, tem por objetivo determinar os padrões

de uma única transação, denominada intratransações, diferenciando assim, uma regra da outra (GOLDSCHMIDT, 2005).

De acordo com Elmasri (2005), o problema de identificar padrões seqüenciais está em, localizar todas as subseqüências de um dado conjunto de encadeamento que tenha um suporte mínimo previamente definido pelo usuário. A seqüência S_1, S_2, S_3,... é um predecessor do fato de que um cliente que tenha efetuado a compra do conjunto S_1, esteja predisposto a comprar S_2, S_3, e assim sucessivamente. Essa previsão baseia-se no suporte dessa série no passado.

Goldschmidt (2005) cita como exemplos de algoritmos para essa aplicação o *Generalized Sequential Patterns* (GSP) e *Sequential Pattern Discovery Using Equivalence Classes* (SPADE). O autor complementa que esses algoritmos baseiam-se na propriedade de antimonotonicidade do suporte: "Uma *k-seqüência* somente pode ser freqüente se todas as suas *(k-1)-subseqüência* forem freqüentes". O suporte de uma seqüência deve ser mantido mesmo que tenha expandido para uma série com mais conjuntos de itens, nunca podendo crescer por conta disto.

2.5 Padrões com Séries Temporais

Uma série temporal caracteriza-se pelo acompanhamento de um fenômeno ordenado que ocorra em um determinado período de tempo, buscando encontrar similaridade na seqüência. A análise de uma série temporal é o método de identificação das peculiaridades, dos modelos e das propriedades importantes da seqüência. Através dessa análise, é possível descrever de forma simplificada, o seu elemento gerador (GOLDSCHMIDT, 2005).

O autor ainda complementa, citando quatros principais tipos de movimentos que normalmente são utilizados para caracterizar as séries temporais, sendo elas:

- **Movimentos de Tendência** – sinaliza a direção genérica que o gráfico utilizará para se mover ao longo do tempo;
- **Movimentos Cíclicos** – refere-se aos movimentos alternados de uma curva, que pode a mesma ser periódica ou não. Indica que, os ciclos não precisam fundamentalmente após intervalos de tempos iguais, seguir exatamente os mesmos padrões;

- **Movimentos Sazonais –** referem-se aqueles eventos que se repetem de tempos em tempos. São movimentos similares um ao outro e sua ocorrência se dá, em função de uma data especial, como aumento das vendas de calçados na semana que antecede o dia das mães ou ainda dia dos namorados;
- **Movimentos Irregulares ou Randômicos** – já esse tipo de movimento ocorre em função de fatores climáticos. Como exemplo é possível citar o aumento nas vendas de botas durante as semanas em que as temperaturas caem;

Segundo Goldschmidt (2005), em séries temporais a busca por similaridade envolve a identificação da seqüencia de dados que, em relação ao padrão apresentado demonstraram pouca variação. As buscas por similaridade podem ser divididas em combinação de subseqüências, que consiste em localizar todas as seqüências de dados da série, que se igualam ao padrão apresentado ou aproximação de seqüencias, que busca encontrar a série de dados que mais se assemelham com a série em análise. A figura 2.4 apresenta o comportamento de três produtos em uma determinada janela de tempo, em que uma série foi analisada. É possível concluir que o produto C é menos volátil do que os produtos A e B e que ambos seguem a mesma política de estoque e também, pode-se fazer predição com boa certeza para o produto C (CARVALHO, 2008).

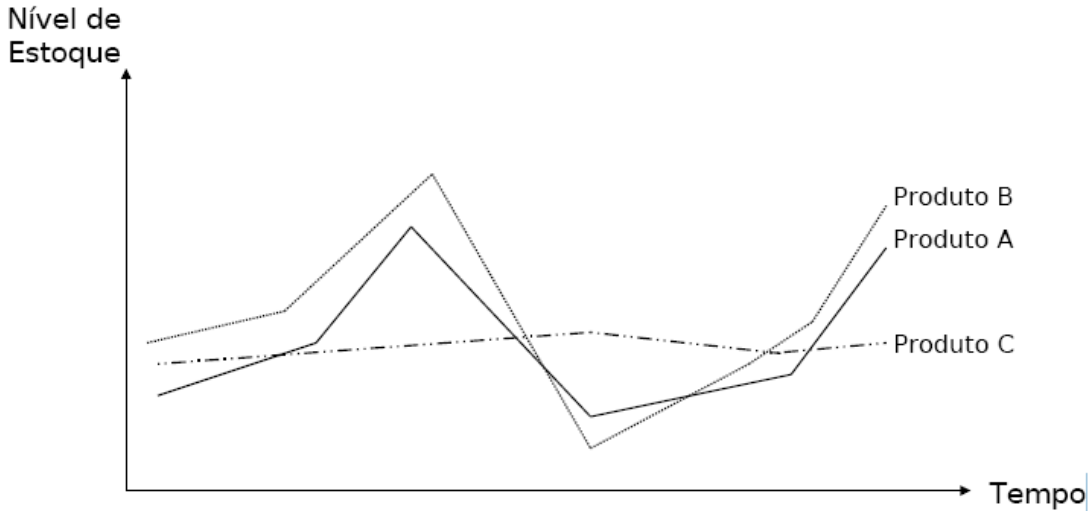

Figura 2.4 – Aplicação de previsão de séries temporais
Fonte: CARVALHO, 2008

Essa técnica pode revelar o comportamento de um determinado produto apresentando os períodos sazonais ou ainda demonstrar a tendência de venda do mesmo. A diminuição dos riscos nas tomadas de decisão e o auxilio ao planejamento da empresa, podem

valer-se da previsão de séries temporais considerando que a eficácia de uma decisão está vinculada, na maioria das vezes, a eventos ocorridos anteriormente.

2.6 *Clustering* (agrupamento)

A técnica de *Clustering* ou agrupamento é citado por Elmasri (2005) como aprendizado não supervisionado. Diferentemente da técnica de classificação citada anteriormente, o agrupamento divide os dados sem a necessidade de uma amostra de treinamento. Através do particionamento dos registros que partilham propriedades comuns entre os elementos do *cluster*, busca aumentar a semelhança interna do *cluster*, reduzindo a similaridade intercluster.

"*Clustering* é uma técnica descritiva comum onde um rótulo identifica um grupo finito de categorias ou *cluster* para descrever os dados" (JAIN; DUBES apud FAYYAD et al., 1996, p.45). O autor ainda cita que as categorias podem ser mutuamente exclusivas e exaustivas ou consistir de uma representação mais rica, como hierárquicas sobrepostas ou categorias (TRADUÇÃO NOSSA).

Goldschmidt (2005) acrescenta que, pelo fato da clusterização auxiliar os usuários a realizarem agrupamentos naturais dos registros em conjuntos de dados, pode ser definida como uma das técnicas básicas da mineração de dados. Geralmente, essa técnica requer que o usuário informe o número de grupos que devem ser considerados. A partir do número informado, os registros podem ser agrupados no mesmo *cluster*, considerando a similaridade existente entre eles. Feito esse agrupamento, é possível efetuar uma análise dos elementos que compõem cada *cluster* e com base nas características comum entre os registros, criar um rótulo para representar o grupo. Na figura 2.5 é possível visualizar no gráfico os 3 *cluster*s gerados a partir da aplicação da técnica sobre dados fictícios.

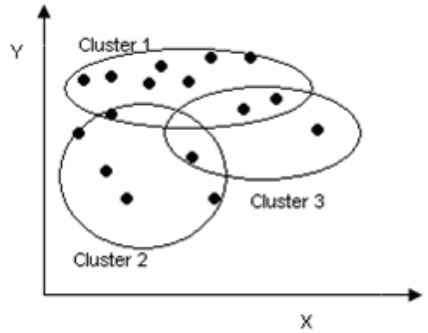

Figura 2.5 – Representação de três *clusters* gerados com a técnica
Fonte: Adaptado de FAYYAD et al., 1996, p. 14

Os métodos de clusterização mais utilizados e conhecidos são os métodos por particionamento e os métodos hierárquicos.

> Os algoritmos de clusterização por particionamento dividem a base de dados em *k* grupos, onde o usuário escolhe o número de *k*. Inicialmente, estes algoritmos escolhem *k* objetos como sendo os centros dos *k* clusters. Os objetos são então divididos entre os *k* clusters de acordo com a medida de similaridade adotada, de modo que cada objeto fique no cluster que forneça o menor valor de distância entre o objeto e o centro do mesmo. Os algoritmos utilizam então, uma estratégia iterativa, que determina se os objetos devem mudar de cluster, fazendo com que cada cluster contenha somente elementos similares entre si. (GOLDSCHMIDT, 2005).

De acordo com Elmasri (2005), o algoritmo inicia com a escolha aleatória de *k* registros para apresentar a centróide (média). Baseados na distância existente entre os registros e a média do agrupamento, o total de registro é colocado em um agrupamento especifico. Após todos os registros terem sido colocados em um agrupamento inicial, é recalculada a média do agrupamento. Esse processo poderá ser repetido várias vezes, onde cada registro é examinado e colocado em um novo agrupamento cuja média é mais próxima.

Existem várias técnicas e processo de clusterização. Entre os principais algoritmos de clusterização podem ser citados: K-Means, Fuzzy K-Means, K-Modes e K-Medoid (GOLDSCHMIDT, 2005).

Essa técnica poderá ser utilizada para criar *clusters* de clientes, considerando alguns atributos como: idade, sexo, renda e calçados adquiridos anteriormente. Esses agrupamentos possibilitariam a visualização de segmentos de clientes que compram um determinado tipo de calçado específico. Isso permitiria montar campanhas de marketing direcionadas de acordo com cada agrupamento de cliente, objetivando uma maior eficiência e eficácia, gerando um maior retorno financeiro com um investimento menor.

3 WEKA

Como foi citado anteriormente, para conseguir garimpar uma grande quantidade de dados e obter conhecimento válido e utilizável, surge a necessidade da utilização de ferramentas capazes de gerar conhecimento a partir dos dados armazenados. Considerando essa necessidade, ferramentas têm sido disponibilizadas no mercado, para auxiliar o homem na execução do processo de KDD e especificamente na mineração dos dados. Algumas delas são pagas, desenvolvidas por grandes empresas de *software*, como: *Intelligent Miner* fabricado pela IBM, Oracle *Data Mining* construído pela Oracle e também o SAS *Enterprise Miner* da empresa SAS. Existem também, ferramentas *open source* como: *Pentaho*, *Rapid Miner* e o Weka.

O Weka será a ferramenta utilizada no desenvolvimento desse estudo, devido sua popularidade no meio acadêmico, ser de fácil utilização, além de ser gratuita. Nas próximas seções serão detalhadas algumas funcionalidades, *interfaces* e algoritmos implementados pelo Weka.

3.1 Características do WEKA

Waikato Environment for Knowledge Analysis – WEKA é uma ferramenta de código aberto disponível na *Web*. *F*oi desenvolvida na linguagem de programação Java pelo curso de ciência da computação da universidade de Waikato na Nova Zelândia (GOLDSCHMIDT, 2005) e está disponível no site da ferramenta (WEKA, 2008). O autor ainda cita que a ferramenta possui quatro diferentes implementações de interface sendo elas:

- ***Simple Client*** – nessa interface o usuário interage com o Weka através de linhas de comando onde requer um conhecimento aprofundado do programa, tornando-se bastante flexível e rápida para usuários avançados;
- ***Explorer*** – essa é a interface de utilização mais comum, e trata de forma separada as etapas de pré-processamento (filtros), mineração de dados (associação, clusterização e classificação) e pós-processamento (apresentação dos resultados);
- ***Experimenter*** – é um ambiente onde é possível fazer experimentos através de testes estatísticos, com o intuito de avaliar o desempenho de diferentes algoritmos;
- ***KnowledgeFlow*** – essa é uma interface gráfica que permite construir o fluxo dos processos de KDD e também efetuar o planejamento das ações;

Diversos métodos de associação, classificação e clusterização estão implementados no Weka. Pelo fato de ser uma ferramenta de código aberto, a inclusão ou remoção de novos métodos pode ser efetuada de forma rápida e simples, tornado uma ferramenta customizável e expansível. Também é possível visualizar os dados de forma gráfica através de histogramas, e os resultados podem ser apresentados na forma de árvores de decisão, diagramas de dispersão e ainda gerar modelos gráficos para montagem de redes neurais (GOLDSCHMIDT, 2005). Um resumo das características do Weka está relacionado no quadro 3.1.

Quadro 3.1 – Características do Weka

Características		Valores
Acesso a Fontes de Dados Heterogêneas		Sim
Integração de Conjuntos de Dados		Não
Facilidade para Inclusão de Novas Operações		Sim
Facilidade para Inclusão de Novos Métodos		Sim
Recursos para Planejamento de Ações		Sim
Processamento Paralelo/Distribuído		Não
Operações/Métodos Disponíveis	Visualização de Dados	Distribuição de Freqüências; Medidas de Dispersão; Histogramas
	Redução de Dados	Amostragem
	Limpeza de Dados	Substituição
	Codificação de Dados	Discretização automática e manual
	Classificação	Árvores de Decisão, Bayes, Redes Neurais...
	Clusterização	Simple-KMeans, Cobweb, FarthestFirst...
	Simplificação de Resultados	N/D
	Organização de Resultados	Agrupamento de Padrões; Ordenamento de Padrões
	Apresentação de Resultados	Conjunto de Regras; Árvores de Decisão
Estruturas para Armazenamento de Modelos de Conhecimento		Sim
Estruturas para Armazenamento de Históricos de Ações		Sim

Fonte: GOLDSCHMIDT, 2005

Segundo Santos (2005) e Weka (2008), o Weka manipula um arquivo com extensão .ARFF que contém texto puro e é composto de três partes:

- **Relação –** a primeira linha do arquivo deve conter a identificação da relação ou tarefa que esta sendo estudada, sendo, antecedida da expressão @relation;
- **Atributos –** logo abaixo, uma lista atributos é relacionada, onde cada linha inicia com @attribute acompanhada do nome do atributo e seguida do seu tipo, que pode ser nominal (as alternativas devem ser relacionadas como uma lista separada por vírgulas e cercadas por chaves) ou numérico (neste caso o nome deve ser seguido do tipo de dado);
- **Dados –** depois, uma linha simples contendo a expressão @data indica o início da relação de dados do arquivo. Cada linha representa uma instância e deve ter valores separados por vírgula correspondentes (e na mesma ordem) dos atributos da seção Atributos;

O arquivo também pode conter linhas de comentários que não serão processadas. Essas linhas devem iniciar com o sinal de percentagem (%). Na figura 3.1 pode-se verificar o formato do arquivo .ARFF.

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%      (a) Creator: R.A. Fisher
%      (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%      (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Figura 3.1 – Arquivo .arff do Weka
Fonte: WEKA, 2008

A ferramenta Weka foi escolhida para o desenvolvimento desse trabalho, por vários motivos: apresentar os principais algoritmos e técnicas de DM, estar sendo utilizada em vários trabalhos acadêmicos sobre mineração de dados, disponível na web e ser de fácil instalação e

utilização. Como exemplo de utilização do Weka, podem ser citado trabalhos como Gonchorosky (2005) e Oliveira (2002).

É importante ressaltar, que este estudo está sendo desenvolvido sobre uma base de dados relacional a partir do SGBD de uma empresa do setor de varejo calçadista. Considerando que cada algoritmo necessita de uma preparação e tratamento específico para ser submetido ao processamento, as informações dos clientes, como idade, sexo, renda, etc e dados de suas respectivas compras, serão organizadas de forma a atender os requisitos de cada algoritmo.

A seguir, serão apresentados alguns algoritmos presentes no Weka. Os algoritmos de padrões seqüenciais e padrões com séries temporais não serão abordados nesse capítulo, porque a ferramenta não os implementa.

3.2 Algoritmo de Regra de Associação

O algoritmo *Apriori* é um clássico de mineração de regras de associação (AGRAWAL, apud GOLDSCHMIDT, 2005). O *Apriori*, apresentado na figura 3.2, busca identificar conjuntos de itens que ocorram concomitantemente na mesma transação, denominando esse conjunto como *itemsets*. Elmasri (2005) e Goldschmidt (2005) citam o principio da antimonotonicidade[4] para ajudar a reduzir o espaço de busca por soluções possíveis. Os autores salientam a necessidade inicial da definição de valores mínimos para suporte e confiança a serem considerados pelo *Apriori*.

[4] "Uma *k-seqüência* somente pode ser freqüente se todas as suas *(k-1)-subseqüência* forem freqüentes" (GOLDSCHMIDT, 2005).

```
Function Apriori(Banco de Transações D, Smin) Grandes Conjuntos de Itemset (L)

L1 = {grande 1-itemsets};
k = 2;

Enquanto Lk-1 <> 0 faça
        Inicio
                Ck = apriori-gen(Lk-1; {gera os novos candidatos}
                Para toda  transação t Є D faça

                        Inicio

                                Ct = subconj(Ck, t); {candidatos contidos em t}

                                Para todo candidato c Є Ct faça

                                        contador(c) = contador(c) + 1

                        fim;

                Lk = {c Є Ck | contador(C) ≥ Smin};

                k = k + 1

        fim;

Retorna (L = Uk Lk).
```

Figura 3.2 – Algoritmo Apriori
Fonte: MONGIOVI , 1998

O quadro 3.2 representa um banco de dados D referente às venda de uma loja de calçados, com 4 transações sendo o suporte mínimo de 2 (S_{min} = 0.5) e confiança mínima (C_{min}) 0.8, que será utilizado para exemplificar as iterações do algoritmo.

Quadro 3.2 – Transações de venda de uma loja de calçados

Transação	Bota	Meia	Bolsa	Sapato	Cinto
1	Sim	Não	Sim	Sim	Não
2	Não	Sim	Sim	Não	Sim
3	Sim	Sim	Sim	Não	Sim
4	Não	Sim	Não	Não	Sim

Mongiovi (1998) e Goldschmidt (2005) decompõem o algoritmo *Apriori* em basicamente duas etapas. A primeira delas se caracteriza por encontrar todos os grandes conjuntos de itens freqüentes denominados, *itemsets* (L_k) que satisfaçam à condição de suporte mínimo (S_{min}). Sendo o suporte assim calculado:

Suporte(X) = Número de registros que contêm X / Número total de registros

Na figura 3.3 está representado o grande conjunto (L_1), após o conjunto de candidatos em potencial (C_1) ter sido avaliado e excluído os elementos que não possuíam o

suporte mínimo previamente definido como 2. Nesse exemplo o elemento “Sapato” não atingiu o suporte mínimo, tendo somente uma venda na transação 1.

C_1

Itemset	Suporte
Bota	2
Meia	3
Bolsa	3
Sapato	1
Cinto	3

→

L_1

Itemset	Suporte
Bota	2
Meia	3
Bolsa	3
Cinto	3

Figura 3.3 – Conjunto de candidatos em potencial (C1) a grandes *itemsets*
Fonte: Adaptado de MONGIOVI, 1998

Conjuntos com dois e três itens também são formados a partir de novas iterações feitas no banco de dados D seguindo o mesmo conceito citado acima para verificar o suporte mínimo. Sendo assim, já é possível gerar os grandes conjuntos como mostra a figura 3.4 e a figura 3.5.

C_2

Itemset	Suporte
{Bota, Meia}	1
{Bota, Bolsa}	2
{Bota, Cinto}	1
{Meia, Bolsa}	2
{Meia, Cinto}	3
{Bolsa, Cinto}	2

→

L_2

Itemset	Suporte
{Bota, Bolsa}	2
{Meia, Bolsa}	2
{Meia, Cinto}	3
{Bolsa, Cinto}	2

Figura 3.4 – Grande conjunto de dois elementos
Fonte: Adaptado de MONGIOVI, 1998

C_3

Itemset	Suporte
{Meia, Bolsa, Cinto}	2

→

L_3

Itemset	Suporte
{Meia, Bolsa, Cinto}	2

Figura 3.5 – Grande conjunto de três elementos
Fonte: Adaptado de MONGIOVI, 1998

A segunda etapa tem o objetivo de gerar as regras de associação a partir de um grande conjunto de itens freqüentes (X), com fator de confiança mínimo (C_{min}) estabelecido previamente. Sendo as regras de associação assim geradas:

$\forall Y \subset X$, *se suporte*$(X \cup Y)$ */ suporte* $(X - Y) \geq C_{min}$, *então gera a regra* $X - Y \Rightarrow Y$,

onde o fator de confiança de uma regra R: $X \Rightarrow Y$, é definido como:

Confiança = $X \cup Y$ / número de registro com X

Ainda nessa etapa, o algoritmo trata de fazer o corte (poda) das combinações que não aparecem na freqüência que foi estipulada anteriormente (Suporte e Confiança). Na figura 3.2 onde o algoritmo é apresentado, é possível notar a chamada da função apriori-gen(L_{k-1}), que tem por objetivo gerar novos candidatos. De acordo com Mongiovi (1998), a idéia deste algoritmo é unir os elementos de L_{k-1}, 2 a 2 e manter apenas aqueles em que todos os seus subconjuntos de tamanho k-1 pertençam a L_{k-1}. Isso ocorre em dois momentos: junção e poda.

