UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA

Departamento de Ciências Exatas

Graduação em Ciência da Computação

Inteligência Empresarial em um Ambiente Acadêmico

José Ramon Trindade Pires

Vitória da Conquista – BA

Janeiro de 2011

Inteligência Empresarial em um Ambiente Acadêmico

José Ramon Trindade Pires

Projeto apresentado ao Curso de Ciência da Computação do Departamento de Ciências Exatas da UESB, orientado pelo professor Fábio Moura Pereira como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Vitória da Conquista – BA

Janeiro de 2011

AGRADECIMENTOS

Agradeço a Deus por sempre ter me proporcionado oportunidades, iluminado meus caminhos e por ter me conduzido por esta importante trajetória de minha vida.

Aos meus pais Maria Lisboa e José Correia, que muito batalharam pela saúde, educação e felicidade de minha família, e por terem me impulsionado e sempre ficado na torcida pra que eu alcançasse meus objetivos.

Aos meus irmãos Gracinha, Teca, Ney, Mônica, Nildo, Cani, Livinha, Marcelo, Valério, Ana e Débora, meus cunhados e sobrinhos, que sempre me ajudaram, incentivaram e apoiaram nos momentos que precisei.

Ao orientador Fábio, por ter ajudado com o tema do trabalho, esclarecimento de dúvidas e por ter me indicado a um emprego de tamanha importância

À professora Maísa que, sempre que precisei, esteve disposta a dar valiosas dicas e conselhos.

À grande amiga Celina, que me ajudou muito nos momentos que precisei, durante todo o curso, oferencendo um ombro amigo e incentivos.

Aos amigos da UESB, Sinthia, Dino, Gabriel, Doug, Jadson, Hilário, Marcos, Elias, Hesdras, Esdras, colegas com quem sempre pude contar nos estudos, horas de diversão e que deram total apoio nos momentos difíceis.

A Lúcio, Marcela, Mara, Roberta, July, Del, pela amizade e pelo apoio.

Eu queria poder citar o nome de cada uma daquelas pessoas que estiveram comigo e me ajudaram a prosseguir nesta etapa da minha vida, mais fica aqui, para aqueles que não citei, o meu sincero e profundo agradecimento.

“Para ganhar conhecimento, adicione coisas todos os dias. Para ganhar sabedoria, elimine coisas todos os dias.”

TSÉ-TUNG

RESUMO

Os avanços na Tecnologia da Informação e o conseqüente crescimento exponencial dos bancos de dados, associados às mudanças ocasionadas por aspectos sociais, econômicos e políticos em ambientes competitivos, exigem que as empresas detenham informações que proporcionem a busca de estratégias de sustentação e de crescimento. Visando permitir às organizações a transformação da grande quantidade de dados brutos em informações e conhecimentos úteis, são utilizadas as tecnologias de Inteligência Empresarial. Os dados do sistema transacional do questionário sócio-cultural da Universidade Estadual do Sudoeste da Bahia (UESB) não são disponibilizados após a sua obtenção através do vestibular. Dessa forma, fez-se necessária a utilização de técnicas de mineração sobre o *data mart* acadêmico, além de tornar possível a visualização dos dados sobre vários pontos de vista. Neste trabalho foi realizada a aplicação de diversos algoritmos de mineração sobre a base de dados do questionário sócio-cultural da Universidade para a obtenção de conhecimentos que facilitem a eficiente tomada de decisões.

Palavras-chave: Inteligência Empresarial, mineração de dados, *data warehouse*, aprendizado de máquina.

ABSTRACT

The advances in Information Technology and the consequent exponential expansion of databases, associated with changes caused by social, economic and political aspects in competitive environments, require companies to have information to provide the search for support and growth strategies. Aiming to enable organizations to transform the vast amount of raw data into useful information and knowledge, are used technologies of Business Intelligence. The data from the transactional system of socio-cultural questionnaire of the State University of Southwest Bahia (UESB) are not available after obtaining it through vestibular contest. Thus, it was necessary to use data mining techniques on data mart academic, and make possible the visualization of data on various points of view. In this study was performed the application of different data mining algorithms on the database of socio-cultural questionnaire of the University to obtain knowledge to facilitate the efficient decision making.

Keywords: Business Intelligence, data mining, *data warehouse*, machine learning.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1. INTRODUÇÃO

Estamos inseridos em um mercado competitivo, no qual as empresas buscam um posicionamento estratégico e, a partir daí, definem as vantagens competitivas sustentáveis a serem desenvolvidas ou aproveitadas. Como conseqüência da Estratégia Competitiva, arena fundamental na qual ocorre a concorrência, surgiram os Sistemas de Apoio à Decisão (SAD). Laudon K. e Laudon J. (2004) afirmam que um SAD tem por objetivo auxiliar o processo de decisão gerencial, combinando dados, ferramentas e modelos analíticos sofisticados e software amigável ao usuário em um único e poderoso sistema que pode dar suporte à tomada de decisão semi-estruturada e não-estruturada. Além disso, um SAD fornece aos usuários um conjunto flexível de ferramentas e capacidades para analisar dados importantes.

No início dos anos 90, começaram a surgir a partir do SAD os conceitos de *data warehouse* e Processamento Analítico On-line (OLAP). Segundo Nextg (2007), o conceito de *data warehouse* surgiu com o objetivo de organizar os dados corporativos da melhor maneira possível, para que pudessem ser acessados e utilizados pelos gerentes e diretores, a fim de auxiliá-los na tomada de decisões.

Os *data warehouses* são criados pelas organizações justamente para fornecer suporte a esse importante aliado dos tomadores de decisão, oferecendo informações precisas e confiáveis aos SADs e esses fornecendo aos gerentes uma visão global da organização, permitindo uma tomada de decisão mais precisa. Como opção para o fornecimento de um retorno mais rápido, as empresas podem utilizar o conceito de *data marts*, subconjuntos de dados do *data warehouse*, que são direcionados a um departamento ou área específica de processos do negócio.

Para Han e Kamber (2001), a construção de *data warehouses* pode ser vista como uma etapa de pré-processamento essencial para a mineração de dados. A mineração, foco deste estudo, é formada por um conjunto de ferramentas e técnicas que, através do uso de algoritmos de aprendizagem ou classificação baseados em redes neurais e estatística, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar padrões nestes dados e auxiliando na descoberta de conhecimento. Esse conhecimento pode ser apresentado por essas

ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão ou grafos.

Como uma recomendação do Ministério da Educação, em cada vestibular é aplicado um questionário sócio-cultural aos candidatos, tendo como objetivos o levantamento de dados sobre o perfil do candidato que pretende ingressar na Universidade e conhecer o público interessado nas vagas da instituição.

Este trabalho foi motivado pelo fato de que os dados do sistema transacional do questionário sócio-cultural da UESB não ficam disponíveis para análise após a realização do vestibular. Assim, a instituição não conhece o perfil dos ingressantes, o que permitiria a tomada de decisões que iriam desde o vestibular a até mesmo projetos de extensão. Outra motivação decorre do fato de que a base de dados do questionário foi analisada em um estudo anterior, no qual foi gerado um *data mart* acadêmico. (SANTOS, 2010)

Dessa forma, o objetivo principal deste trabalho é a aplicação de algoritmos de mineração de dados sobre a base de dados do questionário sócio-cultural, buscando a obtenção de conhecimentos referentes ao perfil dos alunos ingressantes na Universidade. Além disso, visa também analisar a eficiência de cada algoritmo de mineração selecionado e proporcionar à universidade modelos classificatórios de apoio a decisões baseadas em dados do questionário.

A metodologia utilizada na realização do trabalho foi a pesquisa-ação, que consiste em um tipo de pesquisa social com base empírica que é concebida e realizada em estreita associação com uma ação ou com a resolução de um problema coletivo.

O restante deste trabalho está organizado da seguinte maneira: O capítulo 2 abrange os Sistemas de Inteligência Empresarial, onde são aprofundadas as técnicas de mineração de dados, incluindo os algoritmos de classificação e associação utilizados. O terceiro capítulo contém a descrição do *data mart* utilizado no trabalho, além da etapa de pré-processamento realizada antes da aplicação das técnicas de mineração de dados, e é feita uma descrição da ferramenta utilizada. No capítulo 4 são apresentados os resultados alcançados no trabalho através da descrição dos padrões obtidos e de gráficos. A conclusão do trabalho está no capítulo 5.

2. SISTEMA DE INTELIGÊNCIA EMPRESARIAL

Neste capitulo, são apresentados os conceitos de Inteligência Empresarial, *data warehouses* e mineração de dados necessários para o entendimento e a realização deste trabalho.

2.1 INTELIGÊNCIA EMPRESARIAL

A globalização é caracterizada por um ambiente de extrema competitividade e propenso a mudanças ocasionadas por aspectos sociais, econômicos ou políticos. Em um mercado competitivo, as empresas precisam estar atentas a essas mudanças, e principalmente ter capacidade de reagir eficientemente quando elas ocorrerem. Os níveis de competição a que estão subordinadas e a complexidade do ambiente empresarial sugerem que as empresas busquem estratégias de sustentação e de crescimento, além de que sejam conhecedoras e conscientes do ambiente no qual estão inseridas, e consequentemente, um passo à frente dos seus competidores.

Para que as empresas avancem no mercado competitivo, os profissionais responsáveis pela tomada de decisões devem contar com uma gama de informações que englobem os aspectos envolvidos no cenário empresarial e a organização dos processos de trabalho. Portanto, é importante que a informação certa chegue à pessoa certa no momento certo.

Atualmente, grandes empresas detêm um volume enorme de dados e esses estão em diversos sistemas diferentes espalhados por elas. Converter a grande quantidade de valiosos dados em conhecimento é o trabalho das aplicações conhecidas como Inteligência Empresarial (BI – *Business Intelligence*). A BI é um grupo emergente de aplicações projetado para organizar e estruturar os dados da transação de uma empresa de forma que possam ser analisados a fim de beneficiar as operações e o suporte às decisões da empresa. (KALAKOTA e ROBINSON, 2001).

De acordo com Machado (2004), abaixo da Inteligência Empresarial situam-se

as tecnologias de *Customer Relationship Management* (CRM), *Knowledge Management* (KM) e *Data Warehouse* (DW). O CRM é um sistema integrado de gestão com foco no cliente, que objetiva ajudar as companhias a criar e manter um bom relacionamento de forma inteligente com seus clientes. O CRM baseado em dados dos sistemas transacionais pode gerar uma importante fonte de informações; porém nem sempre isso acontece, pois, estando os dados armazenados separadamente em cada departamento, a sua integração estará impossibilitada. Com a tecnologia de *Data Warehouse* é possível integrar e analisar essas bases de dados. Nas próximas seções são abordados diversos conceitos relacionados a BI.

2.1.1 Sistemas de Apoio a Decisão

Sistemas de Apoio à Decisão são sistemas de informação que, através de informações e modelos especializados, ajudam a resolver problemas organizacionais. Além disso, apóiam o processo de tomada de decisão em áreas de planejamento estratégico, controle gerencial e controle operacional. Os SAD's, não só fornecem informações para apoio à tomada de decisões, mas também contribuem para o processo de tomada de decisão. De acordo com Primak (2008), Sistemas de Apoio à Decisão são “Sistemas de Informação complexos que permitem total acesso à base de dados corporativa, modelagem de problemas, simulações e possuem uma interface amigável. Além disso auxiliam o executivo em todas as fases da tomada de decisão, principalmente nas etapas de desenvolvimento, comparação e classificação dos riscos, além de fornecer subsídios para a escolha de uma boa alternativa”.

Utilizando um SAD é possível aos tomadores de decisão buscar informações em bancos de dados diferentes, mesmo que estejam em lugares distintos. A simulação é outra característica importante num SAD, pois demonstra a probabilidade de algo acontecer através de cenários construídos a partir de decisões tomadas, possibilitando ao gestor uma maior segurança para solucionar o problema.

2.2 DATA WAREHOUSE

Data Warehouses são repositórios de informações que fornecem armazenamento, funcionalidade e capacidade de responder consultas acima das capacidades de bancos de dados orientados por transações. Para Imnon (2002), o *data warehouse* é a base para todo o processamento dos Sistemas de Apoio à Decisão. O trabalho do analista de SAD no ambiente de *data warehouse* se torna imensuravelmente mais fácil que o trabalho de um analista em um ambiente clássico.

De acordo com Elmasri e Navathe (2005),

> Um data warehouse também é uma coleção de informações, bem como um sistema de apoio. Porém, existe uma distinção clara. Os bancos de dados tradicionais são transacionais. Os data warehouses têm a característica distinta de que são direcionados principalmente para aplicações de apoio às decisões. Eles são otimizados para a recuperação de dados, não para o processamento rotineiro de transações.

Contrastando com os bancos de dados múltiplos que proporcionam acesso a bancos de dados disjuntos e geralmente heterogêneos, um *data warehouse* é freqüentemente um armazém de dados integrados oriundos de fontes múltiplas, processados para armazenamento em um modelo transacional. Além disso, *data warehouses* são não-voláteis, já que mudam muito menos freqüentemente e podem ser considerados como não sendo de tempo real.

A seguir são explicitadas as principais propriedades dos *data warehouses*:

- Orientado a assunto - O *data warehouse* é organizado em assuntos que realmente são importantes para a análise, eliminando-se assim, os dados que são úteis apenas para dar suporte aos negócios da empresa. Os *data warehouse* são feitos para responder abordagens sobre certos assuntos como, saber mais sobre as vendas da empresa, ou sobre os resultados das atuações das equipes de marketing em determinadas regiões, então podem ser

respondidas questões sobre certos assuntos como: “Quais foram os melhores clientes em um determinado período?”.

- Integrado - Um *data warehouse* é geralmente construído integrando várias bases de dados. A limpeza, envolvendo conversão de datas para um formato único, resolução de conflitos entre nomes e conversão de medidas, deve ser feita para garantir a consistência. Por exemplo, podemos ter o atributo sexo codificado como M/F em uma fonte de dados, outra como H/M e uma terceira como 0/1, sendo necessária a conversão dos dados para um padrão definido antes da inserção no *data warehouse.*

- Não-volátil - Uma das características do *data warehouse* é a não volatilidade, ou seja, os dados do data warehouse não mudam. As alterações que ocorrem nos bancos de dados operacionais não ocorrem no *data warehouse*, pois são repositórios independentes. As únicas operações que ocorrem no DW são: carga de dados e consultas. Pode ser citada como exemplo a atualização do endereço de um cliente. Em uma determinada data um cliente muda-se de endereço, no sistema de dados operacional isto significa uma atualização nos campos de endereçamento do cliente, no *data warehouse* significa a inclusão de um novo registro.

- Dados Históricos - Os bancos de dados operacionais guardam apenas os dados atuais, que estão constantemente sendo atualizados. O objetivo do *data warehouse* é armazenar informações históricas (de até dez anos ou mais), as quais são precisas em um momento específico e geralmente não mudam.

- Dados detalhados e resumidos - O *data warehouse* armazena dados resumidos que atendem a alta gerência da empresa que necessita de informações mais resumidas, e dados detalhados que atendem às necessidades da baixa gerência, pois ajudam a observar aspectos mais táticos da empresa.

2.2.1 Modelo Multidimensional

O Modelo Entidade-Relacionamento (ER) é um modelo de dados conceitual de alto nível, além de muito popular (ELMASRI e NAVATHE, 2005), que tem por base a percepção de que o mundo real é formado por um conjunto de objetos chamados entidades e pelo conjunto de relacionamento entre esses objetos. Já a construção do *data warehouse* é baseado em um modelo multidimensional, o qual é apropriado para o processamento analítico on-line e orientado a assunto. Os modelos multidimensionais são formados por:

- **Medidas -** representam dados numéricos tais como total de vendas, lucro, unidades compradas, etc.
- **Dimensões -** são as diferentes perspectivas pelas quais os dados podem ser analisados.
- **Tabela de Fatos -** contém medidas e chaves para o relacionamento entre as dimensões.

Temos três tipos de modelos multidimensionais:

- Estrela - Esse modelo é formado por uma tabela central – Tabela de Fatos, ligada às suas várias dimensões. Na Figura 1, temos a tabela de fatos Vendas, com as medidas *qt_vendida* e *valor_vendido*, as quais podem ser analisadas pelas dimensões Produto, Vendedor, Tempo e Loja.

