UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

MARAÍSA DA SILVA GUERRA ASSEISS

APLICAÇÃO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS ACADÊMICO UTILIZANDO AS TAREFAS DE AGRUPAMENTO E CLASSIFICAÇÃO

Ilha Solteira
2017

MARAÍSA DA SILVA GUERRA ASSEISS

APLICAÇÃO DO PROCESSO DE DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS ACADÊMICO UTILIZANDO AS TAREFAS DE AGRUPAMENTO E CLASSIFICAÇÃO

Dissertação apresentada à Faculdade de Engenharia do Câmpus de Ilha Solteira - UNESP, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.
Especialidade: Automação.

Prof. Dr. Alexandre Cesar Rodrigues da Silva

Orientador

Ilha Solteira

2017

FICHA CATALOGRÁFICA
Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

Asseiss, Maraísa da Silva Guerra.
A844a Aplicação do processo de descoberta de conhecimento em banco de dados acadêmico utilizando as tarefas de agrupamento e classificação / Maraísa da Silva Guerra Asseiss. -- Ilha Solteira: [s.n.], 2017
115 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia . Área de conhecimento: Automação , 2017

Orientador: Alexandre Cesar Rodrigues da Silva
Inclui bibliografia

1. Agrupamento. 2. Classificação. 3. Descoberta de conhecimento. 4. Data mining. 5. Weka.

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Ilha Solteira

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Aplicação do Processo de Descoberta de Conhecimento em Banco de Dados Acadêmico Utilizando as Tarefas de Agrupamento e Classificação

AUTORA: MARAÍSA DA SILVA GUERRA
ORIENTADOR: ALEXANDRE CESAR RODRIGUES DA SILVA

Aprovada como parte das exigências para obtenção do Título de Mestra em ENGENHARIA ELÉTRICA, área: AUTOMAÇÃO pela Comissão Examinadora:

Prof. Dr. SERGIO AZEVEDO DE OLIVEIRA
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira

Profa. Dra. CHRISTIANE MARIE SCHWEITZER
Departamento de Matemática / Faculdade de Engenharia de Ilha Solteira

Prof. Dr. TÉRCIO ALBERTO DOS SANTOS FILHO
Departamento de Ciências da Computação / Universidade Federal de Goiás

Ilha Solteira, 30 de junho de 2017

AGRADECIMENTOS

A Deus, por estar presente em todos os momentos de minha vida.

Ao Prof. Dr. *Sérgio Azevedo de Oliveira* pela sabedoria, paciência, dedicação e serenidade com que me orientou nesse trabalho. Muito obrigada, professor.

Ao meu amado esposo *Habib Asseiss Neto* pelo apoio, paciência e companheirismo durante as etapas mais críticas desse trabalho, pelo amor a mim dispensado diariamente e por ser um marido maravilhoso.

Aos meus queridos pais *Aparecido Guerra* e *Maura da Silva*, pelas orações em meu favor e por sempre acreditarem e confiarem em mim.

À banca examinadora pelas correções, ideias e sugestões apresentadas.

A todos os amigos que, de alguma forma, me incetivaram a concluir esse trabalho.

Sinceramente,
Maraísa da Silva Guerra Asseiss

RESUMO

Nos últimos anos a quantidade de dados armazenados diarimente em empresas e instituições aumentou consideravelmente e um dos motivos que contribuiu para isso é a crescente importância dada à informação. De forma geral, esses dados são meramente armazenados e, portanto, subutilizados pelos seus detentores, enquanto poderiam ser estudados a fim de obter novos conhecimentos, informações e relacionamentos. Neste contexto, surge o processo de descoberta de conhecimento em banco de dados. Este trabalho apresenta uma introdução a banco de dados, uma revisão bibliográfica sobre o processo de descoberta de conhecimento em banco de dados, a descrição de cada etapa deste processo, uma explanação sobre as tarefas de agrupamento e classificação, além de resumir brevemente as técnicas de particionamento e árvore de decisão. É exposto um estudo sobre o sistema Weka, em que apresenta-se conceitos, funcionalidades e exemplifica-se diversas formas de utilização do sistema. O objetivo principal deste trabalho é propor uma metodologia para descoberta de novos conhecimentos em bancos de dados acadêmicos baseada no processo de descoberta de conhecimento em banco de dados, sendo esta uma metodologia mais simplificada e de execução mais direcionada. Como parte da metodologia este trabalho contribui ainda com uma aplicação desenvolvida em Python como forma de apoio a etapas da metodologia. A metodologia proposta conta com a ferramenta Weka para execução dos algoritmos de *data mining* e prevê a execução das tarefas de agrupamento e classificação. Por fim o trabalho retrata dois estudos de caso envolvendo bancos de dados acadêmicos reais e a execução de todas as etapas da metodologia proposta, com a utilização do sistema Weka. Os estudos de caso abordam as tarefas de agrupamento e classificação e as técnicas de particionamento e árvores de decisão, com a utilização dos algoritmos *SimpleKMeans* e J4.8, respectivamente. Os resultados obtidos através dos estudos mostram que a metodologia proposta é capaz de gerar conhecimentos novos e úteis, tanto na análise de dados de desempenho acadêmico quanto na análise de dados socioeconômicos dos alunos.

Palavras-chave: Agrupamento. Classificação. Descoberta de conhecimento. Estudo de caso. Metodologia. *Data mining*. Weka.

ABSTRACT

In the past years the amount of data stored daily in companies increased considerably and one of the reasons that contributed to this fact is the increasing importance given to information. In general these data are merely stored and therefore underused by its owners, while they could be studied in order to find out new knowledge, information and relationship. In this context, the knowledge discovery in database process arises. This work presents an introduction to databases, a bibliographic review about the knowledge discovery in databases process, a description of each step of this process, an explanation about the clustering and classification tasks and the summarization os the partition and decision tree techniques. A study of the Weka system is shown, in wich are presented concepts, functionalities and examples of use forms for the system. The main objective of this work is the proposal of a methodology for knowledge discovery in academic databases based on the KDD process. The presented methodology is a more simplified and directed version of the KDD. As part of the methodology this work also presents an application developed in Python programming language as a support tool for the methodology steps. The presented methodology uses the Weka tool for running the data mining algorithms and considers the clustering and classification tasks. Lastly this work describes two case studies involving real academic databases and the execution of all the steps from the proposted methodology using the Weka system. The case studies addresses the clustering and classification tasks, as well as the partitioning and decision trees techniques, using the *SimpleKMeans* and J4.8 algorithms respectively. The obtained results show that the methodology is capable of generating new and useful knowledge, both by analyzing academic performance data and by analyzing students' socioeconomic data.

Keywords: Case study. Classification. Clustering. Data mining. Knowledge discovery. Methodology. Weka.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Nas últimas décadas tem se observado um grande crescimento nos volumes de dados gerados por pessoas, empresas e instituições através do uso cada vez mais intenso de sistemas computacionais para os mais diversos fins. Na atualidade, basicamente toda informação gerada é armazenada por meios digitais através de sistemas de bancos de dados.

Com a crescente competitividade do mercado, com a globalização e com o avanço da tecnologia, empresas e instituições acumulam, cada vez mais, grandes volumes de dados e tipos diversificados de informação em seus bancos de dados. Como consequência do aumento da informação armazenada, esses bancos de dados representam informações cada vez mais valiosas sobre os negócios dessas entidades (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

No intuito de descobrir sempre mais informações sobre o mercado, sobre as áreas de atuação, sobre os clientes e os negócios de forma geral, surge a necessidade de que as instituições trabalhem em cima dos seus próprios repositórios de dados. Este trabalho é realizado com o objetivo de gerar conhecimento ou buscar por informações não triviais a partir dos dados que estão simplesmente armazenados nos bancos de dados. A possibilidade é que os dados sejam varridos e lapidados a fim de que sejam transformados em conhecimentos valiosos.

À busca por informações em banco de dados é dado o nome de Processo de Descoberta de Conhecimentos em Banco de Dados (do inglês, *Knowledge Discovery in Database* - KDD). O processo de KDD é composto por cinco etapas, sendo elas: etapa de seleção de dados, etapa de pré-processamento dos dados, etapa de transformação de dados, etapa de *data mining* e etapa de interpretação e avaliação dos dados. Todas as etapas são importantes, uma vez que preparam os dados e fornecem um procedimento organizado para a descoberta de novos conhecimentos, mas destaca-se a importância da etapa de *data mining*, onde algoritmos específicos são aplicados aos conjuntos de dados, permitindo a detecção de padrões e relacionamentos imprescindíveis para a descoberta de conhecimento.

Data mining, além de ser o nome dado a uma etada do processo de KDD, é também o nome de uma área interdisciplinar da ciência, que desperta o interesse de pesquisadores de diversas outras áreas, onde se procura aplicar algoritmos e técnicas específicas com objetivo de extrair informações e relacionamentos de informações em bancos de dados.

Recentes estudos demonstram a viabilidade da aplicação de técnicas de *data mining* em di-

versas áreas da ciência, como medicina (FERREIRA et al., 2014; ARAUJO; SANTANA; SANTOS NETO, 2015), biologia (SILVA, 2016) e engenharia (PAN; MORRIS; ADHIKARI, 2015). Destaca-se, ainda, uma forte tendência em estudos da área da educação em que se aplicam técnicas de *data mining* com objetivos de oferecer predições envolvendo dados acadêmicos (HEREDIA; AMAYA; BARRIENTOS, 2015). Alguns estudos realizam comparações entre diferentes técnicas de *data mining*, como árvores de decisão e redes neurais (NANDESHWAR; MENZIES; NELSON, 2011), comparações entre algoritmos de árvores de decisão (BARADWAJ; PAL, 2011), utilização de classificação e agrupamento (KTONA; XHAJA; NINKA, 2014), ou ainda a aplicação de algoritmos genéticos para predição de evasão de estudantes (MARQUEZ–VERA et al., 2016).

O objetivo desse trabalho é propor uma metodologia para descoberta de novos conhecimentos em banco de dados acadêmico, utilizando a ferramenta Weka como base para execução dos algoritmos de *data mining*. A metodologia proposta é baseada no processo de KDD e foi idealizada para simplificar e agilizar o processo de busca por novos conhecimentos em banco de dados acadêmicos. Também como contribuição deste foi desenvolvida uma aplicação em Python para dar apoio ao processo que precede a *data mining*. A metodologia prevê a aplicação das tarefas de agrupamento e classificação, além da utilização de uma nova aplicação de apoio a *data mining*.

Por fim, são apresentados dois estudos de caso nos quais a metodologia proposta é aplicada e avaliada. Os estudos de caso têm dentre seus objetivos a obtenção de novas informações ou a detecção de padrões previamente desconhecidos. Nos estudos de caso são utilizados diferentes bancos de dados acadêmicos e são executadas as etapas da nova metodologia proposta, empregando as tarefas de agrupamento e classificação, onde os resultados obtidos são analisados e discutidos em cada caso.

1.1 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado como segue: no Capítulo 2 são apresentados uma introdução a banco de dados e os conceitos e etapas do processo de descoberta de conhecimento em banco de dados. No Capítulo 3 são tratadas teorias e funcionalidades do sistema Weka, além de exemplificar diversas formas de utilização do sistema. No Capítulo 4 é proposta uma nova metodologia para descoberta de novos conhecimentos em banco de dados acadêmico e é apresentada uma aplicação de apoio à *data mining*. No Capítulo 5 são detalhadas as aplicações da metodologia em dois estudos de caso distintos, onde foram utilizados o sistema Weka e dados acadêmicos reais. Finalmente, no Capítulo 6 as conclusões e sugestões para trabalhos futuros

são apresentadas.

2 FUNDAMENTAÇÃO TEÓRICA

Neste Capítulo serão apresentadas as fundamentações teóricas sobre banco de dados e uma breve explanação sobre banco de dados relacional. Também serão fundamentados teoricamente o processo de descoberta de conhecimento em banco de dados e suas respectivas etapas.

2.1 BANCO DE DADOS

Korth e Silberschatz (1994) definem banco de dados como "uma coleção de dados inter-relacionados, representando informações sobre um domínio específico", ou seja, todo grupo de informações que esteja organizado e que faça parte de um mesmo mini mundo pode ser considerado um banco de dados. Para exemplificar pode-se citar uma lista telefônica ou um sistema de controle de pacientes de um consultório odontológico.

2.1.1 O Surgimento do banco de dados

Todos os dias, aumenta-se a quantidade de dados que pessoas e empresas precisam armazenar. Um exemplo disso é que antes uma empresa de grande porte conseguia arquivar seus dados em papéis, em salas especificamente preparadas para isto, e hoje em dia essa é uma tarefa notavelmente difícil pela quantidade de informações que são necessárias armazenar.

Com a globalização e a evolução da tecnologia, a quantidade de dados a ser armazenada aumentou consideravelmente e a população cresceu aceleradamente a partir da década de 60, em torno de 1 bilhão de habitantes a cada década, de acordo com a Organização das Nações Unidas- ONU (2014). Nesse ínterim, a quantidade de informação também cresceu, ampliando, assim, a dimensão dos dados a serem armazenados nos bancos de dados.

Portanto, este crescimento não se deve apenas ao aumento da população, nem tão somente ao crescimento dos dados, mas ambos inseridos no contexto global são os responsáveis pela ampliação da quantidade de informação que é preciso armazenar.

Atualmente, o uso de um banco de dados é essencial para armazenar tanta informação de forma organizada e fácil de ser recuperada, pois computacionalmente a recuperação de dados é significativamente mais rápida.

É possível citar vários benefícios do uso de banco de dados em relação aos métodos tradicionais, baseados em papel. Date (1991) cita alguns exemplos:

- Densidade: não é necessário o uso de papel e são evitados dados redundantes;
- Velocidade: um computador pode obter e atualizar dados com rapidez incomparável ao ser humano;
- Menos trabalho monótono: a maior parte do trabalho entediante de manter arquivos à mão é eliminada. Sem contar que as tarefas repetitivas são sempre realizadas com maior qualidade por máquinas;
- Atualidade: a todo momento é possível obter informações atualizadas e precisas;
- Proteção: os dados podem ser melhor protegidos contra perda e acesso ilegal.

2.1.2 Tipos de banco de dados

O modelo de dados relacional é e o mais popular em uso na atualidade. Sua estrutura é baseada no conceito de tabelas, constituídas por linhas e colunas, que armazenam os dados, e no conceito de relacionamento, que permite uma inter-relação entre as tabelas (ELMASRI; NAVATHE, 2011).

Nos últimos anos, foram introduzidos os denominados bancos de dados NoSQL (*Not only SQL*). Sua principal característica, que contrasta com os bancos de dados relacionais, está na estrutura dos seus dados. Neste modelo, os dados são armazenados em formatos específicos, denominados "documentos" ou "pares chave-valor", que possuem uma estrutura flexível em que não é necessário seguir a estrutura de uma tabela, como no modelo relacional (HAN et al., 2011).

Apesar das características apresentadas pelo modelo de banco de dados NoSQL, como desempenho em grande volume de dados e ampla escalabilidade, sua utilização tem sido adotada em projetos muito recentes. Os bancos de dados relacionais ainda são considerados mais adequados para realização de análises e consultas complexas, devido às restrições de consistências impostas pelo seu formato. Os bancos de dados relacionais ainda têm uma utilização amplamente difundida na indústria e a grande maioria dos sistemas dependem de um banco de dados relacional (NAYAK; PORIYA; POOJARY, 2013). A seguir, são descritas brevemente as características dos tipos de bancos de dados: relacional e NoSQL.

2.1.2.1 Banco de dados relacional

Um banco de dados é uma aplicação que permite armazenar e obter de volta dados com eficiência. O que o torna relacional é a maneira como os dados são armazenados e organizados no banco de dados. Segundo Elmasri e Navathe (2011), este tipo de banco de dados surgiu por volta de 1970 e chamou a atenção das pessoas por ser simples e ter sua base na matemática.

Conforme Date (1991) e Elmasri e Navathe (2011), a linguagem padrão desse tipo de banco de dados é a *Structured Query Language*, mais conhecida como SQL. O termo relacional é aplicado tantos aos dados como ao próprio banco de dados. Foi desenvolvido para facilitar o acesso aos dados e possibilita que os usuários utilizem uma grande variedade de abordagens no tratamento de informações.

