CURSO DE CIÊNCIA DA COMPUTAÇÃO

Romeu Cornelius Junior

USO DA MINERAÇÃO DE DADOS NA IDENTIFICAÇÃO DE ALUNOS COM PERFIL DE EVASÃO DO ENSINO SUPERIOR

Santa Cruz do Sul

2015

Romeu Cornelius Junior

USO DA MINERAÇÃO DE DADOS NA IDENTIFICAÇÃO DE ALUNOS COM PERFIL DE EVASÃO DO ENSINO SUPERIOR

Trabalho de Conclusão apresentado ao Curso de Ciência da Computação da Universidade de Santa Cruz do Sul, para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Jacques Nelson Corleta Schreiber

Santa Cruz do Sul
2015

Dedico este trabalho a todos que de alguma forma me deram suporte, apoiaram e incentivaram, acreditando que com esforço se alcança o sucesso.

Romeu Cornelius Junior

AGRADECIMENTOS

Em primeiro lugar, agradeço a minha mãe, Nercy, pelo apoio, incentivo e principalmente pelos ensinamentos de vida que têm me transmitido.

Agradeço à minha esposa Joseane, por compreender a ausência exigida e por sempre incentivar e auxiliar a cada passo que foi dado, mostrando a força do amor que nos une.

Agradecimento especial ao meu orientador, Prof. Dr. Jacques Nelson Corleta Schreiber, pelos conselhos, indicando o caminho correto a ser seguido durante este trabalho. Não deixando de agradecer a todos os professores da UNISC, que contribuíram com o ensinamento necessário para chegar até aqui.

Aos amigos e professores Eduardo Kroth, João Carlos Furtado, Rejane Frozza e Daniela Saccol, os quais considero serem exemplos de mestres.

Enfim, agradeço a todos que, de uma forma ou de outra, contribuíram no decorrer do curso e realização deste trabalho.

RESUMO

As universidades enfrentam o desafio de reduzir os índices de evasão dos alunos nos cursos de graduação. Este problema ocorre tanto em instituições públicas como privadas e seus efeitos estão relacionados com questões financeiras e com a diminuição do número de alunos formados no ensino superior. Este trabalho apresenta um estudo sobre a evasão no ensino superior e a aplicação de técnicas de mineração de dados na tentativa de descobrir, um perfil dos alunos com tendências evasivas e os padrões associados a essas tendências. Como parte da fundamentação teórica, foram verificados os principais conceitos relativos ao tema, levantamento de trabalhos relacionados, estudo das técnicas e algoritmos de mineração de dados, e como metodologia para a realização do estudo, foi escolhida a ferramenta mineradora de dados WEKA, além das técnicas de mineração conhecidas como classificação e associação, para a realização de experimentos de acordo com o domínio dos dados cedidos pela Universidade de Santa Cruz do Sul – UNISC. Esses dados foram cedidos em forma de arquivo texto pelo setor de Informática da universidade e importados para uma base de dados de apoio, em seguida transformados em arquivos .arff, padrão exigido pelo WEKA e então submetidos aos algoritmos de mineração. Os resultados obtidos mostram que o total de disciplinas cursadas e o status final das disciplinas do primeiro semestre são os fatores que mais colaboram para a evasão do curso.

Palavras chave: Evasão no ensino superior. Mineração de dados. Ferramenta WEKA. Perfil do aluno.

ABSTRACT

Reduce students dropout rates in graduate courses is a challenge faced by universities. This problem occurs in public and private institutions, and its effects are related to financial issues and the declining number of graduates in higher education. This paper presents a study on dropout in higher education and the application of data mining techniques. The aim is to find students with evasive trends and the patterns associated with these trends. As part of the theoretical background, were found the main concepts related to the topic, survey related work, study of techniques and algorithms for data mining, and as a methodology for the study, we chose the data mining tool WEKA. In addition to the known mining techniques such as classification and association, to conduct experiments according to the field of data provided by the University of Santa Cruz do Sul - UNISC. These data were given by the university's IT department in text file format and was imported into a supporting database, then transformed into .arff files required by WEKA software and then subjected to data mining algorithms. The results show that the final status of disciplines in the first semester and the number of disciplines taken semester by semester are the factors that collaborate to students dropout.

Keywords: Higher education dropout. Data mining. Tool WEKA. Studant profile.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE GRÁFICOS

LISTA DE ABREVIATURAS

ANDIFES	Associação Nacional dos Dirigentes das Instituições Federais
CSV	*Comma Separated Values*
DM	*Data Mining*
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
MEC	Ministério da Educação e Cultura
REUNI	Reestruturação e Expansão das Universidades Federais
WEKA	*Waikato Environment for Knowledge Analysis*
SQL	*Structured Query Language*
TI	Tecnologia da Informação
UNISC	Universidade de Santa Cruz do Sul

SUMÁRIO

1 INTRODUÇÃO

O problema da evasão ocorre em todas as universidades brasileiras, tanto em instituições públicas como privadas e seus efeitos estão relacionados com questões financeiras e com a diminuição do número de alunos formados no ensino superior gerando prejuízos para os alunos, universidades e para o país.

Reduzir estes índices da evasão dos alunos de graduação é um desafio enfrentado pelas universidades. Segundo o resumo técnico do censo da educação superior realizado pelo MEC em 2011 no qual participaram 2.365 instituições de ensino superior, que registravam 30.420 cursos, 6.739.689 matrículas, 2.346.695 ingressos e 1.016.713 concluintes de graduação, foram destacadas algumas diferenças entre as categorias administrativas das instituições de ensino superior no Brasil (INEP, 2014):

- A categoria pública apresenta 26,3% das matrículas de graduação; possui 32,3% dos cursos de graduação e 12,0% das instituições de ensino superior.
- A categoria privada concentra 73,7% das matrículas de graduação; possui 67,7% dos cursos de graduação e 88,0% das instituições de ensino superior.

Esta análise não contempla os números da educação a distância, somente educação presencial em instituições de ensino superior brasileira foram considerados nesse levantamento (INEP, 2014).

A necessidade de expansão da educação superior em nosso país é visível, segundo o Ministério da Educação, pois a média nacional é de que apenas 24% dos jovens brasileiros, com idade entre 18 e 24 anos, têm acesso ao ensino superior. Os dados estão disponíveis no Relatório de Acompanhamento do REUNI, elaborado pela Associação Nacional dos Dirigentes das Instituições Federais – ANDIFES (INEP, 2014).

As instituições de ensino superior oferecem a cada ano um crescente número de vagas para novos alunos ingressarem nos cursos de graduação. No entanto, parte dos alunos que entram na universidade não concluem o curso, embora existam políticas públicas de incentivo ao ingresso e financiamento de cursos superiores. Porém, o foco destas políticas de reestruturação como o REUNI,

são as instituições públicas de ensino, que por sua vez representam menos de 30% do total das matrículas de graduação.

Tendo por base que a maioria das matrículas não se concentra na rede pública de ensino, neste trabalho, apresentaremos um estudo utilizando dados acadêmicos de alunos de graduação do curso de ciência da computação, de uma universidade comunitária[1] brasileira – UNISC.

A mineração de dados educacionais e seus recursos, possibilitam desenvolver ou adaptar métodos e algoritmos de mineração existentes, para que esses possam apoiar efetivamente processos de detecção de comportamentos ligados à evasão escolar, de tal modo que se seja possível compreender melhor os dados em contextos educacionais, produzidos principalmente por alunos e professores, considerando os ambientes nos quais eles interagem (RIGO, 2012).

Neste contexto, o objetivo principal deste trabalho é auxiliar na busca por razões para a evasão no ensino superior, através da utilização de técnicas de mineração de dados e consequentemente contribuir aos gestores universitários no planejamento de ações efetivas para a retenção de alunos do curso de graduação em Ciência da Computação da Universidade de Santa Cruz do Sul - UNISC.

Pode-se considerar os seguintes objetivos específicos:

- Pesquisar conceitos aprofundados e técnicas ligadas à área de pesquisa, bem como entender como é o seu funcionamento e todos os aspectos envolvidos, principalmente na mineração de dados e seus algoritmos.
- Avaliar trabalhos relacionados com o assunto visando verificar o que já existe na área relacionado a esta proposta de estudo, a fim de estabelecer uma comparação ressaltando aspectos, como: universo de dados utilizados; técnicas de mineração de dados; resultados obtidos; contribuições para evitar a evasão no ensino superior.
- Aprender a arquitetura e funcionamento do programa para mineração de dados, WEKA (WITTEN; FRANK; HALL, 2011).
- Definir o universo de dados a serem utilizados no trabalho, gerando-se uma base de dados.

[1] Nota do autor: O caráter comunitário da UNISC faz com que ela cresça acompanhando os avanços tecnológicos, sem descuidar da atenção ao ser humano e ao meio ambiente, obtendo reconhecimento e destaque nas avaliações realizadas pelo MEC.

- Definir os algoritmos e tecnologias a serem adotados no desenvolvimento como a técnica de mineração de dados, sistema gerenciador de banco de dados entre outros.

Do ponto de vista social, este estudo se justifica pela diminuição da mão de obra qualificada que chega ao mercado de trabalho, cada vez mais escassa uma vez que menos de 24% dos jovens conseguem ingressar no ensino superior. Do ponto de vista empresarial, existe um custo fixo para as universidades manter a infraestrutura para receber os alunos, além deste custo fixo, o custo médio por aluno por ano gira em torno de R$ 9 mil (LOBO 2011), ou seja, cada aluno que evade são R$ 9 mil a menos para a universidade por ano. Do ponto de vista científico, conhecer os conceitos e técnicas ligadas à área de pesquisa, bem como entender como é o seu funcionamento são fatores importantes para o andamento de trabalhos futuros, além de definir o universo de dados e os algoritmos a serem utilizados nos experimentos.

O trabalho está organizado da seguinte forma: O capítulo 2 mostra os números da evasão universitária no Brasil e na UNISC, no capítulo 3 são exibidos os processos de descoberta do conhecimento em banco de dados KDD. O capítulo 4 faz uma breve discussão sobre a mineração de dados educacionais e são apresentados as principais tarefas e técnicas de mineração de dados. No capítulo 5 são exibidos dois dos principais programas mineradores de dados, amplamente utilizados nos trabalhos relacionados que são exibidos no capítulo 6. No capítulo 7, são exibidos a metodologia proposta e a contextualização do fenômeno em estudo, e por fim, no capitulo 8, são exibidas as conclusões e sugestões para trabalhos futuros.

2 NÚMEROS DA EVASÃO UNIVERSITÁRIA NO BRASIL

Este capítulo faz uma breve descrição do problema da evasão universitária no Brasil, além de exibir alguns números deste fenômeno no país e também no curso de ciência da computação da UNISC.

Segundo cálculo do pesquisador do Instituto Lobo para o Desenvolvimento da Educação, da Ciência e da Tecnologia, Oscar Hipólito, com base nos números do Censo do Ensino Superior divulgados pelo Ministério da Educação em dezembro de 2010, as perdas financeiras com a evasão no ensino superior em 2009 giram em torno de R$ 9 bilhões (LOBO, 2011).

Entre 2008 e 2009, os dados do censo mostram que um total de 896.455 alunos abandonaram a universidade, representando uma média de 20,9% do total de alunos. Nas instituições públicas, 114.173 alunos (10,5%) evadiram e nas instituições particulares, um total de 782.282 alunos (24,5%) dos alunos abandonaram seus cursos. Apenas 47,2% dos alunos se formaram após quatro anos de curso (LOBO, 2011).

Entre 2012 e 2013 o número de concluintes de graduação caiu 5,9%, de acordo com dados do Instituto Lobo para o Desenvolvimento da Educação, da Ciência e da Tecnologia. Em 2013, 991.010 alunos terminaram seus cursos contra 1.050.413 em 2012. Entre os alunos que concluíram, 229.278 (23%) eram de instituições públicas, 761.732 (77%) eram provenientes de instituições particulares. Os graus acadêmicos que apresentaram o maior índice de queda foram o bacharelado (7,1% - com 42 mil alunos a menos) e a licenciatura (11,1% - com 22 mil alunos a menos). O número de formandos nas instituições de ensino privadas diminuiu 6,7% (51.135 concluintes a menos que em 2012), enquanto nas instituições públicas a queda foi de 3,6% (8.268 universitários a menos com diploma).

Em média, cada aluno custa em torno de R$ 15 mil por ano nas universidades públicas e R$ 9 mil por ano para as instituições privadas, de acordo com o pesquisador Oscar Hipólito. Ele ainda explica que o cálculo é uma média e tende a ser maior, já que existem vários outros custos envolvidos na educação, como alimentação e transporte entre outros (LOBO, 2011).

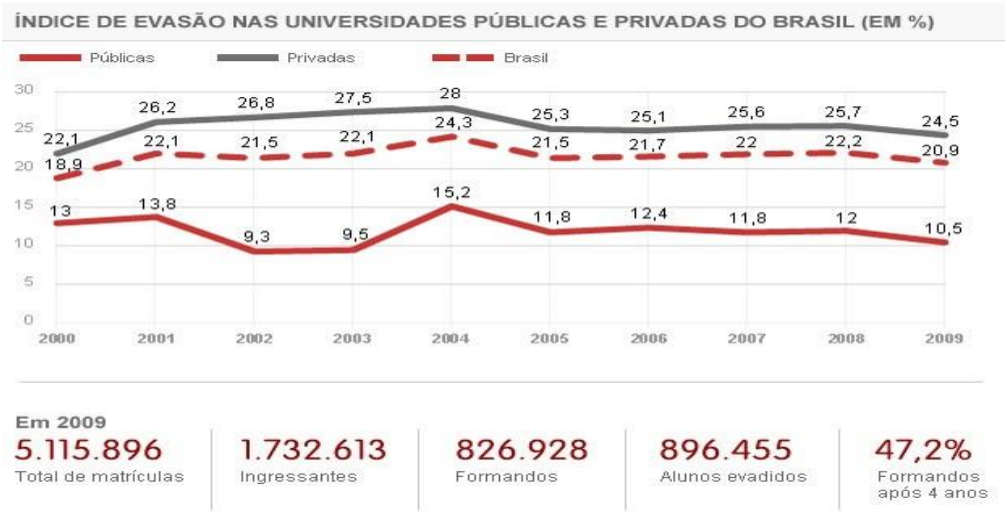
2.1 A evasão no curso de Ciência da Computação da UNISC

Entre 1993 e 2014, o curso de Ciência da Computação da UNISC teve 2.170 alunos matriculados, dos quais 348 (16,03%) são formados, 263 (12,12%) são alunos ativos e 1.559 (71,85%) são alunos que evadiram.

Considerando o valor de R$ 9 mil por ano por aluno para as instituições privadas, conforme mencionando anteriormente, e multiplicando esse valor por 4,5 anos (conforme o currículo do curso) temos o valor de R$ 45 mil para cada aluno evadido. Multiplicando esse valor pelos 1.559 alunos que já abandonaram o curso, chegamos a mais de R$ 70 milhões que deixaram de ser arrecadados pelo curso para a instituição.

Na Figura 01 são exibidos os números da evasão no Brasil entre os anos 2000 e 2009, pode-se ver que os índices da evasão a nível nacional são muito menores se comparados aos números da evasão do curso em estudo neste trabalho.

Figura 01 - Índice da evasão no Brasil



Fonte: (MEC, 2009).

A Figura 02 exibe os números da evolução da evasão a cada semestre do curso de Ciência da Computação da UNISC, onde pode-se ver que a grande maioria dos alunos evadem já nos primeiros semestres e que até o quinto semestre quase a metade dos alunos acabaram evadindo do curso.

Figura 02 - Evolução da evasão na UNISC

	Semestre 1	Semestre 2	Semestre 3	Semestre 4	Semestre 5	Semestre 6	Semestre 7	Semestre 8	Semestre 9	Semestre 10
Nenhuma disciplina	***	380	624	784	893	983	1066	1125	1159	1220
Uma disciplina	171	144	104	89	80	79	61	55	60	51
Duas disciplinas	304	256	207	190	155	133	105	107	88	103
Três disciplinas	721	503	406	330	311	278	260	247	244	224
Quatro disciplinas	195	227	228	205	180	179	172	148	156	138
Cinco disciplinas	415	338	268	209	224	177	165	148	123	98
Seis ou mais disciplinas	54	12	23	53	17	31	31	30	30	26
		20,43%	33,55%	42,15%	48,01%	52,85%	57,31%	60,48%	62,31%	65,59%
			13,12%	8,60%	5,86%	4,84%	4,46%	3,17%	1,83%	3,28%
Evadiram no semestre	***	380	244	160	109	90	83	59	34	61
Total de alunos ativos	1860	1480	1236	1076	967	877	794	735	701	640

Fonte: (Setor de informática da UNISC, adaptado pelo Autor).

Pode-se ver na primeira linha da tabela exibida na Figura 02, identificada pelo rótulo “Nenhuma disciplina”, o avanço da evasão no curso. Não existe alunos com nenhuma disciplina no primeiro semestre pois o pré-requisito da seleção dos alunos para o experimento era eles terem cursado pelo menos uma disciplina do curso.

Porém, na passagem do primeiro para o segundo semestre, encontramos 380 alunos que evadiram do curso. Do segundo para o terceiro semestre, um total de 624 alunos já haviam evadido do curso, do terceiro para o quarto semestre a soma dos alunos que evadiram é 724 e assim por diante.

A penúltima linha da tabela identificada pelo rótulo “Evadiram no semestre” exibe as mesmas informações, porém não de forma acumulativa, ou seja, mostra o número de alunos que evadiram em cada um dos semestres e não o somatório do total de alunos evadidos como é exibido na primeira linha. A última linha da tabela mostra o total de alunos ativos no curso, diminuindo semestre a semestre conforme o número de evasões vai aumentando.

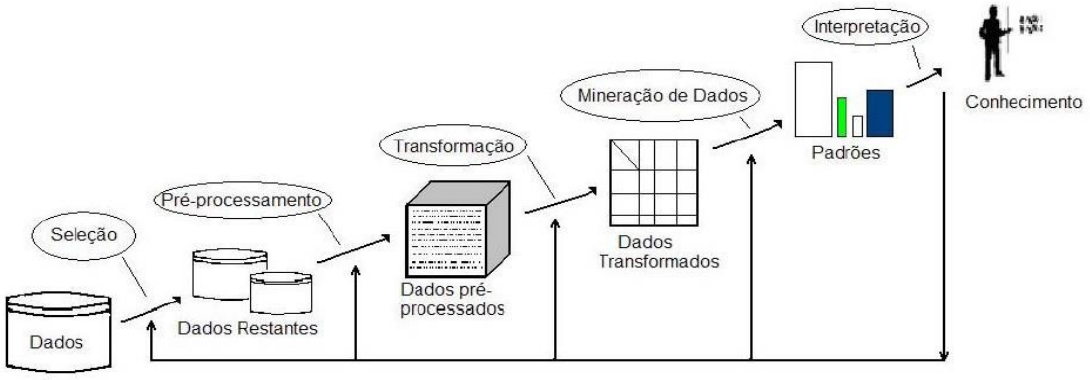
3 DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

Neste capítulo são discutidos os principais conceitos envolvidos no processo de descoberta de conhecimento em bases de dados.

Segundo Castanheira (2008), KDD (*Knowledge Discovery in Databases*) é um processo de descoberta de conhecimento em bases de dados que tem como objetivo principal extrair conhecimento a partir de grandes bases de dados. Para isto envolve diversas áreas de conhecimento, tais como: estatística, matemática, bancos de dados, inteligência artificial, visualização de dados e reconhecimento de padrões. São utilizadas técnicas, em seus diversos algoritmos, oriundos dessas áreas.

Para se iniciar um processo de KDD é preciso ter um claro entendimento do domínio da aplicação e dos objetivos que se deseja alcançar. Este processo é composto por cinco etapas, conforme pode ser visto na Figura 03.

Figura 03 - Fases da descoberta de conhecimento em bases de dados



Fonte: (Castanheira, 2008).

3.1 Etapas da descoberta de conhecimento em base de dados

O processo de KDD é dividido em passos. Desde a seleção da base de dados até a descoberta do conhecimento pode ser considerado um conjunto de atividades contíguas que compartilham conhecimento a partir de bases de dados, (CASTANHEIRA, 2008).

3.1.1 Seleção dos dados

Esta etapa envolve a compreensão do domínio e dos objetivos da tarefa, criação do conjunto de dados envolvendo as variáveis necessárias. Os dados são a espinha dorsal do processo de KDD mas usualmente não estão disponíveis de uma forma pronta para *data mining*, um dos principais problemas em coletar dados é descobrir onde encontrá-los.

De forma similar, para identificar as características de alunos com perfil de evasão em um modelo preditivo, variáveis alvo ou *target* (objetivo, resposta) e de entrada (preditoras), devem ser incluídas (HALL *et al*., 2009).

Assim, para resolver problemas específicos, dados adequados devem ser extraídos de banco de dados ou dados novos coletados que forneçam as exigências da tarefa a ser realizada.

3.1.2 Pré-processamento dos dados

Esta etapa envolve operações como tratar a falta de dados em alguns campos, limpeza de dados como a verificação de inconsistências, redução da quantidade de campos em cada registro, o preenchimento ou a eliminação de valores nulos, remoção de dados duplicados (CASTANHEIRA, 2008).

Inconsistências são bastante comuns neste tipo de tarefa e ocorrem quando um atributo assume valores diferente mas que representam a mesma coisa (GARCIA, 2012). Por exemplo, um atributo que armazena o nome de uma instituição de ensino, pode assumir os valores UNISC e Universidade de Santa Cruz do Sul, que do ponto de vista computacional são diferentes, porém representam a mesma informação.

3.1.3 Transformação dos dados

Esta fase antecede a seleção dos algoritmos de mineração de dados. Os algoritmos possuem padrões que devem ser respeitados, logo esta etapa do processo de KDD é realizada de acordo com o algoritmo de mineração que será utilizado na tarefa escolhida (CASTANHEIRA, 2008).

Estão envolvidas nesta etapa tarefas como identificação de ruídos, *outliers* (valores fora de uma faixa de valores aceitável para um atributo), generalização de atributos, discretização de variáveis.

Dados distorcidos ou que foram improvisados também conhecidos como ruído são bastante comuns. Eles acontecem principalmente quando um sistema é desenvolvido para um propósito específico, e passa a ser utilizado para outro (CASTANHEIRA, 2008). Por exemplo, um sistema de controle de frequência de alunos em eventos é projetado inicialmente para armazenar apenas dados dos participantes de um determinado evento, onde um atributo TIPOPESSOA deveria receber o valor ‘A’ para quem fosse aluno da instituição e o valor ‘N’ para quem não fosse aluno, mas queria participar do evento. Neste campo deveria aparecer apenas dos valores ‘A’ e ‘N’, porém alguns registros recebem o valor ‘P’ que se refere aos palestrantes do evento.

Generalizações podem ser utilizadas quando os dados são muito esparsos e não se consegue resultados satisfatórios com eles. Neste caso, dados primitivos são substituídos por conceitos. Um exemplo de generalização é quando o nome da cidade natal de um aluno é substituído pelo estado a qual essa cidade se refere (HALL *et al*., 2009).

A normalização é outro tipo de transformação de dados. O propósito da normalização é ajustar as escalas de valores dos atributos para um mesmo intervalo, e assim minimizar os problemas oriundos do uso de unidades de medida diferentes entre as variáveis. Um exemplo de normalização é um intervalo de [-2 a 2]. Redes neurais e a análise de *clusters*, tarefas que realizam operações matemáticas de multiplicação sobre o conjunto de dados, se beneficiam com esta técnica (HALL *et al*., 2009).

Alguns algoritmos de classificação e agrupamento trabalham somente com dados no formato nominal, ou seja, não conseguem lidar com os atributos medidos na escala numérica. Desse modo, dados do tipo “renda” devem ser “discretizados” por faixa, como: alta, média ou baixa (HALL *et al*., 2009).

3.1.4 Mineração de dados

Embora muitos autores considerem a mineração de dados como sinônimo de KDD, Fayyad, Piatetsky-Shapiro e Smyth (1996), define o processo de "KDD como sendo o processo geral de descoberta de conhecimento útil a partir dos dados e mineração de dados como uma determinada etapa neste processo, caracterizada como a aplicação algoritmos de específicos para extrair padrões a partir dos dados".

Castanheira (2008) resume em poucas palavras que a "mineração de dados caracteriza-se pela existência de um algoritmo que diante da tarefa proposta será eficiente em extrair conhecimento implícito e útil de um banco de dados. Pode-se dizer que mineração de dados é a fase do KDD que transforma dados puros em informação útil".

Passos adicionais no processo de KDD, tais como preparação de dados, seleção de dados, limpeza de dados, incorporação de conhecimento prévio adequado, e interpretação adequada dos resultados da mineração, são essenciais para assegurar que conhecimento útil será derivado a partir dos dados. Aplicação de métodos de mineração de dados cega pode ser uma atividade perigosa, facilmente levando à descoberta de padrões sem sentido ou inválidos (HALL *et al*., 2009).

Na fase de mineração dentro do processo de KDD, é escolhida a tarefa e definido o algoritmo a ser utilizado, podendo ser executado mais de uma vez já que esta etapa é um processo iterativo, para que haja a extração de padrões (GARCIA, 2012). Uma vez escolhido o algoritmo a ser utilizado, é necessário testá-lo e adaptá-lo a natureza da tarefa escolhida para a resolução do problema.

3.1.5 Avaliação dos resultados

Esta é a última etapa do processo de KDD, no qual os conhecimentos encontrados são interpretados e utilizados em processos de tomada de decisão. As medidas de desempenho (precisão, tempo, outros) também são exibidas nesta fase, podendo, caso necessário, ajustar parâmetros e voltar a alguma etapa anterior para ser executada novamente (REZENDE, 2003).

Os resultados da mineração de dados devem ser apresentados de forma clara, para que as informações possam ser interpretadas e visualizadas de diversas formas, utilizando-se de recursos visuais, como tabelas, gráficos entre outros.

4 MINERAÇÃO DE DADOS EDUCACIONAIS

Este capítulo faz uma breve explicação da mineração de dados, suas tarefas e métodos aplicados no contexto educacional.

A mineração de dados educacionais é um campo de investigação ainda não consolidado, está relacionada à aplicação de técnicas da mineração de dados junto aos mais diversos domínios de dados obtidos em diversos contextos educacionais, em sua grande maioria provenientes de ensino a distância (MANHÃES, 2011).

Esta é uma área de pesquisa nova e em expansão, que necessita de investigações complementares tanto na definição dos atributos a serem utilizados quanto nas técnicas de mineração de dados empregadas, tendo como principais metas os trabalhos relacionados com descoberta em modelos teóricos, modelos de predição, classificação, associação, mineração de relações e tratamento de dados para apoiar decisões (RIGO, 2012), (TINTO, 1975), (DEKKER *et al*., 2009).

Vários pontos que precisam ser aprimorados na utilização de técnicas de mineração em dados educacionais a fim de identificar alunos com padrões de evasão são indicados pelos autores (RIGO, 2012), (MANHÃES, 2011), dentre os quais pode-se destacar:

- Transformação dos dados (os dados selecionados nem sempre estão na forma adequada para serem usados diretamente pelos algoritmos de mineração).
- Identificação dos atributos mais relevantes.
- Identificação dos algoritmos de mineração mais adequados para a tarefa.
- Aplicação dos algoritmos selecionados.

Uma etapa de análise cuidadosa dos dados educacionais a serem minerados deve ser feita, em especial nos itens descritos anteriormente, pois além de ser uma etapa de grande importância dentro deste processo, busca reduzir exigências de hardware bem como tempo de processamento para a obtenção dos resultados (GARCIA, 2012). Também uma boa interpretação dos resultados da mineração de dados deve ser feita após o processamento destes dados, para que se possa aproveitar os resultados obtidos de forma eficaz.

4.1 Tarefas e técnicas da mineração de dados

Nesta seção são explicadas as principais tarefas e técnicas da mineração de dados além de exibir os principais métodos e algoritmos aplicados em cada técnica.

Os objetivos a serem alcançados são o fator responsável pela definição da escolha das tarefas a serem utilizadas na mineração de dados (GARCIA, 2012). Não existe uma definição genérica de tarefa que seja mais ou menos eficiente em qualquer situação, cada caso é um caso.

Após a escolha da tarefa, define-se a técnica a ser utilizada nela. Tarefa se diferencia de técnica de mineração pelo fato de a tarefa especificar qual a informação ou padrão deseja-se encontrar nos dados, e a técnica específica dos métodos que serão aplicados para alcançar os objetivos desejados (WITTEN; FRANK; HALL, 2011).

4.1.1 Tarefas preditivas

A predição é um dos objetivos fundamentais da mineração de dados, utiliza algumas variáveis que encontram-se no banco de dados, com a finalidade de prever valores desconhecidos ou futuros de outras variáveis que sejam de interesse (WITTEN; FRANK; HALL, 2011).

Nas tarefas preditivas (também conhecidas por modelos de descoberta) a abordagem é *bottom-up*, ou seja, a pesquisa é feita de forma a encontrar padrões frequentes, tendências e generalizações, a fim de encontrar informações que estavam escondidas nos dados (GARCIA, 2012).

Nas próximas seções serão apresentadas formas de se realizar as tarefas preditivas em base de dados.

4.1.1.1 Classificação

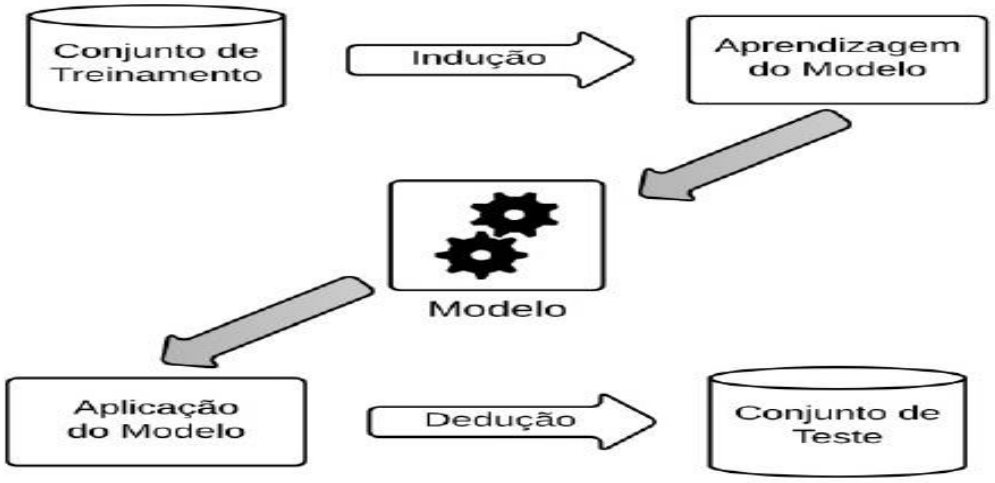
A tarefa de classificação diz respeito ao processo de encontrar um modelo que descreve e distingue classes de dados ou conceitos. Segundo Castanheira (2008), "A tarefa de classificação é uma função de aprendizado que mapeia dados

de entrada, ou conjunto de dados de entrada, em um número finito de classes. Nela cada exemplo pertence a uma classe, entre um conjunto pré-definido de classes".

O objetivo de um algoritmo de classificação é encontrar alguma correlação entre os atributos e uma classe, de modo que o processo de classificação possa usá-lo para predizer a classe de um exemplo novo e desconhecido (COSTA *et al*., 2013).

Um modelo de classificador é representado pela Figura 04, onde a entrada é um conjunto de treinamento, formado por um conjunto de amostras de dados onde a classe já é previamente conhecida. Com base neste conjunto de dados, a etapa de aprendizagem induz um modelo classificador que logo após é testado junto a outro conjunto de teste, que consiste em conjuntos de amostras onde as classes não são conhecidas e precisam ser preditas a partir do modelo.

Figura 04 - Representação de um classificador



Fonte: Costa *et al*. (2013).

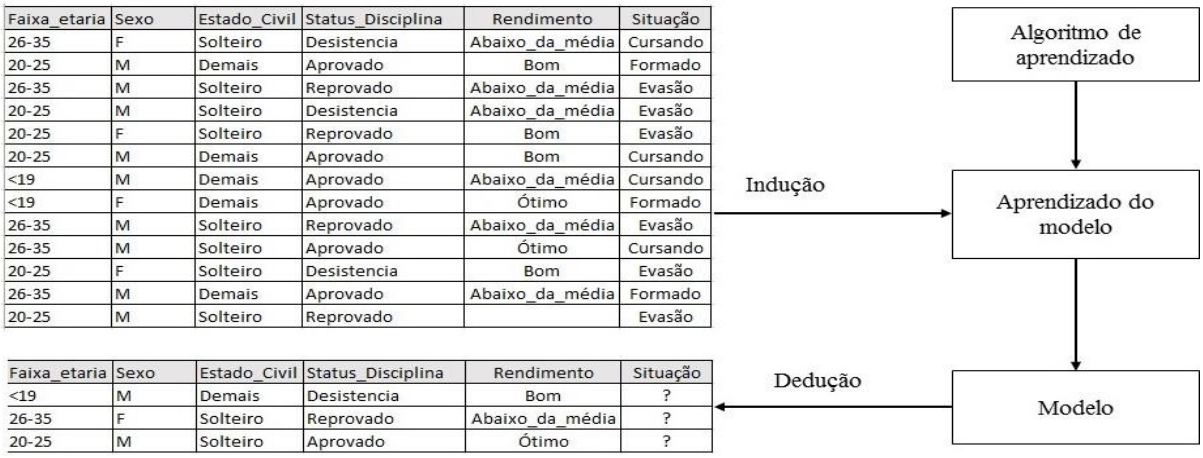
A tarefa de classificação pode ser dividida em duas etapas: Treinamento e classificação.

Na etapa de treinamento, também conhecida como aprendizado, utiliza-se um conjunto de dados denominados amostragem associados a suas classes (rótulos) para criar um modelo que será utilizado na construção do classificador. Este é um tipo de aprendizado conhecido como supervisionado, uma vez que o conjunto de dados utilizados é pré-definido (BUSS, 2011).

Na etapa da classificação, como o próprio nome sugere, faz-se o uso do modelo criado para o classificador. Utilizam-se agora outros conjuntos de dados, também conhecidos como teste, para estimar a precisão do classificador. Esta troca dos dados é importante para evitar o *overfit*, ou seja, evitar que o classificador se ajuste de tal forma que acaba sendo um classificador muito eficaz para os dados de treinamento, porém não tão eficaz para as demais amostragens de teste (BUSS, 2011).

A Figura 05 ilustra um modelo do processo de indução de um classificador e, em seguida, a sua utilização. Primeiro, um conjunto de treinamento, onde os rótulos das classes dos exemplos são conhecidos, é utilizado por um algoritmo de aprendizado para construir um modelo. Após a construção, esse classificador pode ser aplicado para predizer os rótulos das classes dos exemplos do conjunto de teste, ou seja, exemplos cujas classes são desconhecidas.

Figura 05 - Representação do processo de indução de um classificador



Fonte: Bramer (2013).
Obs.: Adaptado pelo autor.

Na tarefa de classificação, as técnicas mais utilizadas nos trabalhos relacionados foram árvores de decisão. A seguir são apresentados alguns trabalhos que aplicam técnicas de classificação em suas abordagens.

Em Dekker *et al*., (2009), são utilizadas árvores de decisão para testar a acurácia de vários classificadores no intuito de buscar perfis de alunos com tendências evasivas. Foram utilizados diversos algoritmos para classificá-los, entre eles o *OneR*, *CART*, *J48*.

Da mesma forma, Manhães *et al*. (2011), buscaram por padrões de evasão de alunos do ensino superior. Eles utilizam *OneR*, *JRip*, *J48* em suas árvores de decisão, para predizer quais métricas explicam a evasão na Escola Politécnica da Universidade Federal do Rio de Janeiro – UFRJ.

Uma técnica muito utilizada na tarefa de classificação são árvores de decisão, modelos estatísticos que utilizam treinamento supervisionado para classificação e predição dos dados (GARCIA, 2012). No conjunto de treinamento as variáveis preditivas são conhecidas, onde cada nó interno (não-folha), pode ser entendido como um atributo de teste, e cada nó-folha (nó-terminal) possui um rótulo de classe (COSTA *et al*., 2013).

Segundo Buss (2011) "A árvore de decisão é composta por estruturas chamadas de raiz, nós internos, arestas e folhas. Os nós internos significam testes sobre um determinado atributo, cada aresta representando um possível valor para esse atributo e cada folha apresentando um valor do atributo classe (rótulo) com que se deseja classificar a tupla de entrada. A raiz é o primeiro atributo a ser testado".

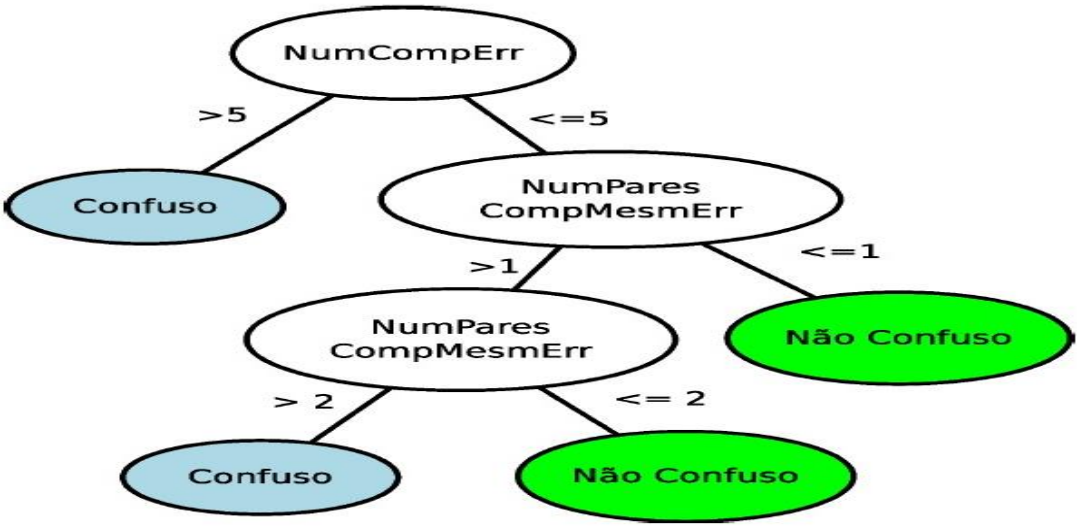
O aprendizado em árvores de decisão é do tipo supervisionado, sua construção é baseada no modelo *Top-down*, partindo do nó raiz em direção às folhas terminais. Os algoritmos dessa categoria se utilizam da técnica de dividir para conquistar, dividindo os problemas em problemas de menores dimensões até encontrar a solução para cada um dos problemas divididos (WITTEN; FRANK; HALL, 2011).

Classificadores com essa técnica procuram dividir sucessivamente o conjunto de dados, até que cada conjunto contemple apenas uma classe, tornando desnecessárias novas divisões (COSTA *et al*, 2013).

A árvore de decisão é montada a partir de dados de treino, a princípio tem-se apenas um nó que contém todas as classes. Recursivamente, escolhe-se um atributo que possa dividir esta classe, até que não haja mais divisões e cada nó folha represente uma única classe ou satisfação de um critério (GARCIA, 2012). A escolha do atributo a ser testado em cada nodo é o que define o sucesso de um algoritmo de aprendizado, que gera a árvore de decisão.

A Figura 06 exibe um exemplo de árvore de decisão que classifica alunos da disciplina de programação entre Confusos e Não Confusos de acordo com os atributos “Número de Compilação Com Erros” e o atributo “Número de Pares de Compilações com o Mesmo Erro”.

Figura 06 - Exemplo de árvore de decisão



Fonte: (Costa *et al*., 2013).

O primeiro teste feito na árvore é sobre a variável *NumCompErr (número de compilação com erros)* onde é verificado se ela é maior que 5, classificando o aluno como confuso, senão ela testa a segunda variável *NumParesCompMesmErr* (*quando o aluno compilou o mesmo erro mais que uma vez, ou seja, um par*), para valores maiores que 1 outro teste é feito sobre a mesma variável *NumParesCompMesmErr* que irá classificar os alunos como confuso quando o valor dessa variável for maior que 2.

4.1.1.2 Regressão Linear

Modelos de regressão linear são muito parecidos com modelos de classificação. Na tarefa de classificação, os atributos alvos da predição são do tipo discreto enquanto na regressão são do tipo numérico e contínuo (WITTEN; FRANK; HALL, 2011).

Esta tarefa também utiliza técnicas de árvores de decisão, porém, diferentemente da tarefa de classificação onde a técnica é utilizada para classificar instâncias, a regressão busca realizar uma estimativa de valor de uma determinada variável, ou seja, mapear um dado em um ou mais valores reais. Enquanto na tarefa anterior, os registros são classificados em uma classe, nesta tarefa os registros são classificados em um valor baseado em uma função matemática (GARCIA, 2012).

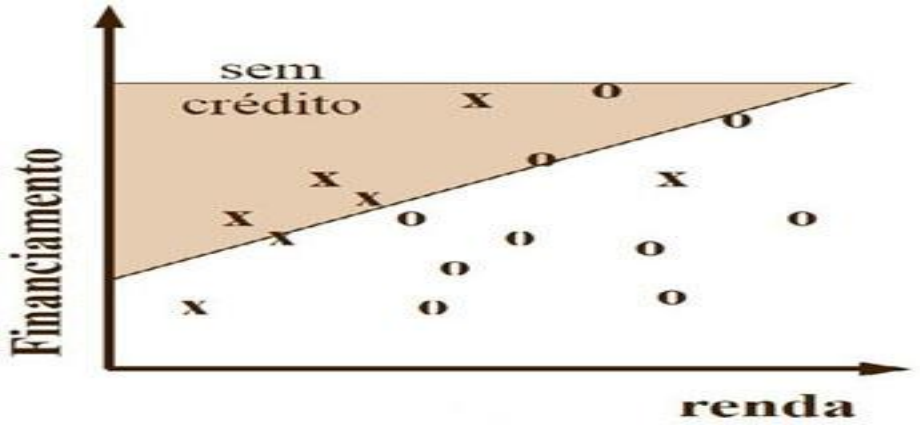
Em seu livro, Witten, Frank e Hall (2011) ressalta que quando o resultado ou a classe alvo da predição é numérica e todos os atributos são numéricos, regressão linear é uma técnica natural e se considerar. É um método baseado em estatísticas. A ideia é expressar a classe como uma combinação linear dos atributos com pesos pré-determinados:

$$X = W_0 + W_1 A_1 + W_2 A_2 + ... + WA_k A_k$$

Onde X é a classe, A_1, A_2,...,A_k são valores de atributos e W_0, W_1,..., W_k são os pesos.

Modelos lineares são fáceis de se visualizar em duas dimensões, que é o equivalente a desenhar uma linha reta através de pontos de dados (WITTEN; FRANK; HALL, 2011). O gráfico 1 mostra um exemplo de uma linha montada sobre o financiamento (crédito) estudantil, onde apenas a renda do aluno é utilizada como entrada. A classe financiamento é mostrada no eixo vertical e a renda no eixo horizontal, ambos são atributos numéricos. A linha traçada representa o melhor ajuste da equação de predição. Os pontos “X” do gráfico representam os alunos sem crédito enquanto os pontos “O” do gráfico representam os alunos com crédito aprovado.

Gráfico 1 - Exemplo de regressão linear



Fonte: (Witten; Frank; Hall, 2011)

Modelos lineares também podem ser aplicados em problemas de classificação binária. Nestes casos, a linha produzida pelo modelo separa as duas classes. Ela define onde a decisão muda de uma classe de valores para a outra, tal

linha é muitas vezes referida como fronteira de decisão (WITTEN; FRANK; HALL, 2011).

4.1.2 Tarefas descritivas

A descrição também é um dos objetivos fundamentais da mineração de dados, busca por padrões que descrevem os dados, de forma que possam ser interpretáveis pelos usuários, a fim de encontrar respostas que confirmem ou neguem as hipóteses (WITTEN; FRANK; HALL, 2011).

Nas tarefas descritivas (também conhecidas por modelos supervisionados, modelos de verificação) a abordagem é do tipo *top-down*, ou seja, existem hipóteses que foram previamente formuladas e são testadas para a verificação da sua veracidade (GARCIA, 2012).

Nas próximas seções serão apresentadas formas de se realizar as tarefas descritivas em base de dados.

4.1.2.1 Regras de Associação

Dentre as tarefas descritivas na mineração de dados mais utilizadas, encontra-se a tarefa de análise de associações ou regras de associação. Esta tarefa consiste na descoberta de regras que mostram condições nos valores dos atributos que sugerem padrões de associação fazendo um levantamento de quanto um conjunto de atributos contribui para a presença de outro conjunto, realizando um estudo de como os itens estão relacionados (GARCIA, 2012).