Junção:

Insert into C_k
From $G_{k-1}(p)$, G_{k-1} (q) {elementos p e q de G_{k-1}}
Select p.item1 = q.item1, ..., p.itemk-2 = q.itemk-2, p.itemk-1 < q.itemk-1;

Considerando a junção do *itemsets* apresentados acima, foi gerado o conjunto de regras possíveis, conforme mostra figura 3.6.

Regra	Fator de confiança
{Bota} ⇒ Bolsa	1,00
{Bolsa}⇒ Bota	0,67
{Meia} ⇒ Bolsa	0,67
{Bolsa}⇒ Meia	0,67
{Meia} ⇒ Cinto	1,00
{Cinto}⇒ Meia	1,00
{Bolsa} ⇒ Cinto	0,67
{Cinto} ⇒ Bolsa	0,67
{Meia,Bolsa} ⇒ Cinto	1,00
{Meia,Cinto} ⇒ Bolsa	0.67
{Cinto,Bolsa} ⇒ Meia	1,00
{Meia} ⇒ Bolsa,Cinto	0,67
{Bolsa} ⇒ Meia,Cinto	0,67
{Cinto} ⇒ Meia, Bolsa	0,67

Figura 3.6 – Regras geradas no momento da junção

Poda:

A idéia da poda é eliminar todo c pertencente a C_k tal que algum (k-1) subconjunto de c não pertence a L_{k-1}.

$C_k = \{c \in C_k \mid \forall s \subset c, s \in L_{k-1}\}$

para todo $c \in C_k$ faça
  para todo $s \subset c$ e $|s| = k-1$ faça
    Se $s \notin L_{k-1}$ então elimina c de C_k.

Aplicando a poda no conjunto de regras possíveis que foram geradas conforme mostra figura 3.6 e considerando o fator de confiança mínimo, a figura 3.7 é obtida como resultado.

Regra	Fator de confiança
{Bota} ⇒ Bolsa	1,00
{Meia} ⇒ Cinto	1,00
{Cinto}⇒ Meia	1,00
{Meia,Bolsa} ⇒ Cinto	1,00
{Cinto,Bolsa} ⇒ Meia	1.00

Figura 3.7 – Regras com confiança acima do mínimo

O interesse em pesquisar o algoritmo *apriori*, é aplicá-lo na base de dados de uma rede de lojas do setor de varejo calçadista, gerando regras de associação para as transações de vendas. Após os resultados serão analisados e verificados se estes são aplicáveis e trazem informações importantes para o setor de *marketing* efetuar campanhas mais eficazes. O exemplo visto anteriormente possui uma quantidade pequena de registros, mas foi útil para compreender a aplicação da técnica de regras de associação.

3.3 Algoritmos de Hierarquias de Classificação

Os algoritmos de classificação consistem basicamente em produzir um modelo de classificação, denominado classificador, a partir de um conjunto de registros existentes, para que posteriormente esse modelo seja utilizado para classificar outros exemplos de classe desconhecida (CARVALHO,2000).

A ferramenta Weka trabalha com dois algoritmos de classificação sendo eles: J48.J48 que é uma versão do algoritmo C4.5 release 8 que foi desenvolvido na linguagem *Java* para a ferramenta Weka e o J48.PART pertencentes a família J48. O algoritmo C4.5 foi substituído posteriormente pelo C5.0 que é uma versão mais recente, disponível apenas comercialmente (QUINLAN, 1999).

O Weka organiza os algoritmos internamente em uma estrutura de pacotes. As implementações de algoritmos de classificação tais como: Weka.classifiers.J48.J48 e o Weka.classifiers.J48.PART, fazem parte do pacote weka.classifiers. Uma breve descrição desses algoritmos é apresentada a seguir.

3.3.1 Algoritmo J48.J48

De acordo com Oliveira (2002) o J48.J48 é considerado um dos mais conhecido algoritmo do Weka, ele desenvolve um modelo de árvore de decisão que baseia-se num conjunto de dados de treinamento utilizando, posteriormente, esse modelo para classificar outras instâncias em um conjunto de teste. Alguns parâmetros como o uso de podas na árvore, número mínimo de instâncias por folha, construção de árvore binária e outros mais, podem ser modificados conforme a tela de parâmetros da ferramenta que mostra a figura 3.8, durante o processo de utilização do algoritmo com o intuito de proporcionar melhores resultados.

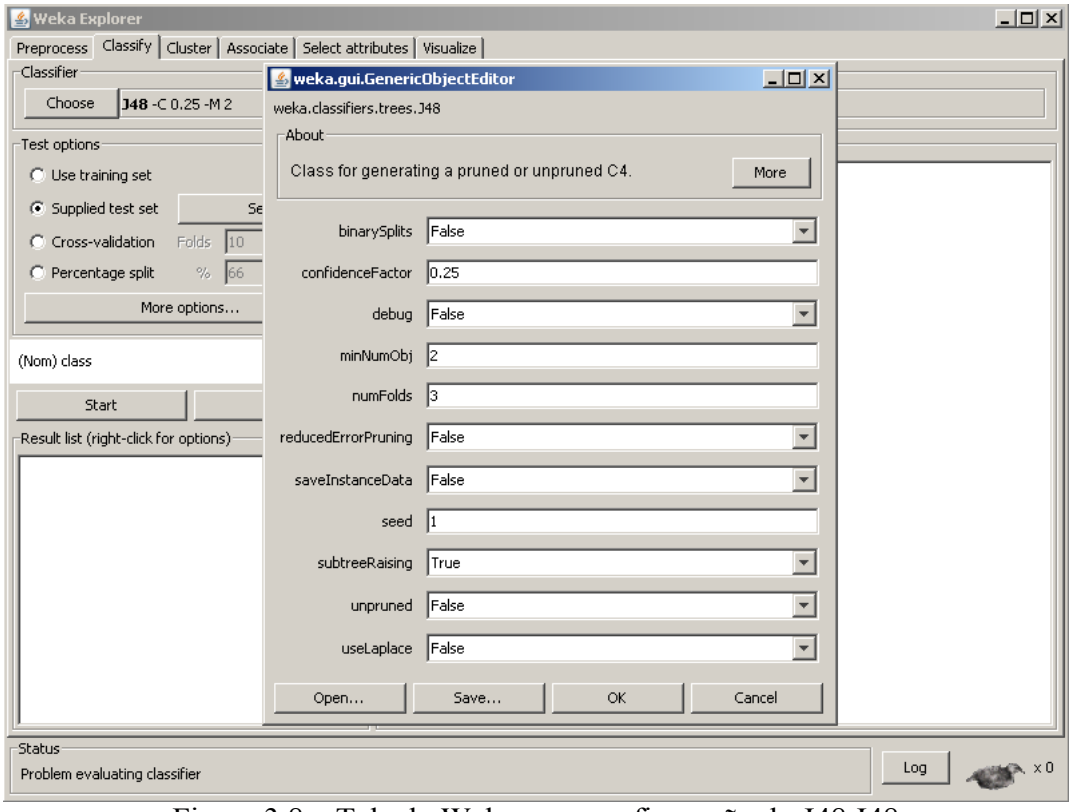

Figura 3.8 – Tela do Weka para configuração do J48.J48
Fonte: WEKA, 2008

Goldschmidt (2005) define árvore de decisão como um padrão de conhecimento onde cada nó interno da árvore representa uma decisão sobre um atributo que determina como os dados são particionados pelos seus nós filhos.

> [...] "Inicialmente, a raiz da árvore contém toda a base de dados com exemplos misturados das várias classes. Um predicado, denominado ponto de separação, é escolhido como sendo a condição que melhor separa ou discrimina as classes. Tal predicado envolve exatamente um dos atributos do problema e divide a base de dados em dois ou mais conjuntos, que são associados cada um a um nó filho. Cada novo nó abrange, portanto, uma partição da base de dados que é recursivamente separada até que cada conjunto associado a cada nó folha consista inteiramente ou predominantemente de registros de uma mesma classe." (GOLDSCHMIDT, 2005).

Elmasri (2005) acrescenta que o nó para particionar as amostras utiliza o atributo com o melhor critério para separação, ou ainda, aquele que maximiza a medida de ganho de informação. O autor ainda menciona, que o uso de entropia como alcance de ganho de informação é motivado, objetivando minimizar nas partições resultantes, os dados de amostra. Conseqüentemente, irá minimizar o número de testes condicionais para classificar um novo registro.

O cálculo da entropia é obtido através da fórmula representada na figura 3.9:

$$E(A = v_j) = -\sum_{i=1}^{n} p(i) \times \log_2(p(i))$$

Figura 3.9 – Função da entropia
Fonte: Adaptado ELMASRI, 2005

Onde:

A = v_j – significa que o atributo A tem o valor v_j

n – é o número de classes diferentes c1,c2,...,cn

p(i) – é a probabilidade de um registro pertencer à classe cn

3.3.2 Algoritmo J48. PART

O algoritmo J48.PART, segundo Oliveira (2002), é uma variação do J48.J48, que a partir da árvore de decisão, constrói regras de produção e Mongiovi (1998) acrescenta, que a meta de um gerador de regras, é gerar um conjunto mínimo de regras (quantidade e comprimento). Segundo Gonchorosk (2005), para a criação da lista de decisão, o algoritmo utiliza uma árvore já estruturada e realiza a indução de regras, que após, as regras vão sendo comprovadas ou alteradas.

O J48.PART possui uma abordagem de "dividir para conquistar" pelo fato de que, a cada iteração uma árvore de decisão é criada de forma parcial, transformando a melhor folha em uma regra e dessa forma, obter maior ganho de informação (tradução nossa) (Weka, 2008).

O método de geração de regras de produção ocorre em dois estágios: primeiramente as regras são induzidas de uma árvore e o segundo estágio, as regras são refinadas. Para cada regra criada é estimada a sua cobertura, através de quanto representam em relação aos demais registros da base. Isso ocorre repetidamente até que todas as instâncias estejam cobertas a fim de refinar a regra. As regras com coberturas mais altas, sempre em relação à quantidade de registros da base, são apresentadas para o usuário e as demais são descartadas (OLIVEIRA, 2002).

Da mesma forma que J48.J48, possui parâmetros configuráveis pela ferramenta Weka, o J48.PART também tem parâmetros que podem ser modificados, porém em uma quantidade menor conforme mostra figura 3.10 da tela do Weka.

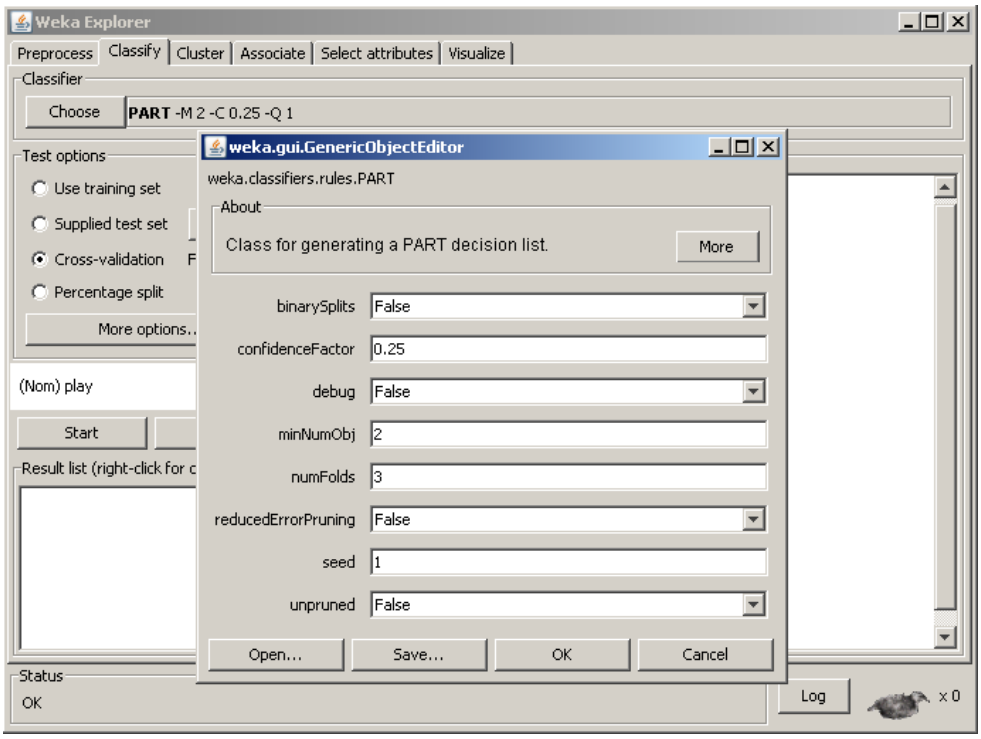

Figura 3.10 – Tela do Weka para configuração do J48.PART
Fonte: WEKA, 2008

A ferramenta Weka ainda dispõe de métodos para aumentar o desempenho, que podem ser utilizados juntamente com os algoritmos J48.J48 e o J48.PART, destacando-se os métodos: *Bagging* e *Boosting*, que serão visto na próxima seção.

3.3.3 Métodos *Bagging* e *Boosting*

De acordo com Oliveira (2002), para a construção de conjuntos de classificadores, o Weka dispõe os métodos de meta aprendizagem. Essas classes são consideradas como aditivos para melhorar os resultados, sendo possível associar esses métodos aos algoritmos de aprendizagem (WEKA, 2008).

O método *Bagging* caracteriza-se por: a partir de conjuntos de amostras de dados contínuos e independentes, gerar os classificadores. As amostras são construídas a partir de um conjunto de dados de treinamento. De forma aleatória, *m* instâncias são extraídas com substituição a partir do conjunto original, ou seja, caso ocorra repetição de instâncias nas amostras. Por default o Weka sugere 10 interações como mostra a figura 3.11 (OLIVEIRA, 2002) (WEKA, 2008).

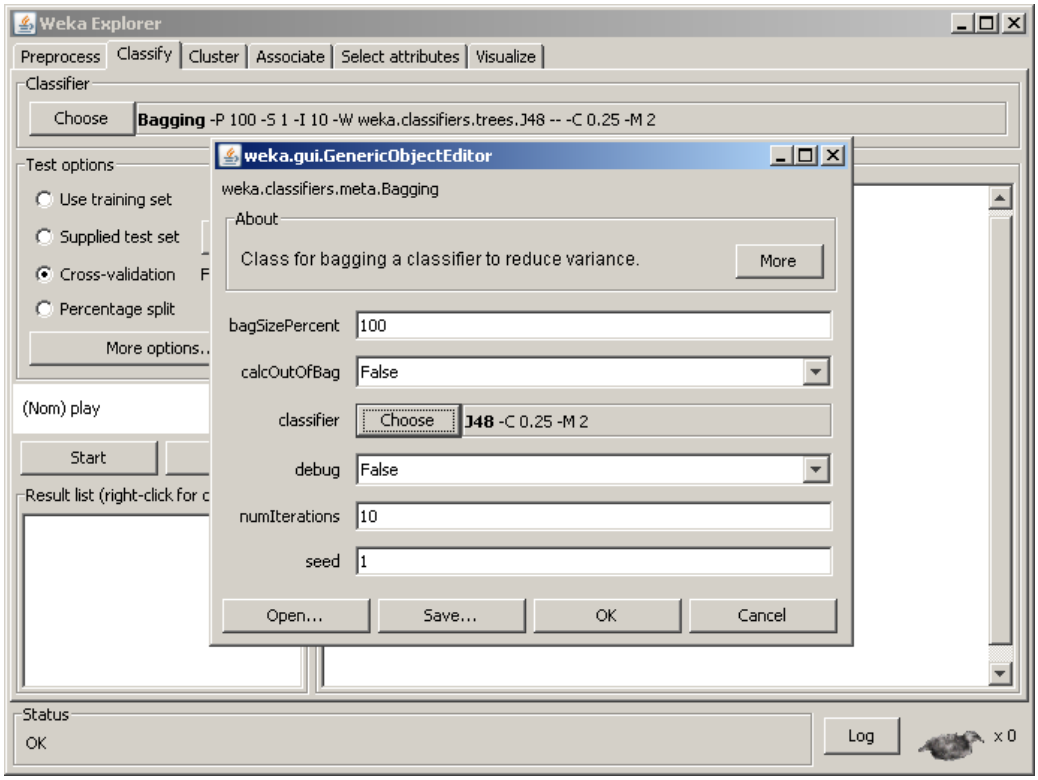

Figura 3.11 – Tela do Weka para configuração do *Bagging*
Fonte: WEKA, 2008

Já o método *Boosting*, caracteriza-se por: cada instância de treinamento recebe um peso associado a ela, onde ao ser induzido o primeiro classificador, todas as instâncias receberão o mesmo peso. Após a primeira indução ser feita, as instâncias de treinamento a qual foram atribuídas um peso incorretamente, receberão novos pesos baseado nos classificadores que anteriormente foram gerados (OLIVEIRA, 2002).

Nos dois métodos de meta aprendizagem, o algoritmo de aprendizagem que será utilizado para criar os classificadores é definido pelo usuário (WEKA, 2008). Outros métodos estão disponíveis no Weka, porém para este estudo, foram destacados os métodos de *Bagging* e *Boosting*.

3.4 Algoritmo de *Clustering* (Agrupamento)

Os algoritmos de agrupamento ou *clustering* tem uma característica interessante que o diferencia dos algoritmos de classificação, onde o agrupamento dos registros é realizado de acordo com características semelhantes apresentadas entre os dados. Elmasri (2005) define este processo como "aprendizado não supervisionado" onde o objetivo é colocar os registros em grupos, de forma que os registros de um grupo sejam semelhantes entre si e diferentes dos componentes dos demais grupos.

A ferramenta Weka trabalha com alguns algoritmos de c*lustering*, podem ser citados: EM, Cobweb, X-means, FarthestFirst e SimpleKMeans. Sendo este último o que será utilizado para realização desse estudo.

Segundo Goldschmidt (2005) o algoritmo seleciona aleatoriamente *k* pontos de dados numéricos, que serão definidos como centróides (elementos centrais) dos *clusters*. Logo após, cada registro da base é atribuído ao *cluster* em que a distância deste ponto (registro) em relação ao centróide do *cluster*, seja a menor entre todas as distâncias calculadas. A cada iteração, um novo centróide é atribuído para cada *cluster*, através da média dos pontos do *cluster*, o que irá caracterizar a configuração do *cluster* para a próxima iteração. Esse processo será concluído, quando os centróides dos *clusters* não se modificarem mais ou ainda quando atingir o número máximo de iterações anteriormente parametrizado pelo usuário. O fluxo do funcionamento do algoritmo está exemplificado na figura 3.12.

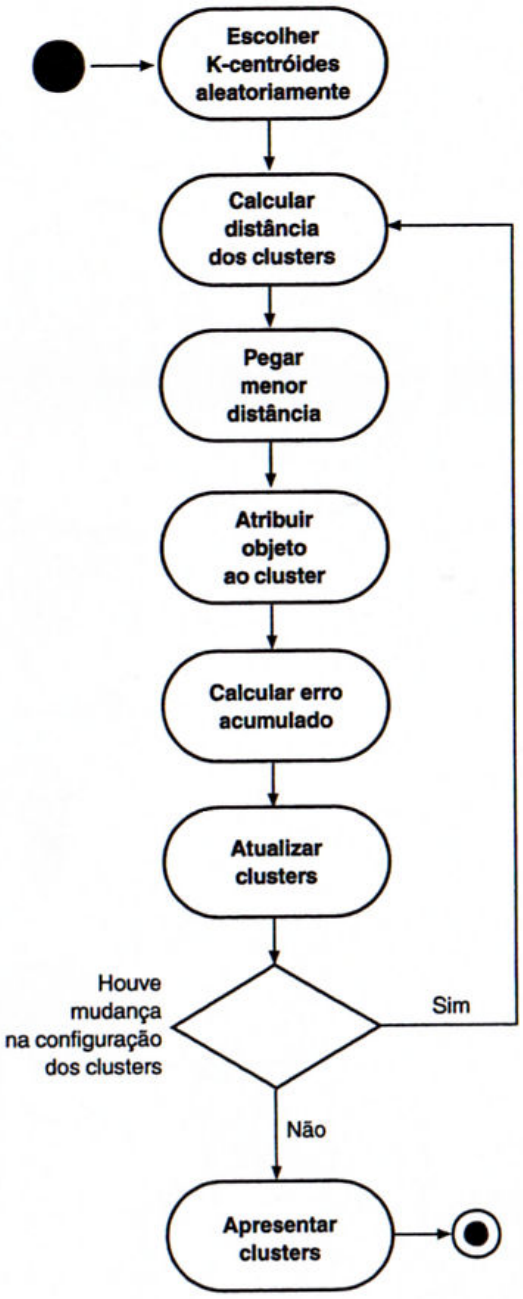

Figura 3.12 – O fluxo do funcionamento do algoritmo KMeans
Fonte: Adaptado GOLDSCHMIDT, 2005

O autor ainda salienta que a partir de um parâmetro *k*, um conjunto de *n* objetos é divido em *k clusters* onde a similaridade intracluster seja alta, mas que a similaridade intercluster, seja baixa. A similaridade de um *cluster* é medida em relação à média de valores dos objetos que compõem o *cluster*, definido como: "centro de gravidade do *cluster*". As figuras 3.13 e 3.14 ilustram a aplicação do algoritmo em um arquivo com 20 registros de dados considerando o parâmetro $k = 3$.