Em um banco de dados relacional todos os dados são guardados em uma coleção de linhas e colunas, constituindo tabelas. Estas têm uma estrutura que se repete a cada linha, como se pode observar em uma planilha, conforme apresentado na Figura 1. Porém, elas se relacionam entre si, e dependendo desse relacionamento carregam dados de outras tabelas consigo como referência à tabela que se relaciona. Um exemplo de relacionameto entre as tabelas apresentadas na Figura 1 é a relação entre a disciplina de Algoritmos e o curso de Informática, esta relação é feita através da coluna Curso da tabela Disciplina onde o número 1, da coluna Curso, indica que Algoritmos pertence ao curso 1 que é Informática. Os relacionamentos entre as tabelas que as tornam "relacionais" (ELMASRI; NAVATHE, 2011).

Figura 1 - Banco de dados relacional.

Curso

Código	Nome
1	Informática
2	Eletrotécnica

Disciplina

Código	Nome	Curso
1	Algoritmos	1
2	LógicaDigital	1
3	Eletricidade1	2
4	LógicaDigital	2

Fonte: Autoria própria.

Este tipo de banco de dados revolucionou o mercado desta área e, apesar das novas tendências como as apresentadas pelos bancos de dados NoSQL, o banco de dados relacional é o mais utilizado na atualidade (MORLEY; PARKER, 2014).

2.2 DESCOBERTA DE CONHECIMENTO EM BANCO DE DADOS

Nos últimos anos, com a globalização e a evolução da tecnologia, a quantidade de dados armazenados aumentou consideravelmente. Bancos de dados de grandes empresas recebem diariamente milhares de novas informações e essas são, na maioria das vezes, "jogadas" no repositório de dados e "esquecidas" lá, quando deveriam ser estudadas, analisadas para que esses registros não representem apenas dados mas, sim, conhecimento.

Bancos de dados podem conter informações desconhecidas que podem ser consideradas importantíssimas, visto que, na atualidade, um dos maiores tesouros é a informação. Com o objetivo de revelar tais informações, surge o processo chamado de descoberta de conhecimento em banco de dados (KDD), que em (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) define-se como sendo: "o processo não trivial de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados".

Em (GOLDSCHMIDT; PASSOS, 2005) o processo de KDD é definido como a descoberta de novos conhecimentos, sejam padrões, tendências, relações, associações, probabilidades ou fatos, que não são óbvios ou de fácil identificação.

2.2.1 Etapas do processo de KDD

O processo de KDD é composto por um conjunto de etapas com a finalidade de obter novos conhecimentos a respeito de um determinado domínio, a partir de uma base de dados em estado bruto (FAYYAD; UTHURUSAMY, 2002). O termo *processo* utilizado no KDD implica que este compreende diversos passos ou etapas, as quais envolvem a preparação dos dados, a busca por padrões, a avaliação do resultado e as etapas adicionais intermediárias que têm o objetivos de preparar a transição das etapas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

O KDD é um processo que envolve a utilização de um banco de dados e de procedimentos adicionais que selecionam, pré-processam, realizam amostragem ou transformam os dados. O processo ainda conta com uma etapa denominada *data mining*, que é a aplicação de métodos (algoritmos) que extraem padrões a partir dos dados. Por fim, a avaliação dos produtos obtidos da *data mining* é realizada a fim de identificar se um subconjunto de padrões obtidos podem ser julgados como novos conhecimentos.

O processo de KDD definido por Fayyad, Piatetsky-Shapiro e Smyth (1996) é composto por nove etapas, descritas a seguir.

1. Desenvolver e entender o domínio da aplicação. Esta etapa envolve o estudo para o apren-

dizado de conhecimentos prévios relevantes, além do conhecimento dos objetivos do usuário ao realizar o processo.

2. Criar um conjunto de dados alvo. O usuário deve selecionar um subconjunto de variáveis (atributos) e instâncias (exemplos) que serão utilizadas para realizar o processo de descoberta. Este passo geralmente inclui consultas aos dados existentes para a seleção de um subconjunto desejado.

3. Limpeza de dados e pré-processamento. Nesta etapa deve-se remover valores distorcidos, tratar ruídos e valores ausentes nos dados, além de lidar com informações temporais e mudanças conhecidas pelo usuário do processo.

4. Redução de dados e projeção. Esta etapa consiste da descoberta de atributos úteis ao se aplicar métodos de redução da dimensionalidade dos dados e métodos de transformação dos dados. A etapa tem o objetivo de encontrar as características nos dados que sejam mais representativas ao objetivo do processo.

5. Escolha da tarefa de *data mining*. Nesta etapa o usuário do processo de KDD deve combinar os objetivos definidos na Etapa 1 com um método de *data mining* específico, tal como classificação, regressão, agrupamento, etc.

6. Escolha do algoritmo de *data mining*. O usuário deve selecionar o método de busca por padrões nos dados e decidir quais modelos e parâmetros do método são apropriados.

7. *Data mining*. Esta etapa realiza a busca por padrões de interesse em uma forma de representação específica, ou conjuntos de representação, incluindo regras de classificação, árvores de decisão, agrupamento, etc. O usuário pode auxiliar o método de *data mining* ao executar corretamente os passos anteriores do processo.

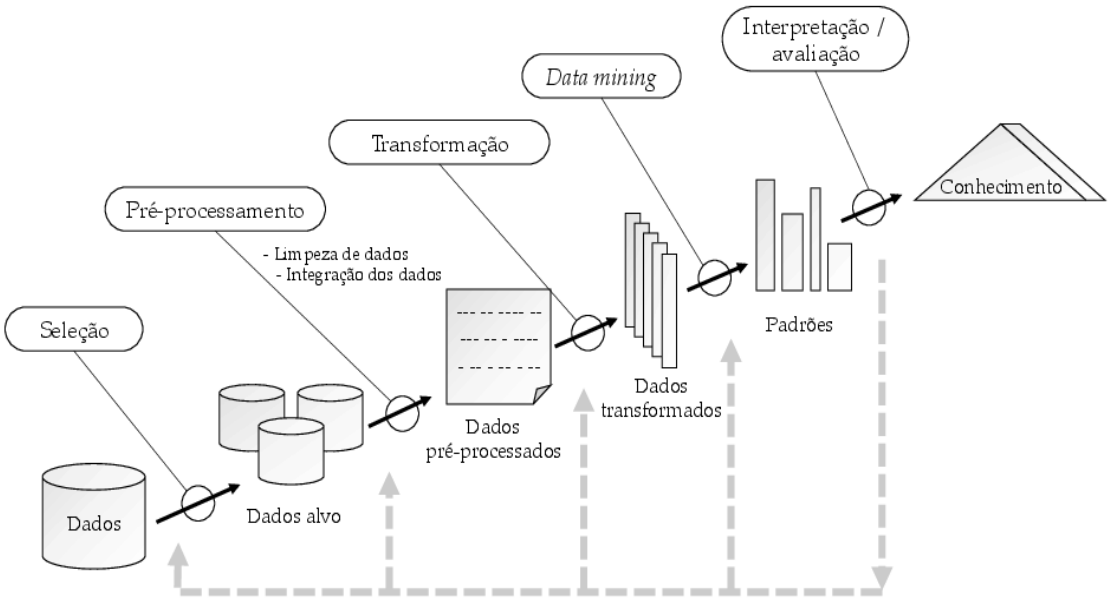
8. Interpretação de padrões minerados. Aqui o usuário visualiza os padrões e modelos extraídos, bem como realiza a visualização dos próprios dados a partir dos modelos extraídos.

9. Consolidação do conhecimento descoberto. A etapa final consiste na utilização direta do conhecimento descoberto, na incorporação do conhecimento em algum outro sistema disponível ou simplesmente na documentação e geração de relatórios para as partes interessadas. Esta etapa também pode envolver a resolução de potenciais conflitos das novas informações obtidas em relação a conhecimentos prévios.

Geralmente uma etapa está relacionada diretamente com sua etapa antecedente, de forma que o processo é considerado um processo iterativo. Esta relação possibilita melhores resultados em cada etapa, uma vez que, trabalhar com o resultado da etapa anterior é mais produtivo e gera mais qualidade nos resultados das etapas subsequentes. Pode-se dizer que o processo de KDD compreende, em outras palavras, todo o ciclo que o dado percorre até que este se transforme em um conhecimento (CEREDA; JOSE, 2014). As principais etapas do processo são apresentadas, na forma de uma visão geral, na Figura 2.

Embora as etapas do processo KDD devam ser executadas na ordem apresentada na Figura 2, o processo é iterativo e interativo (CIOS et al., 2007). Iterativo, por ser uma sequência finita de operações em que o resultado de cada uma depende do resultado das que a precedem. Interativo, pois o usuário pode, e às vezes necessita, intervir e controlar o curso das atividades. Destaca-se, na mesma figura, a possibilidade da execução de "laços" entre quaisquer duas etapas, dados pelas setas tracejadas apresentadas abaixo de cada transição entre etapas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Isso significa que o usuário pode, quando necessário e em qualquer transição, retornar a qualquer etapa anterior para execução com quaisquer alterações desejadas.

Figura 2 - Etapas do processso de KDD.



Fonte: Adaptado de (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

O processo de KDD é uma tarefa intensiva de descoberta de conhecimento e possui interações complexas, feitas ao longo do tempo, entre o homem e o banco de dados através de um

conjunto de diferentes etapas e ferramentas. Apesar de ser descrito como um processo de nove etapas, apenas cinco delas são consideradas etapas principais, enquanto o restante é executado como etapas intermediárias ou de preparação entre etapas (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996), (DINIZ; LOUZADA-NETO, 2000). As etapas principais de seleção de dados, pré-processamento, transformação, *data mining* e interpretação e avaliação dos dados do processo de KDD são descritas nas seções a seguir.

2.2.1.1 Etapa de seleção de dados

Nesta etapa deve-se selecionar o banco de dados que será trabalhado e neste banco quais tabelas e atributos serão utilizados. Um banco de dados pode conter inúmeras tabelas e cada tabela pode ter muitos atributos. Sabe-se ainda que é possível selecionar dados de mais de um banco de dados, ou seja, trabalhar com algumas tabelas de um banco de dados e outras de outro banco de dados e isso também acontece com os atributos, pois tanto pode-se selecionar todos os atributos de uma tabela quanto pode-se selecionar apenas um atributo de determinada tabela.

A etapa de seleção de dados requer um tempo considerável do processo, pois é necessário conhecer e entender o banco de dados em questão. Entender, por exemplo, que tipo de dados são armazenados em uma determinada tabela e com quais tabelas essa tabela se relaciona. Esse estudo do banco de dados é fundamental antes de realizar a seleção dos dados, uma vez que, um dos fatores importantes para que uma *data mining* obtenha sucesso é assegurar a qualidade (completude, veracidade e integridade) dos dados selecionados (BOENTE; GOLDSCHMIDT; ESTRELA, 2008).

2.2.1.2 Etapa de pré-processamento

Para iniciar esta etapa é preciso ter concluído a etapa de seleção, pois é na etapa de seleção que é separado o conjunto de dados pertencentes a um domínio para que futuramente uma análise mais criteriosa possa ser iniciada neste domínio.

Na etapa de pré-processamento é feita uma limpeza nos dados, ou seja, é melhorada a qualidade dos dados. Na limpeza dos dados pode ser retirado algum atributo considerado sem relevância para o domínio em questão, ficando somente atributos que realmente integram o contexto (BOENTE; GOLDSCHMIDT; ESTRELA, 2008).

Segundo Fayyad e Uthurusamy (2002), outra atividade desta etapa é manusear dados inconsistentes ou faltantes, uma vez que é comum encontrar atributos que, na maioria dos registros estão preenchidos, mas que em alguns casos esses atributos ficam nulos. Considera-se que se

um atributo passou pela etapa de seleção e também pela limpeza dos dados é porque o mesmo é importante e não deveria estar nulo. Outro fato importante é que não basta o atributo não estar nulo, é necessário também que este seja consistente no sentido de que seu valor realmente faça parte do domínio. Por isso, nessa fase da etapa de pré-processamento, deve estar normalmente envolvido um perito no negócio ao qual correspondem os dados, uma vez que a detecção e correção de anomalias requerem conhecimento especializado (GOLDSCHMIDT; PASSOS, 2005).

Além da correção de possíveis erros e o preenchimento ou a eliminação de valores nulos ou redundantes, nesta etapa também são identificados e removidos dados duplicados e corrompidos (GOLDSCHMIDT; PASSOS, 2005).

2.2.1.3 Etapa de transformação dos dados

A execução da etapa de pré-processamento produz um conjunto de dados muitas vezes complexos para a execução posterior das etapas do KDD. Assim, como procedimento subsequente, executa-se a etapa de transformação dos dados, cujo objetivo é consolidar o conjunto de dados para possibilitar o trabalho com informações relevantes. Além disso, esta etapa permite otimizar o tempo de processamento do algoritmo de *data mining*, uma vez que reduz o número e a complexidade dos atributos disponíveis para análise (JIANG; TSENG; LIAO, 1999).

Uma das operações que podem estar envolvidas na transformação dos dados é a discretização de atributos numéricos, em que os valores dos dados numéricos são definidos como uma opção dentre um número limitado de opções. Outra possibilidade é a operação de generalização, onde os dados dos atributos são tratados de forma a se tornarem mais genéricos, categorizando os possíveis valores dos atributos em intervalos ou grupos contendo apenas as características mais relevantes (HAN; KAMBER, 2006).

Nesta etapa também pode-se realizar a seleção de atributos, que reduz o conjunto de dados ao remover atributos irrelevantes ou reduntantes. Esta operação pode ser realizada uma vez que o conjunto de dados para análise pode conter muitos atributos que não são aproveitados pela *data mining*. O objetivo desta operação é encontrar um conjunto mínimo de atributos tais que seus dados continuem o mais representativos possíveis em relação ao conjunto de dados original, que contém todos os atributos (HAN; KAMBER, 2006).

2.2.1.4 Etapa de data mining

A *data mining* é considerada a principal etapa do processo de KDD, pois é nesta etapa que é realizada a extração de conhecimentos novos ou são decobertos novos padrões. Esta etapa do KDD consiste na aplicação de algoritmos específicos, que extraem padrões a partir dos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Existem diferentes definições para *data mining*, entre elas:

- *Data mining* é a análise de grandes conjuntos de dados a fim de encontrar relacionamentos inesperados e de resumir os dados de uma forma que eles sejam tanto úteis quanto compreensíveis ao dono dos dados (HAND; MANNILA; SMYTH, 2001).
- *Data mining* é um campo interdisciplinar que junta técnicas de máquinas de conhecimentos, reconhecimento de padrões, estatísticas, banco de dados e visualização, para conseguir extrair informações de grandes bases de dados (CABENA et al., 1997).
- Sob a perspectiva do aprendizado de máquina, *data mining* é um passo no processo de descoberta de conhecimento que consiste na realização da análise dos dados e na aplicação de algoritmos de descoberta que, sob certas limitações computacionais, produzem um conjunto de padrões de certos dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Os objetivos da *data mining* podem ser alcançados através de diferentes métodos, também denominados de tarefas. As tarefas de *data mining* são variadas e distintas devido à variedade de padrões existentes em grandes volumes de dados e, para cada tipo de padrão são necessários diferentes métodos e abordagens. As diferentes tarefas de *data mining* podem ser definidas como sumarização, classificação, agrupamento, associação e regressão (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996; CHEN; HAN; YU, 1996).

A tarefa de sumarização realiza a abstração ou generalização dos dados, cuja aplicação resulta em um conjunto de dados menor, que fornece uma visão geral desses dados e que possuem, normalmente, algum conhecimento agregado. A classificação é a aprendizagem de regras ou funções que determinam a classe de um objeto de acordo com seus atributos. A tarefa de agrupamento realiza a identificação de grupos similares, ou *clusters*, a partir dos dados. O objetivo do agrupamento é agrupar conjuntos de objetos de forma que os objetos pertencentes ao mesmo grupo são mais semelhantes entre si, em relação àqueles em outros grupos. A tarefa de associação envolve a descoberta de uniões ou conexões entre objetos, denominadas regras

de associação, que revelam relacionamentos entre objetos no conjunto de dados. A tarefa de regressão é uma metodologia estatística que visa determinar funções com erros mínimos que modelem os dados. A regressão pode fornecer informações sobre o relacionamento e as interdependências entre os objetos do conjunto de dados (GHEWARE; KEJKAR; TONDARE, 2014; CHEN; HAN; YU, 1996).

