UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

MINERAÇÃO DE DADOS NO MOODLE: ANÁLISE DE PRAZOS DE ENTREGA DE ATIVIDADES

DISSERTAÇÃO DE MESTRADO

Fabieli De Conti

Santa Maria, RS, Brasil

2011

MINERAÇÃO DE DADOS NO MOODLE: ANÁLISE DE PRAZOS DE ENTREGA DE ATIVIDADES

por

Fabieli De Conti

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para a obtenção do grau de

Mestre em Computação

Orientador: Profª.Drª. Andrea Schwertner Charão (UFSM)

Santa Maria, RS, Brasil

2011

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

MINERAÇÃO DE DADOS NO MOODLE: ANÁLISE DE PRAZOS DE ENTREGA DE ATIVIDADES

elaborada por
Fabieli De Conti

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

Profª.Drª. Andrea Schwertner Charão (UFSM)
(Presidente/Orientador)

Prof. Dr. Ricardo Vargas Dorneles (UCS)

Profª. Drª. Roseclea Duarte Medina (UFSM)

Santa Maria, 19 de Dezembro de 2011.

AGRADECIMENTOS

Somos a Luz de Deus e só evoluímos quando entregamos a nossa vida em Suas mãos.

Aos meus amados Pais, obrigada por me ensinar a enfrentar a vida e incentivar sempre na busca pelo crescimento pessoal e profissional. Não seria nada sem o amor e sabedoria de vocês. Pai e Mãe amo muito vocês e tenho muito orgulho de ser filha de vocês.

Meus queridos irmãos, a vida nos ensina que mesmo distantes podemos permanecer juntos. Um especial agradecimento aos meus queridos, minha mana Tatiana e ao meu cunhado Vini pelos conselhos, puxões de orelha e elogios no momento certo. E ao meu mano, Lessandro que me fez ter confiança mas minhas capacidades e ao meu irmão Leandro. Sucesso para vocês.

Ao meu namorado Lisandro, obrigada por entender o momento em que necessitava me dedicar ao mestrado e tinha que abdicar um pouco de você. Sempre com suas palavras e carinho conseguia me acalmar e mostrava que com paciência qualquer problema pode ser resolvido. Te Amo muito, essa vitória é nossa.

Minha Orientadora Andrea, você é uma pessoa que admiro muito. Obrigada por ter confiado na minha capacidade e pelos conselhos amigos. Se não fossem estes não teria chegado até aqui, você demonstra o exemplo de professora que quero seguir.

À professora Rose, que acolhe seus alunos com um carinho tão especial. A todos os professores do curso pelos ensinamentos profissionais e de vida. Ter convivido com vocês me deu oportunidade de crescer muito. Obrigada.

Aos meus colegas de mestrado, juntos realizamos trabalhos, escrevemos artigos e vencemos mais esta etapa. Em especial aos colegas Solange, Marcela, Gustavo, Daniel, Jaziel, Renato e Fernando, com quem tive um contato maior durante o curso.

Aos meus colegas de trabalho pelo seu apoio permanente, e sua compreensão dos momentos de ausência em que me dediquei aos estudos. Meus agradecimento aos colegas e amigos, em ordem alfabética: Alecson, Carol, Cleia, Lidiane e Raquel, que assumiram os projetos e o NAPNE, porque sem vocês não conseguiria me dedicar à escrita deste trabalho. Obrigada e contem sempre comigo.

Aos meus ex-colegas de trabalho que deixaram uma marca muito boa na minha vida: Célio, Heleno e Pedro. Muito obrigada por tudo. Aos meus amigos, acabei me distanciando um pouco pois não sobrava muito tempo conciliando trabalho e estudo, mesmo assim sempre contei com a amizade de vocês. Em especial as minhas amigas, Carolina, Eliana, Fabíola e Maira.

Enfiam tem tantas pessoas especiais na minha vida mas não poderia deixar de expressar meu agradecimento todo especial a, Paulinha e família, Didinha e família, Cristina, Ziara e Família essas pessoas são também meu porto seguro.

A todos que de uma maneira outra colaboraram para a conclusão do curso. Meu muito Obrigada.

“Essa audácia de buscar o novo
Sem pisar no rastro e reacender as brasas
É o contraponto de ter prenda e filhos
E ficar tordilho ao redor ’das casa’ ”
— CRISTIANO QUEVEDO

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

MINERAÇÃO DE DADOS NO MOODLE: ANÁLISE DE PRAZOS DE ENTREGA DE ATIVIDADES

Autor: Fabieli De Conti
Orientador: Profª.Drª. Andrea Schwertner Charão (UFSM)
Local e data da defesa: Santa Maria, 19 de Dezembro de 2011.

Como ferramenta pedagógica, os Ambientes Virtuais de Aprendizagem (AVA) tornaram-se prática comum para o ensino à distância como nos cursos presenciais, por dar apoio à comunicação entre os envolvidos com o ensino. Essa dissertação descreve uma pesquisa realizada sobre os dados gerados pela interação com o AVA Moodle de uma instituição de ensino, focando a análise de prazos e de datas efetivas de submissões de tarefas neste ambiente. O objetivo deste trabalho é identificar padrões relevantes sobre a postagem de tarefas no ambiente, para subsidiar ações em auxílio à postagem muito próximo ao final do período de postagem e propor uma forma transparente e automática de integrar ao Moodle as atividades de KDD. A ferramenta de integração proposta aborta os algoritmos de mineração de dados EM e J.48, selecionados no nosso estudo e os resultados são apresentados de forma simplificada aos usuários na própria interface do Moodle. Para o estudo, são considerados o período em que a tarefa permaneceu aberta para postagem, o curso proveniente da tarefa e o período em que a postagem foi realizada. O estudo foi realizado seguindo as etapas do processo de descoberta de conhecimento, com a utilização da ferramenta Weka. No estudo observou-se a incidência do número de postagens mais próximas ao término do tempo de postagem quando o prazo da mesma era superior a 15 dias. Nos cursos de pós-graduação, observa-se que o tempo para postagem é maior que nos cursos de nível superior e que esse nível apresenta maior quantidade de postagem sendo realizadas no final do prazo de postagem. Nesse contexto, é mais viável a realização de atividades com um prazo menor. Além de um maior número de submissões logo na abertura para postagem, o professor consegue *feedback* mais rápido do processo de aprendizagem do aluno. Isso possibilita tomar atitudes corretivas em tempo mais adequado a fim de evitar o insucesso ou desistência do aluno. Com a implementação da integração do KDD ao Moodle é possível a realização de experimentos por usuários de forma automática e simplificada.

Palavras-chave: Ambiente Virtual de Aprendizagem, KDD, Mineração de Dados.

ABSTRACT

Master's Dissertation
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

ANALYSIS OF ASSIGNMENT SUBMISSIONS DEADLINES IN MOODLE: A CASE STUDY USING DATA MINING

Author: Fabieli De Conti
Advisor: Prof[a].Dr[a]. Andrea Schwertner Charão (UFSM)

Virtual Learning Environments became common practice as a course tool for both distance and presence learning courses, as they support the communication among the parties involved. This study describes research carried out on data that were generated by the interaction with the Moodle VLE within an educational institution, with focus on the analysis of due dates and actual submission dates for assignments in the course environment. The objective of this study is to obtain relevant information about how course assignments are posted in the learning environment, to guide actions supporting the reduction of submissions after the due date or close to the deadline, and to propose a transparent and automatic approach to integrating KDD activities to the Moodle environment, where the data mining stage is restricted to the algorithms selected within this study and the results are presented in a simplified manner within the user interface in the Moodle environment. The study considers the time the assignment remained open for posting, the course to which the assignment was proposed and the actual time when the assignment was posted into the environment. It was carried out following the steps of the knowledge discovery process in databases, using the Weka tool. As a result from the KDD process performed in our database, the number of postings that were closer to the final expiry date were higher for assignments longer than 15 days, graduate courses tended to have longer assignments than undergraduate courses, and they also presented a higher number of postings after the due date or close to the expiry date of the assignments. In this context, shorter assignments are recommended, in order to increase postings soon after the opening of assignments and to enable teachers to obtain faster feedback from the learning process undergone by the student. That makes possible to take corrective actions in shorter time in order to avoid student failure or dismissal. The implementation of the KDD process within Moodle enables the experimentation by users in an automatic and simplified manner.

Keywords: virtual learning environment; moodle; kdd; data mining.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AVA	Ambientes Virtuais de Aprendizagem
API	Application Programming Interface
ARFF	Atribute-Relation File Format
CSV	Comma Separated Value
IA	Inteligência Artificial
JAR	Java Archive
JDBC	Java Database Connectivity
J2EE	Java 2 Platform, Enterprise Edition
KDD	knowledge discovery in databases
MD	Mineração de Dados
MOODLE	Modular Object-Oriented Dynamic Learning Environment
OLAP	On-line Analytical Processing
PHP	Hypertext Preprocessor
SQL	Structure Query Language
URL	Uniform Resource Locator
WEKA	Waikato Environment for Knowledge Analysis

SUMÁRIO

1 INTRODUÇÃO

1.1 Contexto e Motivação

Atualmente, é crescente a utilização de Ambientes Virtuais de Aprendizagem (AVA) como ferramenta de apoio à comunicação entre os envolvidos no processo pedagógico. Tanto no ensino à distância como nos cursos presenciais, o uso de tais ambientes possibilita compartilhar materiais, realizar tarefas e interagir com outros usuários, com o objetivo final de gerar e adquirir conhecimento, tanto em caráter individual como coletivo.

O Modular Object-Oriented Dynamic Learning Environment (Moodle) é um AVA distribuído como Software Livre. Sua concepção foi iniciada nos anos 90, por Martin Dougiamas, com base nas abordagens pedagógicas do construtivismo e do construcionismo social, onde o aluno contribui ativamente no processo de ensino-aprendizagem. O objetivo desse projeto, desde sua concepção, era de suportar a criação e administração de cursos com enfoque no trabalho colaborativo, em um ambiente de simples e intuitiva utilização. Desde então, o Moodle conquistou milhões de usuários, sendo adotado em milhares de instituições de diversos países.

Nos diferentes segmentos da sociedade, as organizações têm buscado na tecnologia recursos que agreguem valor aos seus negócios, seja agilizando operações, suportando ambientes ou viabilizando inovações (SANTOS SILVA, 2004). A inclusão desses sistemas de informação no cotidiano em atividades rotineiras da sociedade gera um grande volume de dados. Esses dados podem ser transformados em informações importantes para sucesso das organizações, com a aplicação do knowledge discovery in databases (KDD), Descoberta de Conhecimento em Banco de Dados. Tal processo consiste na extração de conhecimento de alto nível a partir de dados de baixo nível disponíveis em grandes bancos de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

A mineração de dados é uma etapa do processo de descoberta de conhecimento em grandes bancos de dados (BOENTE, 2006). Segundo (HOLSHEIMER et al., 1996) esta etapa combina métodos e ferramentas de pelo menos três áreas: aprendizagem de máquina, estatística e bancos de dados. Essas técnicas vêm sendo aplicadas para análise do grande volume de dados gerados em AVAs, com resultados promissores.

No presente trabalho, explora-se técnicas de mineração em dados gerados no AVA Moodle, com foco na análise de prazos de entrega de atividades. Tal análise pode auxiliar na gestão do tempo de alunos e professores, que é um aspecto relevante no processo de ensino-aprendizagem.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é explorar o uso de técnicas de mineração de dados para análise de entregas de atividades no AVA Moodle.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Identificar padrões quanto aos prazos e datas efetivas de submissão das tarefas, em uma base Moodle existente, que possam auxiliar no processo de aprendizagem e no planejamento do ensino.
- Propor uma abordagem sistemática para integrar atividades de mineração à base do Moodle, com foco específico na entrega de atividades.

1.3 Metodologia

Dado o caráter exploratório deste trabalho, seu desenvolvimento foi principalmente baseado em um estudo de caso, utilizando a ferramenta Weka para mineração de dados provenientes do AVA Moodle no Instituto Federal Farroupilha - Campus de São Vicente do Sul.

As etapas do trabalho foram organizadas como segue:

- Revisão da literatura: aquisição de conhecimento sobre o estado-da-arte em abordagens e ferramentas de mineração de dados, assim como suas aplicações a ambientes de aprendizagem;
- Coleta de dados: extração do conjunto de dados das tabelas do Moodle na instituição alvo, com foco específico na submissão de tarefas;
- Experimentação: preparação dos dados coletados para serem usados como entrada na ferramenta Weka, seguida da realização de experimentos com os algoritmos de mineração de dados;
- Análise de resultados: análise das saídas obtidas na execução dos algoritmos e a confrontação dos mesmos com o contexto de origem dos dados;

- Sistematização: proposta de ferramenta para integração de algoritmos do Weka ao Moodle.

Os algoritmos que geraram regras condizentes ao problema em estudo são selecionados e executados para apresentar os resultados finais da comparação entre prazos de entrega e submissões de conteúdo no ambiente, e implementados em uma ferramenta Web para a realização da MD em um ambiente simplificado, possibilitando a utilização por pessoas que não tenham domínio do KDD.

1.4 Estrutura do Trabalho

Este trabalho está organizado em cinco capítulos. O capítulo 1 apresenta os motivos, o contexto da realização do trabalho, objetivos gerais e específicos, uma visão geral da metodologia aplicada, e a estrutura de desenvolvimento do trabalho.

O capítulo 2 apresenta a fundamentação teórica sobre os conceitos e desafios relacionados ao processo de Descoberta de Conhecimento, com enfase na Mineração de Dados. Apresenta também os Ambientes Virtuais de Aprendizagem (AVA), com foco específico no ambiente MOODLE, suas características, funcionalidades, e estrutura de base de dados.

O capítulo 3 descreve o estudo de caso de KDD no Conjunto de Dados, gerados pela interação com o conjunto de dados trabalhados no estudo são proveniente das seguintes tabelas:*mdl _assignment,* que armazena informações sobre cada tarefa, e *mdl _assignment _submissions,* que armazena informações sobre cada tarefa submetida.

O capítulo 4 visa descrever uma proposta de integração de mineração de dados ao Moodle, com foco no conjunto de dados gerados pelos prazos de entrega de atividade do Moodle. Neste capítulo também é abordado a implementação de parte dessa proposta de integração.

Por fim, o capítulo 5 apresenta as conclusões obtidas e as propostas de trabalhos futuros, resumindo o resultado das análises e as principais contribuições obtidas com os experimentos realizados e com a implementação do MDS.

2 DESCOBERTA DE CONHECIMENTO EM BASE DE DADOS

Descoberta de Conhecimento em Base de Dados (ou em inglês, *knowledge discovery in databases*) é o processo não trivial que busca identificar, em bases de dados, padrões (tendências, regras, comportamentos) válidos, novos (até então desconhecidos), potencialmente úteis e humanamente compreensíveis.

Dentre os objetivos para seu uso estão melhorar o entendimento de um problema ou suportar um procedimento de tomada de decisão (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Para (HOLSHEIMER et al., 1996), o processo de descoberta de conhecimento é um método semi-automático complexo e iterativo.

Os termos apresentados na definição de KDD são explicados por (SILVA, 2002) e apresentados nos itens que seguem:

- Dados: consistem em unidades de informação que representam um conjunto de fatos F, como instâncias de um banco de dados. Por exemplo, uma coleção de n cadastros de pessoas físicas contendo idade, profissão, renda etc.

- Padrão: representa uma expressão E em uma linguagem L que descreve fatos individuais em um subconjunto FE de F. É dito um padrão E, se este é mais simples do que a enumeração de todos os fatos no subconjunto FE. Por exemplo, o padrão: *"Se renda* $<$ *r então a pessoa não recebe financiamento"* seria aplicável para uma escolha apropriada de r.

- Processo: no contexto de KDD, processo é uma sequência de vários passos que envolve preparação de dados, pesquisa de padrões, avaliação de conhecimento, refinação envolvendo iteração e modificação.

- Validade: este critério determina que os padrões descobertos em novos dados devem ser logicamente corretos com algum grau de certeza. Uma medida de certeza é uma função C mapeando expressões em L para um espaço de medidas MC. Por exemplo, se um limite de padrão de crédito é ampliado, então a medida de certeza diminuiria, uma vez que mais financiamentos seriam concedidos a um grupo até então restrito a esta operação.

- Novo: esse critério trata do caráter de "novidade" em um novo conhecimento, medido por uma função $N(E,F)$, que pode ser uma função booleana ou uma medida que expresse

grau de “novidade” ou “surpresa”. Exemplo de um fato que não é novidade: sejam E = “usa tênis” e F = “alunos de colégio” então $N(E,F) = 0$ ou $N(E,F)$ = false. Por outro lado: sejam E = “bom pagador” e F = “trabalhador da construção civil” então $N(E,F) = 0{,}85$ ou $N(E,F)$ = true.

- Potencialmente útil: os padrões descobertos na abordagem de KDD devem potencialmente levar a alguma atitude prática, conforme medido por alguma função de utilidade. Por exemplo, regras obtidas no processo podem ser aplicadas para aumentar o retorno financeiro de uma instituição.

- Humanamente compreensível: um dos objetivos de KDD é tornar padrões compreensíveis para humanos, com representação intuitiva e granularidade inteligível. Por exemplo: o log de um servidor *Web* não é uma representação compreensível; fatos extraídos deste log, tais como totais de acesso ou classificação dos acessos realizados, fornecem informação mais intuitiva e humanamente compreensível.

Tais características guiam o processo de KDD em multiplicar a capacidade humana em buscar e adquirir conhecimento a partir da presença de configurações de informação e de sua repetida ocorrência.

As atividades de KDD são classificadas por (GOLDSCHMIDT; PASSOS, 2005) em três classes:

1. Pré-Processamento: responsável pelas funções de captação, organização e tratamento de dados;

2. Mineração de Dados: responsável por realizar buscas efetivas por conhecimentos úteis em um KDD; e

3. Pós-Processamento: abrange o tratamento do conhecimento obtido pela etapa de mineração de dados.

As classes de atividades acima apresentadas agregam as seguintes etapas, de acordo com a ilustração da figura 2.1:

1. Definir o tipo de conhecimento a descobrir: compreende o planejamento do domínio da aplicação de conhecimento, e a definição do tipo de decisão que tal conhecimento pode contribuir para melhorar.

2. Criar um conjunto de dados alvo *(Selection)*: abrange a seleção de um conjunto de dados dentro de um banco de dados, ou concentra atenção num subconjunto onde a descoberta deve ser realizada.

3. Limpar dados e pré-processar *(Preprocessing)*: inclui a modelagem ou estimativa de ruídos, sua remoção quando necessário, a escolha de estratégias para manipular campos de dados ausentes, e a formatação de dados para a adequá-los à ferramenta de mineração.

4. Reduzir dados e projeção *(Transformation)*: compreende a localização de características úteis para representar os dados e reduzir variáveis e/ou instâncias dependendo do objetivo da tarefa, e o enriquecimento semântico das informações que devem ser buscadas.

5. Minerar dados *(Data Mining)*: inclui a seleção de métodos para localizar padrões nos dados, a execução de algoritmos para buscar por padrões de interesse, e a definição do melhor ajuste de parâmetros para a tarefa.

6. Interpretar padrões minerados *(Interpretation/Evaluation)*: considera a avaliação e a interpretação dos resultados, com um possível retorno aos passos 1-6 para posterior iteração.

7. Implantar conhecimento descoberto *(Knowledge)*: incorpora os resultados do processo de KDD como novos conhecimentos para o uso do sistema, ou faz com que estes sejam disponíveis em documentos ou base de conhecimento para uso das partes interessadas.