Podem ser aplicadas em estudos de preferência, buscando por afinidade entre os dados. Seu principal objetivo é encontrar conjuntos de itens ou eventos que ocorram junto, baseado na teoria de que a presença de um item em uma determinada transação, implica na ocorrência de outro (BUSS, 2010).

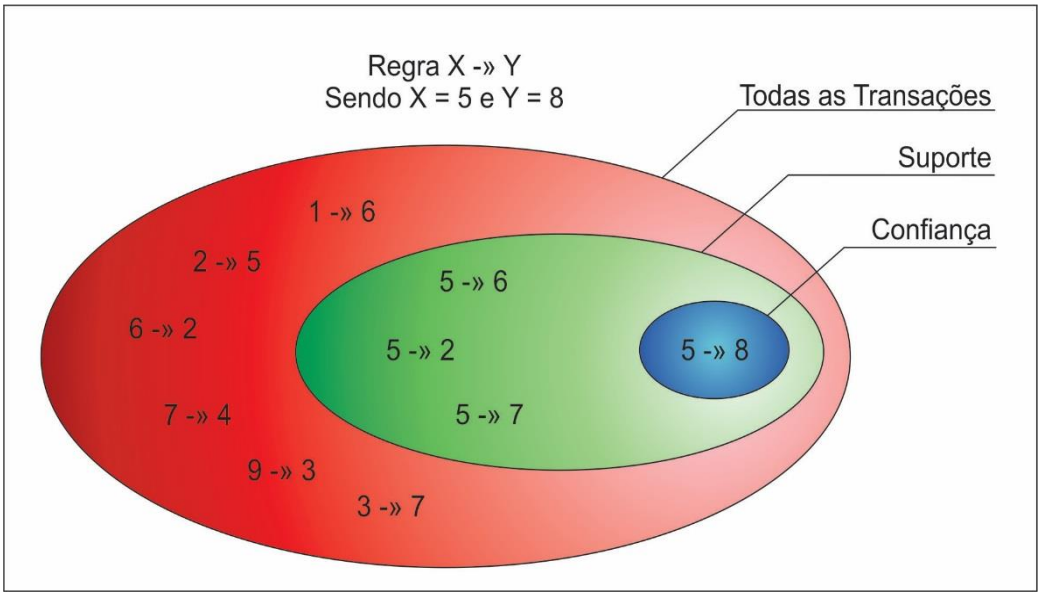
Uma regra de associação é uma expressão de implicação X -> Y, onde X e Y são um conjunto distinto de itens. Na formulação de regras de associação, duas métricas são consideradas importantes: o suporte e a confiança (WITTEN; FRANK; HALL, 2011).

Suporte determina a frequência na qual uma regra se aplica a um conjunto de dados, já a confiança indica a frequência na qual os itens em Y aparecem em transações que contenham X, indicando assim a probabilidade de associação entre o conjunto de dados selecionados. Com isso, um suporte de 0,5 para uma regra de associação indica que apenas 5% de todas as transações sob análise estão aparecendo juntas. Da mesma forma, um nível de 8% de confiança estabelece esse grau de garantia dos itens estarem agrupados (WITTEN; FRANK; HALL, 2011).

A indução de regras é uma técnica comum neste tipo de tarefa, "os algoritmos dessa técnica consistem em regras de previsão, do tipo SE..ENTÃO, em que SE é a condição da regra e ENTÃO prevê o valor de algum atributo solicitado" (GARCIA, 2012). Por exemplo, poderíamos minerar regras com base nas notas dos alunos em suas disciplinas, do tipo “80% dos alunos que têm bom desempenho na disciplina de Lógica tem bom desempenho em estrutura de dados e programação”. Os algoritmos de regras de associação se utilizam de operador lógico AND para gerar regras do tipo conjuntiva (COSTA *et al*, 2013).

No Gráfico 2 é exibido um modelo de diagrama demonstrando possíveis conjuntos nos cálculos de suporte e confiança. Como pode-se observar, no conjunto vermelho nenhuma parte da regra é atendida, no conjunto verde (suporte) a regra é parcialmente atendida e no conjunto azul (confiança) a regra é totalmente atendida.

Gráfico 2 - Possíveis Conjuntos nos Cálculos de Suporte e Confiança



Fonte: (GARCIA, 2012).

O algoritmo mais utilizado nesse tipo de técnica é o *Apriori.* O princípio *Apriori* diz que se um conjunto de itens é frequente, então todos os seus subconjuntos também devem ser frequentes (WITTEN; FRANK; HALL, 2011). Esta ideia pode ser vista no conjunto de dados exibidos no Gráfico 2. Suponha que o conjunto {c,d,e} seja um conjunto frequente, então toda a transação que conter esse conjunto, também deverá conter os seus sub conjuntos {c,d}, {c,e}, {d.e}, {c}, {d}, {e}.

4.1.2.2 Agrupamento *Clustering*

A tarefa de agrupamento, também conhecida como o *clustering* é uma técnica onde os algoritmos de agrupamento possuem aprendizado não supervisionado. Com esta técnica se espera conhecer novos atributos alvos (rótulos) a partir de um conjunto de dados, sem ter classificação prévia (COSTA, *et al*., 2013).

Tem como principal objetivo dividir um conjunto de dados formando grupos onde os dados fiquem agrupados de acordo com a semelhança entre eles, baseando-se em modelos probabilísticos ou medidas de similaridade, determinando quais são estes grupos, dividindo assim grupos heterogêneos de dados em vários sub grupos homogêneos. O agrupamento muitas vezes é uma alternativa apresentada pelas técnicas de classificação, nas quais deve haver a preocupação em rotular e coletar informações para formar o conjunto de treinamento assim como os conjuntos de testes (BUSS, 2010).

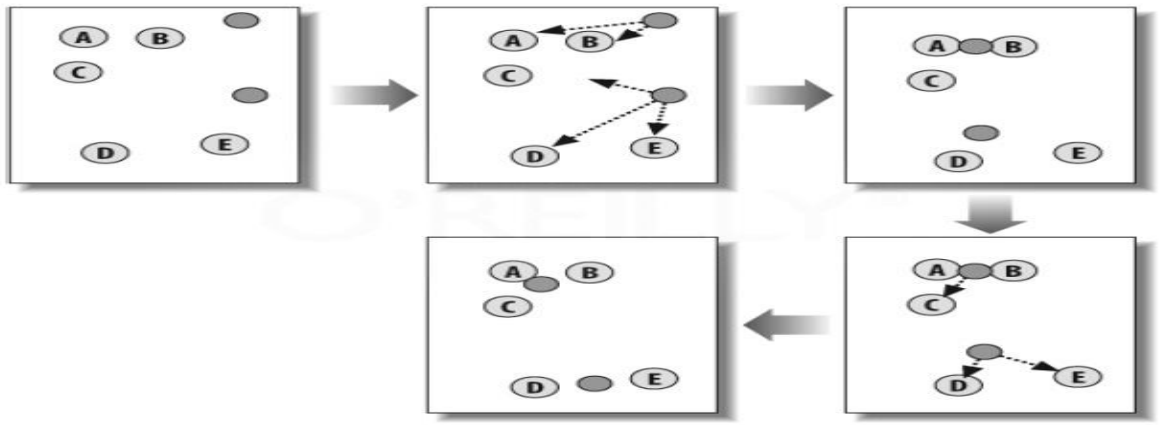
O algoritmo *K-Means* é uma técnica muito utilizada para esta tarefa, em que o objetivo é agrupar n elementos de um banco de dados em k agrupamentos. Onde n corresponde ao número de itens da amostragem selecionada e k o número de agrupamentos desejados. O número k de grupos que se deseja encontrar precisa ser informado de antemão (COSTA *et al*., 2013).

Este algoritmo tem como vantagem a eficiência em tratar grandes conjuntos de dados, porém como desvantagem tem a necessidade de informar o número k de agrupamentos no início do processamento, tornando o algoritmo um tanto limitado, já que geralmente não se sabe em quantos grupos ficam mais bem subdivididos os dados (GARCIA, 2012).

A determinação do número de *clusters* é um processo iterativo, no qual o modelador estima esse número e, após várias simulações, opta pela melhor

alternativa. Em seguida, k pontos são escolhidos aleatoriamente para representar os centroides dos grupos, então para cada registro no banco de dados, encontra-se a semente mais próxima para que este registro faça parte do grupo desta semente. A cada iteração do algoritmo, os centroides são recalculados de acordo com os elementos presentes no grupo e em seguida todos os elementos são realocados para a partição cujo novo centroide se encontra mais próximo (COSTA *et al*., 2013). A Figura 07 exibe o passo a passo do funcionamento do algoritmo *K-Means*.

Figura 07 - Funcionamento do algoritmo *K-Means*



Fonte: Costa *et al*. 2013.

Em seu trabalho Campello e Lins (2008) utilizaram técnicas de agrupamento em conjunto com o algoritmo *K-Means* na construção do modelo de análise e tratamento da evasão e retenção discente no curso de Engenharia de Produção da UFPE, que permitiu aos autores reconhecer seis tipos de classes diferentes distintas de alunos. A partir das características dessas classes foi possível identificar novas alternativas de ação para o problema da evasão. A Figura 08 exibe as classes (*clusters*) dos alunos definidas pelo autor.

Figura 08 - Clusters de alunos associados à evasão/retenção

Fonte: Campello e Lins (2008).

4.1.2.3 Padrões sequências

Os itens de uma cesta de compras contêm informações temporais sobre quando um item foi comprado por um determinado cliente. Estas informações podem ser reunidas a fim de determinar a sequência de transações feitas por um cliente em um determinado período de tempo.

De forma semelhante, dados baseados em eventos coletadas sobre a sequência das disciplinas cursadas por alunos no curso ou mesmo a sequência de páginas WEB visitadas no *site* da universidade, também podem ser reunidas a fim de se determinar se uma sequência de disciplinas cursadas contribui para a evasão ou se alunos que pouco visitam as páginas da biblioteca da universidade têm maior propensão a reprovar nas disciplinas.

Isto significa que uma relação comum, geralmente baseada em precedência temporal ou espacial, existe em eventos que ocorra em tais itens (TAN *et al*., 2009). Em outras palavras, uma sequência é uma lista ordenada de elementos. A seguir é exibida uma lista de exemplo de sequências:

- Sequências de páginas WEB visualizadas por um aluno no site da universidade: ({Homepage}, {Serviços *online*}, {Serviço acadêmicos}, {Mural eletrônico}, {Calendários}).
- Sequências de disciplinas cursadas por um aluno do curso de ciência da computação: ({Algoritmos, Estrutura de dados}, {Sistemas de bancos de dados, Sistemas operacionais I}, {Redes de computadores, Engenharia de software}, {Computação gráfica, Cálculo I}).

Uma sequência pode ser caracterizada pelo seu tamanho e o número de eventos ocorrentes, O tamanho de uma sequência é o número de elementos presentes nessa sequência (TAN *et al*., 2009). Na sequência das páginas WEB do exemplo anterior existem cinco elementos e cinco eventos. Na sequência de disciplinas cursadas, há quatro elementos e 8 eventos.

A descoberta de padrões sequenciais se dá através de um conjunto de dados que contenha uma ou mais sequência de dados. O termo sequência de dados se refere a uma lista ordenada de eventos associada a um único objeto de dados. O suporte de uma sequência *s* é a fração de todas as sequências de dados que contenham *s*. O usuário especifica um *minsup* ou valor mínimo para suporte, e se o suporte para s for maior que esse *minsup*, então ele é declarado como um padrão sequencial (TAN *et al*., 2009).

O algoritmo mais utilizado neste tipo de tarefa é o *FP-Growth*, que apresenta uma distinta diferença do algoritmo *Apriori* pelo fato de não concordar com o paradigma de gerar e testar. Em vez disso, ele codifica o conjunto de dados usando uma estrutura de dados compacta chamada de árvore FP e extrai o conjunto de itens diretamente desta estrutura. Este algoritmo é apresentado de forma sucinta na seção 4.3.5.

4.2 A classificação e árvores de decisão

Nesta subseção é feita uma descrição mais aprofundada do uso de árvores de decisão aplicadas na tarefa de classificação.

Árvores de decisão geralmente apresentam aprendizado indutivo dividido em aprendizado supervisionado e não-supervisionado (GARCIA, 2012). Conforme visto

nos capítulos anteriores, Figura 03, uma tarefa de classificação interessante no escopo deste trabalho pode ser a identificação da situação final do aluno no curso, em que para cada aluno são definidos atributos categóricos ordinais ou atributos contínuos (Exemplo: idade, média no vestibular, média no curso, frequência, nota final nas disciplinas, entre outras) e atributos categóricos não-ordinais (Exemplo: sexo, naturalidade, estado civil, entre outros). A função do classificador é fazer um mapeamento dos atributos para um status que representa a situação final do aluno no curso (Exemplo: cursando, formado, evasão).

Nas tarefas de modelagem descritiva, um modelo de classificador é utilizado como uma ferramenta para diferenciar dados de diferentes classes. Um exemplo disso é utilizar um modelo de classificador para identificar quais são as principais causas da desistência de uma determinada disciplina. Com isso, é possível chegar a conclusões, por exemplo, de que em sua grande maioria, os alunos que desistiram de uma determinada disciplina, apresentaram rendimento abaixo da média e estão na faixa etária entre 26 e 35 anos. Quando há o interesse em análise descritiva, é desejável que o modelo de classificação seja de fácil interpretação, ou seja, que fique evidente ao usuário o porquê de um determinado dado pertencer a uma determinada classe.

Outro fator que torna essa técnica muito utilizada é que o conhecimento adquirido pode ser representado por meio de regras. Essas regras podem ser expressas em linguagem natural, facilitando assim o entendimento por parte dos envolvidos (BUSS, 2011).

4.2.1 Principais conceitos sobre árvores de decisão

Nesta subseção serão vistos de forma mais aprofundada os principais conceitos envolvidos na construção de árvores de decisão.

Após a construção de uma árvore de decisão, pode-se utilizá-la imediatamente e com um custo computacional muito baixo. Além disso, a interpretação da árvore de decisão é uma das suas principais virtudes.

Uma árvore de decisão pode ser estruturada de diversas maneiras a partir de um conjunto de atributos. De forma exaustiva, à medida em que o número de atributos cresce, o número de árvores de decisão possíveis cresce

exponencialmente, tornando impraticável definir a estrutura da árvore de decisão ótima para um determinado problema, devido ao elevado custo computacional envolvido nessa busca (BRAMER, 2013).

Nesse sentido, algoritmos baseados em heurísticas têm sido desenvolvidos para a indução de árvores de decisão. Mesmo que eles não garantam uma solução ótima, apresentam resultados satisfatórios em tempo aceitável. Um desses algoritmos é o algoritmo de Hunt, que é a base de muitos algoritmos de indução de árvores de decisão existentes, como o CART (BREIMAN *et al*., 1994), ID3 (QUINLAN, 1986), C4.5 (QUINLAN, 1993).

4.2.1.1 Modelo de indução Top-Down

Baseado no algoritmo *Top-Down Induction of Decision Tree* que serve como base para os principais algoritmos de indução para árvores de decisão, este modelo gera regras de decisão em uma árvore de decisão, a qual é construída por várias divisões do conjunto de dados de acordo com os valores de seus atributos preditivos (BRAMER, 2013).

Na prática, este modelo é baseado em um algoritmo recursivo de busca gulosa que busca, sobre um conjunto de atributos, aqueles que “melhor” dividem o conjunto dos dados de exemplo em subconjuntos. Primeiramente, todos os dados são colocados em um único nodo, chamado de nodo raiz. Em seguida, um atributo preditivo é escolhido para representar o teste desse nodo e, consequentemente, dividir os dados em sub-conjuntos de dados. Esse processo se repete recursivamente até que todos os dados já estejam classificados ou então até que todos os atributos preditivos já tenham sido utilizados (WITTEN; FRANK; HALL, 2011).

4.2.1.2 Seleção dos atributos preditivos para os nodos das árvores

A escolha por qual atributo preditivo será utilizado em cada nodo da árvore é baseada no critério de seleção. Existem diversos tipos de critérios de seleção, sendo esta uma das diferenças entre os variados algoritmos de indução de árvores de

decisão. Esses critérios são baseados em termos da distribuição de classe dos dados antes e após a divisão (WITTEN; FRANK; HALL, 2011).

A grande maioria dos algoritmos de indução busca dividir os dados de um nodo-pai de forma a minimizar o grau de impureza dos nodos-filhos. Os critérios para a seleção da melhor divisão são baseados em diferentes medidas, tais como dependência, impureza e distância. Quanto menor for o grau de impureza, mais desequilibrada é a distribuição das classes. Se todos os dados pertencem a uma mesma classe em um determinado nodo, a impureza dele é nula. Da mesma forma, se existir o mesmo número de exemplos para cada classe possível, o grau de impureza é máximo neste nodo (BRAMER, 2013).

Algumas das medidas mais utilizadas para a seleção da melhor divisão são apresentadas a seguir.

4.2.1.3 Métricas para a melhor divisão da árvore

Existem muitas métricas que podem ser utilizadas para determinar a melhor forma de dividir os dados. Conforme mencionado anteriormente, essas métricas são definidas em termos da distribuição da classe dos dados antes e após a divisão.

Muitas vezes, o grau de impureza do nodo filho é a base utilizada por essas métricas para selecionar a melhor divisão. Quanto menor o grau de impureza, mais distorcida é a distribuição da classe (BRAMER, 2013).

O Ganho de Informação é uma das medidas baseadas em impureza, o qual utiliza a entropia como medida da impureza. O algoritmo ID3 (QUINLAN, 1986), utiliza essa métrica. Para determinar quão boa é uma condição de teste realizada, é necessário comparar o grau de entropia do nodo-pai (antes da divisão) com o grau de entropia dos nodos-filhos (após a divisão). O atributo que gerar uma maior diferença é escolhido como condição de teste. O ganho é definido pela Equação (1), na forma:

$$\text{ganho} = \text{entropia}(\text{pai}) - \sum_{j=1}^{n}\left[\frac{N(v_j)}{N}\text{entropia}(v_j)\right] \qquad (1)$$

Onde n é o número de valores dos nodo-filhos, N é o número total de objetos do nodo-pai e (vj) é o número de exemplos associados ao nodo-filho vj. O grau de entropia é definido pela Equação (2) a seguir:

$$\text{entropia}(\text{nó}) = -\sum_{i=1}^{c} p(i/\text{nó}) \cdot \log_2[p(i/\text{nó})] \qquad (2)$$

Onde p (i / nó) é a fração dos registros pertencentes à classe i no nó, e c é o número de classes. O atributo-teste que maximiza o ganho de informação é selecionado pelo critério de ganho. O grande problema ao se utilizar o ganho de informação é que ele dá preferência a atributos com muitos valores possíveis (BRAMER, 2013).

Um caso clássico desse problema aconteceria ao utilizar um atributo insignificante (como por exemplo, o código de matrícula de um aluno). Nesse exemplo, seria criado um nodo para cada valor possível, e o total de nodos seria igual ao número de identificadores. Cada um desses nodos teria apenas um exemplo, o qual pertence a uma única classe, ou seja, os exemplos seriam totalmente discriminados. Assim, o valor da entropia seria mínimo porque, em cada nó, todos os exemplos pertencem à mesma classe. Essa divisão geraria um ganho máximo, embora seja totalmente inútil.

A razão de ganho, da sigla em Inglês (*Gain Ratio*), foi proposta por QUINLAN (1993) para solucionar o problema do ganho de informação. Ela nada mais é do que o ganho de informação relativo (ponderado) como critério de avaliação. A razão de ganho é definida pela Equação (3), na forma:

$$\text{razão\_de\_ganho}(\text{nó}) = \frac{\text{ganho}}{\text{entropia}(\text{nó})} \qquad (3)$$

É possível perceber pela Equação (3), que a razão de ganho não é definida quando o denominador é igual a zero. Além disso, favorece atributos cujo denominador, ou seja, a entropia, possui valor pequeno. Em Quinlan (1988), é sugerido que a razão de ganho seja realizada em duas etapas.

Primeiramente calculando o ganho de informação para todos os atributos. Após isso, considerar apenas aqueles atributos que obtiveram um ganho de informação acima da média, e então escolher aquele que apresentar a melhor razão de ganho (BASGALUPP, 2010).

Gini é outra medida bastante conhecida, a qual emprega um índice de dispersão estatística proposto por Corrado Gini em 1912. Este índice é muito

utilizado em análises econômicas e sociais, por exemplo, para quantificar a distribuição de renda em um certo país.

O algoritmo CART (BREIMAN *et al*., 1994) utiliza essa medida. Para um problema de c classes, o gini é definido pela Equação (4), na forma:

$$gini_{index}(\text{nó}) = 1 - \sum_{i=1}^{c} p(i / \text{nó}) \qquad (4)$$

Como no cálculo do ganho de informação, basta calcular a diferença entre o gini antes e após a divisão. Essa diferença, Gini, é representada pela Equação (5):

$$Gini = gini_{index}(\text{pai}) - \sum_{j=1}^{n} \left[\frac{N(v_j)}{N} gini_{index}(v_j) \right] \qquad (5)$$

Onde n é o número de valores do atributo (número de nodos-filhos), N é o número total de objetos do nodo-pai e (Nvj) é o número de exemplos associados ao nodo-filho vj.

4.2.1.4 Atributos categóricos

O desempenho das árvores de decisão induzidas é influenciado de maneira decisiva pela forma de representação dos nodos. Existem diferentes tipos de representação dos nodos para a divisão dos dados, dependendo do tipo de atributo. A seguir, são apresentadas algumas das formas de representação considerando atributos categóricos não-ordinais e ordinais (BRAMER, 2013).

Um ramo por valor de atributo: Uma aresta é criada para cada valor do atributo usado como condição de teste. Embora esse tipo de partição permita extrair do atributo todo o seu conteúdo informativo, possui a desvantagem de tornar a árvore de decisão mais complexa. O algoritmo C4.5 Quinlan (1993) utiliza esse tipo de divisão para atributos categóricos não ordinais.

Atributos categóricos ordinais: Conforme visto nos capítulos anteriores, um atributo é ordinal quando há uma relação de ordem entre os seus possíveis valores. Por exemplo, tem-se um atributo renda que pode possuir os valores ⟨baixa⟩, ⟨média⟩ e ⟨alta⟩. Com atributos desse tipo, é possível realizar uma partição binária do tipo renda < ⟨média⟩, em que todos os exemplos cujo atributo renda tem valor ⟨baixa⟩ seguem por uma aresta e os outros seguem por outra aresta. O algoritmo CART (BREIMAN *et al*., 1994) utiliza esse tipo de partição.

Valores agrupados em dois conjuntos: A divisão binária também pode ser realizada de uma forma mais complexa de acordo com Breiman *et al*. (1994), onde cada um dos dois subconjuntos pode ser formado por registros com mais de um valor para o atributo utilizado como condição de teste. O elevado custo computacional para encontrar a melhor divisão é o grande desafio desse tipo de divisão, pois o número de combinações possíveis é $(2^{n-1}$ - 1), onde n é o número de valores possíveis para o atributo em questão.

Valores agrupados em vários conjuntos: O algoritmo C4.5, Quinlan (1993), gera uma solução de boa qualidade no intuito de permitir o agrupamento de valores em diversos conjuntos com uma complexidade de cálculo razoável. Para isso, inicia criando uma aresta para cada valor do atributo em teste. Após, são verificadas todas as combinações possíveis de dois valores e, caso nenhuma dessas combinações produza um ganho maior que a divisão anterior, o processo é interrompido e a divisão anterior é adotada. Caso contrário, o processo é repetido tendo como base a melhor das soluções anteriores. Percebe-se que não se pode garantir que a divisão encontrada seja a melhor possível, pois é verificado se houve melhoria apenas um passo à frente.

4.2.1.5 Atributos contínuos

Alguns dos testes mais utilizados para partição de atributos contínuos são: testes simples ou pesquisa exaustiva e os testes múltiplos. Os testes múltiplos podem ser de segmentação global ou segmentação ao nível do nó. Fonseca (1994).

Os atributos contínuos permitem uma maior variedade de testes e, consequentemente, implicam uma maior complexidade de cálculo.

O teste simples, também conhecido como pesquisa exaustiva, é o mais utilizado. Um dos algoritmos que o utiliza é o C4.5, e a divisão é sempre binária. Supondo um atributo contínuo Z a ser utilizado como nó teste, mesmo que seu domínio seja infinito, o número de exemplos num conjunto de treinamento Q é finito e, portanto, o número de valores diferentes para esse atributo também é finito.

4.2.1.6 Métodos de poda em árvores de decisão

Um cuidado que se deve ter com árvores de decisão é o crescimento exagerado da árvore. Caso isso ocorra, deve-se contornar a situação com a operação denominada poda da árvore de decisão. Esta operação consiste em substituir os nodos profundos por folhas, removendo as ligações que fornecem um baixo valor de ganho de informação.

Existem diversas formas de realizar poda em uma árvore de decisão, e todas elas são classificadas como pré-poda ou pós-poda (BASGALUPP, 2010).

O método pré-poda é realizado durante o processo de construção da árvore, em que o processo pode simplesmente parar de dividir o conjunto de elementos e transformar o nodo corrente em um nodo folha da árvore.

Um critério de poda que pode ser utilizado é o ganho de informação. Caso todas as divisões possíveis utilizando um atributo Z gerem ganhos menores que um valor pré-estabelecido, então esse nodo vira folha, representando a classe mais frequente no conjunto de dados.

O método pós-poda é realizado após a construção da árvore de decisão, removendo ramos completos, onde tudo que está abaixo de um nodo interno é excluído e esse nodo é transformado em folha, representando a classe mais frequente no ramo.

Para cada nodo interno da árvore, o algoritmo calcula a taxa de erro caso a sub-árvore abaixo desse nó seja podada. Em seguida, é calculada a taxa de erro caso não haja a poda. Se a diferença entre essas duas taxas de erro for menor que um valor predeterminado, a árvore é podada. Caso contrário, não ocorre a poda (BASGALUPP, 2010).

4.2.1.7 Super ajuste ou *Overfiting*

No momento da construção das árvores de decisão, muitas das arestas ou sub-árvores podem refletir ruídos ou erros. Isso acarreta em um problema conhecido como sobre ajuste, que significa um aprendizado muito específico do conjunto de treinamento, não permitindo ao modelo generalizar.

Os erros mais cometidos por um modelo de classificação são geralmente divididos em dois tipos: erros de treinamento e erro de generalização (BASGALUPP, 2010). Erros de treinamento são o número de erros de classificação equivocada contida nos dados de treinamento, enquanto erros de generalização são os erros esperados pelo modelo em dados não vistos anteriormente.

Um bom modelo de classificação deve não apenas se adaptar bem aos dados de treinamento, como também deve classificar com precisão os registros nunca vistos antes por ele. Em outras palavras, um bom modelo deve ter baixa quantidade de erros de treinamento assim como de erros de generalização.

Isto é importante porque um modelo que seja apropriado aos dados de treinamento pode muito bem ter um erro de generalização mais pobre do que um modelo com alto grau de erro de treinamento (BASGALUPP, 2010). Tal situação é conhecida como *overfiting* do modelo.

4.3 Algoritmos de árvores de decisão

Nesta sessão, serão apresentados de forma sucinta os cinco principais algoritmos para indução de árvores de decisão. Os algoritmos em estudo são: *CART* Breiman *et al*. (1994), *ID3* Quinlan (1986) e *C4.5* Quinlan (1993), *Apriori* Agrawal *et al*. (1996), *FP-Growth* TAN *et al*. (2009).

4.3.1 Algoritmo CART

O algoritmo CART (*Classification and Regression Trees*) foi proposto em Breiman *et al*. (1994) e consiste em uma técnica que induz tanto árvores de classificação quanto árvores de regressão, dependendo se o atributo é nominal (classificação) ou contínuo (regressão).

Uma das suas principais vantagens é a grande capacidade de pesquisa de relações entre os dados, mesmo quando elas não são evidentes, bem como a produção de resultados sob a forma de árvores de decisão de grande simplicidade e legibilidade (CAMPELO, 1994).

O algoritmo CART gera árvores binárias, as quais podem ser percorridas da sua raiz até as folhas respondendo apenas a questões simples do tipo “sim” ou “não” (BASGALUPP, 2010).

Os nodos que correspondem a atributos contínuos são representados por agrupamento de valores em dois conjuntos. Utiliza a técnica de pesquisa exaustiva para definir os limiares a serem utilizados nos nodos para dividir os atributos contínuos. Também dispõe de um tratamento especial para atributos ordenados, além de permitir a utilização de combinações lineares entre atributos, ou seja, agrupamento de valores em vários conjuntos. Diferente das abordagens adotadas por outros algoritmos, os quais utilizam pré-poda, o CART expande a árvore exaustivamente, realizando pós-poda por meio da redução do fator complexidade-custo (BREIMAN *et al*., 1994).

4.3.2 Algoritmo ID3

O ID3, Quinlan (1986) é o algoritmo pioneiro em indução de árvores de decisão. É um algoritmo recursivo, baseado em busca gulosa, procurando, sobre um conjunto de atributos, aqueles que “melhor” dividem os dados, gerando sub-árvores. A partir de um conjunto de dados, ele constrói árvores de decisão, sendo a árvore resultante usada para classificar amostras futuras.

O ID3 separa um conjunto de treinamento em subconjuntos, de forma que estes contenham exemplos de uma única classe. A divisão é efetuada através de um único atributo, utilizando o ganho de informação para medir quanto informativo é um atributo (DANKEL, 1997).

O algoritmo ID3 só lida com atributos categóricos não-ordinais, não sendo possível apresentar a ele conjuntos de dados com atributos contínuos, por exemplo. Nesse caso, os atributos contínuos devem ser previamente discretizados. Além disso, o algoritmo ID3 também não apresenta nenhuma forma para tratar valores desconhecidos, ou seja, todos os exemplos do conjunto de treinamento devem ter valores conhecidos para todos os seus atributos, isso acaba tornando necessário gastar um bom tempo com pré-processamento dos dados para utilizar este algoritmo (BASGALUPP, 2010).

O ganho de informação é utilizado pelo ID3 para selecionar a melhor divisão. No entanto, esse critério não considera o número de divisões (número de arestas), e isso pode acarretar em árvores mais complexas. Além disso, o ID3 também não apresenta nenhum método de pós-poda, o que poderia amenizar esse problema de árvores mais complexas.

4.3.3 Algoritmo C4.5 ou J48

O algoritmo C4.5 foi criado por Quinlan (1993), sendo um dos algoritmos mais utilizados para a construção de árvores de decisão. Representa uma significativa evolução do ID3 do mesmo autor.

Consegue trabalhar tanto com atributos categóricos (ordinais ou não-ordinais) como com atributos contínuos. Para tratar com atributos contínuos, o algoritmo C4.5 define um limiar e então divide os exemplos de forma binária: aqueles cujo valor do atributo é maior que o limiar e aqueles cujo valor do atributo é menor ou igual ao limiar (BASGALUPP, 2010).

Consegue tratar valores desconhecidos. Esse algoritmo permite que os valores desconhecidos para um determinado atributo sejam representados como ‘?’, e o algoritmo trata esses valores de forma especial. Esses valores não são utilizados nos cálculos de ganho e entropia.

Utiliza a medida de razão de ganho para selecionar o atributo que melhor divide os dados. Essa medida se mostrou superior ao ganho de informação, gerando árvores mais precisas e menos complexas.

Apresenta um método de pós-poda das árvores geradas. Ele faz uma busca na árvore, de baixo para cima, e transforma em nós folha aqueles ramos que não apresentam nenhum ganho significativo (BASGALUPP, 2010).

4.3.4 Algoritmo *Apriori*

O algoritmo *Apriori* (AG’RAWAL *et al*., 1996) consolidou-se como o primeiro algoritmo de mineração de regras de associação considerado eficiente. Esse algoritmo combina uma estratégia de busca denominada *Breadth-first search* (BFS) com uma estrutura de árvore para contagem de ocorrência de candidatos. O

princípio *Apriori* parte do pressuposto de que se um conjunto de itens é frequente, então todos os seus subconjuntos também devem ser frequentes.

Um conjunto de itens é considerado frequente se o seu suporte for maior ou igual ao limiar *minsup* que é definido pelo usuário antes da execução do algoritmo. Um dos maiores problemas dessa abordagem é justamente a contagem do suporte para cada conjunto de itens, que cresce de forma exponencial (TAN *et al*., 2009).

Apriori foi o primeiro algoritmo de mineração de regras a usar a poda baseada em suporte para controlar o crescimento exponencial dos conjuntos de itens frequentes, ou seja, se um conjunto de itens possuir um suporte menor que o *minsup*, então todos os seus subconjuntos também serão menores, eliminando assim todos esses conjuntos (Agrawal *et al*., 1996).

O processo que determina a frequência com que ocorrem cada conjunto de itens frequentes é a contagem de suporte. Este processo apresenta uma abordagem computacionalmente custosa, principalmente quando os números de transações e conjuntos de itens candidatos a itens frequentes forem grandes (TAN *et al*., 2009).

Este algoritmo utiliza uma abordagem de níveis para gerar regras de associação, onde cada nível corresponde ao número de itens que pertence ao resultado da regra. Inicialmente, regras de confiança alta que tenham apenas um item no resultado da regra são extraídas. Por exemplo, se o conjunto {1,2,3} -> {4} e {1,4,3} -> {2} forem regras de confiança alta, então a regra candidata {1,3} -> {4,2} é criada pela fusão dos resultados de ambas as regras.

4.3.5 Algoritmo FP-Growth

O algoritmo FP-growth (TAN *et al*., 2009) apresenta uma diferença bastante significativa do algoritmo *Apriori*, por não utilizar a técnica gerar e testar utilizada no *Apriori*. O FP-growth utiliza uma árvore FP compacta para codificar o conjunto de dados e extrai o conjunto de itens frequente diretamente desta estrutura.

Uma árvore FP é uma representação compactada dos dados de entrada, construída através da leitura do conjunto de dados de entrada mapeando transação por transação em um caminho na árvore. Como diversas transações podem conter muitos itens em comum, em muitos casos os caminhos dessas transações podem ser sobrepostos, quanto mais caminhos forem sobrepostos, maior a compreensão que se pode obter da árvore FP (TAN *et al*., 2009).

O algoritmo FP-growth explora a árvore FP de baixo para cima para gerar os conjuntos de itens frequentes. Utiliza esta estratégia de *bottom-up* para encontrar conjuntos de itens frequentes terminados em um determinado item em particular, uma vez que cada transação é mapeada para um caminho na árvore, podendo assim examinar apenas os caminhos que contenham o nodo especificado.

O algoritmo encontra todos os grupos de itens frequentes terminados em um sufixo definido utilizando a estratégia de dividir para conquistar, dividindo o problema em pequenos subproblemas (TAN *et al*., 2009). É um algoritmo interessante, por que mostra como uma representação compacta do conjunto de dados da transação ajuda a gerar conjuntos de itens frequentes de modo eficiente.

.

5 FERRAMENTAS DE MINERAÇÃO DE DADOS

Este capítulo tem como objetivo apresentar algumas das principais ferramentas utilizadas na área de mineração de dados para realizar o processo de descoberta de conhecimento no contexto educacional, sendo selecionado os dois principais programas de mineração destacados pelos trabalhos relacionados.

Existem inúmeras ferramentas de mineração de dados disponíveis, de uso comercial e acadêmico, que fornecem as mais variadas coleções de algoritmos de mineração, algoritmos de pré-processamento, técnicas de visualização de dados, entre outras funcionalidades. Entre elas podemos citar o Oracle Data Miner, DBMiner, Clementine, IBM Intelligent Miner, WEKA, Hall *et al*. (2009) e RAPIDMINER (RAPIDMINER, 2014).

Apesar de muitos trabalhos e pesquisas sobre a mineração de dados educacionais e todos os esforços em propor e construir ferramentas de mineração que levem em conta as particularidades da mineração no âmbito educacional, duas dessas ferramentas são as que mais se destacam neste contexto: RAPIDMINER e WEKA. Por este motivo, essas ferramentas foram escolhidas e serão apresentadas de forma sucinta nas subseções a seguir.

5.1 RAPIDMINER

O RAPIDMINER é um sistema para a mineração de dados de código aberto. É um aplicativo distribuído de forma independente para análise de dados, mineração de texto e de dados, além disso, permite a integração com outros produtos desenvolvidos pelo mesmo projeto.

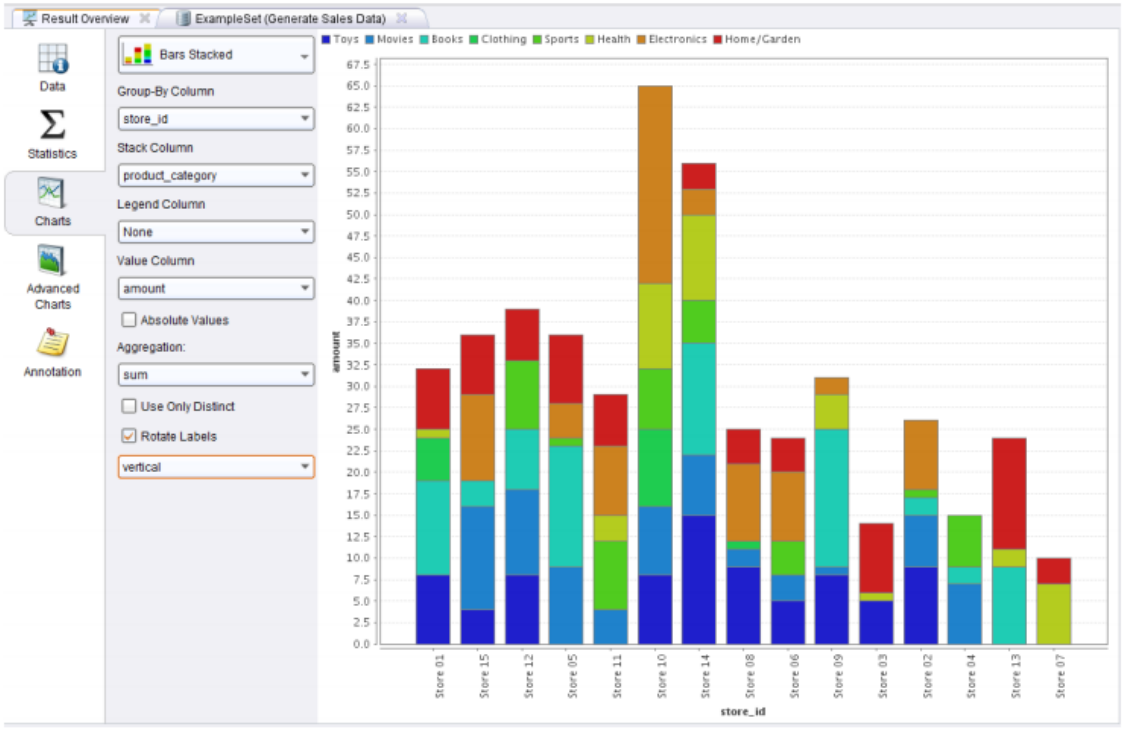
O fato de possuir código aberto e, por isso, ser disponibilizada gratuitamente, é uma das características interessantes dessa ferramenta, além de funcionar na maioria das principais plataformas e sistemas operacionais. Disponibiliza o acesso as suas funcionalidades por meio de uma interface gráfica intuitiva, linhas de comando e API Java, possibilitando a construção de aplicações que a utilizem por meio de um mecanismo simples.

Além disso, o RAPIDMINER possui a biblioteca de algoritmos de aprendizagem do WEKA totalmente integrada e possibilita o acesso a diferentes

fontes de dados, como: Excel, Acess, Oracle, Microsoft SQL Server, MySQL, Postgres, Arquivos de texto, entre outros.

Na Figura 09 é exibida uma de suas telas, onde a base de dados carregada pode ser visualizada.

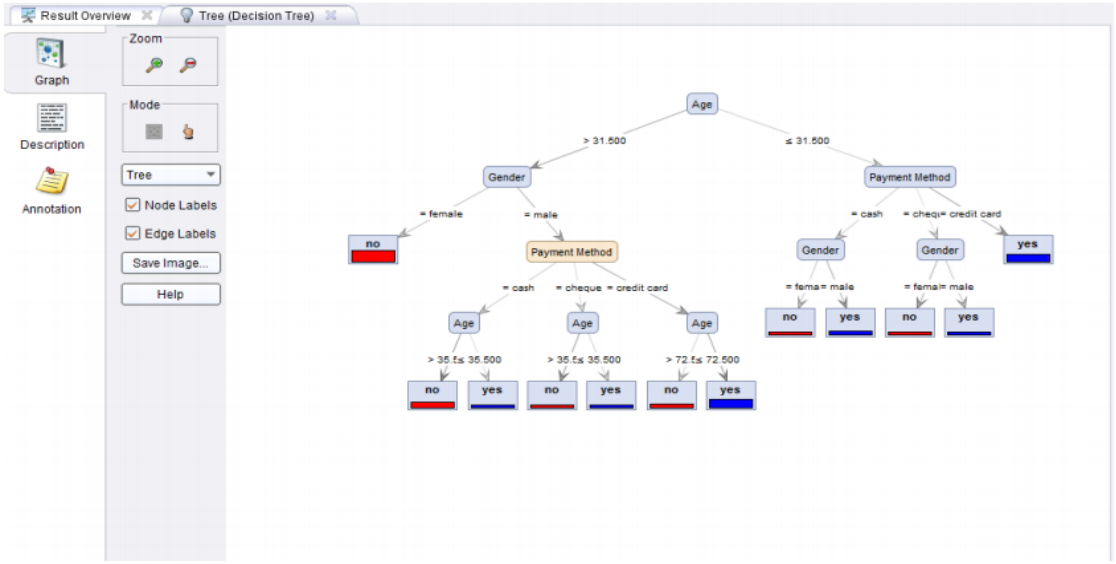
Figura 09 - Interface gráfica do RAPIDMINER



Fonte: Manual do usuário do RAPIDMINER.

Na Figura 10 é apresentada outra de suas telas onde pode ser vista uma árvore de decisão em uma representação gráfica.

Figura 10 - Árvore de decisão de exemplo no RAPIDMINER



Fonte: Manual de usuário do RAPIDMINER.

Além das características apresentadas, são destacadas outras características por desenvolvedores (RAPIDMINER, 2014) como diferenciais da ferramenta. Com mais de 500 operadores de integração e transformação dos dados, mineração, avaliação, visualização; e conceito visualização multicamadas de dados garante a manipulação de dados mais eficiente, entre outras.

O RAPIDMINER oferece uma vasta documentação incluindo tutorias em vídeos, um manual da ferramenta além de guia de instalação. O manual é muito bem elaborado e aborda, além de um passo-a-passo de como utilizar a ferramenta, uma introdução aos conceitos fundamentais e necessários sobre mineração de dados.

5.2 WEKA

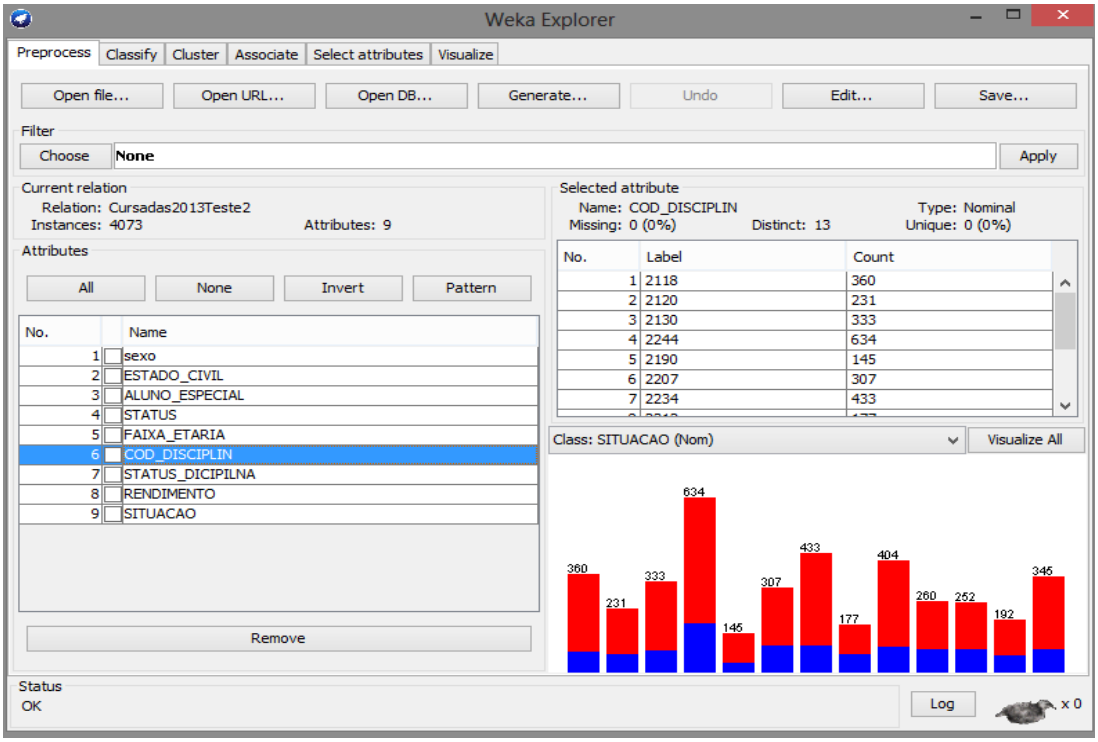
Muito popular no meio acadêmico, o WEKA (*Waikato Environment for Knowledge Analysis*) é um software livre escrito em Java e desenvolvido pela Universidade de Waikato, Nova Zelândia, formado por uma coleção de algoritmos de diversas técnicas de mineração de dados e ferramentas de pré-processamento (HALL *et al*., 2009).