A seguir são descritas com mais detalhes as tarefas de agrupamento e de classificação, sendo que ambas são utilizadas neste trabalho durante a aplicação do estudo de caso, pelas soluções oferecidas em cada tarefa e à adequação aos problemas deste trabalho.

Tarefa de agrupamento

O agrupamento é a identificação de classes, também denominadas de grupos ou *clusters*, a partir um conjunto de dados cujos objetos possuem classes desconhecidas. Em outras palavras, a tarefa de agrupamento realiza a identificação de grupos similares a partir de dados não estruturados. A tarefa se caracteriza por agrupar um conjunto de dados de entrada em grupos, de tal forma que os dados incluídos em um mesmo grupo são os mais similares entre si e menos similares em relação aos dados que estão em outros grupos (GIUDICI, 2005).

Uma vez que os grupos tenham sido definidos a partir do conjunto de dados, os objetos são rotulados de acordo com seu grupo correspondente. A rotulação dos dados em seus *clusters* permite distinguir características comuns de forma sumarizada para compor a descrição de cada *cluster*.

A tarefa de agrupamento pode ser realizada utilizando diferentes técnicas. Tradicionalmente, as técnicas de agrupamento são divididas em técnicas hierárquicas e técnicas de particionamento. Os algoritmos que utilizam as técnicas hierárquicas constroem *clusters* gradativamente, formando uma árvore de *clusters*. Para as técnicas hierárquicas são categorizados algoritmos aglomerativos e divisivos, de acordo com a forma em que os *clusters* são construídos. As técnicas de particionamento, por outro lado, geram os conjuntos diretamente, tentando descobrir *clusters* com a realocação de pontos entre os subconjuntos, ou ainda identificando *clusters* em uma área com alta densidade de dados. As técnicas de particionamento podem ser categorizadas em métodos probabilísticos, métodos *k-medoids*, métodos *k-means* e métodos baseados em densidade (BERKHIN, 2006). Na Tabela 1 apresenta-se a listagem de algoritmos de agrupamento de acordo com suas categorias e subcategorias.

Neste trabalho o algoritmo de *data mining* utilizado para a tarefa de agrupamento é baseado na técnica de particionamento, descrita a seguir.

Tabela 1 - Listagem de algoritmos de agrupamento conhecidos, divididos em técnicas e categorias, de acordo com a metodologia de construção dos *clusters*.

Técnica	Categoria	Algoritmo
Hierárquica	Aglomerativo	CURE
		Chameleon
	Divisivo	SLINK
		COBWEB
Particionamento	Probabilístico	EM (Expectation Maximization)
		SNOB
		AutoClass
		MCLUST
	k-medoids	PAM
		CLARA
		CLARANS
	k-means	SimpleKMeans
		Xmeans
		(diferentes implementações de *k-means*)
	Baseado em densidade	DBSCAN
		OPTICS
		DBCLASD
		DENCLUE

Fonte: Adaptado de (BERKHIN, 2006).

Técnica de particionamento

Na tarefa de agrupamento, a técnica mais comumente utilizada é a de particionamento, que tem o objetivo de verificar a existência de diferentes grupos dentro de um determinado conjunto de dados (SFERRA; CORREA, 2003).

Na técnica de particionamento, diferentes métodos e algoritmos podem ser utilizados. O método mais simples e mais difundido é o método baseado em *k-means* (MAIMON; ROKACH, 2010). Este algoritmo particiona os dados dentro de *k clusters* (C1, C2, ..., C*k*), representados pelos seus centros ou médias (*centroids*). O centro de cada *cluster* é calculado como sendo a média de instâncias pertencentes a cada *cluster*.

O *k-means* inicia com um conjunto de entrada de *clusters* centrais, escolhidos randomicamente ou de acordo com algum procedimento heurístico. A cada iteração, cada instância é atribuída ao centro do seu *cluster* mais próximo, de acordo com uma medida de distância entre ambos. Então, os centros de *cluster* são recalculados. Assim, cada vez mais os dados similares vão ficando mais próximos e os que apresentam menos similaridade vão sendo alocados em outros *clusters*.

A medida de distância definida no algoritmo de agrupamento é usada para a definição de quão próximos os atributos estão, de forma a determinar a similaridade entre os atributos de diferentes instâncias. A medida de distância usada tradicionalmente é a distância euclidiana, que representa a distância em linha reta entre dois pontos no espaço euclidiano (WU et al., 2008). A definição da medida de distância é importante pois engloba o problema da normalização de atributos na tarefa de agrupamento.

Quando diferentes atributos numéricos fazem parte do conjunto de dados na tarefa de agrupamento, é importante observar suas características a respeito da proporção que os valores podem assumir. Quando os valores numéricos são comparados com base na medida de distância euclidiana, essas medidas são afetadas pela escala das variáveis (WITTEN; FRANK; HALL, 2011). Em outras palavras, ao computar a distância entre dois objetos, cada qual com informações de idade e salário, por exemplo, a distância será afetada drasticamente pois a extensão dos valores poderia variar entre 18 e 60 para a idade, e de 900 a 10.000 para o salário. Esse é um problema tratado pelos algoritmos de agrupamento através do processo de normalização de atributos.

A normalização de atributos consiste da aplicação de técnicas que padronizam os dados, geralmente para defini-los no intervalo entre 0 e 1. A normalização pode ser conseguida dividindo todos os valores pelo valor máximo encontrado para o determinado atributo, ou ainda subtraindo o valor mínimo de cada registro e dividindo essa diferença pela variação entre os valores máximo e mínimo (WU et al., 2008). A normalização é um passo geralmente executado na tarefa de agrupamento e é realizada por padrão em implementações comuns de algoritmos de agrupamento (WITTEN; FRANK; HALL, 2011).

Tarefa de classificação

A tarefa de classificação é uma função de aprendizado que mapeia dados de entrada, ou um conjuntos de dados de entrada, em um número finito de categorias pré-definidas, denominadas de classes (GIUDICI, 2005). Uma vez determinada, a função pode ser aplicada a novos registros de forma a estimar a classe em que os registros se enquadram.

O objetivo da classificação é encontrar algum relacionamento entre os atributos e uma classe, de modo que o processo de classificação possa usar esse relacionamento para predizer a classe de um exemplo novo e desconhecido (GIUDICI, 2005; KESAVARAJ; SUKUMARAN, 2013; REN; ZHANG; SUGANTHAN, 2016).

Como exemplo da tarefa de classificação, considere uma padaria que tem um histórico com dados e o comportamento dos seus clientes em relação à frequência de cada cliente à padaria.

Considere dois tipos de clientes: os que vão à padaria todos os dias e os que raramente vão à padaria e isso classifique os clientes como assíduos e não assíduos, respectivamente. Neste caso, a tarefa de classificação pode ser aplicada para determinar uma função que mapeie corretamente os clientes, a partir de seus dados, em uma destas classes. Uma vez descoberta, esta função pode ser utilizada para prever o comportamento de novos clientes que gostariam de criar uma conta na padaria. Esta função pode ser incorporada a um sistema de apoio à decisão que auxilie na filtragem e permissão para abertura de conta somente a clientes classificados como assíduos.

Diferentes técnicas podem ser aplicadas à classificação, entre elas as técnicas de árvores de decisão, redes neurais artificiais, vizinho mais próximo (*nearest neighbor*) e classificação bayesiana. As técnicas de árvore de decisão criam uma estrutura de árvore a partir de um conjunto de dados de entrada na qual cada nó representa atributos ou condições de testes e as folhas representam os resultados dos testes. Os métodos baseados em algoritmos de redes neurais artificiais podem ser usados para modelar relacionamentos complexos entre as entradas para encontrar padrões nos dados. A metodologia de classificação baseada em *nearest neighbor* encontra a distância de uma dada instância com o restante dos dados e calcula a classe dessa instância como sendo a maioria das classes dos vizinhos mais próximos. Por fim, a classificação bayesiana é uma forma de classificação estatística baseada no Teorema de Bayes, em que a classificação de uma instância se baseia na sua probabilidade de assumir determinada classe (BOENTE; GOLDSCHMIDT; ESTRELA, 2008; DEEBA; AMUTHA, 2016). A Tabela 2 apresenta a listagem de algoritmos de agrupamento de acordo com suas categorias e subcategorias.

Neste trabalho o algoritmo de *data mining* utilizado para a tarefa de classificação é baseado na técnica de árvore de decisão, descrita a seguir.

Tabela 2 - Listagem de algoritmos de classificação, divididos de acordo com suas técnicas.

Técnica	**Algoritmo**
Árvore de decisão	Hunt
	ID3
	C4.5 / J4.8
	CART
	REPTree
Redes Neurais Artificiais	Multi-Layer Perceptron
Nearest Neighbor	kNN
Classificação Bayesiana	Naive Bayes
	BayesNet
	(diferentes implementações de Naive Bayes)

Fonte: Adaptado de (BOENTE; GOLDSCHMIDT; ESTRELA, 2008; DEEBA; AMUTHA, 2016).

Técnica de árvore de decisão

A técnica de árvore de decisão consiste basicamente em formar uma árvore composta por nó raiz, nós internos e nós folhas. O nó raiz nunca recebe uma aresta, deste nó apenas saem arestas. Os nós internos recebem apenas uma aresta, porém, destes pode sair mais de uma aresta. Os nós folhas também recebem apenas uma aresta e destes, diferente dos nós internos, não saem arestas.

Em uma árvore de decisão, cada nó interno separa o conjunto de dados em dois ou mais subconjuntos, de acordo com os valores dos atributos de entrada. Um nó folha recebe um rótulo de classe. Os nós internos e o nó raiz contêm condições de testes de atributos que separam registros que possuem características diferentes (MAIMON; ROKACH, 2010).

Ao classificar um registro do conjunto de dados, percorre-se um caminho na árvore que se inicia a partir do nó raiz, onde é realizada a aplicação da condição de teste do nó, seguindo, posteriormente, para a ramificação apropriada baseado no resultado do teste. O próximo nó pode ser um nó interno, onde uma nova condição de teste é aplicada, ou um nó folha. Neste caso, o rótulo deste nó é atribuído ao registro que, por sua vez, é classificado (TAN; STEINBACH; KUMAR, 2009).

Algoritmos que constroem árvores de decisão são baseados em métodos heurísticos e estratégias localmente ótimas, uma vez que induzir uma árvore de decisão a partir de um conjunto de dados é uma tarefa NP-difícil (MAIMON; ROKACH, 2010). Os algoritmos se baseiam em uma estratégia que expande uma árvore de decisão escolhendo localmente quais atributos usar para particionar os dados. Um desses algoritmos é o algoritmo de Hunt, que é a base de diversos algoritmos como ID3 (QUINLAN, 1986), C4.5 (QUINLAN, 1993) e CART (BREIMAN et al., 1984).

Segundo Tan, Steinbach e Kumar (2009), o algoritmo de Hunt gera uma árvore de decisão que expande recursivamente pelo particionamento dos registros em subconjuntos sucessivos. A árvore de decisão inicial contém um único nó raiz com um rótulo de classe. O algoritmo refinará a árvore de modo que a raiz contenha subconjuntos menores, tomando um novo atributo para gerar os subconjuntos. Este procedimento é aplicado recursivamente a cada filho do nó raiz. O algoritmo é interrompido quando todos os registros de cada filho pertencem à mesma classe em um nó folha.

2.2.1.5 Etapa de interpretação e avaliação

É nesta última etapa do processo de KDD que o resultado da *data mining* é interpretado e avaliado. Na interpretação é onde verifica-se se é possível interpretar o resultado, se a resposta do algoritmo é compreensiva para que os tomadores de decisão, que normalmente são os responsáveis por validar os conhecimentos adquiridos, possam analisar os resultados.

Na etapa de avaliação, os resultados que foram considerados interpretáveis são avaliados, ou seja, a pessoa que detém conhecimento do negócio, cujos dados foram utilizados, vai analisar se o resultado extraiu conhecimento e se esse novo conhecimento extraído é útil. Na avaliação do resultado também é validado se o novo conhecimento gerado pode identificar um novo padrão ou gerar uma nova regra. Portanto, é avaliado se o conhecimento extraído pode ser útil na tomada de decisão.

Caso o resultado não for interpretável ou na avaliação não for verificado um novo conhecimento é necessário voltar a realizar parte ou todo o processo de KDD (BOENTE; GOLDSCHMIDT; ESTRELA, 2008).

Ao final desse Capítulo conclui-se que o processo de KDD pode ser interessante para descoberta de novos conhecimentos e que as tarefas de agrupamento e classificação são bastantes pertinentes para atingir o objetivo da etapa de *data mining*. Conclui-se também que as técnicas de particionamento e árvore de decisão são relevantes para o obtenção de bons resultados quando são utilizadas as técnicas de agrupamento e classificação.

3 FERRAMENTA WEKA

Nos últimos anos, o grande aumento do uso de processos de *data mining* pelas empresas motivou o surgimento de muitos softwares proprietários e livres para tal finalidade, por exemplo, o RapidMiner, o Apache Spark, o Weka, entre outros. O Weka é um sistema que teve um destaque considerável.

Segundo Witten, Frank e Hall (2011), o Weka é um software muito popular de aprendizagem de máquina (*machine learning*) e *data mining*. É totalmente livre, do tipo *open source*, distribuído sob os termos da GNU GPL (*General Public License*). O sistema foi desenvolvido em Java e pode ser executado em praticamente qualquer plataforma, tendo sido testado em sistemas operacionais Linux, Windows e Macintosh.

O Weka se consolidou como o sistema de *data mining* mais utilizado em ambientes acadêmicos (WITTEN; FRANK; HALL, 2011). Este sistema é utilizado não apenas em pesquisas científicas, mas principalmente para fins didáticos. Foi exatamente a sua adequação para este último tipo de aplicação que o tornou popular.

O sistema Weka foi criado em uma universidade da Nova Zelândia (The University of Waikato). Weka significa *Waikato environment for knowledge analysis*. O sistema provê uma interface uniforme para diferentes algoritmos de aprendizagem, além de métodos para pré e pós-processamento e avaliação do resultado da aprendizagem em um certo conjunto de dados (HOLMES; DONKIN; WITTEN, 1994).

O sistema oferece implementações de algoritmos de aprendizagem que podem ser facilmente aplicados em conjunto de dados, além de incluir uma variedade de ferramentas para transformação de dados, tais como seleção de atributo e limpeza de dados. É possível pré-processar um conjunto de dados, alimentar um algoritmo de aprendizagem e analisar o resultado e seu desempenho, sem que seja necessário escrever sequer uma linha de código (HALL et al., 2009).

O sistema Weka inclui métodos para os principais problemas de *data mining*: regressão, classificação, agrupamento e associação. O sistema disponibiliza também muitas ferramentas para visualização e pré-processamento de dados. O recurso mais valioso que o Weka oferece é a implementação de algoritmos de aprendizagens reais. Mas, as ferramentas de pré-processamento, chamadas de filtros, também são importantes (WITTEN; FRANK; HALL,

2011).

Para Witten, Frank e Hall (2011), o Weka pode ser considerado uma coleção de classes da linguagem Java, que compreendem, entre outras, classes de árvore de decisão, filtros de pré-processamento ou classes que implementam algoritmos específicos de aprendizagem. No Weka cada implementação de um algoritmo está encapsulada em uma classe e pode depender de outras classes para suas funcionalidades. A vantagem desta estrutura é a possibilidade de se criar códigos próprios em Java e integrar a eles classes de algoritmos de aprendizagem disponibilizados pelo Weka.

Figura 3 - Interface inicial do sistema Weka.

Fonte: Autoria própria.