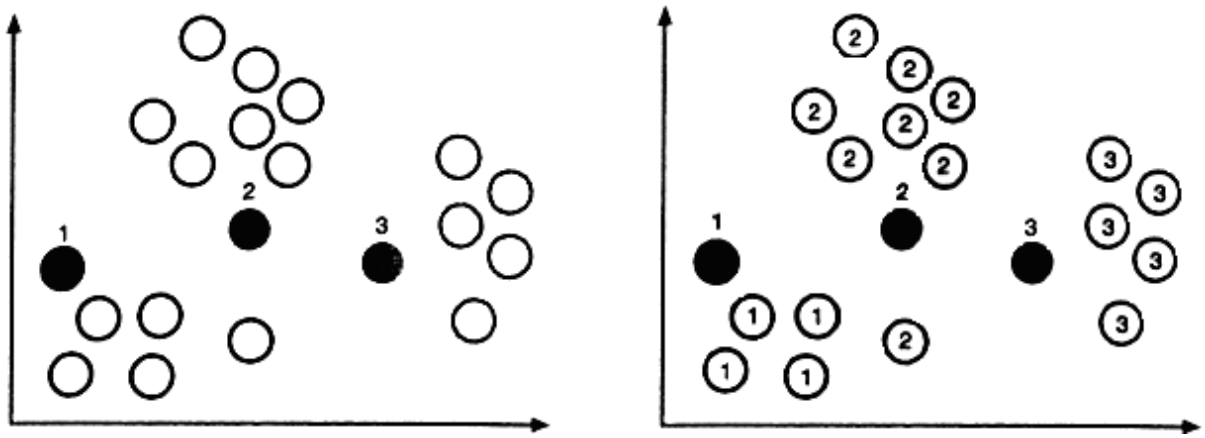

Figura 3.13 – Primeiros passos do algoritmo KMeans
Fonte: Adaptado GOLDSCHMIDT, 2005

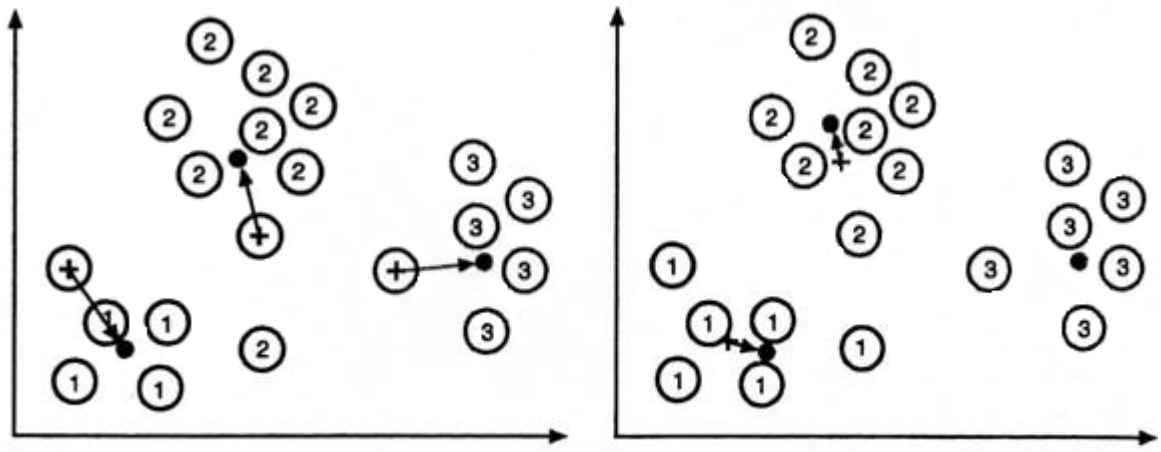

Figura 3.14 – Passos subseqüentes do algoritmo KMeans
Fonte: Adaptado GOLDSCHMIDT, 2005

No próximo capítulo, será detalhada a origem dos dados que serão utilizados para esse estudo, assim como sua modelagem e proposta de tratamento.

4 ESTUDO DE CASO

De acordo com Gonchorosk (2005), um estudo de caso é uma técnica empregada, que tem por objetivo realizar algum tipo de estudo e análise envolvendo uma pesquisa, buscando conhecimento detalhado sobre algo real. Considerando esse conceito, será feito um estudo sobre a aplicação de mineração de dados no setor de varejo calçadista, buscando encontrar informações que permitam desenvolver campanhas de *marketing* de forma eficiente e eficaz.

O estudo será feito utilizando os dados de uma grande empresa do setor de varejo calçadista. Atualmente, a empresa não se beneficia da utilização de nenhuma ferramenta de *data mining* para extrair conhecimento e padrões desconhecidos. Toda a informação armazenada na base de dados hoje, é obtida através das ferramentas convencionais de inferência de dados, como: utilização de consultas SQL sendo executas em forma de relatórios pré-definidos ou ainda por consultas específicas solicitadas a área de TI. Na maioria das vezes, os gestores buscam conhecimento válido e útil, através de grandes planilhas geradas a partir das consultas executadas na base.

Nas próximas seções, será detalhada a forma como os dados estão modelados e como serão tratados para realização desse estudo.

4.1 Modelagem dos dados

Para não interferir no ambiente de produção e nem no ambiente de desenvolvimento, foi definido para esse estudo um universo de tabelas que possuem dados sobre clientes, seus volumes financeiros de compra e suas preferências por tipos de produtos. A base utilizada

para aplicação do *data mining* possui dados de notas fiscais e produtos das notas no período de 02/01/2006 a 04/04/2008.

Um ambiente isolado foi montado. Ele consiste de um SGBD Oracle 10g instalado e configurado, para onde foram importadas as tabelas e dados pré-selecionados. A ferramenta Weka também está instalada nesse mesmo ambiente. Na figura 4.1, é apresentado o Modelo Entidade Relacionamento (MER) das tabelas que serão utilizadas para o desenvolvimento desse estudo.

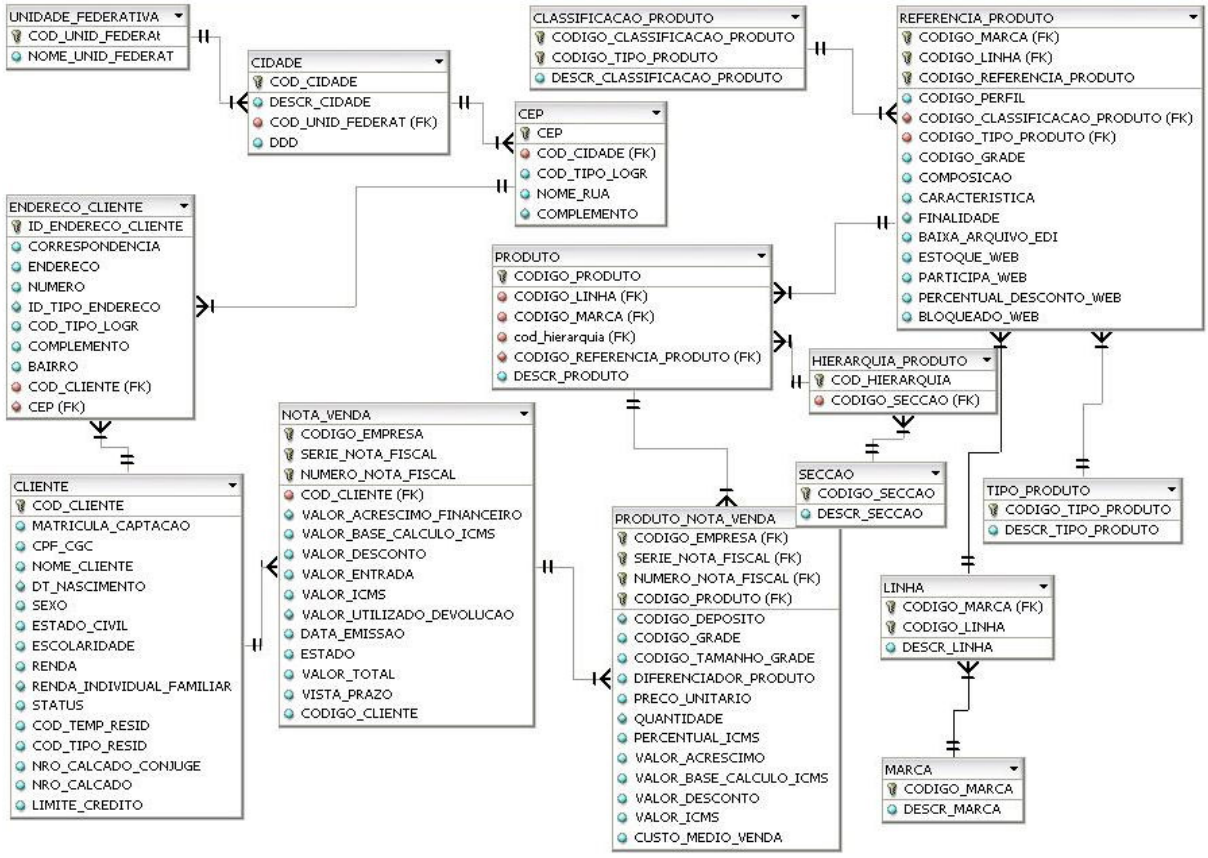

Figura 4.1 – Modelo ER dos dados de vendas, clientes e produtos

Na figura 4.1 é possível visualizar as tabelas pré-selecionadas que estão basicamente divididas em três grandes grupos que são: NOTA_VENDA e PRODUTO_NOTA_VENDA que possuem dados sobre as vendas e os produtos envolvidos nelas, CLIENTE, ENDERECO_CLIENTE, CEP, CIDADE e UNIDADE_FEDERATIVA onde constam os atributos de clientes e seus respectivos detalhamentos e por último os dados sobre produtos, que estão armazenados nas tabelas PRODUTO, REFERENCIA_PRODUTO,

HIERARQUIA_PRODUTO, TIPO_PRODUTO, LINHA, MARCA, SECCAO E CLASSIFICACAO_PRODUTO.

4.2 Seleção dos dados

Os dados do sistema de origem estão armazenados em uma base relacional. Para aplicar as técnicas de *data mining*, poderia se optar por criar uma tabela única "não normalizada" a partir da junção de todas as tabelas envolvidas ou construir um *data mart*. Optou-se por criar a tabela não normalizada por considerar este processo mais simples e que atenderia de forma satisfatória as necessidades para a mineração de dados.

Uma consulta SQL foi gerada, onde cada tupla conterá toda informação necessária para o processamento. Goldschmidt (2005) denomina esse método como "junção orientada", onde os atributos e registros foram selecionados prevendo um potencial para influir na mineração. Caracterizando assim, a etapa de seleção de dados.

Nessa *query* um tratamento foi feito para trazer os produtos das notas como colunas do mesmo registro da nota fiscal. Esse processo foi necessário para conseguir tabular os itens presentes em uma mesma nota e avaliar qual a relação existente entre eles. Foi definido um número máximo de 10 produtos, pois em um levantamento prévio, identificou-se que 9 foi o número máximo de itens encontrados em uma mesma nota. Nesse momento, foi utilizado o caracter coringa para preencher as colunas das notas fiscais que possuíam um número de produtos menor que 10, como mostra a figura 4.2.

SEXO	ESTADO_CIVIL	PRODUTO1	PRODUTO2	PRODUTO3	PRODUTO4	PRODUTO5	PRODUTO6	...
FEMININO	CASADO	CHINELO	SANDALIA	OUTROS	OUTROS	?	?	...
FEMININO	CASADO	MEIA	MEIA	MEIA	MEIA	SAPATO FEM	?	...
FEMININO	CASADO	TENIS	OUTROS	OUTROS	TENIS	?	?	...
FEMININO	CASADO	ACESSORIO ESPORTE	BOTA	MEIA	BOTA	?	?	...
MASCULINO	SOLTEIRO	CHINELO	CHINELO	BOLSA	CHINELO	?	?	...
MASCULINO	SOLTEIRO	TENIS	TENIS	MEIA	MEIA	MEIA	?	...
FEMININO	SOLTEIRO	TENIS	CHINELO	TENIS	OUTROS	?	?	...
MASCULINO	SOLTEIRO	MEIA	MEIA	MEIA	BOTA	?	?	...
FEMININO	CASADO	MEIA	MEIA	MEIA	MEIA	TENIS	?	...
MASCULINO	CASADO	OUTROS	OUTROS	OUTROS	OUTROS	?	?	...
MASCULINO	CASADO	MEIA	MEIA	MEIA	BOLSA	?	?	...
MASCULINO	CASADO	SAPATO MASC	BOTA	MEIA	MEIA	MEIA	?	...
MASCULINO	CASADO	MEIA	MEIA	MEIA	TENIS	?	?	...
MASCULINO	SOLTEIRO	MEIA	MEIA	MEIA	BOTA	?	?	...
FEMININO	SOLTEIRO	SANDALIA	SANDALIA	SAPATO FEM	TENIS	?	?	...

Figura 4.2 – Exemplo de tabela única com dados de clientes, produtos e caracter coringa

Durante o período selecionado, foi acumulado um total de 6.233.443 de transações de vendas, sendo que desse total foram selecionadas as notas de vendas que possuíssem um número maior de dois itens com categorias distintas entre eles. Com esse filtro, obteve-se um

total de 219.888 transações. Segundo Goldschmidt (2005), esse procedimento é chamado de "segmentação do banco de dados".

Outro fator que precisa ser considerado é a limitação de memória e demais recursos computacionais utilizados para esse estudo, que não permitiriam trabalhar com um conjunto maior de dados. O processo de *data mining* exige uma grande disponibilidade de memória RAM e também processadores de alta capacidade.

4.3 Pré-processamento

Após a etapa de seleção dos dados, iniciou-se o pré-processamento. De acordo com Schneider (2007), a eliminação de ruídos é feita nesse momento, preparando a base para a etapa de mineração de dados. Goldschmidt (2005) salienta que a qualidade dos modelos de conhecimento a serem abstraídos sofre influência da qualidade dos dados, justificando essa etapa do processo. Tendo como base a tabela "não normalizada" gerada na etapa da seleção de dados, identificaram-se os registros que possuíam colunas com informações nulas para que os mesmos fossem excluídos, visando com isso à eliminação de ruídos e limpeza da base.

Devido à base de dados atual possuir uma estrutura de cadastro de produtos complexa, identificou-se a necessidade de efetuar uma redução de valores nominais, conforme denominado por Goldschmidt (2005), no atributo referente a categoria do produto. O autor ainda cita que essa operação consiste em reduzir a quantidade de valores distintos em determinados atributos.

> Identificação de hierarquias entre valores [...] Por exemplo, considere um conjunto de dados que contenha informações sobre os produtos vendidos a cada cliente: tênis, sapato, sandália, bermuda, calça, camisa, paletó. Hierarquias podem ser definidas pelo especialista para indicar generalizações de conceito em relação aos valores existentes: tênis $\subset$ calçados, sapato $\subset$ calçados, sandália $\subset$ calçados, bermuda $\subset$ roupa, calça $\subset$ roupa, camisa $\subset$ roupa e paletó $\subset$ roupa. As três primeiras generalizações indicam que tênis, sapato e sandália são casos particulares de calçado. Analogamente, as quatro últimas generalizações indicam que bermuda, calça, camisa e paletó são casos particulares do conceito roupa. A partir desta especificação, os valores originais podem ser substituídos pelas respectivas generalizações, reduzindo um domínio de 7 valores distintos, para apenas 2. (GOLDSCHMIDT, 2005).

Redução ou projeção também podem ser definidas como: "encontrar características úteis para representar os dados, dependendo do objetivo da tarefa. Com redução da dimensionalidade ou métodos de transformação, o número efetivo de variáveis em estudo

pode ser reduzido ou representações invariantes para os dados podem ser encontrados." (tradução nossa) (Fayyad, 1996, p. 42). Para valores nominais, é aceitável apresentar possíveis generalizações para os valores de cada atributo.

Algumas estruturas complexas referente ao produto podem ser citadas como: hierarquia, tipo, classificação, linha e seção do produto, sendo elas necessárias para utilização e organização do setor de compras da empresa. Com isso identificou-se um número muito diversificado de categorias. Para esse estudo, foi reduzida a quantidade de valores distintos de 248, para apenas 20. Alguns exemplos dessa generalização são apresentados na figura 4.3.

DESCR_TIPO_PRODUTO	DESCR_TIPO_PRODUTO_DM
BOTA SOCIAL	BOTA
BOTA	BOTA
CHUTEIRA	CHUTEIRA
CHUTEIRA INDOOR	CHUTEIRA
CLOG	SANDALIA
SANDALIA CLOG	SANDALIA
SPTO BB FEM	SAPATO FEM
SPTO JUV FEM	SAPATO FEM
SPTO BB MASC	SAPATO MASC
SPTO JUV MASC	SAPATO MASC

Figura 4.3 – Categorias de produtos reduzidas

Os dados são formatados para que todos estejam com a fonte maiúscula e não tenham espaços em brancos entre palavras. Por exemplo, se o dado original é "Sapato Masc", o dado passa a ser formatado para "SAPATO_MASC", para padronizar o formato e facilitar a etapa de transformação que será executada após o pré-processamento.

A etapa de transformação é onde os dados de entrada serão recebidos do pré-processamento. Como essa etapa trata os dados de forma específica para aplicação de cada algoritmo, será abordado nos próximos capítulos de forma detalhada com aplicação para os algoritmos objetos desse estudo.

4.4 Algoritmos a serem estudados

Os algoritmos que serão utilizados para esse estudo serão os de associação e de classificação e também a utilização dos algoritmos de meta aprendizagem *Bagging* e *Boosting*. Objetiva-se identificar padrões de compras dos clientes, referente à relação existente entre diferentes tipos de produtos que normalmente ocorrem na mesma transação.

Através desse tipo de extração de conhecimento, poderão ser aperfeiçoadas as campanhas de *marketing*, além de permitir efetuar o reabastecimento (compra) dos produtos em proporções semelhantes e ainda efetuar a aproximação dos mesmos dentro da área de venda da loja.

Nós próximos capítulos será apresentado o funcionamento dos algoritmos bem como a execução e resultados obtidos quando aplicados à base de dados que foi preparada.

5 ALGORITMO DE ASSOCIAÇÃO

Como o objetivo do trabalho é identificar a associação existente entre os produtos que fazem parte de uma mesma nota fiscal de venda, pesquisou-se o funcionamento do algoritmo Apriori utilizado pela ferramenta Weka. Alguns testes iniciais foram feitos com intuído de buscar um maior conhecimento prático da ferramenta juntamente com as particularidades dos parâmetros utilizados pelo Apriori, para posteriormente efetuar a análise na base de dados real de uma grande empresa de varejo calçadista. Este capítulo descreve a transformação dos dados necessária para aplicação do algoritmo de associação, os parâmetros utilizados pelo Weka e também os resultados obtidos como resultado da execução.

5.1 Transformação dos Dados

A etapa de transformação tem a finalidade de modificar e formatar os dados para geração do arquivo arff conforme já apresentado anteriormente. Em um primeiro momento, ocorreram dificuldades em relação à forma de estruturar os dados dentro do arquivo *arff*, pois a construção da ferramenta exige que os atributos sejam informados em sua totalidade. Ou seja, considerando que o arquivo possui doze atributos, é necessário que na sessão @data contenha doze colunas de dados separados por vírgula.

Em um cenário real, não serão todas as notas de vendas que terão o mesmo número de registro sendo necessária a utilização de um recurso denominado como o caracter coringa apresentado por Frank (2008) em seus exemplos. O caracter coringa é o sinal de interrogação (“?”) e serve para indicar ao algoritmo a falta de informação para aquela coluna do registro. A utilização desse coringa é pouco citada na literatura e arquivos disponíveis na web.

Após os dados terem sido recebidos da etapa de pré-processamento, foi executada uma query na tabela não normalizada gerada anteriormente. Os dados foram formatados para gerar o arquivo .arff com toda sua estrutura e no formato exigido para realizar a mineração utilizando o algoritmo de associação Apriori conforme mostra a figura 5.1.