O WEKA também oferece suporte a todo processo de mineração de dados, que inclui suporte à preparação dos dados de entrada, avaliação estatística da aprendizagem, visualização dos dados de entrada e os resultados. Todas as funcionalidades do programa podem ser acessadas através de uma interface simples e intuitiva.

Nesta interface, os algoritmos de aprendizagem e as diversas ferramentas para transformação podem ser aplicados diretamente nas bases de dados sem que seja necessário escrever nenhum código. O WEKA incorpora os principais métodos para as mais diversas áreas de aplicação da mineração de dados, estre elas a classificação, regressão, regras de associação, agrupamento e seleção de atributos (HALL *et al*. 2009). O formato padrão estabelecido para o WEKA é o ARFF, que é um formato de entrada específico da ferramenta e tem a forma de uma tabela relacional simples. O ARFF pode ser construído a partir de uma base de dados ou carregado de um arquivo.

A Figura 11 é exibida a interface gráfica do WEKA onde a base de dados carregada pode ser visualizada e melhor analisada.

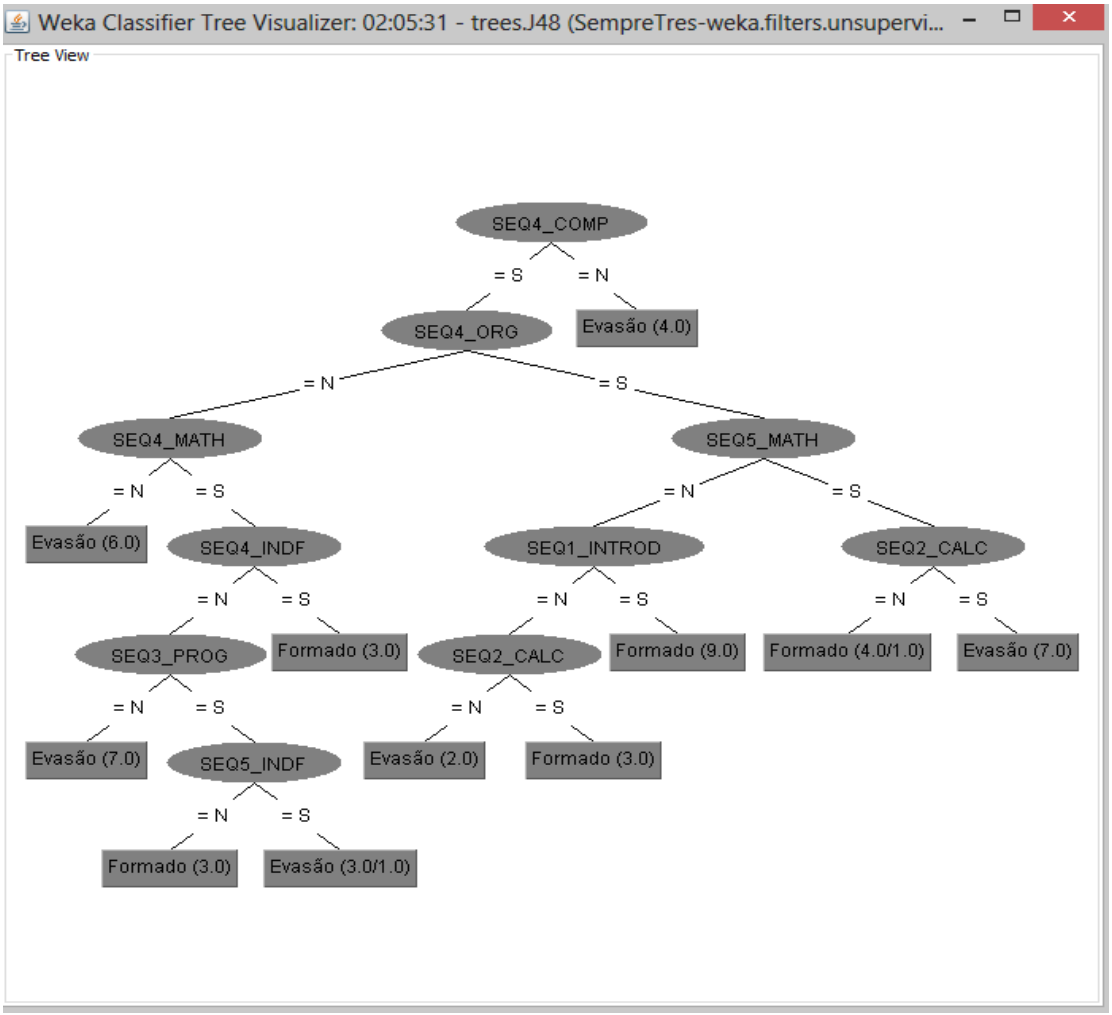
Figura 11 - Interface gráfica do WEKA com dados carregados para análise



Fonte: Hall *et al*. 2009. Adaptado pelo autor.

Na Figura 12 é apresentada uma representação gráfica de uma árvore de decisão de exemplo gerada pelo algoritmo C4.5 ou J4.8 como ele é chamado no WEKA.

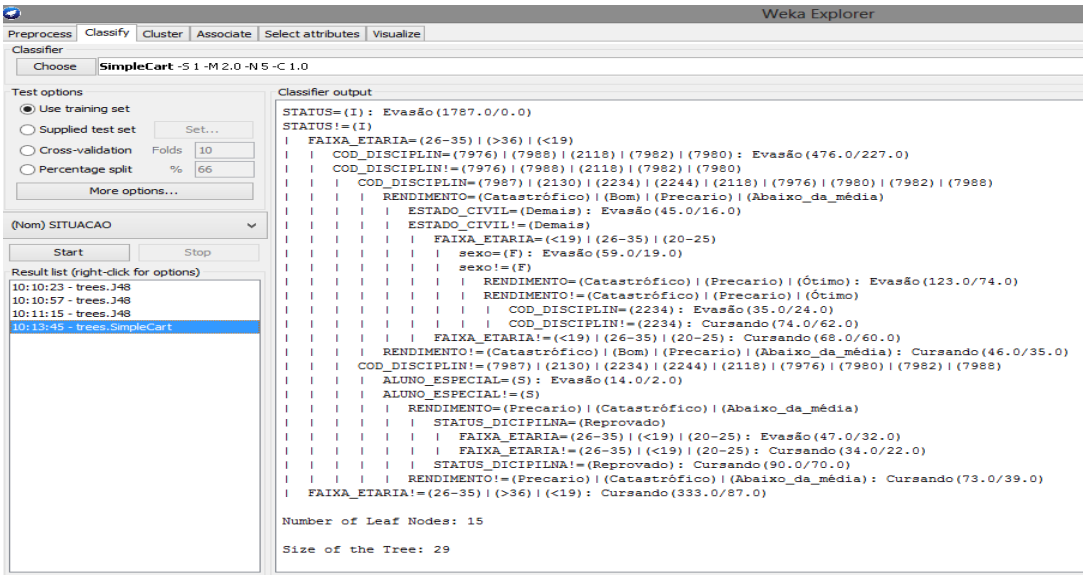
Figura 12 - Árvore gerada pelo algoritmo J48 no WEKA



Fonte: WEKA. Adaptado pelo autor.

A Figura 13 exibe outra forma de representação gráfica de uma árvore de decisão gerada no WEKA, desta vez gera pelo algoritmo CART.

Figura 13 - Árvore gerada pelo algoritmo CART no WEKA



Fonte: Hall *et al*. 2009. Adaptado pelo autor.

O WEKA também oferece sua API Java, que permite a construção de aplicações que utilizem todas as funcionalidades disponibilizadas pela ferramenta. Esta API também oferece suporte a construção do arquivo ARFF que é o formato específico aceito pelos algoritmos implementados pela ferramenta.

Além disso, fornece uma vasta documentação on-line, mantém uma comunidade de entusiastas que mantém o projeto atualizado e em constante evolução, disponibiliza todo o material produzido pelo projeto, como o livro *Data Mining Practical Machine Learning Tools and Techniques* (HALL *et al*., 2009).

6 TRABALHOS RELACIONADOS

Neste capítulo, serão discutidos os trabalhos relacionados ao assunto da evasão no ensino superior, foco principal deste trabalho.

O desempenho acadêmico é um tema amplamente pesquisado, sendo estudado por diversos pesquisadores há algum tempo. Tinto, 1975, definiu um modelo teórico para explicar as causas da evasão discente, considerando o processo de desgaste do aluno como uma interação sócio-psicológica entre as características do aluno universitário e a sua experiência no instituto de ensino.

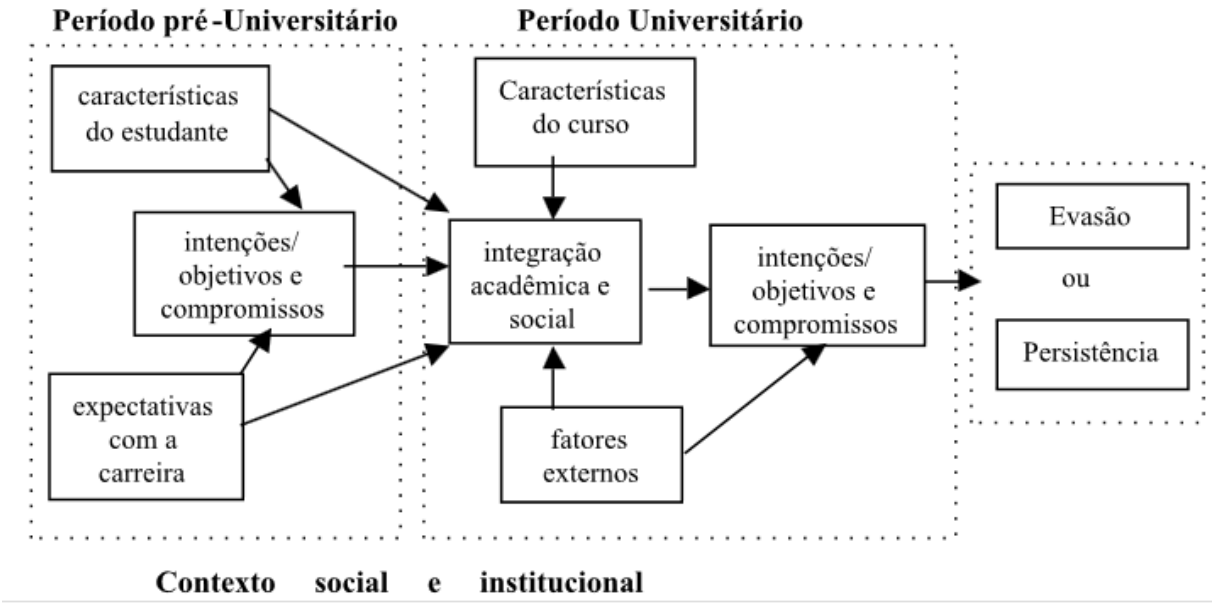
Essa interação entre o passado do aluno e do ambiente acadêmico leva a um grau de integração do aluno para este novo ambiente, sugerindo dessa forma, que o aluno deixa a universidade por problemas causados pela falta de integração com o ambiente acadêmico e social da instituição (RIGO; CAZELLA; CAMBRUZZI, 2012).

Tinto (1987) aperfeiçoou seu modelo e posteriormente outros estudos concluem que essa integração é influenciada, direta ou indiretamente, por características demográficas do discente, tais como: nível sócio-econômico da família, expectativa dos pais a respeito do futuro do filho, habilidades acadêmicas do aprendiz, conhecimentos adquiridos através da educação formal e/ou informal (ANDRIOLA et. al., 2006). Ambos os modelos envolvem a condição social do aluno, atributos como idade, gênero, experiências escolares, habilidades passoais, juntamente com suas expectativas de desenvolvimento pessoal e de carreira, associadas com a motivação para o desempenho acadêmico e o seu reconhecimento.

Estas características são consideradas dentro de um espaço temporal, de modo que a importância e influência de cada uma delas muda de acordo com o tempo no ambiente universitário (RIGO; CAZELLA; CAMBRUZZI, 2012).

A Figura 14 mostra resumidamente este modelo, destacando tanto a influência de aspectos pessoais e sociais existentes antes do ingresso no curso universitário, como aspectos relacionados com o contato acadêmico, metodologia de aprendizagem e integração institucional.

Figura 14 - Modelo proposto por Tinto (1975)



Fonte: (ANDRIOLA *et al*, 2006).

Métodos estatísticos também foram amplamente utilizados para a melhor compreensão do problema da evasão. Johnston (1997) realizou uma análise de registros dos alunos de uma universidade Escocesa, mostrando que mais de um quarto dos alunos que ingressava nos cursos de graduação não estava progredindo ou estava se retirando dos mesmos. A pesquisa sugere que os problemas não acadêmicos são mais suscetíveis de contribuir para o fracasso de um aluno do que problemas acadêmicos e que a gama de problemas não acadêmicos é muito mais complexa e ampla. Além disso, a percepção pessoal do grau de influência exercido por esses problemas nem sempre foi acompanhada pela incidência registrada na pesquisa.

Deve ser destacada a necessidade de avaliação cuidadosa de modelos como os descritos acima, devido à grande dinamicidade observada em relação a estes fatores e seus efeitos (RIGO; CAZELLA; CAMBRUZZI, 2012). Fatores tais como aspectos sociais podem ser superados a partir de níveis motivacionais ou expectativas de carreira, bem como aspectos metodológicos e ações pedagógicas podem ser associados com fatores motivacionais (ADACHI, 2009).

Em sua grande maioria, os trabalhos relacionados à utilização de técnicas de mineração de dados sobre dados educacionais está restrita a identificar

resultados em pequenos contextos, relativos a apenas algumas disciplinas e na maior parte dos casos estão ligados a cursos não presenciais.

Regras de associação foram utilizadas por Minaei-Bidgoli *et al*. (2006), para identificar padrões de informações em bases de dados geradas a partir de sistemas educacionais *online*, utilizando como base a disciplina de Física onde os autores demostraram que um conjunto de regras permite identificar quais os atributos que caracterizam padrão de desempenho dos grupos de alunos, neste caso, oferecida em ambiente *online*.

Um trabalho interessante foi realizado por Dekker *et al*. (2009), onde os autores consideram três conjuntos de dados: um conjunto com os dados de pré-universitários contendo apenas 495 casos (242 instâncias classificadas como sem sucesso, 253 casos classificados como sucesso), cada um descrito com 13 atributos. Um conjunto de dados das notas de alunos de curso universitário contendo apenas 516 casos (253 instâncias classificadas como sem sucesso, 263 casos classificados como bem-sucedido), cada um descrito com 74 atributos (para cada uma das 37 disciplinas disponíveis, dois atributos dizendo quantas tentativas foram feitas, e qual foi a maior nota). Por fim, outro conjunto de dados com os outros dois conjuntos anteriores juntos. Conforme visto, o conjunto com os dados dos alunos universitários possuem apenas dois atributos (tentativa e nota mais alta), enquanto o conjunto com os dados pré-universitários possuíam 13 atributos.

Com o auxílio da ferramenta de mineração de dados WEKA, os autores testaram e compararam alguns algoritmos de mineração de dados nos conjuntos citados, dentre os quais pode-se destacar:

- Árvores de decisão CART (*SimpleCart*) e C4.5 (J48)
- Redes Bayesiana (*BayesNet*)
- Um modelo logístico (*SimpleLogistic*)
- Regras de associação (*JRip, OneR*)

Os resultados obtidos mostram que árvores de decisão ou classificadores bastante simples, podem gerar um resultado satisfatório com precisões entre 75 e 80% conforme podemos observar na Tabela 1.

Tabela 1 - Tabela de resultados comparando diferentes algoritmos

Classifiers	*OneR*	*CART*	*J48 -M 2*	*J48 -M 10*	*BayesNet*	*Logit*	*JRip*	*RF*
Accuracy	0.75	0.79	0.80 ○	0.80	0.75	0.79	0.77	0.79
True positives	0.64	0.79 ○	0.80 ○	0.75 ○	0.72 ○	0.79 ○	0.73 ○	0.82 ○
False negatives	0.36	0.21 ○	0.20 ○	0.25 ○	0.28 ○	0.21 ○	0.27 ○	0.18 ○
True negatives	0.86	0.80 ●	0.80 ●	0.84	0.79	0.80 ●	0.82	0.77 ●
False positives	0.14	0.20 ●	0.20 ●	0.16	0.21 ●	0.20 ●	0.18	0.23 ●

○, ● – statistically significant improvement or degradation

Fonte: (DEKKER *et al*., 2009).

A tabela 1 mostra a pontuação da acurácia dos algoritmos de mineração para obter uma melhor visão sobre o desempenho de classificadores. Pode-se constatar que o algoritmo *OneR* tem uma taxa de Falso negativo mais elevada do que todos os outros algoritmos. Esta é uma descoberta interessante, porque de acordo com os autores é melhor classificar como propenso a evasão erroneamente um aluno que deve realmente ser classificado como não propenso à evasão, do que dar um parecer negativo erroneamente a um aluno que deve ser classificado como positivo.

Um trabalho semelhante foi realizado por Manhães *et al*. (2011) onde os autores analisaram dados de alunos da Escola Politécnica da Universidade Federal do Rio de Janeiro – UFRJ no período de 1994 a 2005. A base de dados foi composta por informações de 543 alunos que concluíram o curso de Engenharia e mais 344 registros de alunos que não concluíram. Com o auxílio da ferramenta de mineração de dados WEKA, foram avaliadas técnicas através de três experimentos, onde foram aplicados dez algoritmos de classificação sobre a base de dados. Os experimentos retornaram dados com acurácia média variando entre 75 a 80%. Os desempenhos obtidos pelos algoritmos de mineração de dados dos mais simples aos mais sofisticados foram semelhantes. Também indicam que a previsão de alunos com risco de evasão pode ser feita a partir de um número reduzido de atributos, por exemplo, verificou-se que o atributo mais importante é o coeficiente de rendimento do primeiro semestre letivo, o segundo é a nota na disciplina de Cálculo Diferencial e Integral I.

Conforme Manhães et. al. (2011) este assunto necessita de muita pesquisa e discussão. Mineração de dados utilizada na busca de padrões de informação em base de dados educacionais a fim de identificar possíveis evasões e suas principais

causas, é um campo de investigação ainda não consolidado que necessita de investigações complementares tanto na definição dos atributos a serem utilizados quanto nas técnicas de mineração de dados empregadas.

Existem pontos que precisam ser pesquisados para aprimorar a utilização da mineração de dados como a transformação dos dados, a identificação dos atributos mais relevantes, qual a melhor técnica e quais os algoritmos são os mais propícios para serem usados em um determinado problema, entre outros (MANHÃES *et al*., 2011).

6.1 Análise dos trabalhos relacionados

A Tabela 2 apresenta uma comparação entre os trabalhos relacionados e este trabalho, quais as técnicas utilizadas nos experimentos, exibindo um conjunto de critérios definidos tais como: Objetivo do trabalho, quais os dados que foram disponibilizados além da tarefa, técnica e algoritmo de Mineração de dados utilizados.

Tabela 2 - Comparativo entre os Trabalhos Relacionados

Trabalho	Objetivo	Dados Utilizados	Tarefa Utilizada	Técnica	Algoritmo
Tinto, 1975	Explicar as causas da evasão discente	Pessoais, sociais e acadêmicos	Associação	Modelo teórico	-
Johnston, 1997	Descobrir por que mais de ¼ dos alunos evadem	Coletadas por acadêmicos, como líderes de curso	Classificação	Estatística	-
Minaei-Bidgoli *et al*. 2006	Identificar padrões em bases de dados de sistemas educacionais online	Pessoais e acadêmicos, tendo a disciplina de física como base	Associação	Regras de associação	*A Priori*
Dekker *et al*., 2009	Prever a evasão do curso de Engenharia Elétrica após o primeiro semestre	Dados pré-universitários e dados acadêmicos.	Classificação, Associação	Árvores de Decisão, Regras de associação, Redes Bayesianas	*SimpleCart, J48, BayesNet, SimpleLogistic, JRip, OneR*
Manhães *et al*., 2011	Identificar precocemente alunos em risco de evasão nos cursos de graduação	Dados da grade curricular do curso de Engenharia	Classificação, Associação	Árvores de Decisão, Regras de associação, Redes Bayesianas	*OneR, JRip, J48, RandomForest, SimpleLogistic, NaiveBayes, A Priori*

Fonte: Do Autor.

A partir dos estudos feitos nos trabalhos relacionados apresentados neste capítulo, pode-se destacar que a tarefa, em sua grande maioria, foi a de classificação de objetos alvos e predominou o uso da técnica de árvores de decisão em conjunto com o algoritmo J48, já no que se diz respeito ao uso de regras de associação, o algoritmo *Apriori* foi o que teve maior destaque.

Para a realização deste estudo, foram escolhidas as técnicas de classificação e associação para serem utilizadas nos experimentos.

Na tarefa de classificação será utilizado o algoritmo J48, assim como em Manhães e Dekker, o resultado da precisão deste algoritmo nos testes realizados durante a faze de familiarização do autor com a ferramenta de mineração se mostraram satisfatórios, ou seja, apresentaram uma boa acurácia.

Na tarefa de associação serão testados dois algoritmos diferentes, *Apriori* e *FPGrowth.* A escolha pelo algoritmo *Apriori* se dá em função da sua utilização nos trabalhos relacionados, já o *FPGrowth* será utilizado para testar o comportamento destes dois algoritmos quando aplicados no mesmo conjunto de dados.

A escolha pela ferramenta de mineração de dados WEKA, Hall *et al*. (2009) se deu em função de ser amplamente utilizado nos trabalhos relacionados, pela facilidade de uso, também por ela ser uma ferramenta livre de código aberto sem custos para utilização e de fácil integração com outras linguagens de programação como Java por exemplo, visando trabalhos futuros, além do grande volume de material escrito e exemplos sobre o mesmo.

7 METODOLOGIA E CONTEXTUALIZAÇAO DO PROBLEMA

Neste capítulo, são apresentados a metodologia utilizada no desenvolvimento do trabalho e a contextualização do fenômeno em estudo na Universidade de Santa Cruz do Sul - UNISC.

A pesquisa e revisão bibliográfica foi o ponto inicial deste trabalho, com base nela foram definidas as tecnologias e técnicas a serem adotadas para o desenvolvimento do estudo. Com relação aos objetivos, este trabalho se originou de uma pesquisa exploratória e explicativa. A pesquisa exploratória proporciona maior familiaridade com o tema, já a pesquisa explicativa, tem por objetivo identificar os fatores que determinam ou que contribuem para a ocorrência do fenômeno estudado (GIL, 2002).

O Curso de Ciência da Computação da UNISC possui um currículo com 55 disciplinas divididas em 9 semestres, totalizando uma carga horária de 3.210 horas, que permite ampla formação científica, cujo embasamento teórico-prático oportuniza, aos alunos, acompanharem novas e dinâmicas tendências em função da rápida evolução tecnológica neste campo. O perfil dos ingressantes no curso são principalmente alunos provenientes do ensino médio, através da realização de vestibular (Setor de informática da UNISC).

Como fonte de dados para este trabalho, foram utilizados dados acadêmicos relativos aos alunos matriculados no curso de ciência da computação da UNISC, preservando os dados pessoais dos alunos por não serem considerados importantes na descoberta por padrões da evasão.

Foram disponibilizados dados em forma de arquivo texto das tabelas: i) ALUNOS (matrícula, data de nascimento, sexo, estado civil, classificação e média no vestibular, ano de ingresso); ii) ALUNOS_CURSOS (matrícula, código do curso, tipo de movimentação do aluno, se ingresso ou desistência, data colação de grau); iii) DISCIPLINAS_CURSADAS (matrícula, ano que foi cursada a disciplina, código da disciplina, carga horária, frequência e notas obtidas); iv) ITENS_CONCLUIDOS (matrícula, código curso, código do currículo, tipo de conclusão da disciplina, créditos obtidos); v) OCORRENCIAS_ALUNOS (matrícula, código do curso, tipo - ingresso ou evasão - e sub tipo da ocorrência, data que o aluno evadiu ou ingressou no curso, ano e média do vestibular); vi) DISCIPLINAS (código da disciplina, nome da disciplina).

O primeiro passo do trabalho foi um estudo detalhado nos dados acadêmicos cedidos pelo setor de informática da UNISC, feito em duas etapas: Em um primeiro momento analisando diretamente os dados, após a importação dos mesmos em uma base de dados de apoio gerada no *Firebird*[2], com o objetivo de encontrar os atributos para melhor organizar os dados, de forma que contemplassem as informações históricas referentes aos alunos em seus períodos na universidade. Em um segundo momento, tratando os dados diretamente na ferramenta WEKA, onde foram submetidos aos algoritmos de mineração de dados selecionados.

Conforme visto nos trabalhos relacionados, a tarefa de classificação é uma das tarefas mais utilizadas na busca por padrões da evasão nas universidades (DEKKER *et al*., 2009; MANHÃES, 2011). Por isto, a classificação será considerada como tarefa base deste trabalho, juntamente com a técnica de árvores de decisão e o algoritmo C4.5 (J48) também amplamente utilizados neste contexto (BUSS, 2011; MANHÃES, 2011).

O método de classificação foi escolhido por melhor responder a questão alvo deste trabalho, ou seja, com o método de classificação pode-se prever que determinados alunos, com um determinado perfil, tendem a evadir do curso. Neste caso, o atributo STATUS é denominado como atributo alvo da classificação e, sobre este, regras de classificação em relação aos outros atributos serão geradas.

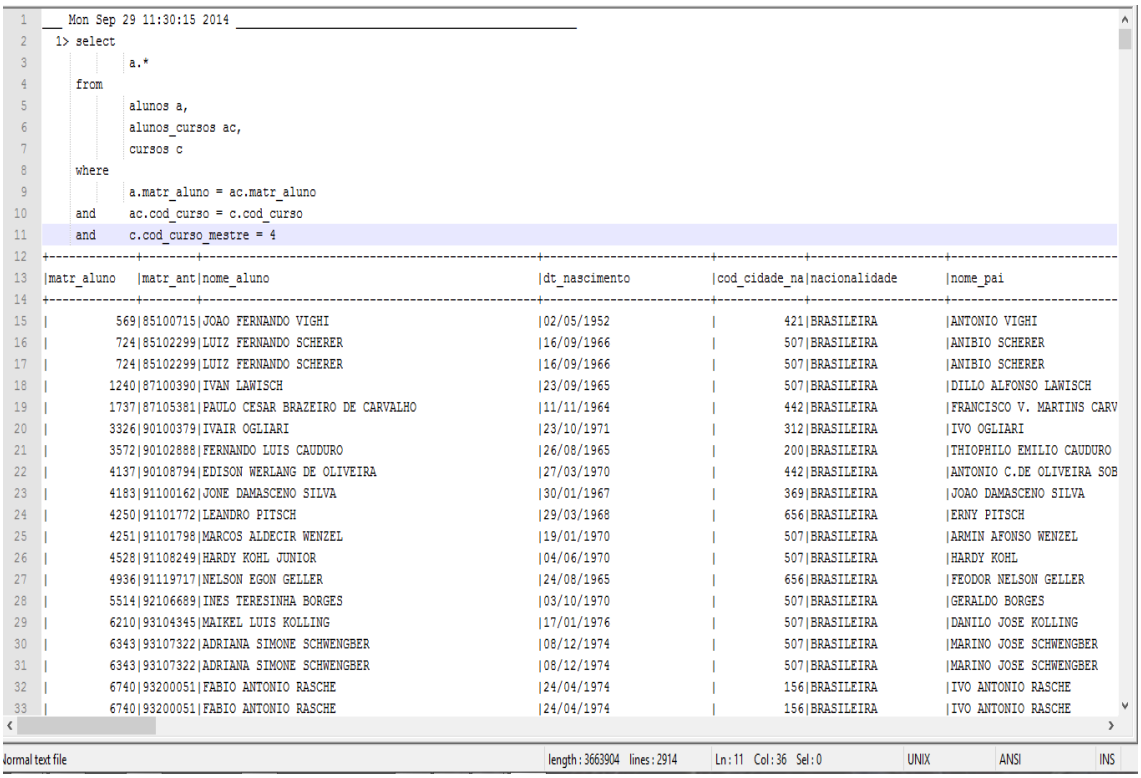
7.1 Domínio da base de dados de apoio

A base de dados utilizada neste trabalho foi diretamente coletada do sistema acadêmico da universidade e disponibilizada em forma de arquivo texto. Ela contém informações sobre os alunos de graduação que ingressaram no curso de Ciência da Computação, em um dos três currículos (códigos 186, 207 e 2509), além de informações sobre as disciplinas cursadas por eles.

Após a fase de seleção dos dados, o primeiro passo foi a importação dos arquivos texto para o banco de dados, a Figura 15 mostra o início de um dos arquivos texto recebidos, com a consulta SQL que originou os dados da tabela ALUNOS. Os demais arquivos texto e consultas podem ser vistos no Anexo A.

[2] Sistema gerenciador de banco de dados de código aberto baseado na plataforma do InterBase da Borland.

Figura 15 - Arquivo texto com a consulta SQL e os dados da tabela Alunos



```
___ Mon Sep 29 11:30:15 2014 ____________________________________________
 1> select
            a.*
    from
            alunos a,
            alunos_cursos ac,
            cursos c
    where
            a.matr_aluno = ac.matr_aluno
    and     ac.cod_curso = c.cod_curso
    and     c.cod_curso_mestre = 4
+-----------+--------+------------------------------------------+-----------------+-------------+--------------+-------------------------+
|matr_aluno |matr_ant|nome_aluno                                |dt_nascimento    |cod_cidade_na|nacionalidade |nome_pai                 |
+-----------+--------+------------------------------------------+-----------------+-------------+--------------+-------------------------+
|       569|85100715|JOAO FERNANDO VIGHI                       |02/05/1952       |          421|BRASILEIRA    |ANTONIO VIGHI            |
|       724|85102299|LUIZ FERNANDO SCHERER                     |16/09/1966       |          507|BRASILEIRA    |ANIBIO SCHERER           |
|       724|85102299|LUIZ FERNANDO SCHERER                     |16/09/1966       |          507|BRASILEIRA    |ANIBIO SCHERER           |
|      1240|87100390|IVAN LAWISCH                              |23/09/1965       |          507|BRASILEIRA    |DILLO ALFONSO LAWISCH    |
|      1737|87105381|PAULO CESAR BRAZEIRO DE CARVALHO          |11/11/1964       |          442|BRASILEIRA    |FRANCISCO V. MARTINS CARV|
|      3326|90100379|IVAIR OGLIARI                             |23/10/1971       |          312|BRASILEIRA    |IVO OGLIARI              |
|      3572|90102888|FERNANDO LUIS CAUDURO                     |26/08/1965       |          200|BRASILEIRA    |THIOPHILO EMILIO CAUDURO |
|      4137|90108794|EDISON WERLANG DE OLIVEIRA                |27/03/1970       |          442|BRASILEIRA    |ANTONIO C.DE OLIVEIRA SOB|
|      4183|91100162|JONE DAMASCENO SILVA                      |30/01/1967       |          369|BRASILEIRA    |JOAO DAMASCENO SILVA     |
|      4250|91101772|LEANDRO PITSCH                            |29/03/1968       |          656|BRASILEIRA    |ERNY PITSCH              |
|      4251|91101798|MARCOS ALDECIR WENZEL                     |19/01/1970       |          507|BRASILEIRA    |ARMIN AFONSO WENZEL      |
|      4528|91108249|HARDY KOHL JUNIOR                         |04/06/1970       |          507|BRASILEIRA    |HARDY KOHL               |
|      4936|91119717|NELSON EGON GELLER                        |24/08/1965       |          656|BRASILEIRA    |FEODOR NELSON GELLER     |
|      5514|92106689|INES TERESINHA BORGES                     |03/10/1970       |          507|BRASILEIRA    |GERALDO BORGES           |
|      6210|93104345|MAIKEL LUIS KOLLING                       |17/01/1976       |          507|BRASILEIRA    |DANILO JOSE KOLLING      |
|      6343|93107322|ADRIANA SIMONE SCHWENGBER                 |08/12/1974       |          507|BRASILEIRA    |MARINO JOSE SCHWENGBER   |
|      6343|93107322|ADRIANA SIMONE SCHWENGBER                 |08/12/1974       |          507|BRASILEIRA    |MARINO JOSE SCHWENGBER   |
|      6740|93200051|FABIO ANTONIO RASCHE                      |24/04/1974       |          156|BRASILEIRA    |IVO ANTONIO RASCHE       |
|      6740|93200051|FABIO ANTONIO RASCHE                      |24/04/1974       |          156|BRASILEIRA    |IVO ANTONIO RASCHE       |
Normal text file  length : 3663904  lines : 2914  Ln : 11  Col : 36  Sel : 0  UNIX  ANSI  INS
```

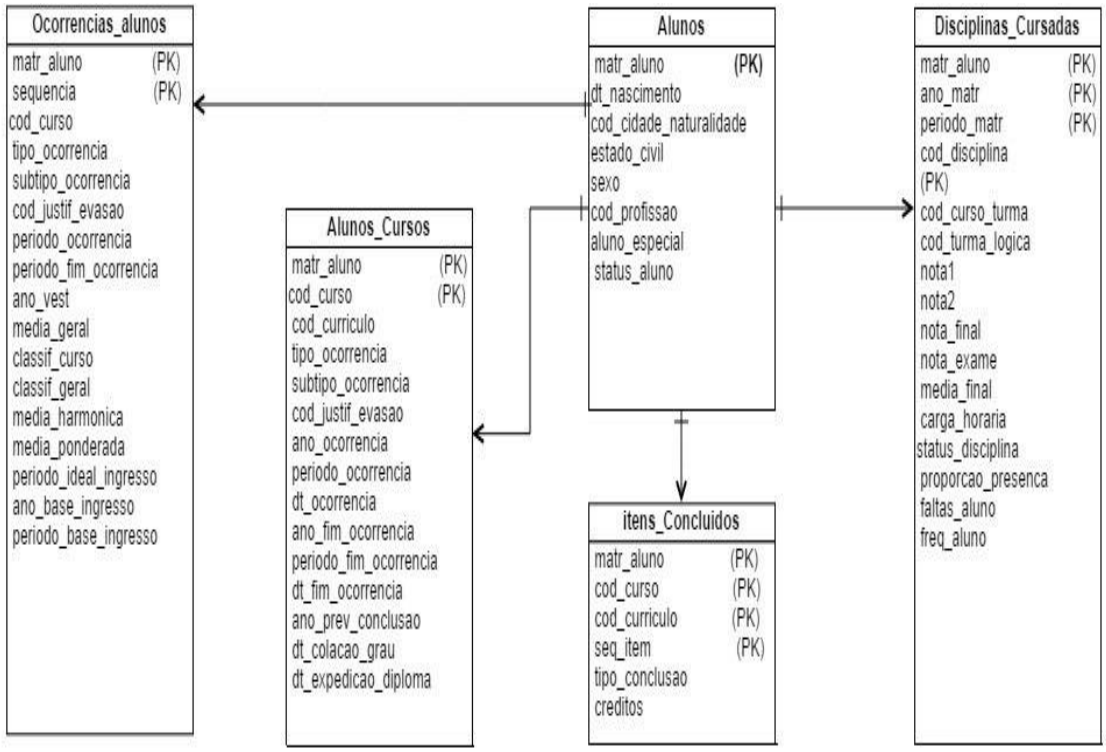
Fonte: Setor de informática da UNISC.

O Anexo B exibe as tabelas que foram recebidas na íntegra, com todos os atributos que foram disponibilizados nos arquivos texto para o estudo, uma breve explicação do significado de cada atributo e o domínio de valores aplicado a ele, caso exista. A tabela ALUNOS continha 70 atributos, a tabela ALUNOS_CURSOS continha 30 atributos, OCORRENCIAS_ALUNO com 31 atributos, DISCIPLINAS_CURSADAS com 17 e a tabela ITENS_CONCLUIDOS com 6 atributos.

Após a importação os dados, seguindo as etapas do processo de KDD, o passo seguinte foi a limpeza da base criada, exclusão dos atributos não considerados importantes para a mineração, como número de RG, CPF, nome do pai, nome da mãe, enfim, atributos que não são capazes de criar uma boa generalização na base de dados. Também foi realizada a remoção dos registros duplicados, como pode-se ver no anexo A, todas as tabelas disponibilizadas pelo setor de informática da UNISC sofreram fusão com outras tabelas, gerando registros duplicados em todas as tabelas importadas.

O modelo relacional da base de dados de apoio criada com os atributos que foram selecionados para o estudo pode ser visto na Figura 16.

Figura 16 - Modelo relacional da base de apoio criada



Fonte: Do Autor.

Com base no modelo relacional exibido na Figura 16, criou-se uma nova tabela HISTORICO com o objetivo de organizar os dados de forma que cada registro contenha o histórico de um aluno, armazenando o maior número de informações possível referente ao perfil do aluno no decorrer do curso, como por exemplo a idade que ele tinha quando ingressou, se ingressou via vestibular ou transferência interna/externa, se trocou ou não de currículo, quantas vezes trancou a matrícula, quantas disciplinas cursou em cada semestre, a ordem em que as disciplinas foram cursadas ordenadas por ano e semestre em que foram cursadas (para disciplinas cursadas no mesmo ano/semestre o critério de ordenação escolhido foi a ordenação numérica pelo código da disciplina).

Os atributos da tabela HISTORICO são descritos na Tabela 3, que apresenta o nome e uma breve descrição dos atributos criados.

Tabela 3 - Atributos da tabela HISTORICO

(continua)

Nome do atributo	Descrição do atributo
matr_aluno (pk)	Matrícula do Aluno
dt_nascimento	Data de nascimento
Sexo	Sexo (M – masculino, F – Feminino)
estado_civil	Estado Civil
codigo_curriculo_curso	Código do currículo em que o aluno está matriculado
tipo_movimentacao	Tipo de movimento do aluno 1 – Ingresso 2 - Evasão
subtipo_movimentacao	1-Vestibular, 2-Transferência Interna, 3-Transferência Externa, 4-Reingresso,5-Diplomado, 6-Aluno Especial,7-Conclusão, 8-Trancamento,9-Cancelamento,10-Desistência, 11-Suspensão,12-Desligamento,13-Falecimento,14-Cancelamento de Matrícula,15-Cancelamento de Rematrícula, 16-Ingresso em Curso Sequencial, 17-Permuta, 18-Trancamento para Vínculo,19-Reopção de Habilitação
meio_acesso	Idem anterior, porém esse armazena o registro da entrada do aluno, já o anterior armazena a situação atual do aluno
idade_ingresso	Com base na data de nascimento e a data da ocorrência da matricula, foi calculado a idade do aluno quando ingressou no curso
trocou_curriculo	Se o aluno trocou ou não de currículo no curso (S- sim, N - não)
qtde_trancamentos	Quantidade de trancamentos realizados
tempo_permanencia	Tempo em semestres que o aluno permaneceu ou permanece no curso, desde o primeiro registro de entrada
media_vestib	A média atingida no vestibular
class_vestib	A classificação no vestibular (classificação no curso, não geral)
total_disciplinas_cursadas	O número total de disciplinas cursadas pelo aluno
disc_01_cod	Código da primeira disciplina cursada
disc_01_ano	Ano que a primeira disciplina foi cursada
disc_01_sem	Semestre em que a primeira disciplina foi cursada
disc_01_status	Status do aluno na disciplina (1-Aprovado, 2-Reprovado,3-Desistente)
disc_01_notafinal	Média final da disciplina
disc_02_cod	Código da segunda disciplina cursada
disc_02_ano	Ano que a segunda disciplina foi cursada
disc_02_sem	Semestre em que a segunda disciplina foi cursada
disc_02_status	Status do aluno na disciplina (1-Aprovado, 2-Reprovado,3-Desistente)
disc_02_notafinal	Média final da disciplina
disc_03_cod	Código da disciplina cursada
disc_03_ano	Ano que a disciplina foi cursada
disc_03_sem	Semestre em que a disciplina foi cursada
disc_03_status	Status do aluno na disciplina (1-Aprovado, 2-Reprovado,3-Desistente)
disc_03_notafinal	Média final da disciplina
disc_04_cod	Código da disciplina cursada
disc_04_ano	Ano que a disciplina foi cursada
disc_04_sem	Semestre em que a disciplina foi cursada
disc_04_status	Status do aluno na disciplina (1-Aprovado, 2-Reprovado,3-Desistente)
disc_04_notafinal	Média final da disciplina
disc ... 05 até 54	Estes campos são repetidos, da disciplina 1 até 55
disc_55_cod	Código da disciplina cursada
disc_55_ano	Ano que a disciplina foi cursada
disc_55_sem	Semestre em que a disciplina foi cursada
disc_55_status	Status do aluno na disciplina (1-Aprovado, 2-Reprovado,3-Desistente)
disc_55_notafinal	Média final da disciplina
tot_disc_sem_01	Total de disciplinas cursadas no primeiro semestre
tot_disc_sem_02	Total de disciplinas cursadas no segundo semestre
tot_disc_sem_03	Total de disciplinas cursadas no terceiro semestre
tot_disc_sem_04	Total de disciplinas cursadas no quarto semestre
tot_disc_sem_05	Total de disciplinas cursadas no quinto semestre
tot_disc_sem_06	Total de disciplinas cursadas no sexto semestre
tot_disc_sem_07	Total de disciplinas cursadas no sétimo semestre

Tabela 3 - Atributos da tabela HISTORICO

(conclusão)

Nome do atributo	Descrição do atributo
tot_disc_sem_08	Total de disciplinas cursadas no oitavo semestre
tot_disc_sem_09	Total de disciplinas cursadas no nono semestre
tot_disc_sem_10	Total de disciplinas cursadas no décimo semestre

Fonte: Autor.

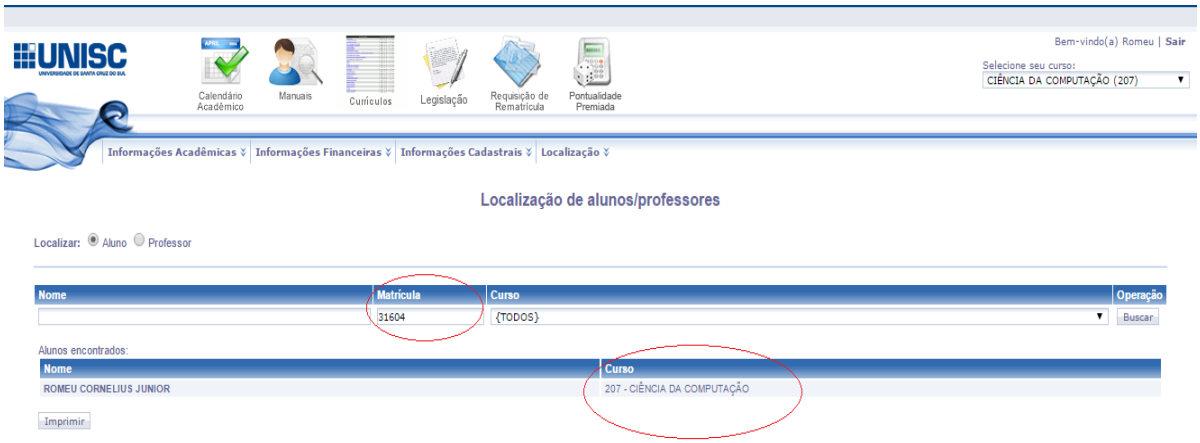
Após criada a tabela, selecionaram-se 1860 alunos para compor a nova base de dados. O critério de seleção utilizado foi o aluno ter cursado pelo menos uma disciplina em um dos três currículos em estudo (186,206,2509), ou seja, alunos que não cursaram disciplina alguma em pelo menos um destes currículos não foram selecionados.