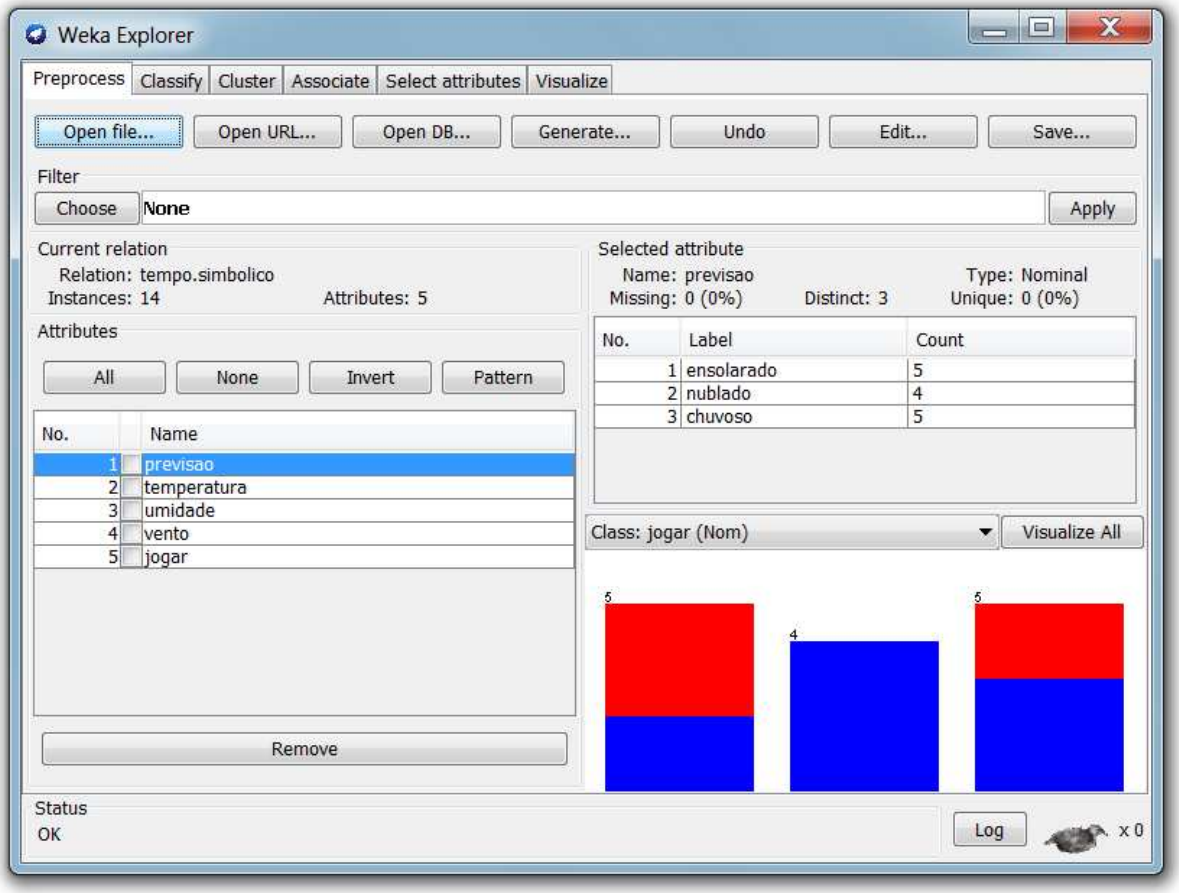
A maneira mais fácil de usar o Weka é através de uma aplicação gráfica denominada *Explorer*, que dá acesso a todos os recursos usando seleção de menus e preenchimento de formulários. Existem duas outras aplicações gráficas, denominadas *Experimenter* e *KnowledgeFlow*. A quarta aplicação dá acesso aos comandos do sistema através de uma interface de linha de comando e é denominada *Simple CLI*. Ao iniciar o sistema Weka, uma interface oferece acesso a cada umas das aplicações mencionadas, como pode ser visto na Figura 3. A seguir é descrita cada aplicação.

3.1 APLICAÇÃO *EXPLORER*

A interface gráfica principal do Weka, a *Explorer*, permite o acesso a todos os seus recursos através de menus e formulários, como pode ser visto na Figura 4. Existem seis diferentes telas,

selecionadas pelas abas localizadas no topo, correspondendo às tarefas de *data mining* que o sistema suporta.

Figura 4 - Interface *Explorer*.



Fonte: Autoria própria.

3.1.1 Preparação de dados

Os conjuntos de dados utilizados em *data mining*, geralmente obtidos a partir de bancos de dados, devem ser estruturados em um dos formatos aceitos pelo Weka, CSV (*Comma Separated Value*) ou ARFF (*Attribute-Relation File Format*). O formato CSV armazena, em cada linha do arquivo, um registro do conjunto de dados. Cada registro consiste de um ou mais campos, cujos valores são separados por vírgula. O formato ARFF consiste de um cabeçalho e uma porção de dados. No cabeçalho são definidos o nome do conjunto de dados (@*relation*) e os atributos e seus respectivos tipos (@*attribute*). A seguir, a região identificada por @*data* contém uma sequência de registros em cada linha e os valores dos seus atributos separados por vírgula. A estrutura de um arquivo no formato ARFF é apresentada na Figura 5.

Figura 5 - Arquivo no formato ARFF.

```
@relation Tempo

@attribute previsao {ensolarado, nublado, chuvoso}
@attribute temperatura real
@attribute umidade real
@attribute vento {verdadeiro, falso}
@attribute jogar {sim, nao}

@data
ensolarado,85,85,falso,nao
ensolarado,80,90,verdadeiro,nao
nublado,83,86,falso,sim
chuvoso,70,96,falso,sim
chuvoso,68,80,falso,sim
chuvoso,65,70,verdadeiro,nao
nublado,64,65,verdadeiro,sim
ensolarado,72,95,falso,nao
ensolarado,69,70,falso,sim
chuvoso,75,80,falso,sim
ensolarado,75,70,verdadeiro,sim
nublado,72,90,verdadeiro,sim
nublado,81,75,falso,sim
chuvoso,71,91,verdadeiro,nao
```

Fonte: Autoria própria.

O CSV é um formato simples e amplamente utilizado em diversas aplicações na área da Computação. O formato ARFF, em contrapartida, foi criado especificamente para o sistema Weka. A maioria dos bancos de dados permite a exportação de dados no formato CSV e a conversão deste formato para o ARFF pode ser feita inserindo os dados do cabeçalho antes dos valores separados por vírgula e posteriormente renomeando o arquivo com a extensão `.arff`.

3.1.2 Aba de pré-processamento

Na aba de pré-processamento, denominada *Preprocess*, é necessário carregar os dados em que a *data mining* será aplicada. Esta interface permite que o arquivo com o conjunto de dados seja importado através da opção *Open file...*. A mesma interface permite que, após carregado o conjunto de dados, sejam alterados os atributos que farão parte dos passos posteriores, através da listagem dos atributos na seção *Attributes*. A mesma tela ainda exibe informações estatísticas dos dados carregados, bem como gráficos de barra indicando os valores quantitativos dos atributos. Na Figura 4 é apresentada a tela inicial do *Weka Explorer* com a aba *Preprocess* selecionada e os dados carregados.

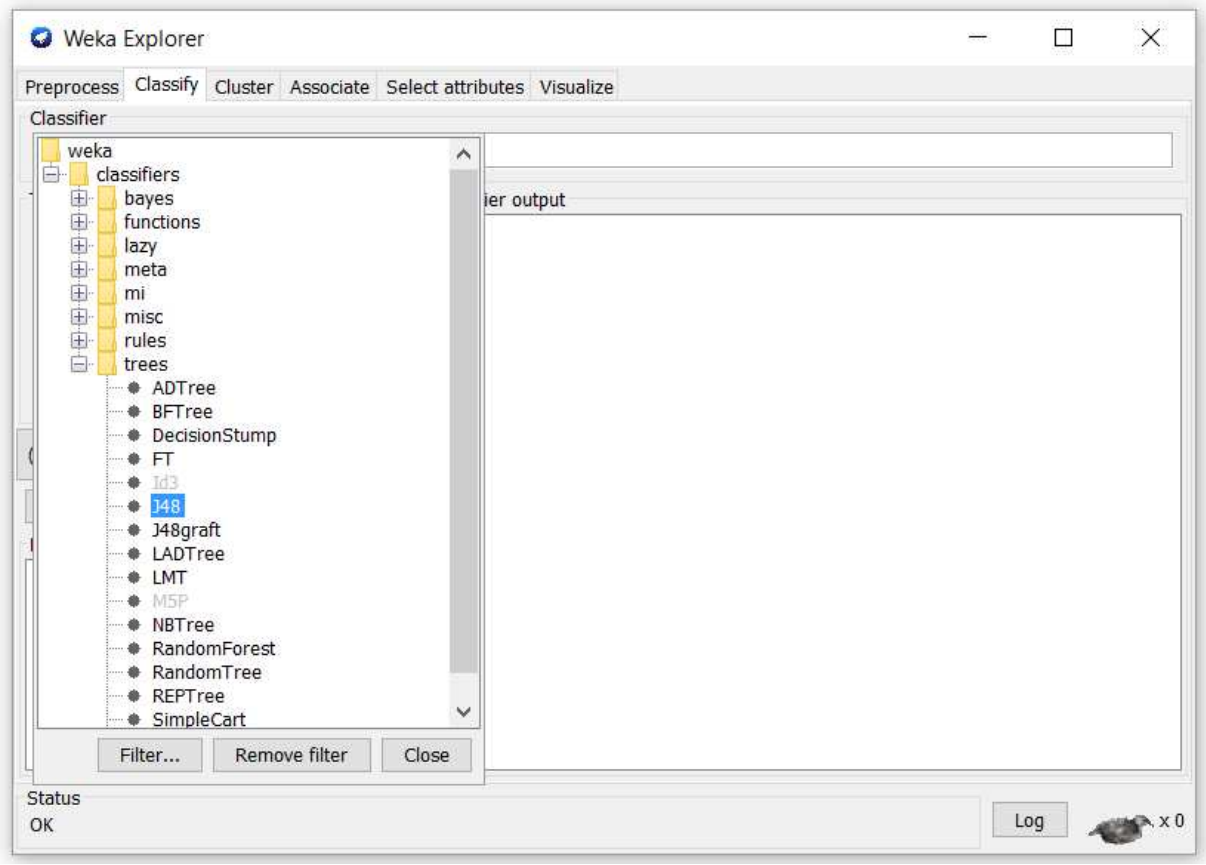
3.1.3 Abas de classificação, agrupamento e associação

Classificar é a segunda aba da aplicação *Explorer*, denominada *Classify*. Nesta aba é possível realizar técnicas pertencentes à tarefa de classificação. Agrupar é a terceira aba da aplicação, chamada de *Cluster*, onde são executados algoritmos da tarefa de agrupamento. Associar é a quarta aba do *Explorer*, rotulada de *Associate*. Nesta aba é realizada a tarefa de associação, através de algoritmos específicos para esta função. A forma de utilizar cada uma das três abas é a mesma, diferenciando apenas pelos algoritmos e técnicas escolhidos e consequentemente seus resultados.

Na utilização de qualquer uma das três abas, primeiramente, deve-se definir o algoritmo que será executado tomando como entrada os dados pré-processados na aba de pré-processamento. Deve-se notar que há um grupo seleto de algoritmos específicos disponíveis na interface de cada aba, de acordo com a tarefa que cada aba se propõe a fazer, sendo elas classificar, agrupar e associar.

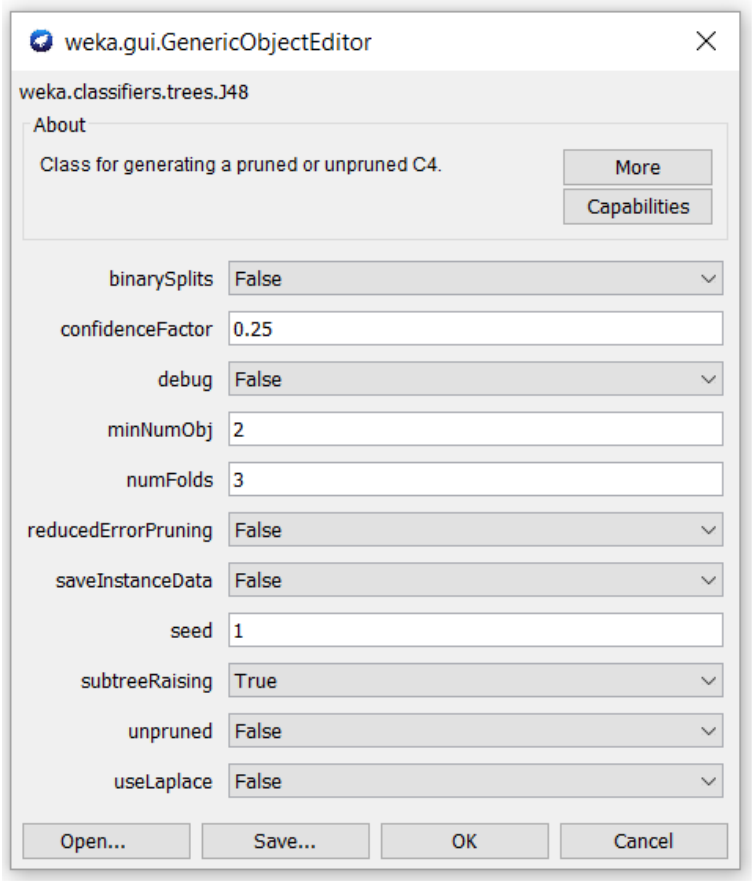
A seleção do algoritmo é realizada através de um menu hierárquico que compreende todos os algoritmos específicos de cada tarefa disponíveis no Weka, como pode ser visto na Figura 6. Uma vez selecionado o algoritmo, a interface permite que seus parâmetros sejam configurados, como apresentado na Figura 7, porém, esses parâmetros já são inicializados com valores padrão, podendo ou não serem alterados (WITTEN; FRANK; HALL, 2011).

Figura 6 - Menu de escolha dos algoritmos.



Fonte: Autoria própria.

Figura 7 - Tela de configuração dos parâmetros do algoritmo J4.8.



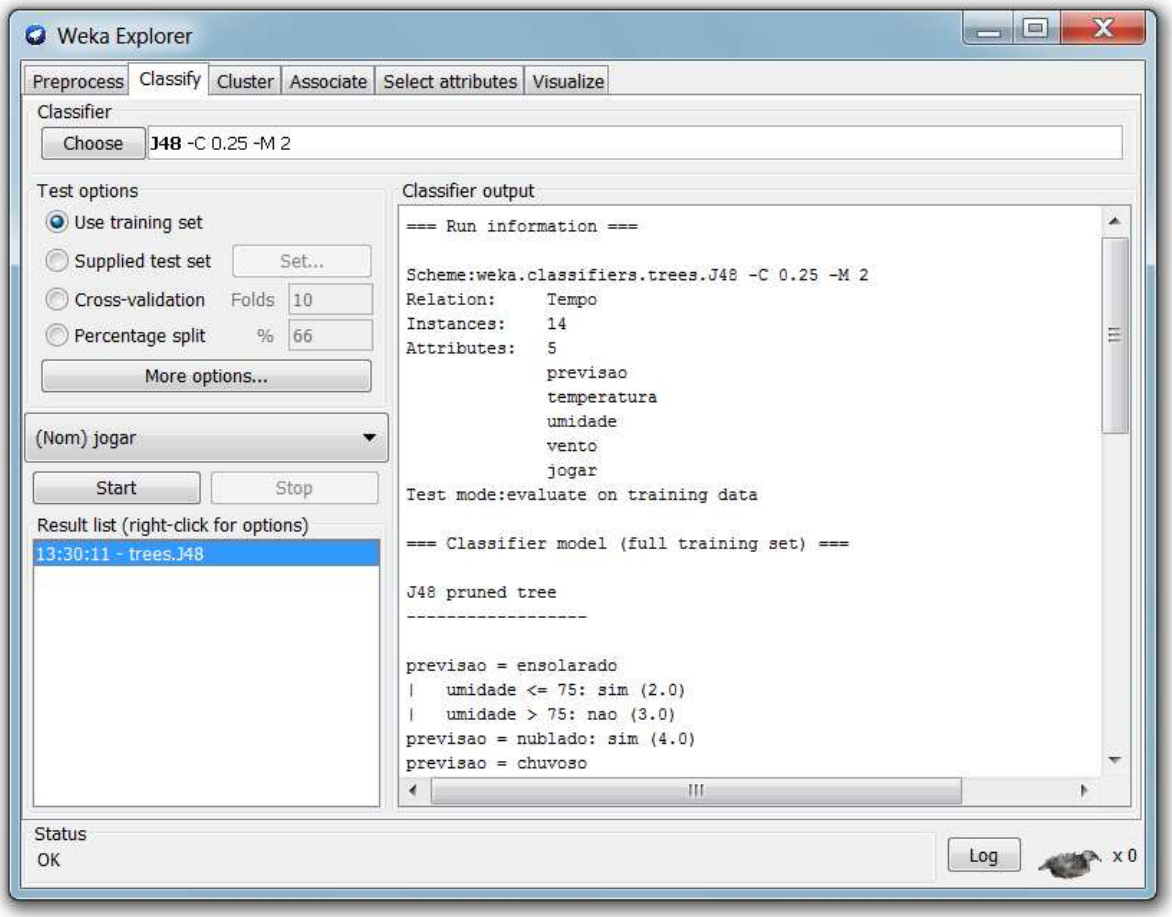
Fonte: Autoria própria.