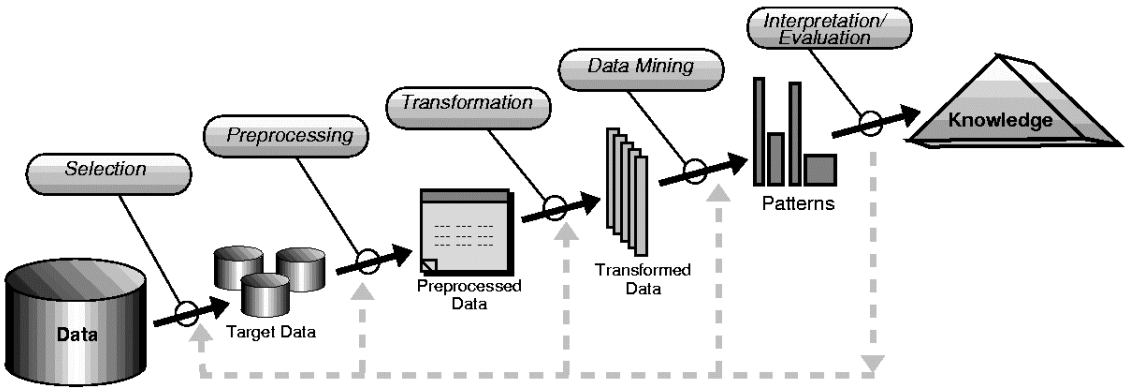


Figura 2.1: Etapas de Descoberta de Conhecimento em Banco de Dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996)

Somente é possível obter padrões relevantes se os dados são selecionados corretamente, e transformados para formatos adequados de acordo com os algoritmos aplicados. Ainda na

figura 2.1, observa-se o encadeamento das atividades relacionadas com o pré-processamento para o sucesso do processo de KDD.

O processo de KDD pode ter as suas metas classificadas em direta, supervisionada ou indireta, não-supervisionada. Na busca de conhecimento direta ou supervisionada tem-se uma meta definida, ou seja, uma ideia do resultado que está sendo buscado. Já na busca de conhecimento indireta ou não-supervisionada não são definidas claramente as metas, sendo que o objetivo é encontrar uma estrutura significativa nos conjuntos de dados.

O KDD é um processo semi-automático, no qual uma imensa quantidade de dados não é garantia de sucesso no modelo final. Isso acontece porque o resultado final depende da interação humana nos processos de escolha dos métodos aplicados, e cada uma das escolhas influência os resultados das etapas seguintes.

2.1 Mineração de Dados

Mineração de dados (ou, em inglês, *Data Mining*) é definida como sendo "o uso de técnicas automáticas de exploração de grandes quantidades de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertos a olho nu pelo ser humano" (CARVALHO, 2001).

2.1.1 Atividades de Mineração de Dados

Essas técnicas são implementadas em forma de algoritmos fundamentados pelos princípios de Indução e/ou Extração de Regras; Redes Neurais; Algoritmos Evolucionários e técnicas de Estatística. Tais princípios fornecem suporte à execução das diferentes tarefas de mineração de dados, através das atividades de descrição e previsão.

Descrição é a busca por padrões que descrevam os dados em um modelo de relações entre as variáveis. Exemplos: agrupamento, associação, sumarização e análise de desvio. Já na previsão, o foco está na generalização de valores e resultados conseguidos através de relacionamentos de variáveis contidos nas bases de dados. Essa generalização serve para prever valores futuros ou desconhecidos para variáveis de interesse. Exemplos: classificação e regressão.

2.1.2 Algoritmos e Técnicas de Mineração de Dados

Segundo (VIEIRA, 2008), as técnicas de MD podem ser aplicadas em tarefas de: classificação, estimativa, associação, agrupamento, sumarização. E pode ser acrescentado a esta lista a tarefa de desvios citada em (CASTANHEIRA, 2008).

- Associação: Essa tarefa consiste em descobrir atributos que ocorrem simultaneamente com grande frequência. Para analisar a qualidade das regras de associação, são utilizados os parâmetros de suporte e confiança, sendo o suporte a porcentagem de transações da base de dados que contêm os itens de A e B. Já a confiança é obtida pela relação das transações que possuem o item de A no total das transações dividido pela quantidade de transações que possuem os itens A e B. O algoritmo Apriori é um exemplo baseado na associação.

- Classificação: Para (HARRISON, 1998) *apud* (VIEIRA, 2008), a tarefa de classificação consiste em construir um modelo que possa ser aplicado em um conjunto de dados não classificados objetivando categorizá-los em classes.

  Um dado é analisado e classificado em uma classe definida, ou seja, busca a descoberta de funções que mapeiem registros em classes pré-definidas. Essas funções, após descobertas, são usadas como sistema de apoio à decisão para prever a classe em que determinados conjuntos de registros se enquadram. Os algoritmos dessa tarefa se utilizam de Redes Neurais, Algoritmos Genéticos e Lógica Indutiva.

- Clusterização ou Agrupamento: Com a aplicação dessa tarefa, os elementos com características semelhantes são agrupados em um mesmo cluster. Cada cluster apresenta internamente grande similaridade e grande diferença em relação aos outros cluster formados pelos conjuntos de dados. Diferente da tarefa de classificação, que tem rótulos pré-definidos, a clusterização identifica automaticamente os grupos de dados aos quais o usuário deve rotular (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). São exemplos dessa tarefa os algoritmos: K-Means, K-item.

- Algoritmos Genéticos: Técnica que trabalha paralelamente com a otimização e busca, baseada nos mecanismos de seleção natural e genética e gera resultados em vários conjuntos de solução simultaneamente. Estão entre eles os algoritmos, Modes, K-Prototypes, K-Medoids e EM.

- Sumarização: Com a tarefa de sumarização é possível indicar características comuns entre conjuntos de dados (GOLDSCHMIDT; PASSOS, 2005) e ainda segundo (VIEIRA, 2008) a sumarização utiliza métodos para encontrar uma descrição compacta para um subconjunto de dados. Geralmente é aplicada nos agrupamentos obtidos na clusterização. Os algoritmos são baseados na Lógica Indutiva e Genéticos.

- Estimativa: Segundo (CARVALHO, 2001), a estimativa tem o objetivo de determinar algum valor mais provável diante de dados já existentes ou de dados semelhantes sobre o qual se tem conhecimento.

  Quando se tem uma variável contínua desconhecida, é possível usar essa tarefa para estimar valores ausentes. Exemplos de estimativas podem ser: estimar o número de filhos em uma família; estimar a renda de um cliente; estimar o tempo de vida do cliente, entre outros (HARRISON, 1998).

- Detecção de Desvios: A tarefa de detecção de desvios tem por objetivo descobrir um conjunto de valores que não seguem padrões definidos. Para esta tarefa é necessário adotar padrões antecipadamente. Esses registros que não seguem os padrões normais do contexto são conhecidos como *outliers* (CASTANHEIRA, 2008).

  Os algoritmos dessa técnica são baseados na estatística para fornecer as funcionalidades necessárias à identificação de desvios nos registros.

Na figura 2.2 é apresentada uma estrutura do relacionamento das atividades e suas tarefas da etapa de Mineração de Dados.

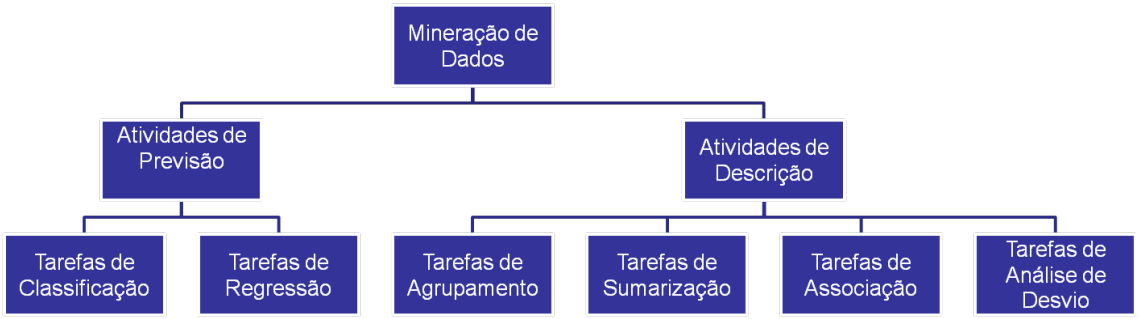


Figura 2.2: Relacionamento entre as atividades e tarefas de Mineração de Dados.

Não existe uma única técnica para resolver todos os problemas de mineração de dados (HARRISON, 1998), visto que cada uma das técnicas oferece vantagens e desvantagens para a sua utilização potencial. Deve-se conhecer o seu funcionamento e assim fazer a melhor escolha do método de acordo com o domínio da aplicação dos dados a serem trabalhados.

2.1.3 Ferramentas de Mineração de Dados

Com o crescimento da Mineração de Dados como ferramenta de descoberta de conhecimento, aumentou também a oferta de *software* para esse tipo de abordagem. Aplicações comerciais das empresas IBM *(Intelligent Miner)*, *SAS (Enterprise Miner)*, SPSS *(Clementine)*, e

opções de ferramentas *Open Source*, como por exemplo o Weka, auxiliam na execução do processo de Mineração de Dados.

Em meio a tantas opções de software de Mineração de Dados, é necessário observar algumas características na escolha da melhor opção. Tais características são abordadas por (GOEBEL; GRUENWALD, 1999) e enumeradas a seguir:

- A habilidade de acesso a uma variedade de fontes de dados, de forma *on-line e off-line*;
- A capacidade de incluir modelos de dados orientados a objetos ou modelos não padronizados (tal como multimídia, espacial ou temporal);
- A capacidade de processamento com relação ao número máximo de tabelas/tuplas/atributos;
- A capacidade de processamento com relação ao tamanho do banco de dados;
- Variedade de tipos de atributos que a ferramenta pode manipular;
- Tipo de linguagem de consulta.

A ferramenta de Mineração de Dados DBMiner teve seu desenvolvimento iniciado em 1989 pelos pesquisadores da Simon Fraser University do Canadá. Esta permite trabalhar com grandes bases de dados com conexão direta ao servidor, e disponibiliza análise *On-line Analytical Processing* (OLAP) que consiste em manipular e analisar de dados sob múltiplas perspectivas, tarefas de classificação, previsão, regressão, associação, clustering e exploração de dados sobre Data Mart ou Data Warehouse.

Os resultados podem ser visualizados através de gráficos, em uma ferramenta comercial, mas sua arquitetura fechada impede a inclusão de novas funcionalidades e a utilização de seus algoritmos em software externos.

Darwin é um produto desenvolvido e comercializado pela Oracle para atuar em grandes volumes de dados. Este suporta arquitetura cliente-servidor, processamento paralelo, é escalável e tem opção de conexão direta com o banco de dados da Oracle. Outros aplicativos SGBD são suportados via ODBC ou ainda mediante arquivos no formato texto. Darwin tem suporte a recursos de pré-processamento de dados, e na Mineração de Dados trabalha com as tarefas de previsão, regressão, classificação, clustering, associação, visualização, análise de dados exploratória, redes neurais e análise estatística. Os resultados podem ser visualizados por gráficos, histogramas, entre outros.

A ferramenta IBM SPSS Modeler pertence à empresa IBM e é comercializada em duas versões: Professional e Premium. A diferença entre as duas é que a última inclui Mineração de Texto. Em geral, a ferramenta suporta acesso direto e fácil aos dados, não necessitando que esses sejam estruturados. O programa disponibiliza os resultados em diversas maneiras, e permite integrá-los facilmente com outras aplicações.

RapidMiner é uma ferramenta de MD livre, que iniciou a ser desenvolvida em 2001, na Universidade de Dortmund. Possui interface parecida com a *Explorer* do Weka, só que disponibiliza mais operadores. Suporta a integração dos seus algoritmos em outras aplicações, utilizando API Java. Esta permite o acesso aos dados do Excel, Access, Oracle, IBM DB2, Microsoft SQL, Sybase, Ingres, MySQL, Postgres, SPSS, dBase, arquivos de texto entre outros. É multiplataforma – 100% Java – disponibiliza mais de 500 recursos para a entrada dos dados, pré-processamento, MD e visualização.

SAS Enterprise Miner usa técnicas de mineração de dados, como predição, descrição de dados, árvores de decisão, algoritmos de redes neurais, entre outros. Possui recursos que auxiliem na etapa de pré-processamento, recursos visuais estatísticos e flexíveis para apresentar os resultados e ajudar a determinar as variáveis mais determinantes no conjunto de dados para distinguir os agrupamentos. Realiza comparação entre os modelos criados e elege o melhor com base em critérios já definidos. É uma ferramenta proprietária e não disponibiliza sua implementação com ambientes externos.

No tópico seguinte será abordada a ferramenta de Mineração de Dados Weka, que foi utilizada neste trabalho.

2.1.3.1 Weka

Neste trabalho, foi utilizada a ferramenta Weka (WEKA, 2011), implementada pelos pesquisadores da Universidade de Waikato de Nova Zelândia, desenvolvida com linguagem de programação Java seguindo a abordagem de *framework*.

Aliadas à característica de ter o seu código fonte aberto, essas características facilitam a adaptação, a inclusão de novas funcionalidades em algoritmos e a portabilidade entre diferentes sistemas operacionais. A grande aceitação dessa ferramenta está relacionada às características elencadas acima, aliadas a uma interface amigável, que agrega um conjunto de algoritmos de classificação, associação, agrupamento, regressão, pré-processamento, visualização e pós-processamento.

Além dos recursos acessíveis através da GUI, também podem ser acessados outros recursos utilizando a *Application Programming Interface (API)* do WEKA. A ferramenta possibilita o acesso aos dados diretamente de bancos de dados via JDBC, através de URL, no formato CSV ou no formato próprio, chamado *Atribute-Relation File Format* (ARFF). Este último que obedece algumas regras para a organização dos dados, como a adoção de um cabeçalho que representa as variáveis e o seu tipo, onde os valores são representados entre chaves “” e separados por vírgulas.

A seguir será apresentada a estrutura do cabeçalho do arquivo ARFF:

- @relation: nome que identifica o conjunto de dados a serem trabalhados.
- @attribute: relaciona os atributos e o tipo que pode ser: Booleano, nominal, categórico ou numérico.
- @data: marca o início da apresentação dos registros da base de dados, cada registro separado por vírgula, e cada linha representando uma transação.

Exemplo de um arquivo formatado seguindo o padrão ARFF:

```
@relation jogar
@attribute ceu {sol, nublado, chuva}
@attribute temperatura {alta, baixa, suave}
@attribute umidade {alta, normal}
@attribute vento {não, sim}
@attribute jogar {não, sim}
@data
nublado,alta,alta,não,não
chuva,suave,normal,sim,não
sol,alta,normal,não,sim
```

O último atributo especificado no cabeçalho será adotado por padrão pelo Weka como sendo a classe a ser testada. Os demais atributos são considerados os atributos preditivos. A figura 2.3 representa a interface gráfica de inicialização do Weka, onde o usuário pode selecionar um dos quatros modos para trabalhar com os seus dados.

- A aplicação *Explorer* é a interface gráfica mais utilizada do Weka, agregando as etapas de pré-processamento, mineração de dados e pós-processamento.

Figura 2.3: Interface gráfica de inicialização do Weka.

- A aplicação *Experimenter* é a interface gráfica destinado à realização de testes estatísticos utilizados na comparação entre diferentes algoritmos de aprendizagem suportados pelo Weka.

- A aplicação *KnowledgeFlo* é uma interface gráfica semelhante ao Explorer, só diferencia pelo fato de trabalhar com fluxos de dados;

- A aplicação *Simple Cli* é a interface que se apresenta no modo texto, sendo a utilização por linhas de comando, é destinado a usuários avançados.

O modo Explorer do Weka (Figura 2.4) apresenta as seguintes opções: *preprocess, classify, cluster, associate, select attribute e visualize*. Neste estudo utilizou-se o modo Explorer, por esse motivo será apresentado o seu funcionamento.

A Figura 2.4 corresponde a interface *Preprocess*. Nela, o usuário seleciona e modifica os dados, observa informações sobre estatística dos atributos pertencentes ao conjunto de dados, que pode ser utilizados como prévia análise e suposição de resultados. Essa interface contém também as rotinas de pré-processamento, implementadas pelos filtros que realizam as atividades de: discretização, normalização, amostragem, seleção de atributos, transformação e combinação de atributos, entre outros.

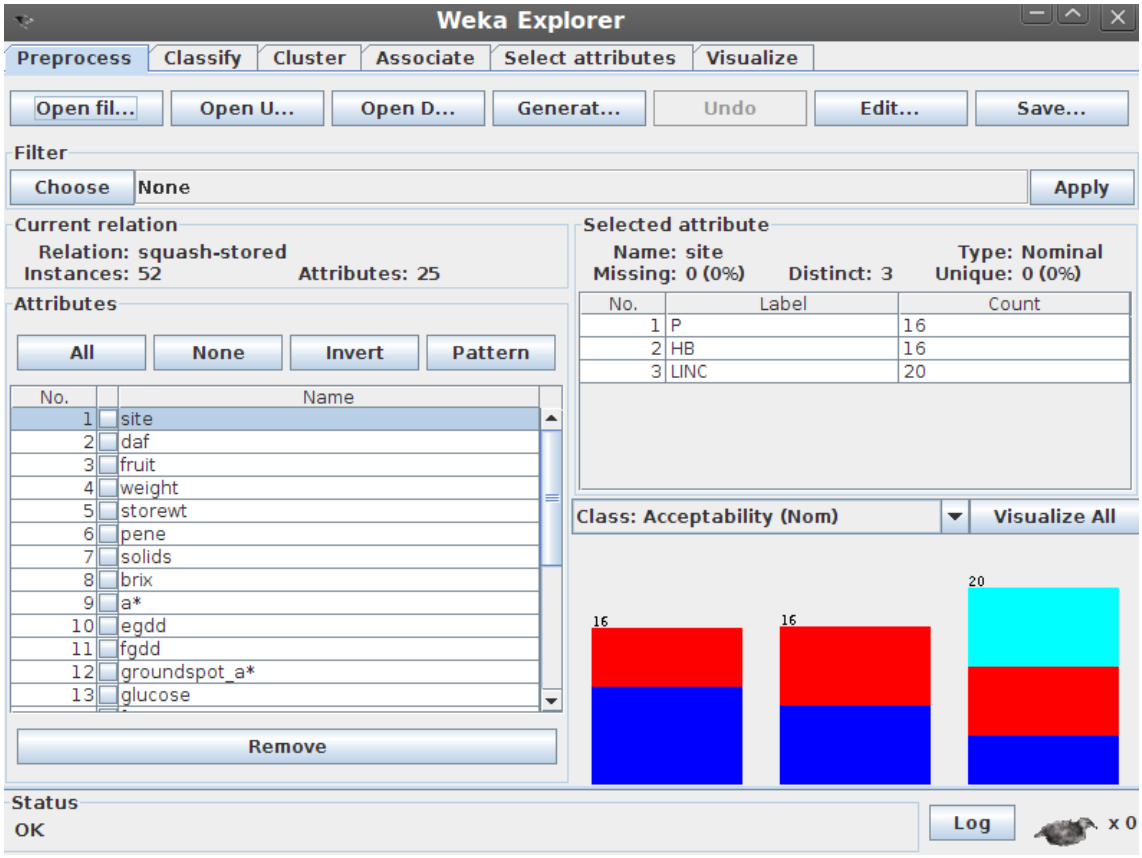


Figura 2.4: Interface do Weka - *Explorer*.

Esses filtros são responsáveis pelas rotinas que transformam atributos numéricos em nominal ou o contrário. Eles são também usados para manter a consistência dos dados (reduzindo a redundância dos mesmos), juntar os dados existentes com outros atributos para dar maior significância, e selecionar atributos mais influentes nos resultados esperados. Tais atividades permitem a transformação de atributos de base de dados em tipos de atributos suportados pela ferramenta Weka.

Classify: Essa interface disponibiliza acesso a vários algoritmos de classificação, onde o usuário pode determinar regras e testes de validação dos padrões obtidos no processo de busca de conhecimento.

Dentro dessa categoria, estão os algoritmos: Id3, C45, J48, BayesNet, Prism, J48, ZeroR, baseados nos métodos de árvore de decisão, Naive Bayes, tabelas de decisão, regressão, perceptron, perceptron multicamada, SVM entre outros.

Árvores de decisão: São baseados na técnica de aprendizado supervisionado, assemelhando-se com a estrutura de uma árvore, onde os nós representam subgrupos de acordo com os valores dos atributos. São implementados nos seguintes algoritmos: ID3, J4.8, LMT.

Vizinhos mais Próximos: Nessa técnica, se existir próxima a uma instância de classe desconhecida uma classe conhecida, as duas são agrupadas na classe já conhecida.