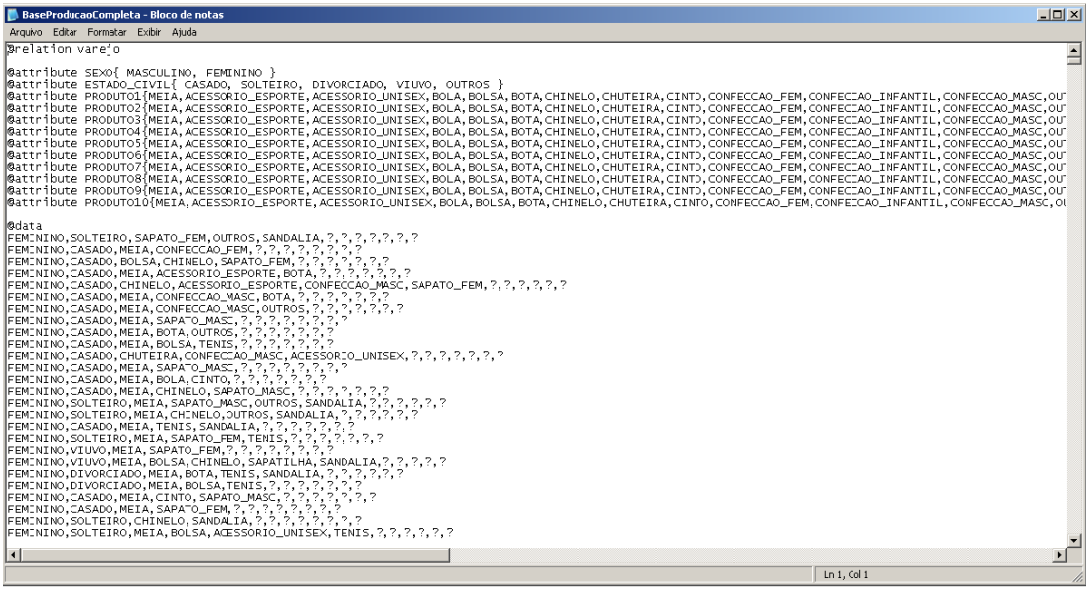

Figura 5.1 – Arquivo Arff gerado na etapa de transformação para o Apriori

Na próxima seção, será detalhado o funcionamento e parâmetros utilizados pelo Weka na execução do algoritmo de associação Apriori e também os testes iniciais feitos para buscar um maior conhecimento na forma de apresentação dos resultados.

5.2 Parametrizações e Testes de Funcionamento

Antes de utilizar a ferramenta e o algoritmo para fazer a mineração dos dados, foi necessário conhecer melhor o funcionamento do algoritmo de associação do Weka, bem como, identificar as parametrizações existentes e seu impacto no funcionamento. Para realização dos testes de funcionamento, foi utilizada a base de dados de exemplo citado anteriormente nesse trabalho no quadro 3.2. A figura 5.2 apresenta o arquivo arff que foi gerado manualmente para validar os resultados.

```
@relation ExemploTCI

@attribute produto1{BOTA,MEIA,BOLSA,SAPATO,CINTO}
@attribute produto2{BOTA,MEIA,BOLSA,SAPATO,CINTO}
@attribute produto3{BOTA,MEIA,BOLSA,SAPATO,CINTO}
@attribute produto4{BOTA,MEIA,BOLSA,SAPATO,CINTO}
@attribute produto5{BOTA,MEIA,BOLSA,SAPATO,CINTO}

@data
BOTA,?,BOLSA,SAPATO,?
?,MEIA,BOLSA,?,CINTO
BOTA,MEIA,BOLSA,?,CINTO
?,MEIA,?,?,CINTO
```

Figura 5.2 – Arquivo Arff baseado no exemplo base de dados

Na figura 5.3 pode-se observar os parâmetros do algoritmo Apriori no Weka.

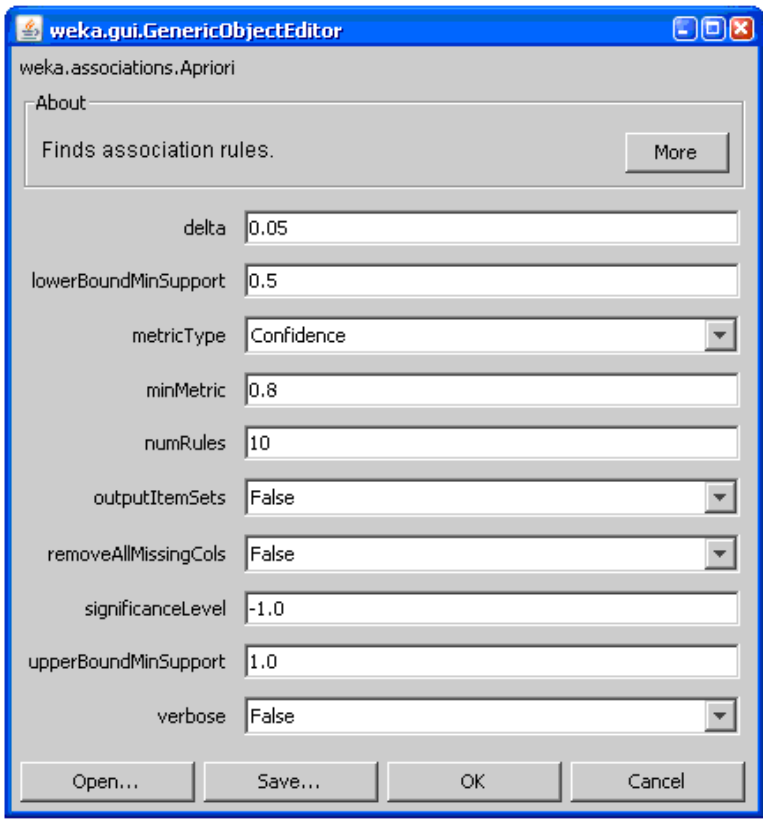

Figura 5.3 – Parâmetros do algoritmo Apriori

Os parâmetros do algoritmo Apriori do Weka (figura 5.3) são (WEKA, 2008):

- **delta –** iterativamente diminui o suporte por esse fator. Reduz o suporte até que o mínimo seja atingido ou o número de regras exigidas tenha sido gerado;
- **lowerBoundMinSupport –** limite inferior para o suporte mínimo;
- **metricType –** define o tipo de métrica pela qual as regras serão classificadas. Que podem ser, confiança, *lift*, alavancagem e convicção;

- **minMetric** – pontuação mínima da métrica escolhida. Considerar apenas regras com pontuações superiores a este valor. No exemplo da figura 5.3, considerou-se a métrica de confiança com pontuação mínima de 0.8;
- **numRules** – número desejado de regras à encontrar;
- **outputItemSets** – se ativado, os elementos que compõem os *itemsets* serão também apresentados na área de *output*;
- **removeAllMissingCols –** remove as colunas com valores em falta;
- **significanceLevel** – o nível de significância. Significância teste (apenas utilizado quando a métrica de confiança for selecionada);
- **upperBoundMinSupport** – limite superior para suporte mínimo. Inicia iterativamente de forma decrescente o apoio mínimo a partir deste valor;
- **verbose** – se ativado o algoritmo será executado em modo verboso/detalhado (demasiadamente grande em palavras que pouco exprimem);

Submeteu-se então o arquivo arff apresentado na figura 5.2 para validar os resultados gerados a partir da base de dados exemplo, que contém quatro transações. Atribuiu-se na tela de parâmetros do algoritmo um suporte mínimo de 0.5 e confiança mínima de 0.8. Na figura 5.4 é apresentado o resultado da execução do Apriori.

Nota-se que foram gerados 3 grandes conjuntos de dados (*itemsets*), o L1 com 4 elementos, L2 também com 4 e L3 com apenas 1. Logo abaixo são apresentadas as regras encontradas que satisfizeram os parâmetros informados.

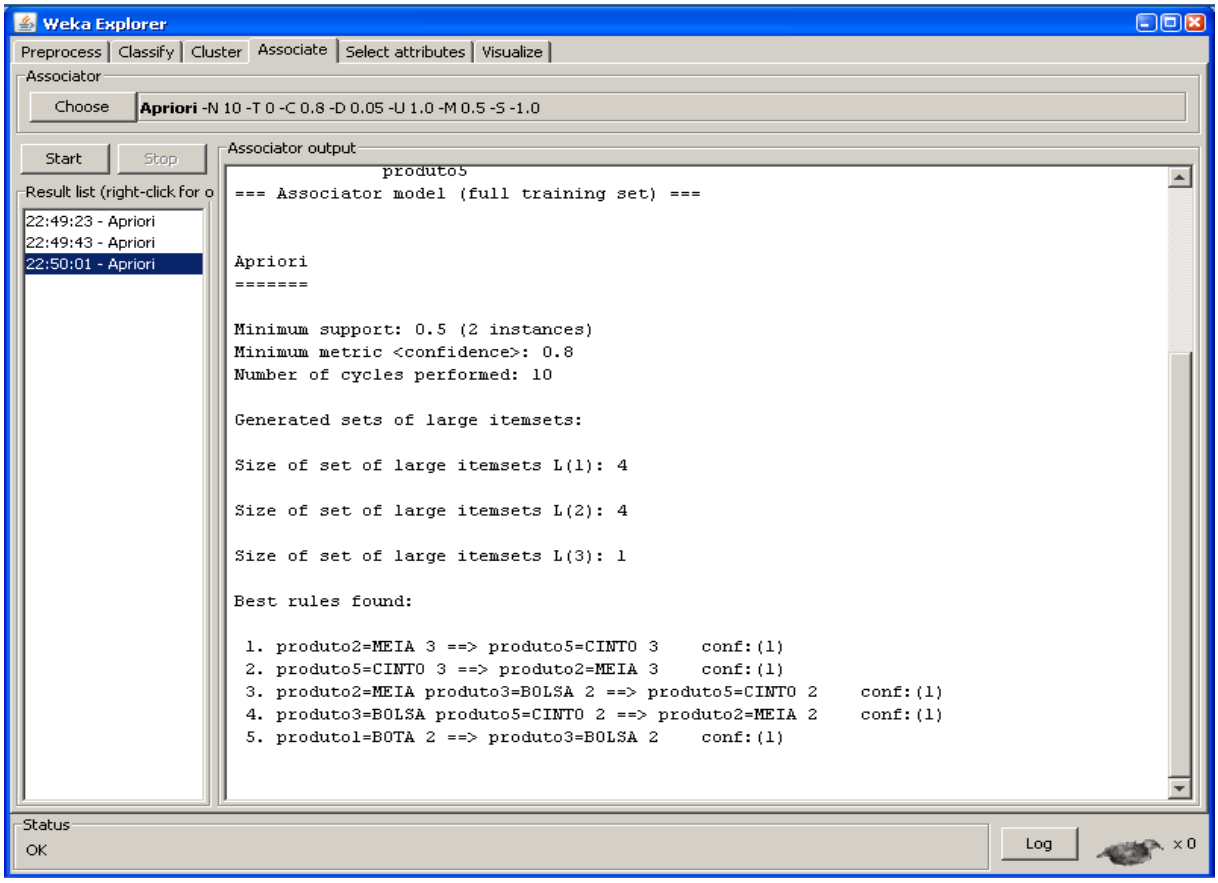

Figura 5.4 – Saída da execução do Apriori

Segundo Santos (2005), é possível efetuar outra formatação para o arff a ser utilizado pelo algoritmo de associação Apriori. O formato considera SIM para identificar que houve a venda de determinado produto e NAO para situação em que não houve venda. Essa é uma opção que não utiliza o caracter coringa “?”. Nesse exemplo foram consideradas as mesmas quatro transações do exemplo anterior. Na figura 5.5 é apresentada a nova estrutura do arquivo arff.

```
@relation ExemploTCI

@attribute BOTA{SIM,NAO}
@attribute MEIA{SIM,NAO}
@attribute BOLSA{SIM,NAO}
@attribute SAPATO{SIM,NAO}
@attribute CINTO{SIM,NAO}

@data
SIM,NAO,SIM,SIM,NAO
NAO,SIM,SIM,NAO,SIM
SIM,SIM,SIM,NAO,SIM
NAO,SIM,NAO,NAO,SIM
```

Figura 5.5 – Nova formatação do arquivo Arff

A figura 5.6 mostra o resultado gerado pela execução do Apriori utilizando o novo arquivo arff. Para esse exemplo foram mantidos os mesmos parâmetros para suporte mínimo de 0.5 e confiança mínima de 0.8. Notou-se que um número maior de *itemsets* foi gerado e ainda uma quantidade muito maior de regras foi gerada, pois para os produtos que não houve venda e estão com atributo igual a NAO, também foram geradas regras para essa situação.

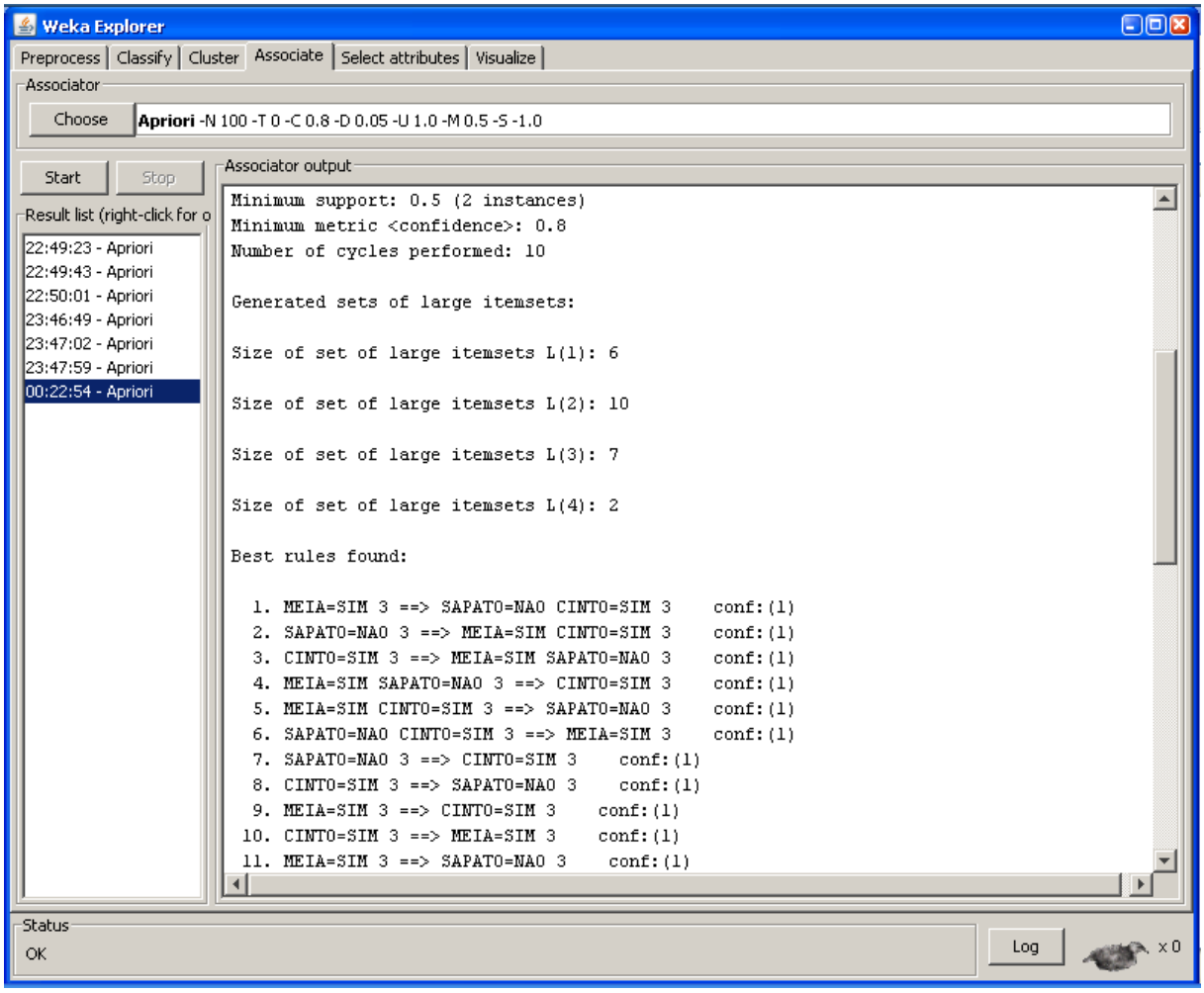

Figura 5.6 – Nova saída da execução do Apriori

Constatou-se com esse exemplo, que os mesmos resultados foram obtidos, porém com um grau maior de dificuldade no momento de encontrar as regras válidas e utilizáveis. A figura 5.7 é referente a tela de saída do arquivo de retorno do exemplo, onde mostra as mesmas regras que foram encontradas de forma simplificada com a utilização do caracter coringa "?". Em todas as demais regras geradas, encontrou-se a associação com o atributo NAO.

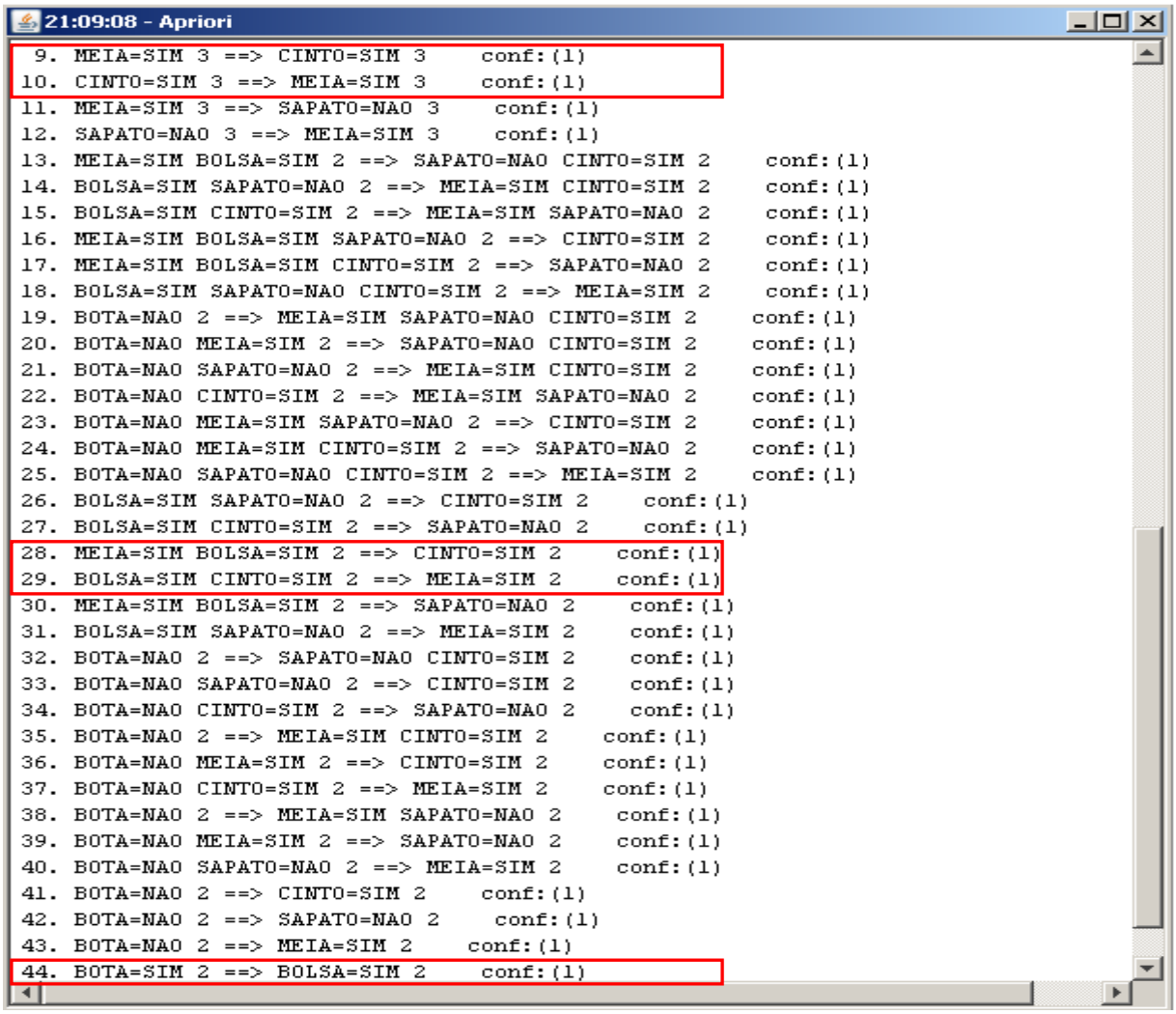

Figura 5.7 – Novas regras geradas na execução do Apriori

Esses testes foram feitos, para validar através do Weka o exemplo citado na seção 3.2 desse trabalho, onde exemplificou-se de forma detalhada o funcionamento do algoritmo Apriori. Após a verificação dos resultados obtidos nessa etapa, pode-se comprovar e ratificar o conceito anteriormente estudado. Dessa forma gerou-se uma maior segurança na validação dos resultados que serão apresentados na próxima seção referente à mineração de dados.