Com o algoritmo já selecionado e os parâmetros devidamente definidos a Weka pode iniciar a execução do algoritmo. Assim que a execução do algoritmo é finalizada, automaticamente é apresentado o resultado na mesma tela. A interface de manipulação para as três tarefas é semelhante.

A seguir é apresentado, como exemplo, na Figura 8, a interface para a tarefa de classificação. Nota-se que para este caso foi utilizado o algoritmo J4.8 e seus parâmetros padrão, como também foi escolhida a opção de teste *Use Training Set*.

Para as tarefas de agrupamento e associação a interface é muito semelhante à apresentada na Figura 8, alterando-se praticamente só os algoritmos correspondentes, motivo pelo qual as mesmas não serão apresentadas.

Figura 8 - Aba *Classify* do *Explorer.*



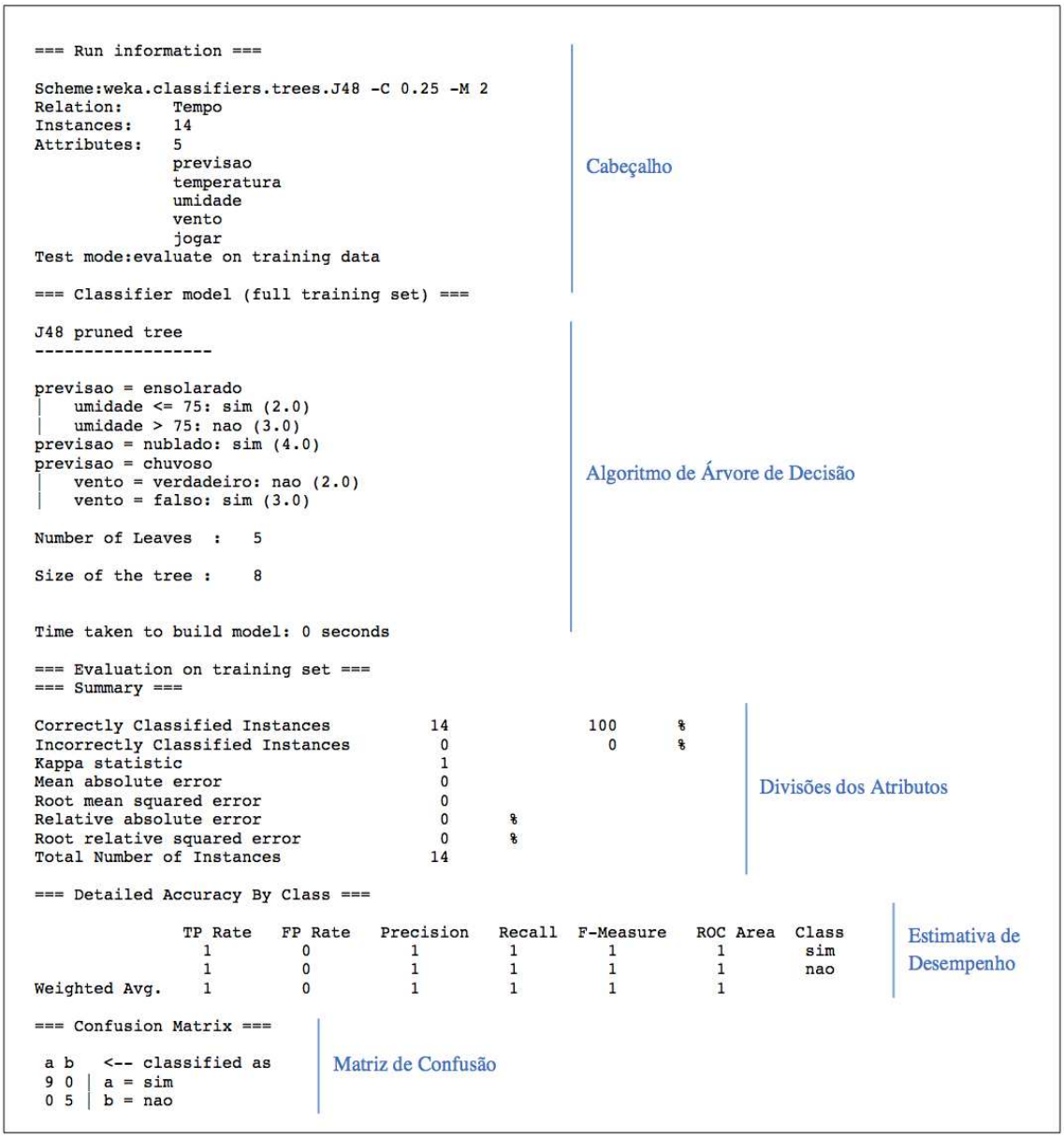
Fonte: Autoria própria.

Apesar da semelhança na interface de escolha e parametrização de algoritmos das três diferentes tarefas citadas nesta seção, seus resultados variam conforme a tarefa executada. Para exemplificar são apresentados e descritos os resultados para cada uma das tarefas.

Nos exemplos a seguir, é utilizado um pequeno conjunto de dados totalmente fictício, que representa as condições de tempo adequados para se jogar ou não um jogo não especificado. Os dados estão disponíveis em (WITTEN; FRANK; HALL, 2011). O objetivo dos exemplos é mostrar como são apresentados os resultados para as diferentes tarefas.

Na Figura 9 mostra-se o resultado completo da execução de um algoritmo de classificação, que compreende, inicialmente, um resumo do conjunto de dados de entrada, seguido por uma árvore de decisão em forma textual e as divisões dos atributos nos nós da árvore. Por fim, são apresentados uma estimativa do desempenho da predição da árvore e uma matriz de confusão.

Figura 9 - Resultado da execução da tarefa de classificação.



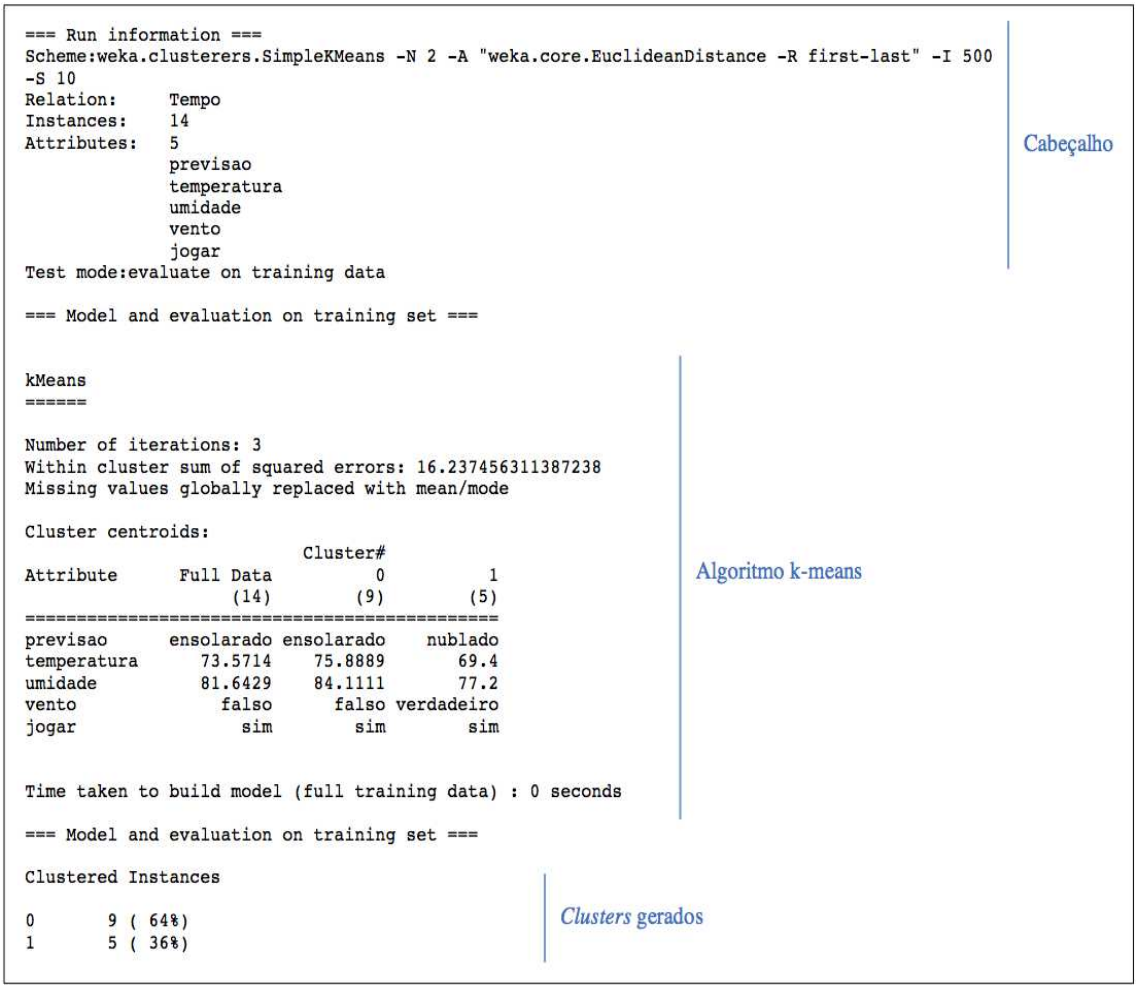
Fonte: Autoria própria.

Como resultado final obteve-se que das quatorze possibilidades de jogar, todas foram classificadas corretamente, onde nove dizem que sim, o tempo é favorável ao jogo e cinco dizem que o tempo não é favorável ao jogo.

Na Figura 10 apresenta-se o resultado da execução de um algoritmo da tarefa de agrupamento. Na parte superior do resultado é apresentado um cabeçalho com informações resumidas dos parâmetros utilizados na execução do algoritmo, bem como dos atributos. Na sequência o resultado do agrupamento é mostrado em forma de uma tabela textual com linhas, que são os nomes dos atributos, e colunas, que correspondem aos *centroids* de cada agrupamento. Por

último é apresentado o número de instâncias que foram alocadas em cada *cluster*.

Figura 10 - Resultado da execução da tarefa de agrupamento.

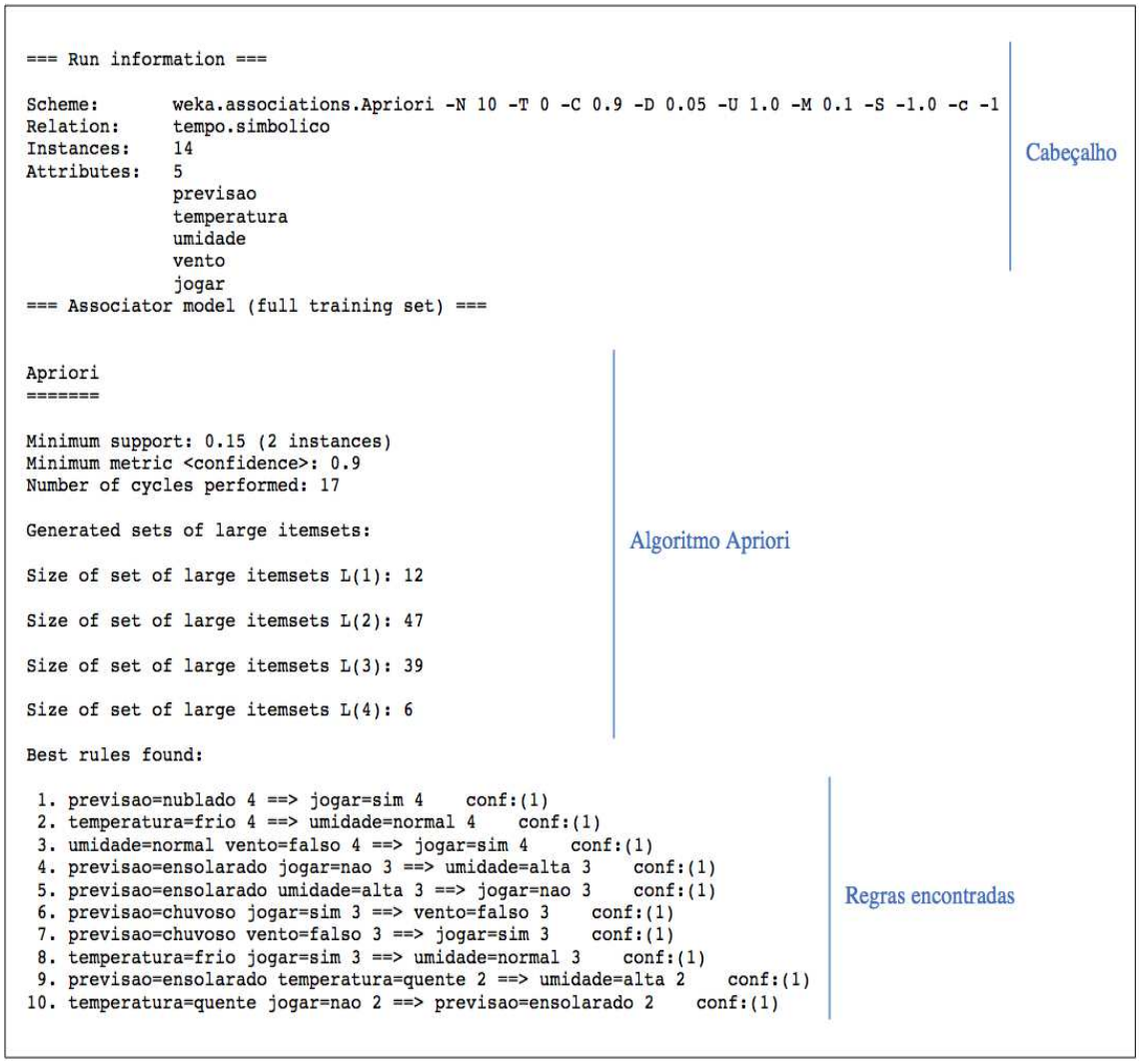


Fonte: Autoria própria.

Como resultado final, após a execução do algoritmo de agrupamento, tem-se as quatorze instâncias divididas em 2 *clusters*, onde o *cluster* 0 (zero) compreende nove instâncias em que o tempo é favorável ao jogo e o *cluster* 1 (um) contém cinco instâncias onde o tempo não é favorável ao jogo.

O resultado da tarefa de associação é mais simples comparado aos de classificação e agrupamento. No início do resultado é apresentado um cabeçalho também com os parâmetros utilizados para execução do algoritmo, seguido dos atributos utilizados. Após o cabeçalho é apresentado o tamanho do conjunto de itens e em seguida as melhores regras encontradas para o caso em questão. Um exemplo do resultado da execução de uma tarefa de associação pode ser visto na Figura 11.

Figura 11 - Resultado da execução da tarefa de associação.