No Anexo C podem ser visto os comandos SQL utilizados para a seleção dos registros e também os comandos para a seleção das disciplinas. Com os alunos já selecionados e importados na base HISTORICO, foram analisadas individualmente as 1860 matrículas selecionadas, com o intuito de preencher alguns campos que não foram preenchidos através de consultas SQL e programação, como por exemplo o tempo de permanência do aluno no curso em semestres, se ele trocou ou não de currículo, quantidade de trancamentos realizados.

Esta foi uma tarefa árdua e cansativa, mas devido à grande quantidade de registros duplicados por causa da forma como os dados foram cedidos, optou-se por fazê-la de forma manual.

Utilizou-se o sistema acadêmico da UNISC para outra análise feita em cada matrícula, para conferir se os dados importados condiziam com a situação atual do aluno importado na base HISTORICO. A Figura 17 mostra a consulta realizada com a matrícula do autor.

Figura 17 - Consulta de alunos matriculados no site da UNISC



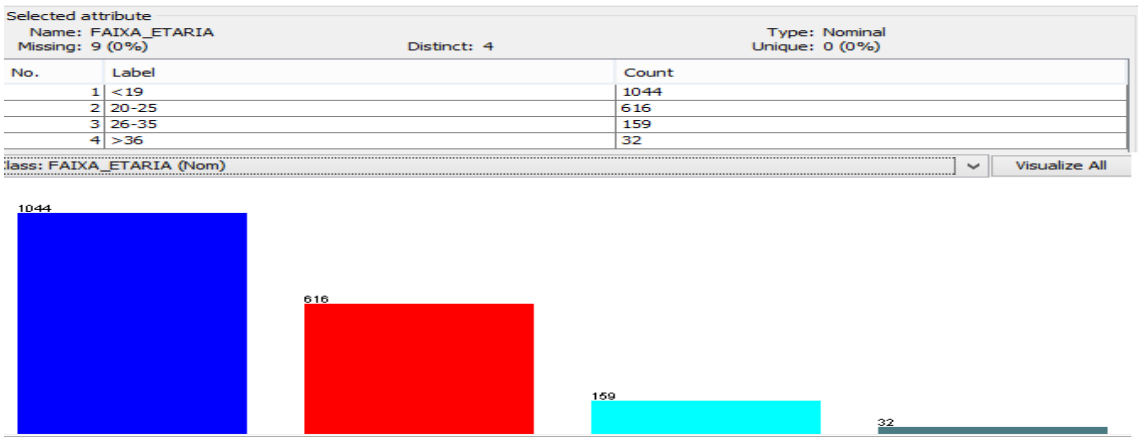
Fonte: Sistema acadêmico da UNISC, adaptado pelo Autor.

7.1.1 Alguns dados estatísticos sobre o domínio criado

Com a tabela HISTORICO devidamente carregada e balanceada, selecionaram-se alguns atributos como faixa etária, número do currículo, o meio de acesso, trocas de currículo, quantidade de trancamentos, o tempo de permanência e a situação atual do aluno no curso, a fim de gerar alguns dados estatísticos para se ter um melhor entendimento sobre os dados importados.

Na Figura 18 é possível visualizar que a grande maioria dos alunos que ingressou no curso, o fazem com menos de 20 anos, outra parcela significativa está na faixa etária entre 20 e 25 anos.

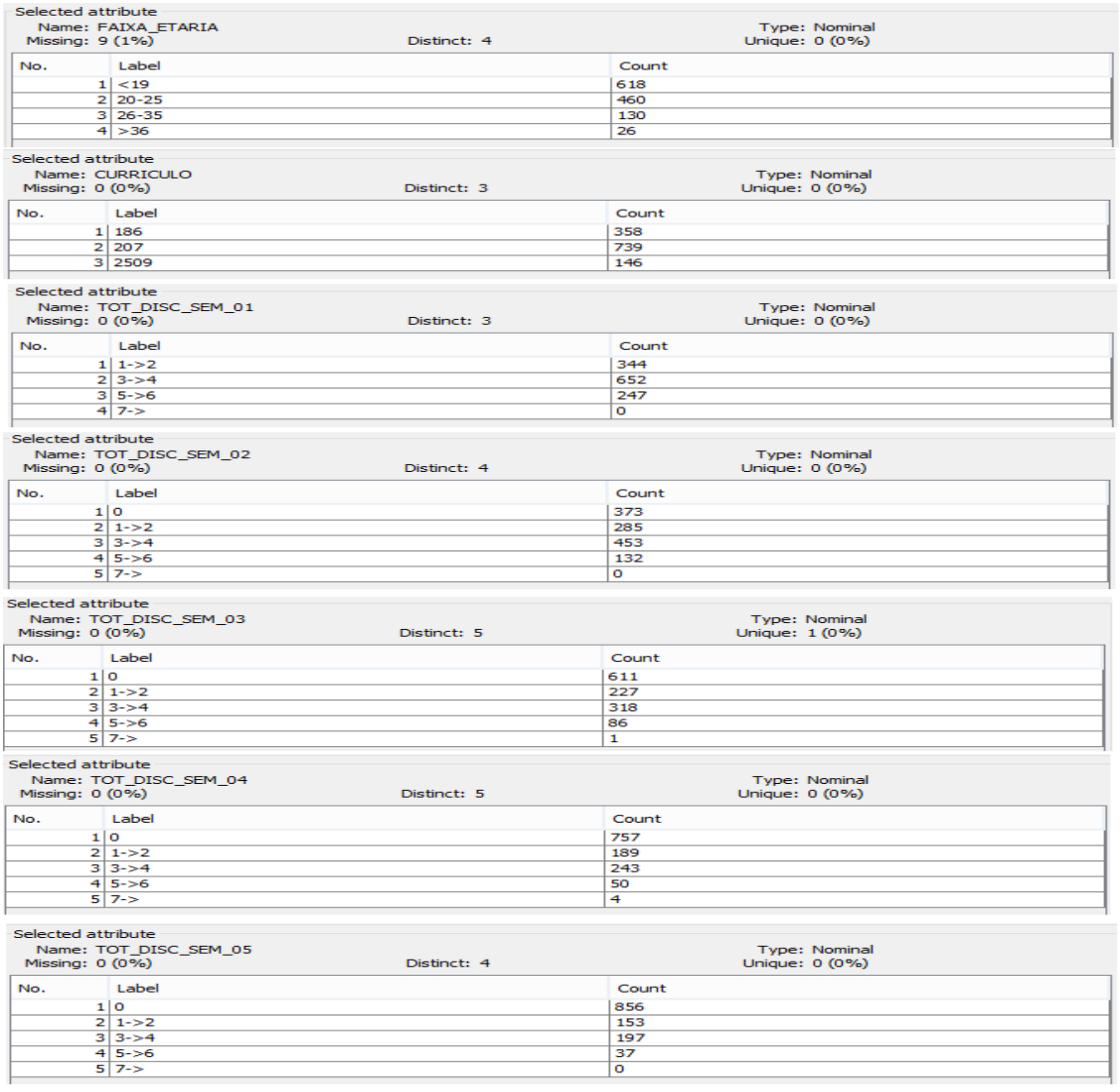
Figura 18 - Faixa etária dos alunos que ingressaram no curso



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 19, são exibidos alguns dados estatísticos dos alunos que evadiram do curso, como a faixa etária, o currículo em que cada aluno estava matriculado e o total de disciplinas cursadas por eles até o quinto semestre. Podemos ver também o número de alunos que evadem semestre a semestre, exibindo a evolução da evasão a partir do segundo semestre, pelo número de disciplinas cursadas, ou seja, pelo número do contador com valor “0” para o total de disciplinas cursadas.

Figura 19 - Dados sobre alunos que evadiram do curso



Selected attribute
Name: FAIXA_ETARIA
Missing: 9 (1%) Distinct: 4 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	<19	618
2	20-25	460
3	26-35	130
4	>36	26

Selected attribute
Name: CURRICULO
Missing: 0 (0%) Distinct: 3 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	186	358
2	207	739
3	2509	146

Selected attribute
Name: TOT_DISC_SEM_01
Missing: 0 (0%) Distinct: 3 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	1->2	344
2	3->4	652
3	5->6	247
4	7->	0

Selected attribute
Name: TOT_DISC_SEM_02
Missing: 0 (0%) Distinct: 4 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	0	373
2	1->2	285
3	3->4	453
4	5->6	132
5	7->	0

Selected attribute
Name: TOT_DISC_SEM_03
Missing: 0 (0%) Distinct: 5 Type: Nominal
Unique: 1 (0%)

No.	Label	Count
1	0	611
2	1->2	227
3	3->4	318
4	5->6	86
5	7->	1

Selected attribute
Name: TOT_DISC_SEM_04
Missing: 0 (0%) Distinct: 5 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	0	757
2	1->2	189
3	3->4	243
4	5->6	50
5	7->	4

Selected attribute
Name: TOT_DISC_SEM_05
Missing: 0 (0%) Distinct: 4 Type: Nominal
Unique: 0 (0%)

No.	Label	Count
1	0	856
2	1->2	153
3	3->4	197
4	5->6	37
5	7->	0

Fonte: WEKA. Adaptado pelo Autor.

Pode-se observar o aumento incremental do campo número 1 com *Label* 0 nas tabelas da Figura 19, onde é indicado o número de alunos que não cursaram

nenhuma disciplina no semestre em questão, nota-se que uma parcela muito grande dos alunos que evadem do curso, o fazem até o quinto semestre conforme pode ser visto na Figura 2.

7.2 Aplicação da mineração de dados na base de apoio

Nesta seção, são apresentadas as tarefas de mineração identificadas para serem aplicadas no domínio de dados criado, a preparação desses dados para serem utilizados nas tarefas escolhidas, os experimentos que foram realizados, bem como uma breve discussão sobre os resultados obtidos nos experimentos.

Com os dados selecionados e pré-processados, a próxima etapa no processo de KDD é a transformação dos dados, principalmente os atributos que apresentam uma sequência de valores muito abrangente, o que dificulta o processamento e entendimento dos resultados, e também para que os mesmos estejam nos padrões dos algoritmos de mineração de dados aos quais eles serão submetidos (WITTEN; FRANK; HALL, 2011).

Nesta etapa, atributos como idade, quantidade de trancamentos de matrícula, tempo de permanência, entre outros, passam por um processo de discretização, ou seja, foram transformados em outros valores que correspondem a mesma informação.

Por exemplo, o atributo idade foi transformado em faixa etária, a quantidade de trancamentos de matrícula transformada em <2 ou >2, o total de disciplinas cursadas transforma-se em < 5 e >= 5, entre outras.

Essas transformações se devem ao fato da escolha pela tarefa de classificação e a técnica de árvores de decisão, pois os algoritmos dessa classe trabalham melhor com atributos nominais (WITTEN; FRANK; HALL, 2011). Todas as consultas SQL e as transformações que foram feitas podem ser visualizadas no Anexo C.

Em conjunto com essas transformações, novos atributos foram criados na base de dados de apoio, nas tabelas DISCIPLINAS e HISTORICO, para ajudar a responder as perguntas que serão feitas no decorrer dos experimentos. A primeira parte dessas transformações, ocorreram na tabela DISCIPLINAS.

Foram criados dois novos atributos, TIPO e SEMESTRE. Todas as disciplinas cadastradas na base de dados, foram divididas em três grupos: i) COMP

(disciplinas consideradas básicas na computação como algoritmos, lógica, programação, redes entre outras.); ii) MATH (disciplinas de matemática como cálculo, geometria, matemática discreta entre outras.); iii) HUMAN (demais disciplinas, como inglês instrumental, língua portuguesa, filosofia, sociologia entre outras.), essas informações foram atribuídas ao atributo TIPO.

O Atributo SEMESTRE foi utilizado para definir em qual semestre a disciplina deveria ser ministrada, e foi preenchida da seguinte forma: As disciplinas do primeiro semestre receberam o valor SEQ1, ou seja, essas disciplinas deveriam aparecer na primeira sequência de disciplinas cursadas pelo aluno. As disciplinas do segundo semestre receberam o valor SEQ2, as disciplinas do terceiro semestre receberam o valor SEQ3 e assim por diante. O objetivo desse atributo é saber se o aluno fez disciplinas fora da ordem curricular.

A Figura 20 mostra o início da tabela DISCIPLINAS com os novos atributos criados para as disciplinas do primeiro e segundo semestre ordenadas de forma alfabética. Pode-se notar que algumas disciplinas sofreram alterações de código e nome ao longo do curso, isso deve-se em função dos três currículos utilizados no estudo (186,207,2509). Essas disciplinas foram unificadas, ou seja, não importa se o aluno cursou a disciplina código 2.120 ou 7.976, nos dois casos ele terá cursado a disciplina de algoritmos, ou se o aluno cursou a disciplina 8.511 ou 2.130 em um determinado semestre, para o estudo ele terá cursado a disciplina de cálculo.

Figura 20 - Tabela DISCIPLINAS

COD	NOME	TIPO	SEMESTRE
2.120	ALGORITMOS	COMP	SEQ1
7.976	ALGORITMOS E PROGRAMACAO	COMP	SEQ1
7.999	COMPUTADOR E SOCIEDADE	HUMAN	SEQ1
2.207	GEOMETRIA ANALITICA	MATH	SEQ1
1.841	INGLES INSTRUMENTAL	HUMAN	SEQ1
2.234	INTRODUCAO A COMPUTACAO	COMP	SEQ1
13.219	INTRODUCAO A COMPUTACAO	COMP	SEQ1
2.244	LOGICA PARA COMPUTACAO	COMP	SEQ1
7.988	MATEMATICA DISCRETA	MATH	SEQ1
10.021	MATEMATICA FUNDAMENTAL	MATH	SEQ1
2.130	CALCULO DIFERENCIAL E INTEGRAL	MATH	SEQ2
8.511	CALCULO I	MATH	SEQ2
2.190	ESTRUTURA DE DADOS	COMP	SEQ2
7.980	ESTRUTURA DE DADOS I	COMP	SEQ2
3.051	FILOSOFIA	HUMAN	SEQ2
3.059	FILOSOFIA E LOGICA	HUMAN	SEQ2
7.989	FISICA APLICADA A COMPUTACAO	COMP	SEQ2
1.884	LINGUA PORTUGUESA	HUMAN	SEQ2
1.999	PORTUGUES INSTRUMENTAL	HUMAN	SEQ2
7.982	PROGRAMACAO ESTRUTURADA	COMP	SEQ2
2.337	PROGRAMACAO I	COMP	SEQ2
7.986	SISTEMAS DIGITAIS	COMP	SEQ2

Fonte: Do Autor.

Com a tabela de disciplinas atualizada, a próxima etapa foi a transformação da tabela HISTORICO, onde foram criados novos atributos a fim de identificar a sequência de disciplinas cursadas por cada aluno em cada semestre. Para isso, o primeiro passo foi identificar quais as disciplinas cursadas pelos alunos, ordenadas por ano/semestre em que foram cursadas, e em seguida separar essas disciplinas em sequências, onde cada uma corresponde a um determinado semestre do aluno no curso.

Após a seleção das disciplinas, as mesmas foram armazenadas em planilhas para se ter noção exata de quais disciplinas aparecem em cada semestre. A Figura 21 exibe a planilha com todas as disciplinas que foram encontradas na segunda sequência de disciplinas (segundo semestre) de todos os alunos.

Figura 21 - Disciplinas encontradas na segunda sequência

Cód	Nome da disciplina
1884	LINGUA PORTUGUESA
2118	ALGEBRA LINEAR
10009	ALGEBRA LINEAR E GEOMETRIA ANALITICA
2120	ALGORITMOS
7976	ALGORITMOS E PROGRAMACAO
2130	CALCULO DIFERENCIAL E INTEGRAL
2132	CALCULO DIFERENCIAL E INTEGRAL II
8511	CALCULO I
2137	CALCULO NUMERICO
2139	COMPILADORES
1127	CONTABILIDADE GERAL
2160	ESTATISTICA
2190	ESTRUTURA DE DADOS
7980	ESTRUTURA DE DADOS I
16719	ESTRUTURA DE DADOS II
3051	FILOSOFIA
3059	FILOSOFIA E LOGICA
7989	FISICA APLICADA A COMPUTACAO
2450	FUNDAMENTOS DE FISICA
2207	GEOMETRIA ANALITICA
7991	GERENCIA E ADMINISTRACAO DE REDES
1841	INGLES INSTRUMENTAL
2234	INTRODUCAO A COMPUTACAO
13219	INTRODUCAO A COMPUTACAO

Cód	Nome da disciplina
15187	LEGISLACAO EM INFORMATICA
9029	LEITURA E PRODUCAO DE TEXTOS
2244	LOGICA PARA COMPUTACAO
7988	MATEMATICA DISCRETA
2294	MATEMATICA FINANCEIRA
10021	MATEMATICA FUNDAMENTAL
3105	METODOS E TECNICAS DE PESQUISA
1142	NOCOES DE CONTABILIDADE
2311	ORGANIZACAO DE ARQUIVOS
2312	ORGANIZACAO DE BANCO DE DADOS
2313	ORGANIZACAO DE COMPUTADORES
1055	ORGANIZACAO E METODOS
1057	PESQUISA OPERACIONAL
7982	PROGRAMACAO ESTRUTURADA
2337	PROGRAMACAO I
2339	PROGRAMACAO III
3176	PSICOLOGIA DO TRABALHO
7998	PSICOLOGIA DO TRABALHO
2346	REDES DE COMPUTADORES I
7986	SISTEMAS DIGITAIS
2350	SISTEMAS OPERACIONAIS I
3227	SOCIOLOGIA DO TRABALHO
7997	SOCIOLOGIA DO TRABALHO
2352	TECNICAS DE PROGRAMACAO

Fonte: Do Autor.

Foram coletadas informações referentes aos cinco primeiros semestres de cada aluno no curso. Com base nas disciplinas encontradas em cada sequência, foram criados atributos para identificar as disciplinas que foram cursadas em cada semestre do aluno no curso, e se ele passou ou reprovou nas disciplinas.

A seguir são listados alguns exemplos dos atributos criados para as disciplinas encontradas no segundo semestre dos alunos no curso, conforme visto na Figura 21.

a) Sem2_Algebra (indica se o aluno fez álgebra ou não)
b) Sem2_Algebra_St (indica se o aluno passou ou não)
c) Sem2_Algoritmo (indica se o aluno fez algoritmo ou não)
d) Sem2_Algoritmo_St (indica se o aluno passou ou não)
e) Sem2_Math (indica se o aluno fez disciplinas do tipo math)
f) Sem2_Comp (indica se o aluno fez disciplinas do tipo comp)
g) Sem2_Algo_Alge (indica se o aluno fez algoritmo e álgebra juntos)

Foram criados atributos deste tipo para todas as disciplinas encontradas em cada um dos cinco primeiros semestres. Estes atributos foram preenchidos com ‘S’ no caso do aluno ter cursado a disciplina ou ‘N’ no caso de não ter cursado a disciplina no semestre em questão.

Em alguns casos estes atributos receberam o valor ‘?’ quando o aluno não cursou a disciplina, ou seja, não receberam o valor ‘N’. Trata-se de disciplinas que apareceram fora da sequência curricular, como por exemplo, quando um aluno fez a disciplina de Cálculo no primeiro semestre. Conforme observa-se na Figura 20, essa é uma disciplina do segundo semestre e não do primeiro, deveria aparecer na segunda ou terceira sequência de disciplinas em diante, caso o aluno tenha feito menos que cinco disciplinas na primeira sequência, mas não deveria estar na primeira sequência de disciplinas cursadas.

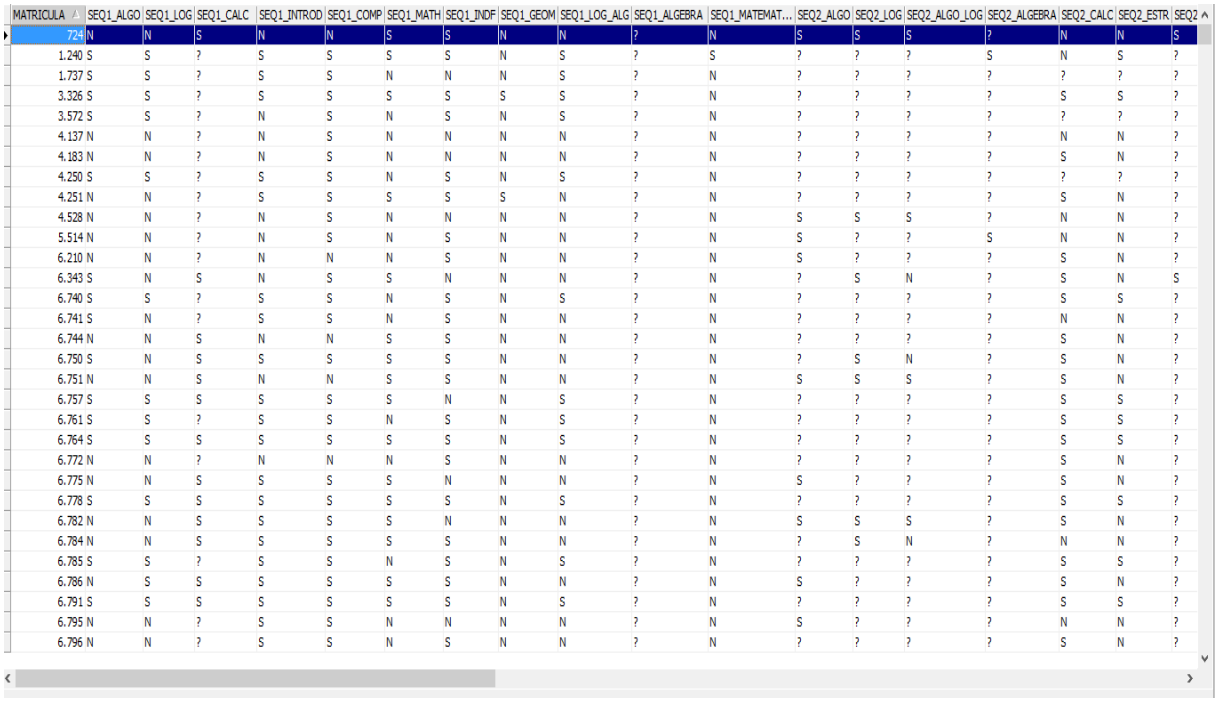
O WEKA oferece um tratamento especial nesses casos onde são encontrados os atributos com valor ‘?’, não considerando esses valores no cálculo de ganho de informação (HALL *et al*., 2009), por isso não foi utilizado o valor ‘N’, pois pela ordem curricular, essa disciplina não deveria ter sido cursada no semestre em questão, assim, os alunos que não cursaram essa disciplina não devem receber o valor ‘N’ como se tivessem que fazer a disciplina e não tivessem feito.

Com base nestes novos atributos criados, foi possível saber quais disciplinas o aluno fez em um determinado semestre, qual o status das disciplinas em questão e se foram cursadas na ordem curricular.

A Figura 22 mostra a tabela HISTORICO ao final da etapa de transformação, são exibidos alguns dos atributos criados, ao total a base ficou composta de 180 atributos e 1591 registros, contendo informação dos alunos que se formaram ou evadiram do curso, por ainda não sabermos se irão evadir ou se formar no curso.

As informações sobre os alunos ativos do curso foram removidas da base, o ANEXO D contém uma lista com as matrículas removidas da base de dados, que podem ser utilizadas para validar o modelo de classificação criado em trabalhos futuros, a fim de confirmar se o conhecimento gerado a partir deste estudo se aplica também naquele grupo de alunos, se conseguiremos prever se cada aluno do curso possui o perfil de evasão ou não.

Figura 22 - Tabela HISTORICO ao final da etapa de transformação



MATRICULA	SEQ1_ALGO	SEQ1_LOG	SEQ1_CALC	SEQ1_INTROD	SEQ1_COMP	SEQ1_MATH	SEQ1_INDF	SEQ1_GEOM	SEQ1_LOG_ALG	SEQ1_ALGEBRA	SEQ1_MATEMAT...	SEQ2_ALGO	SEQ2_LOG	SEQ2_ALGO_LOG	SEQ2_ALGEBRA	SEQ2_CALC	SEQ2_ESTR	SEQ2
724	N	N	S	N	N	S	S	N	N	?	N	S	S	S	?	N	N	S
1.240	S	S	?	S	S	S	S	N	S	?	S	?	?	?	S	N	S	?
1.737	S	S	?	S	S	N	N	N	S	?	N	?	?	?	?	?	?	?
3.326	S	S	?	S	S	S	S	S	S	?	N	?	?	?	?	S	S	?
3.572	S	S	?	N	S	N	S	N	S	?	N	?	?	?	?	?	?	?
4.137	N	N	?	N	S	N	N	N	N	?	N	?	?	?	?	N	N	?
4.183	N	N	?	N	S	N	N	N	N	?	N	?	?	?	?	S	N	?
4.250	S	S	?	S	S	N	S	N	S	?	N	?	?	?	?	?	?	?
4.251	N	N	?	S	S	S	S	S	N	?	N	?	?	?	?	S	N	?
4.528	N	N	?	N	S	N	N	N	N	?	N	S	S	S	?	N	N	?
5.514	N	N	?	N	S	N	S	N	N	?	N	S	?	?	S	N	N	?
6.210	N	N	?	N	N	N	S	N	N	?	N	S	?	?	?	S	N	?
6.343	S	N	S	N	S	S	N	N	N	?	N	?	S	N	?	S	N	S
6.740	S	S	?	S	S	N	S	N	S	?	N	?	?	?	?	S	S	?
6.741	S	N	?	S	S	N	S	N	N	?	N	?	?	?	?	N	N	?
6.744	N	N	S	N	N	S	S	N	N	?	N	?	?	?	?	S	N	?
6.750	S	N	S	S	S	S	S	N	N	?	N	?	S	N	?	S	N	?
6.751	N	N	S	N	N	S	S	N	N	?	N	S	S	S	?	S	N	?
6.757	S	S	S	S	S	S	N	N	S	?	N	?	?	?	?	S	S	?
6.761	S	S	?	S	S	N	S	N	S	?	N	?	?	?	?	S	S	?
6.764	S	S	S	S	S	S	S	N	S	?	N	?	?	?	?	S	S	?
6.772	N	N	?	N	N	N	S	N	N	?	N	?	?	?	?	S	N	?
6.775	N	N	S	S	S	S	N	N	N	?	N	S	?	?	?	S	N	?
6.778	S	S	S	S	S	S	S	N	S	?	N	?	?	?	?	S	S	?
6.782	N	N	S	S	S	S	N	N	N	?	N	S	S	S	?	S	N	?
6.784	N	N	S	S	S	S	S	N	N	?	N	?	S	N	?	N	N	?
6.785	S	S	?	S	S	N	S	N	S	?	N	?	?	?	?	S	S	?
6.786	N	S	S	S	S	S	S	N	N	?	N	S	?	?	?	S	N	?
6.791	S	S	S	S	S	S	S	N	S	?	N	?	?	?	?	S	S	?
6.795	N	N	?	S	S	N	N	N	N	?	N	S	?	?	?	N	N	?
6.796	N	N	?	S	S	N	S	N	N	?	N	?	?	?	?	S	N	?

Fonte: Do Autor.

É importante salientar, que nem todos os atributos descritos no modelo de dados da tabela HISTORICO mencionados anteriormente, irão aparecer no arquivo que será utilizado no WEKA, mas todos os atributos serão necessários para a geração das consultas que serão exportadas e que darão origem ao arquivo (.arff), o qual será utilizado no WEKA.

7.2.1 Identificação das tarefas de mineração aplicáveis ao domínio

Para a identificação das tarefas aplicáveis ao domínio HISTORICO, além de sua análise e entendimento, foi feito um estudo nos trabalhos relacionados. As principais tarefas definidas para o estudo são descritas a seguir:

a) Associação pela ordem das disciplinas cursadas: na tentativa de traçar o perfil dos alunos que cursam determinadas disciplinas em sequência e acabam por abandonar o curso, utilizando-se dos atributos criados para cada uma das disciplinas cursadas, além do ano e semestre em que cada uma delas foram ministradas bem como o status (aprovado, reprovado, desistente);
b) Classificação por tempo de permanência: para descobrir quanto o tempo de permanência do aluno no curso contribui ou não para a evasão, utilizando quantidade de trancamentos, troca ou não de currículo, faixa etária entre outros;
c) Classificação pelo número de disciplinas cursadas a cada semestre: para descobrir se os alunos que cursam mais ou menos disciplinas por semestre apresentam maior propensão à evasão, utilizando o total de disciplinas cursadas, total de disciplina cursada em cada semestre, currículo, entre outros;
d) Classificação pelo tipo de disciplinas cursadas a cada semestre: para descobrir se alunos que sempre fazem disciplinas do tipo COMP (disciplinas base da computação) são menos propensos a evadir do que alunos que fazem somente disciplinas do tipo MATH ou HUMAN, e os que simplesmente seguem o currículo, ou seja, fazem os três tipos de disciplinas misturados (COMP, MATH, HUMAN).

Tendo por base as tarefas identificadas, também foram criadas algumas perguntas para serem respondidas através deste estudo, no intuito de tentar identificar o perfil do aluno que evade do curso. As perguntas criadas são descritas a seguir:

a) Existe uma sequência de disciplinas feitas que provocam a evasão até o terceiro semestre?
b) Quem sempre faz cinco disciplinas nos cinco primeiros semestres evade ou não? E quem sempre faz três ou menos disciplinas?
c) Quem reprova em disciplinas consideradas chave do curso, como lógica e algoritmos desiste?
d) Fazer algoritmos e lógica no mesmo semestre ajuda na permanência do aluno no curso?
e) Reprovar em disciplinas de matemática nos primeiros semestres faz com que o aluno desista do curso?
f) A quantidade de disciplinas do primeiro semestre influencia na permanência ou desistência do aluno?

7.3 Experimentos realizados

Com o intuito de responder as perguntas acima mencionadas, e também aplicar as tarefas de mineração identificadas para o domínio de dados criado, foram realizados cinco tipos de experimentos, listados a seguir:

A. Associação pela ordem das disciplinas cursadas nos três primeiros semestres de cada aluno no curso.
B. Classificação dos alunos que reprovam nas primeiras disciplinas da grade curricular do curso.
C. Classificação dos alunos pela quantidade de disciplinas cursadas no primeiro semestre de cada aluno no curso.
D. Classificação dos alunos pela quantidade de disciplinas cursadas nos cinco primeiros semestres do aluno no curso.
E. Classificação dos alunos que cursaram Algoritmo e Lógica juntos, no mesmo semestre.

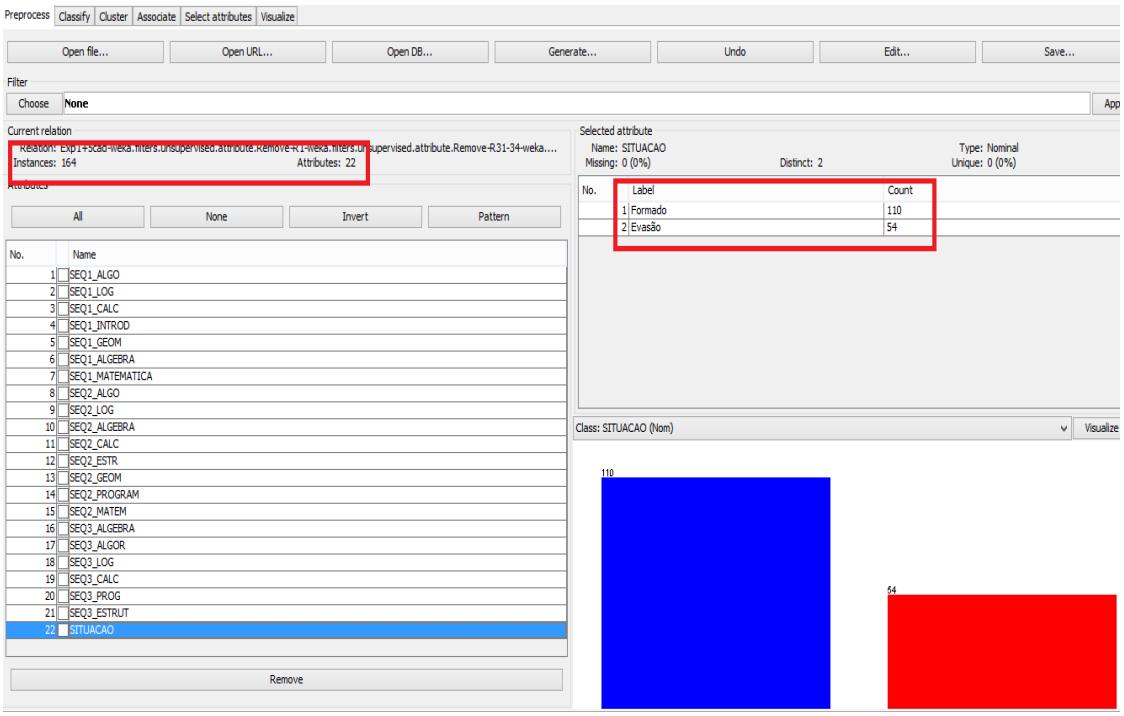
7.3.1 Experimento A

Para a realização do experimento A, foram analisadas as três primeiras sequências de disciplinas cursadas de cada aluno dividindo a base de dados em dois perfis de alunos.

Um perfil com os alunos que sempre fizeram cinco ou mais disciplinas nos três primeiros semestres e outro perfil dos alunos que sempre fizeram menos que cinco disciplinas nos três primeiros semestres. Cada perfil da base de dados será submetido a dois algoritmos de associação temporal, *Apriori* e *FpGrowth*. Em ambos os casos, utilizando as configurações padrão do WEKA.

A Figura 23 mostra o perfil dos alunos que sempre fizeram cinco ou mais disciplinas nos três primeiros semestres. Foram encontrados 164 alunos que satisfaziam esse critério, dos quais 110 são formados e 54 evadiram do curso. Após vários testes, selecionaram-se 22 atributos correspondentes às disciplinas cursadas nesse período para o experimento.

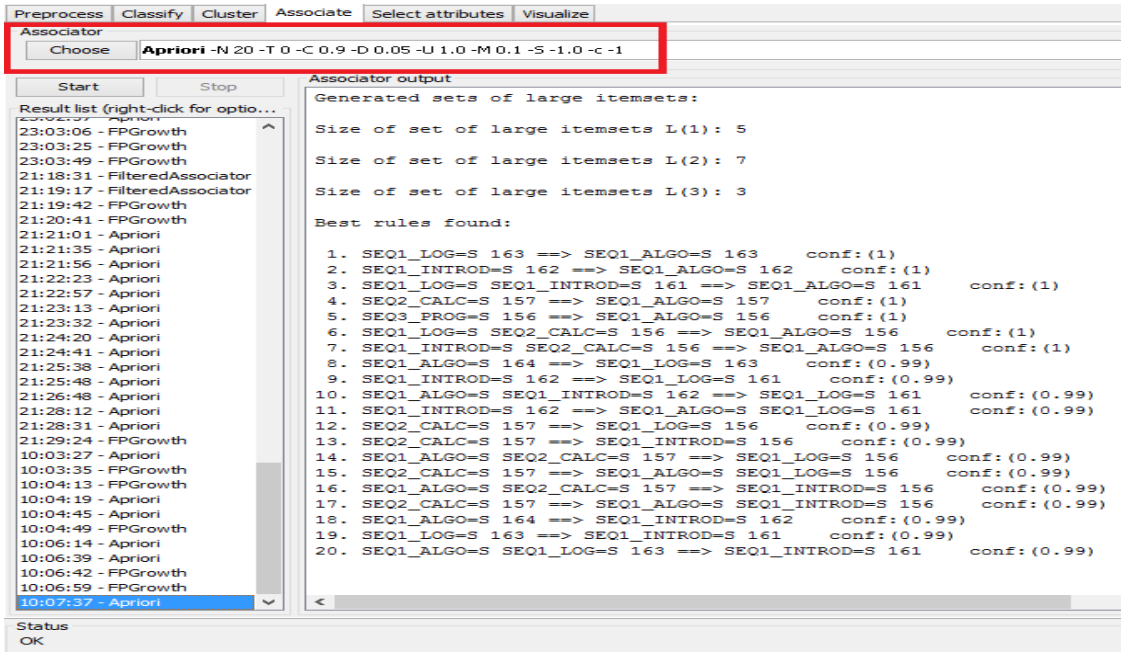
Figura 23 - Perfil 1 dos alunos do experimento A



Fonte: WEKA. Adaptado pelo Autor.

A figura 24 mostra o resultado do experimento A1, a associação destes 22 atributos feita pelo algoritmo *Apriori.* Foram listadas as 20 principais regras geradas pelo algoritmo, todas elas com grau de confiança superior a 98%, mas nenhuma delas associada diretamente a evasão. As regras geradas foram do tipo: SEQ1_LOG=S SEQ1_INTROD=S 163 -> SEQ1_algo=S 162 conf:(0.99), ou seja, quem fez lógica e introdução na primeira sequência de disciplinas cursadas, também fez algoritmos, o 163 significa que essa sequência apareceu cento e sessenta e três vezes e se confirmou em 162 vezes, ou seja, esta regra tem um grau de confiança de 99%.

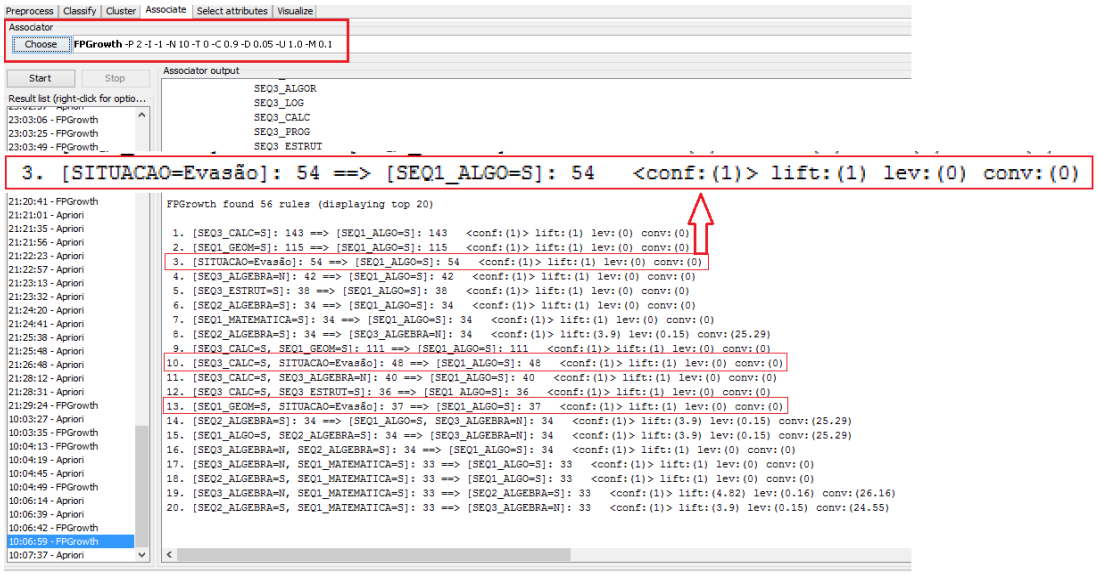
Figura 24 - Resultado experimento A1 - *Apriori*



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 25 é exibido o resultado obtido do experimento A1, obtido pelo algoritmo *FpGrowth*. Foram geradas 20 regras com grau de confiança de 100%, desta vez o algoritmo de mineração associou pelo menos três regras com a evasão, a principal regra é que dos 54 casos de evasão deste grupo de alunos analisados, todos fizeram algoritmos no primeiro semestre, dos 48 alunos que fizeram algoritmos no primeiro semestre e cálculo no terceiro semestre, 48 evadiram.

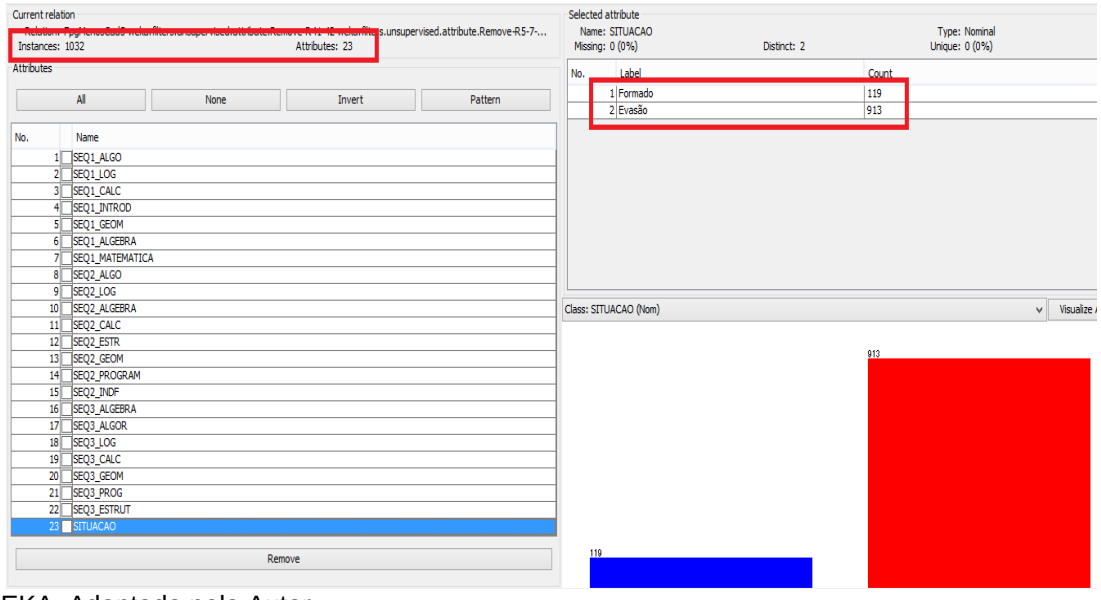
Figura 25 - Resultado experimento A1 – *FpGrowth*



Fonte: WEKA. Adaptado pelo Autor.

A Figura 26 mostra o perfil dos alunos que sempre fizeram menos de cinco disciplinas nos três primeiros semestres. Foram encontrados 1032 alunos que satisfaziam esse critério, dos quais 119 são formados e 913 evadiram do curso. Novamente selecionaram-se 23 atributos correspondentes às disciplinas cursadas pelos alunos neste período para análise.

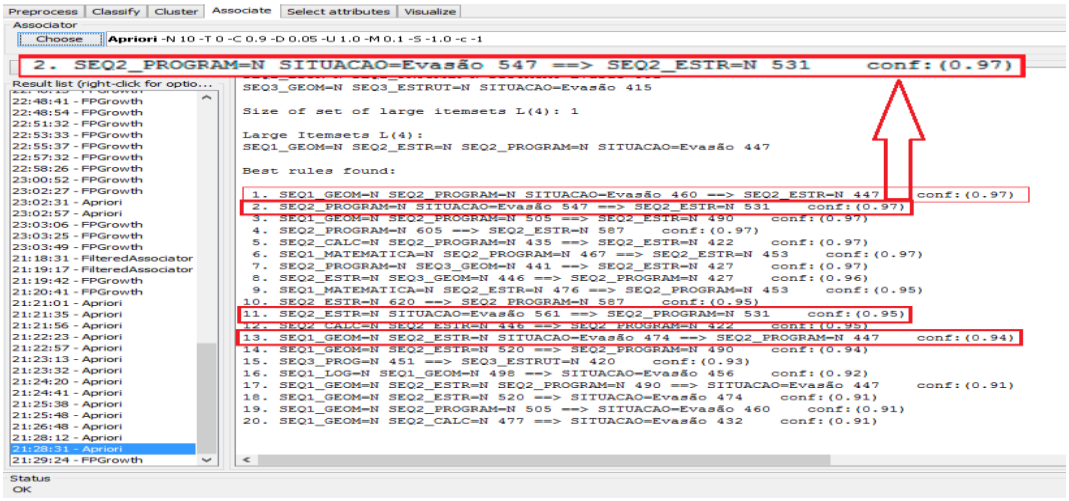
Figura 26 - Perfil 2 dos alunos do experimento A



Fonte: WEKA. Adaptado pelo Autor.

A figura 27 mostra o resultado do experimento A2, associação dos 23 atributos feita pelo algoritmo *Apriori* e na Figura 28 é exibido o resultado obtido do experimento A2, obtido pelo algoritmo *FpGrowth*.