Redes Neurais: Técnica de aprendizado supervisionado na qual são construídos modelos matemáticos através da aprendizagem alcançada com o estudo das transações realizadas e armazenadas no banco de dados. Podendo ser: Perceptron quando existe para a entrada apenas um vetor de valores e o retorno é calculado em cima de uma combinação linear de atributos; Perceptron Multi-Camadas (MLP) é formado por múltiplas camadas de valores de entrada interconectadas onde o resultado é dependente do processamento realizado em camadas anteriores.

Naive Bayes: O algoritmo Naive Bayes é um classificador probabilístico, que utiliza cálculos de probabilidade baseando no Teorema de Bayes associado a fórmulas estatísticas para encontrar modelos. Esses modelos são gerados utilizando uma fórmula onde os resultados dos cálculos probabilísticos são considerados conhecimento prévio.

Clustering: Os algoritmos contidos nessa opção buscam verificar no conjunto de dados, grupos com atributos de características semelhantes e os une em classes homogêneas entre si e heterogêneas entre as outras classes do conjunto de dados. As classes são organizadas garantindo similaridade intraclasse e diferenciação extraclasse. Os principais algoritmos implementados no Weka são: *Cobweb, SimpleKmeans, Xmeans, Opiticis, DBScan, EM*.

Xmeans: É um algoritmo não-supervisionado, pois não tem classes definidas logo no início do processo. Esse algoritmo trabalha com atributos numéricos e busca formar grupos de objetos com menor distância entre eles, levando em conta a função da análise e comparações dos valores dos atributos mais próximos.

SimpleKMeans: Esse algoritmo trabalha com atributos categóricos e numéricos, normalizando os atributos através de cálculos de distância, que pode ser através da distância Euclidiana, e agrupa os atributos conforme a distância entre eles em classes, onde o número de classes que o algoritmo vai retornar é definido pelo usuário.

Cobweb: Utilizando esse algoritmo o usuário tem como resultado um modelo estatístico que representa os grupos definidos no conjunto de dados, por métodos estatísticos.

Associate: Neste painel o usuário tem acesso aos algoritmos implementados seguindo as técnicas de associação, que consistem em encontrar relação entre atributos que ocorrem com frequência juntos em diferentes transações. Os principais algoritmos são: *Apriori, Tertius*, entre outros.

Apriori: identifica os conjuntos de itens que aparecem com frequência no conjunto de da-

dos, após são geradas as regras que satisfazem as metas de confiança estabelecidas e assim expressam os resultados.

Tertius: busca regras sobre relações e co-ocorrências em conjuntos de dados, muito usado na verificação de associações em tabelas de associações.

Select Attributes: nesta aba o usuário encontra através de combinações um subconjunto que melhor representa uma previsão.

Visualize: A última opção na interface *Explorer* é a Visualize. Nela o usuário visualiza os atributos em uma parcela de matriz de pontos, que podem ser selecionados e expandidos.

Pode ser usada tanto no início do processo de mineração de dados para vislumbrar a distribuição dos dados, suas relações e propor hipóteses, como no processo posterior à descoberta de modelos para visualizar os resultados, regras, grupos, associações e contextualizá-las a fim de, confirmar ou não o resultado obtido.

Para validar os modelos obtidos a ferramenta disponibiliza os seguintes recursos:

- cross validation, suplied test set, use training set, percentage split: recursos que possibilitam teste e validação, através de parâmetros de validade e confiabilidade.

- matriz de confusão, índice de correção e incorreção de instâncias mineradas, estatística kappa, erro médio absoluto, erro relativo médio, precisão, F-measure: são os indicadores estatísticos que possibilitam a análise dos resultados.

A escolha por esta ferramenta deve-se pelo fato da mesma possuir as seguintes características: Código fonte aberto; Pacote de API; Disponibilizar grande quantidade de algoritmos das tarefas de Classificação, Agrupamento, Associação; Suportar funcionalidades para o pré-processamento; Visualização dos resultados também graficamente, que dependendo do algoritmo pode ser gráficos ou árvore; Ambiente amigável; Diferente opões de acesso aos dados, podendo ser direto ao Banco de Dados ou no formato CSV, ou ainda ARFF; e diferente possibilidades de configuração dos parâmetros dos algoritmo.

2.2 Ambientes Virtuais de Aprendizagem

Os Ambientes virtuais de aprendizagem (AVAs) são utilizados como locais para compartilhamento de materiais, realização de tarefas, criação e execução de projetos, postagem de dúvidas e busca de respostas e interações efetivas entre sujeitos autônomos. Servindo como plataforma para tais tarefas, esses ambientes proporcionam a construção de conhecimento, su-

perando os materiais e propostas de atividades iniciais e favorecendo as aprendizagens individuais e coletivas (FAGUNDES, 2006).

Esses ambientes são ricos em recursos e mídias, que dinamizam a interação entre as pessoas e objetos de conhecimento, por possibilitar o compartilhamento de informações de maneira mais organizada. Tais plataformas possibilitam apresentar informações diversas de formas diferenciadas, contribuindo para a aprendizagem dos diferentes tipos de pessoas e interesses específicos.

Os benefícios da utilização de um AVA vão além de uma plataforma para o ensino a educação a distância. Este também contribui como ferramenta de aprendizagem no ensino presencial da sala de aula. Atualmente, dentre os AVAs mais adotados no Brasil estão Moodle, Teleduc e Tidia – Ae. No tópico seguinte será abordado mais detalhadamente o Moodle, por ser desse ambiente o conjunto de dados trabalhados no processo de descoberta de conhecimento.

2.2.1 Moodle

O Modular Object-Oriented Dynamic Learning Environment (Moodle) é um software *Open Source*, que teve seu desenvolvimento iniciado nos anos de 1990, por Martin Dougiamas, com base nas filosofias de aprendizagem do construtivismo e do construcionismo social, suportando a criação e administração de cursos com enfoque no trabalho colaborativo em um ambiente de simples e intuitiva utilização.

Entre as principais características do Moodle, pode-se elencar a tradução do ambiente para 50 idiomas, e o seu sistema modular que possibilita incluir diferentes recursos e atividades durante a oferta de um curso. Assim, o sistema se adapta facilmente a diferentes contextos e níveis de trabalho (exigências) conforme a necessidade do momento.

2.2.1.1 Principais características e funcionalidades do Moodle

O acesso aos recursos do ambiente Moodle é controlado por classificação de perfis, sendo eles:

- Administrador: responsável pela administração, configurações do sistema, inserção de participantes e criação de cursos;
- Tutor: responsável pela edição, inserção e gerenciamento de tarefas e material, e pelo gerenciamento do curso; e
- Usuário: têm apenas acesso aos cursos em que esteja inscrito ou naqueles que não exigem inscrições.

As principais ferramentas disponibilizadas pelo ambiente são de comunicação, recursos de atividades, avaliação, administração e organização. As ferramentas de comunicação envolvem os fóruns de discussões e chat; avaliação pode ser de cursos, pesquisas de opinião, enquetes e questionários.

Os recursos e atividades estão relacionados com as maneiras de trabalhar com os conteúdos, sendo que a postagem desses pode ser realizada por meio de páginas de texto simples, páginas Web e links para arquivos ou endereços da Internet. Outra opção é a tarefa que permite criar textos online, gerar arquivos e enviá-los; o Wiki gera documentos cooperativos, a opção calendário possibilita visualizar e marcar datas e prazos.

2.2.1.2 Modalidades de uso: interfaces

As ferramentas de administração e organização envolvem o gerenciamento dos participantes e dos cursos. Nesse estudo, será trabalhado com o conjunto de dados que o Moodle disponibiliza sobre as atividades referentes às tarefas de quatro tipos distintos: Modalidade avançada de carregamento de arquivos; Texto Online; Envio de arquivo único; Atividade Offline. As interfaces das atividades tarefa seguem o padrão apresentado na figura 2.5, alterando apenas alguns atributos conforme a modalidade da mesma.

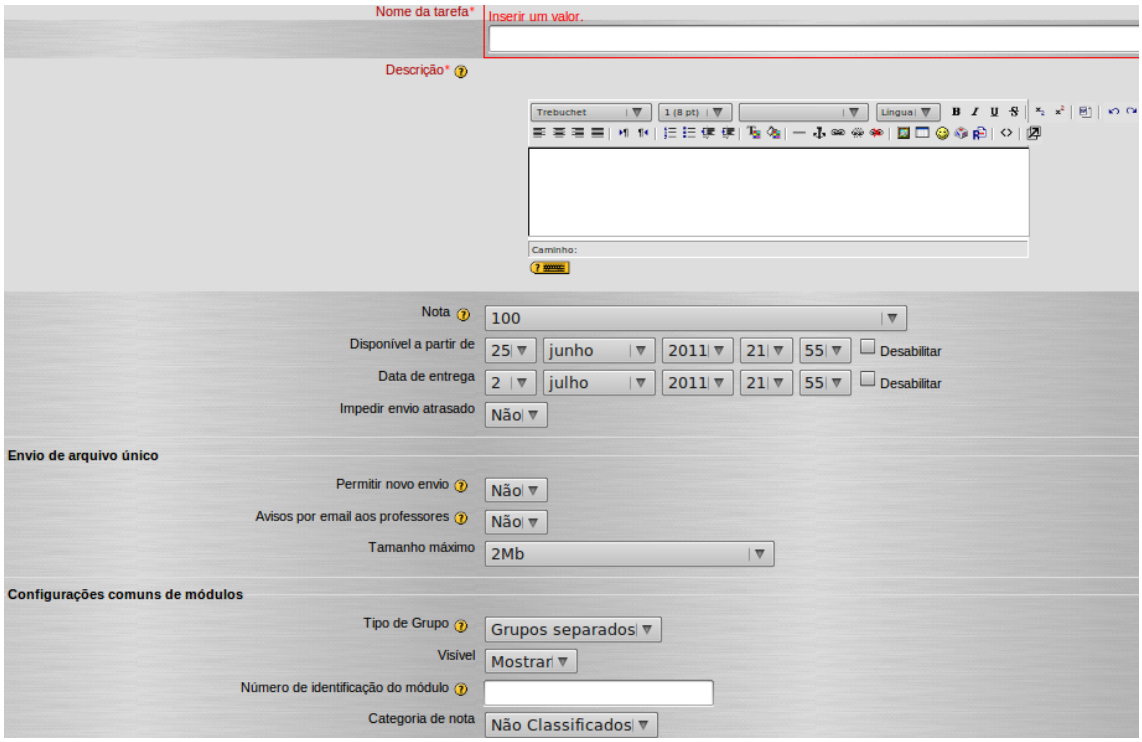


Figura 2.5: Interface da atividade tarefa.

2.2.1.3 Atributos das tarefas

As interfaces das diferentes modalidades de tarefas apesar de diferentes são flexíveis e baseiam em um conjunto de atributos com uma estrutura padrão de classes e objetos. Os mesmos serão abordados resumidamente na tabela 2.1.

Tabela 2.1: Atributos atividade tarefa.

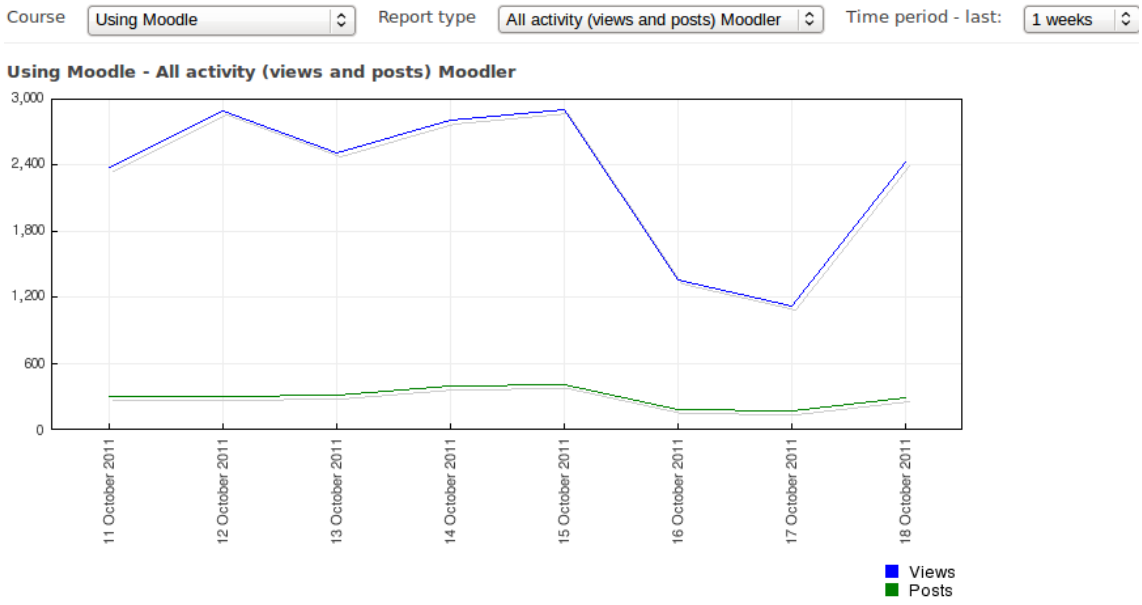
Nome da tarefa	Nome da tarefa
Descrição	Descrição da tarefa
Nota	Nota correspondente à tarefa
Disponível a partir de	Quando será aberta para postagem
Data de entrega	Data limite para a realização da postagem
Impedir envio atrasado	Impede o envio após o prazo estipulado
Permitir novo envio	Permite reenvio de uma tarefa
Alertar os tutores por e-mail	Envio de e-mail na postagem
Comentário inserido na frase	Envio da frase para tela de comentários
Tamanho máximo	Tamanho máximo de arquivo
Número máximo de arquivos carregado	Quantidade de arquivos para postagem
Esconder descrição antes da data de abertura	A tarefa só é exibida após a abertura
Permitir notas	Permite anotações no texto
Habilitar Envio para Avaliação	Aviso aos professores sobre a postagem
Tipo de Grupo	Seleciona o tipo de grupo
Número de identificação do módulo	Identificação para cálculo de avaliação
Categoria de nota	Categoria para representação de notas

A configuração destes atributos influencia na realização das tarefas e nos resultados obtidos na realização das mesmas. De fato, recursos como o envio atrasado de tarefa incentivam ao não cumprimento do prazo estipulado. Outros atributos favorecem o sucesso da atividade, tais como: Comentário na frase, habilitar envio para avaliação, permitir notas, que possibilitam um meio de comunicação eficiente entre os envolvidos. Dos atributos apresentados na tabela 2.1 foram empregados no estudo os seguintes: Data de Entrega e Disponível a partir de. Sendo os outros atributos do estudo gerados pela interação da submissão da tarefa.

2.2.1.4 Relatórios e Estatísticas

Os relatórios e estatísticas do Moodle são gerados com base nos dados de acesso, cursos, atividades e usuários. Utiliza-se de filtros para a configuração dos relatórios, devendo selecionar o curso, participantes, grupo, dia ou período, atividades, recursos, ações realizadas, entre outras opções. Os resultados podem ser visualizados no Moodle ou em arquivos nos formatos pdf, ods ou excel.

Esse conjunto de relatórios e estatística possibilita o controle sobre o andamento das disciplina, interação dos envolvidos e aproveitamento. Na figura 2.6 observa-se um gráfico e uma tabela gerados pelas estatísticas de acesso ao ambiente pelos usuários.



Period ending (day)	Views	Posts	Logs
17 October 2011	2427	282	Course Logs
16 October 2011	1116	169	Course Logs

Figura 2.6: Apresentação do resultado estatístico do Moodle (MOODLE, 2011).

Os relatórios são importantes para o gerenciamento do Moodle, mas os distribuídos juntamente com a plataforma são limitados visto a grande quantidade de dados gerados pelo ambiente. Para facilitar o acompanhamento dos cursos muitas instituições desenvolvem seus próprios relatórios afim de suprir com as demandas de informações, como esse processo de desenvolvimento requer conhecimento de programação algumas empresas acabam vendendo o Moodle com um produto customizado de acordo com as necessidades do cliente.

O GMoodle, figura 2.7, é um exemplo de customização do Moodle desenvolvido pela empresa Badiu. Essa ferramenta consiste em um sistema de gerenciamento da plataforma Moodle que permite a integração de vários ambientes Moodle, inclusive com versões diferentes e em servidores distintos. Destaca-se por disponibilizar relatórios gerenciais com dados de vários Web sites Moodle, apresentação gráfica e em tabela, relatórios flexíveis onde os filtros possibilitam diferentes configurações de seleção dos dados. Sua utilização pode ser livre ou comercial, variando entre as duas distribuição apenas a quantidade de usuários, que na opção livre fica em até 250 usuários ativos.

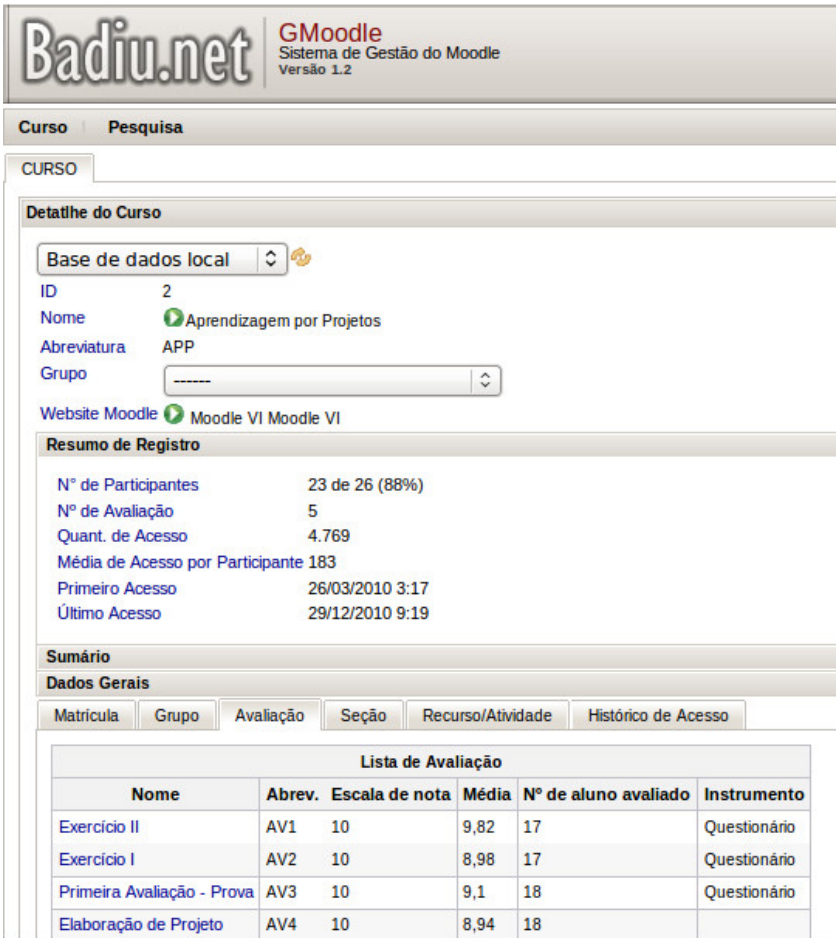


Figura 2.7: Interface do GMoodle (BADIU, 2011).

2.3 KDD em dados gerados por AVA

Existem na literatura diversos trabalhos relacionados com o nosso tema de pesquisa. Por exemplo, (DIAS, 2008) aplicaram técnicas de mineração de dados nos dados coletados a partir de um ambiente virtual de aprendizagem, voltado para o ensino de *Structured Query Language* (SQL) chamado LabSQL. Foram considerados os seguintes parâmetros:

- atributos pessoais dos alunos, do curso, turma e disciplina;
- o tempo que cada aluno demorou para se inscrever na turma;
- se o aluno realizou atividades em equipe;
- se o aluno utilizou agenda e anotações do sistema;
- o total de problemas resolvidos e dos pontos obtidos na realização dos mesmos;
- o nível de dificuldade dos exercícios resolvidos; e,

- a quantidade de acessos ao ambiente.