Ainda na próxima seção, será apresentado a forma como o arquivo gerado .arff, foi submetido a ferramenta Weka para processamento do algoritmo Apriori.

5.3 Mineração do Dados

Após efetuar a preparação dos dados e geração do arquivo arff, o arquivo foi submetido a mineração através do algoritmo de associação Apriori implementado pela ferramenta Weka, utilizando a interface *explorer* (citada anteriormente na seção 3.1), por permitir uma fácil utilização dos recursos disponíveis. Após selecionar o arquivo .arff anteriormente gerado, são apresentadas algumas informações sobre os atributos contidos no arquivo. Uma representação gráfica também é gerada com objetivo de visualizar um resumo sobre cada atributo e também informações sobre a quantidade que os mesmos se repetem conforme mostra a figura 5.8.

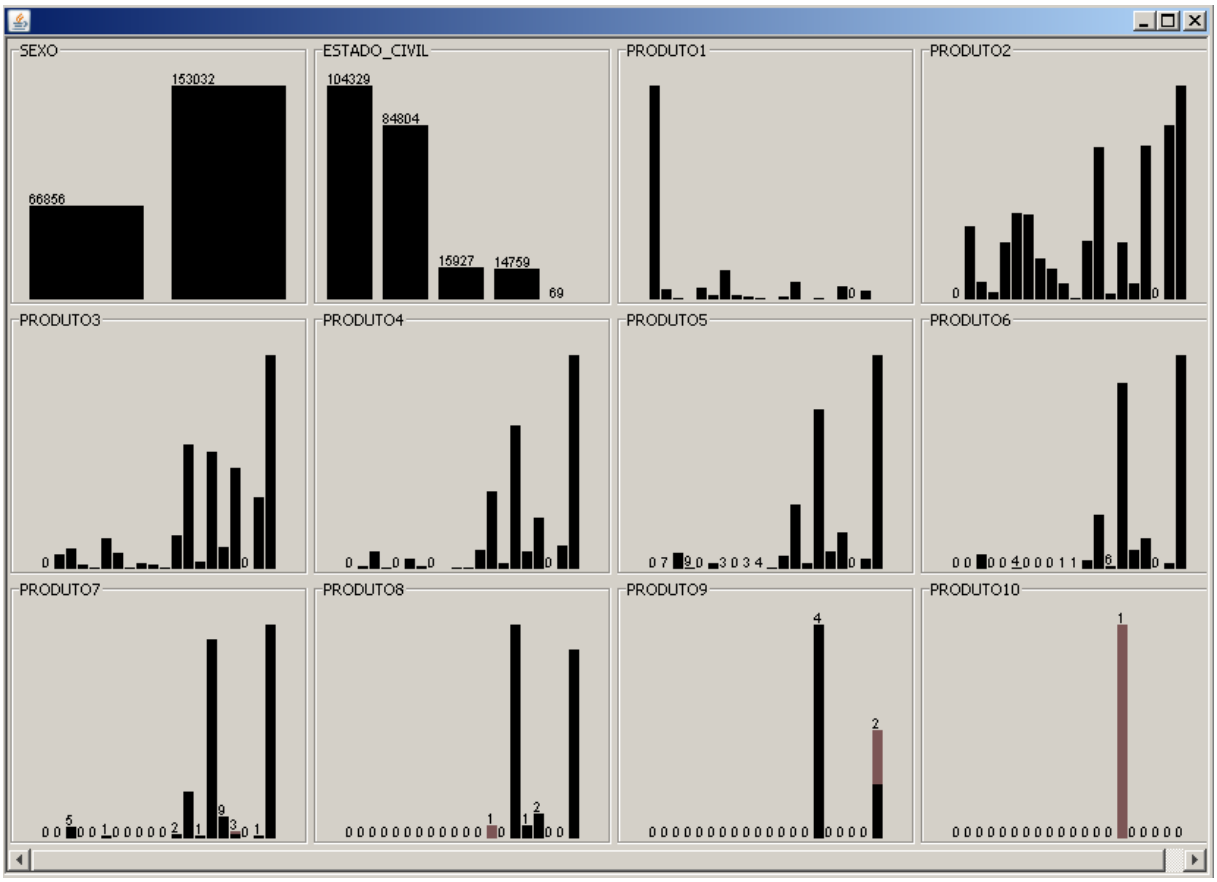

Figura 5.8 – Representação gráfica de resumo sobre cada atributo

Primeiramente o arquivo .arff foi submetido a execução do algoritmo Apriori utilizando os parâmetros *default* conforme sugerido pela ferramenta. O parâmetro de confiança mínima inicialmente possui valor de 0.9 que representa um fator de confiança maior ou igual a 90%. Com essa configuração o algoritmo encontrou três grandes *itemsets*

porém não encontrou nenhuma regra que possuísse os fatores mínimos de confiança inicialmente selecionado.

A partir desse resultado, foi alterado o fator mínimo de confiança para 0.5 e também o parâmetro referente ao número de regras gerado foi aumentado de 10 para 20. Os outros parâmetros foram mantidos e após isso foi iniciado o processamento novamente. O resultado, como mostra a figura 5.9, também gerou três grandes *itemsets* e um total de 17 regras que atenderam o novo fator mínimo de confiança passado como parâmetro.

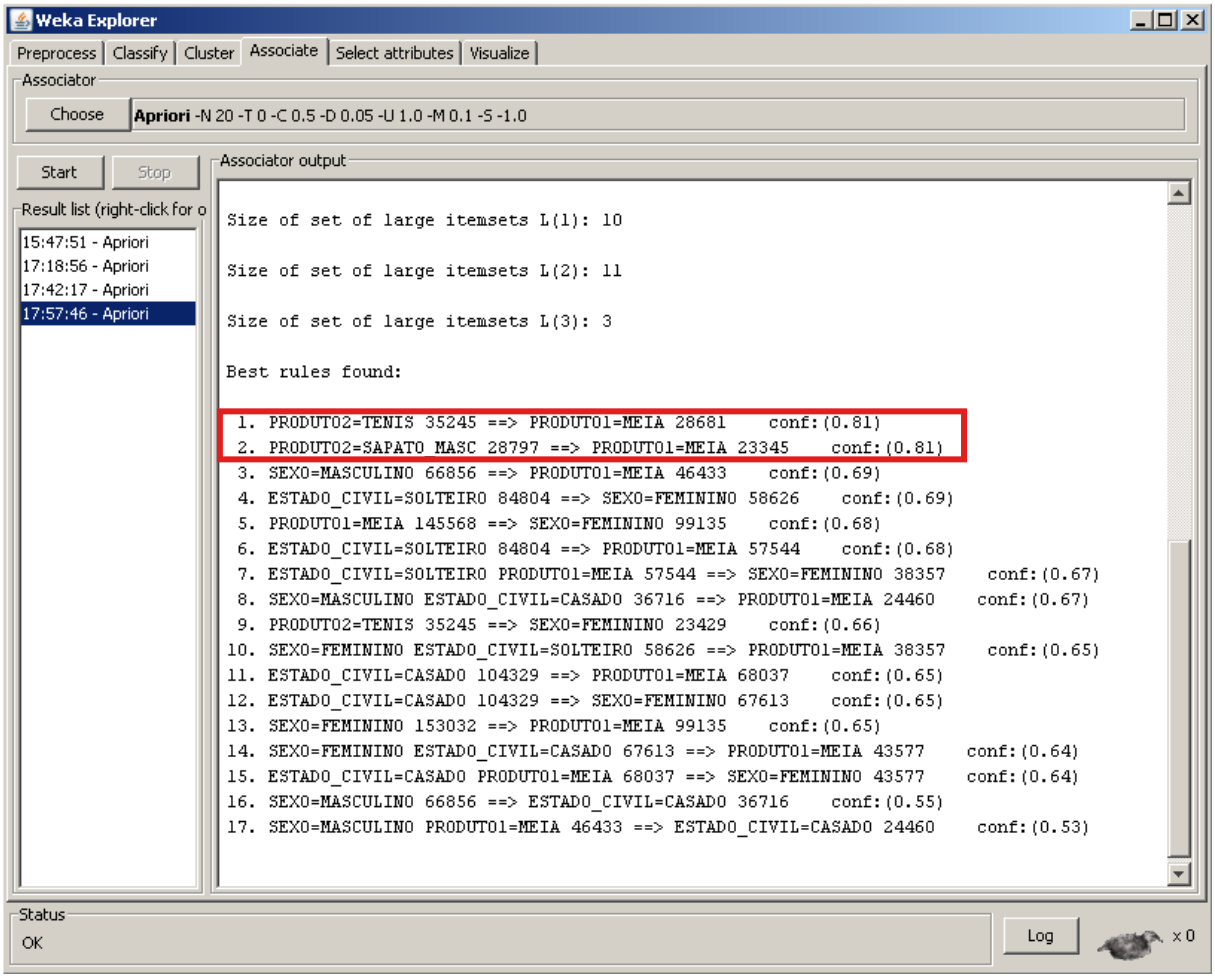

Figura 5.9 – Resultado da nova execução do Apriori – Fator confiança alterado

Nessa etapa do processo, os resultados foram apresentados e analisados na tentativa de encontrar a associação entre os produtos que compõem as transações de vendas. Tendo como base a figura 5.9 é possível perceber que as regras que tiveram um percentual maior de confiança, foram as regras 1 e 2 com fator 0.81. A regra (1) indica que a compra de tênis pode levar a compra de meia e a regra (2), que a compra de sapato masculino, pode também levar a compra de meia. Considerando que para utilizar um sapato ou ainda um tênis é necessária a

utilização de um par de meias, percebeu-se que os resultados apresentados foram de certa forma muito óbvios. Outro fator que gera essa distorção é o baixo valor da meia quando comparado com outros produtos, o que propícia um maior volume de vendas desse produto.

Analisando os dados estatísticos apresentados pela ferramenta Weka conforme mostra a figura 5.10, nota-se que o produto "meia" encontra-se em um grande número de transações e em uma quantidade alta se comparada com os outros produtos.

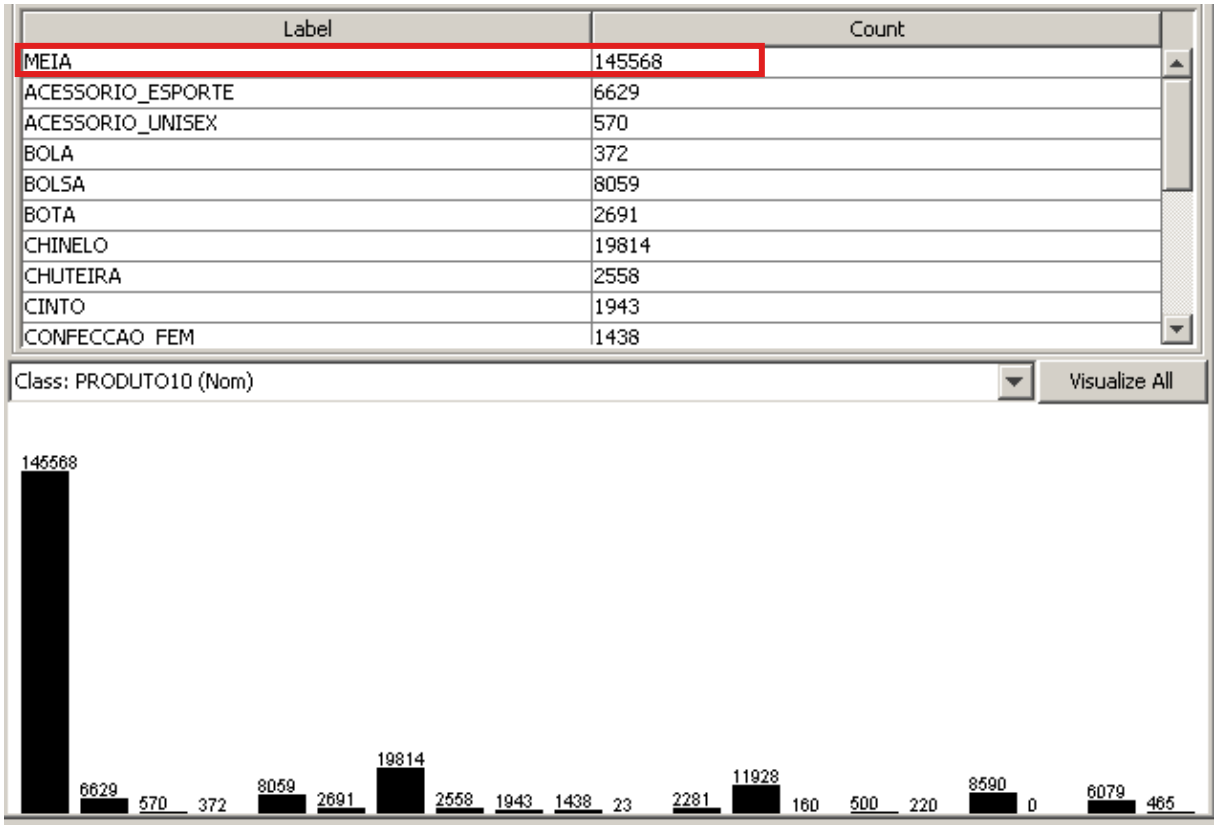

Figura 5.10 – Gráfico estatístico da quantidade de produtos

Como foi citado anteriormente, o processo de descoberta de novos conhecimentos é interativo e iterativo e composto por etapas seqüenciais (FAYYAD, 1996). Conforme citado por Goldschmidt (2005) interativo por indicar a necessidade de envolvimento do homem controlando o processo e iterativo por sugerir a possibilidade de repetições total ou parcial do processo buscando conhecimento satisfatório por meio de refinamentos sucessivos.

Com base nesses conceitos, optou-se por gerar a consulta novamente com base na tabela não normalizada, tirando o produto meia, com intuído de identificar a associação existente entre os outros produtos. Nessa nova consulta foi feita uma decodificação na seleção dos dados, para que em cada tupla retornada, o produto meia fosse substituído pelo caracter coringa "?". A partir dessa nova consulta, um novo arquivo .arff foi gerado.

Os parâmetros do algoritmo foram mantidos os mesmos que foram utilizados para a validação anterior. O fator mínimo de confiança ficou em 0.5 e o parâmetro referente ao número de regras geradas em 20. O resultado, como mostra a figura 5.11, gerou dois grandes *itemsets* e um total de quatro regras que atenderam o fator de suporte e confiança passado como parâmetro.

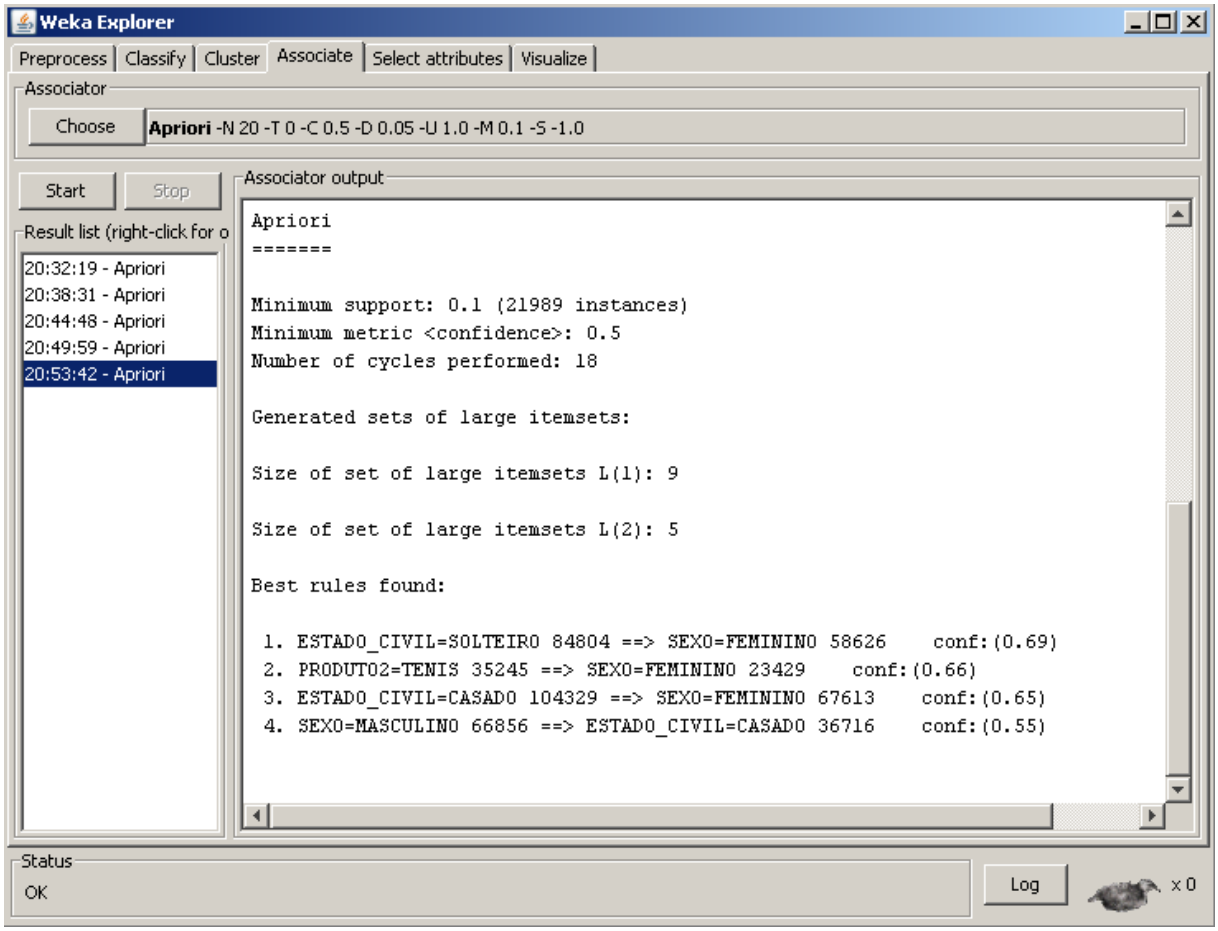

Figura 5.11 – Resultado da nova execução do Apriori sem produto meia

Tendo como base a figura 5.11 é possível perceber que mesmo sem o produto meia, não se obteve nenhum novo conhecimento, não gerando um resultado satisfatório. Nenhuma associação entre os produtos foi apresentada.

Em busca da obtenção de novos conhecimentos, no próximo capítulo, o mesmo arquivo arff será minerado pelo algoritmo classificador J48 também implementado pela ferramenta Weka.

6 ALGORITMO DE CLASSIFICAÇÃO

Uma das propostas iniciais desse trabalho é identificar o comportamento de alguns dos algoritmos implementados pelo Weka, buscando identificar qual dos algoritmos analisados possui uma maior aderência a base de dados de uma empresa do setor de varejo calçadista. Como descrito no capítulo anterior, o Apriori não apresentou um resultado satisfatório. Nesse capítulo será descrita a análise e resultados apresentados pelo algoritmo de classificação J48. Os métodos de meta aprendizagem *Bagging* e *Boosting* disponíveis no Weka, também serão aplicados e descritos nesse capítulo.

6.1 Transformação dos Dados

Na etapa de transformação dos dados para gerar o arquivo arff a ser submetido ao processamento do algoritmo classificador J48, algumas mudanças foram feitas em relação ao arquivo gerado para o algoritmo de associação Apriori.

A tabela não normalizada recebeu a adição de 20 novas colunas com os nomes de cada categoria de produto existente na base. Essa operação é denominada por Goldschmidt (2005), como construção de atributos. O autor ainda justifica sua utilização, pela redução do conjunto de dados tornando o processamento do algoritmo mais simplificado, além de agregar ao problema informações que sejam úteis ao processo de mineração.

Após as colunas serem adicionadas, foi executado um *script* em PL/SQL que a partir das colunas produto1 até produto10 identificava qual categoria de produto estava armazenado e atribuía o valor “SIM” a coluna com o nome da respectiva categoria onde se caracterizava uma venda do produto referido. As colunas foram inicializadas com valor “NAO”, sendo alterado para “SIM” quando uma categoria era identificada no registro. Optou-se por essa

transformação, por ter gerado um modelo mais simples de avaliar os resultados durante a aplicação de teste inicial.

Conforme mostra a figura 6.1, após o término da execução do *script*, os dados foram selecionados considerando as novas colunas adicionadas a tabela contendo a informação "SIM" representando uma venda daquele produto e "NÃO" caso o produto não tenha sido adquirido na mesma nota de compra.