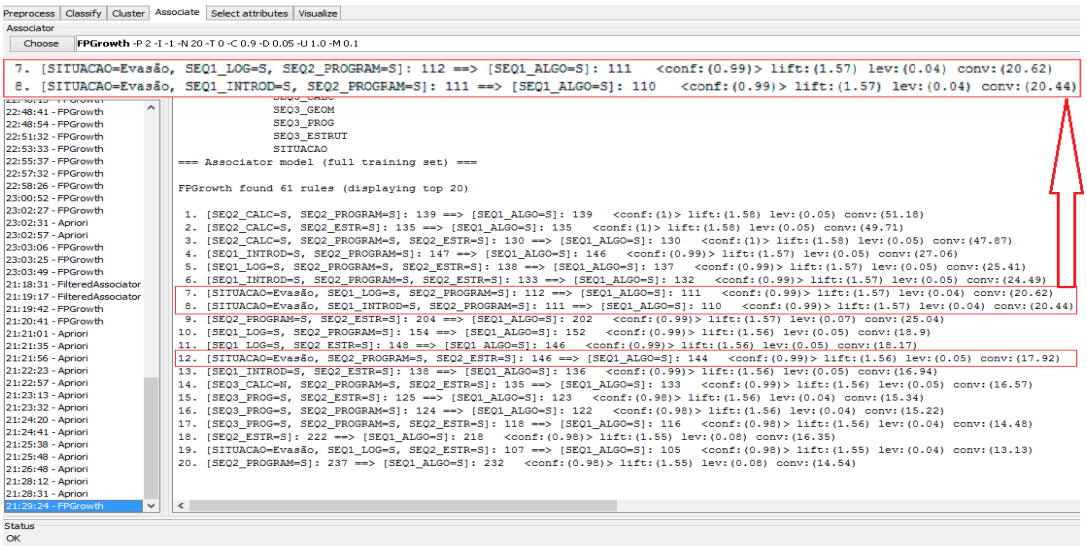
Figura 27 - Resultado experimento A2 - *Apriori*



Fonte: WEKA. Adaptado pelo Autor.

A principal regra gerada pelo *Apriori* foi que quem não fez geometria no primeiro semestre e não fez programação no segundo semestre e evadiu do curso (aconteceu em 460 casos), também não fez estrutura de dados no segundo semestre (se confirmou em 447 casos).

Figura 28 - Resultado experimento A2 - *FpGrowth*



Fonte: WEKA. Adaptado pelo Autor.

A principal regra gerada pelo *FpGrowth* foi que quem fez lógica no primeiro semestre e fez programação no segundo semestre e evadiu do curso (aconteceu em 112 casos), também fez algoritmos no primeiro semestre (se confirmou em 111 casos).

Pelos resultados do experimento A, podemos observar que no perfil dos 164 alunos que sempre fizeram as cinco disciplinas nos três semestres analisados, a sequência de disciplinas que mais se confirmou é a de quem faz algoritmos, lógica e introdução no primeiro semestre, fez cálculo no segundo semestre. Todos os 54 alunos que evadiram deste grupo fizeram algoritmos no primeiro semestre, sendo que 48 deles fizeram cálculo apenas no terceiro semestre. Já no perfil dos 1032 alunos que fizeram menos que cinco disciplinas nos três semestres, a regra mais significativa foi a de que 447 destes alunos que não fizeram geometria no primeiro semestre, também não fizeram estrutura de dados e programação estruturada no segundo semestre, e acabaram abandonando do curso.

7.3.2 Experimento B

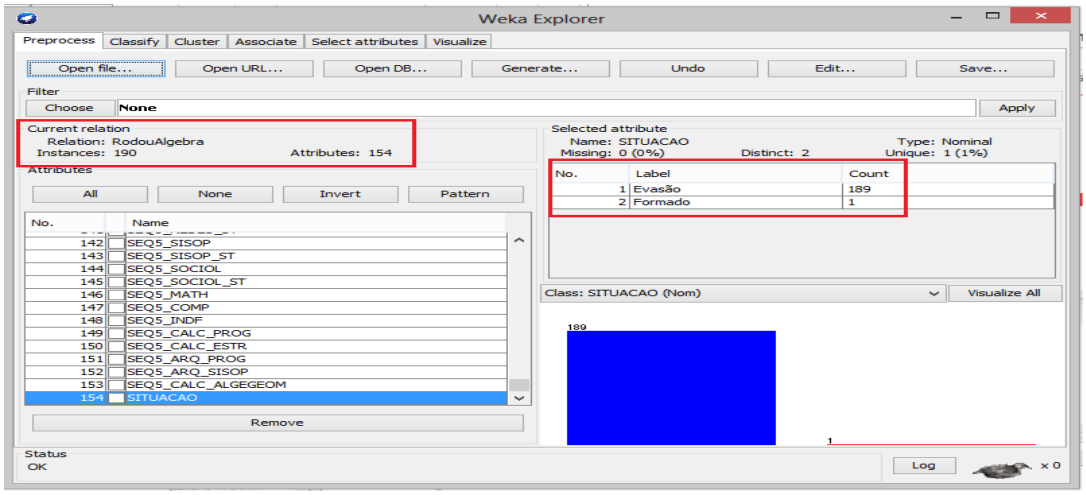
Para a realização do experimento B, classificação dos alunos que reprovam nas primeiras disciplinas da grade curricular, foram selecionados os alunos que em algum momento do curso reprovaram nas seguintes disciplinas:

1. Álgebra linear (190 ocorrências)
2. Algoritmos (426 ocorrências)
3. Cálculo (302 ocorrências)
4. Introdução a computação (170 ocorrências)
5. Lógica (401 ocorrências)

Neste experimento, foram criados cinco arquivos, um para cada uma das disciplinas, selecionando todos os alunos que satisfaziam esses critérios, ou seja, que reprovaram em alguma das disciplinas. A ideia é submeter cada um dos arquivos ao algoritmo de classificação J48 em validação cruzada, utilizando as configurações padrão do WEKA, para tentar descobrir se existe um padrão entre os alunos que reprovam nestas disciplinas.

A Figura 29 mostra o perfil dos alunos que reprovaram em álgebra. Foram selecionados 190 alunos que atenderam a esse quesito e 154 atributos da tabela HISTORICO, contendo todas as informações reunidas sobre a vida acadêmica dos alunos para o começo do experimento.

Figura 29 - Perfil dos alunos que reprovam em Álgebra

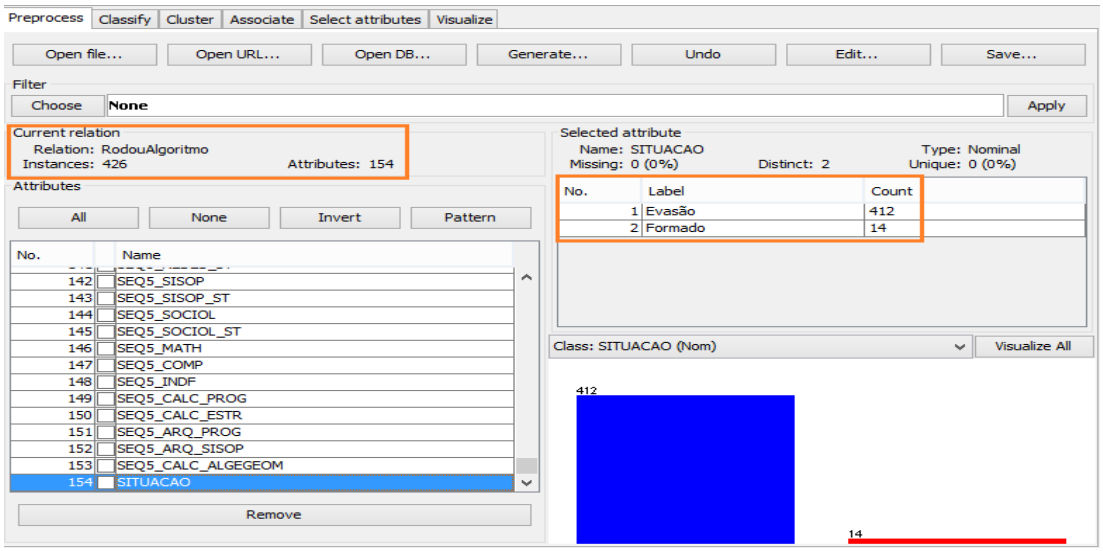


Fonte: WEKA. Adaptado pelo Autor.

Neste caso, dos 190 alunos que reprovaram na disciplina, apenas um aluno se formou, os outros 189 alunos desistiram do curso, logo, o arquivo nem precisou ser submetido a testes no WEKA para procurar um padrão para a evasão, devido a quantidade de alunos que evadiram, as árvores geradas pelos algoritmos classificaram todos os alunos como evasão.

A Figura 30 mostra o perfil dos alunos que reprovaram em algoritmos. Foram selecionados 426 alunos que atenderam a esse quesito, destes 14 são alunos formados e 412 alunos que evadiram, e os mesmos 154 atributos da tabela HISTORICO selecionados no experimento anterior.

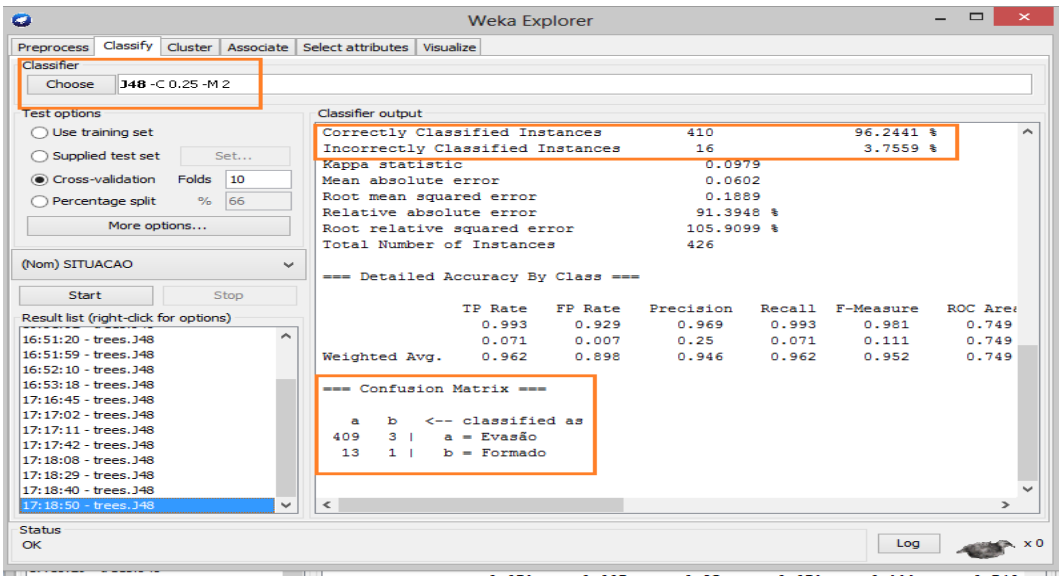
Figura 30 - Perfil dos alunos que reprovam em Algoritmo



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 31 é exibido o resultado do experimento de classificação dos alunos que reprovaram em Algoritmos. O classificador apresentou uma precisão de mais de 96%, a matriz de confusão mostra que o modelo foi capaz de classificar com sucesso quase todos os casos de evasão dos alunos, dos 412 alunos que evadiram, 409 foram classificados de forma correta, porém o modelo se mostrou ineficaz para classificar os alunos que se formaram, dos 14 casos apenas um foi classificado corretamente.

Figura 31 - Resultado do experimento B2

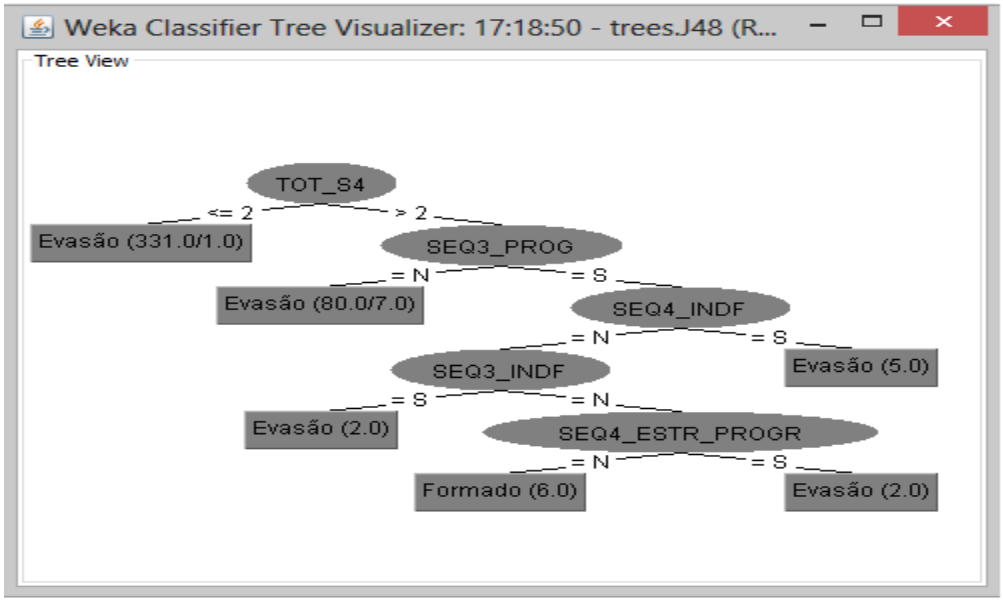


Fonte: WEKA. Adaptado pelo Autor.

A Figura 32 mostra a árvore de decisão gerada pelo algoritmo J48 para o experimento B2, onde pode-se ver que alunos que fazem duas ou menos disciplinas no quarto semestre TOT_S4 (ou quarta sequência de disciplinas) têm possibilidades de evadir, e os alunos que fazem mais que duas disciplinas no quarto semestre, mas que não fizeram a disciplina de programação no terceiro semestre SEQ3_PROG, também têm chances de abandonar o curso.

Já os alunos que fizeram programação no terceiro semestre e fizeram disciplinas do tipo HUMAN no quarto semestre SEQ4_INDF, acabaram evadindo. Os alunos que não fizeram este tipo de disciplina no quarto semestre e não cursaram estrutura de dados e programação juntas no quarto semestre se formaram.

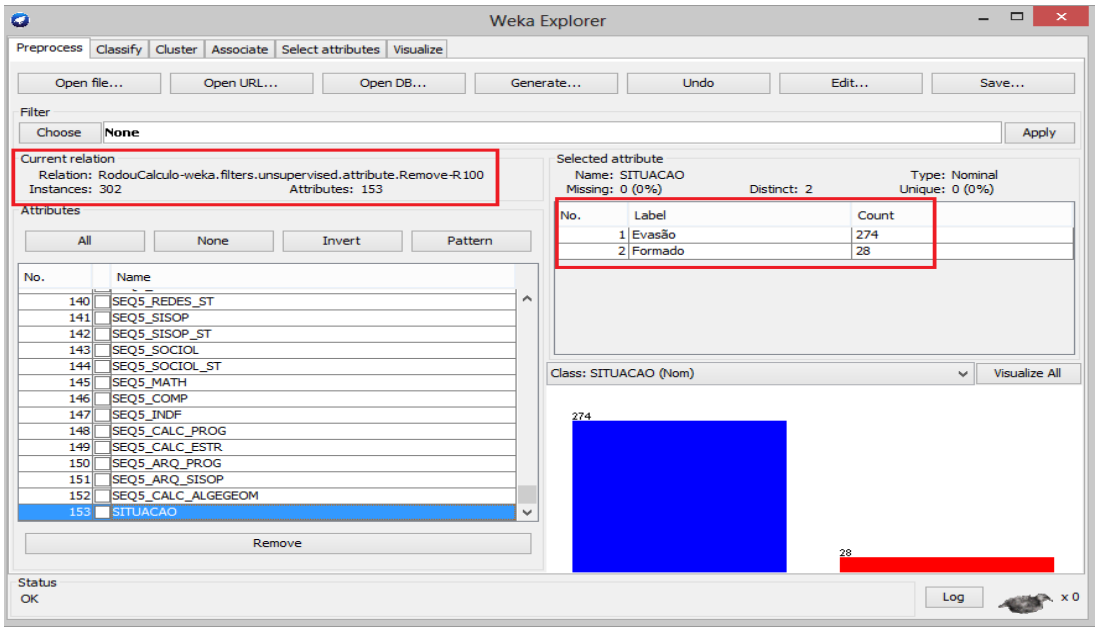
Figura 32 - Árvore gerada pelo experimento B2



Fonte: WEKA. Adaptado pelo Autor.

A Figura 33 mostra o perfil dos alunos que reprovaram em Cálculo. Foram selecionados 302 alunos que atenderam a esse quesito, destes 28 são alunos formados e 274 alunos que evadiram, e os mesmos atributos dos experimentos anteriores foram selecionados.

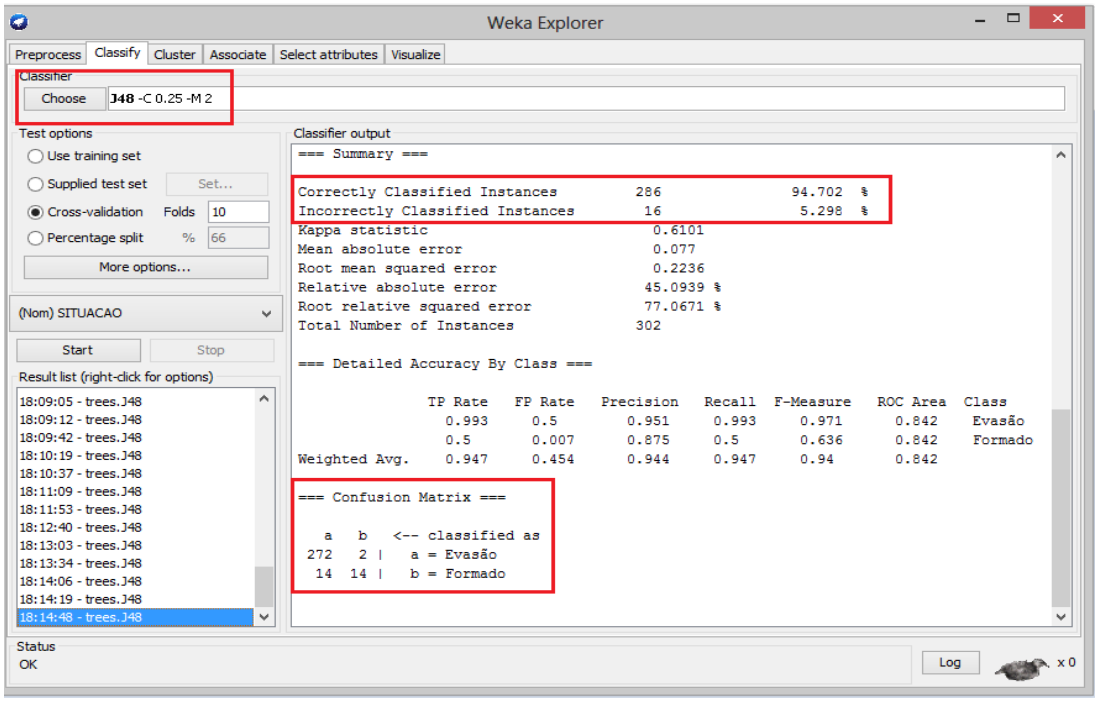
Figura 33 - Perfil dos alunos que reprovaram em Cálculo



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 34 é exibido o resultado do experimento de classificação dos alunos que reprovaram em Cálculo, novamente o classificador apresentou um alto percentual de precisão, acima de 94% de acertos.

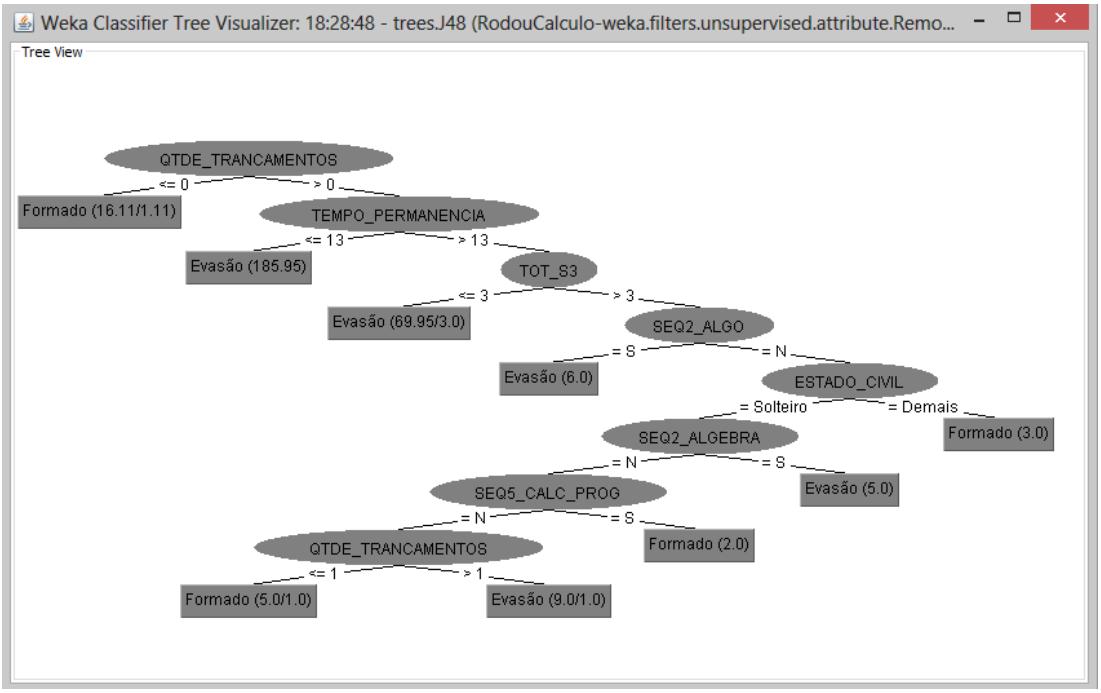
Figura 34 - Resultado do experimento B3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 35 mostra a árvore de decisão gerada pelo algoritmo J48 para o experimento com os alunos que reprovaram em Cálculo, onde pode-se ver que o tempo de permanência e o total de disciplinas cursadas no terceiro semestre TOT_S3 do aluno no curso tem forte influência no abandono ou não do curso pelos alunos que reprovam em Cálculo.

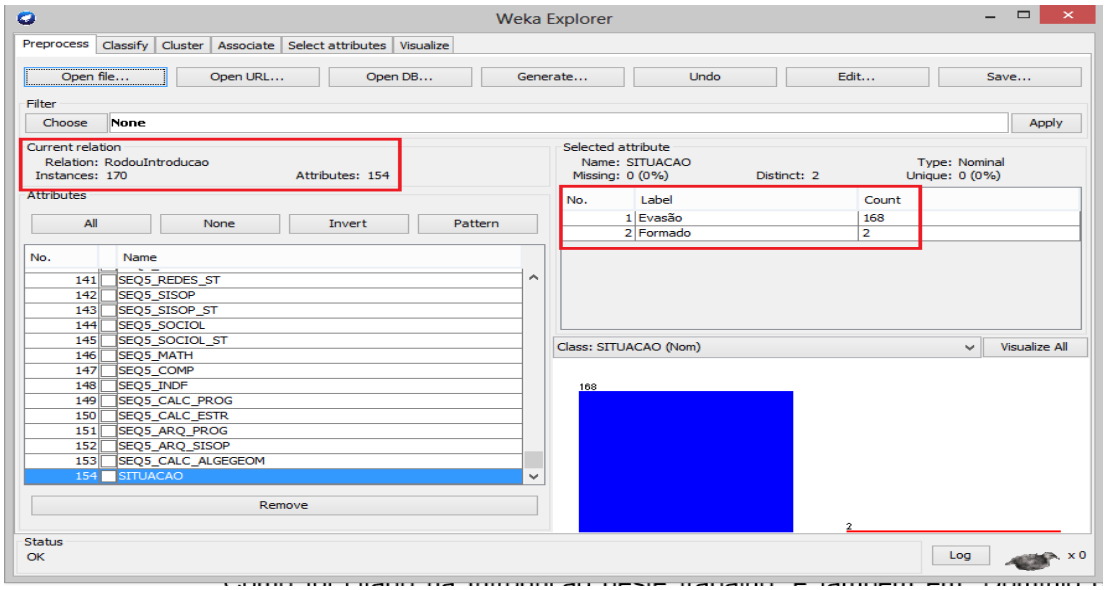
Figura 35 - Árvore gerada pelo experimento B3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 36 mostra o perfil dos alunos que reprovaram em Introdução a Computação. Foram selecionados 170 alunos que atenderam a esse quesito, destes apenas dois alunos são formados e 168 alunos evadiram do curso.

Figura 36 - Perfil dos alunos que reprovam em Introdução à Computação

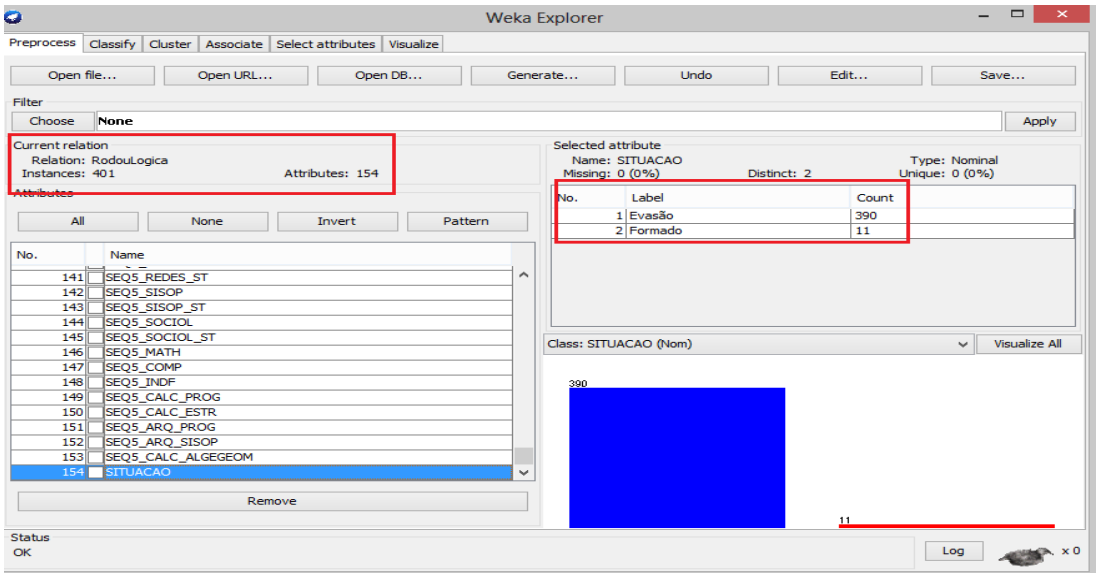


Fonte: WEKA. Adaptado pelo Autor.

Novamente a maioria dos alunos que reprovaram, 168 dos 170 alunos selecionados desistiram do curso, logo, o arquivo nem precisou ser submetido a testes no WEKA para procurar um padrão, uma vez que os classificadores nem geram uma árvore de decisão nesses casos, classificam todos alunos como evasão.

A Figura 37 mostra o perfil dos alunos que reprovaram em Lógica. Foram selecionados 401 alunos que atenderam a esse quesito, destes 11 são alunos formados e 390 alunos que evadiram, e os mesmos atributos dos experimentos anteriores foram selecionados.

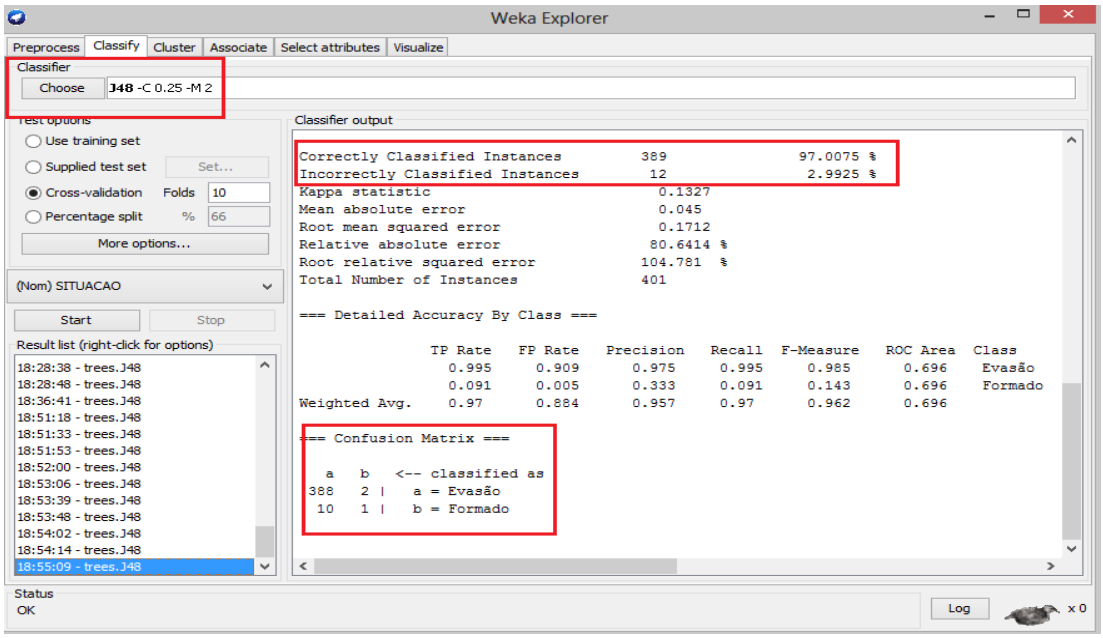
Figura 37 - Perfil dos alunos que reprovaram em Lógica



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 38 é exibido o resultado do experimento de classificação dos alunos que reprovaram em Lógica, como o número de formados também é muito pequeno para esse grupo de alunos, o classificador apresenta um altíssimo percentual de acertos, acima de 97%.

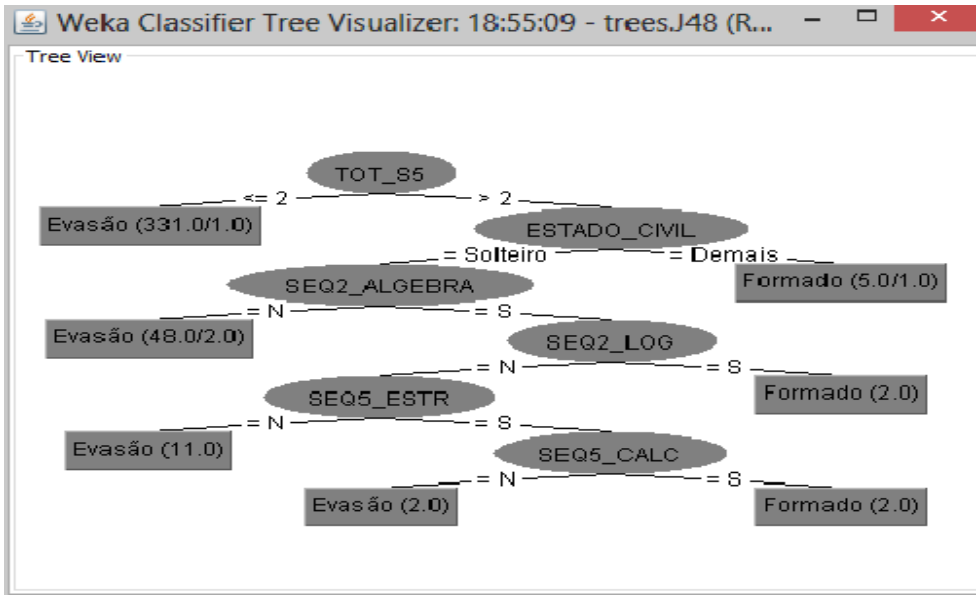
Figura 38 - Resultado do experimento B4



Fonte: WEKA. Adaptado pelo Autor.

A Figura 39 mostra a árvore de decisão gerada pelo algoritmo J48 para o experimento com os alunos que reprovaram em Lógica, onde pode-se ver que o total de disciplinas cursadas no quinto semestre destes alunos no curso tem forte influência no abandono ou não do curso, onde o aluno que faz duas ou menos disciplinas no quinto semestre tem maior probabilidade de evadir.

Figura 39 - Árvore gerada pelo experimento B4



Fonte: WEKA. Adaptado pelo Autor.

Os resultados do experimento B deixam claro que reprovar nas primeiras disciplinas do curso pode ser catastrófico para a permanência do aluno no curso, no melhor dos casos, apenas 9% dos alunos que reprovaram em cálculo seguiram no curso, nas demais disciplinas o percentual dos alunos que seguem no curso não chega a 3%.

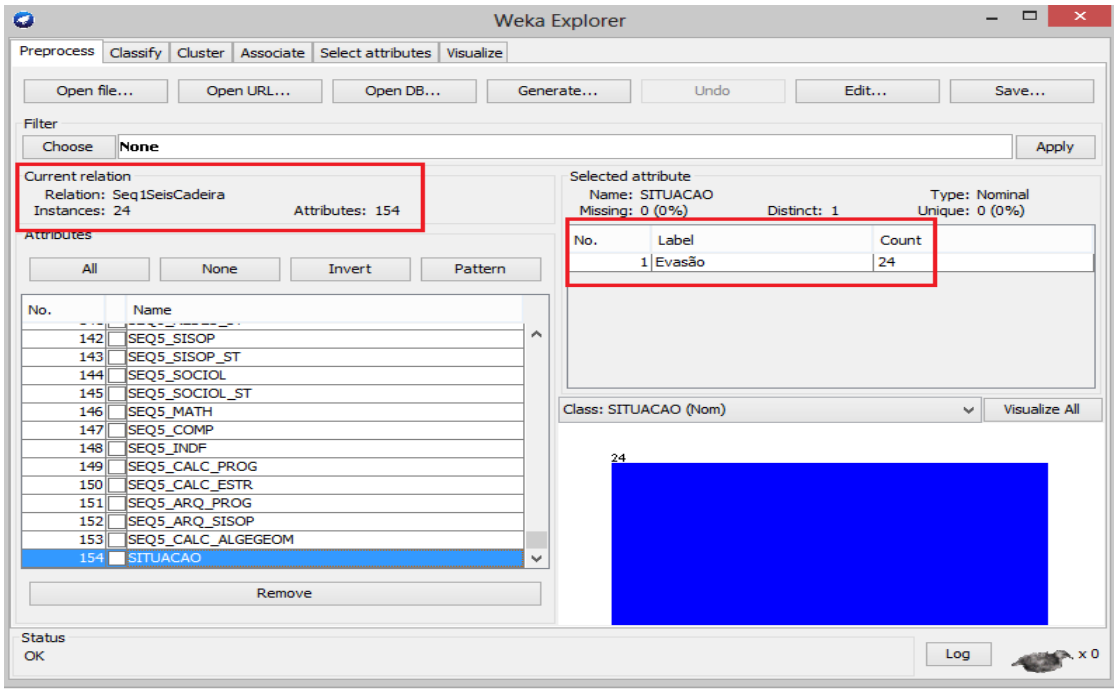
7.3.3 Experimento C

Para a realização do experimento C, os dados da tabela HISTORICO foram separados pela quantidade de disciplinas que cada aluno cursou no primeiro semestre, sendo encontrado alunos que cursaram de uma até seis disciplinas.

Neste experimento, foram então criados seis arquivos, um para cada quantidade de disciplinas encontrada. Os dados novamente foram submetidos ao algoritmo de classificação J48 em validação cruzada, utilizando as configurações padrão do WEKA, para tentar descobrir se existe um padrão entre a quantidade de disciplinas cursada na primeira sequência de disciplinas do curso.

A Figura 40 mostra o perfil dos alunos que fizeram seis disciplinas no primeiro semestre, foram selecionados 24 alunos que atenderam a esse quesito, onde todos os 24 alunos evadiram.

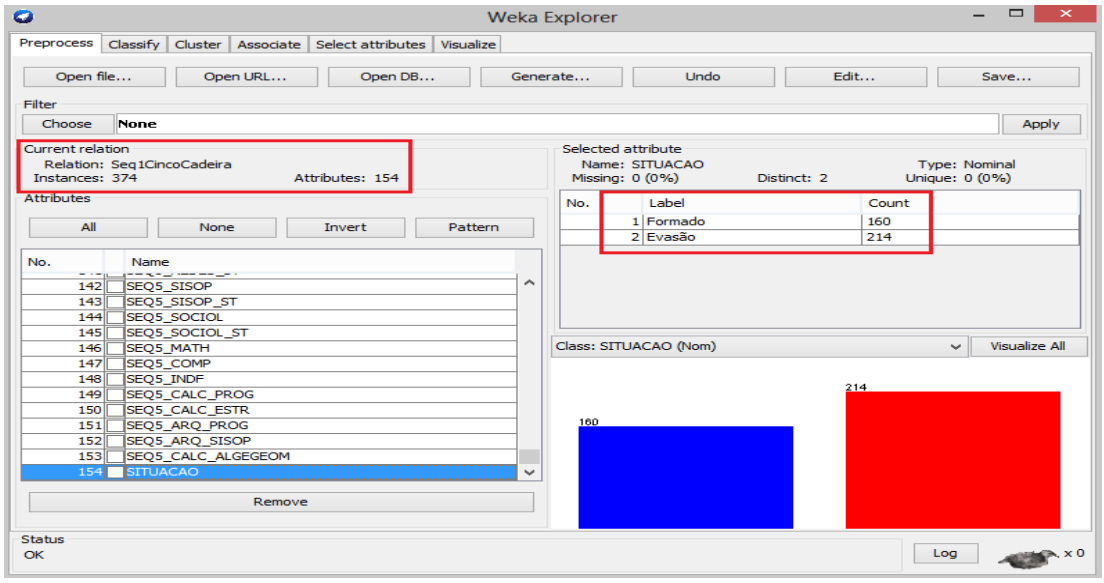
Figura 40 - Perfil dos alunos que fizeram seis disciplinas no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

A Figura 41 mostra o perfil dos alunos que fizeram cinco disciplinas no primeiro semestre, foram selecionados 374 alunos que atenderam a esse quesito, destes 160 são alunos que se formaram e 214 alunos que evadiram.

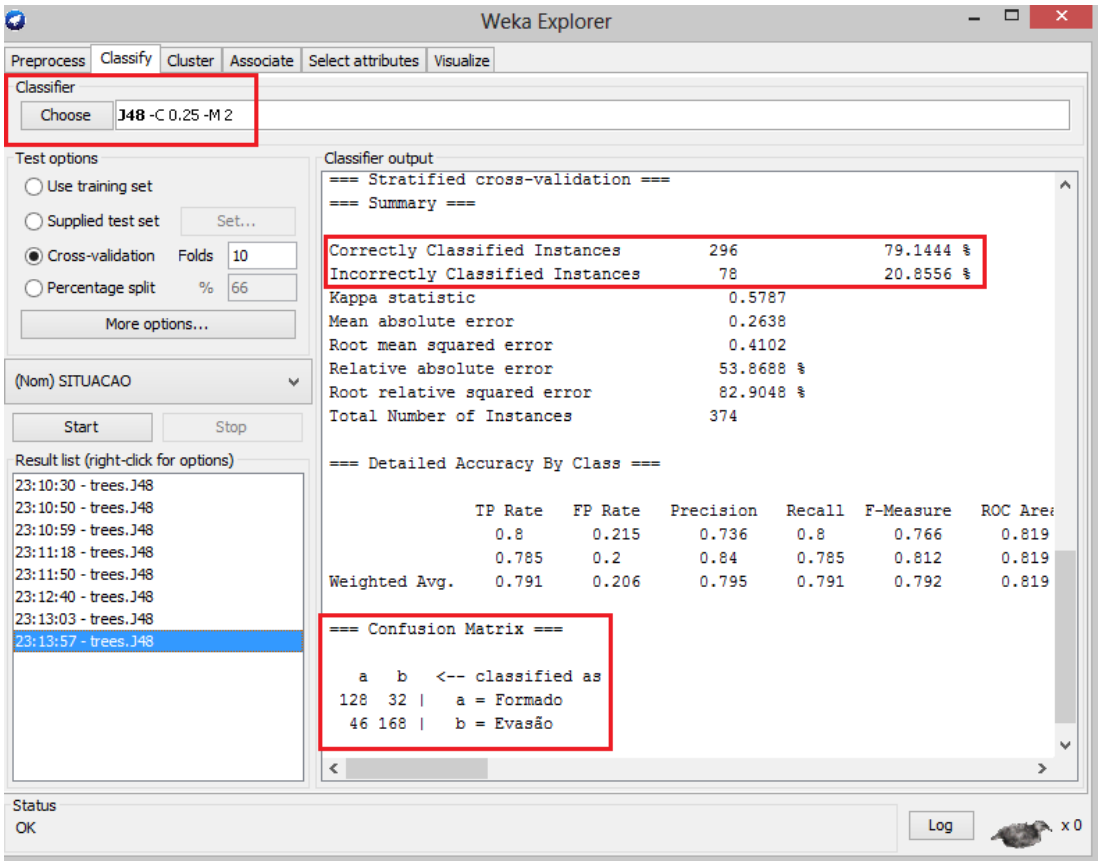
Figura 41 - Alunos que fizeram cinco disciplinas no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 42 é exibido o resultado do experimento de classificação dos alunos que fizeram cinco disciplinas no primeiro semestre. A precisão de acertos do classificador neste caso foi de mais de 79%. A matriz de confusão mostra que dos 214 alunos que evadiram, 168 foram classificados de forma correta.

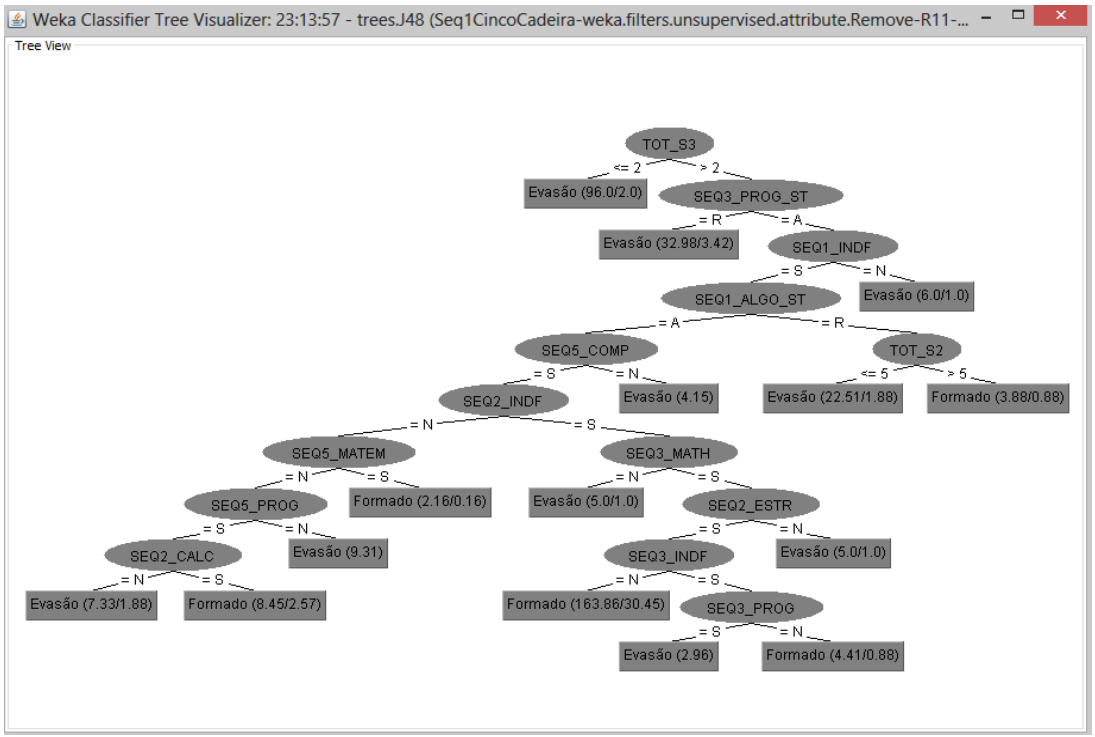
Figura 42 - Resultado do experimento C2



Fonte: WEKA. Adaptado pelo Autor.

A Figura 43 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que fizeram cinco disciplinas no primeiro semestre, onde pode-se ver que o total de disciplinas cursadas no terceiro semestre destes alunos no curso tem forte influência no abandono ou não do curso, outro fator que contribui para a evasão nesses casos é o aluno não fazer disciplina de programação no terceiro semestre.