No estudo apresentado em (BARUQUE, 2007) foram analisadas as ferramentas do Moodle (chat, lição quiz, fórum, wiki, etc) que são preferidos pelos alunos de diferentes cursos, a fim de disponibilizar os recursos conforme os perfis. Este trabalho utilizou da ferramenta de mineração de dados Magnum Opus, que se baseia nas regras de associação para a descoberta de conhecimento.

No estudo realizado por (ROMERO; VENTURA; GARCÍA, 2007), observa-se a busca pela classificação e predição dos alunos, conforme o desempenho obtido, com base nos dados coletados de sete cursos do Moodle da Universidade de Córdoba. Esse modelo vai ao encontro da adequação das atividades propostas através do Moodle para o andamento dos cursos. Analisando os recursos que mais surtiram resultados positivos por cursos, uma ferramenta em Java foi implementada utilizando o QUILHA, *framework* de código aberto para construir modelos de mineração de dados (ROMERO; VENTURA; GARCÍA, 2007).

O estudo realizado por (PRADO LIMA; WEBBER; GUIMARÃES, 2011) incluiu métodos automáticos de análise de dados nos ambientes virtuais de aprendizagem para facilitar o acompanhamento do desenvolvimento cognitivo individual ou de grupos de alunos pelo professor, com informações que fornecem uma visão sintetizada do processo de aprendizagem em tempo hábil.

Para a tarefa de Mineração de Dados, foi utilizado o algoritmo EM da técnica de clustering, implementado na ferramenta Weka. O conjunto de dados utilizado foi gerado a partir da disciplina lógica de programação, na qual os alunos devem resolver algoritmos estruturados e que futuramente podem ser executados computacionalmente. Como resultado, obteve-se o reconhecimento de grupos de alunos com o mesmo perfil de aprendizagem, o que permite ao professor tomar decisões baseados no nível de desenvolvimento cognitivo do aluno ou do grupo de alunos, em cada uma das etapas de aprendizagem.

O estudo desenvolvido na tese de (KAMPFF, 2009) busca suportar maior acompanhamento do andamento das atividades mediadas por AVA ao professor, para que o professor consiga intervir com ações pedagógicas em tempo nos casos com tendências problemáticas, como reprovação ou evasão. A ferramenta de MD utilizada foi RapidMiner e algoritmos de classificação, sendo os seguintes elementos monitorados pela MD: frequência de acesso ao ambiente virtual, participação nas atividades, prazos de entrega de atividades, desempenho, aliados às características demográficas e comportamental. No sistema de alertas em AVA proposto, os

alertas podem ser fixos, configurados pelo professor ou gerados através de regras geradas pela MD. Esses alertas servem para notificar situações sobre um determinado aluno ou grupo de alunos. O professor recebe alertas quando os alunos não estão nos parâmetros estipulados. Esse acompanhamento pode ser explicito, como na realização de uma atividade, ou pode indicar períodos de notificação sobre os alunos que estejam realizando acessos ao material. Já os alertas baseados em padrões utilizam-se das regras de classificação nos dados históricos, aplicadas aos dados de turmas em andamento. Os alertas foram analisados em um estudo com alunos de cursos presenciais em uma disciplina totalmente EAD. No mesmo observou-se as tendências:

- alunos que não realizaram todas as atividades propostas obtiveram notas abaixo da média;
- alunos com uma frequência maior de acesso ao material de estudo obtiveram nota acima da média.

Como resultado, observou-se que os alertas contribuíram para o aumento das aprovações e diminuição nas evasões, isso devido às intervenções que o professor realizou com base nos alertas, direcionadas a grupos que necessitavam de uma atenção especial.

Dentre os trabalhos relacionados, não foi encontrado análises com foco em prazos de entrega de atividades. No entanto, há indícios de que este tipo de análise é importante para apoiar decisões e formulação de estratégias para um adequado gerenciamento do tempo dedicado a atividades de uma ou mais disciplinas (PILLING-CORMICK, 1997).

3 ESTUDO DE CASO

Após estudo sobre KDD e o Moodle, foi necessário conhecer a estrutura do banco de dados do Moodle e o relacionamento entre tabelas, para identificar os relacionamentos do conjunto de dados e os possíveis atributos a serem trabalhados para a descoberta de conhecimento.

A primeira etapa a ser realizada é a pré-processamento que, segundo (A. P. A. BATISTA, 2003), tem como objetivo garantir e aprimorar a qualidade dos dados, já que a qualidade dos dados influi diretamente no conhecimento a ser extraído. As próximas seções descrevem as etapas do Estudo de Caso para a descoberta de regras observando os prazos de postagem das tarefas do Moodle.

3.1 Compreensão do negócio

Conforme já apresentado no capítulo 2, a principal funcionalidade do Moodle é como ferramenta para o apoio às atividades pedagógicas nos cursos presenciais e à distância. Nesse ambiente, professores e alunos interagem através dos recursos disponibilizados pelo Moodle.

Entender do contexto do negócio de onde são extraídos os dados para o KDD é primordial para a correta seleção, transformação dos dados e análise das regras geradas na Mineração de Dados. Portanto, no tópico seguinte aprofundado o estudo sobre a estrutura do Moodle.

3.1.1 Estrutura da Base de Dados do Moodle

A estrutura da base de dados do Moodle é composta por 203 tabelas, que armazenam todas as informações sobre os cursos, usuários cadastrados, acessos, atividades e avaliações, sendo desenvolvida em SQL. A estrutura da base de dados é apresentada na Figura 3.1.

Dentro da estrutura do banco de dados do Moodle, se destacam duas tabelas, referentes a atividade de tarefa e tarefa submetida, fontes do conjunto de dados trabalhados nesse estudo: *mdl_assignment* e *mdl_assignment_submissions*. Dos 55 cursos, somente 23 utilizaram o recurso de tarefa em sua realização, onde 276 usuários interagiram em 63 atividades, totalizando 677 registros.

Na realização deste trabalho trabalhou-se com uma base de dados real, gerada pelo AVA Moodle versão 1.9.19, do Instituto Federal Farroupilha – Campus de São Vicente do Sul. O ambiente contém 55 cursos cadastrados, sendo a distribuição entre os níveis de ensino apresentada na figura 3.2.

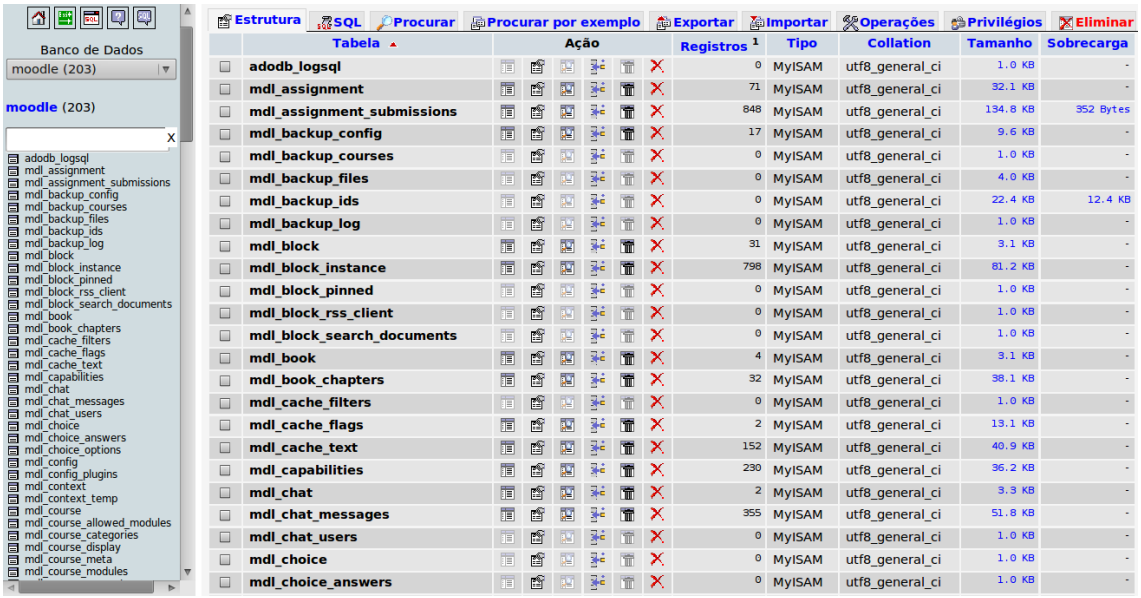


Figura 3.1: Estrutura da Base de Dados do Moodle.

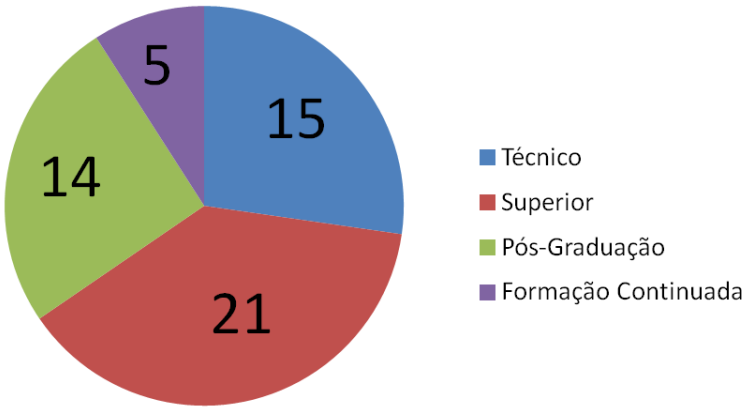


Figura 3.2: Diferentes níveis de ensino

3.1.2 Estrutura das tabelas

A tabela *mdl_assignment* armazena informações sobre cada tarefa (Figura 3.3). Já a tabela *mdl_assignment_submissions* (Figura 3.4) armazena informações sobre cada tarefa submetida. As tabelas 3.1 e 3.2 apresentam a relação entre o atributo presente na base de dados e o apresentado na interface do Moodle para os usuários.

3.2 Seleção dos Dados

Para a seleção dos dados, foram realizadas consultas no banco de dados utilizando a linguagem *Structured Query Language(SQL)*. A tabela 3.3 apresenta os atributos selecionados para gerar o arquivo ARFF utilizado pela ferramenta Weka, o conjunto dos dados selecionados passou pelas etapas dos tópicos seguintes.

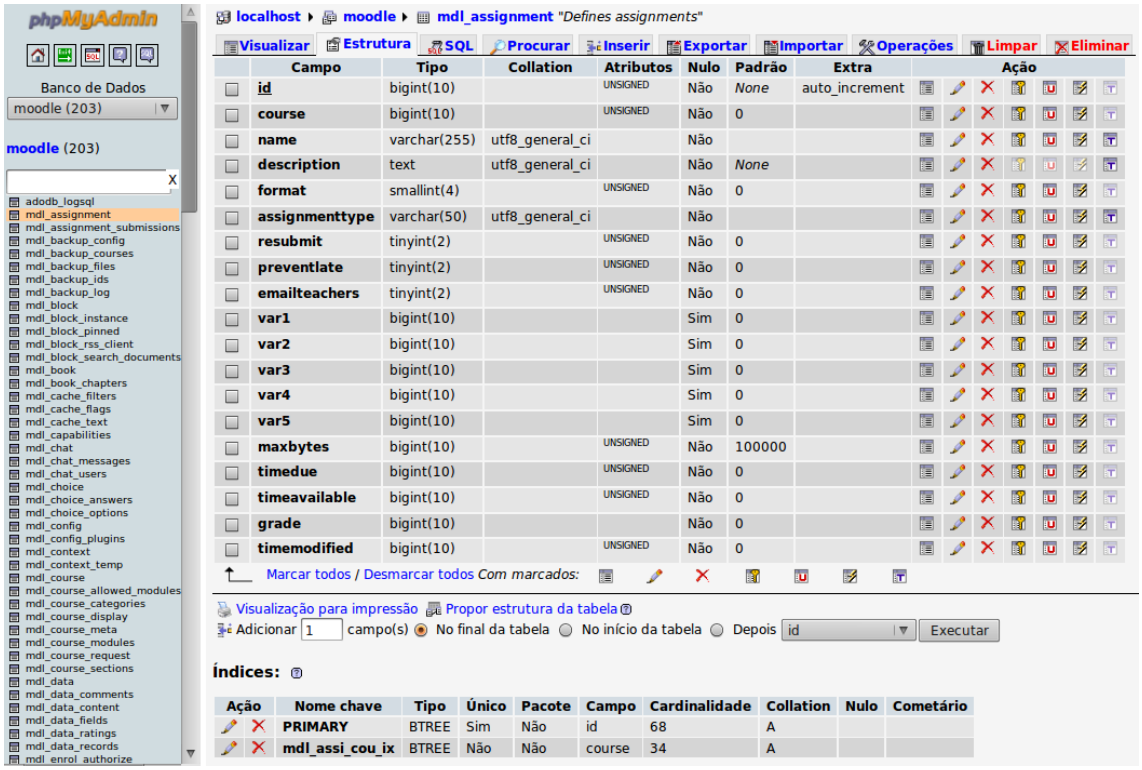


Figura 3.3: Estrutura da tabela referente a atividade - Tarefa.

Tabela 3.1: Relação dos atributos da tabela Tarefa.

Tabela	Atributo	Descrição
Assignment Atividade	ID	Identificação da atividade
	Course	Código do Curso
	Name	Nome da tarefa
	Description	Descrição
	Format	Formato
	Assignmenttype	Tipo da atividade
	Resubmit	Reenviar
	Preventlat	Permitir envio fora do prazo
	E-mailteachers	Envio de aviso para o e-mail do professor
	Var1	Número de arquivos que podem ser enviados
	Var2	Permitir notas
	Var3	Esconder a descrição antes da data de abertura
	Var4	Habilitar envio para avaliação
	Maxbytes	Tamanho máximo do arquivo a ser enviado
	Timedue	Prazo final para a entrega
	Timeavailable	Prazo inicial para a entrega
	Grade	Nota
	Timemodified	Data em que foi salvo as alterações

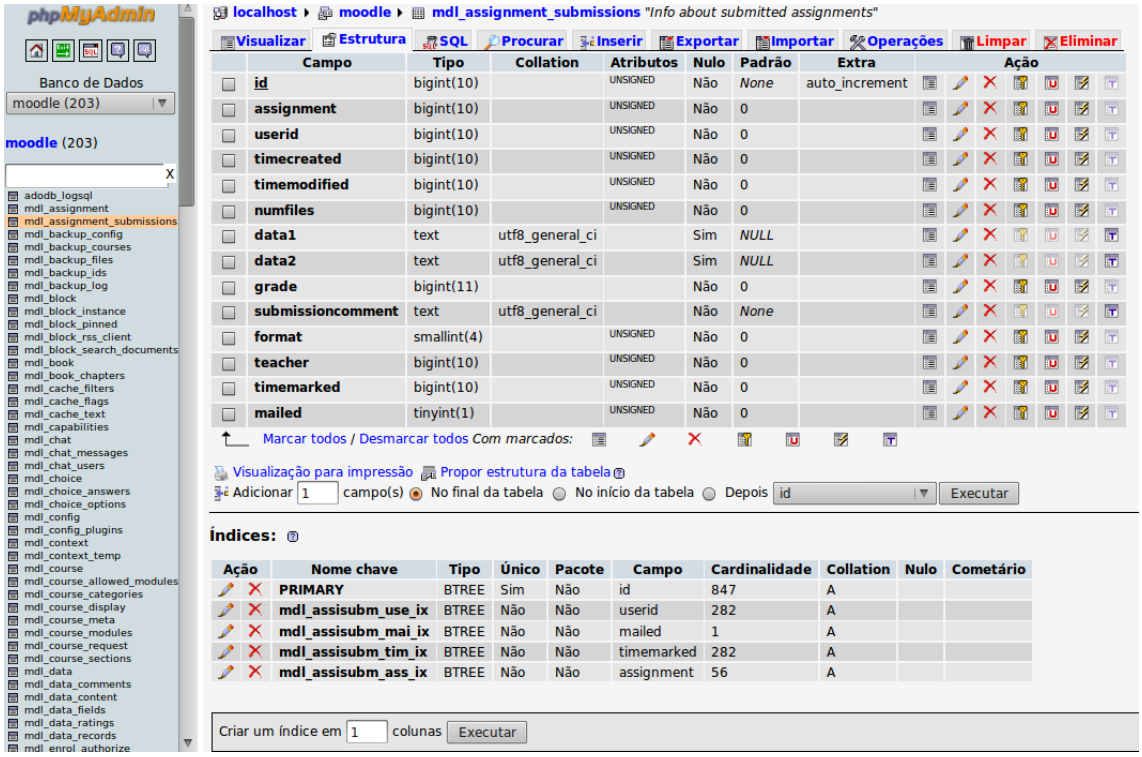


Figura 3.4: Estrutura da tabela referente a atividade - Tarefa Submetida.

Tabela 3.2: Relação dos atributos da tabela Tarefa Submetida.

Tabela	Atributo	Descrição
Assignment Submissions	ID	Identificação da atividade submetida
	Assignment	Identificação da atividade
	Userid	Identificação do usuário
	Timecreated	Quando foi criado a atividade
	Timemodified	Quando foi enviado para o ambiente
	Data 1	Notas da submissão
	Data 2	Se foi submetido
	Grade	Nota
	Submissioncomment	Comentários sobre a submissão
	Format	Formato
	Teacher	Identificação do professor
	Timemarked	Data em que foi enviado para o ambiente
	Mailed	Enviado

3.3 Limpeza e transformação do dados

Nessa etapa foram removidos os atributos que não agregam informações para a análise dos dados, restando o conjunto de dados formado pelos atributos, identificação do nível do curso, data de início, data de término para postagem e a data em que foi efetivada a postagem.

Em algumas atividades os atributos estavam com valores faltantes, sendo que devido à falta

Tabela 3.3: Relação dos atributos selecionados das tabelas atividade e atividade submetida.

Tabela	Atributo	Descrição
Assignment	ID	Identificação da atividade
	Course	Curso
	Timedue	Prazo final para a entrega
	Timeavailable	Prazo inicial para a entrega
	Timemodified	Data modificação
	Userid	Identificação do usuário
Assignment Submissions	Timemodified	Quando foi enviado para o ambiente
	Timemarked	Data em que foi enviado para o ambiente

desses houve situações em que se fez necessário a remoção da atividade para não comprometer o conjunto dos dados.

Depois da execução de alguns algoritmos foi observada a dificuldade de interpretar os dados expressos na forma de datas, por isso foi adotado outros parâmetros para esses atributos. De fato, conforme cita (VIEIRA, 2008), "Quando o algoritmo minerador a ser utilizado não é capaz de analisar certo dado, este geralmente é transformado em outra informação que o algoritmo de MD é capaz de analisar".

Para facilitar a geração e interpretação de regras foram criados os atributos: Período de Postagem; Entrega Realizada e Porcentagem Diferença entre Postagem.

- Período de postagem: o esquema para a geração deste atributo é apresentado na figura 3.5.

Figura 3.5: Atributo Período de Postagem.

- Entrega realizada: o esquema para a geração deste atributo é apresentado na figura 3.6.
- Porcentagem Diferença entre postagem: o esquema para a geração deste atributo é apresentado na figura 3.7.

Outra transformação realizada foi no formato dos atributos gerados a partir das datas, pois no banco de dados do Moodle elas são armazenadas no formato *Timestamp*. Este controla o tempo em uma soma cumulativa de segundos desde a zero hora do dia 01 de janeiro de 1970. A fim de

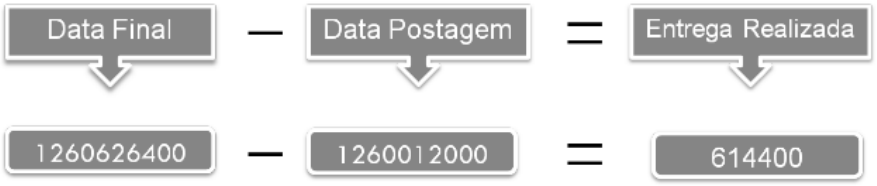


Figura 3.6: Atributo Entrega Realizada.