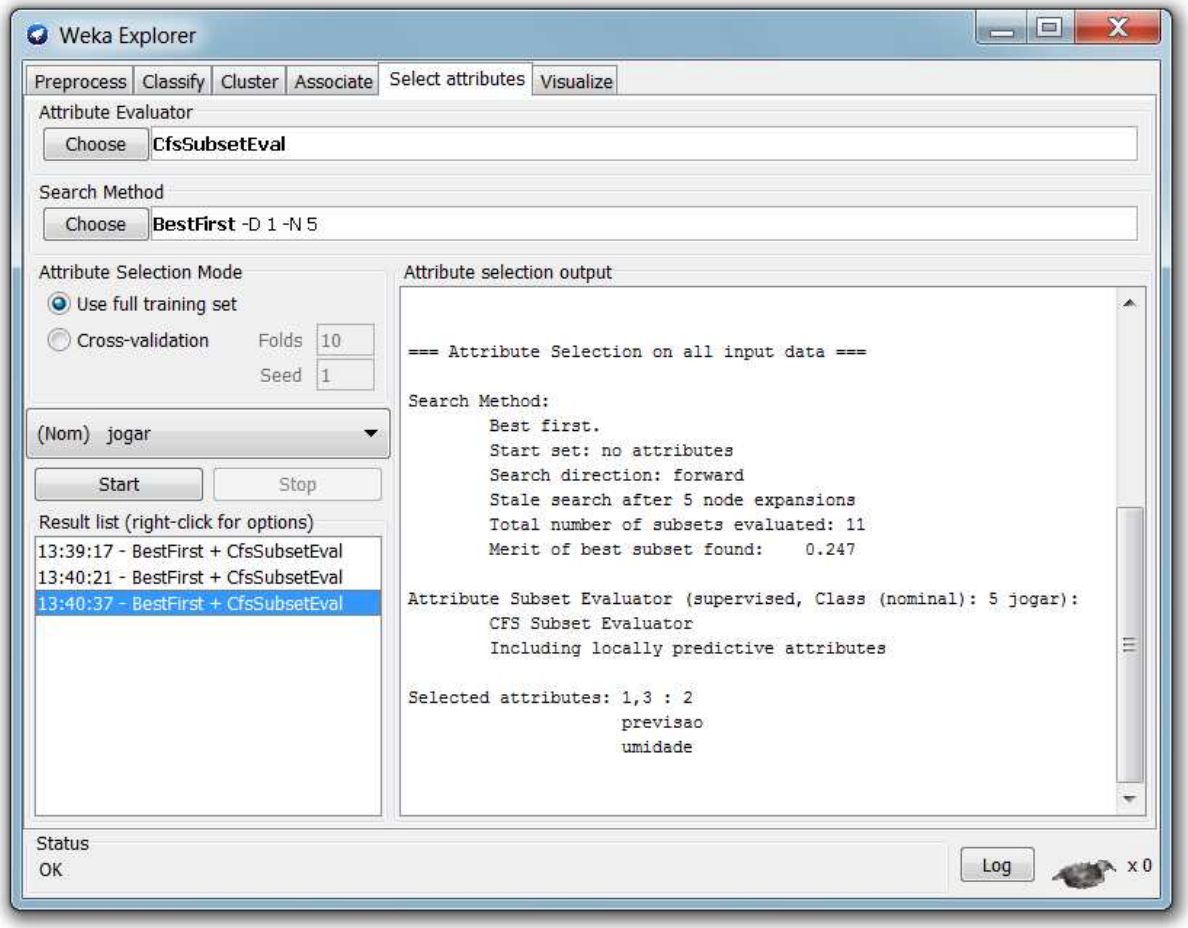
Fonte: Autoria própria.

O resultado da tarefa de associação mostra que foram encontradas dez regras de associação. Onde cada linha é uma regra e em cada linha o valor apresentado à esquerda da seta é o número de instâncias que o antecedente é verdadeiro e o valor à direita da seta é o número de instâncias para os quais o consequente também é verdadeiro. O valor entre parênteses no final de cada linha é o valor de confiança da regra, que é taxa entre o valor antecedente e o valor consequente. Exemplo: regra 4, todas as vezes que a `previsao` for `ensolarado` e `jogar` for `nao` a `umidade` será `alta`.

3.1.4 Aba de seleção de atributos

A quinta aba, seleção de atributos (*Select Attributes*), apresentada na Figura 12, é utilizada para selecionar os atributos mais relevantes, uma vez que os algoritmos de aprendizagem apresentam efeitos negativos na presença de atributos irrelevantes que impactam negativamente o desempenho desses algoritmos.

Figura 12 - Aba *Select Attribute* do *Explorer*.



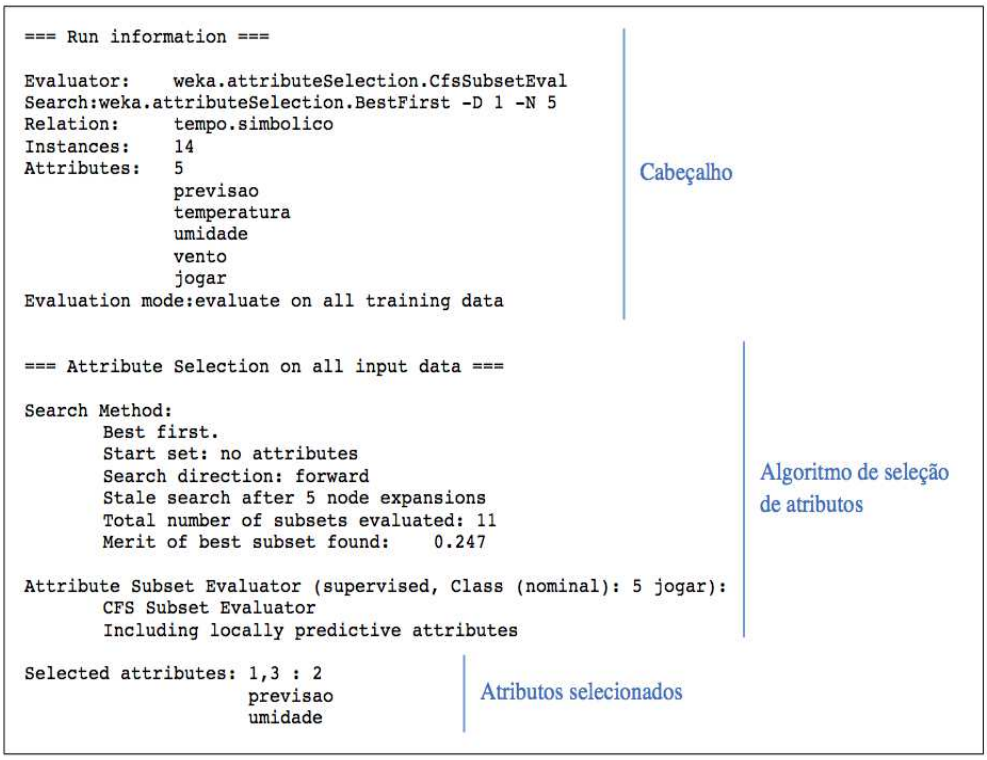
Fonte: Autoria própria.

A etapa de seleção de atributos é pouco realizada no Weka, pois a melhor forma de seleção de dados relevantes para *data mining* é realizada manualmente, baseado em um entendimento profundo do problema e do que os atributos realmente significam (WITTEN; FRANK; HALL, 2011). No entanto, métodos automáticos podem ser úteis pois podem auxiliar na tarefa manual diminuindo as possibilidades de existir um atributo irrelevante no conjunto de dados.

Para a seleção de atributos devem ser escolhidos: o algoritmo de seleção de atributos, o método de busca no conjunto de dados e o atributo alvo. Um exemplo de resultado de seleção

de atributos é mostrado na Figura 13. O resultado apresenta um resumo de informações de parâmetros utilizados na execução do algoritmo escolhido, os atributos, informações do método utilizado e por último os atributos que foram selecionados como mais relevantes.

Figura 13 - Resultado da execução da seleção de atributos.



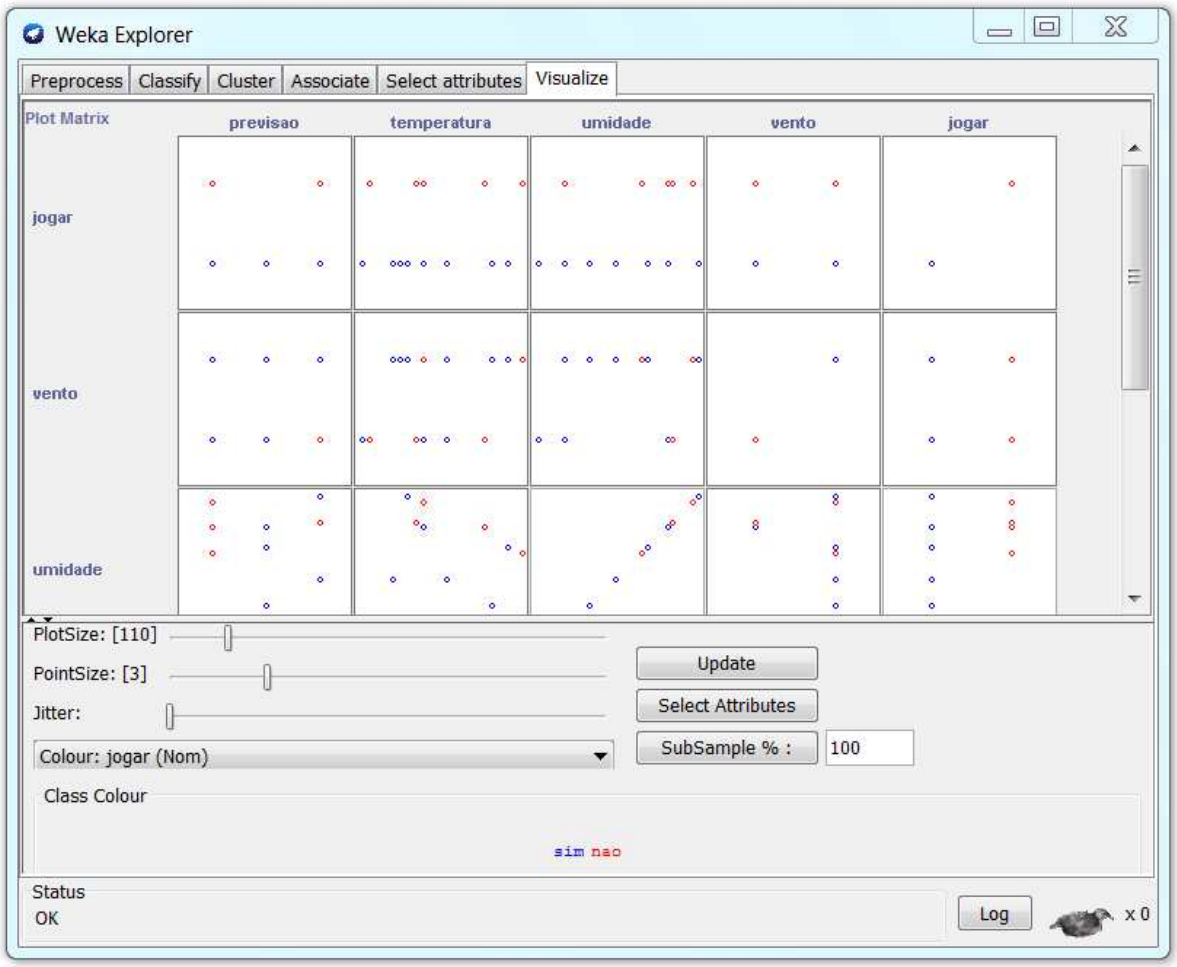
Fonte: Autoria própria.

Como resultado do algoritmo de seleção de atributos são apresentados os dois atributos mais independentes entre si e que ao mesmo tempo são os mais relacionados ao atributo alvo. No caso em questão o atributo alvo é o `jogar` e os dois atributos selecionados mais independentes entre si e mais relacionados ao atributo `jogar` são `previsao` e `umidade`.

3.1.5 Aba de visualização

A última aba, denominada *Visualize*, é utilizada para visualizar os atributos do conjunto de dados de entrada. Sua interface exibe uma matriz de duas dimensões de plotagem espalhada para cada par de atributos. Além disso, é possível alterar alguns parâmetros desta interface, como cor e tamanho, para melhorar a vizualização dos dados no espaço bidimensional. Na Figura 14 mostra-se um exemplo de resultado visualizado através desta interface, onde as interrelações dos atributos são caracterizados em cada região pelo posicionamento dos pontos no formato (x, y).

Figura 14 - Aba *Visualize* do Explorer.



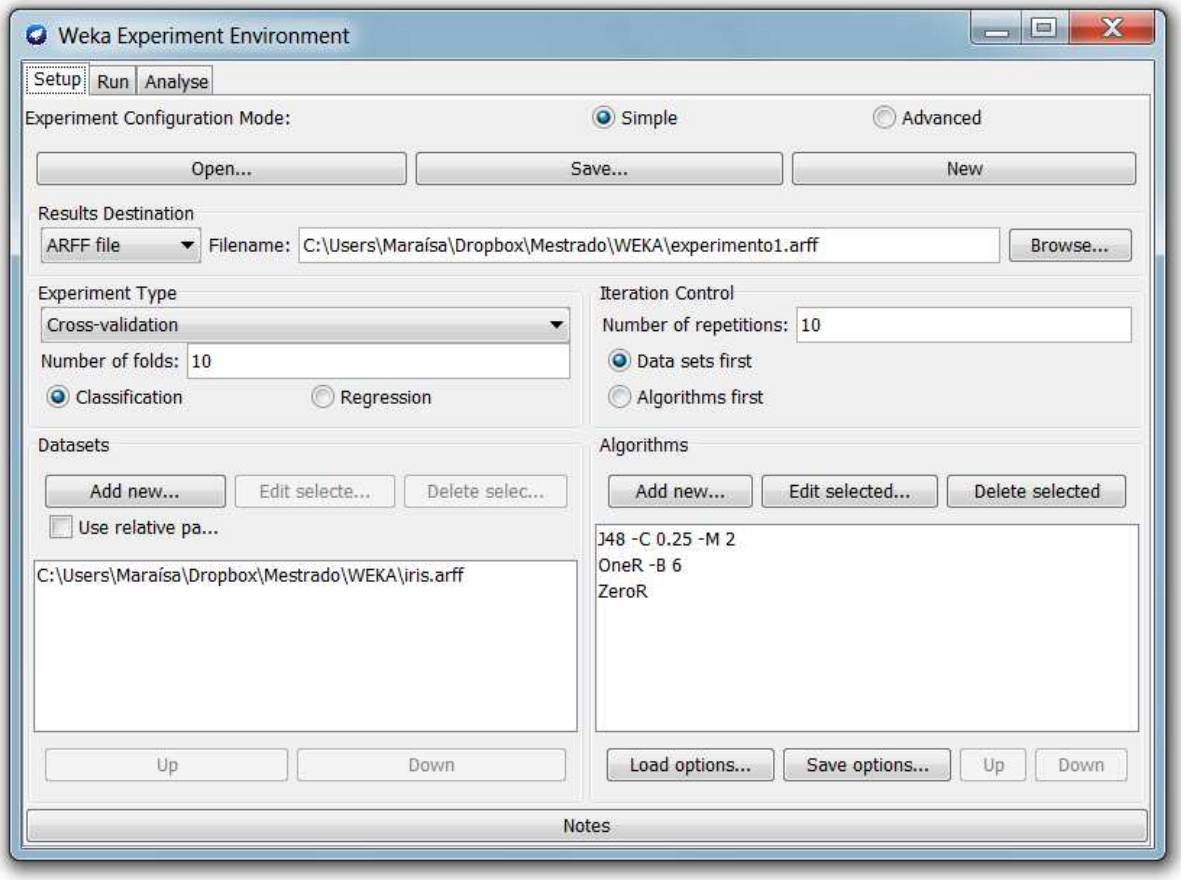
Fonte: Autoria própria.

3.2 APLICAÇÃO *EXPERIMENTER*

As aplicações *Explorer* e *KnowledgeFlow* executam *data mining* em conjuntos de dados específicos e utilizam apenas um algoritmo por vez. Já a aplicação *Experimenter* permite executar experimentos em grande escala, utlizando normalmente vários algoritmos de *data mining* em diferentes conjuntos de dados. Sua interface permite gerar diferentes configurações de parâmetros para cada algoritmo. A execução da aplicação *Experimenter* permite automatizar o processo de *data mining*.

Na Figura 15, apresenta-se a interface inicial da aplicação *Experimenter*. Através desta interface é possível selecionar os conjuntos de dados e os algoritmos que farão parte do experimento, além de permitir modificar outras configurações e parâmetros.