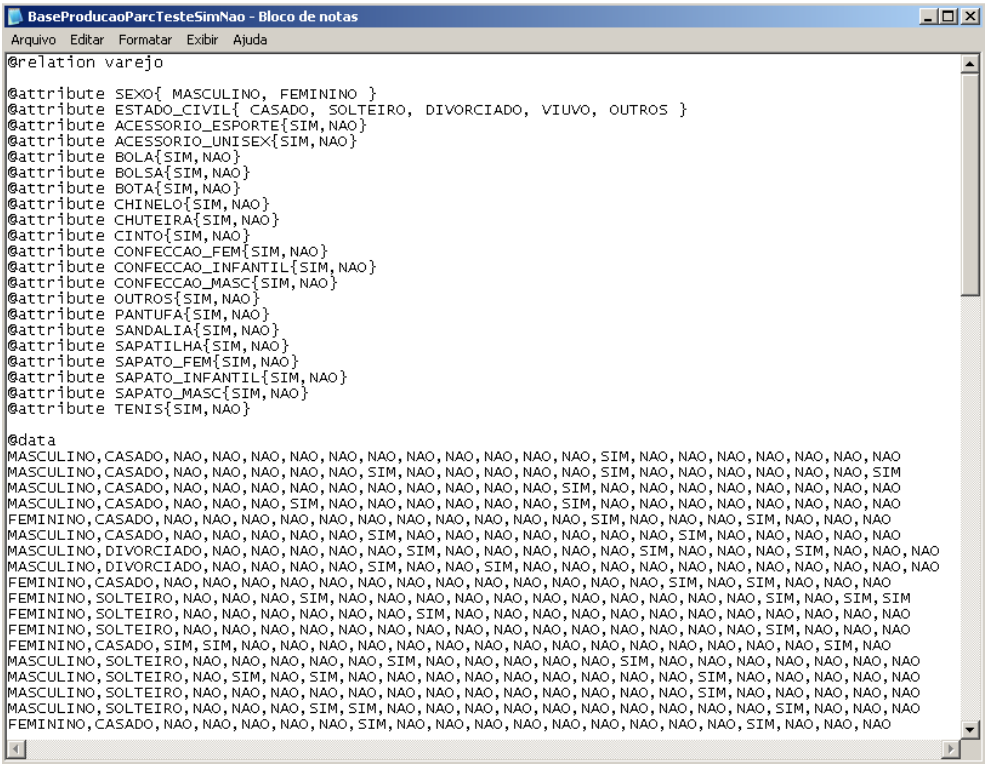

Figura 6.1 – Arquivo Arff gerado na etapa de transformação para o J48

Testes iniciais foram feitos com objetivo de conhecer as particularidades e parâmetros utilizados pelo J48 e pelos métodos de meta aprendizagem, para posteriormente efetuar a análise na base de dados real de uma grande empresa de varejo calçadista.

6.2 Parametrizações e Testes de Funcionamento

Para realização dos testes iniciais, foi utilizada uma quantidade reduzida de 100 registros que foram selecionados de forma aleatória da base real que será analisada posteriormente. O primeiro teste foi realizado com os parâmetros *default* do Weka, apresentados na figura 6.2.

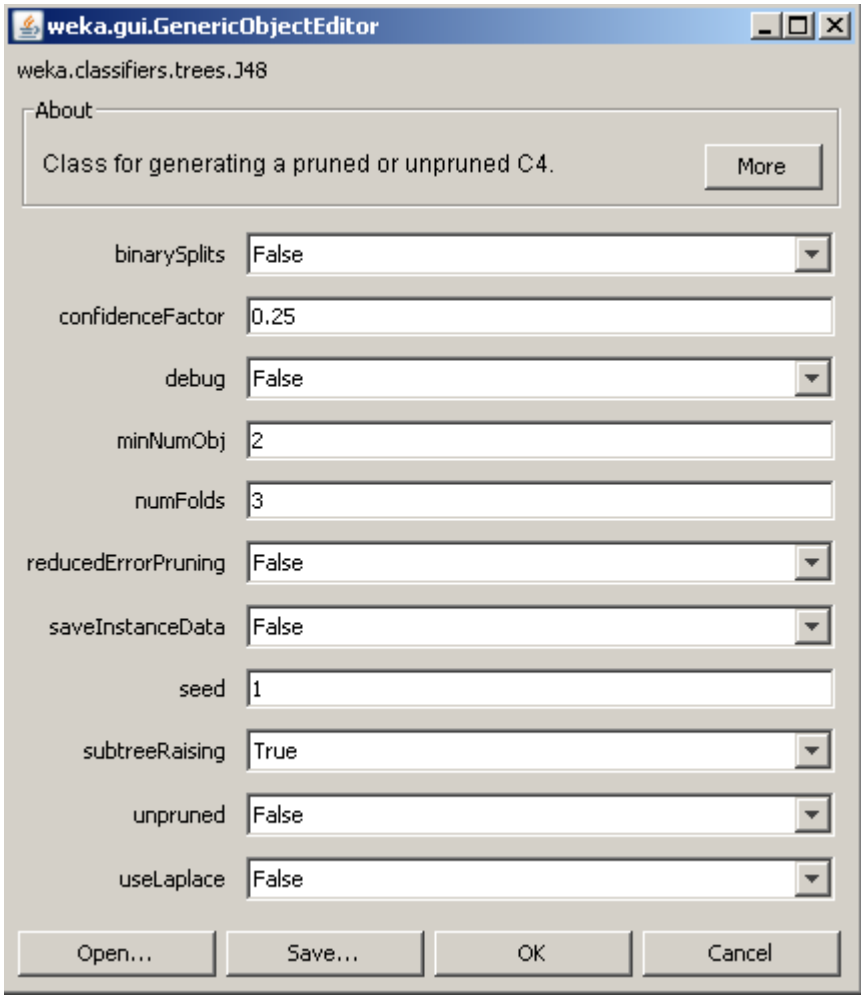

Figura 6.2 – Parâmetros do algoritmo J48

Os parâmetros do algoritmo J48 do Weka (figura 6.2) são (WEKA, 2008):

- **binarySplits** – relevância para usar divisão binária sobre atributos nominais para construir as árvores;
- **confidenceFactor** – o fator confiança utilizado para a poda (valores menores, indicam maior poda);
- **debug** – essa opção quando alterada para verdadeiro (*true*), apresenta informações adicionais sobre a execução do algoritmo;
- **numMinObj** – serve para informar o número mínimo de instâncias por folha;

- **numFolds** – determina a quantidade de dados utilizados para a redução de erro na poda. Uma dobra (ou f*old*) é utilizada para a poda e o restante para o cultivo da árvore;
- **reducedErrorPruning** – serve para ativar ou não a redução de erros na poda;
- **saveInstanceData** – determina se os dados de treinamento devem ser salvos para posterior visualização;
- **seed** – parâmetro que informa o número de sementes que serão selecionadas aleatoriamente quando for utilizado o parâmetro de redução de erros na poda;
- **subTreeRaising** – informa se uma operação para substituir o nó interno da árvore por um dos nós que estão abaixo deve ser utilizada. Esta é uma ação utilizada no momento da poda;
- **unpruned** – não utilizar poda;
- **useLaplace** – conta o número de folhas excluídas suavizadas pela base de Laplace, "que é um método simples para transformar um Problema com Valores Iniciais (PVI), em uma equação algébrica, de modo a obter uma solução deste PVI de uma forma indireta, sem o cálculo de integrais e derivadas para obter a solução geral da Equação Diferencial" (SODRÉ, 2003, p. 1, apud GONCHOROSK, 2005);

O algoritmo J48 utiliza um processo de aprendizado supervisionado conforme definido por Elmasri (2005), onde o primeiro modelo é gerado a partir de um conjunto de treinamento que foi anteriormente classificado.

Além dos parâmetros apresentados na figura 6.2, a ferramenta permite definir algumas configurações sobre o conjunto de testes que será utilizado. Como mostra a figura 6.3, é possível escolher um conjunto de treinamento ou ainda ativar a validação cruzada e o percentual de separação, que serve para determinar o tamanho do conjunto de dados que serão separados para a geração de novos classificadores (1) (WEKA, 2008).

Conforme citado por Goldschmidt (2005), caso o conjunto de treinamento não seja suficientemente representativo, o classificador pode ter bom desempenho apenas no conjunto de treinamento, mas não no conjunto de testes. Esse fenômeno é definido pelo autor como *overfitting* onde o classificador ajustou-se em excesso ao conjunto de treinamento. O autor ainda fala do fenômeno de *underfitting* que é a situação oposta, onde o classificador ajusta-se pouco ao conjunto de treinamento. Esse fenômeno tende a ocorrer em função de

parametrizações impróprias do algoritmo de aprendizagem ou ainda um volume grande de variáveis.

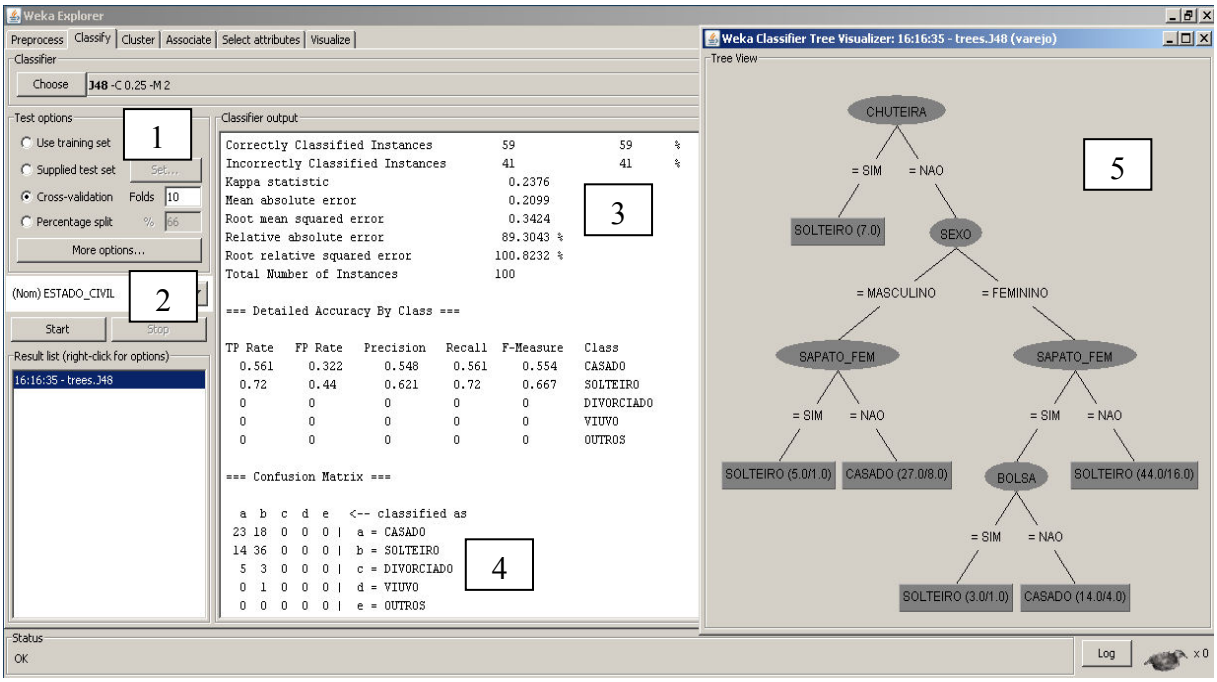

Figura 6.3 – Parâmetros e resultados dos testes iniciais

A figura 6.3 também mostra o resultado gerado com base no arff contendo o número reduzido de registros. Para o teste inicial, foi utilizado o atributo alvo ESTADO_CIVIL para que a árvore fosse gerada a partir dele (2). Na tela de resultados são apresentadas algumas informações sobre a quantidade e percentual de instâncias que foram classificadas corretamente, as que foram incorretamente classificadas e ainda informações sobre os erros gerados durante a classificação (3). No final da página de resultado, uma matriz de confusão é apresentada (4). Segundo Goldschmidt (2005), essas matrizes servem para fornecer um detalhamento do desempenho e qualidade do modelo de classificação. O weka apresenta também uma representação gráfica da árvore de classificação que foi gerada através da indução dos dados ao algoritmo classificador (5).

Com os parâmetros *default* do weka para o J48, verificou-se que 59% das instâncias foram corretamente classificadas. A opção *binarySplits* que serve para ativar a divisão binária quando atributos nominais são utilizados, foi alterada para *true*, mas os resultados se mantiveram os mesmos.

Quando o parâmetro para redução de erros na poda (*reducedErrorPruning*) foi alterado para *true* o número de instâncias corretamente classificadas baixou para 48%. Como

visto anteriormente, o parâmetro *numFolds* está diretamente relacionado com o *reducedErrorPruning*, então optou-se em aumentá-lo de 3 para 4 para aumentar o número de folhas utilizadas na redução de erros o que passou para 53% a quantidade de instâncias classificadas corretamente.

Outros parâmetros como o *seed*, que informa o número de sementes que serão selecionadas no momento da redução de erros na poda também foram alterados, mas o melhor resultado foi obtido com os parâmetros *default* do Weka conforme mostra a figura 6.4.

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------ 1

CHUTEIRA = SIM: SOLTEIRO (7.0)
CHUTEIRA = NAO
|   SEXO = MASCULINO
|   |   SAPATO_FEM = SIM: SOLTEIRO (5.0/1.0)                    2
|   |   SAPATO_FEM = NAO: CASADO (27.0/8.0)
|   SEXO = FEMININO
|   |   SAPATO_FEM = SIM                                        3
|   |   |   BOLSA = SIM: SOLTEIRO (3.0/1.0)
|   |   |   BOLSA = NAO: CASADO (14.0/4.0)
|   |   SAPATO_FEM = NAO: SOLTEIRO (44.0/16.0)

Number of Leaves  :     6

Size of the tree :      11


Time taken to build model: 0.02 seconds

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

Correctly Classified Instances          59               59      %
Incorrectly Classified Instances        41               41      %
=== Confusion Matrix ===

  a  b  c  d  e   <-- classified as
 23 18  0  0  0 |  a = CASADO
 14 36  0  0  0 |  b = SOLTEIRO
  5  3  0  0  0 |  c = DIVORCIADO
  0  1  0  0  0 |  d = VIUVO
  0  0  0  0  0 |  e = OUTROS
```

Figura 6.4 – Classificação obtida nos testes iniciais

A figura 6.4, no item marcado como (1), também apresenta a árvore de classificação gerada a partir do algoritmo classificador J48. Interpretando o resultado apresentado na figura, é possível identificar que clientes do sexo masculino e solteiros, compram calçados femininos, porém não costumam comprar chuteiras, com 20% de acerto (2). Outro grupo de cliente apresentado no resultado é do sexo feminino, onde aparece uma relação entre a compra

de sapato feminino e bolsa para as clientes com estado civil igual a solteiro, com 33,33% de acerto e as casadas não costumam comprar os dois produtos juntamente na mesma nota de compra, com 28,58% de acerto (3).

O arquivo a ser submetido à mineração foi gerado com um total de 219.888 instâncias ou registros referentes às notas de vendas armazenadas. Na próxima seção, será detalhada a forma como os dados foram submetidos à mineração, utilizando o algoritmo de classificação J48 e também a interpretação dos resultados apresentados.

6.3 Mineração do Dados

Após efetuar a preparação dos dados e geração do arquivo arff, o arquivo foi submetido à mineração através do algoritmo classificador J48. Da mesma forma que o algoritmo de associação Apriori, utilizou-se a interface *explorer* para execução do algoritmo. Como na fase inicial de teste os parâmetros *default* apresentaram o maior percentual de classificação, optou-se por submeter o arquivo ao J48 com os parâmetros sugeridos pela ferramenta. Nas opções do conjunto de treinamento, foi selecionado o item *cross-validation* com o valor inicial de 10 *folds* sugerido pela ferramenta. O atributo ESTADO_CIVIL foi selecionado para ser o alvo sendo utilizado como base para geração da árvore.

O processamento do algoritmo utilizando 10 *folds* para o *cross-validation* levou 9 minutos e 29 segundos desde o inicio até o fim da execução conforme apresentado na figura 6.5 item (1). Seguindo o exemplo de Gonchorosky (2005), o número de *folds* foi alterado para 2, pois é o mínimo que a ferramenta permite e os resultados se mantiveram praticamente os mesmos, com uma pequena variação no percentual de instâncias corretamente classificadas, porém com uma redução considerável no tempo de execução, levando um total de 1 minuto e 36 segundos (2).

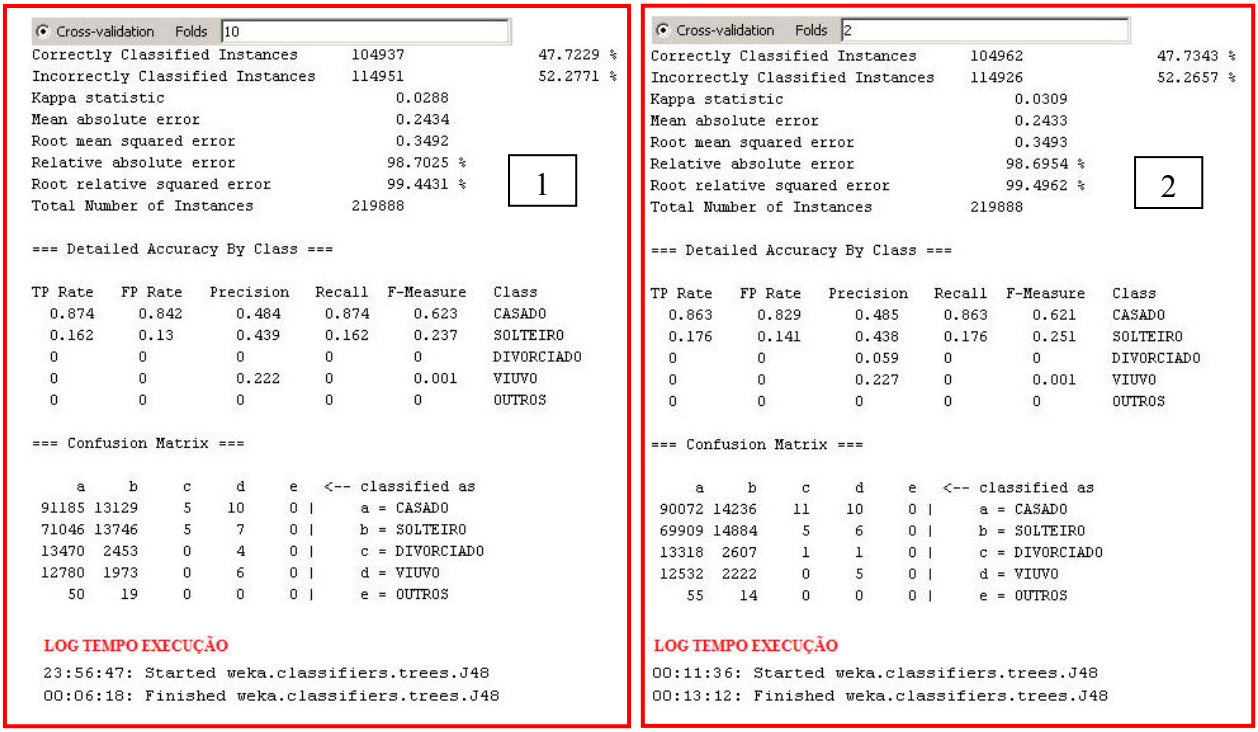

Figura 6.5 – Resultados estatísticos da execução do J48 atributo alvo estado civil

Com base nos resultados semelhantes apresentados na execução do J48 com valor de 2 ou 10 atribuídos ao parâmetro *folds*, optou-se por efetuar as análises informando apenas 2 *folds,* por executar o processamento em um tempo menor. A árvore foi gerada inicialmente com tamanho igual a 325 com 163 folhas como apresenta a figura 6.6 no item marcado como (1).

Analisar a árvore gerada e identificar um novo conhecimento potencialmente utilizável tornou-se uma tarefa árdua, em decorrência à grande extensão tanto em sua altura, quanto em sua largura. Para facilitar essa análise e identificação das regras relevantes, foi necessário separar alguns pontos considerados mais interessantes.

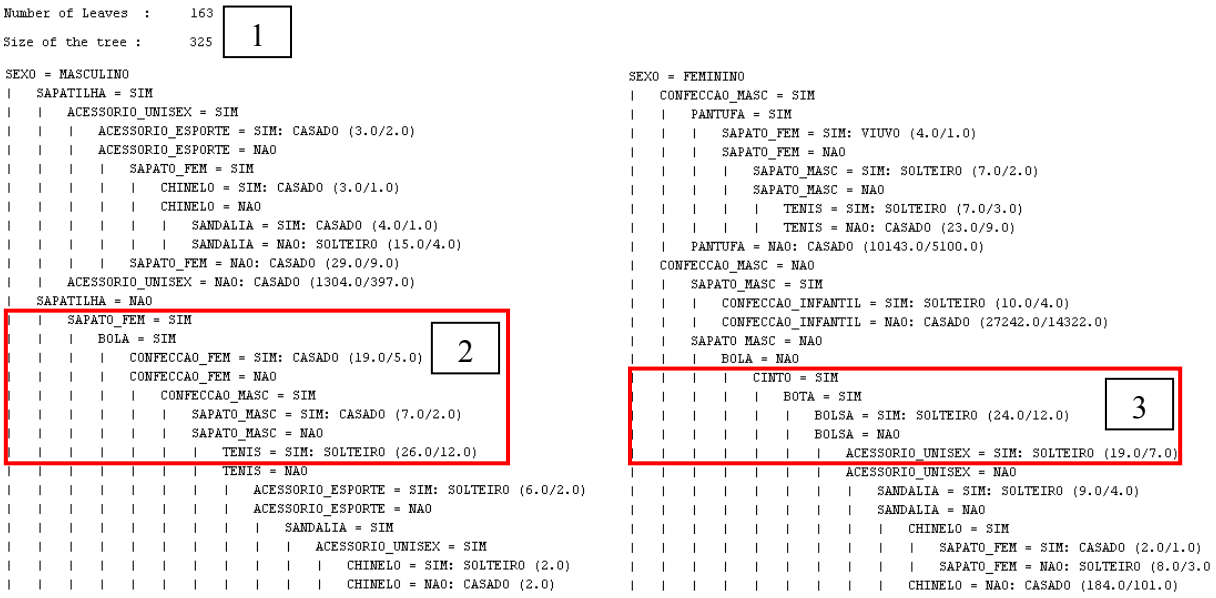

Figura 6.6 – Árvore de classificação do J48 atributo alvo estado civil

Na figura 6.6 marcada como item (2), algumas relações entre produtos adquiridos na mesma nota fiscal foram classificadas. Pode-se observar que clientes do sexo masculino casados, compraram na mesma transação, sapato feminino, bola e confecção feminina, com 26,32% de acerto. Outra relação interessante também é apresentada, que o mesmo perfil de cliente, masculino e casado, tende a comprar sapato feminino, bola, confecção masculina e sapato masculino, com 28,57% de acerto. Já os clientes masculinos, porém solteiros, apresentaram uma relação entre sapato feminino, bola, confecção masculina e tênis, com 46,15% de acerto.

No item marcado como (3), outro perfil interessante é apresentado, clientes do sexo feminino com estado civil igual a solteiro, compram cinto, bota e bolsa no mesmo cupom fiscal, com 50% de acerto. Ainda com esse perfil de cliente, é possível destacar a relação existente entre os itens cinto, bota e acessórios unisex, com 36,84% de acerto. Importante destacar que com a mineração de dados houve a confirmação de vários padrões já conhecidos como, homens solteiros que compram bola e acessório esporte, os quais não foram destacados no trabalho, visto que o objetivo é descobrir novos padrões.

Conforme apresentado anteriormente na figura 6.5, o percentual de instâncias classificadas corretamente ficou baixo, em apenas em 47,73 %. Com objetivo de buscar um percentual maior, foi alterado o parâmetro *confidenceFactor*, que é o fator de confiança utilizado para a efetuar a poda, para 0.20 e o percentual se manteve o mesmo, porém com diminuição do tamanho da árvore para 205 com 103 folhas. Uma nova tentativa foi feita,

agora alterando o parâmetro *reducedErrorPruning* para *true* com intuito de reduzir os erros na poda, porém não foi obtido sucesso pelo fato do percentual ter se mantido o mesmo. Nessa última execução a árvore foi gerada com um tamanho de 1213 com total de 607 folhas.

Uma nova execução do algoritmo foi iniciada, porém o atributo alvo utilizado foi alterado para o SEXO e não mais o ESTADO_CIVIL. O resultado desse processamento foi mais satisfatório que o anterior, pois um total de 70,15% das instâncias foram classificadas corretamente como mostra a figura 6.7. Para essa validação, foram mantidos os parâmetros *default* da ferramenta e também o número de *folds* foi mantido a quantidade de dois, conforme validação anterior.