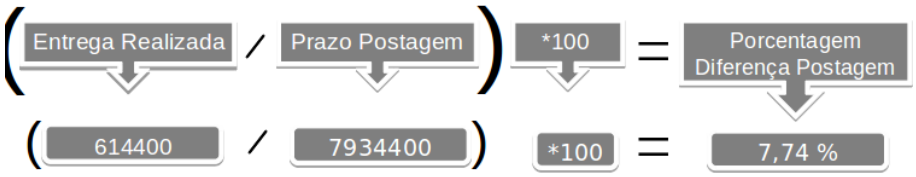


Figura 3.7: Atributo Período de Postagem.

facilitar a legibilidade dos resultados, o período expresso em *Timestamp*, foi transformado para o formato dias e horas. Para isso, utilizou-se a fórmula apresentada na figura 3.8.

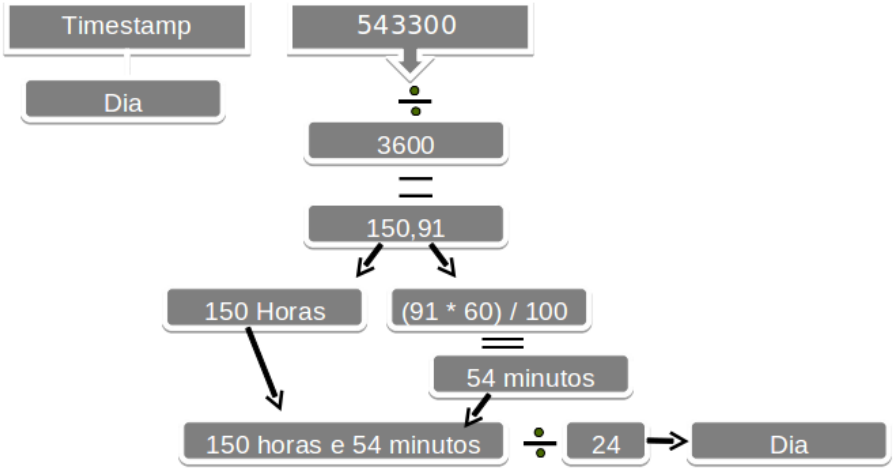


Figura 3.8: Esquema para a transformação do formato data *Timestamp* para Dia.

Nos atributos apresentados na tabela 3.4, ocorrem um grande número de valores diferentes. Esse fato pode dificultar a geração de regras e sua análise, para tal, pode ser aplicado na etapa de pré-processamento um algoritmo de discretização, que tem como entrada valores de um atributo contínuo e a saída é gerada em lista com os intervalos ordenados. Cada intervalo é apresentado pela seguinte delimitação: [Limite Inferior : Limite Superior]. Foi selecionado o filtro da categoria *supervised*, agrupando em períodos os dados numéricos. Assim, os dados são apresentados por meio de agrupamento de seus valores reduzindo o número de valores dos atributos.

Sendo assim, o atributo período de entrega, que possuía 64 valores diferentes, foi agrupado

em 37 períodos. O atributo Entrega realizada que continha 565 valores diferentes passou a ser classificado em 20 faixas de tempo. Já o atributo Porcentagem Diferença entre postagem, que apresentava 598 valores diferentes, passou a ser classificado em 9 faixas de período.

Essa transformação se faz necessária para aumentar a legibilidade das regras encontradas e também para possibilitar a execução dos algoritmos de associação.

Tabela 3.4: Relação dos atributos, seus valores mínimo, máximo e a média.

Atributo	Mínimo	Máximo	Média
Período de Postagem	0,3	4.560	363,6
Entrega Realizada	-1.048,11	4.393,01	130,2
Porcentagem Diferença entre Postagem	-7.894,28	99,96	11,77

Outro atributo criado foi situação da postagem, o qual é classificado analisando o atributo Porcentagem Diferença Postagem. Como resultado obteve-se 6 classes, tabela 3.5.

Tabela 3.5: Classificação segundo a situação da postagem.

Classe	Limite Inferior	Limite Superior
Classe 1	-	< 25%
Classe 2	>= 25%	< 50%
Classe 3	>= 50%	< 75%
Classe 4	> = 75%	< 95%
Classe 5	> = 95%	< = 100%
Classe 6	> 100%	-

- Classe 1: quando postado logo na abertura da atividade, ou seja, a postagem ocorreu entre o intervalo dos 25% iniciais do tempo total para a postagem;
- Classe 2: quando a postagem ocorreu entre o intervalo superior ou igual a 25% do tempo total da postagem e inferior a 50% para a postagem;
- Classe 3: quando postado entre o intervalo superior ou igual a 50% do tempo total da postagem e inferior ou igual a 75% do tempo;
- Classe 4: quando postado entre o intervalo superior ou igual a 75% do tempo total da postagem e inferior ou igual a 95% do tempo;
- Classe 5: quando postado no intervalo de tempo superior a 95% do período em que a atividade está aberta para postagem; e,

- Classe 6: quando postado após o término do período de postagem da atividade.

Na tabela 3.6 é demonstrada a relação da situação de postagem com a quantidade de ocorrência das mesmas.

Tabela 3.6: Relação das classes e a quantidade de ocorrência de cada uma.

Classe	Classe 1	Classe 2	Classe 3	Classe 4	Classe 5	Classe 6
Quantidade	318	62	70	48	106	73

Com o intuito de obter uma identificação mais simplificada do prazo para a postagem, essas foram classificadas em 5 classes, com base na quantidade de horas que permaneceram abertas e podem ser visualizadas na Figura 3.9

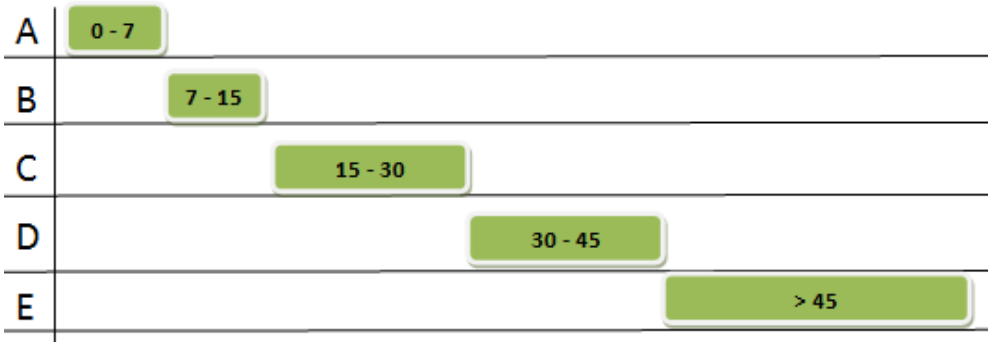


Figura 3.9: Classificação Prazo de Postagem.

- Classe A: atividades com o período de postagem de até 168 horas, ou seja, até 7 dias.
- Classe B: atividades com período de postagem acima de 168 horas e até 360 horas, ou seja, acima de 7 dias e até 15 dias.
- Classe C: atividades com período de postagem acima de 360 e até 720 horas, ou seja, acima de 15 dias e até 30 dias.
- Classe D: atividades com período de postagem acima de 720 e até 1080 horas, ou seja, acima de 30 dias e até 45 dias.
- Classe E: atividades com período de postagem acima de 1080 horas, ou seja, acima de 45 dias.

A distribuição nas diferentes Classes de períodos de entrega relacionadas com o Nível de Ensino, podem ser observadas na Figura 3.10.

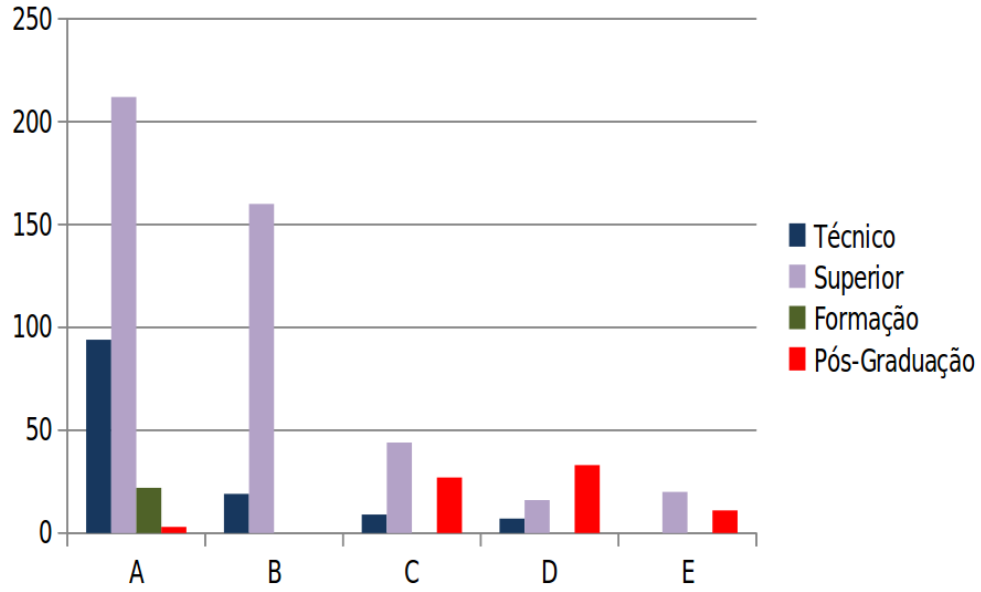


Figura 3.10: Atividades classificadas por tempo de postagem e Nível de Ensino.

3.4 Mineração

A primeira atividade realizada foi de agrupamento, na qual foi buscado encontrar grupos que tenham semelhança entre eles, relacionando os atributos *período de entrega, porcentagem diferença entre postagem e situação*.

Para tal, foram executados os algoritmos: *SimpleKmeans, Cobweb, DBScan e o Expectativa*(EM), sendo que esse último apresentou regras que relacionam os prazos das tarefas com o nível de curso, com resultados estatísticos melhores que nos outros.

Na busca de associações entre os atributos período de postagem, entrega realizada, porcentagem diferença entre postagem, situação e curso, foram executados os algoritmos: APRIORI e Tertius.

Outro questionamento era quanto a relação do tempo para a postagem e o período no qual ocorreu a mesma, relacionando com os níveis de ensino. Para encontrar essas regras utilizou-se os algoritmos baseados na atividade de classificação, na qual é possível com base em um subconjunto de dados determinar o valor de um atributo.

Da execução dos algoritmos: *Id3, J4.8, ADTree, UserClassifier, PredictionNode, ClassifierTree, Prism, Part, Naive Bayes*. O algoritmo selecionado foi o *J4.8*, por apresentar resultados com estatísticas melhores.

3.4.1 Algoritmo J4.8

O algoritmo J4.8 faz parte da ferramenta Weka, sendo esse algoritmo uma implementação derivada dos algoritmos ID3 e C4.5, fundamentados na descoberta de conhecimento baseado na estrutura de uma árvore de decisão. Nesta técnica, é construída uma árvore de decisão baseado num conjunto dados de treino. Cada determinação de nó acontece escolhendo um atributo do conjunto de dados que mais eficazmente divide os dados. Tem um nó principal o nó pai que representa um atributo e suas ramificações que são cada nó interno da árvore, os quais representam uma decisão sobre um atributo que determina como os dados são particionados pelos nós filhos, os resultado são apresentados em Top-Down, do sentido nó raiz para as folhas. O tamanho da árvore está relacionado com a quantidade de nós determinados pelo fator de confiança informado, esse fator significa o percentual estatístico para medir a confiança dos dados classificados corretamente.

Nas figuras 3.11 e 3.12 são apresentados alguns parâmetros que podem ser alterados na execução do algoritmo J4.8, a fim de obter melhores resultados.

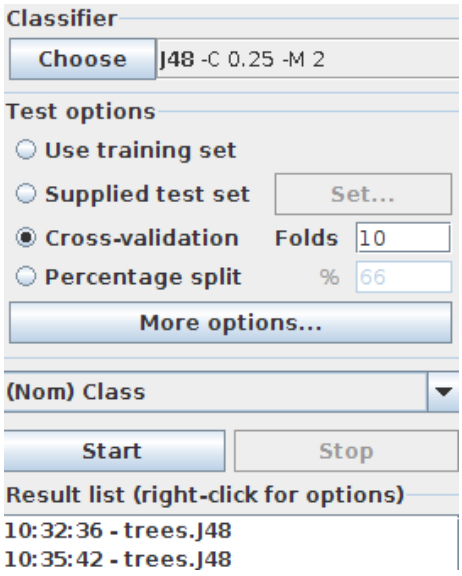


Figura 3.11: Parâmetros do algoritmo J4.8 do WEKA.

Na figura 3.11 visualiza-se algumas opções de configurações para a execução, como:

Use training set: seleção automática por parte do algoritmo do atributo melhor classificador do conjunto de atributos;

Supplied test set: definição de um conjunto de dados de um arquivo para servir de treinamento;

Cross-validation: definição do número de cruzamentos para a validação cruzada;

Percentage split: pré-definição do percentual de dados pertencentes ao conjunto de dados que será utilizado na avaliação.

Logo abaixo ao botão *More Options* consta uma caixa de seleção onde são exibidos todos os atributos do conjunto de dados para a seleção do atributo que será utilizado de predição. Por padrão é utilizado o último atributo do conjunto.

A Figura 3.12 nos apresenta outros parâmetros do algoritmo J4.8, abordados nos tópicos a seguir.

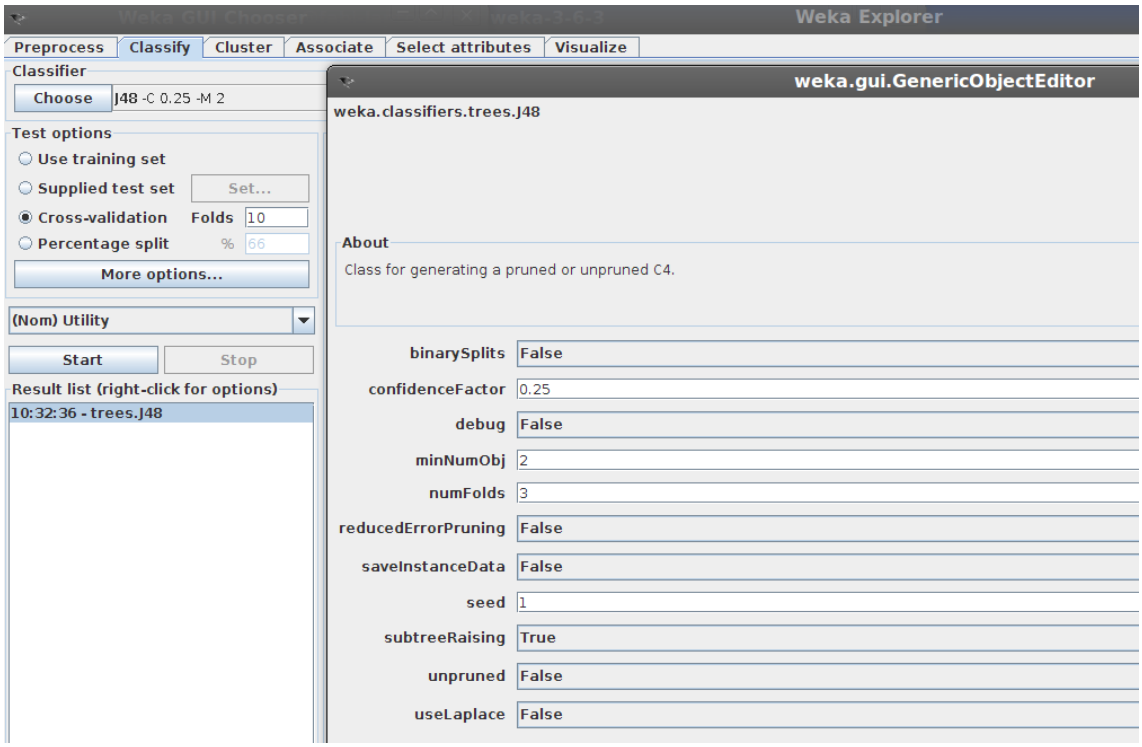


Figura 3.12: Parâmetros do algoritmo J4.8 do WEKA.

- BinarySplits: determina se será usada divisão binária;
- ConfidenceFactor: determina o fator de confiança inicial para a poda;
- Debug: se verdadeiro exibe mais informações sobre a execução;
- NumMinObj: determina o número mínimo de instâncias por folha;
- NumFolds: determina número de partições com a redução de erro na poda;
- ReducedErrorPruning: realiza a poda com redução de erros;
- SaveInstanceData: pode ou não apagar a árvore depois de construída;

- Seed: define o número de sementes que serão selecionadas para a execução com o parâmetro de redução de erros ativado na poda;
- SubTreeRaising: define a utilização de subárvore de poda ou não;
- Unpruned: constrói a árvore sem poda; e
- UseLaplace: ativa ou não a contagem do número de folhas excluídas.

Na Figura 3.13 pode ser observadas as seguintes informações: que a árvore foi podada; apresentação textual da árvore de decisão, sendo que em cada folha da árvore é indicada a quantidade de instâncias classificadas naquela folha; percentual e quantidade de instâncias classificadas corretamente e também sobre erros que aconteceram na classificação.

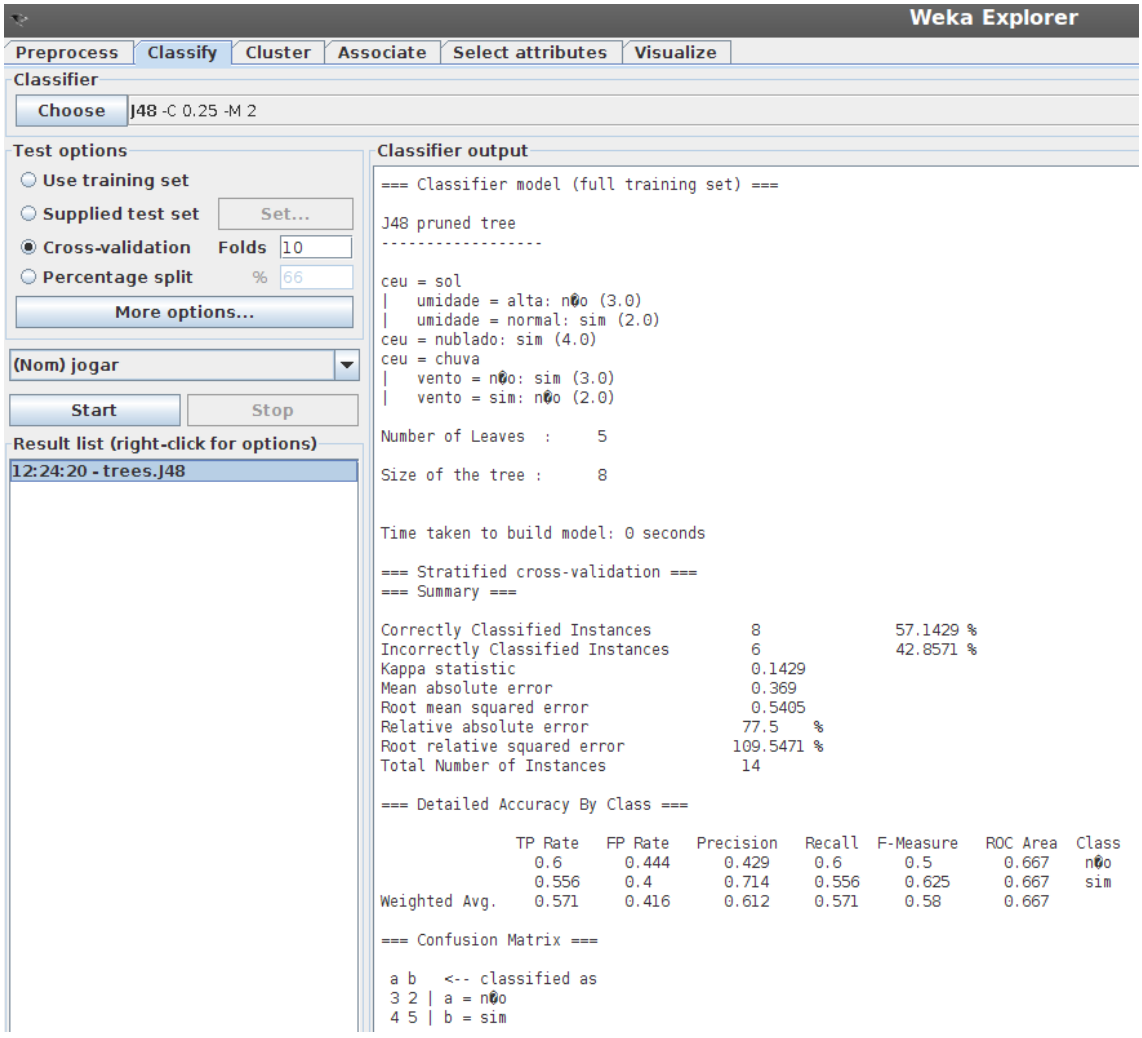


Figura 3.13: Resultados do algoritmo J4.8 do WEKA.