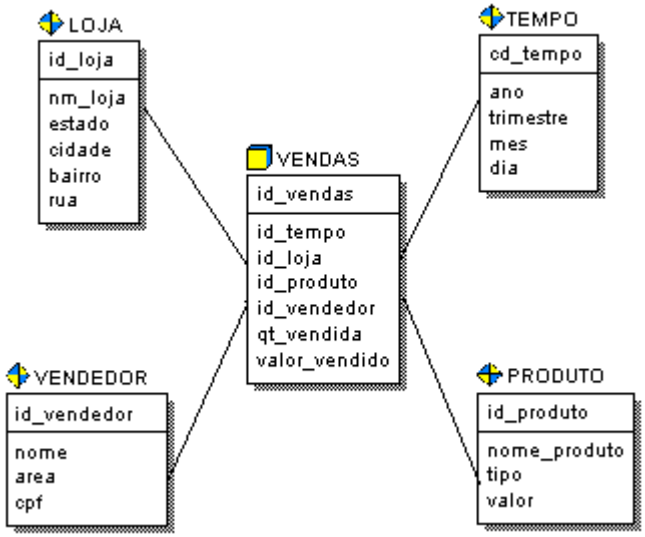

Figura 1: Modelo Estrela

- Floco de neve - O modelo floco de neve é uma variação do esquema estrela, no qual algumas dimensões são normalizadas, dividindo-as em outras tabelas adicionais. A vantagem dos dados normalizados é a economia de espaço em disco para armazenamento, entretanto, a consulta torna-se mais complexa, pois precisa executar a união das várias tabelas envolvidas. Na Figura 2, temos a dimensão Loja que foi normalizada, dando origem à dimensão Endereço e a dimensão Produto que deu origem à dimensão Marca.

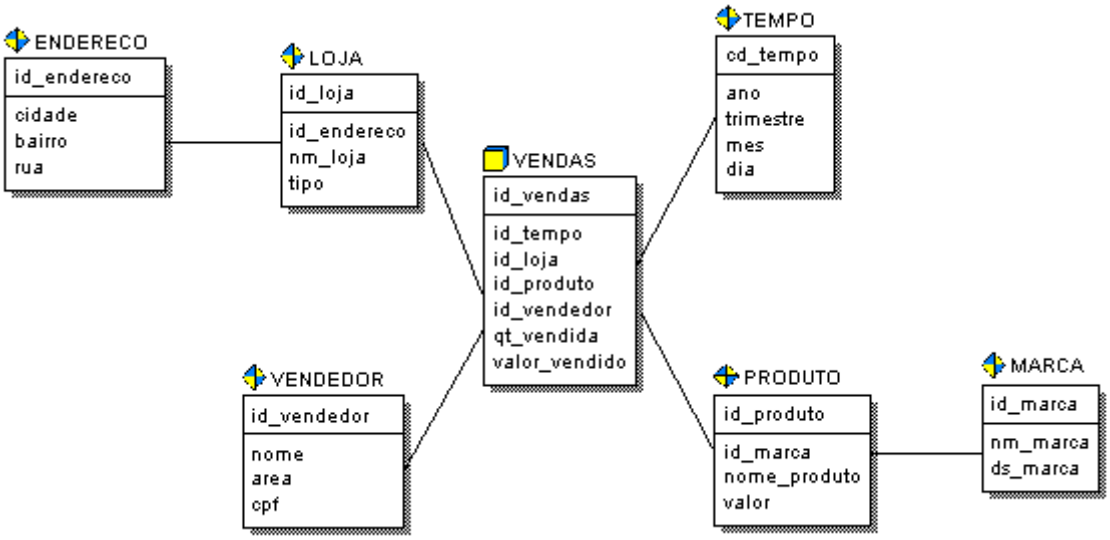

Figura 2: Modelo Snowflake

- Constelação - Nesse modelo multidimensional as tabelas de fatos podem compartilhar dimensões. Na Figura 3, temos duas tabelas de fatos, Vendas e Compras, que compartilham as dimensões Produto e Tempo.

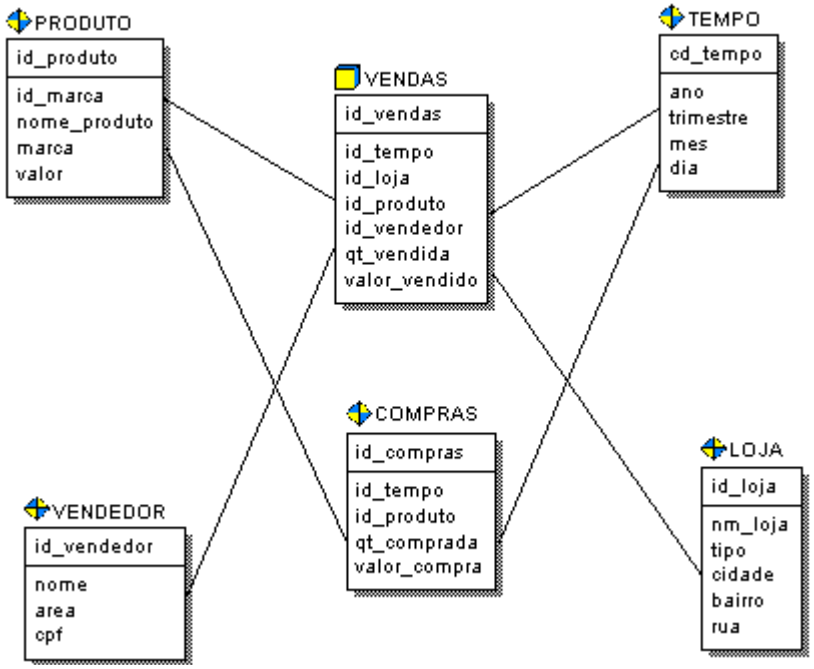

Figura 3: Modelo Constelação

A construção de *data warehouse* pode ser vista como uma importante etapa de pré-processamento para mineração de dados. Portanto, data warehouses fornecem ferramentas OLAP (seção 2.3) para a análise interativa de dados multidimensionais de granularidades variadas, o que facilita a mineração de dados efetiva.

2.3 OLAP

O processamento analítico on-line (On-Line Analytical Processing) ou simplesmente OLAP pode ser definido como:

> "uma tecnologia de software que permite aos analistas, gerentes e executivos obterem os dados de uma forma rápida, consistente e com acesso interativo para uma grande variedade de possíveis visões da informação na empresa. Mais sucintamente, OLAP é um conjunto de funcionalidades que tem como principal objetivo facilitar a análise multidimensional" (INMON, 1999).

Assim, os usuários responsáveis pela tomada de decisão podem realizar pesquisas a partir de várias perspectivas diferentes através de ferramentas que possibilitam fazer previsões sobre determinados cenários, sintetizar informações corporativas por meio de visões personalizadas, realizar análises históricas, dentre outras funcionalidades. Para alcançar essa visualização, os dados são apresentados em termos de medidas e dimensões, ou seja, sobre um modelo de dados dimensional. A maneira mais popular de se entender o mundo dimensional é utilizando-se da idéia de um cubo, que é uma metáfora para se referir como as dimensões desse modelo estão organizadas (MACHADO, 2004). Um cubo é composto por fatos, dimensões e medidas, que foram explicados na sessão anterior.

Existem muitas ferramentas OLAP no mercado, cada uma com propósito diferente tornando complicada a aquisição de um produto OLAP, pois é necessário levar em consideração a funcionalidade, a arquitetura, interfaces e o impacto sobre a organização e todos esses requisitos devem envolver tanto os gerentes de TI quanto os usuários finais. Vale lembrar que a escolha de um produto inadequado leva a algumas conseqüências como prejuízos financeiros para aquisição de software, no treinamento de pessoas para usar uma ferramenta que apresenta benefícios temporários e imperceptíveis, pode acontecer falhas no projeto fazendo com que aconteça perda de credibilidade para conclusão do projeto, já que esse frustrou as expectativas dos executivos, os maiores interessados na informação.

De modo geral, as ferramentas OLAP permitem ao usuário realizar algumas operações no cubo dimensional, as principais para Kimball e Ross (2002) são: *Drill down*, *Drill up*, *Drill across*, *Slice and Dice* e *Pivoting*.

2.4 MINERAÇÃO DE DADOS

Para Elmasri e Navathe (2005), a área de interesse popular conhecida como mineração de dados “se refere à mineração ou à descoberta de novas informações em função de padrões ou regras em grandes quantidades de dados. Por ser útil, na prática, a mineração precisa ser realizada eficientemente em grandes arquivos e bancos de dados. Atualmente ela não possui uma boa integração com os sistemas

gerenciadores de bancos de dados".

De acordo com Witten e Frank (2005), a mineração de dados trata da resolução de problemas através da análise de dados já existentes no banco de dados. Como exemplo, pode ser citado o problema da infidelidade instável de clientes em um mercado altamente competitivo. Um banco de dados das escolhas dos clientes, juntamente com os perfis de clientes, detém a chave para esse problema. Padrões de comportamento dos clientes antigos podem ser analisados para identificar características distintas entre os clientes cuja tendência é mudar a preferência de produtos e os que têm probabilidade de permanecer fiéis.

A mineração de dados é um campo interdisciplinar, a confluência de um conjunto de disciplinas, incluindo sistemas de banco de dados, estatísticas, aprendizado de máquina, visualização e ciência da informação (HAN e KAMBER, 2001). Além disso, dependendo da abordagem de mineração de dados utilizada, técnicas de outras disciplinas podem ser aplicadas, tais como redes neurais, *fuzzy*, representação do conhecimento, programação lógica indutiva, ou computação de alto desempenho.

Devido à diversidade de disciplinas que contribuem para a mineração de dados, espera-se que as pesquisas na área gerem uma grande variedade de sistemas de mineração. Portanto, é necessário estabelecer uma clara classificação dos dados dos sistemas de mineração.

A seguir são descritas as funcionalidades da mineração de dados.

2.4.1 Funcionalidades da Mineração de Dados

Quando se aplica a mineração, busca-se extrair informações para atingir determinados objetivos. São vários os tipos de informações que podemos obter através da mineração de dados, dentre os quais podemos citar:

- Associações - Acontecem quando as ocorrências estão ligadas a algum evento. Temos um clássico exemplo de uma grande rede varejista americana que descobriu, através da mineração de dados, que as vendas de fraldas

estavam intimamente ligadas às vendas de cervejas, pois os pais que saíam para comprar fraldas, compravam cerveja também.

- Classificação - Encontra modelos que descrevem e distinguem classes ou conceitos para previsão futura. Pode também ajudar a encontrar perfis e características dos clientes. Com isto, providenciar um modelo, utilizado para prever suas ações e desejos. Também pode ajudar, por exemplo, a determinar os tipos de promoções que são mais eficientes, para manter determinados tipos de clientes, procurando direcionar melhor os gastos necessários para mantê-los.

- Previsões - O tipo mais comum de uso da mineração de dados é a previsão. Com ela podemos antever, por exemplo, se um cliente irá renovar uma assinatura, se ele irá comprar um determinado tipo de produto, e baseado em padrões, estimar o valor futuro de variáveis contínuas: número de vendas, porcentagem de lucro, entre outros.

- Análise de Cluster - Ao contrário da classificação, cujos grupos são conhecidos previamente, na análise de cluster eles são desconhecidos. A formação desses grupos é feita com base na similaridade dos dados comparados aos demais.

- Análise de exceções - Uma base de dados pode conter objetos que não concordam como o comportamento geral ou modelo dos dados. A aplicação da mineração descobre dados que estão fora do padrão, mais conhecidos como ruídos ou exceções.

Neste trabalho, foram utilizadas técnicas de mineração por classificação, pelo fato de que, através da classificação são encontrados modelos e características para classes já definidas. Técnicas de mineração por associação foram utilizadas também, para encontrar atributos que estivessem associados. Nas seções seguintes, as técnicas de mineração de dados por classificação e por associação são descritas com mais detalhes.

2.4.2 Técnicas de Classificação

Classificação, que é a tarefa de organizar objetos em uma entre diversas categorias pré-definidas, é um problema universal que engloba muitas aplicações diferentes. Exemplos incluem a detecção de mensagens de spam em e-mails baseada no cabeçalho e conteúdo da mensagem, a categorização de células como malignas ou benignas baseada nos resultados de varreduras MRI (Ressonância Magnética por Imagem) e a classificação das galáxias baseada nos seus formatos. (TAN, STEINBACK e KUMAR, 2009)

O processo de classificação é realizado em duas etapas. Na primeira etapa, representada na Figura 4, é construído um modelo que descreve um conjunto pré-determinado de classes de dados ou conceitos. O modelo é construído através da análise de linhas do banco de dados descritas por atributos. Cada linha é assumida como pertencendo a uma classe predefinida, como determinada por um dos atributos, chamado **atributo rótulo da classe**. As linhas analisadas para construir coletivamente o modelo formam o **conjunto de treinamento**. As linhas individuais que compõem o conjunto de treinamento são referenciadas como amostras de treinamento e são selecionadas aleatoriamente da população estudada. Desde que o rótulo de classe de cada amostra de treinamento é fornecido, esta etapa também é conhecida como aprendizado supervisionado. Contrasta com o aprendizado não-supervisionado (ou clustering), em que os rótulos de classe das amostras de treinamento não são conhecidas, e o número ou conjunto de classes a serem aprendidas não pode ser conhecido antecipadamente.

Na segunda etapa, como pode ser visto na Figura 5, o modelo é usado para classificação. O método de validação é uma técnica simples que utiliza um **conjunto de teste** de classe rotulada. As amostras de teste são selecionadas aleatoriamente e são independentes das amostras de treinamento. A precisão de um modelo em um conjunto de ensaio em questão é o percentual de amostras que estão definidos e corretamente classificados pelo modelo. Para cada amostra, o rótulo de classe conhecido é comparado com a previsão do modelo de classe aprendido. (HAN e KAMBER, 2001)

Normalmente, o modelo de aprendizagem é representado sob a forma de árvores de decisão, regras de classificação ou fórmulas matemáticas.