```
Correctly Classified Instances      154261               70.1544 %
Incorrectly Classified Instances     65627               29.8456 %
Kappa statistic                          0.0635
Mean absolute error                      0.4013
Root mean squared error                  0.4484
Relative absolute error                 94.8356 %
Root relative squared error             97.4858 %
Total Number of Instances           219888

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.071     0.023      0.575     0.071     0.126    MASCULINO
  0.977     0.929      0.706     0.977     0.82     FEMININO

=== Confusion Matrix ===

     a      b   <-- classified as
  4727  62129 |      a = MASCULINO
  3498 149534 |      b = FEMININO
```

Figura 6.7 – Resultados estatísticos da execução do J48 atributo alvo sexo

Com base na nova árvore de classificação gerada, uma nova análise tornou-se necessária na busca por novos conhecimentos utilizáveis. O processamento do algoritmo utilizando 2 *folds* para o *cross-validation* levou 44 segundos para execução, conforme apresentado na figura 6.8 item (1). Conforme citado por Goldschmidt (2005) o tempo de processamento tende a ser menor, quanto menor o número de variáveis do atributo. A árvore foi gerada inicialmente com tamanho igual a 214 com 112 folhas como apresenta a figura 6.8 no item marcado como (2).

LOG TEMPO EXECUÇÃO 1

```
22:51:20: Started weka.classifiers.trees.J48
22:52:04: Finished weka.classifiers.trees.J48
```

2

```
Number of Leaves  :     112
Size of the tree :     214
```

Figura 6.8 – Informações sobre execução do J48 com atributo alvo sexo

Para apresentar a árvore obtida como resultado do processamento do algoritmo, foi adotado o mesmo critério anterior de separação dos pontos considerados com maior relevância para facilitar a visualização das regras classificadas. A figura 6.9 item (1) apresenta uma relação interessante entre produtos adquiridos na mesma nota fiscal. Pode-se observar que clientes do sexo masculino, compraram na mesma transação, bola, sapato feminino e sandália. O item sandália pode ser masculino ou feminino, pois esse item não está classificado de forma distinta.

1

```
BOLA = SIM
|   SAPATO_FEM = SIM
|   |   SANDALIA = SIM: MASCULINO (584.0/38.0)
|   SANDALIA = NAO
|   |   ESTADO_CIVIL = CASADO
|   |   |   SAPATO_MASC = SIM: MASCULINO (14.0/5.0)
|   |   |   SAPATO_MASC = NAO: FEMININO (55.0/18.0)
|   |   ESTADO_CIVIL = SOLTEIRO
|   |   |   SAPATILHA = SIM: FEMININO (4.0)
|   |   |   SAPATILHA = NAO
|   |   |   |   TENIS = SIM
|   |   |   |   |   CONFECCAO_MASC = SIM: FEMININO (2.0)
|   |   |   |   |   CONFECCAO_MASC = NAO: MASCULINO (7.0)
|   |   |   |   TENIS = NAO
|   |   |   |   |   BOLSA = SIM: MASCULINO (2.0)
|   |   |   |   |   BOLSA = NAO
|   |   |   |   |   |   CONFECCAO_MASC = SIM: MASCULINO (12.0/4.0)
|   |   |   |   |   |   CONFECCAO_MASC = NAO: FEMININO (25.0/10.0)
|   |   ESTADO_CIVIL = DIVORCIADO: FEMININO (15.0/3.0)
|   |   ESTADO_CIVIL = VIUVO: FEMININO (10.0)
|   |   ESTADO_CIVIL = OUTROS: FEMININO (0.0)
|   SAPATO_FEM = NAO
|   |   ESTADO_CIVIL = CASADO
|   |   |   SANDALIA = SIM: FEMININO (62.0/20.0)
|   |   |   SANDALIA = NAO
|   |   |   |   SAPATILHA = SIM
|   |   |   |   |   TENIS = SIM: FEMININO (2.0)
```

2

```
BOLA = NAO
|   ESTADO_CIVIL = CASADO: FEMININO (102927.0/35947.0)
|   ESTADO_CIVIL = SOLTEIRO
|   |   CONFECCAO_MASC = SIM
|   |   |   BOTA = SIM
|   |   |   |   ACESSORIO_UNISEX = SIM: FEMININO (8.0)
|   |   |   ACESSORIO_UNISEX = NAO
|   |   |   |   CHINELO = SIM: FEMININO (4.0)
|   |   |   |   CHINELO = NAO
|   |   |   |   |   OUTROS = SIM: MASCULINO (9.0/3.0)
|   |   |   |   |   OUTROS = NAO
|   |   |   |   |   |   CHUTEIRA = SIM: MASCULINO (6.0/2.0)
|   |   |   |   |   |   CHUTEIRA = NAO
|   |   |   |   |   |   |   BOLSA = SIM: FEMININO (21.0/1.0)
|   |   |   |   |   |   |   BOLSA = NAO
|   |   |   |   |   |   |   |   CINTO = SIM: MASCULINO (4.0)
|   |   |   |   |   |   |   |   CINTO = NAO: FEMININO (81.0/22.0)
|   |   |   BOTA = NAO
|   |   |   |   SAPATILHA = SIM
|   |   |   |   |   OUTROS = SIM: MASCULINO (4.0/1.0)
|   |   |   |   |   OUTROS = NAO: FEMININO (27.0/7.0)
|   |   |   |   SAPATILHA = NAO
|   |   |   |   |   SANDALIA = SIM
|   |   |   |   |   |   CHINELO = SIM: FEMININO (127.0/36.0)
|   |   |   |   |   CHINELO = NAO
|   |   |   |   |   |   CHUTEIRA = SIM
|   |   |   |   |   |   |   SAPATO MASC = SIM: MASCULINO (2.0)
```

3

Figura 6.9 – Árvore de classificação do J48 atributo alvo sexo

Com base na figura 6.9 marcada como item (2), pode-se salientar o perfil de clientes do sexo feminino com estado civil igual a solteiro, compram confecção masculina, bota e acessório unisex na mesma nota. Ainda na mesma figura, marcada como item (3), outra relação interessante é apresentada, onde clientes do sexo feminino compram sandália e chinelo na mesma transação de venda. Importante salientar, que essas classificações não tinham sido evidenciadas no processamento anterior, onde o atributo alvo utilizado foi o ESTADO_CIVIL.

```
|   |   |   |   SAPATO_FEM = NAO
|   |   |   |   |   CINTO = SIM
|   |   |   |   |   |   BOTA = SIM
|   |   |   |   |   |   |   BOLSA = SIM: FEMININO (37.0/1.0)
|   |   |   |   |   |   |   BOLSA = NAO
|   |   |   |   |   |   |   |   TENIS = SIM: MASCULINO (11.0/3.0)
|   |   |   |   |   |   |   |   TENIS = NAO: FEMININO (66.0/28.0)
|   |   |   |   |   |   BOTA = NAO
|   |   |   |   |   |   |   SANDALIA = SIM
|   |   |   |   |   |   |   |   ACESSORIO_ESPORTE = SIM: MASCULINO (9.0/3.0)
|   |   |   |   |   |   |   |   ACESSORIO_ESPORTE = NAO
|   |   |   |   |   |   |   |   |   TENIS = SIM
|   |   |   |   |   |   |   |   |   |   OUTROS = SIM: FEMININO (3.0/1.0)
|   |   |   |   |   |   |   |   |   |   OUTROS = NAO: MASCULINO (10.0/3.0)
|   |   |   |   |   |   |   |   |   TENIS = NAO: FEMININO (108.0/43.0)
|   |   |   |   |   |   |   SANDALIA = NAO: MASCULINO (1351.0/562.0)
|   |   |   |   |   CINTO = NAO: FEMININO (11544.0/4888.0)
|   |   |   SAPATO_MASC = NAO: FEMININO (60851.0/15380.0)
```

Figura 6.10 – Classificação obtida com atributo alvo distintos

Já a figura 6.10, apresenta a mesma classificação identificada anteriormente, onde evidencia clientes do sexo feminino que compram cinto, bota e bolsa no mesmo cupom fiscal. Vale salientar que o número de instâncias classificadas corretamente com o atributo alvo SEXO passou para 37 contra as 24 instâncias do atributo alvo ESTADO_CIVIL.

Nessa etapa, o parâmetro *confidenceFactor*, também foi alterado para 0.20 e o percentual de classificações corretas se manteve o mesmo, da mesma forma ocorreu com a alteração do parâmetro *reducedErrorPruning* para *true* onde o percentual se manteve o mesmo, com alterações apenas em relação a altura e largura da árvore.

Buscando um maior percentual de instâncias classificadas corretamente e também a descoberta de novos conhecimentos diferentes dos já obtidos, nas próximas seções, será avaliado o comportamento dos métodos de meta aprendizagem *bagging* e *boosting* disponíveis na ferramenta Weka. A utilização desses métodos em outros trabalhos como o Gonchorosky (2005), que obteve um ganho significativo no percentual de instâncias classificadas corretamente, motivou o emprego desses métodos na base utilizada para esse trabalho buscando melhores resultados como os obtidos pelo autor.

6.4 Validação *Bagging*

O método *bagging*, tem a função de aumentar o desempenho de um algoritmo classificador e também sua capacidade de geração de regras, construindo classificadores a partir do conjunto de amostra, que de forma independente e sucessiva são processadas. Uma

diferença em relação ao J48 está no número de sementes que é utilizado, pois o *bagging* define um número aleatório de sementes para gerar o conjunto de reamostragem (GONCHOROSKY, 2007). Nesse trabalho, foi utilizado o algoritmo J48 incorporado ao método *Bagging* implementado pelo Weka .

A primeira execução foi efetuada com os parâmetros *default* do algoritmo e apresentou um resultado de 70.09% de instâncias classificadas corretamente. Em comparação com a execução direta do J48, que obteve um percentual de 70.15, os resultados gerados foram semelhantes. Os dados foram submetidos novamente ao algoritmo, porém o parâmetro *seed*, que segundo Weka(2008), serve para definir o número aleatório de sementes a ser utilizada, foi alterado de um para dois. O resultado dessa nova execução também se manteve semelhante a execução direta do J48, tanto em relação as regras geradas, quanto ao percentual de 70.11 instâncias corretamente classificadas conforme mostra a figura 6.11.

```
Time taken to build model: 503.32 seconds

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

Correctly Classified Instances      154166               70.1111 %
Incorrectly Classified Instances     65722               29.8889 %
Kappa statistic                          0.0682
Mean absolute error                      0.399
Root mean squared error                  0.447
Relative absolute error                 94.2807 %
Root relative squared error             97.1669 %
Total Number of Instances           219888

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.079     0.027      0.56      0.079     0.138    MASCULINO
  0.973     0.921      0.707     0.973     0.819    FEMININO

=== Confusion Matrix ===

      a      b   <-- classified as
   5253  61603 |      a = MASCULINO
   4119 148913 |      b = FEMININO
```

Figura 6.11 – Resultado estatístico do método *Bagging*

6.5 Validação *Boosting*

Segundo Weka (2008), o método *Boosting* serve para impulsionar um classificador nominal, onde apenas problemas de classes nominais podem ser resolvidos. Conforme

Gonchorosk (2005), no método *Boosting* inicialmente todas as instâncias são geradas com um peso igual. Cada instância classificada como erro na primeira indução tem o seu peso alterado, tendo como base os classificadores anteriormente construídos através de reponderação.

Para esse estudo, foi utilizado o método AdaboostM1 implementado pelo Weka. A primeira execução foi efetuada com os parâmetros *default* do algoritmo e obteve-se um resultado de 69.77% de instâncias classificadas corretamente, gerando um resultado semelhante a execução direta do J48, que alcançou um percentual de 70.15. Importante salientar que além de não ter classificado corretamente um maior percentual de instância, o processo de análise tornou-se muito oneroso e lento, pois o parâmetro *numIterations* que define o número de iterações que será executado pelo método, possui valor *default* de 10, fazendo com que 10 árvores fossem geradas, cada uma possuindo um peso diferente.

Uma nova execução foi iniciada, porém com o parâmetro *useResampling* alterado para *true*. Esse parâmetro serve para utilizar o processo de reamostragem ao invés de utilizar a reponderação (WEKA, 2008). O resultado dessa nova execução gerou um percentual de 69.70 de instâncias classificadas corretamente como mostra a figura 6.12, sendo, ainda menor que o resultado de 69.77% alcançado na execução anterior.

```
Time taken to build model: 1431.07 seconds

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

Correctly Classified Instances      153270               69.7037 %
Incorrectly Classified Instances     66618               30.2963 %
Kappa statistic                          0.0918
Mean absolute error                      0.3914
Root mean squared error                  0.4449
Relative absolute error                 92.4744 %
Root relative squared error             96.7132 %
Total Number of Instances           219888

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.126     0.053      0.507     0.126     0.202    MASCULINO
  0.947     0.874      0.713     0.947     0.813    FEMININO

=== Confusion Matrix ===

     a      b   <-- classified as
  8410  58446 |      a = MASCULINO
  8172 144860 |      b = FEMININO
```

Figura 6.12 – Resultado estatístico do método *Boosting*

Segundo Weka (2008), muitas vezes a utilização do *Boosting* melhora significativamente o desempenho, mas por vezes, pode ocorrer *overfitting* devido ao grande número de variáveis, o que faz o classificador ajustar-se em excesso ao conjunto de treinamento perdendo a capacidade de generalização. Essa citação justifica a diferença de resultados obtidos no exemplo do trabalho do Gonchorosk (2005) em relação aos resultados obtidos nesse trabalho.

6.6 Aplicação em Período Sazonal

Em uma conversa informal com o especialista de marketing juntamente com o analista de negócio da área, tendo como objetivo a discussão dos resultados obtidos com a mineração, foi sugerido analisar um período sazonal. A partir dessa sugestão uma base de dados foi preparada buscando identificar as características de consumo dos clientes de um período específico do ano. Para o varejo, algumas datas comemorativas, como natal, dia das mães, dia dos pais e também o dia dos namorados, representam um volume maior de venda. Para fazer essa análise, foi sugerido o dia dos namorados, selecionando os dados referentes ao período de 20/05 a 12/06 dos anos de 2006 e 2007.

Uma nova tabela não normalizada foi gerada com um total de 13.687 transações. A partir dessa tabela, um novo arquivo com extensão arff foi gerado. Logo após os dados foram submetidos à aplicação do J48 com todos os parâmetros *default* utilizando o atributo ESTADO_CIVIL como atributo alvo. Como apresentado na análise anterior, o atributo alvo ESTADO_CIVIL apresentou um percentual baixo de instâncias corretamente classificadas se comparado com o atributo alvo SEXO. O parâmetro *reduceErrorPruning* foi então alterado para *true* buscando um maior percentual de classificação correta. Mesmo após a alteração do parâmetro, o percentual de classificação se manteve baixo como mostra a figura 6.13.

```
Correctly Classified Instances        6363               46.537  %
Incorrectly Classified Instances      7310               53.463  %
Kappa statistic                          0.0265
Mean absolute error                      0.2449
Root mean squared error                  0.354
Relative absolute error                 98.6133 %
Root relative squared error            100.4516 %
Total Number of Instances            13673

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.814     0.781      0.478     0.814     0.602    CASADO
  0.219     0.192      0.42      0.219     0.288    SOLTEIRO
  0         0          0         0         0        DIVORCIADO
  0         0.001      0         0         0        VIUVO
  0         0          0         0         0        OUTROS

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 5202 1181    3    5    0 |    a = CASADO
 4133 1161    2    4    0 |    b = SOLTEIRO
  772  226    0    1    0 |    c = DIVORCIADO
  784  195    1    0    0 |    d = VIUVO
    1    2    0    0    0 |    e = OUTROS
```

Figura 6.13 – Resultado estatístico do período sazonal com atributo alvo ESTADO_CIVIL

Após a execução do algoritmo, algumas regras interessantes podem ser relacionadas para o período de data proposto. A figura 6.14 marcada com item (1) apresenta o perfil de clientes casados que compram confecção feminina. No item (2) aparecem os clientes casados que compram acessório esporte e botas na mesma NF.