Na parte inferior dessa figura é apresentada uma matriz de confusão, que segundo (GOLDSCHMIDT; PASSOS, 2005), serve para fornecer um detalhamento do desempenho e qualidade

do modelo de classificação. Na matriz de confusão observa-se os acertos e erros na classificação por atributo.

Outra representação do resultado da execução do algoritmo J4.8 é a árvore de classificação gerada através da indução dos dados (Figura 3.14).

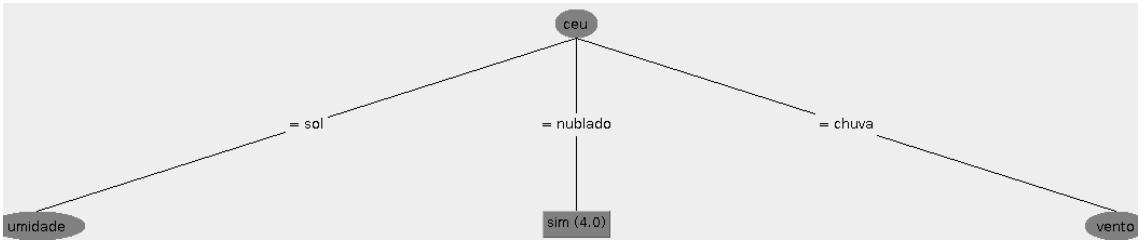


Figura 3.14: Árvore de decisão gerada pelo algoritmo J4.8 do WEKA.

3.4.2 Algoritmo EM

Nas tarefas de agrupamento, o sistema identifica as classes, isto é, agrupa os dados e descobre subconjuntos de objetos relacionados ao conjunto de treinamento, encontrando descrições de cada um destes subconjuntos (DILLY, 1995).

Um exemplo de algoritmo de tarefa de Agrupamento é o *Expectation Maximization*(EM) ou em português Esperança-Maximização (BUTZEN, 2008). Esse algoritmo consiste em um método interativo para estimar e maximizar a verossimilhança dos parâmetros de um conjunto de dados incompletos ou não conhecidos. Os dados são agrupados em Cluster onde é possível associar os dados com diferentes cluster e para definir a probabilidade de pertencer ao cluster é atribuído um peso.

Os cálculos do EM consistem em dois passos: E (expectativa) que calcula o valor de uma função de verosimilhança com base no conjunto de dados observados e as estimativas atuais dos parâmetros a fim de, determinar dados incompletos ou não conhecidos; e M (Maximização) é o passo onde são reestimados os parâmetros do modelo estimado em busca da maximização da verosimilhança que resulta em uma nova estimativa. A cada passo executado cresce a verossimilhança dos dados agrupados.

A Figura 3.15 apresenta as opções de configurações para a execução, sendo as três primeiras semelhantes às apresentadas na Classificação. Já a quarta opção é o botão *Ignore attributes*, que quando executa abre uma nova janela que possibilita a seleção de atributos para serem ignorados na execução do algoritmo.

O resultado da execução do algoritmo é apresentado na Figura 3.16, onde no início da tela

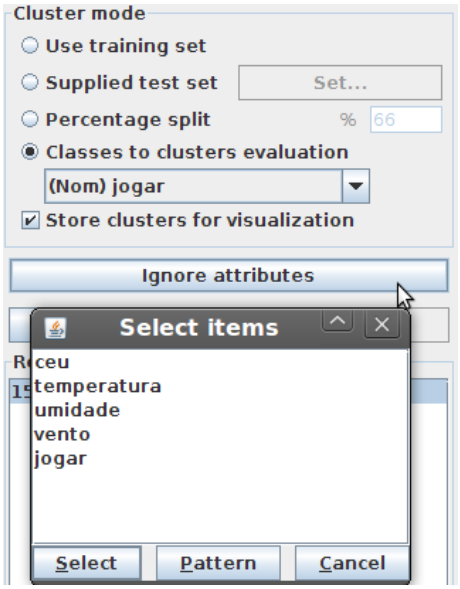


Figura 3.15: Parâmetro *Ignore attributes* do algoritmo EM do WEKA.

é apresentado o número de cluster, logo abaixo a descrição com a probabilidade anterior do cluster e distribuição de probabilidade para todos os atributos do conjunto de dados, o número de instâncias de formação em cada cluster, e o log de verossimilhança dos dados de treinamento em relação ao agrupamento.

3.4.3 Algoritmo APRIORI

O algoritmo Apriori segue a tarefa de Associação, sendo segundo (SANTOS, 2005) o algoritmo mais conhecido de identificação de associações. Este realiza buscas sucessivas em toda a base de dados, no intuito de encontrar relacionamentos entre os atributos e combinações.

Para a execução desse algoritmo é necessário que os dados estejam no formato discretos ou nominais. Para tal aplica-se um filtro que discretiza dados usando intervalos iguais, trocando a categoria numérica por uma nominal correspondente ao intervalo. Esta rotina é implementada no aplicativo weka.filters.unsupervised.

Como resultado do APRIORI é apresentado o tamanho dos conjuntos de *itemsets* com suporte mínimo e as melhores regras de associação com os números de instâncias ou ocorrências para as quais a associação acontece. As regras são ordenadas em ordem de confiança.

3.4.4 Execução dos Algoritmos

O algoritmo de associação Apriori apresentou duas regras, conforme descrição na Figura 3.18. Ambas as regras têm em comum que os valores do atributo situação fazem relação as

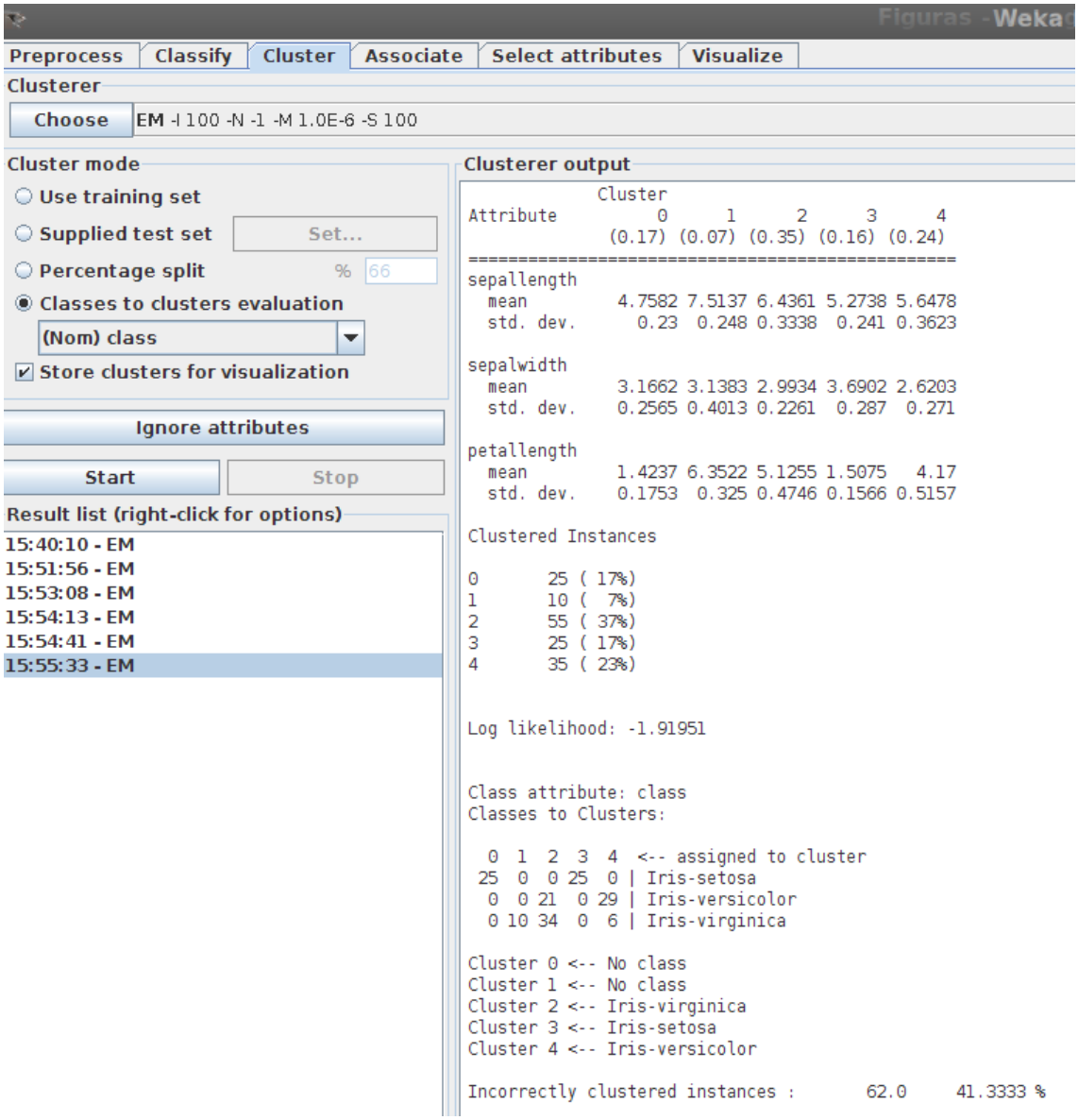


Figura 3.16: Exibição dos resultados do Algoritmo EM do WEKA.

atividades postadas em até 50% do período para postagem.

Na primeira regra, a situação inicial representa 318 ocorrências, sendo que estas estão no período de postagem de até 19.42% do período de postagem. A segunda regra apresenta que 106 ocorrências foram realizadas no intervalo de 25% a 50% do período de postagem. As execuções dos algoritmos EM e J4.8 são discutidas na próxima seção, pois os mesmos geraram regras que podem identificar relações analisando o prazo de postagem das tarefas do Moodle.

3.5 Resultados

Conforme descrito no tópico anterior, o Apriori não apresentou regras satisfatórias sendo nessa seção abordados os resultados apresentados pelos algoritmos EM e J4.8.

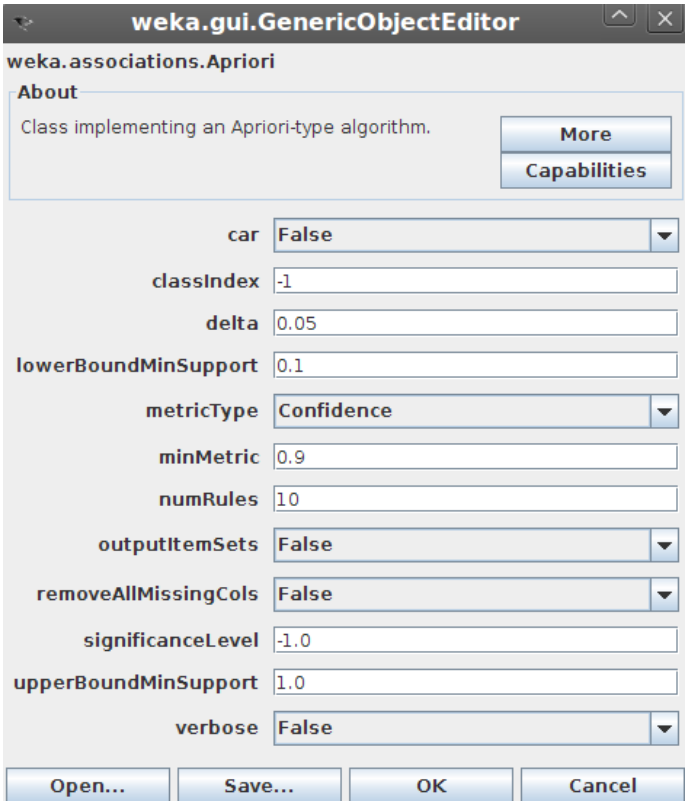


Figura 3.17: Parâmetros do algoritmo Apriori.

Na execução com o algoritmo de agrupamento EM, que busca identificar semelhança máxima entre parâmetros de modelos estatísticos, obtive-se a divisão dos dados em cinco cluster, apresentados na Figura 3.19.

- Cluster 0: Neste cluster tem-se um maior número de atividades dos cursos de pós-graduação, onde pode ser observada a incidência de atividades com tempo de postagem acima de 15 dias e também onde tem-se a maior ocorrência de postagem de atividade realizadas após 50% do período para a postagem.

- Cluster 1: Observa-se um grupo de cursos do nível de ensino superior onde tem-se a maior incidência de postagem realizada mais para o final da expiração do prazo. Isso nos mostra uma tendência em realizar a postagem para o final, mesmo tendo um grande período para realizar essa atividade.

- cluster 2: Nesse grupo tem-se a maior incidência de cursos do nível superior e técnico, com o prazo de postagem na grande maioria de até 30 dias, sendo que as atividades foram postadas com maior incidência na metade do período total para a postagem.

- Cluster 3: Nesse grupo também tem-se a maior incidência de cursos do nível e técnico e

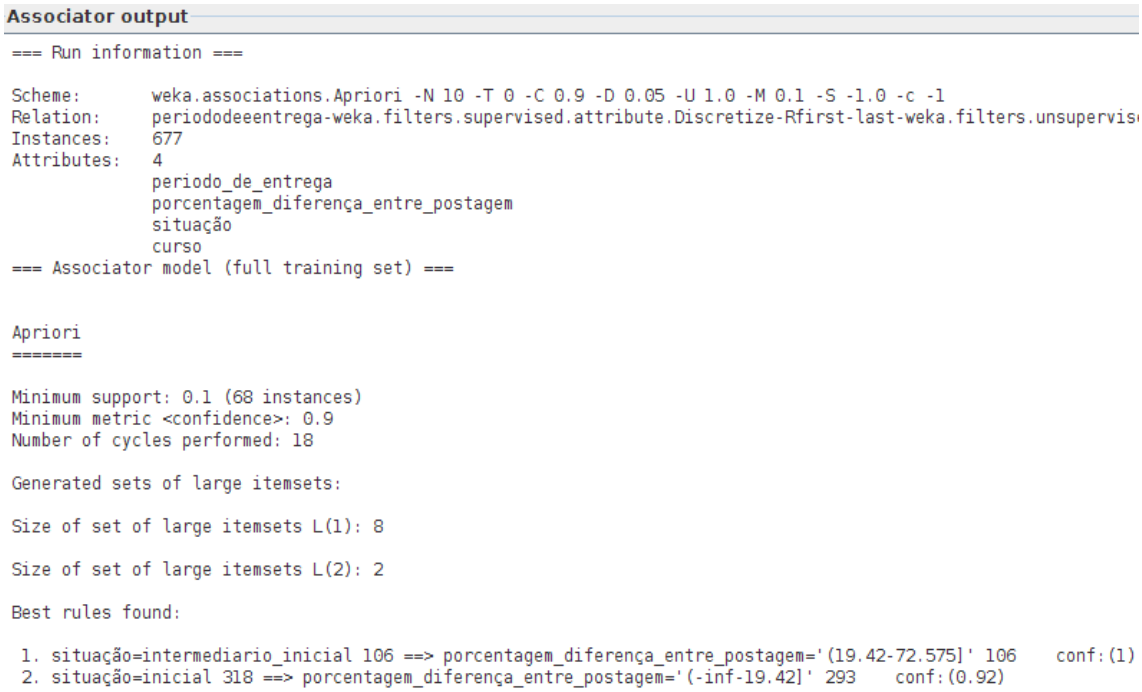


Figura 3.18: Resultado algoritmo APRIORI.

mais da metade de todas as atividades do nível superior estão neste agrupamento, com o prazo de postagem concentrado em até 15 dias, sendo que as atividades foram postadas com maior incidência nos 25% iniciais do período para a postagem.

- Cluster 4: Observa-se a tendência dos cursos de nível técnico e de formação continuada, com período de postagem na grande maioria de até 7 dias, mas ao contrário dos outros clusters, neste não tem-se a concentração de postagem em determinada classificação de postagem.

Nas figuras 3.20, 3.21 e 3.22, pode ser observado os resultados obtidos com a execução do algoritmo J48. É apresentada a árvore de classificação obtida pela execução do algoritmo, onde verifica-se que os cursos de nível superior têm suas atividades criadas com o tempo para postagem em sua grande maioria de até 30 dias e a postagem ocorre geralmente nos primeiros 25% do período total para postagem.

Já para os cursos técnicos, mesmo com período de postagem na grande maioria de até 7 dias, observa-se um maior ocorrência de postagem nos 50% finais ao período de postagem. Com os cursos de pós-graduação nota-se que os mesmos possuem tempo para postagem em média superiores a 30 dias e a efetivação da postagem ocorre no intervalo superior a 75% do tempo total de postagem e com entregas após o término do período para a postagem.

Observou-se que é comum a conduta de alguns alunos de realizar a postagem das tarefas

```
                         Cluster
Attribute                      0       1       2       3       4
                          (0.11)  (0.04)  (0.29)  (0.45)  (0.11)
=================================================================
classe_periodo_entrega
  A                       1.5033  5.0634 155.3759 101.1946 71.8627
  B                       1.0616  5.0243  10.4837 165.0472  1.3832
  C                       29.373  1.3776  29.3687  24.3782  1.5025
  D                      35.5449  1.4953   3.3756  15.9168  5.6674
  E                      11.4712 18.9244   1.3718   3.1312  1.1015
  [total]                 78.954 31.8851 199.9756  309.668 81.5173
situação
  Classe_1               25.0321  5.5348  61.6085 226.4639  4.3606
  Classe_2               13.5541  1.4807  62.0934  32.0077  1.8641
  Classe_3                8.2855  1.4333  41.4448   8.5726 18.2638
  Classe_4               25.9326  1.4574  10.9258  12.5374  2.1467
  Classe_5                 1.904  20.983   3.3362   4.6311 44.1457
  Classe_6                5.2457  1.9957   21.567  26.4553 11.7363
  [total]                 79.954 32.8851 200.9756  310.668 82.5173
nivel_curso
  Superior                3.3218 26.5467 149.6412 273.1865  5.3038
  Pós                     70.772  1.7896  2.4005   1.3667  2.6712
  Técnico                 2.8586  1.3221 45.8417  33.0845 50.8931
  Formação                1.0017  1.2266  1.0922   1.0304 21.6491
  [total]                 77.954 30.8851 198.9756  308.668 80.5173
Clustered Instances

0       71 ( 10%)
1       25 (  4%)
2      176 ( 26%)
3      312 ( 46%)
4       93 ( 14%)


Log likelihood: -3.37075
```

Figura 3.19: Agrupamento utilizando o algoritmo EM.

no final do período de postagem, podendo comprometer o processo de aprendizagem, visto que a configuração de aceitar ou não a submissão de trabalhos após o término pode ser ou não selecionada no momento em que são criadas as atividades. Outra tendência observada é que nas atividades com período de tempo maior que 15 dias, tem-se o aumento no número de postagem efetuadas mais próximas ao término do período de postagem do que em atividades com o tempo de até 7 dias.

Nos cursos de pós-graduação, observou-se que o tempo aberto para a postagem era superior na maioria das vezes do que o tempo das atividades dos cursos de nível superior e foi nesse nível que teve maior incidência de postagem sendo realizadas no final do prazo de postagem ou após o término da mesma.

Com as análises realizadas, observa-se de maneira geral que nas atividades com um prazo menor, tem-se um índice maior de postagem na abertura para postagem.