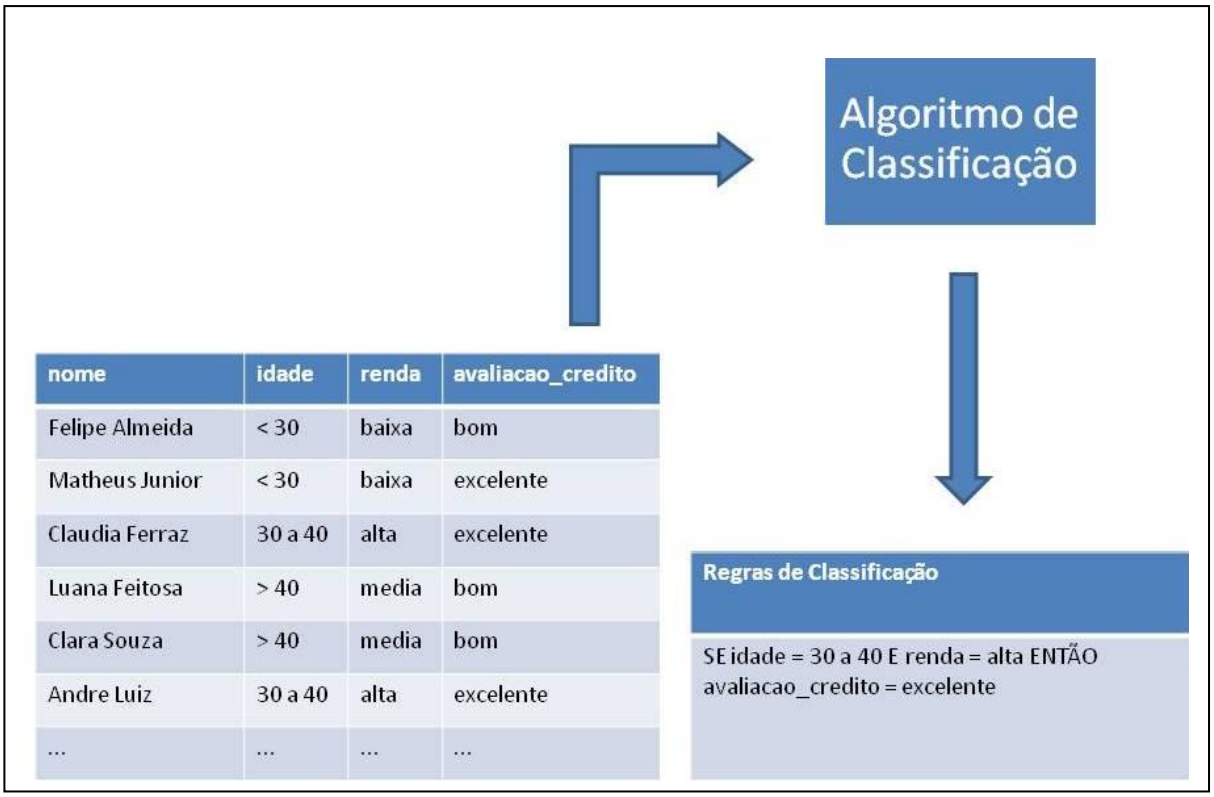

Figura 4: Etapa de criação do modelo de classificação

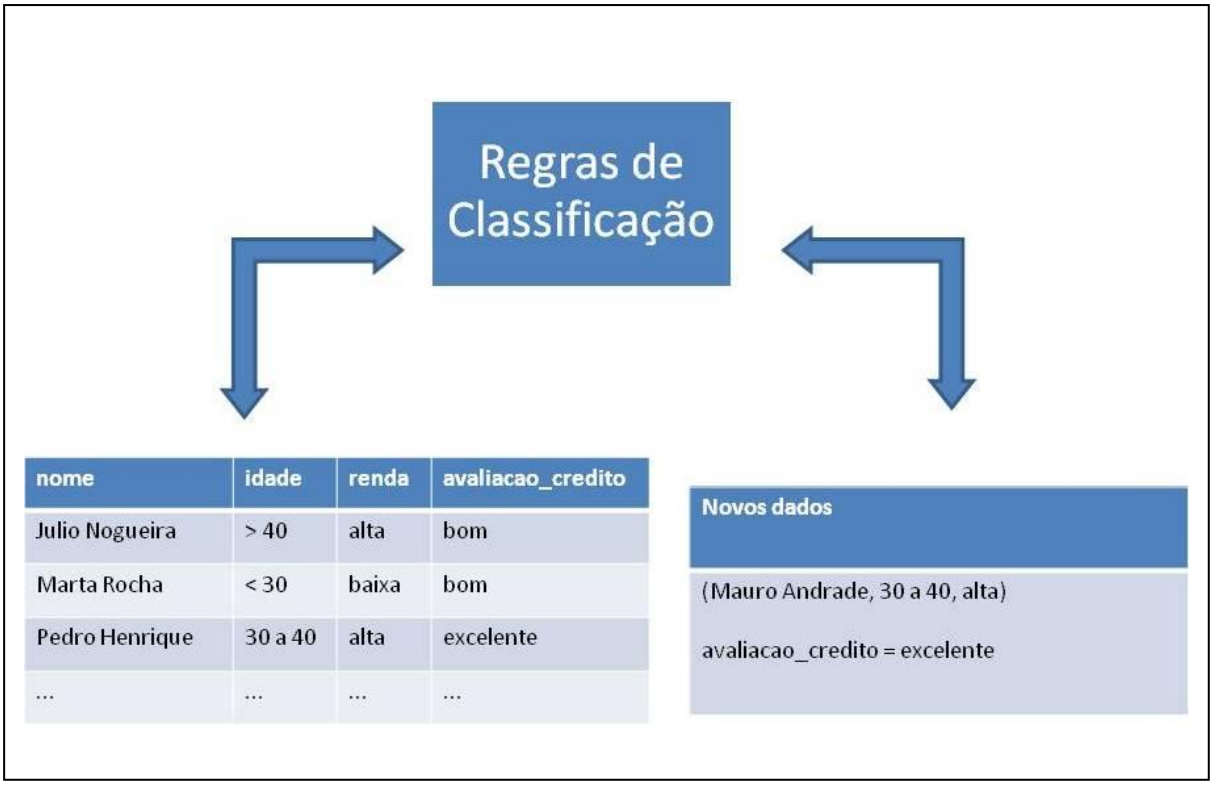

Figura 5: Etapa de classificação utilizando o conjunto de teste

Árvores de Decisão

Árvores de decisão funcionam como uma técnica de classificação simples, porém muito utilizada. Um problema de classificação pode ser resolvido fazendo uma série de questões cuidadosamente organizadas sobre os atributos do registro de teste. Cada vez que recebemos uma resposta, uma questão seguinte é feita até que cheguemos a uma conclusão sobre o rótulo da classe do registro. A série de questões e suas respostas possíveis podem ser organizadas na forma de uma árvore de decisão, com sua estrutura hierárquica consistindo de nós e arestas direcionadas. (TAN, STEINBACK e KUMAR, 2009).

A Figura 6 mostra um exemplo de árvore de decisão. Através da árvore fictícia, percebe-se, por exemplo, que os candidatos que cursaram o Ensino Médio em escola pública e têm filhos, tendem a ser aprovados.

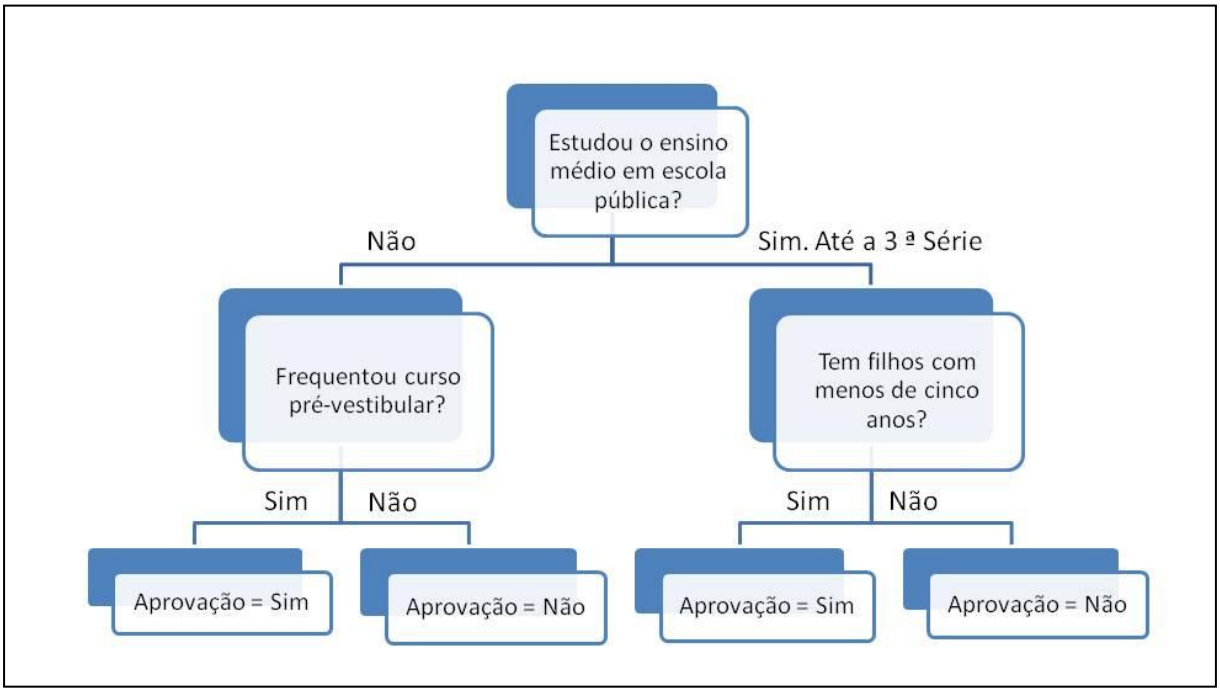

Figura 6: Exemplo de árvore de decisão para dados de aprovação em vestibular

A seguir, são apresentados os algoritmos de classificação utilizados no trabalho:

- O algoritmo *J48* (*weka.classifiers.trees.J48*) é uma implementação do algoritmo

C4.5 release 8, que gera árvore de decisão e é considerado o mais popular algoritmo do *Weka*. O *J48* constrói um modelo de árvore de decisão baseado num conjunto de dados de treinamento, sendo que esse modelo é utilizado para classificar as instâncias do conjunto de teste (ALMEIDA, 2003). O *J48* constrói árvores de decisão a partir de um conjunto de dados de treinamento rotulados com o conceito de entropia[1] da informação. Ele usa o fato de que cada atributo dos dados pode ser usado para tomar uma decisão, dividindo os dados em subconjuntos menores. *J48* examina o ganho de informação normalizada (diferença de entropia) que resulta da escolha de um atributo para dividir os dados. Para tomar a decisão, o atributo com maior ganho de informação é utilizado. Então o algoritmo repete nos subconjuntos menores. O processo de divisão é interrompido se todas as ocorrências em um subconjunto pertencem à mesma classe. Em seguida, um nó folha com a classe selecionada é criado na árvore de decisão. (QUINLAN, 1993)

- O algoritmo *PART (weka.classifiers.rules.PART)* é uma variação do *J48*, que constrói regras a partir da árvore de decisão. O processo de geração de regras para classificação de sistemas normalmente atua em dois estágios: Regras são induzidas inicialmente e posteriormente refinadas. Isto é feito através de dois métodos, através da geração da árvore de decisão e posteriormente o mapeamento da árvore de decisão em regras aplicando processos de refinamento, ou pela utilização do paradigma dividir-para-conquistar. Witten e Frank (2005) combinam estas duas aproximações no algoritmo *PART*, que trabalha construindo a regra e estimando sua cobertura como no processo de dividir-para-conquistar repetidamente até que todas as instâncias estejam cobertas. Constrói uma árvore de decisão parcial em cada interação e converte os ramos com a mais alta cobertura em regras.

- O *JRip* (*weka.classifiers.rules.JRip*) ou *Ripper* (*Repeated Incremental Pruning to Produce Error Reduction* - Poda Incremental Repetida para Produzir Redução de Erro), consiste em um aprendizado de regra proposicional, que foi proposto por

[1] Na teoria da informação, mede o grau de desordem de um conjunto de dados. Este conceito é utilizado para encontrar o próximo melhor atributo de um dado para ser utilizado como nó de uma árvore de decisão.

William W. Cohen como uma versão otimizada do *IREP* (*Incremental Reduced Error Pruning*). O método *IREP* utiliza árvores de decisão e as simplifica pela redução do erro, com um algoritmo que trabalha a técnica dividir-para-conquistar. Depois que uma regra é encontrada, todos os exemplos que são cobertos por ela são deletados. Um caminho para melhorar a abordagem incremental do IREP é adiar o processo de produção de regras deste método, assim esse método se aproxima do método de poda pelo erro, forma de otimização pelo qual o algoritmo *JRip* é conhecido. (Cohen, 1995)

2.4.3 Técnicas de Associação

Muitas empresas acumulam enormes quantidades de dados das suas operações do dia-a-dia. Por exemplo, grandes quantidades de dados de compras de clientes são juntadas diariamente nos balcões das mercadorias. A Tabela 1 ilustra um exemplo desses dados, conhecidos comumente como transações de cestas de compras. Cada linha nesta tabela corresponde a uma transação, que contém um identificador único rotulado como *TID* e um conjunto de itens comprados por um determinado cliente (TAN, STEINBACK e KUMAR, 2009). Utilizando algoritmos de associação sobre os dados pode-se chegar a regras de associação que definem o comportamento de compra dos clientes.

TID	Itens
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Cola}
4	{Pão, Fraldas, Leite, Cerveja}
5	{Pão, Leite, Fraldas, Cola}

Tabela 1: Cestas de compras

Regra de Associação

Segundo Tan, Steinback e Kumar (2009), uma regra de associação pode ser definida como uma expressão de implicação no formato X → Y, onde X e Y são conjuntos disjuntos de itens, isto é, X ∩ Y = ∅. A força de uma regra de associação pode ser medida em termos do seu **suporte** e sua **confiança**. O suporte determina a freqüência na qual uma regra é aplicável a um determinado conjunto de dados, enquanto que a confiança determina a freqüência na qual os itens em Y aparecem em transações que contenham X. Como exemplo, temos a regra de associação que pode se obtida na cesta de compras da Tabela 1: {Fraldas → Cerveja}, o que significa que os clientes que têm fraldas em suas cestas de compras tendem a levar cerveja também. O suporte de tal regra é igual a 60%, já que 3/5 dos registros tem fraldas e cerveja, e a confiança é igual a 75%, pois entre os 4 registros que possuem fraldas 3 possuem cerveja.

Logo abaixo temos a descrição do algoritmo de associação utilizado no trabalho:

- O algoritmo *Apriori* (*weka.associations.Apriori*), proposto por Rakesh Agrawal e Ramakrishnan Srikant, é o mais utilizado para a descoberta de regras de associação. Para isto, o algoritmo executa múltiplas passagens sobre o banco de dados de transações. Na primeira passagem é contado o suporte de cada item, e os que têm o suporte individual maior que o suporte mínimo são considerados como freqüentes. Em cada uma das passagens subseqüentes, é efetuada a contagem do suporte dos itens candidatos, verificando se é maior que o suporte mínimo. Como método de otimização do tempo de execução, o algoritmo *Apriori* parte do princípio de que se X ⊂ Y, e X não é freqüente, logo Y também não é freqüente. Deste modo, a cada nova passagem sobre o banco de dados, o algoritmo não precisa ler novamente todo o banco, mas somente o conjunto de itens candidatos selecionados na parte anterior. O algoritmo *Apriori* pode trabalhar com um número grande de atributos, gerando várias alternativas combinatórias entre eles. São realizadas buscas sucessivas em toda a base de dados, mantendo um ótimo desempenho em termos de tempo de processamento. (AGRAWAL e SRIKANT, 1994)

3. MINERAÇÃO DE DADOS EM UM AMBIENTE ACADÊMICO

Este capítulo fornece informações referentes ao *data mart* Acadêmico e às diversas alterações que foram necessárias, além da descrição da principal ferramenta utilizada e da etapa de pré-processamento realizado antes da aplicação dos algoritmos de mineração de dados.

As universidades estão cada vez mais investindo em técnicas de mineração. Torna-se importante conhecer seus clientes estudantes, atraí-los e mantê-los. Para isso as universidades vêm utilizando esses sistemas para planejar estratégias de marketing, planejar a utilização das receitas para os projetos de extensão, evitando superávit e déficit no orçamento e aplicando de maneira mais produtiva as receitas da universidade (KIMBALL e ROSS, 2002). Como exemplos de aplicação da mineração de dados em universidades, podemos citar:

- A Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ), utilizando as técnicas de mineração de dados, depois de examinar milhares de alunos forneceu a seguinte regra: se o candidato é do sexo feminino, trabalha e teve aprovação com boas notas, então não efetiva matrícula. Seria muito difícil obter regras como esta, sem estar utilizando a mineração.
- Na Universidade Federal de Minas Gerais (UFMG), as técnicas de mineração vêm sendo utilizadas na determinação do perfil dos alunos, com base nos dados de pesquisa econômico-social preenchidos quando da admissão. Essas informações vêm sendo correlacionadas com o desempenho dos alunos no vestibular e mesmo durante o curso de graduação. (CESAR, 2000)
- A base de dados do Vestibular de 2008 da Universidade Federal de Santa Maria (UFSM) foi utilizada com o propósito de identificação de padrões e conhecimento. Na base de dados foi possível deparar com informações sobre questões, provas, cotas, naturalidade dos candidatos e respostas, nas quais técnicas de mineração de dados puderam ser aplicadas. (LOPES e PRASS, 2009)

Faz-se necessária a aplicação de técnicas de mineração no ambiente acadêmico da Universidade Estadual do Sudoeste da Bahia, visando aumentar a eficiência nas campanhas de vestibular e conhecer os candidatos a vagas nos

cursos da universidade. A seguir, serão descritas as fases para o desenvolvimento do trabalho.

3.1 O *DATA MART* ACADÊMICO DA UESB

O potencial de mineração de dados pode ser melhorado se os dados apropriados tiverem sido coletados e armazenados em um *data warehouse.* Em trabalhos de conclusão de curso anteriores, “fez-se necessária a construção de um *data mart* que permitisse a visualização dos dados do questionário sócio-cultural da Universidade Estadual do Sudoeste da Bahia sobre vários pontos de vista. Através da utilização de ferramentas de processamento analítico online (OLAP) foi possível a realização de consultas e análises complexas na base de dados, permitindo a definição do perfil do candidato ao vestibular” (SANTOS, L. F. D. S., 2010). Este trabalho utilizou o *data mart* Acadêmico da UESB como ponto de partida para a etapa de pré-processamento de dados.

Para facilitar a compreensão, serão descritas de forma genérica informações acerca do questionário sócio-cultural e do *data mart*, seguidas da etapa de pré-processamento dos dados.