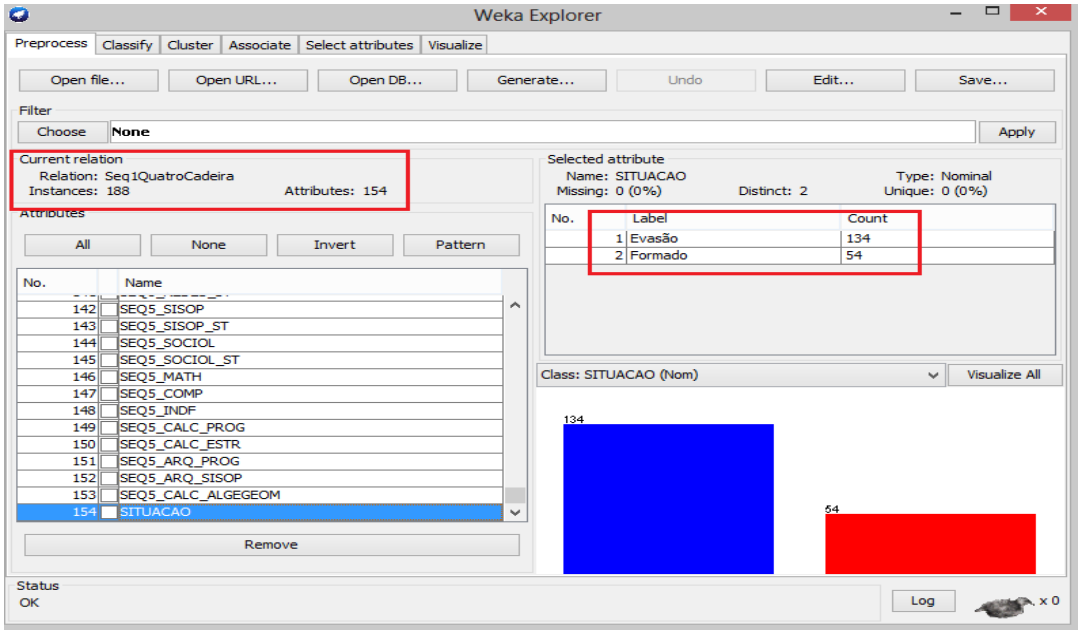
Figura 43 - Árvore gerada pelo experimento C2



Fonte: WEKA. Adaptado pelo Autor.

A Figura 44 mostra o perfil dos alunos que fizeram quatro disciplinas no primeiro semestre. Foram selecionados 188 alunos que atenderam a esse quesito, destes 54 são alunos que se formaram e 134 alunos que evadiram.

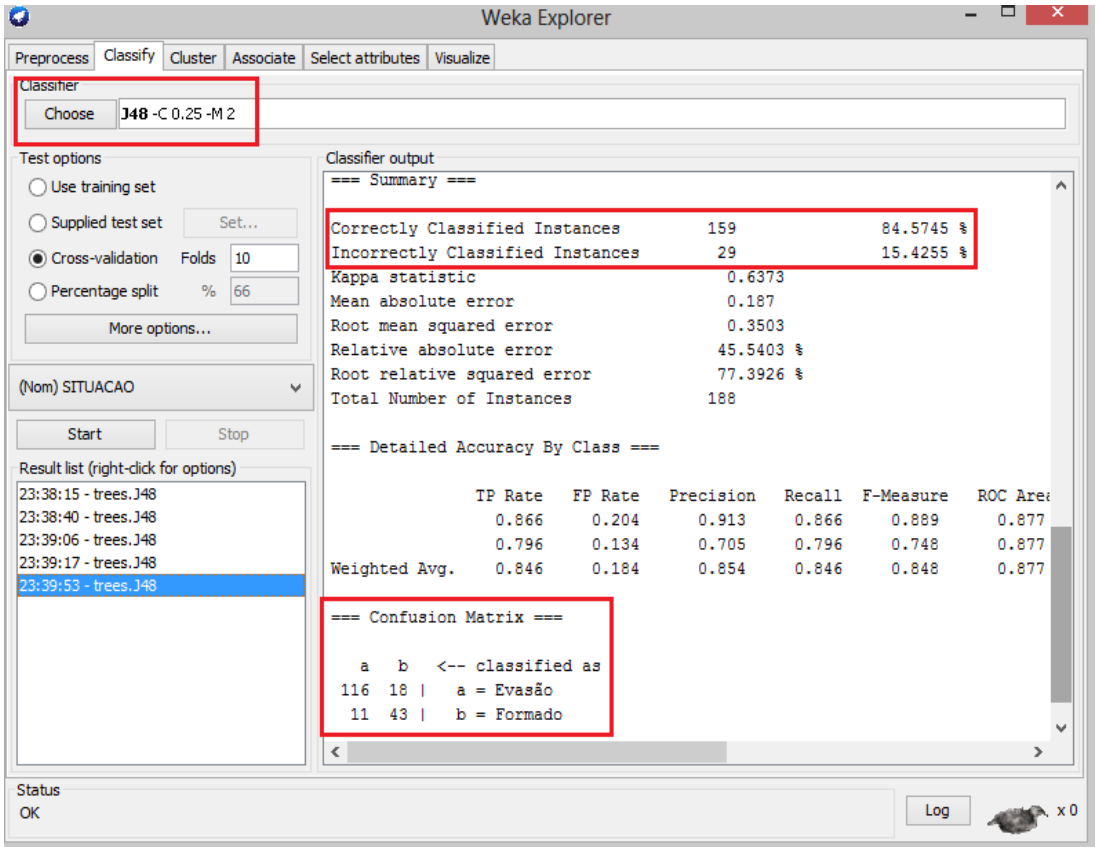
Figura 44 - Alunos que fizeram quatro disciplinas no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 45 é exibido o resultado do experimento de classificação dos alunos que fizeram quatro disciplinas no primeiro semestre. A precisão do classificador desta vez foi de mais de 84% de acertos, classificando apenas 18 dos 134 alunos que evadiram de forma incorreta.

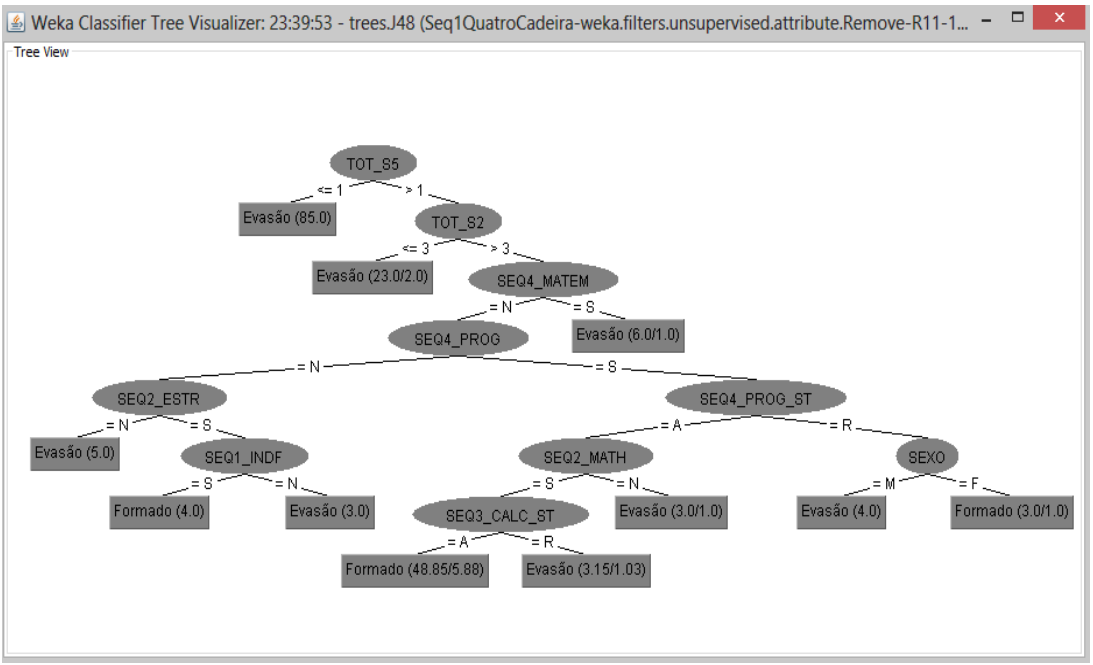
Figura 45 - Resultado do experimento C3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 46 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que fizeram quatro disciplinas no primeiro semestre, onde pode-se ver que o total de disciplinas cursadas no quinto e no segundo semestre destes alunos no curso tem forte influência no abandono ou não do curso por parte desses alunos.

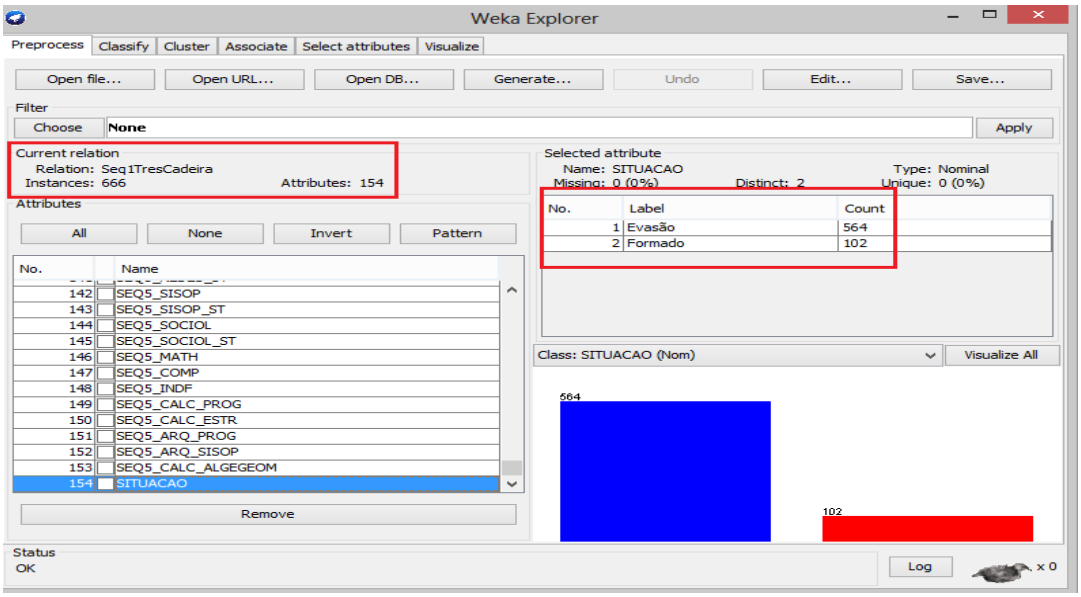
Figura 46 - Árvore gerada pelo experimento C3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 47 mostra o perfil dos alunos que fizeram três disciplinas no primeiro semestre. Foram selecionados 666 alunos que atenderam a esse quesito, destes 102 são alunos que se formaram e 564 alunos que evadiram.

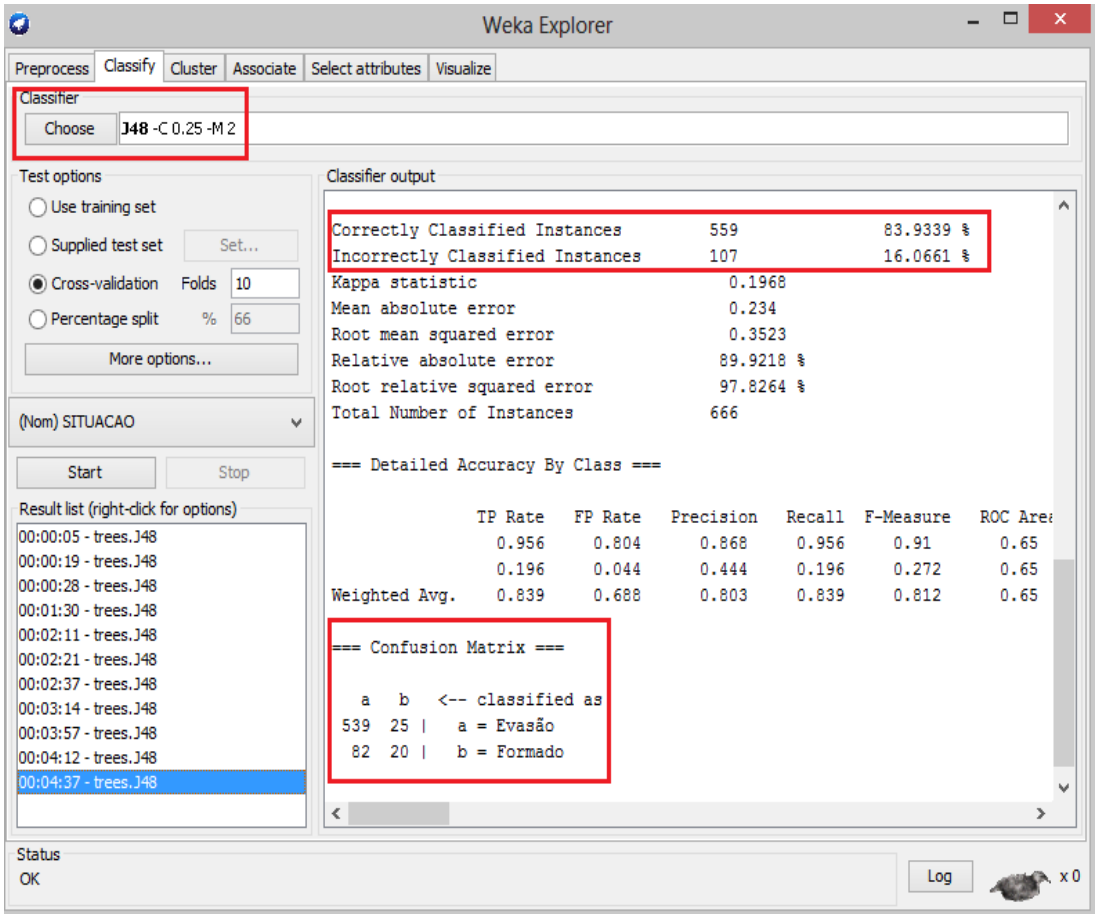
Figura 47 - Alunos que fizeram três disciplinas no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 48 é exibido o resultado do experimento de classificação dos alunos que fizeram três disciplinas no primeiro semestre. Com precisão de mais de 83%, o classificador foi capaz de acertar 539 dos 564 casos de evasão dos alunos.

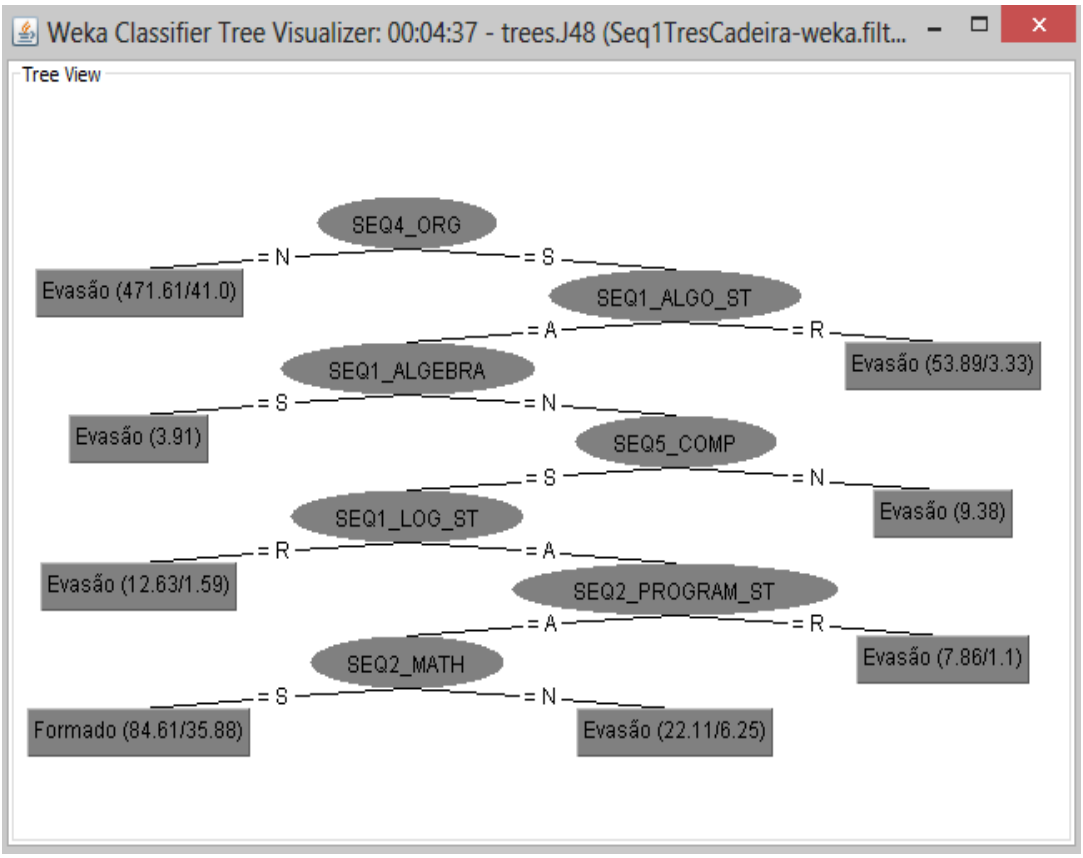
Figura 48 - Resultado do experimento C4



Fonte: WEKA. Adaptado pelo Autor.

A Figura 49 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que fizeram três disciplinas no primeiro semestre, onde pode-se ver que os alunos que não cursaram as disciplinas de Organização de computadores no quarto semestre têm grande probabilidade de abandonar o curso, seguido dos alunos que reprovaram em Algoritmos no primeiro semestre.

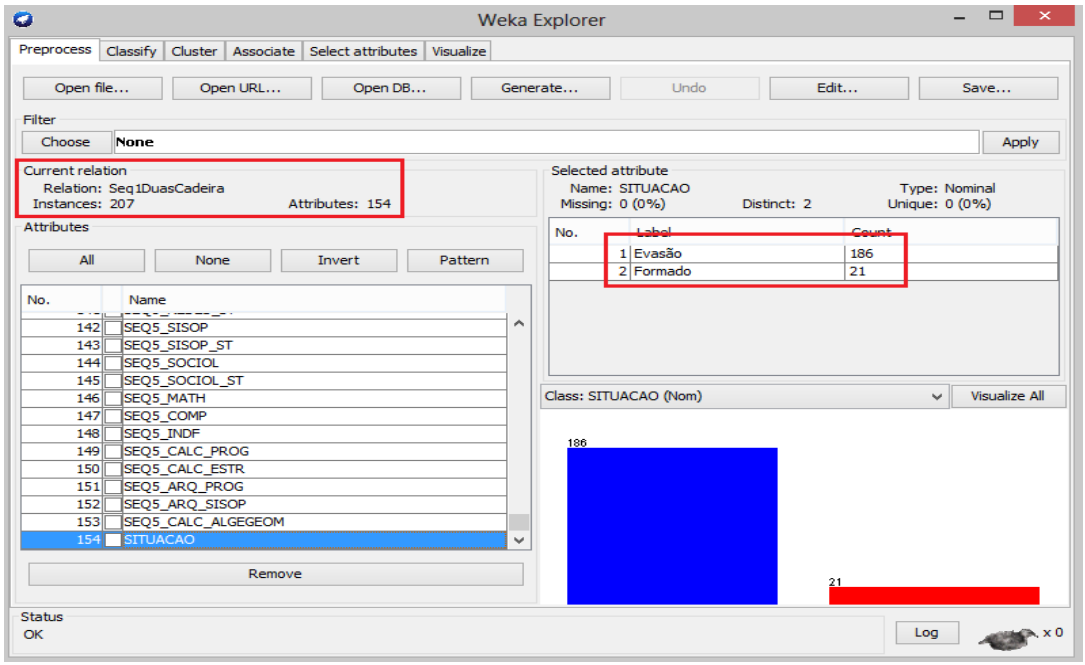
Figura 49 - Árvore gerada pelo experimento C4



Fonte: WEKA. Adaptado pelo Autor.

A Figura 50 mostra o perfil dos alunos que fizeram duas disciplinas no primeiro semestre. Foram selecionados 207 alunos que atenderam a esse quesito, destes 21 são alunos que se formaram e 186 alunos evadiram.

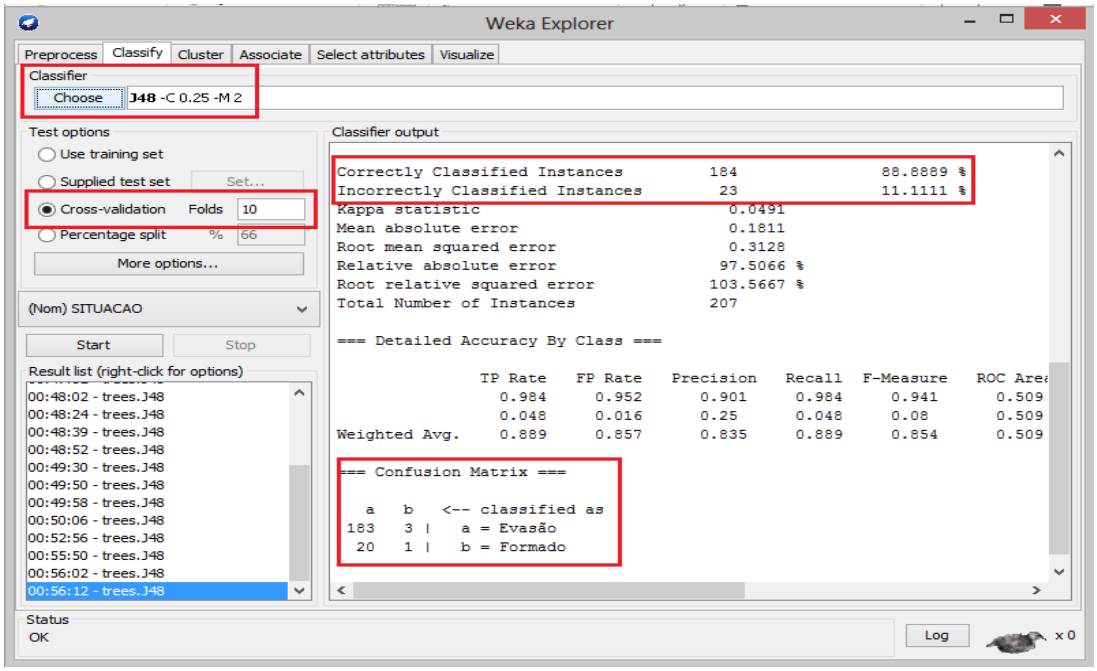
Figura 50 - Alunos que fizeram duas disciplinas no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 51 é exibido o resultado do experimento de classificação dos alunos que fizeram duas disciplinas no primeiro semestre. Com mais de 88% de acertos, o classificador acertou 183 dos 186 casos de evasão deste grupo de alunos.

Figura 51 - Resultado do experimento C5

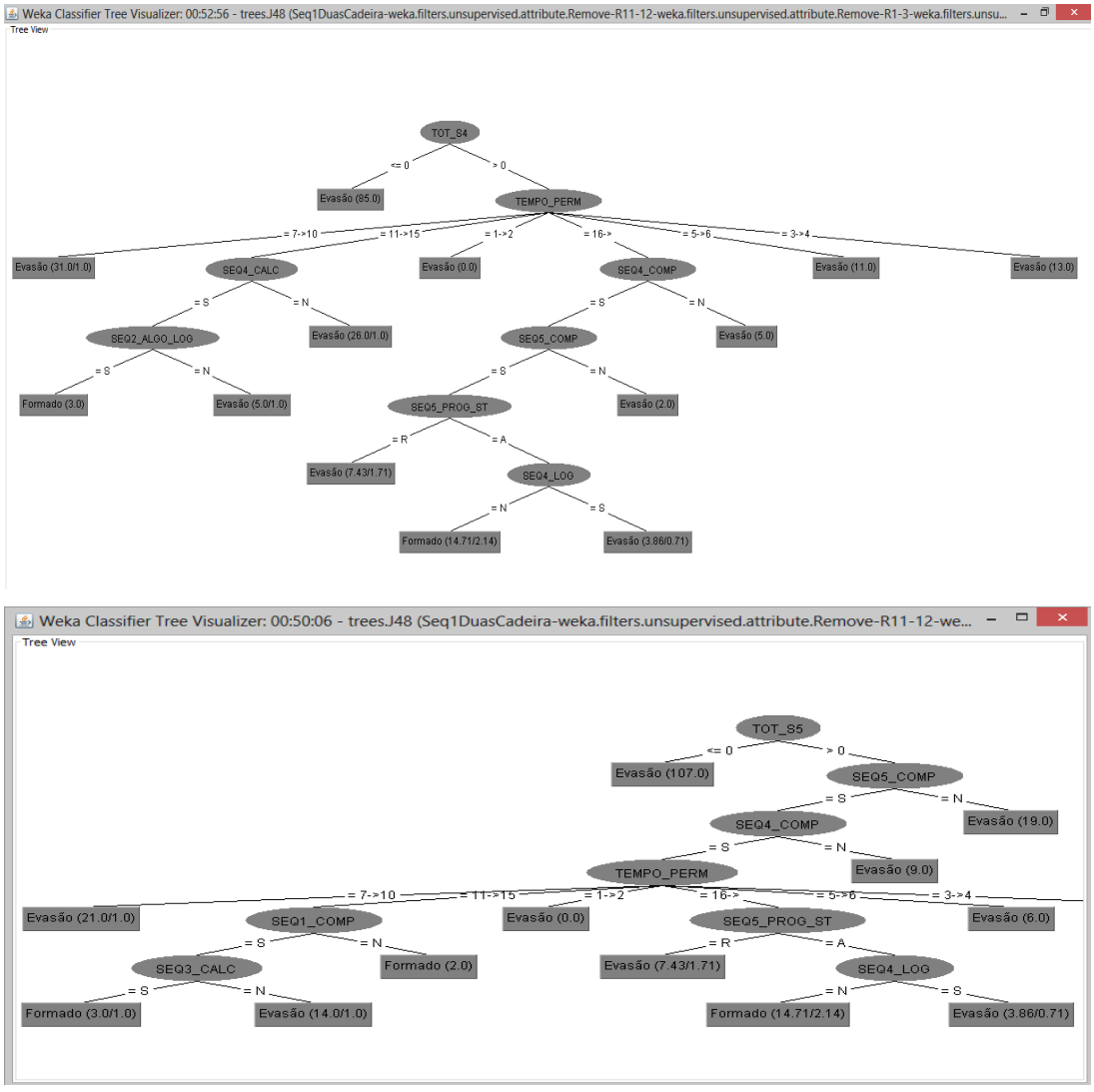


Fonte: WEKA. Adaptado pelo Autor.

A Figura 52 mostra duas árvores de decisão geradas pelo algoritmo de classificação J48 para o experimento com os alunos que fizeram duas disciplinas no primeiro semestre, durante os experimentos, vários atributos são testados, alguns são removidos, depois adicionados e outros removidos, até que se ache os atributos que façam algum sentido entre si.

Neste experimento, pode-se ver que foi removido o atributo TOT_S5 (total de disciplinas cursadas no semestre 5) da segunda árvore, tornando assim o atributo TOT_S4 (total de disciplinas cursadas no semestre 4) como o atributo com maior ganho de informação. Esses testes são feitos de forma exaustiva durante todos os experimentos, até que se encontre alguma informação útil que possa estar oculta nos dados analisados.

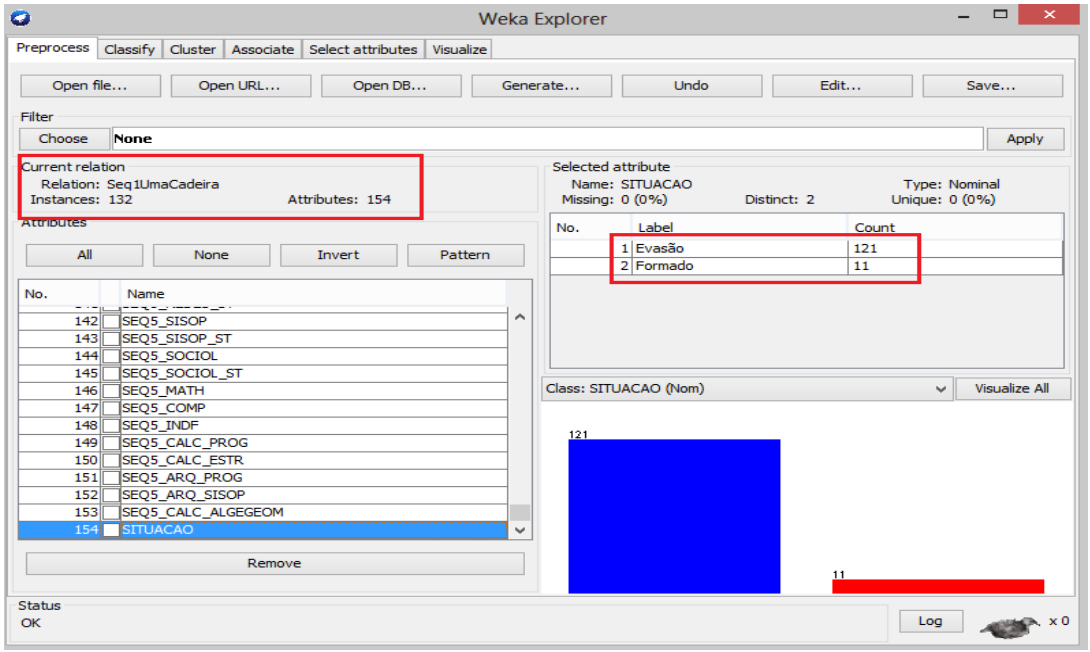
Figura 52 - Árvores gerada pelo experimento C5



Fonte: WEKA. Adaptado pelo Autor.

A Figura 53 mostra o perfil dos alunos que fizeram uma disciplina no primeiro semestre. Foram selecionados 132 alunos que atenderam a esse quesito, destes 11 são alunos que se formaram e 121 alunos que evadiram.

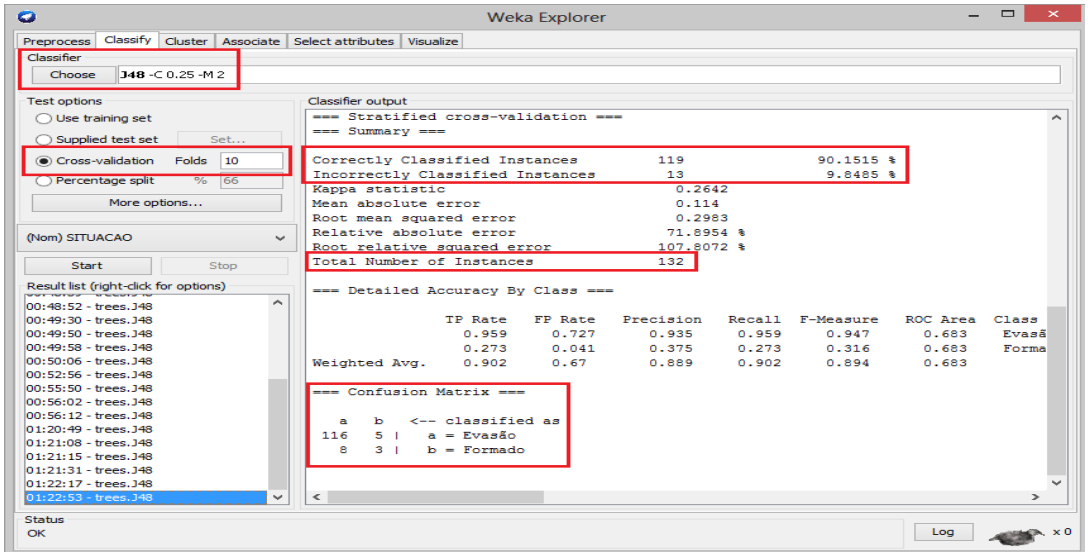
Figura 53 - Alunos que fizeram uma disciplina no 1° semestre



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 54 é exibido o resultado do experimento de classificação dos alunos que fizeram uma disciplina no primeiro semestre. O classificador apresentou precisão de mais de 90% acertando a maioria dos casos de evasão.

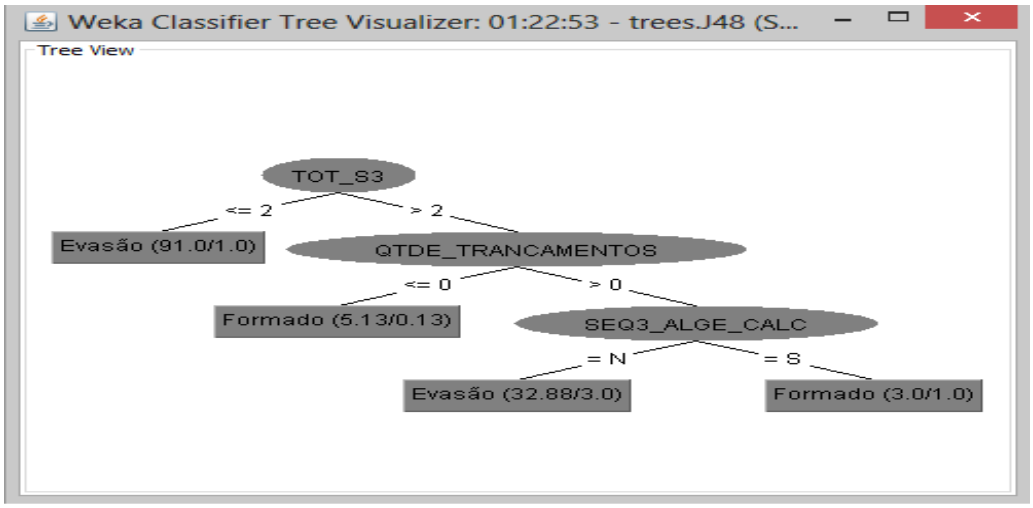
Figura 54 - Resultado do experimento C6



Fonte: WEKA. Adaptado pelo Autor.

A Figura 55 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que fizeram uma disciplina no primeiro semestre, onde pode-se ver que a maioria dos alunos que cursaram uma disciplina no primeiro semestre e menos que três no segundo semestre evadiram do curso.

Figura 55 - Árvores gerada pelo experimento C6



Fonte: WEKA. Adaptado pelo Autor.

Podemos ver pelos resultados obtidos neste experimento que todos os alunos que fizeram mais que cinco disciplinas no primeiro semestre evadiram. Também fica visível que a grande maioria dos alunos que fazem menos que três disciplinas no primeiro semestre acabam evadindo do curso, menos de 9% destes se formam.

7.3.4 Experimento D

Para a realização do experimento D, os dados da tabela HISTORICO foram separados pela quantidade de disciplinas que cada aluno cursou em cada um dos primeiros cinco semestres.

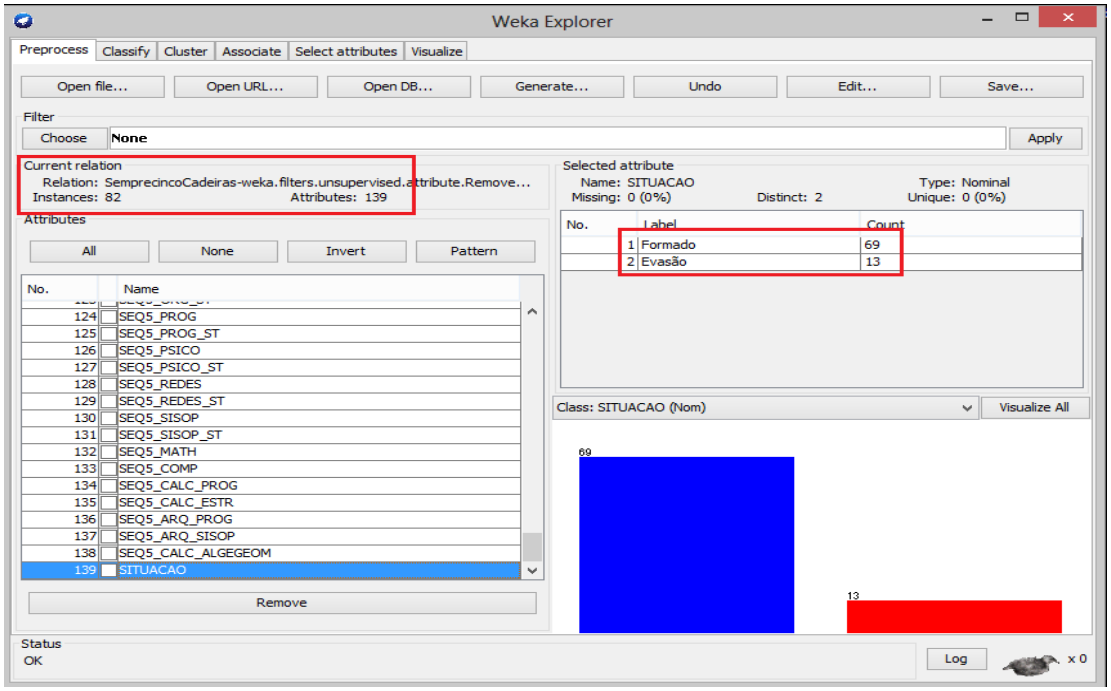
Neste experimento foram criados quatro arquivos, um para os alunos que sempre cursaram as cinco disciplinas nos cinco semestres, um para os que sempre cursaram três disciplinas, um para os alunos que sempre cursaram três ou mais disciplinas, mas que não estão no primeiro grupo que sempre cursou as cinco

disciplinas e por fim, um arquivo para os alunos que sempre cursaram menos que três disciplinas nos cinco primeiros semestres do curso.

Os dados novamente foram submetidos ao algoritmo de classificação J48 em validação cruzada, utilizando as configurações padrão do WEKA, para tentar descobrir se existe um padrão entre a quantidade de disciplinas cursada nos cinco primeiros semestres do aluno no curso e a evasão destes alunos.

A Figura 56 mostra o perfil dos alunos que sempre fizeram cinco disciplinas nos cinco primeiros semestres do curso, foram selecionados 82 alunos que atenderam a esse quesito, destes 69 se formaram e 13 alunos evadiram, um dos poucos casos onde os números da evasão são menores do que número de formandos.

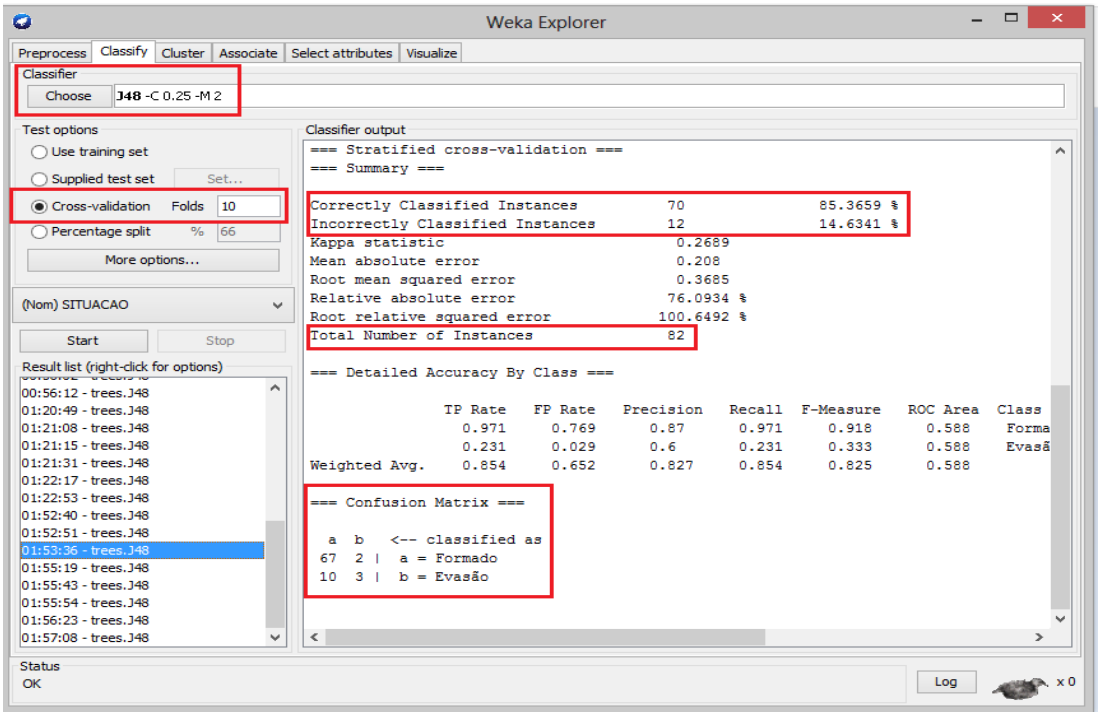
Figura 56 - Alunos que sempre fizeram cinco disciplinas



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 57 é exibido o resultado do experimento de classificação dos alunos que sempre fizeram cinco disciplinas nos cinco primeiros semestres. A precisão do classificador foi de mais de 85%, acertando 67 dos 69 alunos que se formaram, e classificando de forma correta três dos 13 alunos que evadiram.

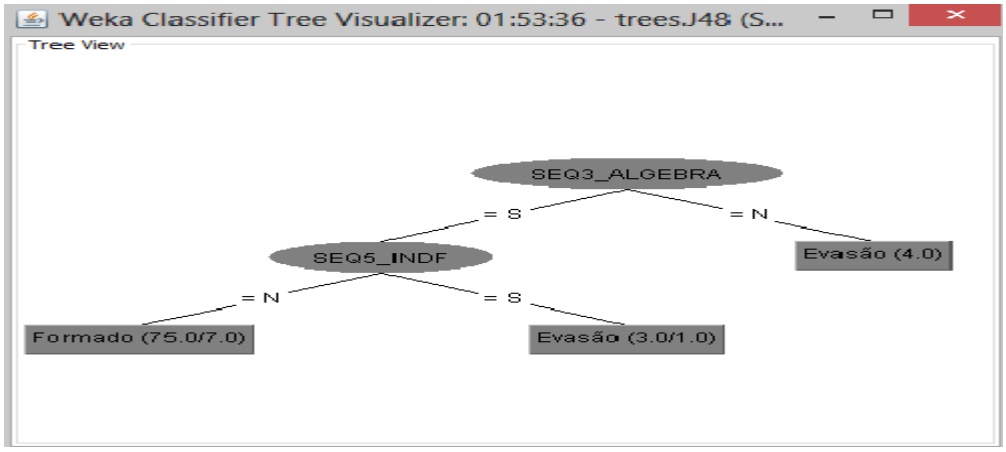
Figura 57 - Resultado do experimento D1



Fonte: WEKA. Adaptado pelo Autor.

A Figura 58 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que sempre fizeram cinco disciplinas nos primeiros semestres, onde pode-se ver que quem faz álgebra no terceiro semestre e não faz disciplinas do tipo HUMAN SEQ5_INDF no quinto semestre, tem grande probabilidade de se formar.

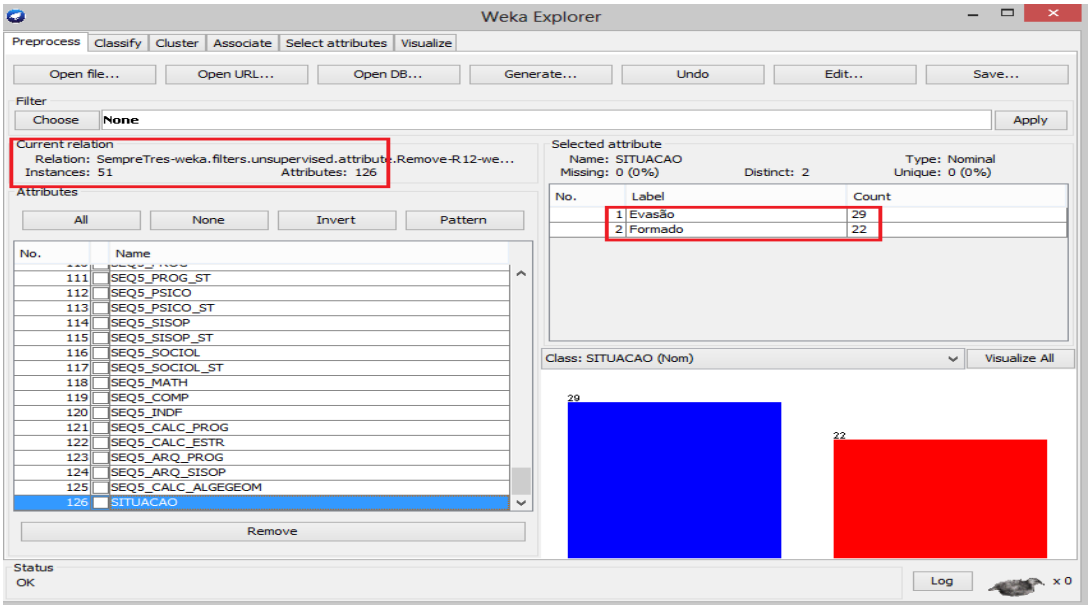
Figura 58 - Árvore gerada pelo experimento D1



Fonte: WEKA. Adaptado pelo Autor.