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     periododeeentrega-weka.filters.unsupervised.attribute.Remove-R1,3-4,6
Instances:    677
Attributes:   3
              classe_periodo_entrega
              situação
              nivel_curso
Test mode:    evaluate on training data

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

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

classe_periodo_entrega = A
|   situação = Classe_1: Superior (123.0/26.0)
|   situação = Classe_2: Superior (60.0/8.0)
|   situação = Classe_3: Técnico (50.0/24.0)
|   situação = Classe_4: Superior (14.0/3.0)
|   situação = Classe_5: Técnico (46.0/26.0)
|   situação = Classe_6: Superior (37.0/16.0)
classe_periodo_entrega = B: Superior (178.0/18.0)
classe_periodo_entrega = C: Superior (81.0/37.0)
classe_periodo_entrega = D
|   situação = Classe_1: Superior (24.0/11.0)
|   situação = Classe_2: Pós (6.0/2.0)
|   situação = Classe_3: Técnico (3.0/1.0)
|   situação = Classe_4: Pós (18.0/1.0)
|   situação = Classe_5: Técnico (4.0/1.0)
|   situação = Classe_6: Superior (2.0)
classe_periodo_entrega = E
|   situação = Classe_1: Superior (10.0/5.0)
|   situação = Classe_2: Superior (0.0)
|   situação = Classe_3: Pós (1.0)
|   situação = Classe_4: Pós (1.0)
|   situação = Classe_5: Superior (15.0)
|   situação = Classe_6: Pós (4.0)
```

Figura 3.20: Classificação Algoritmo J48.

```
Number of Leaves  :     20

Size of the tree :      24


Time taken to build model: 0 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances         498               73.5598 %
Incorrectly Classified Instances       179               26.4402 %
Kappa statistic                          0.3829
Mean absolute error                      0.1766
Root mean squared error                  0.2971
Relative absolute error                 70.0039 %
Root relative squared error             83.7882 %
Total Number of Instances              677

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.927     0.554      0.772     0.927     0.843      0.811    Superior
                 0.365     0.005      0.9       0.365     0.519      0.956    Pós
                 0.395     0.095      0.495     0.395     0.44       0.746    Técnico
                 0         0          0         0         0          0.975    Formação
Weighted Avg.    0.736     0.389      0.709     0.736     0.704      0.82
```

Figura 3.21: Classificação Algoritmo J48.

```
=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 413  12  27   1 |   a = Superior
  47  25   2   0 |   b = Pos
  84   6  35   4 |   c = Tecnico
   2   0  17   2 |   d = Formação
```

Figura 3.22: Classificação Algoritmo J48.

4 PROPOSTA DE INTEGRAÇÃO DE MINERAÇÃO DE DADOS SIMPLIFICADA COM O MOODLE

Com base na experiência adquirida no estudo de caso, propôr-se uma forma de integrar ao Moodle as atividades de mineração usando Weka, mantendo o foco na análise de prazos de entrega de tarefas. O objetivo da integração é facilitar a repetição dos experimentos realizados usando outros dados, tornando esta funcionalidade de mineração acessível na interface do Moodle.

Este capítulo descreve a proposta de integração de mineração de dados ao Moodle. Nesta a mineração é vista como uma atividade simplificada, restringindo-se ao conjunto de dados gerados pela atividade tarefa do Moodle e aos algoritmos que produziram resultados relevantes para este tipo de análise. Parte desta proposta foi implementada e seu desenvolvimento é descrito neste capítulo.

4.1 Visão Geral da Proposta

A Figura 4.1 apresenta um diagrama que fornece uma visão geral da proposta de integração. Tudo tem início nos dados gerados pela interação dos usuários com o AVA Moodle. Tais dados são armazenados na base de dados e precisam ser selecionados e pré-processados, gerando um arquivo de entrada para o Weka (formato ARFF). No presente trabalho, esta etapa foi feita manualmente, mas é possível automatizá-la através de *scripts* que seguem os passos descritos no estudo de caso. A proposta é que esses *scripts* sejam invocados de forma transparente via interface do Moodle.

Dispondo do arquivo ARFF, é dado a entrada dos dados para algoritmos do Weka, invocados programaticamente via interface do Moodle. Neste ponto, para que a integração seja possível, é necessário um mecanismo para interoperabilidade entre a linguagem usada no Moodle (PHP) e a linguagem usada no Weka (Java). Em particular, é necessário invocar, por uma interface Web, um programa Java que executa algoritmos do Weka. Uma vez executado o algoritmo, seus resultados devem ser apresentados de forma simplificada na interface do Moodle.

No presente trabalho, devido ao tempo disponível, concentrou-se apenas na parte central da proposta, que consiste num mecanismo com interface Web integrada ao Moodle para execução de algoritmos do Weka sobre um arquivo ARFF previamente gerado. A este mecanismo, deu-se o nome de ferramenta de Mineração de Dados Simplificada - MDS. No estudo de caso

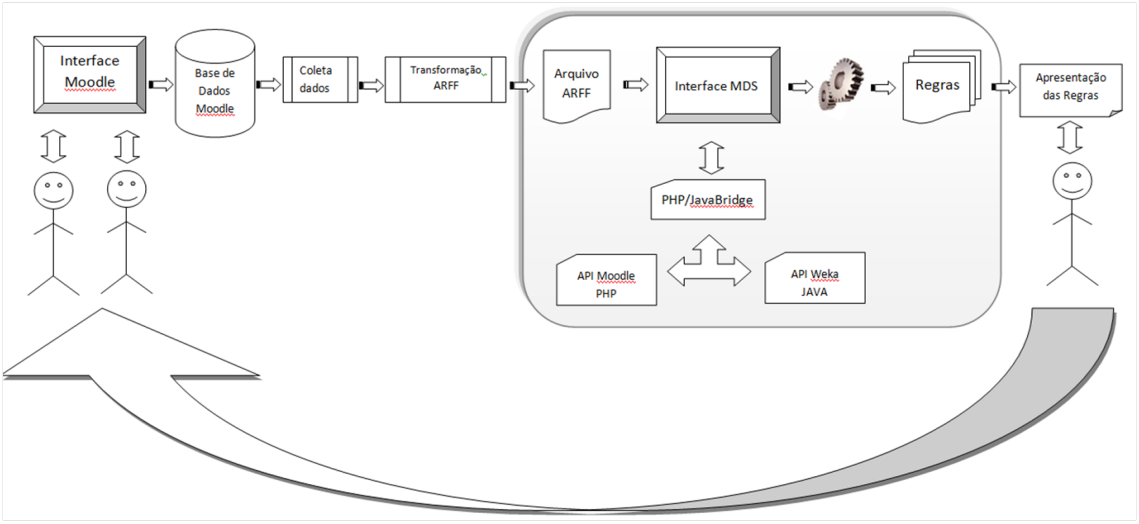


Figura 4.1: Proposta de Integração de Mineração de Dados Simplificada com o Moodle.

apresentado, estão presentes todos os passos necessários para implementação da parte inicial da proposta (geração do ARFF). Quanto à implementação da parte final da proposta (apresentação dos dados), descreve-se uma solução rudimentar, que pode ser alvo de trabalhos futuros visando seu aprimoramento.

4.2 Ferramenta de Mineração de Dados Simplificada - MDS

A ferramenta recebe com entrada um conjunto de dados gerados referentes às atividades de ensino-aprendizagem no Moodle, concentrando-se especificamente em variáveis relativas aos prazos de postagem da tarefa atividade. Estes dados são recebidos no formato ARFF, já descrito anteriormente.

O desenvolvimento da ferramenta envolveu a utilização das seguintes tecnologias:

- Java, como plataforma de implementação dos algoritmos de Mineração de Dados, em função de sua compatibilidade com a interface de programação de aplicativos (API) do Weka;
- PHP, como plataforma de comunicação entre os algoritmos de Mineração de Dados e o Moodle, já que o último é também desenvolvido com essa linguagem de programação; e,
- JavaBridge, como uma arquitetura híbrida de passagem de informações que emprega um mecanismo Java representado em plataforma PHP, e funciona como interface entre as aplicações da ferramenta proposta.

O trabalho com duas linguagens diferentes possibilitou implementar o modelo de uma ferramenta Mineração de Dados Simplificada (MDS), que é a parte central da proposta que auxilia aos professores responsáveis por um curso, em ambiente Moodle, a identificar os possíveis atrasos nas postagem das tarefas. A implementação da ferramenta de Mineração de Dados Simplificada, é apresentada na parte em destaque da Figura 4.1.

A interface MDS foi implementada em linguagem PHP com chamadas à API do Moodle. Nela, o usuário seleciona o arquivo contendo o conjunto de dados a ser minerado, e define o algoritmo para definir as regras de conhecimento a se extrair dos dados. Realizadas essas configurações, a ativação da interface executa os algoritmos do Weka sobre os dados do Moodle e mostra então os resultados.

A linguagem Java serve para implementar as tarefas de Mineração de Dados, utilizando os algoritmos da ferramenta Weka através de sua API. Para desenvolver tal ferramenta ligando arquiteturas de sistema com linguagens de programação diferentes, é necessário o uso de um componente que faça a passagem de informações entre essas arquiteturas. Para tal, será utilizada a ferramenta JavaBridge, cuja estrutura de funcionamento é representada na Figura 4.2.

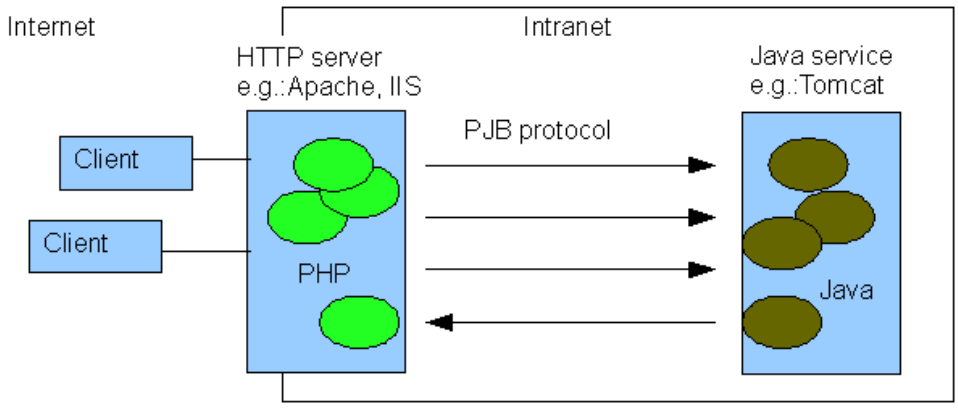


Figura 4.2: Arquitetura do PHP/JavaBridge (PHP/JAVA BRIDGE, 2011).

O JavaBridge funciona com uma ponte entre PHP e Java, conectando um mecanismo de *script* nativa, o PHP, com uma máquina virtual Java. As transações de passagem de dados são feitas a partir de requisições de dados de plataformas clientes, onde se tem um *front-end* PHP associado a um *back-end* Java. Nesse contexto, o servidor de HTTP (PHP) repassa dados para o serviço Java, que os processa e entrega de volta ao servidor para o repasse final às estações clientes.

4.2.1 Funcionamento da Ferramenta MDS

Todo o processo de Mineração de Dados até a obtenção das regras com a ferramenta MDS pode ser observado a partir da arquitetura do sistema, representada na Figura 4.1, e para o caso de uso da ferramenta representado na Figura 4.3. Os casos definem o processo de utilização da Ferramenta MDS em conjunto com os ambientes de aprendizagem e de Mineração de Dados. Basicamente, o uso consiste em selecionar o conjunto de dados desejado, traduzir para o formato do Weka (ARFF) e executar o algoritmo pertinente.

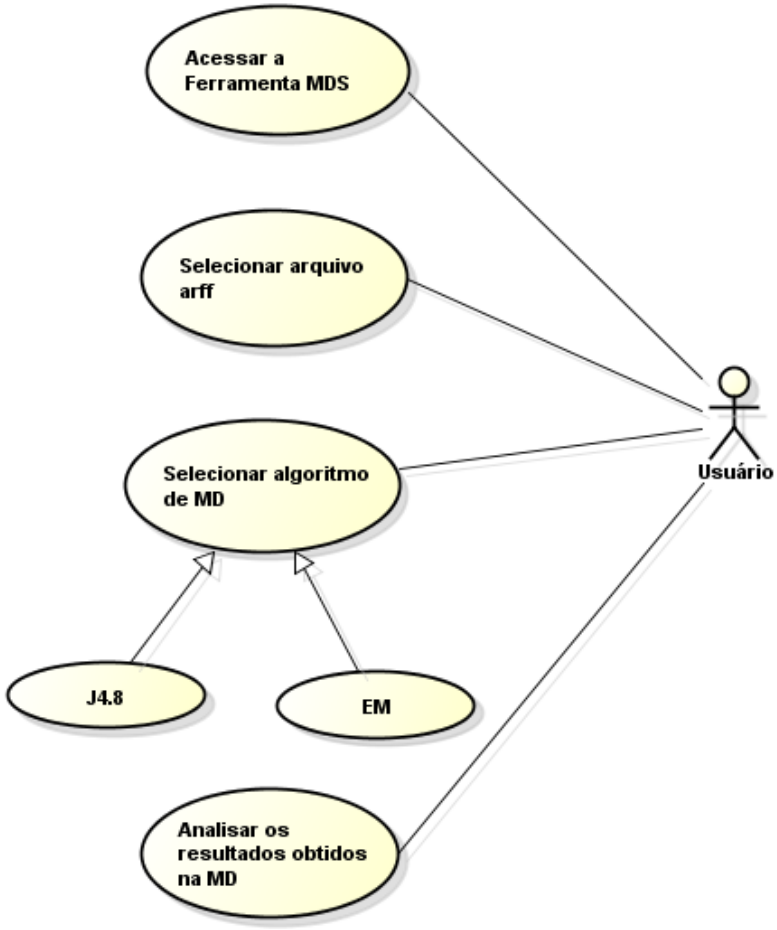


Figura 4.3: Caso de Uso da Ferramenta.

Na Figura 4.4 observa-se como ocorrem os fluxos de atividades na ferramenta. Na presente implementação, a abordagem requer que o usuário traduza o conjunto de dados desejado em formato ARFF, e selecione o algoritmo (J4.8 ou EM) cujo processamento melhor se ajusta à sua necessidade de descoberta de conhecimento. A Figura 4.5 é a tela principal da ferramenta. Nela ocorre a inclusão dos dados, seleção do algoritmo de MD, a execução, visualização dos resultados. Para a seleção do conjunto de dados usa-se o botão "Gerenciar arquivos"presente

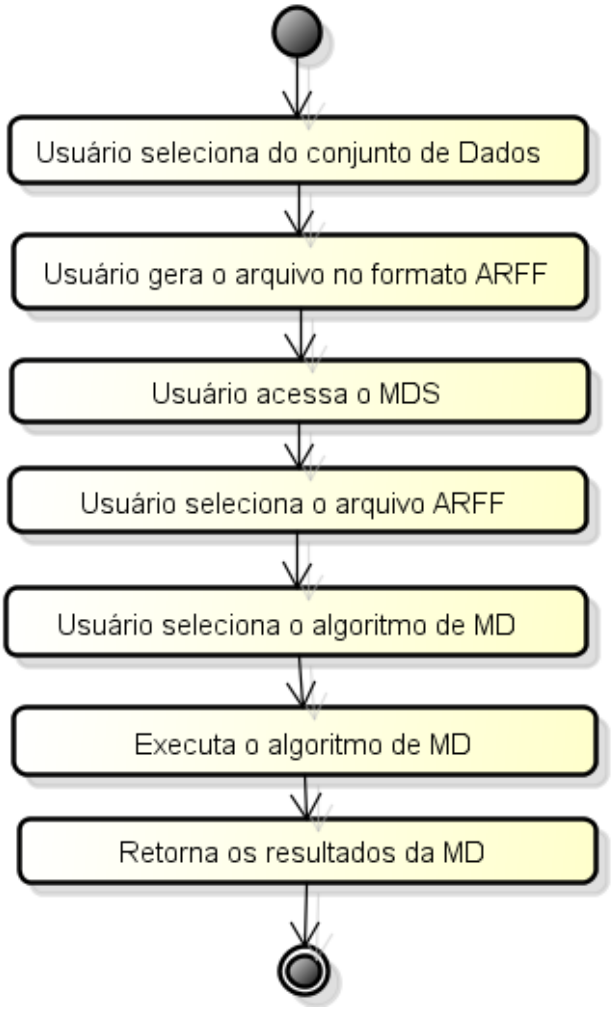


Figura 4.4: Diagrama de Atividade

na tela inicial, que apresentará a tela da Figura 4.6. Esta, por sua vez, apresentará as bases de dados já carregadas e a possibilidade de selecionar uma nova base de dados (Figura 4.7). Posterior à definição do caminho dos dados, retorna-se para a tela inicial para proceder com a seleção do algoritmo, conforme Figura 4.8. As opções de algoritmo são: J4.8, baseado na tarefa de Classificação e *Expectation Maximization* (EM) baseado na tarefa de Agrupamento.

Enquanto o primeiro separa diferentes tipos de informação considerando a 'distância' entre informações individuais, o segundo se baseia na proximidade entre informações para gerar grupos baseado na proximidade dessas mesmas informações no conjunto de dados. O resultado da execução dos algoritmos é apresentado no formato textual, conforme ilustra, mais adiante, a Figura 4.10.

Figura 4.5: Interface Inicial da Ferramenta MDS.

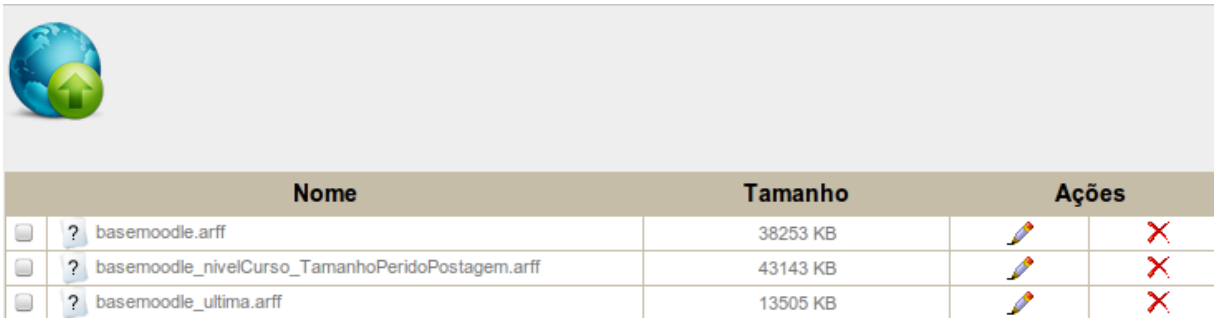


Figura 4.6: Interface para inclusão do arquivo ARFF.

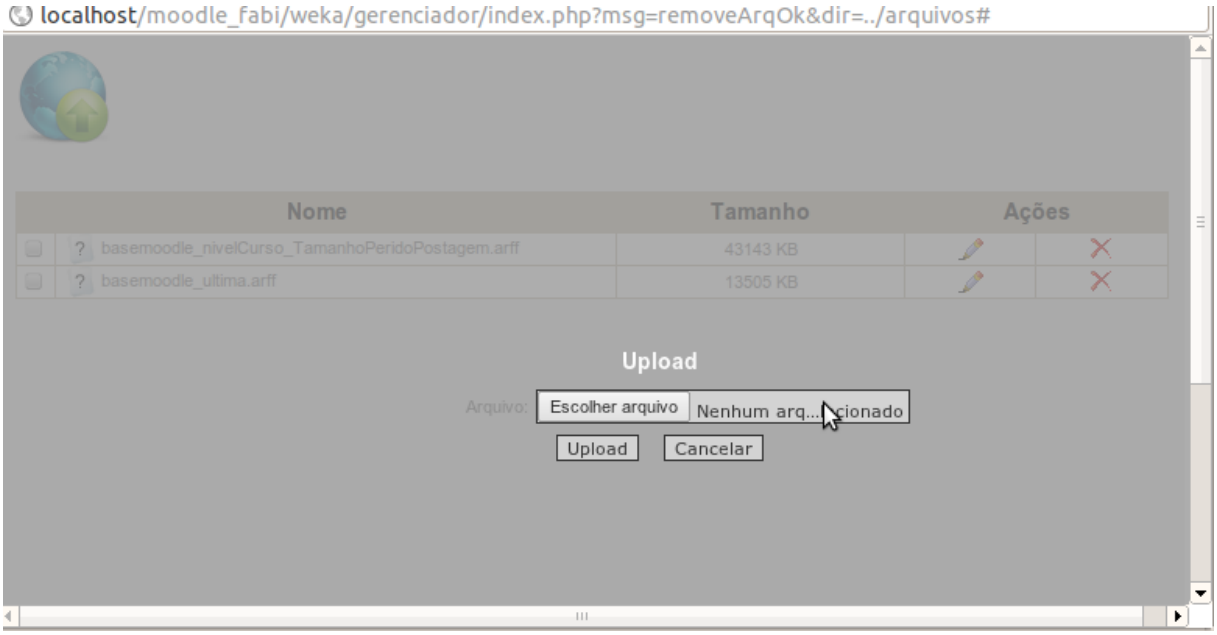


Figura 4.7: Interface de seleção do arquivo ARFF.

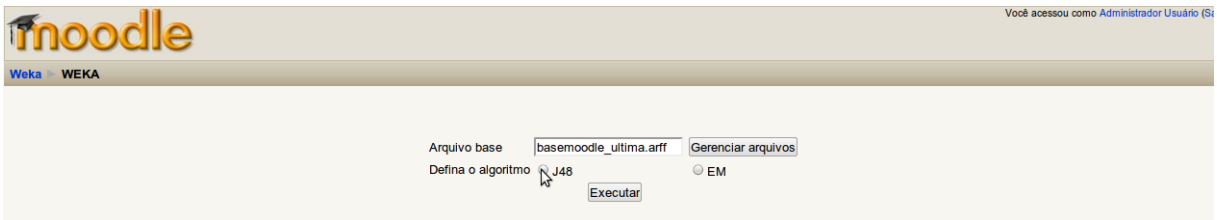


Figura 4.8: Interface de seleção do Algoritmo de MD.

4.2.2 Resultados

Na tela de resultados, é possível navegar pelos mesmos, selecionar outra base de dados ou algoritmo. Para explicitar a ação da ferramenta, presenta-se parte dos códigos utilizados na implementação dos algoritmos e o seu resultado. Utiliza-se os mesmos dados do estudo de caso, obtendo-se resultados que, antes, foram obtidos operando a ferramenta Weka manualmente, usando sua interface gráfica.

4.2.2.1 Algoritmo J4.8.

Na figura 4.9 é visualizado parte do resultado corresponde a execução do algoritmo J4.8.

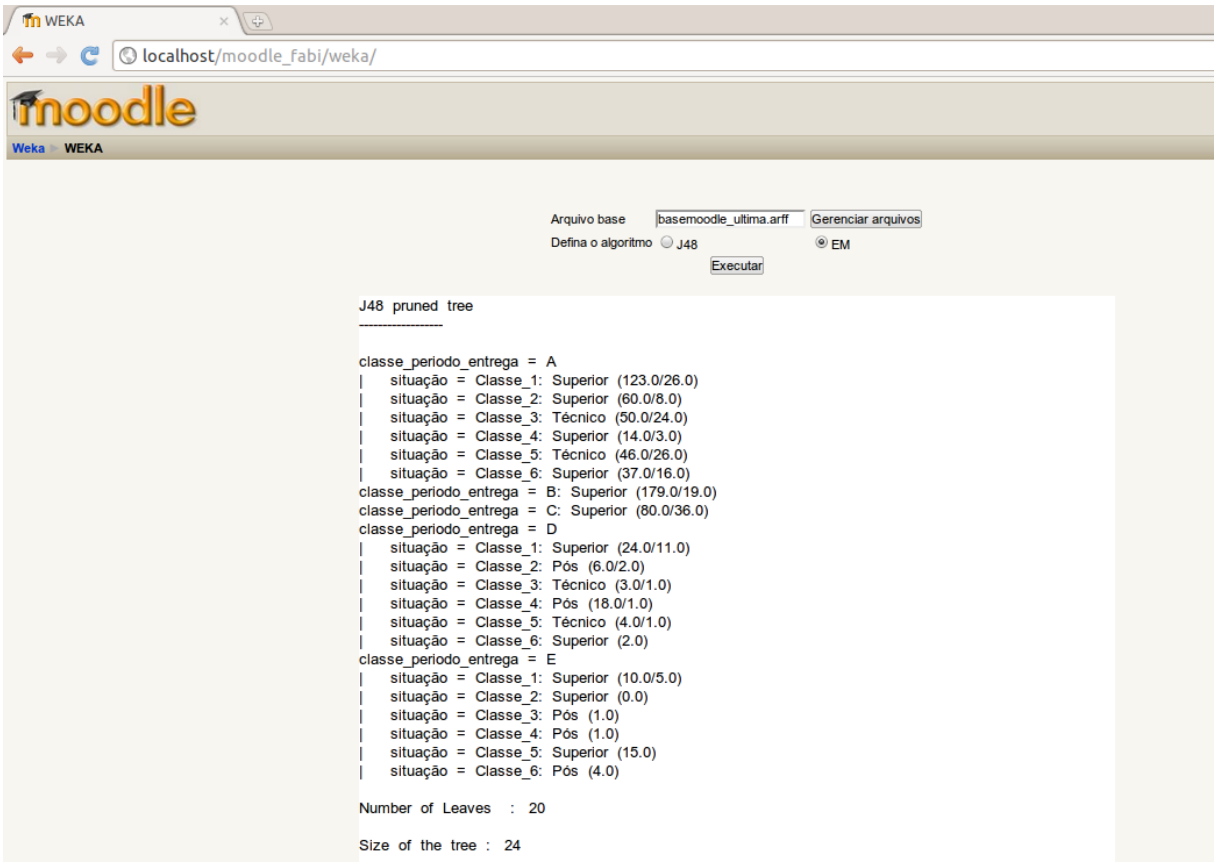


Figura 4.9: Regras Geradas pelo Algoritmo J4.8 da ferramenta MDS.

Sendo o resultado da execução dessa algoritmo as informações que seguem:

- J48 pruned tree, indicando que a árvore foi podada;
- Lista com as regras e os valores para os atributos, classificados corretamente e os não classificados corretamente;
- *Number of leaves*, que são os números de níveis da árvore;

- *Size of the tree*, que representa o tamanho da árvore;
- *Correctly Classified Instances*, que representa o número de instâncias corretamente classificadas;
- *Incorrectly Classified Instances*, que representa o número de instâncias que não foram corretamente classificadas;
- *Kappa Statistic*, que exibe o resultado da Kappa estatística;
- *Mean Absolute Error*, que corresponde ao cálculo do erro médio absoluto;
- *Root Mean Squared Error*, que apresenta o resultado da raiz quadrada do erro médio;
- *Total Number of Instancers*, número total de instâncias.

Parte do código responsável pela geração do resultado da execução do algoritmo J4.8, é apresentado a seguir:

```
J48 thisClassifier = new J48();
thisClassifier.buildClassifier(instância);
variavel = thisClassifier.toString();
J48 tree = new J48();
tree.buildClassifier(dados);
saida = tree.toString();
Evaluation avaliacao;
avaliacao = new Evaluation(dados);
avaliacao.evaluateModel(tree, dados);
String strSummary = avaliacao.toSummaryString();
saida += "\n Sumário:\n" + strSummary;
saida += "Detailed Accuracy By Class:\n";
avaliacao = new Evaluation(dados);
avaliacao.evaluateModel(tree, dados);
saida += avaliacao.toClassDetailsString();
saida += "Avaliacao cruzada: \n";
Evaluation avalCruzada;
avalCruzada = new Evaluation(dados);
```

```
avalCruzada.crossValidateModel(tree, dados, 10, new Random(1));
saida += avalCruzada.toMatrixString();
```

Primeiramente, é criada uma nova instância da classe árvore J4.8; o passo seguinte consiste na invocação do método que realiza o treinamento e passa por parâmetro o conjunto de dados; então, vem o comando para realizar a impressão das regras já geradas; com a chamada ao método que avalia as regras geradas no treinamento;é impresso o sumário. Posteriormente ocorre a busca de detalhes da avaliação realizada nas regras geradas no treinamento. Por último é chamado o método que realiza a avaliação cruzada e determina que a mesma seja realizada 10 vezes; então é realizada a impressão da matriz de confusão gerada.

4.2.2.2 Algoritmo EM.

O resultado da execução do algoritmo EM pode ser observado na Figura 4.10, sendo as informações deste resultado as seguintes:

- Os clusters criados relacionando os atributos do conjunto de dados;
- É apresentado também o tamanho em porcentagem de cada cluster.

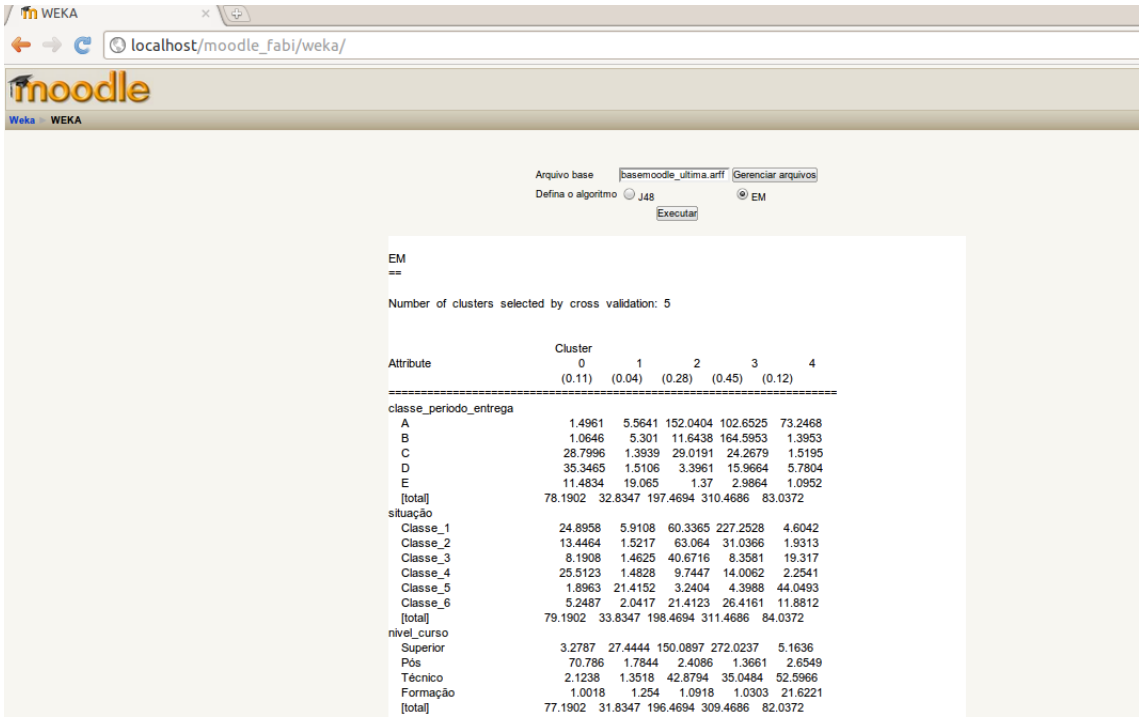


Figura 4.10: Resultados Gerados pelo Algoritmo EM da ferramenta MDS.

- Na primeira linha um novo objeto EM é criado e atribuído a Clusterer;

- Na segunda linha é chamado o método *buildClusterer* de Clusterer, com parâmetro o conjunto de dados carregado para a MD, o qual gera cluster;
- Na terceira linha um novo objeto avaliação de Cluster é criado e atribuído à avaliação;
- Na quarta linha é chamado o método *setClusterer* de avaliação, com o parâmetro o objeto Clusterer, cluster de avaliação;
- Na quinta linha é chamado o método *setClusterer* de avaliação, com parâmetro o conjunto de dados carregado para a MD, ou seja os dados para avaliar o clusterer;
- Comando para realizar a impressão dos resultados já gerados.

O trecho de código fonte seguinte é responsável pela geração do resultado apresentado anteriormente.

```
EM clusterer = new EM();
clusterer.buildClusterer(dados);
ClusterEvaluation avaliacao = new ClusterEvaluation();
avaliacao.setClusterer(clusterer);
avaliacao.evaluateClusterer(dados);
variavel = avaliacao.clusterResultsToString();
```

As regras encontradas na atividade de MD devem ser transformadas para um formato simplificado, evitando a apresentação dos resultados diretamente conforme obtidos, a fim de possibilitar o entendimento das mesmas, por usuários que não dominam a área de KDD. Outra maneira de expressar os resultados é graficamente. Para tal, é interessante relacionar os atributos, período em que a atividade permaneceu aberta, nível de ensino e situação da postagem.

5 CONCLUSÃO

A utilização do processo de KDD é bastante difundida, mas pode ser observado ainda dificuldades no entendimento das etapas para chegar ao conhecimento, bem como, na interpretação das regras geradas pelas ferramentas de Mineração de Dados. Sendo esse o motivo para o qual a proposta de integração sugerida, apresenta a visualização dos resultados em uma maneira mais clara, onde os relacionamentos entre os dados gerados pelas atividades operacionais podem ser interpretados sem conhecimento prévio do funcionamento da ferramenta e suas estatísticas.

Apesar da proposta de integração ser apresentado ao usuário de maneira simplificada, isso não anula a exigência sobre o domínio da área de negócio e o conhecimento das técnicas e etapas de KDD são úteis para contribuir no processo de tomada de decisão.

A utilização do processo de KDD está relacionado a diferentes áreas de atuação, podendo ser aplicada a conjunto de dados gerados por empresas, instituições de ensino, profissionais liberais, etc, que buscam a descoberta de conhecimento sobre um determinado conjunto de dados para auxiliar na tomada de decisão.

O estudo realizado teve como ambiente gerador dos dados o AVA Moodle, que é um excelente ambiente facilitador da comunicação entre os envolvidos no processo pedagógico, nos cursos à distância e nos presenciais. No estudo realizado observa-se a importância da utilização de ferramentas que gerem informações aos docentes acerca da trajetória do aluno, para que o mesmo possa elaborar estratégias pedagógicas que atendam as necessidades pedagógicas individuais de cada aluno ou grupo.

As principais contribuições desta dissertação são:

- Identificação de quais tarefas de MD são adequadas para o conjunto de dados em estudo, sendo selecionadas a classificação e o agrupamento;
- Identificação dos algoritmos das tarefas selecionadas que geram regras melhores para o conjunto de dados em estudo. Como resultado chegamos aos algoritmos EM e J4.8;
- Descoberta de conhecimento com base nos dados gerados pela atividade tarefa do Moodle, utilizando os algoritmos de Mineração de Dados EM e J4.8 da ferramenta WEKA;
- Proposta de integração de atividades de Mineração de Dados ao Moodle, de forma simplificada e automática para o usuário;

- Implementação de parte da proposta de integração sugerida, que corresponde a tarefa de Mineração de Dados simplificada;

5.1 Trabalhos Futuros

Como trabalhos futuros, podem ser sugeridas as seguintes opções:

- Implementação do restante da proposta sugerida;
- Agregação da ferramenta MDS a implementação;
- Repetição de novos experimentos usando dados gerados por outras Instituições de Ensino;
- A inclusão de outros dados para investigar as relações com a postagem das tarefas e o aproveitamento do aluno;
- Também a análise dos dados da participação e o perfil do aluno podem gerar regras importantes se relacionadas com a postagem da tarefa e aproveitamento;

REFERÊNCIAS

A. P. A. BATISTA, G. E. de. **Pré-processamento em aprendizado de máquina supervisionado**. 2003. Tese (Doutorado) — USP.

BADIU. **GMoodle**. Acessado em Dez/2011, http://www.badiu.net.

BARUQUE, C. B. et al. Analysing users' access logs in Moodle to improve e learning. In: EURO AMERICAN CONFERENCE ON TELEMATICS AND INFORMATION SYSTEMS, 2007. **Anais...** [S.l.: s.n.], 2007.

BOENTE, A. **Descoberta de Conhecimento em Bases de Dados**. 2006. Tese (Doutorado) — AWU – American World University.

BUTZEN, E. **PROPOSTA DE UM MÓDULO DE DATA MINING PARA SISTEMA DE SCOUT NO VOLEIBOL**. [S.l.]: Centro Universitário Feevale Instituto de Ciências Exatas e Tecnológicas, 2008.

Datamining A Mineração de Dados no Marketing, Medicina, Economia, Engenharia e Administração. [S.l.]: Editora Érica Ltda, 2001.

CASTANHEIRA, L. G. **Aplicação de Técnicas de Mineração de Dados em Problemas de Classificação de Padrões**. 2008. Dissertação (Mestrado) — PPGEE/UFMG.

DIAS, M. M. et al. Aplicação de Técnicas de Mineração de dados no Processo de Aprendizagem na Educação a Distância. In: XIX SIMPóSIO BRASILEIRO DE INFORMáTICA NA EDUCAçãO, 2008. **Anais...** SBC, 2008.

DILLY, R. **Data Mining - an introduction. Parallel Computer Centre - Queen's**. [S.l.]: University of Belfast, 1995.

FAGUNDES, L. et al. Projetos de Aprendizagem – Uma experiência mediada por ambientes Telemáticos. In: RBIE, 2006, http://www.br-ie.org/pub/index.php/rbie/article/view/37/31. **Anais...** [S.l.: s.n.], 2006.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge Discovery and Data Mining: towards a unifying framework. In: SECOND INTERNATIONAL CONFERENCE ON KD & DM, 1996. **Anais...** [S.l.: s.n.], 1996.

A survey of data mining and knowledge discovery software tools. [S.l.]: SIGKDD Explorations, 1999.

Data mining: um guia prático. Segunda.ed. [S.l.]: Elsevier, 2005.

Intranet data warehouse. [S.l.]: Berkeley, Editora, 1998.

HOLSHEIMER, M.; KERSTEN, M.; MANNILA, H.; TOIVONEN, H. A perspective on databases and data mining. In: FIRST INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 1996. **Anais...** [S.l.: s.n.], 1996. p.447–46.

KAMPFF, A. J. C. **Mineração de Dados Educacionais para Geração de Alertas em Ambientes Virtuais de Aprendizagem como Apoio à Prática Docente**. 2009. Tese (Doutorado) — PPGIE/UFRGS.

MOODLE. **Documentação Estatísticas Moodle**. Acessado em Out/2011, http://docs.moodle.org/22/en/Statistics.

PHP/JAVA Bridge. Acessado em Out/2011, http://php-java-bridge.sourceforge.net/pjb/.

Transformative and Self-Directed Learning in Practice. [S.l.]: New Directions for Adult and Continuing Education, 1997.

PRADO LIMA, M. de Fátima W. do; WEBBER, C. G.; GUIMARÃES, B. B. ESTUDO DO DESENVOLVIMENTO COGNITIVO INDIVIDUAL E DE GRUPOS ATRAVÉS DA ANÁLISE AUTOMÁTICA. In: CINTED-UFRGS, 2011. **Anais...** Novas Tecnologias na Educação CINTED-UFRGS, 2011.

ROMERO, C.; VENTURA, S.; GARCÍA, E. **Data mining in course management systems**: moodle case study and tutorial. [S.l.]: Elsevier Science, 2007.

SANTOS, R. **Um guia para uso do Weka em scripts e integração com aplicações em Java**. http://www.lac.inpe.br/ rafael.santos/Docs/CAP359/2005/weka.pdf: Instituto Nacional de Pesquisas Espaciais – INPE, 2005.

SANTOS SILVA, M. P. dos. Mineração de Dados - Conceitos, Aplicações e Experimentos com Weka. In: IV ESCOLA REGIONAL DE INFORMÁTICA, 2004. **Anais...** [S.l.: s.n.], 2004.

SILVA, M. P. S. **Uma Linguagem Declarativa de Especificação de Consultas e Processos para Descoberta de Conhecimento em Bancos de Dados e sua Implementação**. 2002. Dissertação (Mestrado) — Universidade Federal de Pernambuco.

VIEIRA, L. A. **FERRAMENTAS PARA ESTIMAR VALORES FALTANTES EM UMA BASE DE DADOS NA ETAPA DE PRÉ-PROCESSAMENTO DE UM KDD**. [S.l.]: Universidade do Vale do Itajaí, 2008.

WEKA. **Data Mining Software in Java**. Acessado em Jul/2011, Disponível em http://www.cs.waikato.ac.nz/ml/weka.