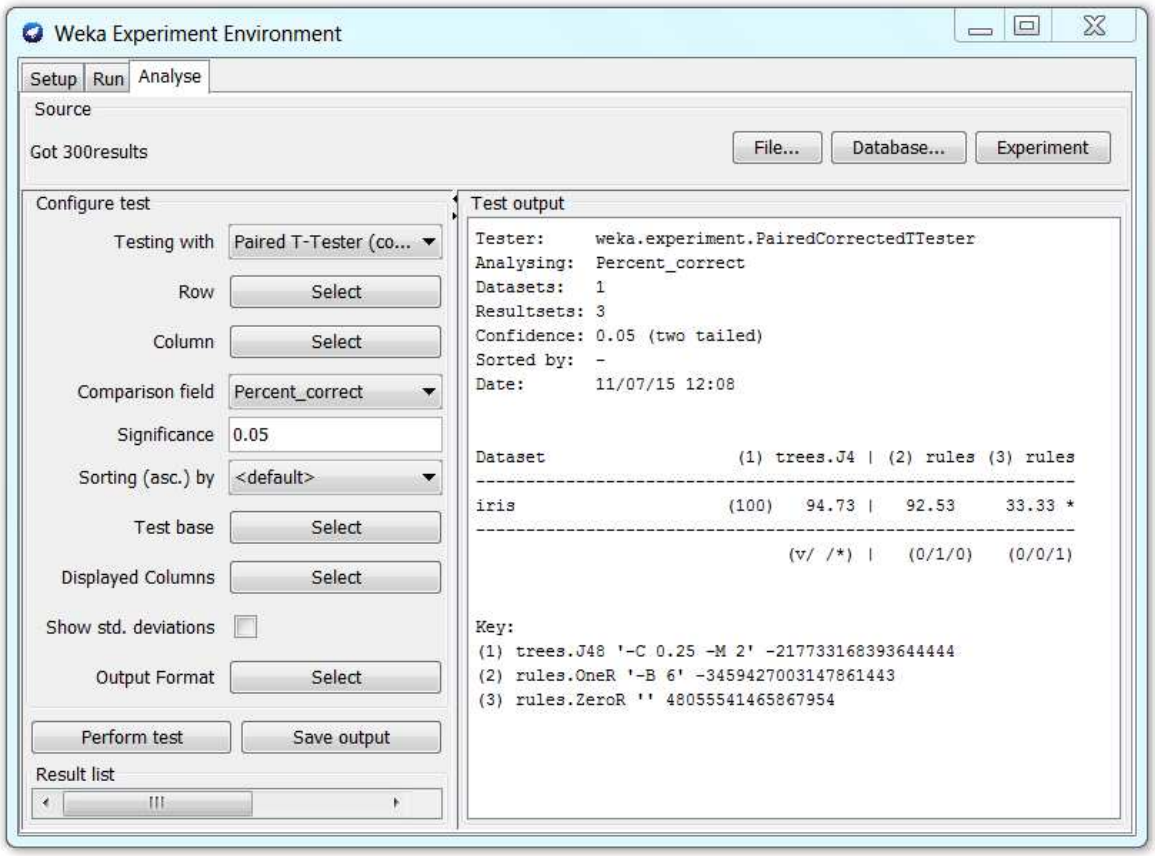
Figura 15 - Aba *Experimenter* do *Explorer*.



Fonte: Autoria própria.

O resultado do experimento é apresentado em outra aba da aplicação, denominada *Analyse*, que deve ser vizualizada após ser acionada a aba *Run*, com os parâmetros associados. Na Figura 16 mostra-se o resultado de um experimento que compara três diferentes algoritmos de *data mining* em um mesmo conjunto de dados. A estrutura do resultado é composta por um cabeçalho e em seguida uma tabela textual que apresenta a comparação dos resultados de cada algoritmo para o conjunto de dados.

Figura 16 - Aba *Analyse* do *Experimenter*.



Fonte: Autoria própria.

Como resultado do experimento são apresentados os três algoritmos em uma tabela, onde o desempenho do algoritmo mais à esquerda (J4.8), denominado base, é comparado com o desempenho dos outros dois (OneR e ZeroR). Os rótulos 1, 2 e 3 das colunas são repetidos no final do resultado, pois não há espaço suficiente para seus nomes na tabela. O valor entre parênteses (100) representa o número de execuções experimentais.

O símbolo (*v/ /**) a baixo da coluna 1 (algoritmo base) denota uma contagem no estilo (*x/y/z*) onde x representa o número de vezes que o algoritmo foi melhor que o base, y representa o número de vezes que o algoritmo não foi nem melhor nem pior que o base e z representa o número de vezes que o algoritmo foi pior que o base. No resultado, o algoritmo 2 é equivalente ao algoritmo base e o algoritmo 3 foi pior que o base. O símbolo * indica o pior resultado na tabela.

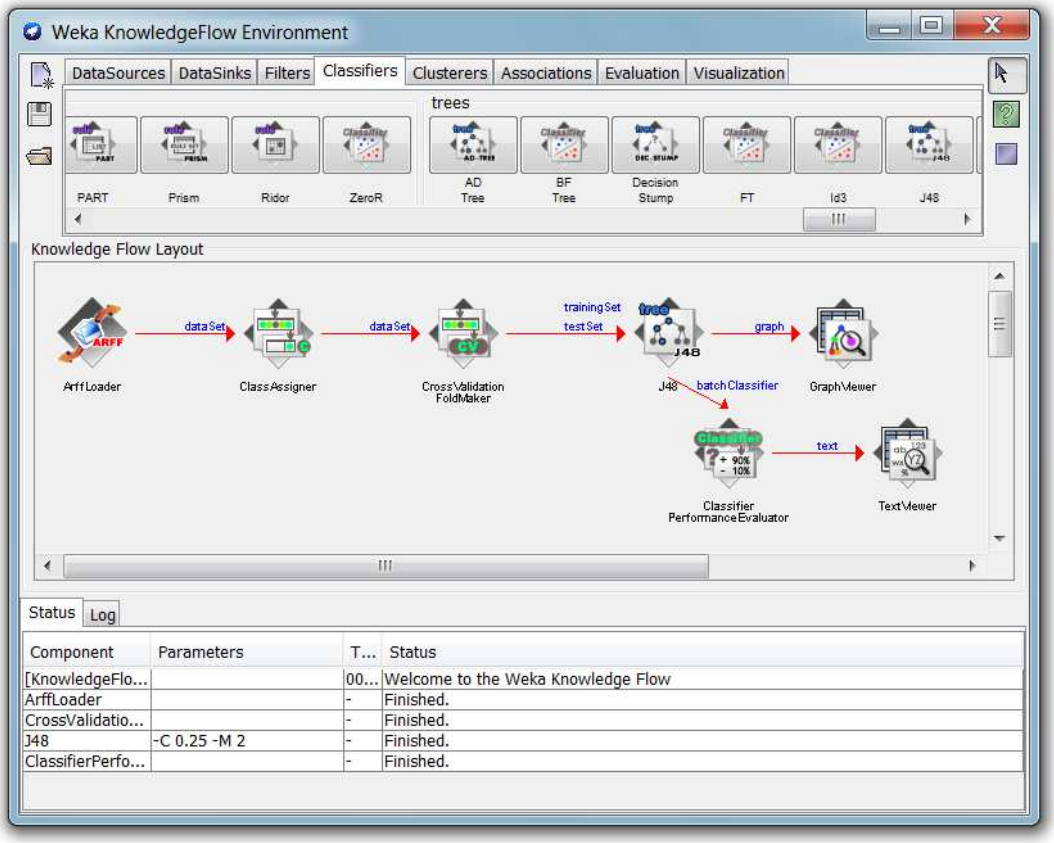
A interface *Analyse* permite também a alteração de alguns parâmetros do experimento e a verificação da variação do resultado conforme os parâmetros são alterados.

3.3 APLICAÇÃO *KNOWLEDGEFLOW*

A aplicação *KnowledgeFlow* possibilita a execução de todas as funcionalidades disponíveis na aplicação *Explorer* de forma gráfica, selecionando componentes e alocando-os em um espaço e conectando-os através um grafo direcionado que processa e analiza os dados. Cada componente desta interface representa um recurso que o *Explorer* oferece, por exemplo, algoritmos de classificação ou agrupamento, visualizadores gráficos de resultados, entre outros.

Assim como no *Explorer*, os componentes da aplicação permitem alterar seus parâmetros de configuração. No entando, o *KnowledgeFlow* permite o desenho e a execução de configurações para o processamento de dados em fluxo, que o *Explorer* não pode fazer (WITTEN; FRANK; HALL, 2011).

Figura 17 - Exemplo *KnowledgeFlow*.



Fonte: Autoria própria.

Na Figura 17 é apresentado um exemplo de componentes posicionados que carregam um arquivo `arff` e realizam a execução de um algoritmo de árvore de decisão, permitindo a posterior exibição dos resultados em formato gráfico e de texto. Independente da interface de execução, *Explorer* ou *KnowledgeFlow*, desde que os parâmetros estejam igualmente configurados o resultado da execução do algoritmo de *data mining* é sempre o mesmo.

3.4 APLICAÇÃO *SIMPLE CLI*

De acordo com (WITTEN; FRANK; HALL, 2011), a *Simple CLI*, também chamada de interface de linha de comando, oferece um painel de texto simples onde pode-se inserir comandos. Tais comandos compreendem os mesmos recursos que são oferecidos pelas intefaces gráficas das aplicações descritas nas seções anteriores.

Esta interface facilita a integração entre o Weka e códigos desenvolvidos, independentemente, em Java. Além disso, a interface permite uma execução mais direta de algoritmos específicos, como, por exemplo, o ID3:

```
java weka.classifiers.trees.Id3 -t dados.arff
```

Esta linha de código executa o Java e o instrui a executar o algoritmo de árvore de decisão Id3, passando como parâmetro o arquivo dados.`arff`, que contém os dados de treinamento. O resultado é apresentado na mesma interface de linha de comando de uma maneira semelhante àqueles aprentados nos exemplos de resultado da aplicação *Explorer*, conforme a Figura 13, na página 43.

4 METODOLOGIA PARA DESCOBERTA DE CONHECIMENTO EM BANCOS DE DADOS ACADÊMICOS

Neste capítulo propõe-se uma metodologia para a execução do processo de descoberta de novos conhecimento em bancos de dados, baseada no processo de KDD, voltada a dados acadêmicos e à descoberta de informações novas e úteis a respeito desses dados. As tarefas de *data mining* cobertas pela metodologia são as tarefas de agrupamento e de classificação, que são ambas aplicadas a dois estudos de caso distintos: o estudo de caso do IFMS e o estudo de caso da UNESP, descritos no Capítulo 5.

4.1 METODOLOGIA PROPOSTA

Inicialmente sugere-se que seja disponibilizado um banco de dados para acesso a consultas de forma que sejam conhecidas suas estruturas e relacionamentos principais, a fim de detectar as possibilidades de extração de informação. A partir do estudo do banco de dados, as etapas de seleção, pré-processamento e transformação dos dados, que precedem a execução de algum algoritmo de *data mining*, são realizadas de forma distinta entre as execuções de classificação e agrupamento. Nos estudos de caso, a tarefa de classificação é aplicada em diferentes disciplinas relacionadas a um mesmo curso, o que aumenta a dificuldade de obtenção dos dados na etapa de seleção, e de tratamento da informações nas etapas de pré-processamento e transformação dos dados. Por outro lado, a tarefa de agrupamento é realizada com informações mais gerais dos cursos e dos estudantes, não necessitando especificação detalhada de matrículas específicas em diversas disciplinas.

Para a tarefa de classificação, portanto, foi desenvolvida uma ferramenta computacional específica para apoio à execução das etapas iniciais da metodologia, denominada Aplicação para Apoio à *Data Mining* (AADM). Os detalhes do desenvolvimento e da execução da ferramenta computacional de apoio são apresentados na seção seguinte. Já para a tarefa de agrupamento, os passos foram executados manualmente, sem o auxílio da ferramenta, uma vez que os dados necessários são mais gerais e demandam menos consultas e manipulações no banco de dados.

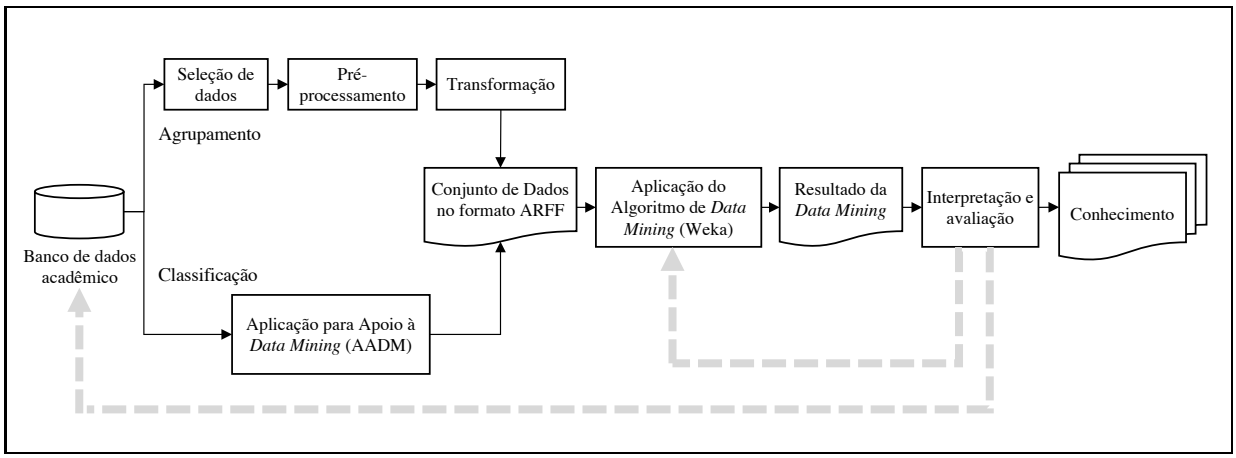
A metodologia concentra a execução de algoritmos de *data mining* através da ferramenta Weka, em que as etapas anteriores são executadas de forma independente, seja através da AADM ou de forma manual. Como a metodologia proposta é baseada no KDD, após a execução dos algoritmos de *data mining*, uma etapa final é necessária para avaliação e compreensão

dos resultados obtidos. Esta etapa é realizada de forma detalhada, analisando cada caso individualmente, e os resultados obtidos são apresentados para cada um dos estudos de caso.

A metodologia apresentada para a descoberta de conhecimento em banco de dados acadêmico pode ser sumarizada por etapas gerais conforme o fluxograma apresentado na Figura 18. Nesta figura, pode ser visto que, a partir da etapa de interpretação e avaliação, é possível retornar a algum passo anterior definido caso o resultado obtido não seja satisfatório. Os estudos de caso apresentados no capítulo seguinte são realizados e alterados diversas vezes através das etapas apresentadas, muitas vezes retornando-se aos passos anteriores permitidos, até que se obtenha resultados interessantes e úteis.

A metodologia proposta difere do processo de KDD apresentado na Figura 2, do Capítulo 2, página 21, no sentido em que o KDD permite, de forma genérica, a repetição de execução de algum passo anterior nos casos em que algum resultado não seja satisfatório (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Na metodologia proposta neste trabalho, sugere-se que a possibilidade de repetição de etapas anteriores pode ocorrer somente na etapa de interpretação e avaliação, já que só se verifica com clareza a necessidade de alguma correção após interpretar e avaliar o resultado obtido. Além disso, a volta pode ocorrer apenas para a etapa de *data mining* e para o passo inicial. Isto é, voltar para a etapa de *data mining* significa executar novamente o algoritmo de mineração, utilizando modos de execução diferentes, até que se obtenha um resultado útil. Voltar ao passo inicial é necessário quando se nota que os dados e atributos que fizeram parte do estudo não foram capazes de gerar resultados satisfatórios. Neste caso, a metodologia apresentada não sugere voltar a uma etapa anterior arbitrária, como no processo de KDD original, em que a volta é definida de forma mais abrangente.

Figura 18 - Fluxograma das etapas gerais que compõem a metodologia proposta para a descoberta de conhecimento em bancos de dados acadêmicos.



Fonte: Autoria própria.