A Figura 59 mostra o perfil dos alunos que sempre fizeram três disciplinas nos cinco primeiros semestres do curso, foram selecionados 51 alunos que atenderam a esse quesito, onde 22 alunos se formaram e 29 alunos evadiram.

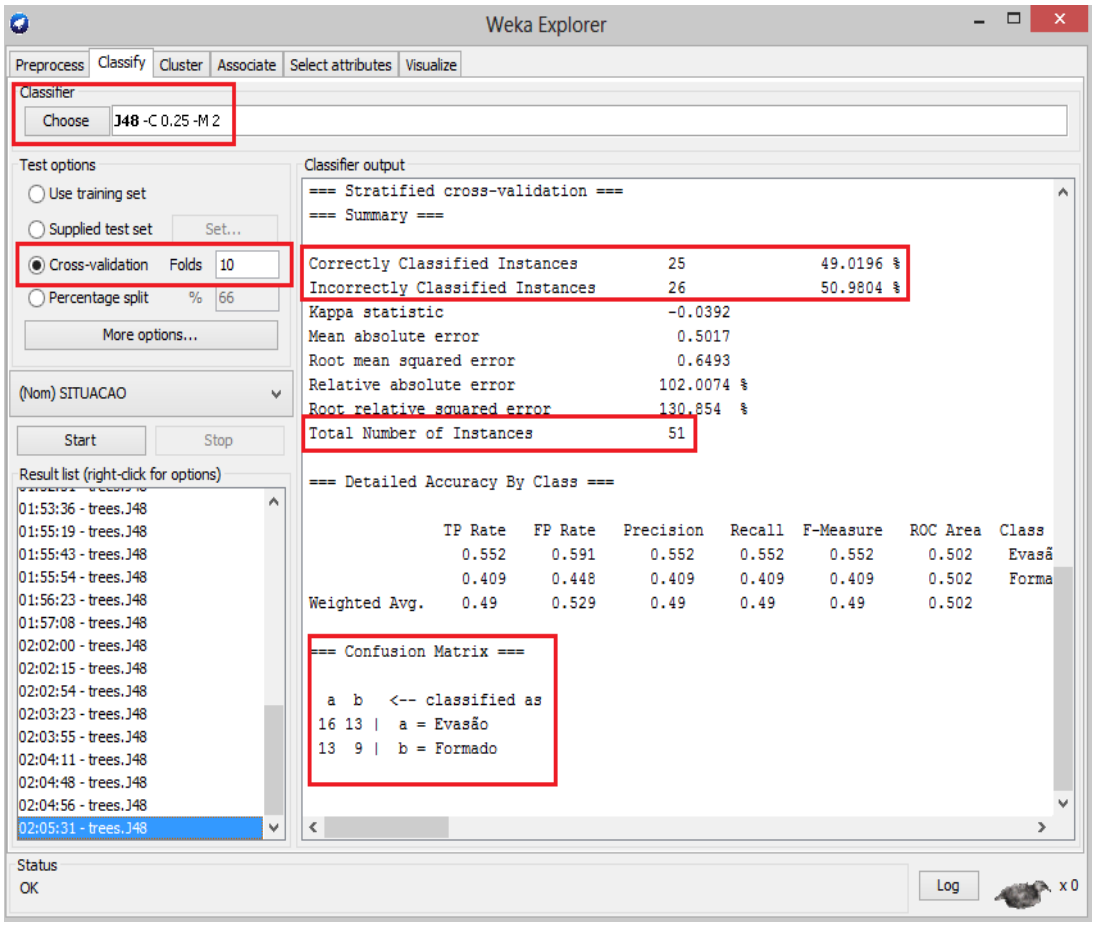
Figura 59 - Alunos que sempre fizeram três disciplinas



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 60 é exibido o resultado do experimento de classificação dos alunos que sempre fizeram três disciplinas nos cinco primeiros semestres. Neste experimento a precisão do classificador caiu consideravelmente, apresentando precisão de menos de 50% de acertos, ou seja, errou a metade dos alunos que classificou. Dos 29 alunos que evadiram, apenas 16 foram classificados de forma correta.

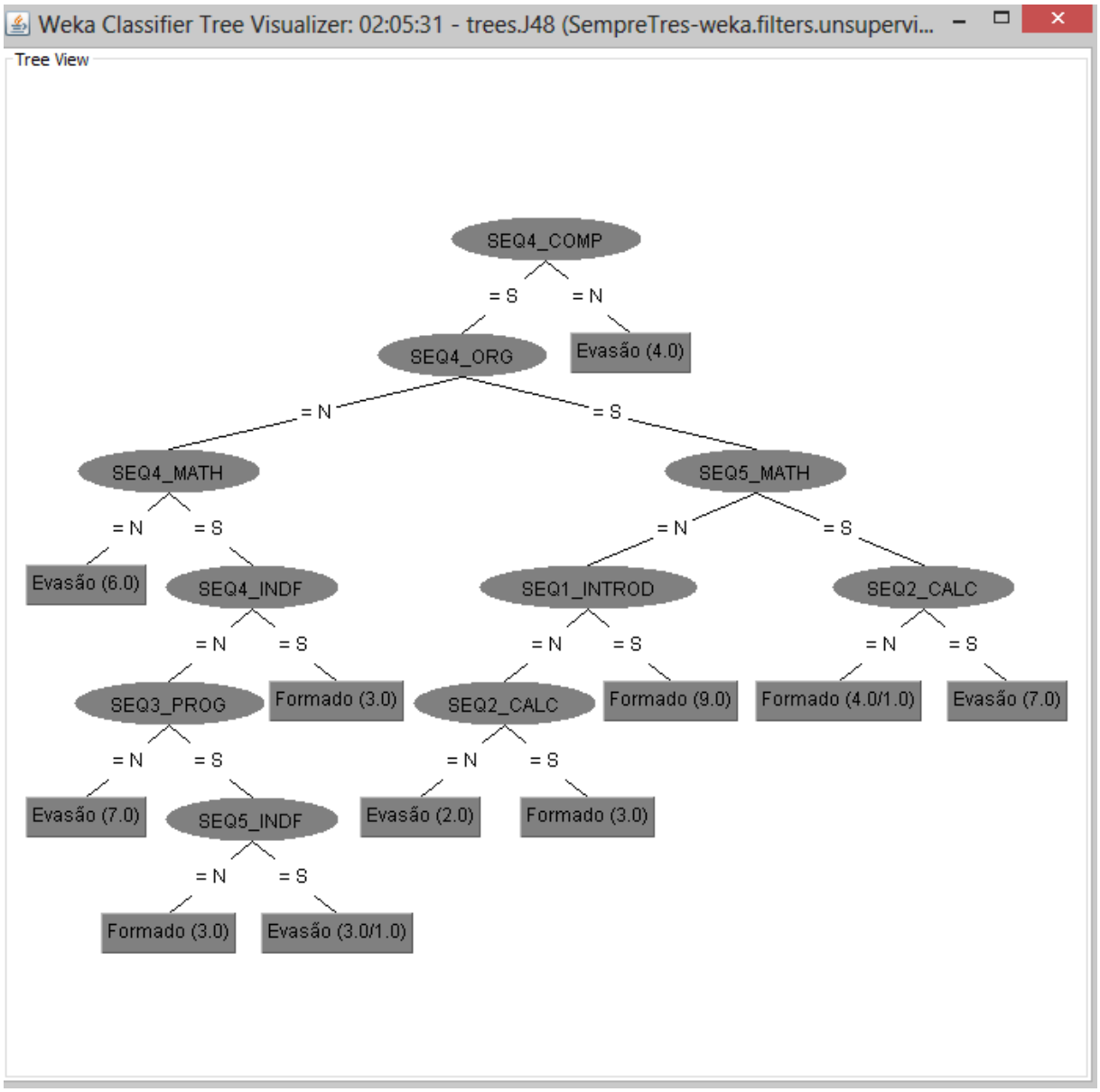
Figura 60 - Resultado do experimento D2



Fonte: WEKA. Adaptado pelo Autor.

A Figura 61 mostra a árvore de decisão gerada pelo algoritmo de classificação J48 para o experimento com os alunos que sempre fizeram três disciplinas nos primeiros semestres, onde pode-se ver que o tipo de disciplina, MATH, COMP, HUMAN tem bastante influência sobre a evasão ou sucesso do aluno no curso.

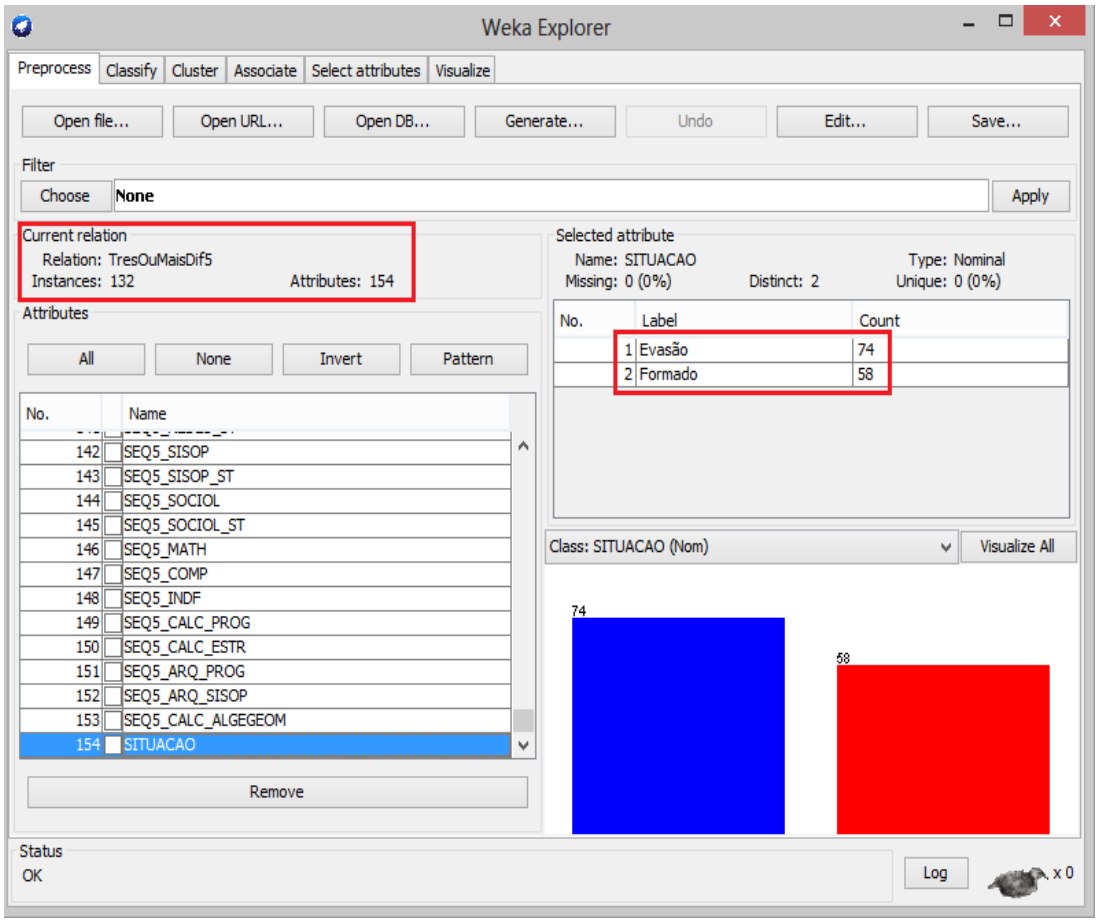
Figura 61 - Árvore gerada pelo experimento D2



Fonte: WEKA. Adaptado pelo Autor.

A Figura 62 mostra o perfil dos alunos que sempre fizeram três ou mais disciplinas nos cinco primeiros semestres do curso, mas que não estão no perfil dos que sempre fizeram as cinco disciplinas. Foram selecionados 132 alunos que atenderam a esse quesito, onde 58 alunos se formaram e 74 alunos evadiram.

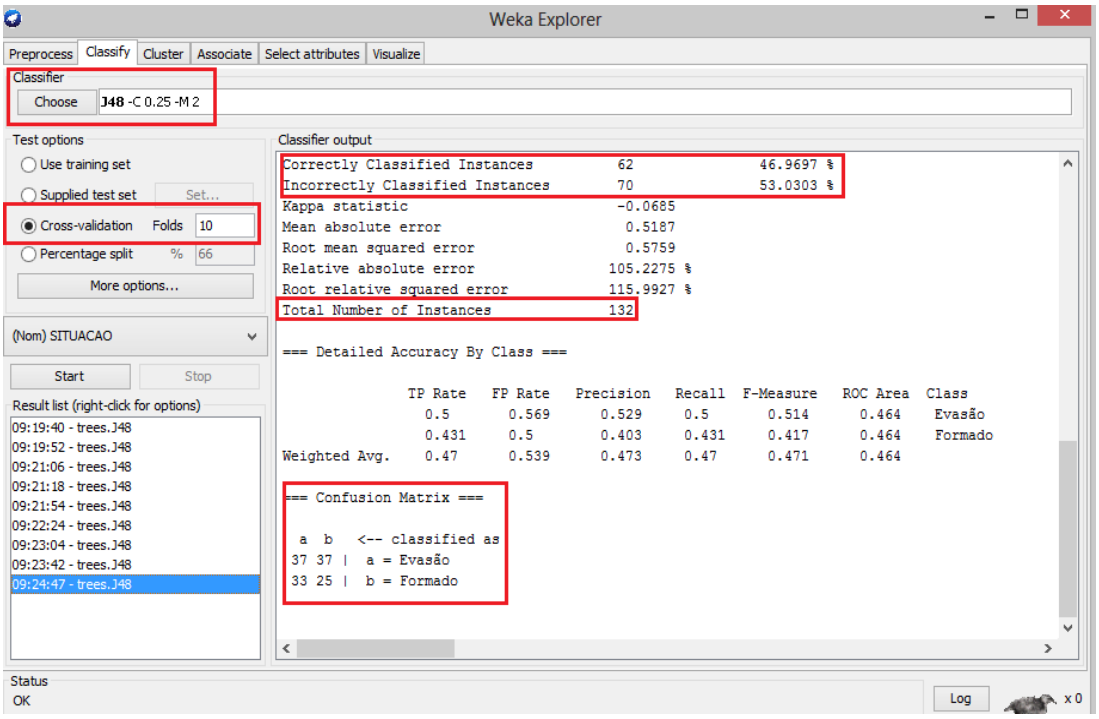
Figura 62 - Alunos que fizeram três ou mais disciplinas



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 63 é exibido o resultado do experimento de classificação dos alunos que sempre fizeram três ou mais disciplinas nos cinco primeiros semestres do curso, mas que não estão no perfil dos que sempre fizeram as cinco disciplinas. Novamente a precisão do classificador ficou abaixo de 50%.

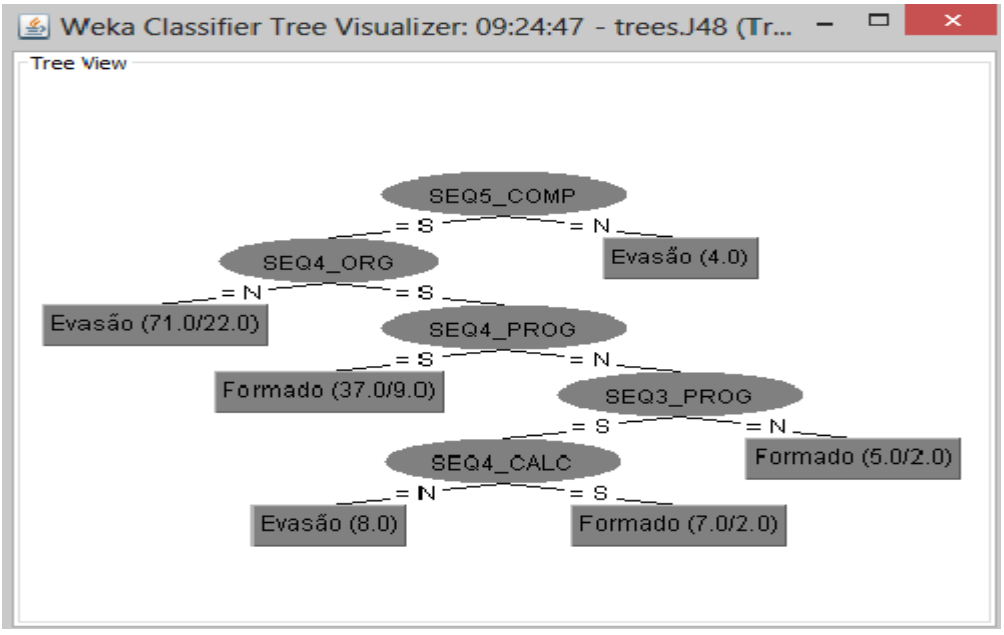
Figura 63 - Resultado do experimento D3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 64 mostra a árvore de decisão gerada pelo experimento D3, o atributo fez disciplina de computação no semestre 5 é o atributo com maior ganho de informação.

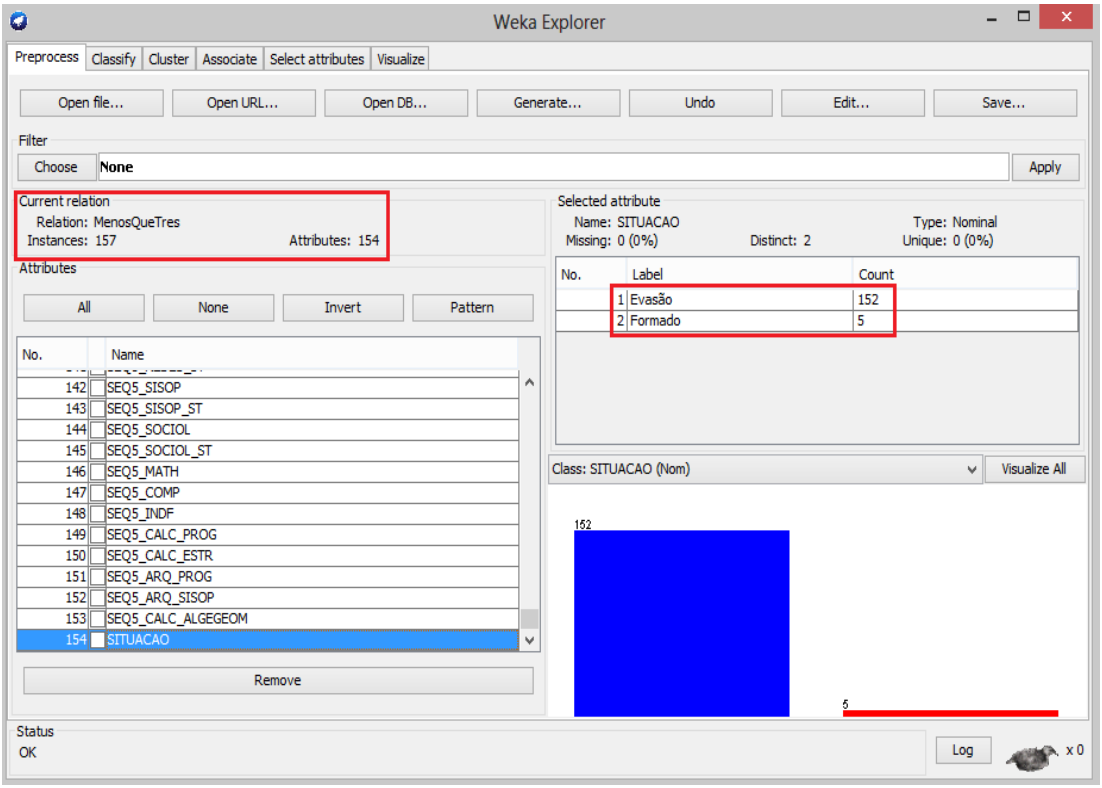
Figura 64 - Árvore gerada pelo experimento D3



Fonte: WEKA. Adaptado pelo Autor.

A Figura 65 mostra o perfil dos alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestres do curso. Foram selecionados 157 alunos que atenderam a esse quesito, onde 5 alunos se formaram e 152 alunos evadiram.

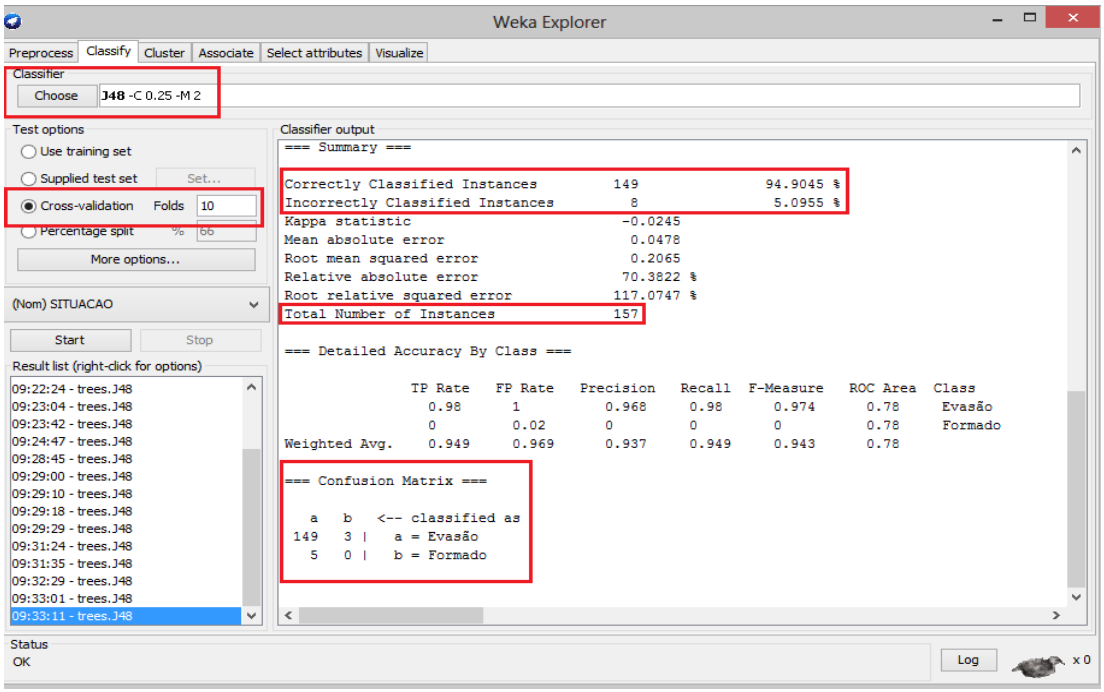
Figura 65 - Perfil dos alunos que fizeram menos que três disciplinas



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 66 é exibido o resultado do experimento de classificação dos alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestres. A precisão do classificador desta vez ficou acima dos 94% de acertos, acertando 149 dos 154 alunos que evadiram, mas não acertou nenhum dos 5 alunos que se formaram.

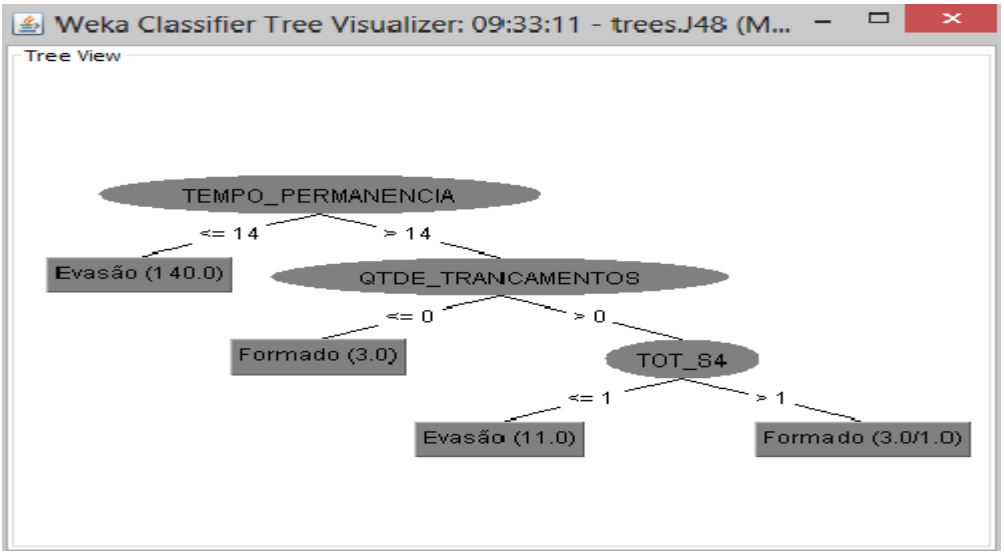
Figura 66 - Resultado do experimento D4



Fonte: WEKA. Adaptado pelo Autor.

A Figura 67 mostra a árvore de decisão gerada pelo experimento D4 com os alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestres, mostra que os poucos alunos que se formam, o fazem depois do 14° período, desde que entrou no curso.

Figura 67 - Árvore gerada pelo experimento D4



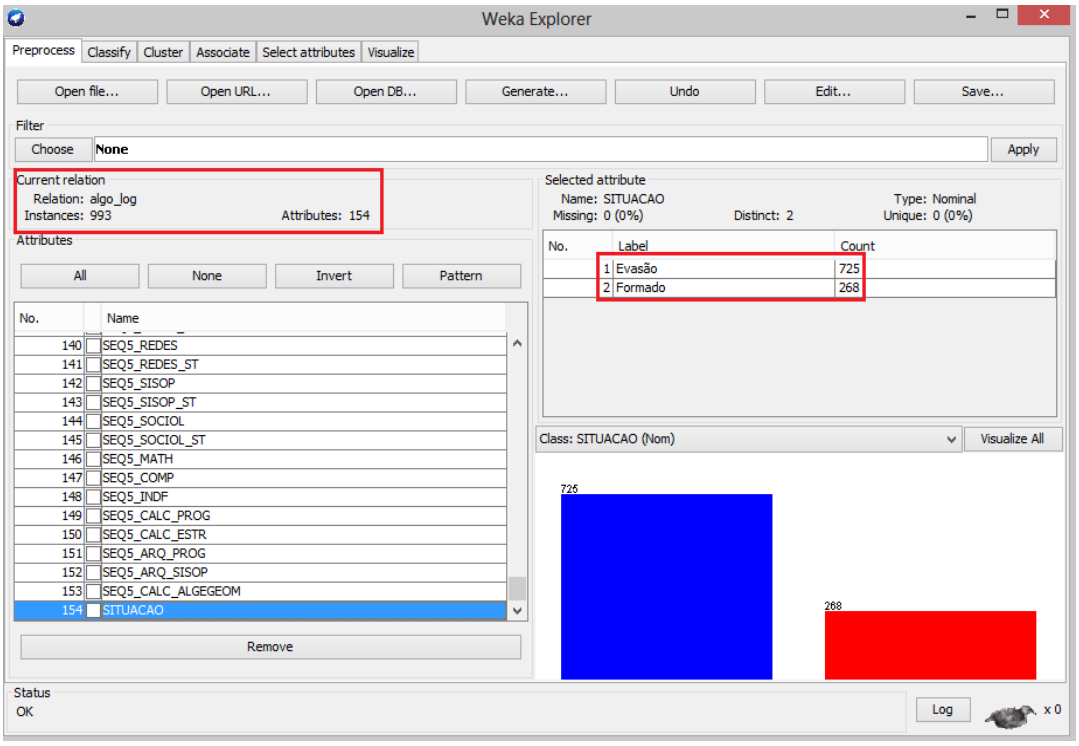
Fonte: WEKA. Adaptado pelo Autor.

Os resultados deste experimento, mostram que alunos que seguem um padrão do número de disciplinas cursadas, e se esse número de disciplinas é maior ou igual a três, são menos propensos a evadir. Por exemplo, dos 82 alunos que sempre fizeram 5 disciplinas nos cinco primeiros semestres, 69 são formados e apenas 13 evadiram. Outro caso é o dos alunos que sempre fizeram três disciplinas nos cinco semestres, foram encontrados 51 alunos que se encaixavam nesse perfil, dos quais 22 são formados. Os alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestre apresentam um outro padrão, a evasão, uma vez que dos 157 alunos fizeram menos que três disciplinas, apenas cinco concluíram o curso.

7.3.5 Experimento E

A Figura 68 mostra o perfil dos alunos que fizeram Algoritmos e Lógica no mesmo semestre. Foram selecionados 993 alunos que atenderam a esse quesito, onde 268 alunos se formaram e 725 alunos evadiram.

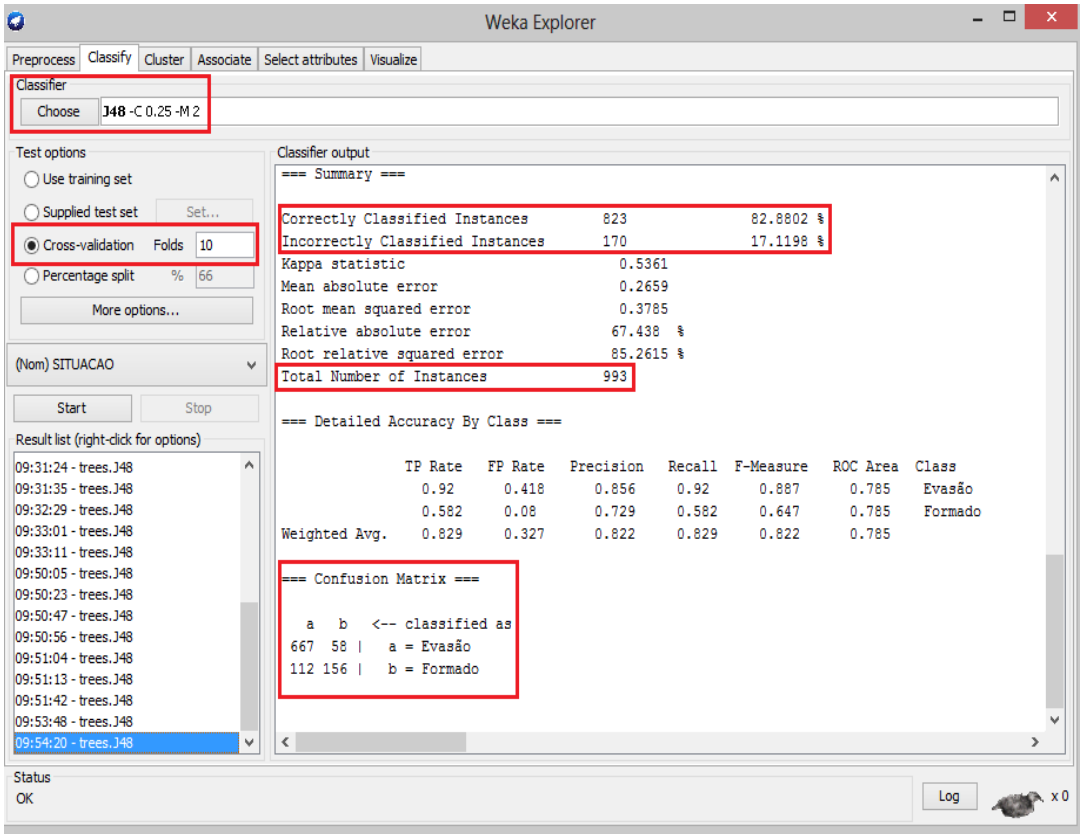
Figura 68 - Perfil dos alunos que fizeram Algoritmos e Lógica Juntos



Fonte: WEKA. Adaptado pelo Autor.

Na Figura 69 é exibido o resultado do experimento de classificação dos alunos que fizeram Algoritmos e Lógica no mesmo semestre. A precisão do classificador ficou acima dos 82%, dos 725 alunos que evadiram, 667 foram classificados de forma correta.

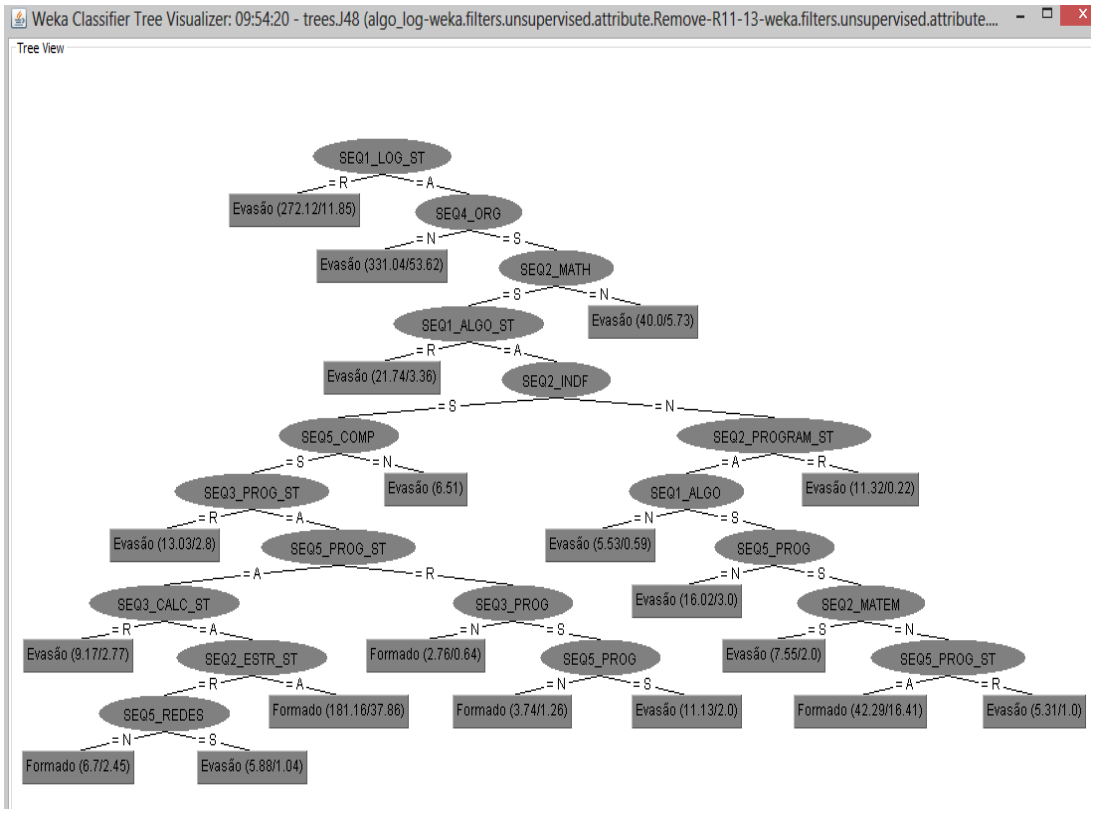
Figura 69 - Resultado do experimento E1



Fonte: WEKA. Adaptado pelo Autor.

A Figura 70 mostra a árvore de decisão gerada pelo experimento E1, onde pode-se ver que quem faz lógica e algoritmos no mesmo semestre, e reprova em lógica tem grande probabilidade de evadir, assim como os que fazem essas disciplinas juntas e não reprova em lógica, mas não faz a disciplina de organização de banco de dados no quarto semestre, também tem grande chance de evadir do curso.

Figura 70 - Árvore gerada pelo experimento E1



Fonte: WEKA. Adaptado pelo Autor.

O resultado do experimento E mostra que cursar lógica e algoritmos no mesmo semestre não é exatamente a solução para a evasão do curso, menos de 30% dos alunos que cursam essas disciplinas no mesmo semestre chega ao final do curso.

7.4 Discussão dos resultados obtidos

Como foi citado na Introdução deste trabalho, e também em “Domínio da base de dados de apoio” (seção 7.1), não foram utilizados os dados pessoais dos alunos da base, nem tampouco dados de ordem socioeconômica. Assim, o trabalho ficou limitado à vida curricular do aluno.

Comparando-se os resultados dos experimentos realizados, verifica-se que as precisões dos classificadores criados na maior parte dos experimentos ficam em torno de 84 a 97%, comparando-se com os trabalhos relacionados (DEKKER *et al*., 2009), (MANHÃES, 2011; WITTEN; FRANK; HALL, 2011), essa pode ser considerada uma boa acurácia, exceto os experimentos D2 e D3 onde a precisão

ficou abaixo de 50%, devido a remoção dos atributos total de disciplinas cursadas e total de disciplinas de cada semestre.

Outra análise que deve-se levar em conta é em relação aos valores das taxas de acerto e erro em que um algoritmo pode diferir do outro na classificação. Um classificador que possui uma grande taxa de erro para falso positivo não é adequado para a solução do problema da evasão, pois ele pode classificar um aluno com risco de evasão como sendo sem risco. Já o erro do algoritmo de classificar um aluno no grupo de risco de evasão sem ele de fato evadir, falso negativo, pode ser considerado um erro menos grave (MANHÃES 2011; DEKKER *et al*., 2009). Neste contexto, a maioria dos experimentos se mostraram eficientes em classificar os alunos no grupo propenso à evasão, foco principal deste trabalho.

Com relação às regras construídas através da associação das disciplinas cursadas nos três primeiros semestres do curso, a principal delas se deu no grupo de alunos que fizeram menos de cinco disciplinas através do algoritmo *Apriori*, onde quem não fez geometria no primeiro semestre, não fez programação no segundo semestre e evadiu do curso (460 ocorrências), também não fez estrutura de dados no segundo semestre (447 ocorrências).

O objetivo do experimento B era investigar se alunos que reprovam nas primeiras disciplinas do curso possuem maior tendência a evadir do curso e se existe algum padrão entre eles. Os resultados mostraram que alunos que reprovam nas primeiras disciplinas, dificilmente continuam no curso. Dos 190 alunos que reprovaram em álgebra, apenas um se formou. Dos 170 alunos que reprovaram em Introdução à computação, dois se formaram. Dos 302 alunos que reprovaram em Cálculo, menos de 10% chegaram ao final do curso. Na disciplina de Algoritmos, 426 alunos reprovaram e apenas 14 se formaram, o mesmo acontece com a disciplina de Lógica para programação, onde penas 11 dos 401 alunos que reprovaram conseguiram se formar.

Os experimentos C e D utilizaram a quantidade de disciplinas cursadas por cada aluno como base. No primeiro experimento os dados foram separados e investigados pelo total de disciplinas cursadas somente no primeiro semestre, e no segundo experimento foram utilizados os totais de disciplinas do primeiro ao quinto semestre do aluno no curso.

O experimento C mostrou que fazer mais que cinco disciplinas no primeiro semestre é uma péssima ideia, todos alunos que fizeram isso evadiram. Também

mostra que a grande maioria dos alunos que fazem menos que três disciplinas no primeiro semestre acabam evadindo do curso.

Com o experimento D, pode-se perceber que alunos que seguem um padrão do número de disciplinas cursadas, ou seja, semestre após semestre fazem a mesma quantidade de disciplinas, são menos propensos a evadir, por exemplo, dos 82 alunos que sempre fizeram cinco disciplinas nos cinco primeiros semestres, 69 são formados e apenas 13 evadiram (um dos poucos casos onde o número de evasão é menor do que o número de formados). Outro caso é o dos alunos que sempre fizeram três disciplinas nos cinco semestres, foram encontrados 51 alunos que se encaixavam nesse perfil, dos quais 22 são formados. Já para os alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestres, o padrão estabelecido é a evasão, uma vez que dos 157 casos em que alunos fizeram menos que três disciplinas, apenas cinco se formaram.

O objetivo do experimento E era responder se os alunos que cursam algoritmos e lógica no mesmo semestre têm maiores chances de se formarem, porém os dados mostram que isso não é exatamente o que acontece na realidade. Dos 993 alunos que atenderam a esse quesito, ou seja, que cursaram algoritmos e lógica no mesmo semestre, apenas 268 deles se formaram, menos de 30% do total de alunos que fazem essas duas disciplinas juntas obtém sucesso.

Os resultados mostram que ao fazerem essas disciplinas juntas, os alunos que reprovam em Lógica acabam por evadir na grande maioria dos casos, conforme pode ser visto na Figura 70, e quando são aprovados em Lógica mas reprovam na disciplina de organização de banco de dados no quarto semestre, ou não cursam disciplinas do tipo MATH no semestre seguinte, também acabam evadindo.

Portanto, a resposta para as seguintes perguntas foram:

a) Existe uma sequência de disciplinas feitas que provocam a evasão até o terceiro semestre? Não ficou claro nos resultados obtidos pelos experimentos se existe uma sequência de disciplina cursadas até o terceiro semestre que faça os alunos evadirem, os algoritmos penderam mais para as disciplinas que não foram cursadas pelos alunos que evadiram neste período do que para as disciplinas cursadas por eles.

b) Quem sempre faz cinco disciplinas nos cinco primeiros semestres evade ou não? E quem sempre faz três ou menos disciplinas? Devido ao alto índice da evasão no curso, mesmo entre os alunos que sempre cursam as cinco disciplinas nos cinco primeiros semestres ocorre evasão. Porém nota-se que os alunos que seguem um padrão no número de disciplinas cursadas por período, e quando esse número de disciplinas é maior ou igual a três, tem muito mais chances de obter sucesso. Dos alunos que sempre fizeram as cinco disciplinas, 84% se formaram, 56% dos que sempre fizeram três disciplinas também chegaram ao final do curso. Já os alunos que sempre fizeram menos que três disciplinas nos cinco primeiros semestres do curso andam na contramão, pois pouco mais de 3% desse grupo obtém sucesso no curso.

c) Quem reprova em disciplinas consideradas chave do curso, como lógica e algoritmos desiste? Nem todos os alunos que reprovam em lógica ou algoritmos desiste do curso, porém os experimentos mostraram que quando essas disciplinas são cursadas juntas, reprovar em lógica pesa muito mais do que reprovar em algoritmos na decisão de seguir ou abandonar o curso.

d) Fazer algoritmos e lógica no mesmo semestre ajuda na permanência do aluno no curso? Infelizmente cursar lógica e algoritmos no mesmo semestre não é a solução para o problema da evasão do curso, como podemos ver, apenas 30% dos alunos que fazem essas disciplinas no mesmo semestre chegam ao final do curso.

e) Reprovar em disciplinas de matemática nos primeiros semestres faz com que o aluno desista do curso? Sim, reprovar em disciplinas do primeiro semestre, tanto de matemática como nas demais se mostrou altamente perigoso para a permanência do aluno no curso, a grande maioria dos alunos que reprovam nessas disciplinas evadiram.

f) A quantidade de disciplinas do primeiro semestre influencia na permanência ou desistência do aluno? A quantidade de disciplinas cursadas no primeiro semestre se mostrou muito influente na decisão dos alunos em abandonar ou não o curso. No caso dos alunos que cursaram 6 disciplinas no primeiro período do curso, houve 100% de abandono. No caso dos alunos que cursaram cinco disciplinas, cerca de 42% deles obtiveram sucesso, dos alunos que cursaram quatro disciplinas, cerca de 28% se formaram. No grupo dos alunos que cursaram três disciplinas, pouco mais de 15% conseguiu se formar, dos alunos que fizeram duas disciplinas apenas 10% e por fim, apenas 8% dos alunos que fizeram uma disciplina no primeiro semestre conseguiu chegar ao final do curso. Nota-se claramente que o percentual de sucesso do aluno no curso está diretamente ligado ao número de disciplinas cursadas no primeiro semestre, quanto maior o número de disciplinas, maior é o percentual dos alunos que se formaram, exceto nos casos onde esse número ultrapassa cinco disciplinas.

Com base nas respostas encontradas nos experimentos realizados, foram identificados alguns perfis de alunos evasores do curso de Ciência da Computação da UNISC:

1. O aluno que sempre faz menos de três disciplinas.
2. O aluno que faz mais que cinco disciplinas no primeiro semestre.
3. O aluno que reprova nas primeiras disciplinas do curso.
4. O aluno que não segue um padrão na quantidade de disciplinas cursadas a cada semestre, como por exemplo, alunos que fazem em um semestre cinco disciplinas, no semestre seguinte três disciplinas, no seguinte duas disciplinas.

8 CONCLUSÕES

A partir dos estudos realizados, foi possível identificar a importância da pesquisa de medidas que possam contribuir no combate à evasão universitária, devido à grande variedade de fatores que influenciam este problema e o impacto que pode causar na sociedade. Também foi possível notar a importância das etapas do KDD e da mineração de dados neste contexto, pois permitem a realização de estudos detalhados do padrão de alunos com perfil de evasão, além de permitir a avaliação de medidas preventivas para tal problema.

A aplicação das técnicas de Mineração de Dados com o objetivo de descobrir novos conhecimentos auxilia no processo de exploração de uma base de dados, permitindo gerar informações úteis para os gestores e assim auxiliar as tomadas de decisões. Desta forma, este trabalho apresentou uma proposta de estudo de padrões da evasão universitária, utilizando uma abordagem com mineração de dados, tendo como público alvo os alunos do curso de Ciência da Computação da UNISC.

Analisou-se um domínio de dados disponibilizados pela universidade contendo informações sobre os alunos e as disciplinas por eles cursadas, além das movimentações feitas pelos alunos na universidade, como matrícula, trancamentos, trocas de currículos ou conclusão do curso.