Para participação de um candidato no concurso vestibular da UESB, é necessário o preenchimento de um formulário informando dados pessoais para efetivação da inscrição, além do preenchimento do questionário sócio-cultural. Ao fim desse processo, é gerado um boleto para pagamento da taxa de inscrição e informado um número de protocolo para confirmação da inscrição. Esse sistema é muito importante para a instituição, pois agiliza o processo de inscrição. Porém, o sistema não permite a realização de relatórios simples com informações estratégicas como o perfil do aluno em determinado ano ou semestre, participação na renda familiar, meios de transporte e alimentação a serem utilizados durante o curso superior, etc. Com esta finalidade, o *data mart* Acadêmico da UESB foi construído.

A Figura 7 ilustra o modelo multidimensional resultante do trabalho realizado em 2009, com os fatos, as medidas e as dimensões estabelecidas no esquema estrela, que consiste em um fato ligado por junções às dimensões de análise.

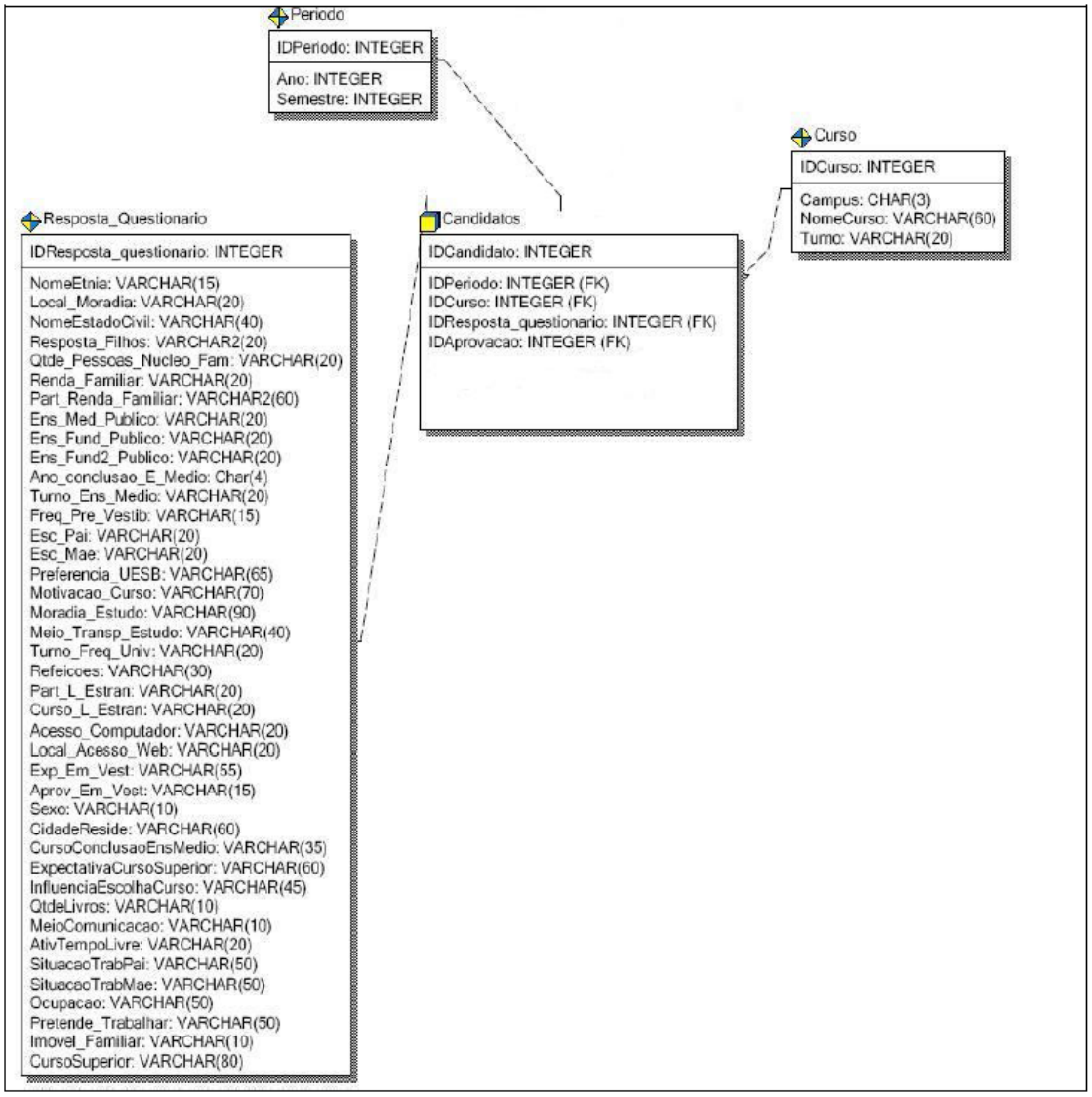

Figura 7: Modelo Multidimensional do *Data Mart* acadêmico da UESB. (SANTOS, L. F. D. S., 2010)

Visando a facilidade para obtenção dos dados em *ARFF (Attribute-Relation File Format),* formato nativo de conjunto de dados utilizado no *Weka* (ferramenta de mineração descrita na seção 3.3), a base de dados do modelo multidimensional foi reestruturada. A primeira alteração foi a migração do Sistema de Gerenciamento MySql para PostgreSQL, que é o SGBD utilizado atualmente pela instituição, e traz vantagens como boa performance, ser multi-plataforma, integridade referencial e suporte nativo a transações.

Destacam-se também entre as mudanças a inclusão na tabela *Resposta_Questionario,* de uma coluna categorizada – *aprovacao* – cujo domínio,

{SIM, NÃO}, define se o candidato obteve aprovação no concurso vestibular ao qual aquele questionário está associado.

É bom ressaltar que quando for utilizado o termo aprovação como positiva, engloba apenas os candidatos que foram convocados em primeira chamada. Com aprovação igual a Não, estão todos os candidatos referidos como classificados (mas não convocados) e reprovados nos livros de Classificação Geral de Candidatos por Curso mantidos na Comissão Permanente de Vestibular (COPEVE).

Para que sejam aplicadas técnicas de mineração por classificação, objetivando que um candidato com um determinado perfil definido pelas suas respostas no questionário sócio-cultural seja classificado como aprovado ou não-aprovado, é indispensável a utilização das listas de aprovação dos concursos de vestibular da universidade. Infelizmente, a COPEVE não dispõe de bancos de dados ou arquivos de texto contendo a lista de convocados pela instituição. Os únicos meios possíveis de obtenção da relação de convocados foram através do site da UESB e do site da Consultoria em Projetos Educacionais e Concursos (CONSULTEC).

As listas de aprovados nos processos seletivos tiveram de passar, individualmente, por extensos processos de filtragem manual. No *Microsoft Office Word 2007*, através de procedimentos de Localização – Substituição, e utilizando localizações especiais de letras e alguns caracteres, foram eliminados os nomes dos estudantes, cursos, e mantidos apenas números de inscrição, RG e códigos de cursos. Por meio de funcionalidades do *Microsoft Office Excel 2007*, como a conversão Texto para Colunas e os Filtros, foi possível, respectivamente, definir com exatidão as colunas do número de inscrição e RG, e ordenar os registros para facilitar a remoção dos códigos de cursos (representados por números muito menores que os de inscrição). Além do mais, foram acrescentadas, no Excel, as colunas *ano* e *semestre,* para resolver o problema referente ao fato de que o mesmo candidato pode ter tentado o vestibular mais de uma vez, levando em conta que o número de inscrição não foi acrescentado no *data mart*. Sem o número de inscrição, a única maneira possível de diferenciação dos candidatos seria pelo RG, mas poderiam ser encontradas duplicações caso o candidato tenha participado por mais de uma vez. A Tabela 2 contém alguns registros de aprovados no vestibular 2007.1

após os processos de filtragem. O arquivo foi salvo no formato *CSV*[2] (*Comma-Separated Values*), e convertido para *SQL* objetivando a criação de uma tabela que seria utilizada em uma função que definiria, na tabela *Resposta_Questionario,* os alunos aprovados.

Inscrição	RG	Ano	Semestre
103240	839612753	2007	1
103247	1169037593	2007	1
103276	998690473	2007	1
103282	1156081840	2007	1
103298	1133145078	2007	1

Tabela 2: Primeiros registros de aprovados no vestibular 2007.1

Dessa forma, a tabela *aprovados* foi acrescentada no banco de dados. A tabela poderia ter sido criada como temporária, mas foi mantida, pois pode facilitar trabalhos futuros que venham a utilizar a mesma base de dados. A figura 8 apresenta a base de dados reestruturada.

3.2 PRÉ-PROCESSAMENTO DE DADOS

Nessa etapa são descritas a atribuição de uma variável de aprovação a cada candidato e as demais etapas de pré-processamento necessárias para a obtenção dos dados no formato nativo do *Weka*, o *ARFF*.

A forma encontrada para associar cada registro da tabela *resposta_questionario* à aprovação foi a atualização da coluna *aprovacao* de todos os registros como *NÃO*, e a execução de uma função, *definirAprovados()* utilizando a ferramenta *SQL Manager 2007 for PostgreSQL.* Tal função visa atribuir *SIM* ao

[2] CSV é um formato de arquivo que armazena dados tabelados. É uma implementação particular de arquivos de texto separados por um delimitador, que usa a vírgula e a quebra de linha para separar os valores.

campo *aprovação* de todos os registros de respostas do questionário associados aos RG's da tabela *aprovados*. O código da função encontra-se no Anexo 1.

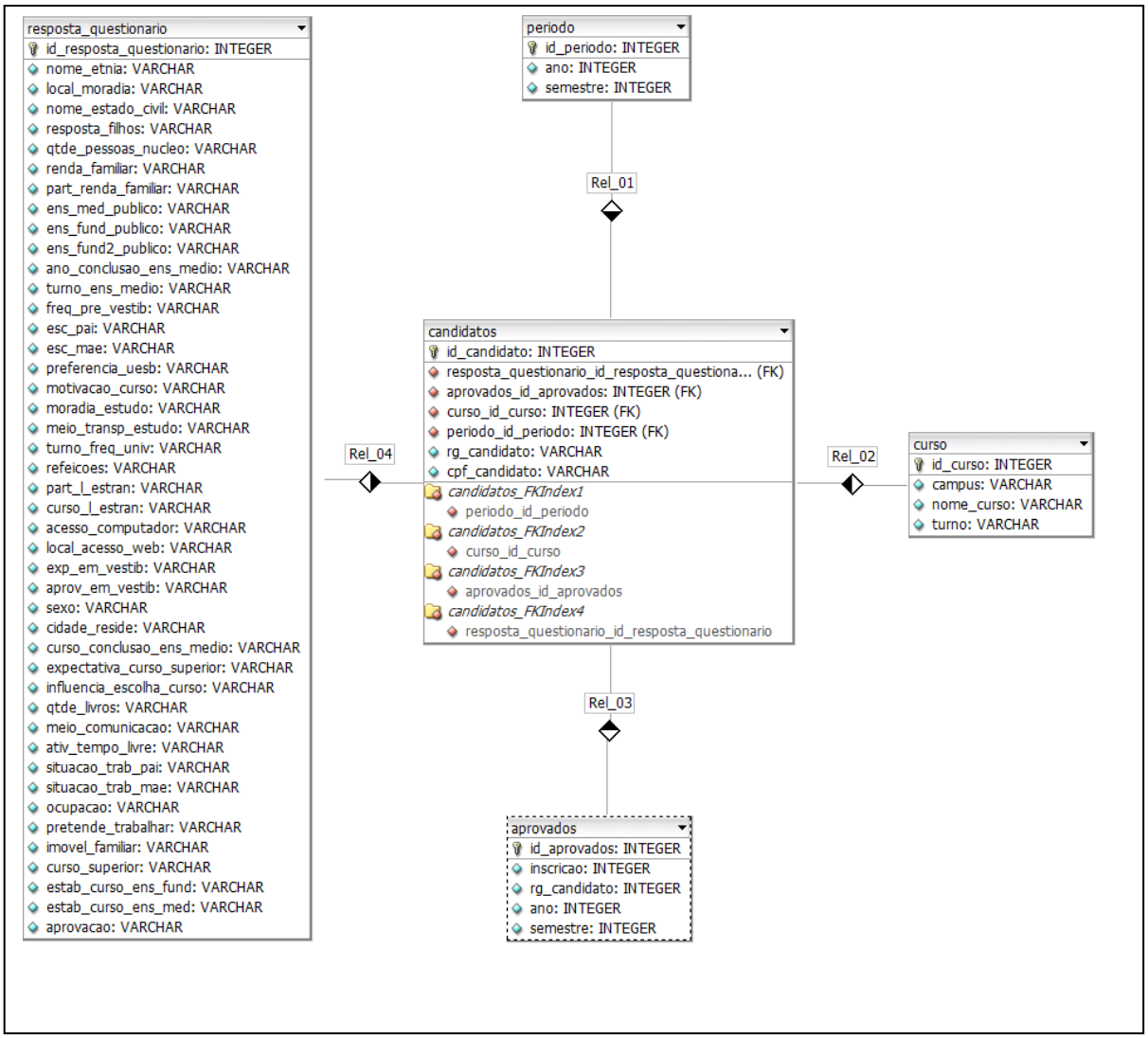

Figura 8: Modelo Multidimensional Reestruturado

Outra característica presente no *data mart* e que precisava de atenção especial era a grande quantidade de valores nulos. Analisando apenas um atributo escolhido, *renda_familiar*, foi constatado que 35,13% dos registros tinham valor nulo para tal atributo. Verificando todos os atributos, observou-se que 40,53% dos registros possuíam pelo menos um valor nulo.

O maior problema na construção de *data warehouses* é a qualidade dos

dados. Para evitar o princípio de GIGO[3] (*garbage in garbage out*), os dados devem ter valores nulos mínimos, porque isso afeta os resultados da mineração de dados.

Para qualquer tipo de atributo, usualmente existe também um símbolo importante que significa **desconhecido**, ou seja, a ausência de um valor para aquele atributo. O principal símbolo utilizado nesses casos em aprendizado de máquina é o ponto de interrogação (?). No entanto, devido ao alto percentual de valores nulos, o símbolo não foi utilizado, pois poderia gerar regras como: se *nome_etnia* = ? e *renda_familiar* = ? então *aprovacao* = não. Por esse motivo, foram selecionados apenas os registros que tivessem todos os seus atributos definidos. Para minimizar o número de registros que seriam descartados, foram feitas 3 seleções diferentes de dados: Perfil Sócio-econômico, Perfil Educacional e Perfil de Expectativas. Essa escolha facilitou também carregar os arquivos no padrão *ARFF* na interface *Explorer* do *Weka.* Na Tabela 3 podem ser vistos os atributos pertencentes a cada perfil.

Perfil Sócio-Econômico	**Perfil Educacional**	**Perfil de Expectativas**
nome_etnia	*ens_med_publico*	*preferencia_uesb*
local_moradia	*ens_fund_publico*	*motivacao_curso*
nome_estado_civil	*turno_ens_medio*	*moradia_estudo*
resposta_filhos	*freq_pre_vestib*	*meio_transp_estudo*
qtde_pessoas_nucleo	*part_l_estran*	*turno_freq_univ*
renda_familiar	*curso_l_estran*	*refeicoes*
part_renda_familiar	*acesso_computador*	*exp_em_vestib*
aprovacao	*aprovacao*	*aprov_em_vestib*
		aprovacao

Tabela 3: Atributos que compõem os perfis

Devido à grande quantidade de inscritos para as poucas vagas disponíveis estava ocorrendo a **prevalência de classe**[4]. A discrepância entre o número de aprovados e o de reprovados, demonstrada na Figura 9, dificultaria a obtenção de

[3] Lixo que entra, lixo que sai é um axioma da informática que ressalta que se dados incorretos forem submetidos a processamento, o resultado serão dados igualmente incorretos.
[4] Um ponto importante do aprendizado de máquina que se refere ao desbalanceamento de classes em um conjunto de dados. A prevalência é indesejável quando as classes minoritárias possuem uma informação muito importante.

padrões referentes ao perfil dos estudantes que ingressaram na universidade. Para sanar o problema da prevalência, foi utilizado um conceito de Estatística (disciplina contida na mineração de dados) muito útil em técnicas de mineração, o **método de amostragem**[5]. Tendo como meta a obtenção de padrões referentes ao perfil de aprovados, todos os registros com esta característica foram mantidos. Foi selecionada uma amostra aleatória de reprovados, permitindo estimar uma medida verdadeira. Na Figura 10 pode ser observado o balanceamento entre as classes para o Perfil Sócio-Econômico. Nos outros perfis também houve aplicação de método de amostragem, mantendo o número de aprovados igual ao de reprovados.