```
SEXO = MASCULINO
|   CONFECCAO_INFANTIL = SIM: CASADO (6.0)
|   CONFECCAO_INFANTIL = NAO
|   |   CONFECCAO_FEM = SIM: CASADO (70.0/28.0)
|   |   CONFECCAO_FEM = NAO
|   |   |   ACESSORIO_UNISEX = SIM
|   |   |   |   CHUTEIRA = SIM: CASADO (3.0/1.0)
|   |   |   |   CHUTEIRA = NAO
|   |   |   |   |   ACESSORIO_ESPORTE = SIM
|   |   |   |   |   |   SAPATO_MASC = SIM: CASADO (5.0/2.0)
|   |   |   |   |   |   SAPATO_MASC = NAO
|   |   |   |   |   |   |   BOTA = SIM: CASADO (5.0)
|   |   |   |   |   |   |   BOTA = NAO: SOLTEIRO (5.0/1.0)
|   |   |   |   |   ACESSORIO_ESPORTE = NAO
|   |   |   |   |   |   PANTUFA = SIM: CASADO (4.0/1.0)
|   |   |   |   |   |   PANTUFA = NAO: SOLTEIRO (110.0/57.0)
|   |   |   ACESSORIO_UNISEX = NAO
|   |   |   |   CONFECCAO_MASC = SIM
|   |   |   |   |   CHINELO = SIM: CASADO (2.0)
|   |   |   |   |   CHINELO = NAO
|   |   |   |   |   |   SAPATO_MASC = SIM
|   |   |   |   |   |   |   ACESSORIO_ESPORTE = SIM: SOLTEIRO (3.0/1.0)
|   |   |   |   |   |   |   ACESSORIO_ESPORTE = NAO
|   |   |   |   |   |   |   |   TENIS = SIM: CASADO (11.0/3.0)
|   |   |   |   |   |   |   |   TENIS = NAO: SOLTEIRO (20.0/8.0)
|   |   |   |   |   |   SAPATO_MASC = NAO
|   |   |   |   |   |   |   SAPATO_FEM = SIM
|   |   |   |   |   |   |   |   BOLA = SIM: SOLTEIRO (2.0)
|   |   |   |   |   |   |   |   BOLA = NAO: CASADO (2.0)
```

1

2

```
SEXO = MASCULINO
|   |   |   SAPATO_MASC = NAO
|   |   |   |   SAPATO_FEM = SIM
|   |   |   |   |   BOLA = SIM: SOLTEIRO (2.0)
|   |   |   |   |   BOLA = NAO: CASADO (2.0)
|   |   |   |   SAPATO_FEM = NAO
|   |   |   |   |   BOLA = SIM: CASADO (10.0/3.0)
|   |   |   |   |   BOLA = NAO: SOLTEIRO (243.0/122.0)
|   CONFECCAO_MASC = NAO
|   |   BOLA = SIM
|   |   |   ACESSORIO_ESPORTE = SIM
|   |   |   |   SAPATO_MASC = SIM: CASADO (2.0)
|   |   |   |   SAPATO_MASC = NAO: SOLTEIRO (3.0)
|   |   |   ACESSORIO_ESPORTE = NAO: CASADO (13.0/1.0)
|   |   BOLA = NAO
|   |   |   CHUTEIRA = SIM
|   |   |   |   CINTO = SIM: CASADO (3.0)
|   |   |   |   CINTO = NAO
|   |   |   |   |   BOTA = SIM: CASADO (18.0/3.0)
|   |   |   |   |   BOTA = NAO
|   |   |   |   |   |   TENIS = SIM
|   |   |   |   |   |   |   BOLSA = SIM: SOLTEIRO (4.0/1.0)
|   |   |   |   |   |   |   BOLSA = NAO: CASADO (30.0/9.0)
|   |   |   |   |   |   TENIS = NAO
|   |   |   |   |   |   |   BOLSA = SIM: CASADO (2.0)
|   |   |   |   |   |   |   BOLSA = NAO
|   |   |   |   |   |   |   |   ACESSORIO_ESPORTE = SIM: CASADO (11.0/4.0)
|   |   |   |   |   |   |   |   ACESSORIO_ESPORTE = NAO: SOLTEIRO (51.0/19.0)
|   |   |   CHUTEIRA = NAO: CASADO (3247.0/1447.0)
```

3

4

Figura 6.14 – Resultados 1 do período sazonal com atributo alvo ESTADO_CIVIL

Ainda na figura 6.14 duas regras podem ser destacadas para clientes masculinos e solteiros. Marcada com o item (3), aparecem os clientes solteiros que compram sapato feminino e bola na mesma NF e no item (4), apresenta os clientes solteiros que compram chuteira, tênis e bolsa na mesma NF.

Na figura 6.15, algumas regras interessantes são destacadas para clientes do SEXO feminino considerando o mesmo período sazonal. O item (1) apresenta o perfil de clientes solteiras que compram confecção feminina e chuteira na mesma NF. Já o item (2) mostra as clientes casadas que adquirem bola de futebol. No item marcado como (3), aparece o perfil de clientes solteiras que compram bolsa e chuteira na mesma NF e também as que compram bolsa e confecção masculina na mesma transação de compra.

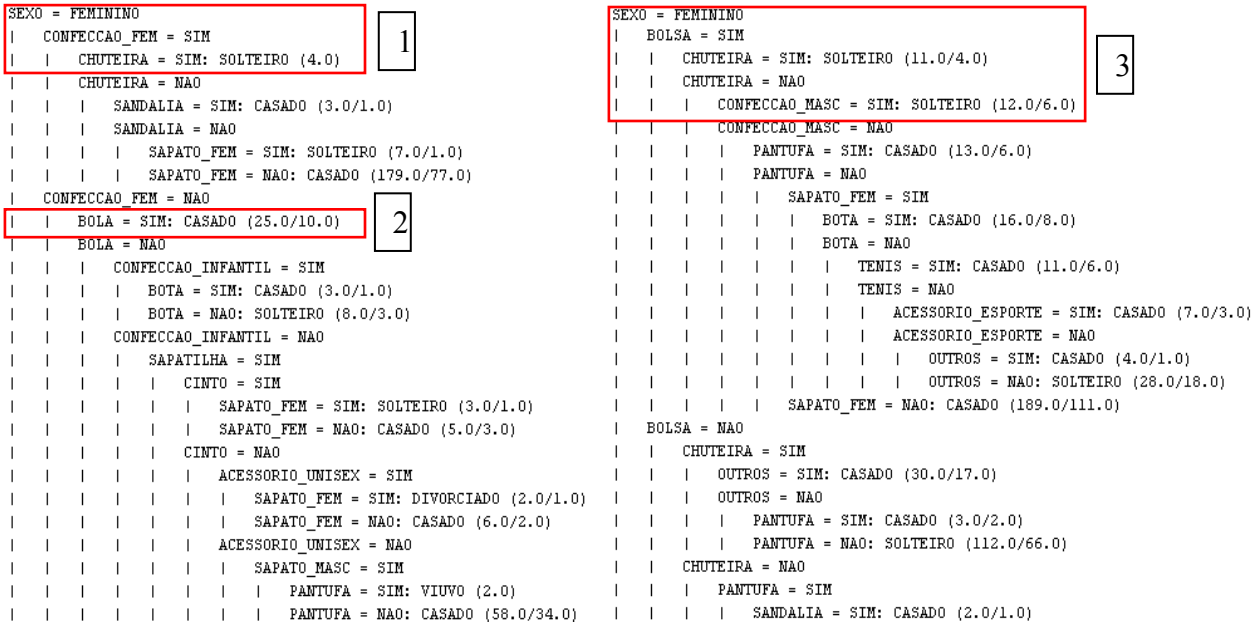

Figura 6.15 – Resultados 2 do período sazonal com atributo alvo ESTADO_CIVIL

Com base nos resultados apresentados anteriormente, onde foi obtido um percentual maior de instâncias corretamente classificadas com a utilização do atributo SEXO como atributo alvo, uma nova análise foi efetuada com esse atributo. Primeiramente o algoritmo foi executado com os parâmetros *default*, gerando uma árvore de apenas uma folha. Uma nova execução foi efetuada, agora alterando o parâmetro *reduceErrorPruning* para *true* o que apresentou um resultado satisfatório com um percentual de 70.94 de instâncias corretamente classificadas como mostra a figura 6.16.

```
Number of Leaves  :     115

Size of the tree :      226


Time taken to build model: 0.58 seconds

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

Correctly Classified Instances        9700               70.9427 %
Incorrectly Classified Instances      3973               29.0573 %
Kappa statistic                          0.0177
Mean absolute error                      0.3863
Root mean squared error                  0.4448
Relative absolute error                 94.968  %
Root relative squared error             98.6203 %
Total Number of Instances            13673

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.036     0.023      0.381     0.036     0.066    MASCULINO
  0.977     0.964      0.719     0.977     0.828    FEMININO

=== Confusion Matrix ===

    a    b   <-- classified as
  141 3744 |    a = MASCULINO
  229 9559 |    b = FEMININO
```

Figura 6.16 – Resultado estatístico do período sazonal com atributo alvo SEXO

Algumas regras interessantes foram obtidas após a execução do algoritmo. Como mostra a figura 6.17 marcada com item (1), apresenta o perfil de clientes do sexo masculino e casados compram sapato feminino e chuteira, já no item (2) aparecem os homens casados que compram botas, tênis e bolsa na mesma NF.

```
ESTADO_CIVIL = CASADO
|   BOLA = SIM: MASCULINO (30.0/13.0)
|   BOLA = NAO
|   |   CONFECCAO_INFANTIL = SIM
|   |   |   CONFECCAO_MASC = SIM: MASCULINO (3.0/1.0)
|   |   |   CONFECCAO_MASC = NAO: FEMININO (3.0/1.0)
|   |   CONFECCAO_INFANTIL = NAO
|   |   |   SAPATO_FEM = SIM
|   |   |   |   CHUTEIRA = SIM: MASCULINO (6.0/2.0)
```

1

```
|   |   |   |   |   BOTA = SIM
|   |   |   |   |   |   TENIS = SIM
|   |   |   |   |   |   |   BOLSA = SIM: MASCULINO (5.0/2.0)
|   |   |   |   |   |   |   BOLSA = NAO: FEMININO (36.0/13.0)
|   |   |   |   |   |   TENIS = NAO: FEMININO (99.0/30.0)
|   |   |   |   |   BOTA = NAO: FEMININO (409.0/157.0)
```

2

Figura 6.17 – Resultados 1 do período sazonal com atributo alvo SEXO

A figura 6.18 apresenta um perfil interessante de clientes do sexo feminino e solteira que compram confecção masculina e também outro perfil com as mesmas características quanto a sexo e estado civil, que compram chuteira.

```
ESTADO_CIVIL = SOLTEIRO
|   |   SAPATO_MASC = NAO
|   |   |   CONFECCAO_MASC = SIM: FEMININO (210.0/89.0)
|   |   |   CONFECCAO_MASC = NAO
|   |   |   |   CHUTEIRA = SIM: FEMININO (83.0/34.0)
|   |   |   |   CHUTEIRA = NAO
|   |   |   |   |   TENIS = SIM
|   |   |   |   |   |   CHINELO = SIM: MASCULINO (7.0/2.0)
|   |   |   |   |   |   CHINELO = NAO
|   |   |   |   |   |   |   SANDALIA = SIM: FEMININO (7.0)
|   |   |   |   |   |   |   SANDALIA = NAO
|   |   |   |   |   |   |   |   PANTUFA = SIM: FEMININO (15.0/2.0)
```

Figura 6.18 – Resultados 2 do período sazonal com atributo alvo SEXO

Esta relação entre a compra de chuteiras por esse perfil de cliente, já foi mencionada na análise feita considerando o atributo alvo estado civil, porém associado com outros produtos. Dessa forma torna mais evidente, a escolha do produto chuteira pelas clientes do sexo feminino, para presentear namorados nessa data comemorativa escolhida para análise. Importante salientar que as regras destacadas nesse trabalho, foram consideradas não óbvias, pois outras regras já conhecidas como, mulheres casadas que compram sapato feminino e bolsa na mesma nota fiscal, não foram citadas. Com essas regras pode-se comprovar que esse comportamento já conhecido de fato ocorre nas transações de vendas.

CONCLUSÃO

A partir da revisão bibliográfica e das validações práticas feitas para esse estudo, foi possível compreender os ganhos competitivos que uma empresa de varejo calçadista pode obter, através da garimpagem e extração de conhecimento do seu volumoso banco dos dados. Volume esse que é gerado, durante os dias de funcionamento de cada loja, onde ocorrem centenas de transações de vendas, refletindo o perfil do seu público consumidor.

Considerando o setor de varejo, que trabalha com margens de lucros reduzidas, a utilização de ferramentas e técnicas de *data mining*, pode representar um aumento de competitividade frente a seus concorrentes. Para que as técnicas de *data mining* fossem aplicadas, as etapas anteriores do processo de KDD como: seleção dos dados, limpeza, enriquecimento, transformação ou codificação foram executadas, a fim de preparar os dados para aplicação dos algoritmos. Importante salientar que esse é um processo que necessita grande envolvimento humano, pois requer uma interação muito grande do analista de KDD juntamente com o analista de marketing, validando os resultados obtidos e também propondo novas análises ou até mesmo um enriquecimento dos dados facilitando o processo de mineração.

Com base nesse estudo, foi possível verificar que o algoritmo J48 apresentou maior aderência a base de dados da referida empresa de varejo calçadista. Foram observadas algumas regras interessantes que apresentaram novos conhecimentos sobre o perfil de compra dos clientes. Como exemplo, pode-se citar clientes do sexo masculino, que compraram bola, sapato feminino e sandália ou ainda o perfil de clientes do sexo feminino com estado civil igual a solteiro, que compram confecção masculina, bota e acessório unisex na mesma nota, além dos demais resultados apresentados anteriormente.

A partir das análises feitas levando em consideração o período sazonal conforme sugerido pelo especialista em marketing, novas regras foram descobertas. Alguns perfis de padrão de compra dos clientes para o dia dos namorados sugerem campanhas de marketing

específicas para esse público. A sugestão de aquisição conjunta de determinados produtos em uma mesma experiência de venda, podem ser feitas tanto por parte de propagandas e mídia quanto pela sugestão dos vendedores, que podem fomentar a venda tendo posse dessa informação.

Considerando a importância da etapa de apresentação do conhecimento descoberto, pretende-se direcionar formalmente os resultados obtidos para o setor de marketing da empresa. Será indicado como trabalho futuro a criação de métricas de vendas para determinados produtos antes da aplicação da mineração na base. Após a criação das métricas, sugere-se desenvolver uma campanha de marketing com base nos resultados obtidos nesse trabalho, para depois validar se houve incremento nas vendas dos produtos apresentados nos resultados. Outra sugestão seria efetuar as mesmas validações, utilizando outra ferramenta de mineração ou ainda outros algoritmos para comparar com os resultados obtidos nesse trabalho.

REFERÊNCIAS BIBLIOGRÁFICAS

AGRAWAL, Rakesh; IMIELINSKI, Thomas; SWAMI, Arun. Mining Association Rules between Sets of Items in Large Databases. In: ACM SIGMOD International Conference on Management of Data 5/93, 1993, Washington/USA. **Proceedings of SIGMOD 5/93**. Washington/USA, 1993. p. 207-216.

AGRAWAL, Rakesh; SRIKANT, Ramakrishnan. Fast Algorithms for Mining Association Rules. In: 20th International Conference on Very Large Databases (VLDB) CONFERENCE, 1994, Santiago/Chile. **Proceedings of the 20th International Conference on Large Databases**. Santiago/Chile, 1994. p. 487-498.

BISPO, C. A. F. **Uma análise da nova geração de sistemas de apoio à decisão**. São Carlos, 1998. 160 p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo. Disponível em: < http://www.teses.usp.br/teses/disponiveis/18/18140/tde-04042004-152849>. Acesso em: 19 mar. 2008.

CABENA, Peter et al. **Discovering Data Mining from Concept to Implementation**. New Jersey-USA: Prentice Hall PTR, 1997. 193 p.

CARVALHO, Juliano Varella, **Mineração de Dados e a Descoberta de Conhecimento**, 2008. Apresentação cedida pelo professor.

CARVALHO, Juliano V. Reconhecimento de Caracteres Manuscritos Utilizando Regras de Associação. In: Dissertação de Mestrado Curso de Pós-Graduação em Informática da Universidade Federal da Paraíba, 2000. Disponível em: < http://www.ourgrid.org/twiki-public/bin/view/COPIN/DissertacoesMestrado?sortcol=1&table=3&up=0 >. Acesso em: 19 mar. 2008.

DATE, C.J. **Introdução a Sistemas de Bancos de Dados.** Rio de Janeiro: Elsevier, 2003.

ELMASRI, Ramez; NAVATHE, Shamkant B. Conceitos de Data Mining. In: **Sistemas de banco de Dados.** 4 ed. São Paulo: Addison, 2005. p.625-657.

FAYYAD, Usama M.; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From Data Mining to Knowledge Discovery**: FAYYAD et al. G. Cambridge-Mass:AAAI/MIT Press. 1996. p. 37-51 http://www.aaai.org/aitopics/assets/PDF/AIMag17-03-2-article.pdf. Acesso em: 4 mar. 2008.

FRANK, Eibe. **Machine Learning with WEKA:** Department of Computer Scienci, University of Waikato, New Zealand. http://sourceforge.net/project/downloading.php?groupname=weka&filename=weka.ppt&use_mirror=ufpr . Acesso em: 29 set. 2008.

GIUDICI, Paolo**. Applied Data Mining : Statistical Methods for Business and Industry.** England: Wiley, 2003.

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

GOLDSCHMIDT, Ronaldo. **KDD e Mineração de Dados Métodos Baseados em Lógica Nebulosa.** Material Didático Utilizado nas disciplinas. Disponível em: http://br.geocities.com/ronaldo_goldschmidt/top_esp_ia.htm - Métodos de Mineração de Dados. Acessado em: 8 jun. 2008.

GONCHOROSKY, Sidinei Pereira. **Utilização de técnicas de KDD em um call center ativo**. Novo Hamburgo: 2007. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto de Ciências Exatas e Tecnológicas, Feevale, 2007. Disponível em: <http://ead.feevale.br/tc/files/1015.pdf>. Acesso em: 13 set. 2007.

MACKENNA, Regis. **Estratégias de Marketing em Tempos de Crise.** 5 ed. Rio de Janeiro: Campus, 1989.

MACKENNA, Regis. **Competindo em Tempo Real: Estratégias Vencedoras para a era do Cliente Nunca Satisfeito.** 3 ed. Rio de Janeiro: Campus, 1998.

MACKENNA, Regis. **Marketing de Relacionamento.** Rio de Janeiro: Campus, 1999.

MACKENNA, Regis. **Acesso Total: O Novo Conceito de Marketing de Atendimento.** Rio de Janeiro: Campus, 2002.

MONGIOVI, Giuseppe. **T.E.I. Data Mining**. Notas de Aula Data Mining. Campina Grande, 1998. p. 1-102.

PIATETSKY-SHAPIRO, Gregory. **From Data Mining to Knowledge Discovery**: An Introduction. S.l., 2005, 32p. Disponível em: http://www.kdnuggets.com/data_mining_course/x4-data-mining-to-knowledge-discovery.ppt> Acessado em: 28 mai. 2008.

OLIVEIRA Fernando L. et al. Utilização de Algoritmos Simbólicos para a Identificação do Número de Caroços do Fruto Pequi, In: IV Encontro de Estudantes de Informática do Estado do Tocantins**,** 2002, Palmas. **Encontro de Estudantes de Informática do Tocantins – Encoinfo**. Palmas, 2002. p. 34-43.

PRODANOV, Cleber C. **Manual de Metodologia Científica**. 3 ed. Novo Hamburgo: Feevale, 2006. 77p.

QUINLAN, Ross. Improved use of continuous attributes in C4.5. In: Journal Of Artificial Intelligence Research 4, 1996, p. 77-90. Disponível em: <http://www.jair.org/media/279/live279-1538-jair.pdf>. Acessado em: 19 jun. 2008

QUINLAN, Ross; KOHAVI, Ron. Decision Tree Discovery. 10, 1999, p. 1-16. Disponível em: < http://ai.stanford.edu/~ronnyk/treesHB.pdf >. Acessado em: 19 jun. 2008

SANTOS, Rafael. **Um guia para uso do Weka em *scripts* e integração com aplicações em Java**. Tutorial publicado no Instituto Nacional de Pesquisas Espaciais – INPE, abr 2005, 20p. Disponível em: <http://www.lac.inpe.br/~rafael.santos/Docs/CAP359/2005/weka.pdf>. Acesso em: 14 jun. 2008.

SILVA, Marcelino Pereira dos Santos. **Mineração de Dados - Conceitos, Aplicações e Experimentos com Weka.** IV Escola Regional de Informática RJ/ES (IV ERI RJ/ES) Nov 2004. Artigo publicado na Sociedade Brasileira de Computação – SBC. Disponível em: www.sbc.org.br/bibliotecadigital/download.php?paper=35. Acessado em: 14 jun. 08.

SCHNEIDER, Luis Felipe. **Mineração de dados**: Conceitos. Universidade Federal do Rio Grande do Sul, 9p, 2007. Disponível em: <http://www.inf.ufrgs.br/~clesio/cmp151/cmp15120021/artigo_lfelipe.pdf>. Acesso em: 29 maio. 2008.

SPRAGUE, R. H.; WATSON, H. J. **Sistemas de Apoio à Decisão: Colocando a Teoria em Prática.** Rio de Janeiro: Campus, 1991.

WEKA 3: **Data Mining Software in Java**. Nova Zelândia. Universidade de Waikato, 2008. Apresenta todas as características do projeto e do software Weka. Disponível em <http://www.cs.waikato.ac.nz/ml/weka/index.html>. Acesso em: 05 mar. 2008.