Neste domínio de dados, foram identificadas algumas tarefas de *data mining* que poderiam ser aplicadas, a grande maioria delas utilizando-se das técnicas de Associação e Classificação, em conjunto com os algoritmos *Apriori*, *FpGrowth e* J48, utilizando a ferramenta de mineração de dados WEKA para a resolução das tarefas.

Algumas decisões e situações prejudicaram o andamento do trabalho, causaram muito retrabalho e perda de tempo. Outras serviram apenas para constatar a relação entre as necessidades da teoria e as dificuldades da prática, como por exemplo, a etapa de importação dos arquivos texto para um banco de dados, que foi muito onerosa e tomou mais tempo do que o esperado. Recomenda-se que o pesquisador faça uma análise prévia do universo de dados a ser trabalhado, antes da elaboração de um cronograma ou planejamento de trabalho.

O entendimento de algumas informações sobre os atributos também foi prejudicado porque as tabelas originais passaram por processo de fusão com outras tabelas, gerando muitos dados duplicados. O ideal seria manter contato constante

com o administrador da base de dados fornecida, visando conhecer melhor a base de dados a ser trabalhada, uma vez que a escolha equivocada de um atributo ou mesmo de uma linha de pensamento, ocasionaram a elaboração equivocada de visões e de novas bases de dados, causando muito retrabalho.

Para encontrar um modelo de tarefa que forneça resultados interessantes, devemos executar repetidas vezes os algoritmos de mineração de dados combinando diversos atributos da tabela até que se obtenha resultados satisfatórios, requerendo muita paciência do minerador de dados.

Entretanto, também é importante mencionar que o conhecimento extraído neste trabalho reflete a realidade do curso de ciência da computação da UNISC, onde o total de disciplinas cursadas e o status final das disciplinas do primeiro semestre são os fatores que mais colaboram para a evasão do curso, com os quais pode-se traçar um perfil para os alunos que evadem, como o perfil dos alunos que sempre fazem menos de três disciplinas, o perfil dos alunos que fazem mais que cinco disciplinas no primeiro semestre, os alunos que reprovam nas primeiras disciplinas do curso ou mesmo o perfil dos alunos que não seguem um padrão na quantidade de disciplinas cursadas a cada semestre.

8.1 Trabalho futuros

Com base no estudo abordado neste trabalho, estabelecem-se algumas recomendações para pesquisas nesta mesma área. Alguns assuntos merecem aprofundamento em pesquisas ou trabalhos futuros. Destes, os principais são:

a) Utilização das matrículas dos alunos que se encontram no ANEXO D, que não foram utilizados neste estudo, por se tratar de matrículas de alunos que estavam ativos no curso durante a realização deste trabalho, para previsão/validação do conhecimento que foi gerado neste estudo;
b) Utilização de outras técnicas para Mineração de Dados não utilizadas neste trabalho, como por exemplo, Clusterização e Redes Neurais;
c) Criação de uma base para a armazenagem dos dados nos moldes que os algoritmos de mineração exigem, possibilitando a geração dos arquivos no formato apropriado para a Mineração de Dados e a

visualização dos resultados, acoplado ao ambiente de gestão da universidade;

d) Incorporar ao modelo um questionário socioeconômico, na tentativa de Identificar precocemente os alunos com perfil propenso à evasão, possibilitando que a instituição de ensino utilize dados em tempo real e não apenas dados estatísticos na análise deste problema. Em vários casos do experimento, alunos que evadiram no primeiro semestre apresentavam desempenho acadêmico excelente, sem reprovar em nenhuma das disciplinas cursadas, mas mesmo assim acabaram não se matriculando para o segundo semestre. Acredita-se que nesses casos, a evasão esteja mais diretamente vinculada a fatores de ordem socioeconômica do que da vida acadêmica do aluno.
e) Criação de uma ferramenta de fácil utilização por parte dos coordenadores do curso, para que os próprios possam fazer a mineração dos dados gerados pelos alunos no decorrer do curso, permitindo uma gestão adequada do mesmo.

REFERENCIAS

ADACHI, Ana Amélia Chaves Teixeira. **Evasão e evadidos nos cursos de graduação da Universidade Federal de Minas Gerais.** [Dissertação de Mestrado]. Faculdade de Educação, UFMG, 2009.

ANDRIOLA, W. B.; ANDRIOLA, C. G.; MOURA, C. P. **Opiniões de docentes e de coordenadores acerca do fenômeno da evasão discente dos cursos de graduação da Universidade Federal do Ceará (UFC).** Rio de Janeiro-RJ: Ensaio: Avaliação e Políticas Públicas em Educação, 2006.

AGRAWAL, R.; MANNILA, H.; SRIKANT, R.; TOIVONEN, H.; VERKAMO, A. I**. Fast Discovery of Association Rules. Advances in knowledge discovery and data mining**, v. 12, n. 1, p. 307-328, 1996.

BASGALUPP, Márcio Porto. **LEGAL-Tree**: um algoritmo genético multi-objetivo para indução de árvores de decisão. [Tese de Doutorado]. São Paulo: Universidade de São Paulo, 2010.

BRAMER, Max. **Principles of data mining**. [s.l]: Springer, 2013.

BREIMAN, l; FRIEDMAN, J; STONE, C. J.; OLSHEN, R. A. **"CART**: Classification and regression trees." Wadsworth: Belmont, CA 156 (1994).

BUSS, Djonatan. **Utilização de técnicas de inteligência de negócios para descoberta em bases de dados acadêmicas**. Monografia (Conclusão de Curso). Bacharelado em Ciências da Computação. Universidade Federal de Pelotas UFPE, 2011.

CAMPELLO, A. V. C.; LINS, L. N. Metodologia de análise e tratamento da evasão e retenção em cursos de graduação instituições federais de ensino superior. **Anais**. XXVIII Encontro Nacional de Engenharia de Produção, Rio de Janeiro, 2008.

CASTANHEIRA, Luciana Gomes. **Aplicação de Técnicas de Mineração de Dados em Problemas de Classificação de Padrões**. Belo Horizonte: UFMG, 2008.

COSTA, E.; BAKER, R. S. J. D.; AMORIN, L.; MAGALHÃES, J.; MARINHO T. **Mineração de Dados Educacionais:** Conceitos, Técnicas, Ferramentas e Aplicações. Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 1-29, 2013.

DANKEL, D. The ID3 Algorithm. Disponível em: http://www.cise.ufl.edu/ ~ddd/cap6635/Fall-97/Short-papers/2.htm, 1997.

DEKKER, G. W.; PECHENIZKIY, M.; VLEESHOUWERS, J. M. **Predicting Students Drop Out**: A Case Study. Cordoba-Espanha: Proceedings of the International Conference on Educational Data Mining, v. 9, p. 41-50, 2009.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From data mining to knowledge discovery in databases.** AI magazine, v. 17, n. 3, p. 37, 1996.

GARCIA, Alan Cássio. **Mineração de dados aplicada a sistemas de recomendação**. [Trabalho de Conclusão de Curso]. Santa Cruz do Sul: Universidade de Santa Cruz do Sul, 2012.

GIL, Antônio Carlos. Como classificar as pesquisas. _____. Como elaborar projetos de pesquisa, v. 4, p. 41-56, 2002.

HALL, M. A.; WITTEN, I. H.; FRANK, E. **The WEKA Data Mining Software**: An Update**.** SIGKDD Explorations, 2009.

INEP - Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Censo da Educação Superior. Brasília: INEP, 2014. Disponível em: <http://portal.inep.gov.br/basica-levantamentos-acessar> Acesso em: 12 mar. de 2014.

JOHNSTON, Véronique. **Why do first year students fail to progress to their second year?** An academic staff perspective**.** Department of Mathematics, Napier University. British Educational Research Association Annual Conference, Nova Iorque, 1997.

LOBO, Maria Beatriz de Carvalho Melo. Esclarecimentos metodológicos sobre os cálculos de evasão. Instituto Lobo para o Desenvolvimento da Educação, da Ciência e da Tecnologia. Mogi das Cruzes, SP: 2011. Disponível em: <http://www.institutolobo.org.br/imagens/pdf/artigos/art_087.pdf> Acesso em: 12 mar. de 2014.

MANHÃES, L. M. B.; CRUZ, S. M. S.; COSTA, R. J. M.; ZAVALETA, J.; ZIMBRÃO, G. **Previsão de Estudantes com Risco de Evasão Utilizando Técnicas de Mineração de Dados**. Instituto de Ciências Exatas – Universidade Federal Rural do Rio de Janeiro (UFRRJ), 2011.

MEC - Ministério da Educação e Cultura. INEP. Resumo Técnico do Censo da Educação Superior 2009, 2009. Disponível em <http://portal.inep.gov.br/web/censo-da-educacao-superior/resumos-tecnicos>. Acesso em: 10 de nov. de 2014.

MINAEI-BIDGOLI, Bhrouz. **Association analysis for a web-based educational system**. Data Mining in E-Learning. WitPress. Southampton, Boston, 2006.

QUINLAN, J. Ross. **Induction of decision trees**. Machine learning, v. 1, n. 1, p. 81-106, 1986.

QUINLAN, J. Ross. **C4.5**: programs for machine learning. Vol. 1. Morgan kaufmann, 1993.

RAPIDMINER. **Manual RAPID-I**. 2014. Disponível em <http://rapidi.com/content/view/181/190/lang,en/>. Acesso em: 10 de nov. de 2014.

RIGO, S. J.; CAZELLA, S. C.; CAMBRUZZI, W. Minerando Dados Educacionais com foco na evasão escolar: oportunidades, desafios e necessidades. In: **Anais do Workshop de Desafios da Computação Aplicada à Educação**. 2012.

REZENDE, Solange Oliveira. **Sistemas inteligentes:** fundamentos e aplicações. Barueri-SP: Editora Manole Ltda, 2003.

TAN, P.; STEINBACH, M.; KUMAR, V.; FERNANDES, A. **Introdução ao datamining: mineração de dados**. Rio de Janeiro: Editora Ciência Moderna, 2009.

TINTO, V. **Dropout from higher education:** a theoretical synthesis of recent research. Washington, Review of Educational Research, 1975.

TINTO, V. **Leaving college:** rethinking the causes of student attrition. Chicago: University of Chicago Press, 1987.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining:** Practical machine learning tools and techniques. 3. ed.. Morgan Kaufmann, 2011.

ANEXO A: Arquivos

Alunos: Dados pessoais dos alunos, como número de matrícula, nome e outros.

```
__ Mon Sep 29 11:30:15 2014 ____________________________________
 1> select
          a.*
    from
          alunos a,
          alunos_cursos ac,
          cursos c
    where
          a.matr_aluno = ac.matr_aluno
    and   ac.cod_curso = c.cod_curso
    and   c.cod_curso_mestre = 4
```

matr_aluno	matr_ant	nome_aluno	dt_nascimento	cod_cidade_na	nacionalidade	nome_pai
569	85100715	JOAO FERNANDO VIGHI	02/05/1952	421	BRASILEIRA	ANTONIO VIGHI
724	85102299	LUIZ FERNANDO SCHERER	16/09/1966	507	BRASILEIRA	ANIBIO SCHERER
724	85102299	LUIZ FERNANDO SCHERER	16/09/1966	507	BRASILEIRA	ANIBIO SCHERER
1240	87100390	IVAN LAWISCH	23/09/1965	507	BRASILEIRA	DILLO ALFONSO LAWISCH
1737	87105381	PAULO CESAR BRAZEIRO DE CARVALHO	11/11/1964	442	BRASILEIRA	FRANCISCO V. MARTINS CARV
3326	90100379	IVAIR OGLIARI	23/10/1971	312	BRASILEIRA	IVO OGLIARI
3572	90102888	FERNANDO LUIS CAUDURO	26/08/1965	200	BRASILEIRA	THIOPHILO EMILIO CAUDURO
4137	90108794	EDISON WERLANG DE OLIVEIRA	27/03/1970	442	BRASILEIRA	ANTONIO C.DE OLIVEIRA SOB
4183	91100162	JONE DAMASCENO SILVA	30/01/1967	369	BRASILEIRA	JOAO DAMASCENO SILVA
4250	91101772	LEANDRO PITSCH	29/03/1968	656	BRASILEIRA	ERNY PITSCH
4251	91101798	MARCOS ALDECIR WENZEL	19/01/1970	507	BRASILEIRA	ARMIN AFONSO WENZEL
4528	91108249	HARDY KOHL JUNIOR	04/06/1970	507	BRASILEIRA	HARDY KOHL
4936	91119717	NELSON EGON GELLER	24/08/1965	656	BRASILEIRA	FEODOR NELSON GELLER
5514	92106689	INES TERESINHA BORGES	03/10/1970	507	BRASILEIRA	GERALDO BORGES
6210	93104345	MAIKEL LUIS KOLLING	17/01/1976	507	BRASILEIRA	DANILO JOSE KOLLING
6343	93107322	ADRIANA SIMONE SCHWENGBER	08/12/1974	507	BRASILEIRA	MARINO JOSE SCHWENGBER
6343	93107322	ADRIANA SIMONE SCHWENGBER	08/12/1974	507	BRASILEIRA	MARINO JOSE SCHWENGBER
6740	93200051	FABIO ANTONIO RASCHE	24/04/1974	156	BRASILEIRA	IVO ANTONIO RASCHE
6740	93200051	FABIO ANTONIO RASCHE	24/04/1974	156	BRASILEIRA	IVO ANTONIO RASCHE

```
Normal text file   length : 3663904   lines : 2914   Ln : 11   Col : 36   Sel : 0   UNIX   ANSI   INS
```

Alunos_cursos: Tabela de vínculo dos alunos com o curso.

```
__ Mon Sep 29 10:10:38 2014 ____________________________________
 1> select a.*
    from alunos_cursos a, cursos c
    where a.cod_curso = c.cod_curso
    and c.cod_curso_mestre = 4
```

matr_aluno	cod_curso	cod_cu	tipo_o	subtip	cod_justif_ev	ano_oc	period	dt_ocorrencia	ano_fi	period	dt_fim_ocorrencia
6740	186	1	2	2	0	2001	1	02/02/2001	0	0	
6741	186	1	2	9	2	1998	1	18/03/1998			
6744	186	1	2	7	0	2002	2	16/12/2002			
6746	186	1	1	1	0	1993	2	16/07/1993			
6748	186	1	1	1	0	1993	2	16/07/1993			
6750	186	1	2	2	11	2008	2	24/07/2008			
6751	186	1	2	7	0	2000	2	20/12/2000			
6756	186	1	1	4	0	2000	1	02/02/2000	0	0	
6757	186	1	2	7	0	1998	1	15/07/1998			
6761	186	1	2	7	0	2000	2	20/12/2000			
6762	186	1	1	1	0	1993	2	16/07/1993			
6763	186	1	2	8	0	1995	2		1900	0	
6764	186	1	2	7	0	1999	1	12/07/1999			
6772	186	1	2	2	0	1997	2	05/08/1997	0	0	
6775	186	1	1	1	0	1993	2	16/07/1993			
6778	186	1	2	7	0	1998	1	15/07/1998			
6781	186	1	2	9	0	1993	2				
6782	186	1	2	7	0	2001	1	17/12/2001			
6783	186	1	1	1	0	1993	2	16/07/1993			
6784	186	1	1	1	0	1993	2	16/07/1993			
6785	186	1	2	7	0	1998	1	15/07/1998			
6786	186	1	2	7	0	2001	2	17/12/2001			

Discipinas_cursadas: Tabela das disciplinas cursadas pelos alunos.

```
___ Mon Sep 29 11:33:37 2014 ____________________________________________
 1> select
           dc.*
    from
           alunos_cursos ac,
           cursos c,
           disciplinas_cursadas dc
    where
           ac.cod_curso = c.cod_curso
    and    c.cod_curso_mestre = 4
    and    ac.matr_aluno = dc.matr_aluno
```

matr_aluno	ano_ma	period	cod_disciplin	cod_curso_tur	cod_turma_log	nota1	nota2	nota_final	nota_exame	media_final	carga_horari
569	1985	1	1096	0	0	0.000	0.000	7.750		7.750	64.0
569	1985	1	2271	0	0	0.000	0.000	9.000		9.000	60.0
569	1985	2	1097	0	0	0.000	0.000	7.000		7.000	62.0
569	1985	2	1363	0	0	0.000	0.000	8.500		8.500	45.0
569	1985	2	2274	0	0	0.000	0.000	8.250		8.250	60.0
569	1986	1	1049	0	0	0.000	0.000	8.750		8.750	60.0
569	1986	1	1206	0	0	0.000	0.000	9.000		9.000	62.0
569	1986	1	1996	0	0	0.000	0.000	7.000		7.000	62.0
569	1986	2	1065	0	0	0.000	0.000	7.500		7.500	60.0
569	1986	2	1115	0	0	0.000	0.000	8.750		8.750	64.0
569	1986	2	3102	0	0	0.000	0.000	8.250		8.250	62.0
569	1987	1	1066	0	0	0.000	0.000	7.500		7.500	60.0
569	1987	1	1098	0	0	0.000	0.000	8.750		8.750	60.0
569	1987	1	1365	0	0	0.000	0.000	8.000		8.000	32.0
569	1987	1	2162	0	0	0.000	0.000	7.000		7.000	60.0
569	1987	1	2276	0	0	0.000	0.000	10.000		10.000	64.0

Itens_concluidos: Itens concluídos pelos alunos durante o curso.

```
___ Mon Sep 29 11:39:47 2014 ____________________________________________
  1> select
           ic.*
     from
           alunos_cursos ac,
           cursos c,
           itens_concluidos ic
     where
           ac.cod_curso = c.cod_curso
     and    c.cod_curso_mestre = 4
     and    ac.matr_aluno = ic.matr_aluno
```

matr_aluno	cod_curso	cod_cu	seq_it	tipo_c	creditos
569	102	15	1	1	4.0000
569	102	15	2	1	4.0000
569	102	15	3	1	4.0000
569	102	15	4	1	4.0000
569	102	15	5	1	4.0000
569	102	15	6	1	4.0000
569	102	15	7	1	4.0000
569	102	15	8	1	4.0000
569	102	15	9	1	4.0000
569	102	15	10	1	4.0000
569	102	15	11	1	9.0000
569	102	15	12	1	9.0000
569	102	15	13	1	2.0000
569	102	15	14	1	4.0000

Arquivo: Ocorrencias_alunos: Tabela com todas as ocorrências dos alunos no curso, desde a matrícula, trancamentos, cancelamentos ou formatura.

```
__ Mon Sep 29 11:37:52 2014 ____________________________________
 1> select
            oa.*
    from
            alunos_cursos ac,
            cursos c,
            ocorrencias_alunos oa
    where
            ac.cod_curso = c.cod_curso
    and     c.cod_curso_mestre = 4
    and     ac.matr_aluno = oa.matr_aluno
```

matr_aluno	sequencia	cod_curso	tipo_o	subtip	cod_ju	ano_oc	period	dt_ocorrencia	ano_fi	period	dt_fim_ocorrencia	ano_ve	period	med
569	1	102	1	1	0	1985	1	19/01/1985				1985	1	
569	2	102	2	7	0	1990	1							
569	3	207	1	5	0	1998	2	04/08/1998						
724	1	102	1	1	0	1985	1	19/01/1985				1985	1	
724	2	102	2	2	0	2000	2	03/08/2000						
724	3	186	1	2	0	2000	2	03/08/2000				1985	1	
724	4	186	2	8	4	2001	2	10/10/2001	2002	1	15/01/2002			
724	5	186	1	4	0	2002	2	23/07/2002				1985	1	
724	6	186	2	2	0	2002	2	26/07/2002						
724	7	207	1	2	0	2002	2	26/07/2002				1985	1	
724	8	207	2	10	0	2003	1	14/05/2003						
724	1	102	1	1	0	1985	1	19/01/1985				1985	1	
724	2	102	2	2	0	2000	2	03/08/2000						
724	3	186	1	2	0	2000	2	03/08/2000				1985	1	
724	4	186	2	8	4	2001	2	10/10/2001	2002	1	15/01/2002			
724	5	186	1	4	0	2002	2	23/07/2002				1985	1	
724	6	186	2	2	0	2002	2	26/07/2002						
724	7	207	1	2	0	2002	2	26/07/2002				1985	1	
724	8	207	2	10	0	2003	1	14/05/2003						

Fonte: Setor de informática da UNISC

ANEXO B: Quadros

ALUNOS					
Coluna	**Tipo Dado**	**Null**	**PK**	**Descrição**	**Domínio/Nota**
matr_aluno	Integer	Não	Sim	Matrícula do Aluno	
matr_antiga_aluno	char(8)	Não	Não	Matrícula do aluno no sistema antigo	
nome_aluno	varchar(50)	Não	Não	Nome do aluno	
dt_nascimento	date	Não	Não	Data de nascimento	
cod_cidade_naturalidade	integer	Não	Não	Código do local de nascimento	
Nacionalidade	char(20)	Não	Não	Nacionalidade	
nome_pai	varchar(50)	Não	Não	Nome do pai	
nome_mae	varchar(50)	Não	Não	Nome da mãe	
Sexo	char(1)	Não	Não	Sexo	M, F
estado_civil	integer	Não	Não	Estado Civil	
cod_logradouro_cobranca	integer1	Não	Não	Código do logradouro do endereço de cobrança	
endereco_cobranca	varchar(50)	Não	Não	Endereço para cobrança	
bairro_cobranca	varchar(20)	Não	Não	Bairro do endereço de cobrança	
cod_cidade_cobranca	integer	Não	Não	Código da cidade do endereço de cobrança	
cep_cobranca	char(8)	Não	Não	CEP referente ao endereço de cobrança	
fone_cobranca	char(20)	Não	Não	Fone para o contato em caso de cobrança	
outro_endereco	varchar(50)	Não	Não	Outro endereço do aluno	
outro_bairro	varchar(20)	Não	Não	Bairro correspondente ao outro endereço	
outro_cod_cidade	integer	Não	Não	Código da cidade do outro endereço	
outro_cep	char(8)	Não	Não	CEP referente ao outro endereço	
outro_fone	char(20)	Não	Não	Fone para contato	
cod_profissao	integer	Não	Não	Código da profissão ou atividade que o aluno exerce	
endereco_profissao	varchar(50)	Não	Não	Endereço profissional	
bairro_profissao	varchar(20)	Não	Não	Bairro referente ao endereço profissional	
cod_cidade_profissao	integer	Não	Não	Código da cidade do endereço profissional	
cep_profissao	char(8)	Não	Não	CEP referente ao endereço profissional	
fone_profissao	char(20)	Não	Não	Fone profissional para contato	
cic_numero	char(11)	Não	Não	Número do CPF	
cic_valido	integer1	Não	Não		
ci_numero	char(15)	Não	Não	Número da carteira de identidade	
ci_orgao	char(3)	Sim	Não	Órgão expedidor da carteira de identidade	
ci_uf	char(2)	Sim	Não	UF onde foi expedida a carteira de identidade	
ce_numero	char(10)	Sim	Não	Número da carteira de identidade para estrangeiros|	

ce_orgao	char(3)	Sim	Não	Órgão Expedidor da carteira de identidade para estrangeiros	
ce_pais	char(10)	Sim	Não	País do órgão expedidor da carteira	
sm_numero	char(12)	Sim	Não	Número de registro da carteira de serviço militar	
sm_tipo_doc	varchar(30)	Sim	Não	Natureza ou espécie de documento	
sm_serie	char(1)	Sim	Não	Série da carteira de serviço militar	
sm_ministerio	varchar(20)	Sim	Não	Ministério referente ao cumprimento do Serviço Militar	
sm_ano_expedicao	integer	Sim	Não	Ano da expedição da Carteira do Serviço Militar	
sm_quitacao	char(1)	Sim	Não	Situação do aluno quanto a quitação do serviço militar	S, N
te_numero	char(15)	Sim	Não	Número do título eleitoral	
te_zona	integer	Sim	Não	Zona referente ao Título Eleitoral	
te_secao	integer	Sim	Não	Seção referente ao Título Eleitoral	
te_local	integer	Não	Não	Código da cidade referente ao Título Eleitoral	
te_dt_votacao	date	Sim	Não	Data referente à última votação	
aluno_especial	char(1)	Não	Não	Condição do aluno quanto aos cursos	S, N
autorizacao_matr	char(1)	Não	Não	Indicador para a autorização da matrícula	S,N
cod_justif_matricula	integer	Não	Não	Código da justificativa de autorização de matrícula	
status_aluno	char(1)	Não	Não	Situação do aluno na instituição	A, I
sm_dt_expedicao	date	Sim	Não	Data de expedição da carteira de serviço militar	

ALUNOS_CURSOS					
Coluna	**Tipo Dado**	**Null**	**PK**	**Descrição**	**Domínio/Nota**
matr_aluno	Integer	Não	Sim	Matrícula do Aluno	
cod_curso	Integer	Não	Sim	Código do curso do aluno	
cod_curriculo	Smallint	Não	Não	Código do currículo	
tipo_ocorrencia	integer1	Não	Não	Tipo de movimento do aluno	1-Ingresso, 2-Evasão
subtipo_ocorrencia	integer1	Não	Não	Subtipo de ocorrência para o aluno	1-Vestibular, 2-Transferência Interna, 3-Transferência Externa, 4-Reingresso,5-Diplomado, 6-Aluno Especial,7-Conclusão, 8-Trancamento,9-Cancelamento,10-Desistência, 11-Suspensão,12-Desligamento,13-Falecimento,14-Cancelamento de Matrícula,15-Cancelamento

					de Rematrícula, 16-Ingresso em Curso Sequencial, 17-Permuta, 18-Trancamento para Vínculo,19-Reopção de Habilitação
cod_justif_evasao	Integer	Não	Não	Código da justificativa de evasão	
ano_ocorrencia	Smallint	Não	Não	Ano em que o aluno ingressou ou evadiu do curso	
periodo_ocorrencia	integer1	Não	Não	Período em que o aluno ingressou ou evadiu do curso	
dt_ocorrencia	Date	Não	Não	Data em que o aluno ingressou ou evadiu do curso	
ano_fim_ocorrencia	Smallint	Sim	Não	Ano limite referente à possibilidade de retorno do aluno	
periodo_fim_ocorrencia	integer1	Sim	Não	Período limite referente à possibilidade de retorno do aluno	
dt_fim_ocorrencia	Date	Sim	Não	Data limite referente à possibilidade de retorno do aluno	
ano_prev_conclusao	Smallint	Não	Não	Ano previsto para a conclusão do curso	
periodo_prev_conclusao	integer1	Não	Não	Período previsto para a conclusão do curso	
dt_colacao_grau	Date	Não	Não	Data referente à colação de grau	
dt_expedicao_diploma	Date	Não	Não	Data de expedição do doploma	
autorizacao_matr	char(1)	Não	Não	Indicador de autorização de matrícula pelo responsável	S-Sim, N-Não
autorizacao_requisitos	char(1)	Não	Não	Indicador para a autorização da matrícula nos itens curriculares que tem Vide-Coordenação como requisito	
prev_formando	char(1)	Não	Não	Previsão de o aluno ser um provável formando	
forma_cobranca	integer1	Não	Não	Indica a forma de cobrança default para o aluno	1-DOC, 2-Débito em folha, 3-Pagamento de monitoria, 4-Débito em conta, 5-Não emissão DOC

pagto_entrada	char(1)	Não	Não	Indica o pagamento referente à entrada	S-Sim, N-Não
cod_justif_pagto	Integer	Não	Não	Código da justificativa de dispensa do pagamento de entrada	
sequencia_atual	Integer	Não	Não	Número sequencial atual de identificação da ocorrência	
dt_provao	Date	Não	Não	Data de realização do Exame Nacional de Curso pelo aluno	
cod_status_provao	integer4	Sim	Não		

OCORRENCIAS_ALUNOS					
Coluna	**Tipo Dado**	**Null**	**PK**	**Descrição**	**Domínio/Nota**
matr_aluno	Integer	Não	Sim	Matrícula do Aluno	
Sequencia	Integer	Não	Sim	Número sequencial de identificação da ocorrência	
cod_curso	Integer	Sim	Não	Código do curso	
tipo_ocorrencia	Integer	Não	Não	Tipo de ocorrência	1-Ingresso, 2-Evasão
subtipo_ocorrencia	Integer	Não	Não	Subtipo de ocorrência	1-Vestibular, 2-Transferência Interna, 3-Transferência Externa, 4-Reingresso,5-Diplomado, 6-Aluno Especial,7-Conclusão, 8-Trancamento,9-Cancelamento,10-Desistência, 11-Suspensão,12-Desligamento,13-Falecimento,14-Cancelamento de Matrícula,15-Cancelamento de Rematrícula, 16-Ingresso em Curso Sequencial, 17-Permuta, 18-Trancamento para Vínculo,19-Reopção de Habilitação
cod_justif_evasao	Integer	Não	Não	Código da justificativa de evasão	
ano_ocorrencia	SmallInt	Não	Não	Ano em que o aluno ingressou ou evadiu do curso	
periodo_ocorrencia	Integer	Não	Não	Período em que o aluno ingressou ou evadiu do curso	
dt_ocorrencia	date	Não	Não	Data em que o aluno ingressou ou evadiu do curso	
ano_fim_ocorrencia	smallint	Sim	Não	Ano limite referente à possibilidade de retorno do aluno	
periodo_fim_ocorrencia	Integer	Sim	Não	Período limite referente à possibilidade de retorno do aluno	
ano_vest	smallint	Sim	Não	Ano em que o aluno prestou vestibular	
periodo_vest	Integer	Sim	Não	Período em que o aluno prestou vestibular	
media_geral	float	Sim	Não	Média geral obtida pelo aluno	

				no vestibular	
classif_curso	smallint	Sim	Não	Classificação obtida no curso pelo aluno no vestibular	
classif_geral	smallint	Sim	Não	Classificação geral obtida pelo aluno no vestibular	
cod_instituicao_2g	Integer	Sim	Não	Código da instituição de ensino do segundo grau	
ano_evasao_2g	smallint	Sim	Não	Ano da evasão do curso de segundo grau	
cod_instituicao_3g	Integer	Sim	Não	Código da instituição de ensino superior de origem	
ano_evasao_3g	smallint	Sim	Não	Ano de evasão do aluno em relacão ao curso de origem	
periodo_evasao_3g	Integer	Sim	Não	Período de evasão do aluno em relação ao curso de origem	
cod_curso_2g	Integer	Sim	Não	Código do curso de segundo grau concluído pelo aluno	
cod_curso_3g	Integer	Sim	Não	Código do curso superior concluído pelo aluno	
media_harmonica	float	Sim	Não	Média hasrmônica obtida pelo aluno no vestibular	
media_ponderada	float	Sim	Não	Média ponderada obtida pelo aluno no vestibular	
periodo_ideal_ingresso	Integer	Sim	Não	Semestre de ingresso	
ano_base_ingresso	smallint	Sim	Não	Ano base para cálculo do semestre atual	
periodo_base_ingresso	Integer	Sim	Não	Período base para cálculo do semestre atual	

DISCIPLINAS_CURSADAS					
Coluna	**Tipo Dado**	**Null**	**PK**	**Descrição**	**Domínio/Nota**
matr_aluno	Integer	Não	Sim	Matrícula do Aluno	
ano_matr	smallint	Não	Sim	Ano em que foi cursada a disciplina	
periodo_matr	integer1	Não	Sim	Período em que foi cursada a disciplina	
cod_disciplina	integer	Não	Sim	Código da disciplina cursada	
cod_curso_turma	integer	Não	Não	Código do curso da turma na qual o aluno cursou a disciplina	
cod_turma_logica	integer	Não	Não	Código referente à turma lógica na qual o aluno cursou a disciplina	

nota1	float4	Não	Não	Primeira nota parcial	
nota2	float4	Não	Não	Segunda nota parcial	
nota_final	float4	Não	Não	Nota final resultante de uma média entre as notas parciais	
nota_exame	float4	Não	Não	Nota do exame final	
media_final	float4	Não	Não	Média final da disciplina	
carga_horaria	decimal(13,4)	Não	Não	Carga horária lecionada para a disciplina (aulas dadas)	
status_disciplina	integer1	Não	Não	Situaçáo do aluno em relação à disciplina	1-Aprovado, 2-Reprovado,3-Desistente
proporcao_presenca	char(10)	Não	Não	Proporção da presença x hora aula para o controle de frequência	
faltas_aluno	smallint	Não	Não	Número de faltas do aluno	
freq_aluno	float4	Não	Não	Frequência do aluno às aulas	

ITENS_CONCLUIDOS					
Coluna	**Tipo Dado**	**Null**	**PK**	**Descrição**	**Domínio/Nota**
matr_aluno	Integer	Não	Sim	Matrícula do Aluno	
cod_curso	integer	Não	Sim	Código do curso	
cod_curriculo	smallint	Não	Sim	Código do currículo	
seq_item	integer1	Não	Sim	Número sequencial referente ao item do currículo	
tipo_conclusao	integer1	Não	Não	Tipo de conclusão da disciplina	1-Cursado, 2-Aproveitado,3-Dispensado
Créditos	decimal(13,4)	Não	Não	Número de créditos concluídos	

Fonte: Departamento de TI da UNISC

ANEXO C: Consultas

Consulta 01: Disciplinas cursadas

```
SELECT distinct a.matr_aluno, A.cod_disciplin, A.ano_ma, A.period, A.status, A.media_final
FROM r_disciplinas_cursadas A
where  A.cod_curso_tur IN (186,207,2509)
ORDER BY a.matr_aluno, A.ano_ma, A.period, A.cod_disciplin
```

Consulta 02: Total disciplinas cursadas por ano/semestre por aluno

```
select count (*), d.ano_ma, d.period
from DISCIPLINAS_CURSADAS d
where d.matr_aluno = :MATR
group by d.ano_ma, d.period
order by d.ano_ma, d.period
```

Consulta 03: Ocorrência alunos

```
select a.matr_aluno,a.cod_curso, a.tipo_o, a.subtip
from R_OCORRENCIAS_ALUNOS a , historico H
where a.matr_aluno = :MATR AND A.matr_aluno = H.matricula
AND A.cod_curso IN (186,207,2509)
order by a.sequencia
```

Consulta 4: Experimento 1

```
select h.sexo,
case when h.estado_civil = 1 then 'Solteiro' else 'Demais'  end as estado_civil,
CASE
   WHEN (h.idade_ingresso < 20) THEN '<19'
   WHEN (h.idade_ingresso > 19 AND h.idade_ingresso < 26) THEN '20-25'
   WHEN (h.idade_ingresso > 25 AND h.idade_ingresso < 36) THEN '26-35'
   WHEN (h.idade_ingresso > 35 ) THEN '>36'
   else '?'
END AS faixa_etaria,
h.codigo_curriculo_curso as curriculo,
```

```
CASE
   WHEN (h.meio_acesso = 1) THEN 'Vestibular'
   WHEN (h.meio_acesso = 2) THEN 'Transf_Interna'
   WHEN (h.meio_acesso = 3) THEN 'Transf_Externa'
   WHEN (h.meio_acesso = 5) THEN 'Diplomado'
   WHEN (h.meio_acesso = 26) THEN 'Outro'
   else '?'
END AS meio_acesso,
coalesce (h.trocou_curriculo,'?') as trocou_curriculo,
case
  when h.qtde_trancamentos = 0 then  'Nenhuma'
  when h.qtde_trancamentos = 1 then  'Uma'
  when h.qtde_trancamentos >= 2 then  'Mais_que_uma'
  else '?'
end as qtde_trancamentos,
h.total_disciplinas_cursadas,
case
 when h.tot_disc_sem_01 in (1,2) then '1->2'
 when h.tot_disc_sem_01 in (3,4) then '3->4'
 when h.tot_disc_sem_01 in (5,6) then '5->6'
 when h.tot_disc_sem_01 > 6 then '7->'
 else '0'
end as tot_disc_sem_01,
case
 when h.tot_disc_sem_02 in (1,2) then '1->2'
 when h.tot_disc_sem_02 in (3,4) then '3->4'
 when h.tot_disc_sem_02 in (5,6) then '5->6'
 when h.tot_disc_sem_02 > 6 then '7->'
 else '0'
end as tot_disc_sem_02,
case
 when h.tot_disc_sem_03 in (1,2) then '1->2'
 when h.tot_disc_sem_03 in (3,4) then '3->4'
 when h.tot_disc_sem_03 in (5,6) then '5->6'
```

```
 when h.tot_disc_sem_03 > 6 then '7->'
 else '0'
end as tot_disc_sem_03,
case
 when h.tot_disc_sem_04 in (1,2) then '1->2'
 when h.tot_disc_sem_04 in (3,4) then '3->4'
 when h.tot_disc_sem_04 in (5,6) then '5->6'
 when h.tot_disc_sem_04 > 6 then '7->'
 else '0'
end as tot_disc_sem_04,
case
 when h.tot_disc_sem_05 in (1,2) then '1->2'
 when h.tot_disc_sem_05 in (3,4) then '3->4'
 when h.tot_disc_sem_05 in (5,6) then '5->6'
 when h.tot_disc_sem_05 > 6 then '7->'
 else '0'
end as tot_disc_sem_05,
case
  when h.status_movimentacao = 1 then
     case
      when h.subtipo_movimentacao in (1,2,3,4,5) then 'Cursando'
     end
  when h.status_movimentacao = 2 then
     case
      when h.subtipo_movimentacao in (2,3,8,9,10,13,14,15,18) then 'Evasão'
      When h.subtipo_movimentacao = 7 then 'Formado'
     end
end as Situacao
from historico h
```

Consulta 5: Experimento 2

```
select h.sexo,
case when h.estado_civil = 1 then 'Solteiro' else 'Demais'  end as estado_civil,
CASE
```

```
   WHEN (h.idade_ingresso < 20) THEN '<19'
   WHEN (h.idade_ingresso > 19 AND h.idade_ingresso < 26) THEN '20-25'
   WHEN (h.idade_ingresso > 25 AND h.idade_ingresso < 36) THEN '26-35'
   WHEN (h.idade_ingresso > 35 ) THEN '>36'
   else '?'
END AS faixa_etaria,
h.codigo_curriculo_curso as curriculo,
CASE
   WHEN (h.meio_acesso = 1) THEN 'Vestibular'
   WHEN (h.meio_acesso = 2) THEN 'Transf_Interna'
   WHEN (h.meio_acesso = 3) THEN 'Transf_Externa'
   WHEN (h.meio_acesso = 5) THEN 'Diplomado'
   WHEN (h.meio_acesso = 26) THEN 'Código 26'
   else '?'
END AS meio_acesso,
coalesce (h.trocou_curriculo,'?') as trocou_curriculo,
case
   when h.qtde_trancamentos = 0 then 'Nenhuma'
   when h.qtde_trancamentos = 1 then 'Uma'
   when h.qtde_trancamentos > 1 then 'Mais_que_uma'
   else '?'
end as  qtde_trancamentos ,
case
   when h.tempo_permanencia = 1 then '1'
   when h.tempo_permanencia in (2,3) then '2->3'
   when h.tempo_permanencia in (4,5) then '4->5'
   when h.tempo_permanencia > 5 then '6->'
end as Semestres,
case
  when h.status_movimentacao = 1 then
     case
      when h.subtipo_movimentacao in (1,2,3,4,5) then 'Cursando'
     end
  when h.status_movimentacao = 2 then
```

```
    case
     when h.subtipo_movimentacao in (2,3,8,9,10,13,14,15,18) then 'Evasão'
     When h.subtipo_movimentacao = 7 then 'Formado'
    end
end as Situacao
from historico h
```

Consulta 6: Experimento 3

```
select h.sexo,
case when h.estado_civil = 1 then 'Solteiro' else 'Demais'  end as estado_civil,
CASE
  WHEN (h.idade_ingresso < 20) THEN '<19'
  WHEN (h.idade_ingresso > 19 AND h.idade_ingresso < 26) THEN '20-25'
  WHEN (h.idade_ingresso > 25 AND h.idade_ingresso < 36) THEN '26-35'
  WHEN (h.idade_ingresso > 35 ) THEN '>36'
  else '?'
END AS faixa_etaria,
h.codigo_curriculo_curso as curriculo,
coalesce (h.trocou_curriculo,'?') as trocou_curriculo,
case
  when h.qtde_trancamentos = 0 then 'Nenhuma'
  when h.qtde_trancamentos = 1 then 'Uma'
  when h.qtde_trancamentos > 1 then 'Mais_que_uma'
  else '?'
end as  qtde_trancamentos ,
case
  when h.media_vestib <= 2 then 'Muito_baixa'
  when h.media_vestib > 2 and h.media_vestib <= 3  then 'Baixa'
  when h.media_vestib > 3 and h.media_vestib <= 5  then 'Media'
  when h.media_vestib > 5 then 'Boa'
  else '?'
end as  media_vestibular ,
case
  when h.class_vestib <= 10 then 'Top_10'
  when h.class_vestib > 10  then 'Demais'
```

```
   else '?'
end as  class_vestibular ,
case
  when h.status_movimentacao = 1 then
     case
      when h.subtipo_movimentacao in (1,2,3,4,5) then 'Cursando'
     end
  when h.status_movimentacao = 2 then
     case
      when h.subtipo_movimentacao in (2,3,8,9,10,13,14,15,18) then 'Evasão'
      When h.subtipo_movimentacao = 7 then 'Formado'
     end
end as Situacao
from historico h
```

ANEXO D: Matriculas dos alunos ativos

M-21619	M-51771	M-60046	M-72566	M-80027	M-89867	M-96619
M-24085	M-51778	M-60047	M-72567	M-80874	M-89870	M-96621
M-29019	M-51784	M-60049	M-72571	M-81623	M-89875	M-96622
M-30424	M-51786	M-60072	M-72573	M-81624	M-89878	M-96624
M-30939	M-51791	M-60073	M-72575	M-81632	M-89879	M-96631
M-31020	M-51793	M-60074	M-72578	M-81639	M-89886	M-96637
M-31293	M-51803	M-60075	M-72581	M-81645	M-89887	M-97331
M-31523	M-51808	M-60076	M-72583	M-81651	M-89889	M-97332
M-31604	M-52732	M-60079	M-72589	M-81657	M-89892	M-97334
M-31703	M-53120	M-60081	M-72595	M-82473	M-89895	M-97941
M-31761	M-53372	M-61078	M-72598	M-82474	M-89899	M-98190
M-31853	M-53513	M-61777	M-72600	M-82791	M-90269	M-98673
M-32303	M-53600	M-61930	M-72602	M-83838	M-91215	
M-32331	M-55755	M-62003	M-72619	M-83840	M-91216	
M-32334	M-55819	M-62014	M-72628	M-83849	M-92314	
M-33504	M-55821	M-62322	M-72887	M-83850	M-92892	
M-33573	M-55828	M-62798	M-73941	M-83853	M-92968	
M-34474	M-55835	M-62822	M-74221	M-83856	M-93340	
M-35687	M-55837	M-62824	M-74300	M-83862	M-93465	
M-36579	M-55838	M-62827	M-76310	M-83864	M-93466	
M-36587	M-55840	M-64121	M-77574	M-83869	M-93473	
M-36593	M-55842	M-65605	M-77575	M-85648	M-93476	
M-38253	M-55846	M-65615	M-77581	M-85771	M-93479	
M-39436	M-55847	M-65616	M-77583	M-85786	M-93483	
M-39438	M-55857	M-65619	M-77585	M-85895	M-93484	
M-39733	M-55859	M-65621	M-77588	M-87312	M-93487	
M-40150	M-55860	M-65626	M-77589	M-87315	M-93488	
M-40931	M-55867	M-65633	M-77591	M-87318	M-93492	
M-40939	M-56210	M-65634	M-77602	M-87327	M-93493	
M-41532	M-56298	M-65636	M-77603	M-87336	M-93497	
M-41617	M-56692	M-65642	M-77610	M-87347	M-93517	
M-42103	M-56708	M-66782	M-77611	M-87360	M-94019	
M-42585	M-57329	M-67072	M-77613	M-87362	M-94945	
M-43050	M-57662	M-67454	M-77619	M-87363	M-95929	
M-43953	M-57789	M-68663	M-77654	M-87709	M-96600	
M-45375	M-58184	M-68689	M-77662	M-88275	M-96603	
M-46450	M-58198	M-69689	M-77724	M-88281	M-96605	
M-48008	M-59070	M-69692	M-77783	M-88613	M-96606	
M-48074	M-59658	M-69693	M-78273	M-89857	M-96610	
M-50871	M-59859	M-70326	M-78606	M-89859	M-96612	
M-51765	M-60041	M-70795	M-79196	M-89862	M-96613	
M-51767	M-60041	M-71169	M-79544	M-89863	M-96616	
M-51770	M-60042	M-72565	M-79577	M-89866	M-96618	