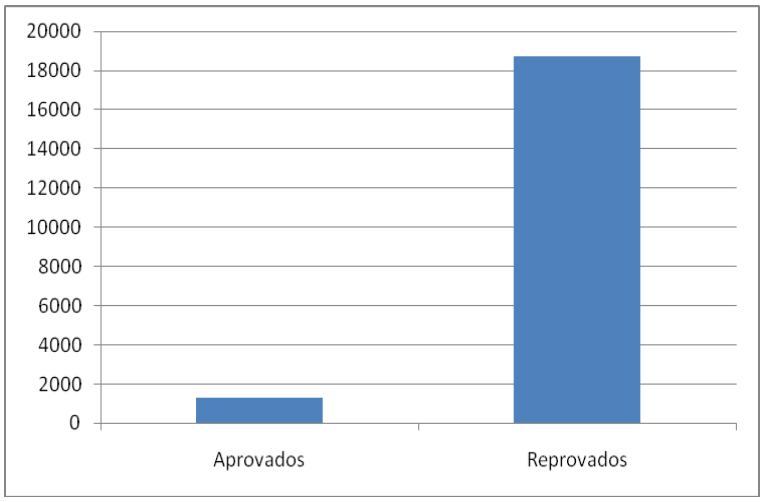

Figura 9: Ocorrência de prevalência de classe com números do Perfil Sócio-Econômico

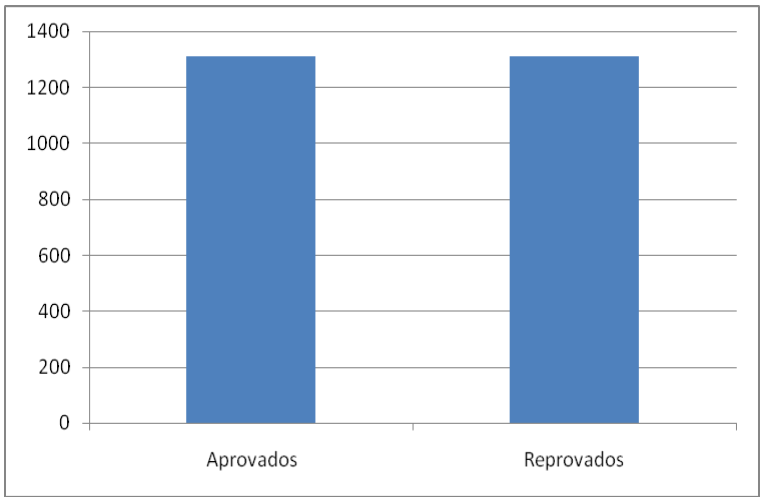

Figura 10: Perfil Sócio-Econômico após a aplicação do método de amostragem

[5] Seleção de amostras, que pode ocorrer de duas formas: aleatória e não aleatória. No método aleatório cada elemento tem uma probabilidade igual (e não nula) de ser selecionado do Universo.

A Figura 11 demonstra os atributos gerados no padrão *ARFF* para o Perfil Sócio-Econômico. Os outros perfis seguiram os mesmos princípios

```
@relation perfil1

@attribute "nome_etnia" { "BRANCO" , "PARDO" , "PRETO" , "AMARELO" ,
"INDÍGENA" }
@attribute "local_moradia" { "ZONA URBANA - CENTRO" , "ZONA URBANA -
PERIFERIA" , "ZONA RURAL" }
@attribute "nome_estado_civil" { "SOLTEIRO" , "CASADO" , "EM UNIÃO
ESTÁVEL/UNIÃO CIVIL" , "DIVORCIADO" , "VIÚVO" }
@attribute "resposta_filhos" { "NÃO" , "SIM - DE 0 A 3 ANOS" , "ACIMA DE
3 ANOS" }
@attribute "qtde_pessoas_nucleo" { "DE 3 A 4 PESSOAS" , "DE 5 A 7
PESSOAAS" , "OUTROS/NÃO RESPONDEU" , "ATÉ 02 PESSOAS" }
@attribute "renda_familiar" { "DE 1 A 2 SALÁRIOS MÍNIMOS" , "DE 6 A 10
SALÁRIOS MÍNIMOS" , "ATÉ 1 SALÁRIO MÍNIMO" , "DE 11 A 20 SALÁRIOS
MÍNIMOS" , "ACIMA DE 20 SALÁRIOS MÍNIMOS" }
@attribute "part_renda_familiar" { "NÃO TRABALHO. RECEBO AJUDA
FINANCEIRA DA FAMÍLIA" , "TRABALHO E RECEBO AJUDA FINANCEIRA DA FAMÍLIA"
, "TRABALHO. NÃO RECEBO AJUDA FINANCEIRA DA FAMÍLIA" , "TRABALHO E
CONTRIBUO PARCIALMENTE PARA O SUSTENTO DA FAMÍLIA" , "TRABALHO E SOU
RESPONSÁVEL PELO SUSTENTO DA FAMÍLIA" }
@attribute "aprovacao" { "SIM" , "NÃO" }


@data
(...)
```

Figura 11: Arquivo *ARFF* do Perfil Sócio-econômico

3.3 FERRAMENTA

Para a realização do trabalho proposto foi utilizada a ferramenta *Weka* (*Waikato Environment for Knowledge Analysis*). O *Weka* é um produto da Universidade de Waikato (Nova Zelândia) e foi implementado pela primeira vez em sua forma moderna em 1997. Ele usa a GNU *General Public License* (GPL) e foi escrito na linguagem Java. O *Weka* contém uma GUI para interagir com arquivos de dados e produzir resultados visuais, além de possuir uma API geral que permite incorporá-lo, como qualquer outra biblioteca, a outros aplicativos para realizar tarefas de mineração de dados automatizadas no lado do servidor, por exemplo. A versão estável mais atualizada do *Weka* é a 3.6.3.

Weka é uma coleção de algoritmos de aprendizagem de máquina para tarefas de mineração de dados. Os algoritmos podem ser aplicados diretamente a um

conjunto de dados ou chamados a partir do seu próprio código Java. *Weka* contém ferramentas para os dados de pré-processamento, classificação, regressão, clusterização, regras de associação e visualização. É também ideal para o desenvolvimento de novos modelos de aprendizagem de máquina.

Além dos algoritmos de aprendizagem que podem facilmente ser aplicados aos *datasets* (conjuntos de dados), *Weka* fornece uma variedade de ferramentas para transformação de *datasets,* como os algoritmos de discretização.

Como pode ser visto na Figura 12, há quatro interfaces disponíveis para utilização do *Weka*. Logo abaixo é feita uma descrição da utilidade de cada interface.

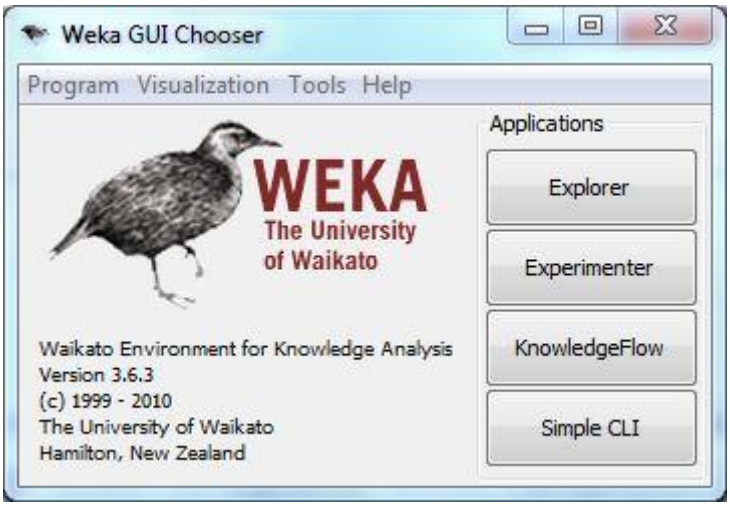

Figura 12: Janela inicial do *Weka*

- Explorer - A interface gráfica *Explorer* dá acesso a todas as facilidades usando seleção de menus e preenchimento de formulários. Por exemplo, você pode ler rapidamente em um conjunto de dados de um arquivo *ARFF* e construir uma árvore de decisão a partir dele. Ela orienta o usuário através da apresentação de opções, como menus, por forçá-lo a trabalhar em uma ordem apropriada, apresentando opções como formulários a serem preenchidos. (WITTEN e FRANK, 2005)

- Experimenter - A interface *Experimenter* foi projetada para ajudar o usuário a responder uma questão prática básica quando técnicas de classificação e

regressão estão sendo aplicadas: Quais métodos e valores de parâmetros atuam melhor para o problema dado? Normalmente não há uma maneira de responder a esta pergunta a priori, e uma razão pela qual foi desenvolvido o software foi proporcionar um ambiente que permitisse aos usuários *Weka* comparar uma variedade de técnicas de aprendizagem. Isso pode ser feito interativamente usando o *Explorer*. No entanto, o *Experimenter* permite automatizar o processo. (WITTEN e FRANK, 2005)

- Knowledge Flow - A interface *Knowledge Flow* permite que o usuário projete configurações de processamento de dados. Uma desvantagem fundamental do *Explorer* é que ele mantém tudo na memória principal. Quando se abre um *dataset*, ele imediatamente carrega tudo. Isso significa que só pode ser aplicado a problemas de pequeno e médio porte. No entanto, *Weka* contém alguns algoritmos incrementais que podem ser usados para processar conjuntos de dados muito grandes. O *Knowledge Flow* permite que seja especificado um fluxo de dados pela conexão de componentes que representam fontes de dados, ferramentas de pré-processamento, algoritmos de aprendizagem, métodos de avaliação, e os módulos de visualização. (WITTEN e FRANK, 2005)

- Simple CLI - Por trás destas interfaces interativas reside a funcionalidade básica do *Weka* que permite o acesso na sua forma bruta, inserindo comandos textuais, que dão acesso a todas as funcionalidades do sistema. (WITTEN e FRANK, 2005)

3.4 APLICAÇÃO DAS TÉCNICAS DE MINERAÇÃO

Como pode ser notado na Figura 11 mostrada anteriormente, que representa um dos arquivos no padrão *ARFF*, os dados não necessitaram de qualquer processo de discretização, já que não haviam atributos contínuos ou numéricos. Os atributos dos *datasets* são definidos como **categorizados** ou **quantitativos.** Além do mais, são tidos como **nominais,** pois os valores são formados apenas por nomes diferentes, e oferecem informações suficientes apenas para distinguir um objeto de

outro (=, ≠). As diversas classes de filtragem fornecidas no *Weka* não foram necessárias. Outro detalhe interessante é a conversão de *CSV* para *ARFF.* A interface *Explorer* do *Weka* permite que os *datasets* entrem no formato *CSV* e, utilizando duas classes de conversão, *weka.core.converters.CSVLoader* e *weka.core.converters.ArffSaver*, converte-os automaticamente para o formato padrão. Não foi possível fazer a conversão dessa forma, pois estavam sendo utilizados extensos conjuntos de dados. A solução encontrada foi a implementação de uma classe em Java, importando as duas classes de conversão do *Weka*, que recebia um arquivo no formato *CSV* e o convertia para *ARFF*.

A principal interface do *Weka* utilizada neste trabalho foi a *Explorer*. Em alguns casos, foi necessária a especificação de fluxos de dados na interface *Knowledge Flow,* pois falhas referentes à insuficiência da memória estavam sendo apresentadas no *Explorer*. O *Experimenter* foi utilizado também como uma forma de confirmar de forma automatizada o estudo comparativo entre as técnicas de aprendizagem feito interativamente com o *Explorer*.

É importante ressaltar que em todos os algoritmos de classificação utilizados neste trabalho, a opção de teste escolhida foi a validação cruzada (*cross-validation*) que garante que cada registro é usado o mesmo número de vezes para treinamento e exatamente uma vez para teste. Além do mais, os parâmetros *default* de cada algoritmo de classificação foram mantidos. Foi necessário realizar modificações nos parâmetros do algoritmo de associação, como será explicado no próximo capítulo.

4. RESULTADOS ALCANÇADOS

Após a conversão de todos os *datasets* para o formato padrão do *Weka*, e a definição dos algoritmos de classificação (*J48*, *PART* e *JRip*) e do algoritmo de associação (*Apriori*), o próximo passo consiste na aplicação dos algoritmos nos datasets particionados, a análise e a visualização do resultados. O *Weka* possui uma aba pra visualização de dados, *Visualize*. No entanto, esta opção permite apenas visualizar o conjunto de dados em si, e não os resultados de uma classificação, associação ou clusterização. O que a ferramenta *Weka* fornece para classificação por árvores de decisão é a visualização de árvores, útil apenas para árvores que contenham poucos nós. Como foram geradas árvores extensas, esta funcionalidade não foi utilizada. Para grandes árvores de decisão, foi feita uma visualização parcial, que percorre da raiz aos nós folhas aos quais estejam associadas uma grande quantidade de instâncias.

O total de vestibulandos inscritos no concurso vestibular da UESB (e contidos no d*ata mart* acadêmico) entre os anos de 2007 e 2009 nos três campi, desprezando os que possuíam pelo menos um atributo com valor nulo e desconsiderando o particionamento em 3 perfis, é de 20.030, onde 2.240 obtiveram aprovação e 17.790 foram reprovados.

A seguir são apresentados os resultados obtidos por cada algoritmo.

4.1 J48

Para o Perfil Sócio-Econômico, o algoritmo *tree.J48* gerou uma árvore de decisões contendo 13 nós, sendo que 10 deles eram folhas. Por não ter sido uma árvore extensa, foi possível demonstrá-la integralmente no trabalho, como pode ser vista na Figura 13. Note que na figura há apenas 12 nós, pois dentre os 13 que foram gerados havia uma folha vazia (sem qualquer registro associado). Há dois números em cada nó folha, onde o primeiro e o segundo indicam, respectivamente, o número de instâncias corretamente classificadas e incorretamente classificadas por aquele nó na etapa de treinamento.

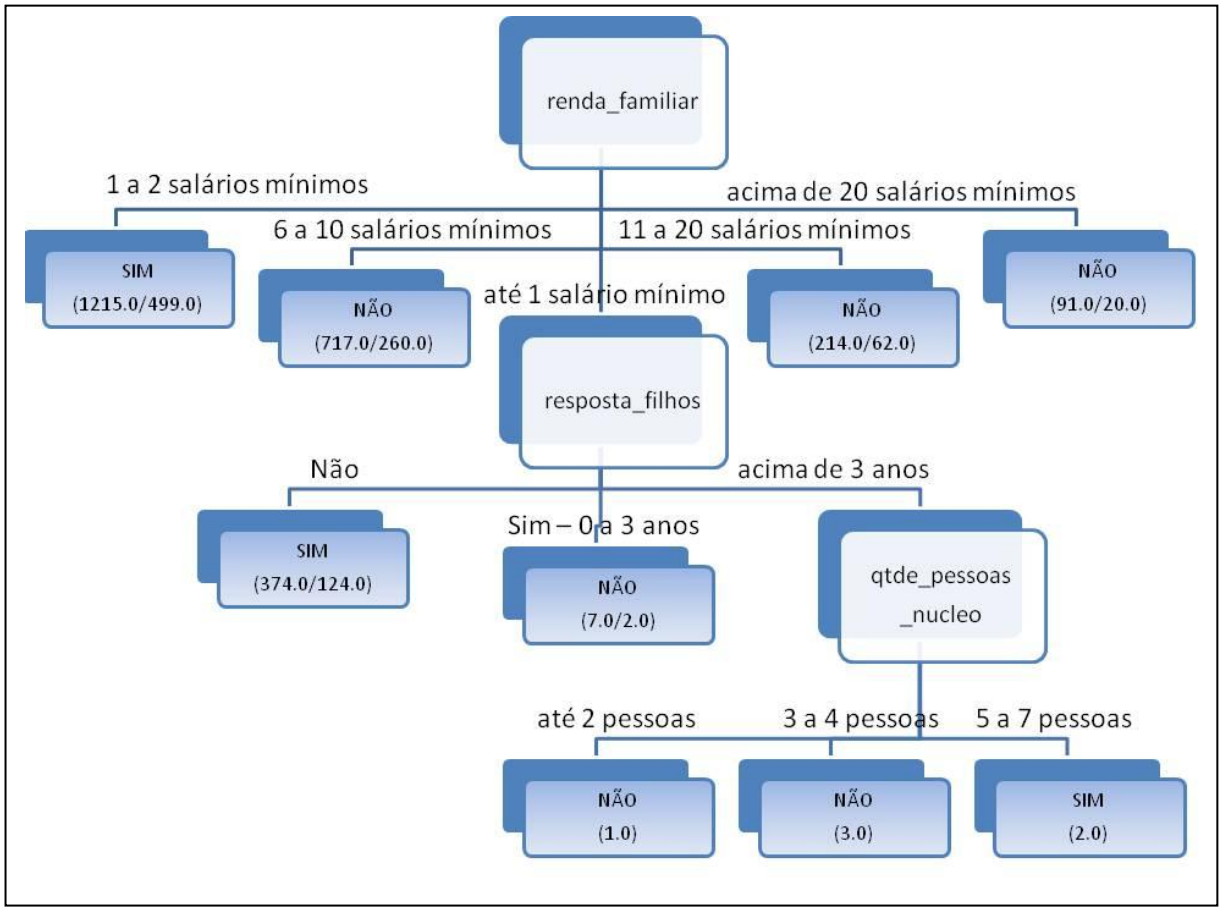

Figura 13: Árvore de decisão do Perfil Sócio-Econômico obtida pelo algoritmo J48

Analisando a árvore de decisão, pode-se observar que a renda familiar é um importante atributo que define o perfil dos ingressantes na universidade. É possível notar que os estudantes com renda familiar entre 1 e 2 salários mínimos são considerados como aprovados, já que o número de ingressantes é muito maior do que o número de reprovados. Outro ponto importante diz respeito a ter filhos, quando a renda familiar é até 1 salário mínimo. Para estudantes que têm filhos, é maior a dificuldade de obter aprovação.

A Figura 14 contém o resumo do resultado da aplicação do algoritmo para o Perfil Sócio-Econômico, onde pode ser observado que 61,96% das instâncias foram classificadas corretamente e 38,03% obtiveram erro na classificação. Pela **matriz de confusão**[6] pode ser visto que 943 candidatos aprovados foram classificados com o rótulo aprovação igual a SIM e 369 foram classificados incorretamente. Dentre os

[6] Tabela onde são expostas as contagens de registros de testes previstos correta e incorretamente pelo modelo de classificação. É utilizada na avaliação de desempenho do modelo.

reprovados, 683 tiveram o rótulo aprovação como NÃO e 629 foram incorretamente classificados como aprovados.

```
Correctly Classified Instances        1626        61.9665 %
Incorrectly Classified Instances      998         38.0335 %

=== Confusion Matrix ===
   a   b   <-- classified as
 943 369 |   a = SIM
 629 683 |   b = NÃO
```

Figura 14: Algoritmo *tree.J48* aplicado sobre o Perfil Sócio-econômico

O *tree.J48* aplicado ao Perfil Educacional obteve uma árvore de 93 nós, dentre os quais 71 eram nós folha. O percentual de registros classificados corretamente, como pode ser visto na Figura 15, foi igual a 60,33%, e com classes incorretas temos 39,66%.

```
Correctly Classified Instances        2703        60.3348 %
Incorrectly Classified Instances      1777        39.6652 %

=== Confusion Matrix ===

    a    b   <-- classified as
 1454  786 |    a = SIM
  991 1249 |    b = NÃO
```

Figura 15: Algoritmo *tree.J48* aplicado sobre o Perfil Educacional

Logo abaixo temos a árvore de decisão obtida para o Perfil Educacional. Foram selecionadas folhas que tivessem números expressivos. Analisando a árvore na Figura 16, pode ser observado que candidatos que fizeram todo o Ensino Médio em instituições públicas de ensino têm maior tendência de obter vagas na Universidade. Atributos como curso de língua estrangeira e pré-vestibular, além do turno no qual foi feito o Ensino Médio, serviram como uma valiosa forma de diferenciação.

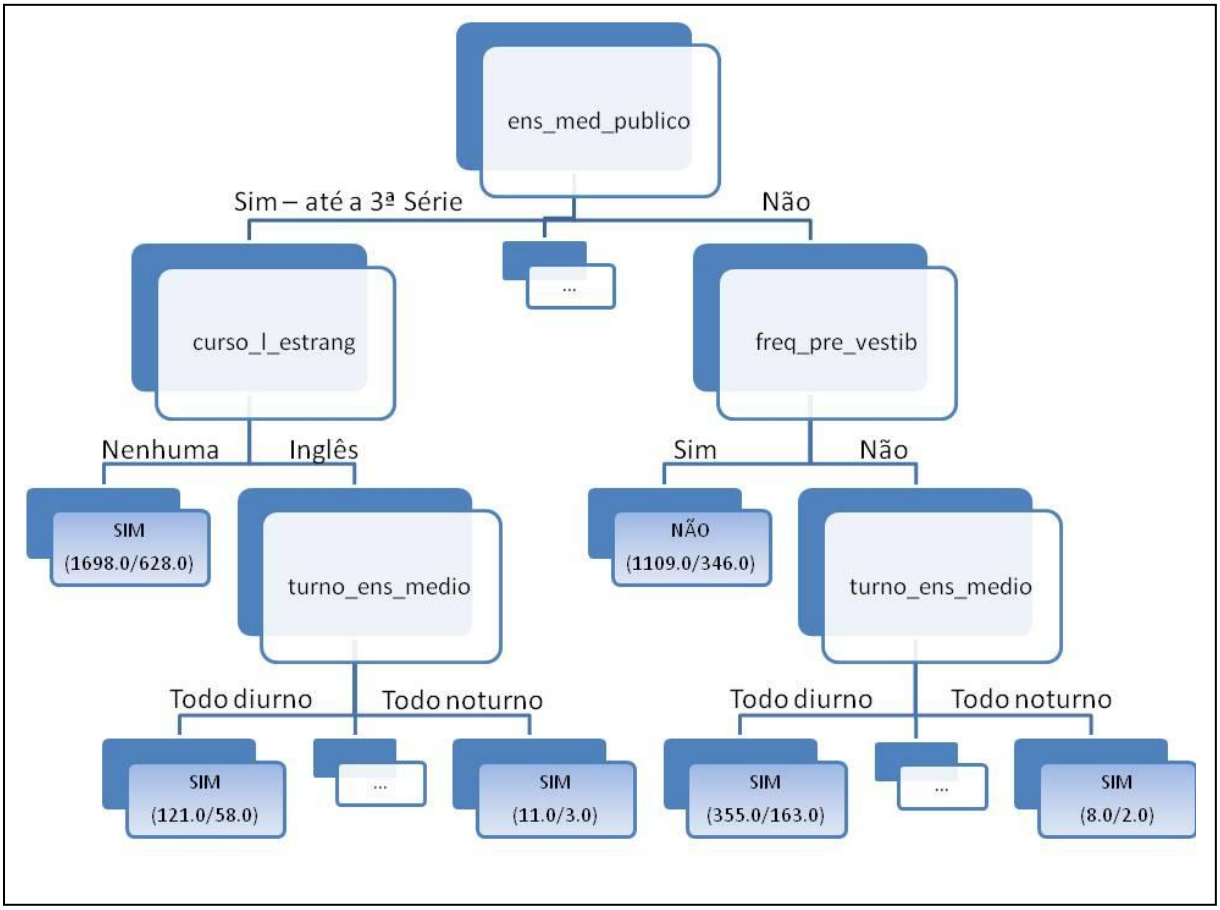

Figura 16: Árvore de decisão do Perfil Educacional obtida pelo algoritmo J48

A maior árvore gerada pelo *tree.J48* foi para o Perfil de Expectativas, que conteve 296 nós, dos quais 245 eram nós terminais. Como pode ser visto na Figura 17, o desempenho relacionado à porcentagem de classificação correta foi de 64,93%, e 35,06% dos registros não foram bem classificados.

```
Correctly Classified Instances        2909               64.933  %
Incorrectly Classified Instances      1571               35.067  %

=== Confusion Matrix ===

    a    b   <-- classified as
 1407  833 |    a = SIM
  738 1502 |    b = NÃO
```

Figura 17: Algoritmo *tree.J48* aplicado sobre o Perfil de Expectativas

Na Figura 18 há uma visão parcial da árvore de decisão. Pode ser observado que os candidatos que moram com a família no município da instituição (ou pretendem se mudar), têm dificuldade para obter aprovação na primeira tentativa, mas com a experiência adquirida tendem a conseguir na segunda ou na terceira vez. Outro detalhe interessante está no fato de que estudantes que têm parentes no município de um dos campi da UESB, e já foram aprovados em uma ou duas tentativas anteriores, mas por algum motivo tentaram o vestibular novamente, não são aprovados com a experiência obtida anteriormente.

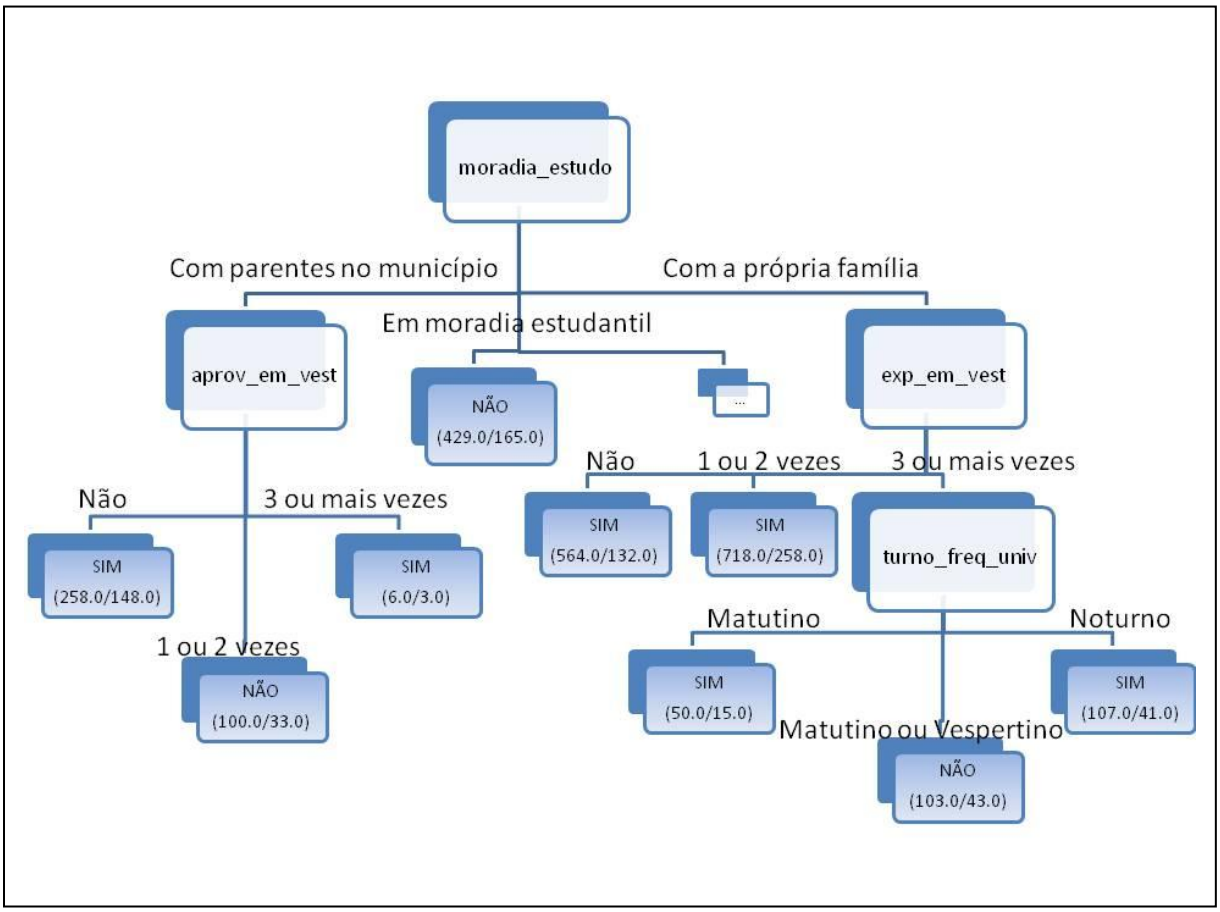

Figura 18: Árvore de decisão do Perfil de Expectativas obtida pelo algoritmo J48

4.2 PART

O algoritmo de classificação *rules.PART* sobre o Perfil Sócio-econômico gerou um total de 61 regras de classificação. Dentre elas, 5 foram selecionadas levando

em conta as que abrangessem um maior número de candidatos, tendo o conseqüente de cada regra (o atributo aprovação) como SIM:

- SE renda_familiar = até 1 salário mínimo E resposta_filhos = não E local_moradia = zona urbana – periferia ENTÃO aprovação = SIM (174.0/42.0)
- SE nome_estado_civil = solteiro E renda_familiar = 1 a 2 salários mínimos E local_moradia = zona urbana – periferia E nome_etnia = preto ENTÃO aprovação = SIM (104.0/31.0)
- SE nome_estado_civil = solteiro E renda_familiar = 1 a 2 salários mínimos E part_renda_familiar = trabalho e contribuo parcialmente para o sustento da família E qtde_pessoas_nucleo = de 3 a 4 pessoas ENTÃO aprovação = SIM (30.0/7.0)
- SE resposta_filhos = não E nome_estado_civil = solteiro E part_renda_familiar = trabalho. Não recebo ajuda financeira da família ENTÃO aprovação = SIM (23.0/10.0)
- SE nome_estado_civil = solteiro E part_renda_familiar = não trabalho. Recebo ajuda financeira da família ENTÃO aprovação = SIM (310.0/148.0)

Há dois valores numéricos para cada regra, onde o primeiro representa a quantidade de instâncias às quais a regra é aplicável, e o segundo indica as instâncias que seguem corretamente o antecedente, mas que têm o conseqüente diferente do esperado. Por meio da primeira regra, pode ser visto que candidatos cuja renda familiar seja de até 1 salário mínimo, moram na periferia e não têm filhos, tendem a ser aprovados no vestibular. A regra é aplicável a um número expressivo de instâncias, já que muitas regras foram geradas e a maior parte delas servia para poucos registros. A Figura 19 demonstra o percentual de instâncias corretamente classificadas e a matriz de confusão para o primeiro perfil.

```
Correctly Classified Instances        1563               59.5655 %
Incorrectly Classified Instances      1061               40.4345 %

=== Confusion Matrix ===

   a   b   <-- classified as
 817 495 |   a = SIM
 566 746 |   b = NÃO
```

Figura 19: Algoritmo *rules.PART* aplicado sobre o Perfil Sócio-econômico

A técnica de mineração por classificação utilizando o algoritmo *rules.PART* sobre o Perfil Educacional alcançou um total de 55 regras. Obteve um percentual de acertos igual a 61,47%, como pode ser constatado na Figura 20. Seguem abaixo 5 das regras de classificação obtidas sobre o perfil, cuja seleção foi realizada utilizando os mesmos critérios da anterior:

- SE curso_l_estran = nenhuma E ens_med_publico = sim – até a 3ª série E ens_fund_publico = sim – até a 4ª série ENTÃO aprovação = SIM (1242.0/426.0)
- SE curso_l_estran = nenhuma E turno_ens_medio = todo diurno E ens_med_publico = sim – até a 3ª série ENTÃO aprovação = SIM (322.0/133.0)
- SE curso_l_estran = nenhuma E ens_med_publico = sim – até o profisionalizante E turno_ens_medio = todo diurno ENTÃO aprovação = SIM (177.0/55.0)
- SE curso_l_estran = espanhol E ens_fund_publico = não ENTÃO aprovação = SIM (16.0/4.0)
- SE curso_l_estran = inglês ENTÃO aprovação = SIM (75.0/28.0)

Por meio das regras, é possível verificar que, apesar de língua estrangeira ser uma disciplina obrigatória no vestibular, onde o candidato tem o direito de escolher entre inglês e espanhol, a maior parte dos candidatos que obtiveram aprovação não fizeram qualquer curso objetivando aprender um segundo idioma. As últimas regras, que contém o atributo *curso_l_estran* como inglês e espanhol, tiveram números menos expressivos.

```
Correctly Classified Instances        2754               61.4732 %
Incorrectly Classified Instances      1726               38.5268 %

=== Confusion Matrix ===

    a    b   <-- classified as
 1501  739 |    a = SIM
  987 1253 |    b = NÃO
```

Figura 20: Algoritmo *rules.PART* aplicado sobre o Perfil Educacional

O total de regras obtidas para o Perfil de Expectativas foi extremamente

maior: 227. No entanto, as regras se adequavam melhor ao conjunto de dados já que o percentual de acertos na classificação foi igual a 64,01%, com 2.868 registros classificados corretamente e 1.612 classificados incorretamente, como pode ser notado na Figura 21. Seguem abaixo as regras selecionadas com os mesmos critérios:

- SE moradia_estudo = com minha própria família E exp_em_vest = não. É a primeira vez E meio_transp_estudo = transporte coletivo (ônibus) E preferencia_uesb = oferece o melhor curso da minha opção ENTÃO aprovação = SIM (83.0/17.0)
- SE moradia_estudo = com minha própria família E turno_freq_univ = matutino E meio_transp_estudo = transporte coletivo (ônibus) ENTÃO aprovação = SIM (174.0/31.0)
- SE turno_freq_univ = noturno E motivacao_curso = afinidade pessoal – vocação – realização pessoal E aprov_em_vest = não. É a primeira vez ENTÃO aprovação = SIM (220.0/43.0)
- SE motivacao_curso = outro motivo E aprov_em_vest = 1 vez E exp_em_vest = 2 vezes ENTÃO aprovação = SIM (10.0)
- SE motivacao_curso = mercado de trabalho garantido E turno_freq_univ - noturno ENTÃO aprovação = SIM (16.0/4.0)

Através da terceira regra de classificação, aplicável a um maior número de registros, é possível observar que a opção por cursos realizados no turno noturno facilita o ingresso na Universidade, mesmo sem ter obtido aprovação em algum vestibular realizado anteriormente. É possível que esta informação esteja relacionada a outros fatores, como menor concorrência, por exemplo.

```
Correctly Classified Instances        2868               64.0179 %
Incorrectly Classified Instances      1612               35.9821 %

=== Confusion Matrix ===

    a    b   <-- classified as
 1382  858 |    a = SIM
  754 1486 |    b = NÃO
```

Figura 21: Algoritmo *rules.PART* aplicado sobre o Perfil de Expectativas

4.3 JRIP

O algoritmo *rules.JRip* funciona, para um problema com duas classes, definindo uma como positiva e a outra como negativa, onde a positiva é aquela que contém o menor número de registros. As regras são geradas para a classe positiva, e a classe negativa é tida como a *default*. Apesar das duas classes conterem o mesmo número de registros em cada perfil, a classe de aprovados foi tratada como a positiva. Dessa forma, as regras classificam os aprovados, e os registros que não se enquadram a qualquer regra pertencem à regra padrão de reprovados.

```
JRIP rules:
===========

(local_moradia = ZONA URBANA - PERIFERIA) and
(renda_familiar = DE 1 A 2 SALÁRIOS MÍNIMOS) =>
aprovacao=SIM (525.0/169.0)

(renda_familiar = ATÉ 1 SALÁRIO MÍNIMO) => aprovacao=SIM
(387.0/133.0)

(renda_familiar = DE 1 A 2 SALÁRIOS MÍNIMOS) and
(qtde_pessoas_nucleo = DE 3 A 4 PESSOAS) => aprovacao=SIM
(372.0/169.0)

 => aprovacao=NÃO (1339.0/498.0)

Number of Rules : 4

=== Summary ===
Correctly Classified Instances      1607        61.2276 %
Incorrectly Classified Instances    1017        38.7724 %

=== Confusion Matrix ===

   a   b   <-- classified as
 854 457 |   a = SIM
 560 752 |   b = NÃO
```

Figura 22: Algoritmo *rules.JRip* aplicado sobre o Perfil Sócio-econômico

Para o Perfil Sócio-Econômico foram obtidas 4 regras de classificação,

incluindo a regra *default*. As regras classificam corretamente 854 candidatos aprovados e 752 candidatos reprovados, obtendo um percentual de acertos igual a 61,22%, como pode ser verificado na Figura 22. Tais regras reforçam o que foi observado na árvore de decisão do algoritmo *tree.J48*: renda familiar de 1 a 2 salários mínimos é uma característica dos ingressantes na Universidade. O atributo *local_moradia* definido como periferia foi considerado como relevante também pelo algoritmo *rules.PART*.

```
JRIP rules:
===========

(ens_med_publico = SIM - ATÉ A 3ª SÉRIE) => aprovacao=SIM
(1991.0/771.0)

(part_l_estran = NÃO) and (ens_med_publico = SIM - ATÉ O
PROFISSIONALIZANTE) and (turno_ens_medio = TODO DIURNO) =>
aprovacao=SIM (180.0/56.0)

 => aprovacao=NÃO (2309.0/896.0)

Number of Rules : 3
=== Summary ===

Correctly Classified Instances      2733       61.0045 %
Incorrectly Classified Instances    1747       38.9955 %

=== Confusion Matrix ===

    a    b   <-- classified as
 1386  854 |    a = SIM
  893 1347 |    b = NÃO
```

Figura 23: Algoritmo *rules.JRip* aplicado sobre o Perfil Educacional

A quantidade de regras de classificação encontradas para o Perfil Educacional foi igual a 3. O número de instâncias corretamente classificadas foi igual a 2.733, e 1.747 registros não foram classificados de forma correta, fazendo com que 61% das instâncias tivessem uma boa classificação. As regras, apresentadas na Figura 23 juntamente com a matriz de confusão, permitem verificar, assim como os resultados dos algoritmos aplicados anteriormente, que o curso de Ensino Médio em uma instituição pública de ensino é a característica mais comum entre os alunos

aprovados na UESB entre 2007 e 2009. Assim como foi verificado com o algoritmo *tree.J48*, os atributos referentes a curso de língua estrangeira e pré-vestibular são também bastante úteis para a classificação no Perfil Educacional.

```
JRIP rules:
===========

(moradia_estudo = COM MINHA PRÓPRIA FAMÍLIA) and
(aprov_em_vest = NÃO. É A PRIMEIRA VEZ) => aprovacao=SIM
(1161.0/328.0)

(turno_freq_univ = NOTURNO) => aprovacao=SIM (446.0/146.0)

(moradia_estudo = COM MINHA PRÓPRIA FAMÍLIA) and
(meio_transp_estudo = TRANSPORTE COLETIVO(ÔNIBUS)) =>
aprovacao=SIM (197.0/73.0)

(exp_em_vest = NÃO. É A PRIMEIRA VEZ) and (moradia_estudo =
FORA DO CAMPUS ONDE ESTUDO - VIAJANDO) => aprovacao=SIM
(51.0/13.0)

(aprov_em_vest = NÃO. É A PRIMEIRA VEZ) and (refeicoes =
NÃO) and (motivacao_curso = AFINIDADE PESSOAL - VOCAÇÃO -
REALIZAÇÃO PESSOAL ) and (meio_transp_estudo = TRANSPORTE
COLETIVO(ÔNIBUS)) => aprovacao=SIM (171.0/75.0)

(aprov_em_vest = NÃO. É A PRIMEIRA VEZ) and (refeicoes =
SIM - CANTINA) and (turno_freq_univ = VESPERTINO) =>
aprovacao=SIM (32.0/11.0)

 => aprovacao=NÃO (2422.0/828.0)

Number of Rules : 7
=== Summary ===

Correctly Classified Instances      2917       65.1116 %
Incorrectly Classified Instances    1563       34.8884 %

=== Confusion Matrix ===

    a    b   <-- classified as
 1401  839 |    a = SIM
  724 1516 |    b = NÃO
```

Figura 24: Algoritmo *rules.JRip* aplicado sobre o Perfil de Expectativas

O algoritmo *rules.JRip* aplicado ao Perfil de Expectativas gerou 7 regras de classificação, mostradas na Figura 24. As regras geradas garantem que atributos como moradia, aprovação em concursos de vestibular anteriores, experiência em vestibular e turno são bastante expressivos e valiosos para a classificação.

4.4 COMPARAÇÃO DOS RESULTADOS

A Tabela 4 resume o percentual de instâncias corretamente classificadas para os 3 perfis utilizados, por meio das técnicas de mineração por classificação com os algoritmos *tree.J48*, *rules.PART* e *rules.JRip*. Pode-se observar que cada perfil teve um algoritmo diferente que melhor se adequasse e obtivesse melhor resultado.

Comparação entre as taxas de classificação correta obtidas por diferentes algoritmos de classificação sobre os *datasets* utilizados			
Conjuntos de dados	Algoritmos de Classificação		
	tree.J48	***rules.PART***	***rules.JRip***
Perfil Sócio-econômico	**61.9665 %**	59.5655 %	61.2276 %
Perfil Educacional	60.3348 %	**61.4732 %**	61.0045 %
Perfil de Expectativas	64.933 %	64.0179 %	**65.1116 %**

Tabela 4: Resumo comparativo dos resultados de classificação

Para confirmar os resultados obtidos interativamente através da interface *Explorer*, foi utilizado o *Experimenter,* que possibilita uma avaliação de forma automatizada de uma variedade de técnicas de aprendizagem. Através do *Experimenter*, muitos algoritmos podem estar sendo investigados ao mesmo tempo, incluindo o mesmo algoritmo com parâmetros diferenciados para que sejam descobertos os melhores.

Como pode ser visto na Figura 25, na seção *Datasets* da aba *Setup* do ambiente de experimentos foram adicionados os conjuntos de dados utilizados neste trabalho. Na seção *Algorithms* foram selecionados os algoritmos de classificação a

serem investigados, e mantidos os parâmetros *default*, assim como foi feito no *Explorer*.

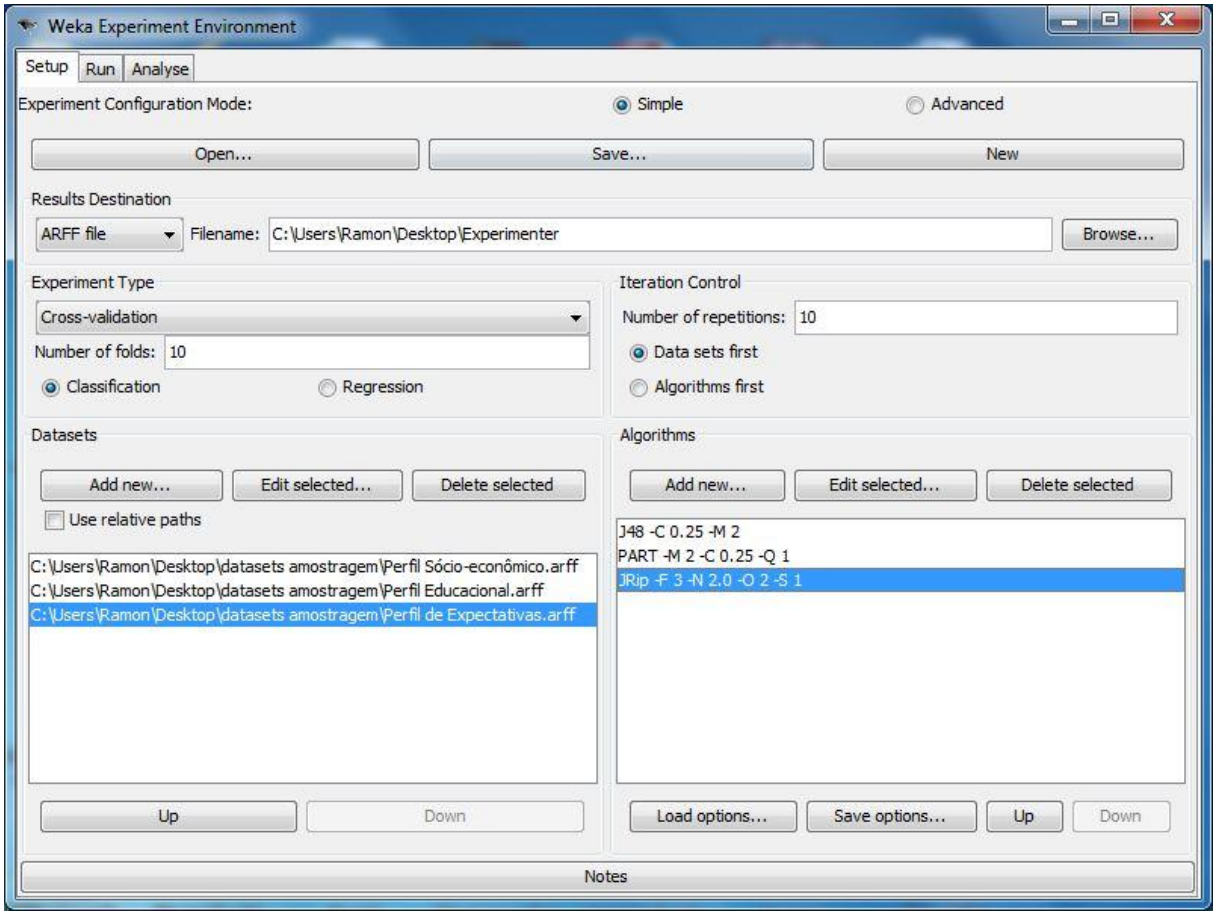

Figura 25: Interface *Experimenter* no *Weka*

O *Weka* cria um arquivo, geralmente em formato *ARFF*, que é utilizado para análise das informações referentes às comparações dos algoritmos aplicados aos *datasets*. O arquivo gerado para essa comparação foi intitulado *Experimenter.arff*. Na aba *Run*, após clicarmos em *Start*, o processo de comparação é iniciado. Com o arquivo de experimento pronto, a análise pode ser realizada na aba *Analyse*. A Figura 26 apresenta os resultados obtidos com a ferramenta.

Os valores encontrados não são exatamente os mesmos obtidos interativamente, mas são bem próximos. Como pode ser notado no *Explorer* e no *Experimenter,* o Perfil Sócio-Econômico obteve melhores resultados com o algoritmo *tree.J48*, o Perfil Educacional se adequou melhor ao algoritmo *rules.PART* e o Perfil de Expectativas obteve um modelo de classificação com melhor desempenho com o

algoritmo *rules.JRip.*

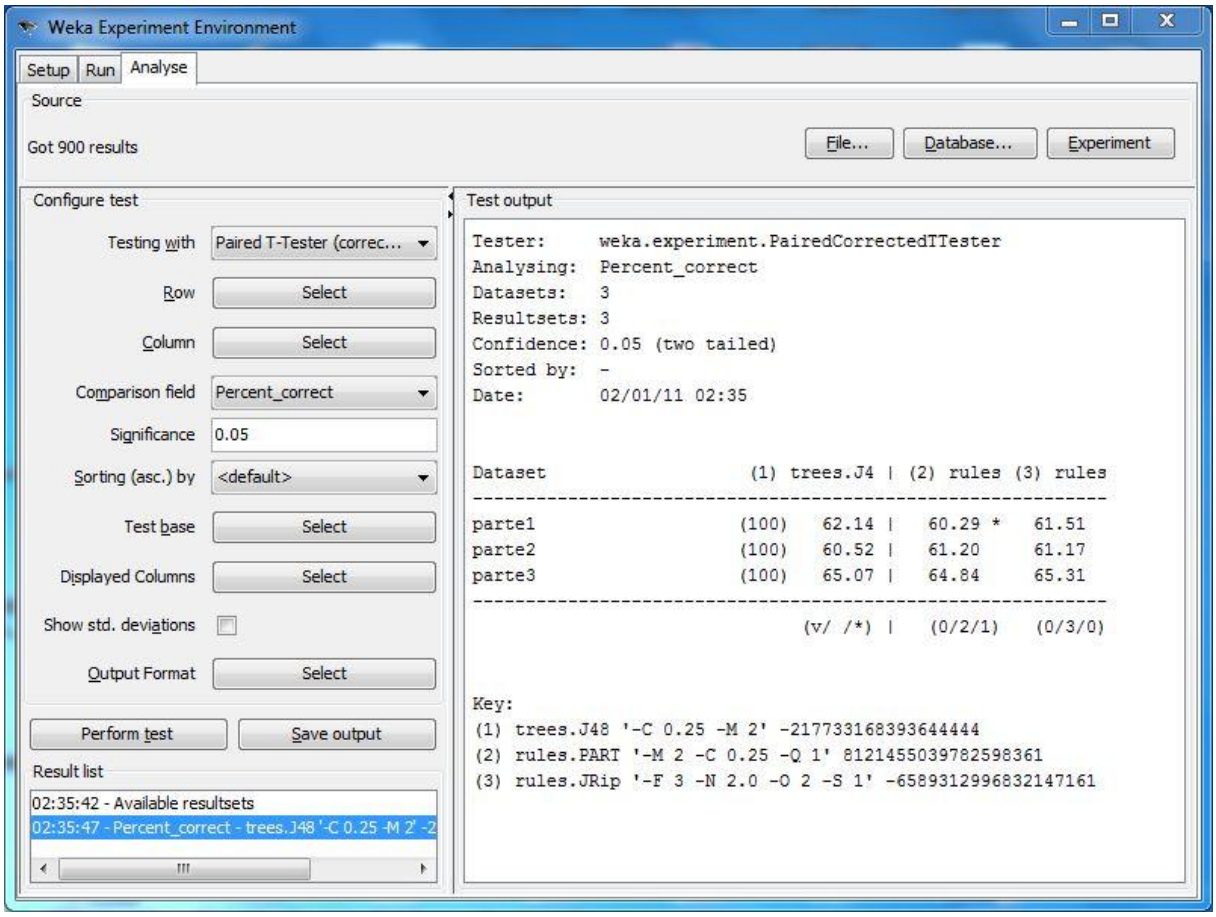

Figura 26: Aba *Analyse* da interface *Experimenter* no *Weka*

4.5 APRIORI

Para encontrar valores de atributos que estão associados com determinada confiança acima de uma métrica mínima pré-definida, foi utilizado o algoritmo *Apriori.* Para possibilitar a obtenção de atributos associados à aprovação como positiva, foi necessária a alteração de alguns parâmetros no *Weka.* O primeiro parâmetro alterado foi o *minMetric*, que define a confiança mínima das regras de classificação a serem geradas. O valor padrão para o parâmetro é 0.9, mas não foi possível obter regras com confiança tão elevada que associassem outros parâmetros ao de aprovação, e, portanto este valor foi alterado para 0.5 em todos os perfis. Outro parâmetro modificado foi o *numRules*, no qual é definido o número máximo de

regras de classificação. Tendo 10 como o valor *default*, apenas as 10 regras mais confiáveis seriam obtidas, mas não incluíam o atributo aprovação no conseqüente. Por esse motivo, valores mais elevados para o número de regras foram colocados, e as regras foram selecionadas seguindo o princípio de obter as regras mais confiáveis que implicassem na aprovação como positiva.

A Figura 27 contém as regras de classificação selecionadas para o Perfil Sócio-Econômico. Após o antecedente de cada regra, há um número que indica a quantidade de instâncias que possuem os atributos mostrados, e o número após o conseqüente revela o número de aprovados contido no conjunto de dados definido pelo antecedente. Através das regras geradas pelo algoritmo *Apriori* foi possível confirmar que a maior parte dos ingressantes são alunos que moram na periferia. Solteiros, sem filhos e com renda familiar de 1 a 2 salários mínimos são informações consistentes, e que permitiram conhecer o perfil dos aprovados. Após cada regra, é demonstrada a sua confiança.

```
1. local_moradia = "ZONA URBANA – PERIFERIA" nome_estado_civil =
   "SOLTEIRO" 824 ==> aprovacao = "SIM" 538    conf:(0.65)
2. local_moradia = "ZONA URBANA – PERIFERIA" resposta_filhos = "NÃO"
   part_renda_familiar = "NÃO TRABALHO. RECEBO AJUDA FINANCEIRA DA
   FAMÍLIA" 608 ==> aprovacao = "SIM" 396    conf:(0.65)
3. nome_etnia = "PARDO" resposta_filhos = "NÃO" renda_familiar = "DE 1 A
   2 SALÁRIOS MÍNIMOS" 639 ==> aprovacao = "SIM" 394    conf:(0.62)
4. nome_estado_civil = "SOLTEIRO" resposta_filhos = "NÃO" renda_familiar
   = "DE 1 A 2 SALÁRIOS MÍNIMOS" 1115 ==> aprovacao = "SIM" 666
   conf:(0.6)
5. renda_familiar = "DE 1 A 2 SALÁRIOS MÍNIMOS" part_renda_familiar =
   "NÃO TRABALHO. RECEBO AJUDA FINANCEIRA DA FAMÍLIA" 854 ==> aprovacao =
   "SIM" 497    conf:(0.58)
```

Figura 27: Regras de Associação para o Perfil Sócio-econômico

As regras de associação obtidas para o Perfil Educacional, apresentadas na Figura 28, consolidam o que foi verificado em classificação com o algoritmo *tree.J48*: ter cursado o Ensino Médio em instituições públicas de ensino é uma característica de uma grande parcela de estudantes da UESB. É possível observar também que

candidatos que têm acesso a computador com internet tendem a ser aprovados.

```
1. ens_med_publico = "SIM - ATÉ A 3ª SÉRIE" turno_ens_medio = "TODO
   DIURNO" part_l_estran = "NÃO" 1094 ==> aprovacao = "SIM" 724
   conf:(0.66)
2. ens_med_publico = "SIM - ATÉ A 3ª SÉRIE" freq_pre_vestib = "SIM"
   curso_l_estran = "NENHUMA" 925 ==> aprovacao = "SIM" 600
   conf:(0.65)
3. ens_med_publico = "SIM - ATÉ A 3ª SÉRIE" turno_ens_medio = "TODO
   DIURNO" curso_l_estran = "NENHUMA" acesso_computador = "SIM - COM
   INTERNET" 945 ==> aprovacao = "SIM" 611    conf:(0.65)
4. ens_med_publico = "SIM - ATÉ A 3ª SÉRIE" freq_pre_vestib = "SIM"
   acesso_computador = "SIM - COM INTERNET" 946 ==> aprovacao = "SIM" 583
   conf:(0.62)
5. ens_med_publico = "SIM - ATÉ A 3ª SÉRIE" 1991 ==> aprovacao = "SIM"
   1220    conf:(0.61)
```

Figura 28: Regras de Associação para o Perfil Educacional

A segunda regra obtida pelo algoritmo *Apriori* para o Perfil de Expectativas, como pode ser visto na Figura 29, confirma, com alta confiança, que candidatos que optam por cursos realizados no turno noturno têm mais facilidade de ingressar na UESB, como foi visto também no algoritmo de classificação *rules.PART*. A regra número 3 não contradiz a informação obtida pelo algoritmo *tree.J48* de que os candidatos que moram com a família têm dificuldade para obter aprovação na primeira tentativa, já que a regra não leva em conta que o candidato não tenha experiência, mas sim o fato de não ter sido aprovado anteriormente.

```
 1. motivacao_curso = "AFINIDADE PESSOAL - VOCAÇÃO - REALIZAÇÃO PESSOAL"
    moradia_estudo = "COM MINHA PRÓPRIA FAMÍLIA" aprov_em_vest = "NÃO. É
    A PRIMEIRA VEZ" 718 ==> aprovacao = "SIM" 536    conf:(0.75)
 2. turno_freq_univ = "NOTURNO" 777 ==> aprovacao = "SIM" 564
    conf:(0.73)
 3. moradia_estudo = "COM MINHA PRÓPRIA FAMÍLIA" aprov_em_vest = "NÃO. É
    A PRIMEIRA VEZ" 1161 ==> aprovacao = "SIM" 833    conf:(0.72)
 4. exp_em_vest = "NÃO. É A PRIMEIRA VEZ" aprov_em_vest = "NÃO. É A
    PRIMEIRA VEZ" 1274 ==> aprovacao = "SIM" 798    conf:(0.63)
 5. exp_em_vest = "1 VEZ" aprov_em_vest = "NÃO. É A PRIMEIRA VEZ" 860 ==>
    aprovacao = "SIM" 503    conf:(0.58)
```

Figura 29: Regras de Associação para o Perfil de Expectativas

5. CONCLUSÃO

As tecnologias de Inteligência Empresarial permitem às organizações a transformação de dados brutos em informações e conhecimento úteis que proporcionam a busca de estratégias de sustentação e de crescimento a serem desenvolvidas ou aproveitadas. As universidades estão cada vez mais investindo em técnicas de mineração, devido à importância de conhecer seus clientes estudantes, atraí-los e mantê-los, além da busca de conhecimentos relacionados ao desempenho dos alunos no decorrer do curso, facilitando a tomada de decisões.

A aplicação das técnicas de mineração de dados sobre o *data mart* Acadêmico permitiu a obtenção de informações valiosas e concretas sobre o perfil dos estudantes. As árvores de decisão e as regras de classificação são muito úteis para serem utilizadas de forma preditiva, possibilitando prever se determinado candidato ou grupo de candidatos obterá aprovação no próximo concurso de vestibular da Universidade.

Foi possível perceber que os estudantes da UESB, em sua maioria, moram com a família na periferia, são solteiros, não têm filhos e têm renda familiar de no máximo 2 salários mínimos. Como conhecimentos concretos, temos também que a maior parte dos ingressantes cursou o Ensino Médio em rede pública, em turno diurno, não fizeram curso de língua estrangeira, e estão incluídos digitalmente com computadores com acesso a internet em suas casas. Percebe-se também que a maior parte dos alunos não fez curso pré-vestibular. Através de regras com alta confiança, foi possível notar que candidatos que optam por curso no turno noturno têm maior facilidade de serem aprovados.

Por meio de informações comprovadas, onde já se conhece os alunos, diversas atitudes podem ser tomadas, visando melhorar as deficiências e fornecer um melhor ambiente que auxilie os estudantes de baixa renda e a comunidade. Como exemplo, a Universidade pode traçar formas de fornecer aos estudantes um curso gratuito de inglês ministrado pelos alunos de Letras Modernas, cursos de pré-vestibular em instituições de Ensino Médio que sirvam de estágio para estudantes de Pedagogia e de outros cursos da UESB, além de outros projetos de extensão que forneçam estágios aos estudantes, facilitando-lhes o ingresso no mercado de trabalho, e que tragam benefícios à comunidade.

Utilizando ferramentas *Open Source* como o *Weka*, as universidades podem contar com sistemas com qualidade sem o alto investimento em licenças de utilização e não impactando seus orçamentos, podendo assim administrar baseando-se em dados que já estão presentes em seus sistemas.

A principal dificuldade encontrada neste trabalho foi a ineficiência da universidade no quesito da manutenção de dados que definem a classificação dos candidatos e de relatórios do vestibular em bases de dados. A COPEVE possui tais informações apenas em grandes livros, intitulados como Relatório de Vestibular e Classificação Geral de Candidatos por Curso. Outro empecilho encontrado foi o fato de que o *data mart* não continha o número de inscrição. Como a Uinfor não podia passar informações pessoais como RG e CPF (única forma possível de associar as respostas com o candidato), os dados obtidos no vestibular de 2010 não puderam ser incluídos. Como citado anteriormente, a grande quantidade de valores nulos no *data mart* fez com que muitos registros, incluindo aprovados e reprovados, tivessem que ser desprezados.

5.1. TRABALHOS FUTUROS

Como sugestões para trabalhos futuros, é recomendado que o *data mart* seja reformulado, para que não seja necessário eliminar um alto percentual de candidatos. Seria interessante também que, sendo obtidas bases de dados relacionadas à classificação dos estudantes na Universidade, as técnicas de mineração de dados por classificação sejam utilizadas não apenas para classes binárias (aprovados e reprovados), mas para classes como aprovado, reprovado em redação, reprovado em português, reprovado em matemática, reprovado em língua estrangeira etc. Isto permitiria conhecer as dificuldades dos alunos, além dos atributos que se associariam a cada dificuldade, e permitiria formular projetos que melhorassem os números encontrados.

Trabalhos futuros utilizando técnicas de mineração para conhecer o desempenho dos alunos durante o curso seriam bastante úteis, pois facilitaria a tomada de decisões relacionadas à reformulação do quadro de disciplinas, disciplinas de verão e optativas que podem ser oferecidas, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

AGRAWAL, R. & SRIKANT, R. **Fast algorithms for mining association rules.** Proc. of the 20th Int'l Conference on Very Large Databases. Santiago, Chile, set. 1994.

ALMEIDA, L. M. et al. **Uma Ferramenta para Extrações de Padrões.** CEULP – ULBRA, 2003.

ELMASRI, R., NAVATHE, S. B. **Sistemas de Banco de Dados.** 4 ed. São Paulo: Pearson, 2005.

HAN, Jiawey; KAMBER, Micheline. **Data Mining: Concepts and Techniques.** Morgan Kaufman Publishers, 2001.

INMON, **Willian H. Building the Data Warehouse.** 4 ed. Wiley, 2002.

KALAKOTA, Ravi; ROBINSON, Marcia. **E-business: Estratégias para alcançar o sucesso no mundo digital.** 2. ed. Bookman, 2001.

KIMBALL, Ralph; ROSS, Margy. **The Data Warehouse Toolkit: The complete guide to dimensional modeling.** 2. ed. USA: Wiley, 2002.

LAUDON, K. C., LAUDON, J. P. **Sistemas de Informação Gerencias: Administrando a empresa digital.** 5. ed. São Paulo. Prentice-Hall, 2004 .

LOPES, L. P.; PRASS, F. S. **Técnicas de Data Mining aplicadas na base de dados do vestibular da UFSM.** Santa Maria: ULBRA, 2009.

MACHADO, F. N. R. **Tecnologia e projeto de Data Warehouse: uma visão multidimensional.** São Paulo: Érica, 2004.

MINERAÇÃO DE DADOS. Disponível na Internet. http://www.din.uem.br/~ia/mineracao/geral/index.html. Acessado em 11/12/2010.

NEXTG, Nextgeneration center. **CRM - Base de dados.** Disponível em: http://www.nextg.com.br - Acessado em: 02 de Junho de 2007.

PINTO, F. S. **A Construção de um Data Warehouse em um Ambiente Acadêmico – Caso UESB.** Vitória da Conquista: UESB, 2002.

PRIMAK, Fábio Vinícius. **Desicões com BI (Business Intelligence).** Ciência Moderna, 2008.

QUINLAN, R. **C4.5: Programs for Machine Learning**. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

RUD, Olivia Parr. **Data Mining Cookbook: Modeling Data for Marketing, Risk and Customer Relationship Management.** Wiley, 2001.

SANTOS, L. F. D. **Construção de um data mart acadêmico da UESB.** Vitória da Conquista: UESB, 2010.

TAN, P. N., STEINBACK, M, KUMAR, V. **Introdução ao Data Mining – Mineração de Dados.** Editora Ciência Moderna, 2009.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.** 2 ed. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2005.

ANEXO 1

Função criada para atribuir SIM ao campo aprovação nas respostas associadas aos candidatos convocados em primeira chamada.

```
CREATE OR REPLACE FUNCTION "public"."definirAprovados" () RETURNS boolean AS
$body$
DECLARE
 minimo int := min(id_aprovados) FROM aprovados;
 maximo int := max(id_aprovados) FROM aprovados;
 rgSel aprovados.rg_candidato%TYPE;
 anoSel aprovados.ano%TYPE;
 semestreSel aprovados.semestre%TYPE;
 periodoSel periodo.id_periodo%TYPE;
 id_resposta_questionarioSel resposta_questionario.id_resposta_questionario%TYPE;
 str1 varchar(20);
 num numeric;
BEGIN
 WHILE minimo<=maximo LOOP
        rgSel := rg_candidato FROM aprovados WHERE id_aprovados = minimo;
        anoSel := ano FROM aprovados WHERE id_aprovados = minimo;
        semestreSel := semestre FROM aprovados WHERE id_aprovados = minimo;

        periodoSel := id_periodo FROM periodo WHERE ano = anoSel AND semestre = semestreSel;

        str1 := '0' || rgSel;
        num := count(*) FROM candidatos WHERE id_periodo = periodoSel AND rg_candidato = rgSel OR
                rg_candidato = str1;
        IF num = 1 THEN
                id_resposta_questionarioSel := id_resposta_questionario FROM candidatos WHERE
                        id_periodo = periodoSel AND rg_candidato = rgSel OR rg_candidato = str1;
                UPDATE resposta_questionario SET aprovacao = 'SIM' WHERE id_resposta_questionario =
                        id_resposta_questionarioSel;
        ELSE
                id_resposta_questionarioSel := DISTINCT id_resposta_questionario FROM candidatos WHERE
                        id_periodo = periodoSel AND rg_candidato = rgSel OR rg_candidato = str1 ORDER BY
                        id_resposta_questionario LIMIT 1;
                UPDATE resposta_questionario SET aprovacao = 'SIM' WHERE id_resposta_questionario =
                        id_resposta_questionarioSel;
                RAISE NOTICE 'Houve Duplicação';
        END IF;
     minimo := minimo + 1;
 END LOOP;
 RETURN true;
END;
$body$
LANGUAGE 'plpgsql' VOLATILE CALLED ON NULL INPUT SECURITY INVOKER;
```