O conjunto de etapas da metodologia proposta tem como ponto de partida o acesso ao banco de dados acadêmico. Nesta fase inicial que precede todos os outros passos, deve ser realizado o estudo dos dados e da estrutura do banco de dados a fim de detectar as tabelas mais importantes, atributos úteis e os relacionamentos entre diferentes tabelas. Após o estudo do banco de dados, pode-se dar início às etapas da metodologia que, inicialmente, se dividem de acordo com a tarefa a ser executada, conforme apresentado na Figura 18. Para a tarefa de agrupamento, as etapas de seleção de dados, pré-processamento e transformação são executadas de forma independente. Para a tarefa de classificação, é executada a Aplicação para Apoio à *Data Mining*, que integra os passos de forma automatizada. Para ambas as tarefas a saída é compatível com as etapas posteriores, que são comuns para o restante da metodologia. Na Figura 19 são apresentadas as etapas a serem realizadas nas tarefas de agrupamento e classificação.

Figura 19 - Etapas das tarefas de agrupamento e classificação.

Agrupamento	**Classificação**
Etapa de seleção de dados	
Etapa de pré-processamento	Aplicação para apoio a *data mining* (AADM)
Etapa de transformação	
Etapa de aplicação do algoritmo de *data mining*	Etapa de aplicação do algoritmo de *data mining*
Etapa de interpretação e avaliação	Etapa de interpretação e avaliação

Fonte: Autoria própria.

Nas seções a seguir são descritos os detalhes da execução de cada etapa da metodologia.

4.2 ETAPA DE SELEÇÃO DE DADOS

A primeira etapa é onde devem ser selecionados os atributos que irão formar um conjunto de dados para ser trabalhado. Os atributos podem ser selecionados a partir de diversas tabelas do banco de dados, inclusive por dados provenientes de mais de um banco de dados. Os campos selecionados podem ser de domínios diferentes, desde que exista uma relação que possa ser considerada confiável entre eles. A execução desta etapa requer algum conhecimento sobre a estrutura do banco de dados e sobre os próprios dados para que os atributos possam ser selecionados de forma estratégica e, assim, aumentar as possibilidades de sucesso no resultado da aplicação da metodologia.

Também é possível realizar a seleção de dados de formas alternativas, isto é, existe a pos-

sibilidade de extração de dados indiretamente a partir de bancos de dados, seja por meio de relatórios gerados pelo sistema que acessa o banco de dados, seja por meio da extração manual dos dados em planilhas.

4.3 ETAPA DE PRÉ-PROCESSAMENTO

A etapa de pré-processamento tem o objetivo de melhorar a qualidade dos dados selecionados na etapa anterior, tornando-os o mais confiável possível. A obtenção de um conjunto de dados fidedigno é essencial para atingir resultados satisfatórios na aplicação da metodologia, bem como para que os resultados possam ser úteis em seus contextos originais.

Esta etapa consiste em analisar os atributos selecionados e verificar se os dados têm integridade, ou seja, garantir a exatidão e a consistência dos dados no conjunto de dados. Nesta etapa, os dados selecionados dever ser analisados detalhadamente, para que registros incompletos, duplicados, inconsistentes ou com ruídos possam ser tratados ou mesmo eliminados. É preciso verificar de forma crítica o conjunto de dados selecionado, pois se algum registro no conjunto de dados não transmitir confiabilidade em seu valor, este atributo deve ser descartado. Registros com dados ausentes também devem ser excluídos ou, se existir a possibilidade de preenchimento da informação de maneira manual, esta deve ser utilizada e o atributo pode ser mantido.

Em alguns casos pode ser verificado que atributos selecionados na execução da etapa de seleção podem não ter relevância para a aplicação da metodologia. Nesse caso, esses dados dever ser eliminados na etapa atual para que o conjunto de dados obtido tenha o mais alto grau de qualidade possível.

4.4 ETAPA DE TRANSFORMAÇÃO

A etapa de transformação consiste em aplicar procedimentos sobre os dados de forma que eles sejam modificados para execução das etapas posteriores. As modificações sobre os dados são realizadas em cada atributo de acordo com seu tipo de valor e de acordo com alguma simplificação que se deseja obter. Entre os procedimentos mais comuns realizados na etapa de transformação está a discretização de atributos numéricos, em que valores numéricos contínuos são discretizados de acordo com algum parâmetro. Além disso, outro procedimento pode ser realizado nos atributos nominais, os quais podem ter seu valor de exibição ajustado, mas mantendo seu valor significativo original. Como exemplo do ajuste do valor de exibição está a alteração de valores de palavras completas para siglas, ou vice-versa, ou ainda a generalização

de palavras masculinas e femininas.

O objetivo da etapa de transformação é simplificar a forma em que os valores dos atributos são expressos, consolidando o conjunto de dados para as etapas posteriores da metodologia. Em especial, o tempo de execução do algoritmo de *data mining* pode ser melhorado dada a relativa simplificação aplicada ao conjunto de dados.

Uma outra vantagem que a etapa de transformação dos dados oferece para a metodologia em geral é que, em sua etapa final, o processo de interpretação dos resultados pode ser facilitado, uma vez que os atributos presentes já foram consolidados e simplificados na etapa em questão.

O passo final da etapa de transformação é a construção do arquivo que compreende o conjunto de dados preparado para posterior execução do algoritmo de *data mining*. O formato de arquivo que a metodogia propõe é o ARFF, que dispõe os atributos e seus valores no trecho `@relation` do arquivo, seguido pelo trecho de dados `@data`, onde se encontram os registros. Um exemplo contendo um trecho de um conjunto de dados no formato ARFF é apresentado na Figura 20.

Figura 20 - Trecho de conjunto de dados no formato ARFF gerado pela aplicação AADM, contendo os atributos e os dados para execução da classificação.

```
@relation Aluno

@attribute programacao {S, N}
@attribute web1 {S, N}
@attribute matematica {S, N}
@attribute logica {S, N}
@attribute algoritmos {S, N}
@attribute sexo {M, F}
@attribute idade real
@attribute solteiro {S, N}

@data

N,S,S,S,S,M,34,S
N,S,S,S,N,M,37,N
S,S,S,S,S,M,25,S
N,S,N,S,N,F,30,S
S,S,S,S,S,M,30,S
S,S,S,S,S,M,48,N
S,S,N,N,S,M,20,S
N,S,N,N,N,M,30,S
S,S,S,S,S,M,35,S
S,S,N,N,N,F,22,S
S,S,S,N,N,M,23,S
S,N,S,N,N,M,31,S
S,S,S,N,S,M,21,S
⋮
```

Fonte: Autoria própria.

4.5 APLICAÇÃO PARA APOIO À *DATA MINING* (AADM)

As etapas até aqui descritas são realizadas manualmente, de acordo com a metodologia proposta, para a execução da tarefa de agrupamento, como pode ser visto na Figura 18. Por outro lado, quando se deseja aplicar a tarefa de classificação, a mesma figura sugere uma execução diferente. Neste caso, as etapas de seleção de dados, pré-processamento e transformação são unificadas em um único passo denominado Aplicação para Apoio à *Data Mining* (AADM).

A AADM consiste de uma aplicação de *software* desenvolvida, como parte deste trabalho, com o objetivo de automatizar e otimizar as etapas iniciais da metodologia proposta, especificamente para a tarefa de classificação. Como já mencionado, os passos necessários para a obtenção dos dados acadêmicos de maneira apropriada para aplicação da tarefa de classificação exigem consultas complexas ao banco de dados e diversas manipulações a fim de se obter um conjunto de dados ideal.

A aplicação foi desenvolvida na linguagem de programação Python, realiza a comunicação com um banco de dados, executa consultas complexas e manipula os resultados obtidos utilizando estruturas de dados como *arrays* e matrizes da linguagem Python. A aplicação é executada através da linha de comando juntamente com a passagem de parâmetros necessários para a sua execução.

A AADM realiza diversas operações e contempla as etapas de seleção, pré-processamento e transformação dos dados, de forma que as operações sejam realizadas da forma mais automática e direta possível. A aplicação recebe, como entrada, a data de matrícula dos alunos, as disciplinas que se deseja filtrar e o *campus* da instituição onde se oferta o curso. Com os dados de entrada, a aplicação constrói uma sentença SQL utilizando as tabelas e os relacionamentos de interesse e realiza a consulta ao banco de dados, obtendo uma lista de registros dos alunos que cursaram todas as disciplinas definidas. Este processo compõe a etapa de seleção dos dados. A seguir, a lista de registros é manipulada de forma que os alunos que cursaram a mesma disciplina mais de uma vez, em semestres diferentes, são eliminados. Os alunos que apresentam algum atributo nulo são retirados do conjunto de dados. Este processo pertence à etapa de pré-processamento dos dados. Por fim, a aplicação executa a etapa de transformação dos dados, onde atributos específicos dos registros são ajustados em seu valor de exibição, como situação em cada disciplina, estado civil, sexo ou data de nascimento.

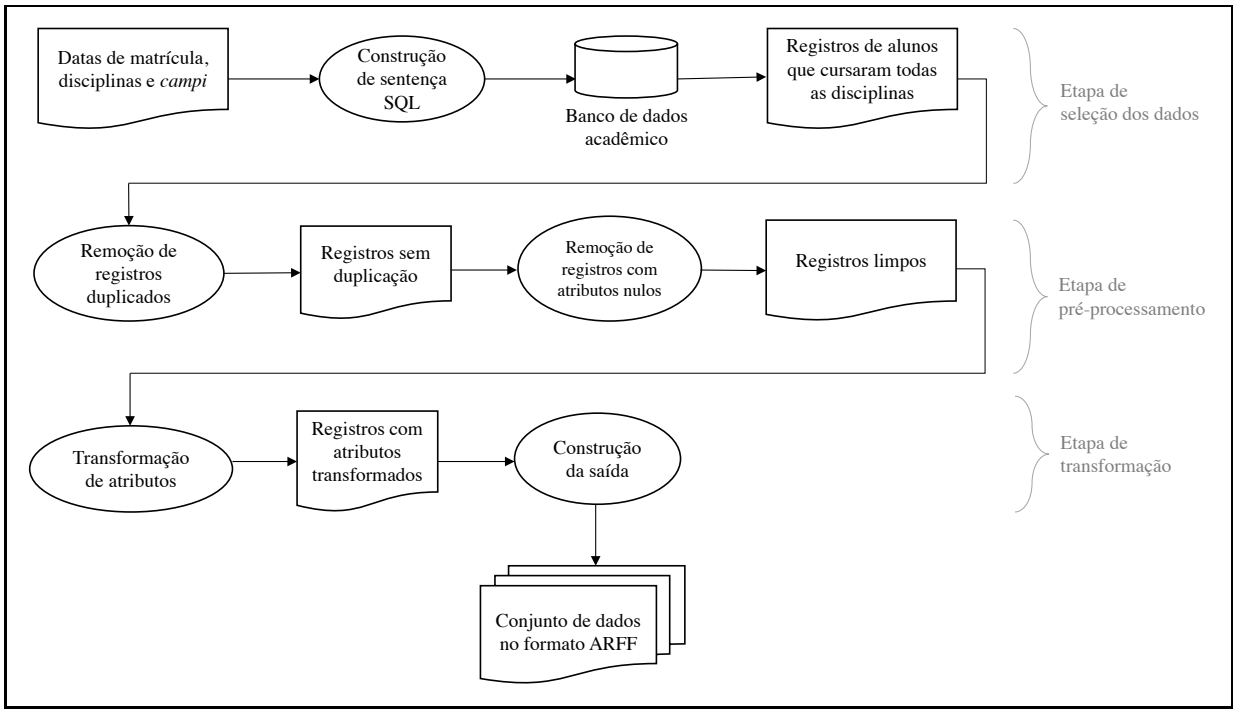
Na Figura 21 apresenta-se um fluxograma que traz, em uma visão geral, os passos executados pela AADM como parte da metodologia proposta neste capítulo. Nesta figura apresenta-se os passos divididos horizontalmente em três níveis, cada qual representando uma etapa da me-

todologia baseada no KDD.

Após a execução das etapas da metodologia, a AADM gera, como saída, um relatório contendo dados estatísticos dos registros e, também, um arquivo no formato ARFF com o conjunto de dados obtido e os valores dos atributos considerados. O formato ARFF é compatível com a ferramenta Weka e também com a metodologia proposta.

De posse do conjunto de dados gerado, já pré-processado e transformado de acordo com a metodologia para a tarefa de classificação, têm-se o necessário para executar as etapas posteriores da metodologia conforme apresentado no fluxograma da Figura 18, na página 50.

Figura 21 - Fluxograma dos passos gerais que a ferramenta AADM executa como parte da metodologia proposta.



Fonte: Autoria própria.

4.6 ETAPA DE APLICAÇÃO DO ALGORITMO DE *DATA MINING*

Após a execução das etapas anteriores, seja através dos passos individuais de seleção de dados, pré-processamento e transformação aplicados à tarefa de agrupamento, seja por meio da Aplicação de Apoio à *Data Mining* (AADM) aplicada à tarefa de classificação, a etapa de *data mining* é executada de forma unificada para ambas as tarefas. Isto ocorre pelo fato de que, em ambos os casos, a saída gerada nas etapas anteriores, no formato ARFF, é compatível com a etapa atual.

A etapa de *data mining* consiste na aplicação de um método ou algoritmo de *data mining* sobre o conjunto de dados resultante da execução das etapas anteriores, de acordo com a tarefa estabelecida previamente. Para cada uma das tarefas de agrupamento ou classificação, uma variedade de técnicas e algoritmos diferentes podem ser utilizados.

Nesta etapa, para a aplicação da *data mining*, devem ser definidos a técnica e o algoritmo a serem utilizados, respeitando a tarefa definida. Para a tarefa de agrupamento, podem ser definidos algoritmos baseados em *k-means*, baseados em densidade, probabilísticos, entre outros. Para a tarefa de classificação, os algoritmos podem ser divididos em algoritmos de árvore de decisão, baseados em redes neurais artificiais, entre outros. A descrição das tarefas de agrupamento e classificação, juntamente com o detalhamento das técnicas e dos algoritmos conhecidos foram apresentados no Capítulo 2, Seção 2.2.1.4, página 24.

Na metodologia proposta, os algoritmos definidos para execução da *data mining* são fornecidos pela ferramenta Weka. Esta ferramenta é, então, utilizada para carregar o conjunto de dados obtido das etapas anteriores e, de acordo com a definição da tarefa, é possível selecionar o algoritmo a ser utilizado. A ferramenta Weka apresenta duas interfaces distintas para a execução de *data mining* com as tarefas e agrupamento e classificação. Para cada tarefa, diferentes modos de execução do algoritmo definido podem ser utilizados.

A ferramenta Weka apresenta quatro modos de execução do algoritmo selecionado, de acordo com a tarefa a ser executada. Para o agrupamento, os modos de execução disponíveis são *Use training set*, *Supplied test set*, *Percentage split* e *Classes to clusters evaluation*. Para a tarefa de classificação, os modos de execução disponíveis são *Use training set*, *Supplied test set*, *Cross-validation* e *Percentage split*. Cada um dos modos de execução executam o algoritmo de *data mining* definido sobre o conjunto de dados de entrada de uma forma diferente, com características distintas de divisão dos dados em porções de treinamento e de teste. Cada modo de execução apresenta resultados específicos e, dependendo da maneira que a porção de treinamento é definida sobre os dados, podem ser obtidos resultados mais realistas. Os diferentes modos de execução, suas características específicas e a forma que tratam os conjuntos de dados são descritos em detalhes nos estudos de caso tratados neste trabalho, apresentados no Capítulo 5 seguinte.

Uma vez obtido o resultado da execução do algoritmo de *data mining* através da ferramenta Weka, este resultado deve ser interpretado e analisado de forma a se obter alguma informação sobre os dados, isto é, deve ser verificado se a *data mining* foi capaz de gerar um novo conhecimento. Este procedimento é realizado na próxima e última etapa da metodologia, descrita na seção a seguir.

4.7 ETAPA DE INTERPRETAÇÃO E AVALIAÇÃO

A última etapa da metodologia proposta prevê a interpretação e avaliação dos resultados obtidos após a execução do algoritmo de *data mining* da etapa anterior. Como os algoritmos têm sua execução na ferramenta Weka, as saídas de seus resultados são exibidas na interface da própria ferramenta. Os resultados obtidos podem ser exibidos de maneiras diferentes, dependendo da tarefa realizada. Logo, é necessário, antes de dar início ao processo de interpretação, analisar a forma em que o resultado está disposto.

Os resultados gerados a partir da tarefa de agrupamento são basicamente compostos por três partes, sendo elas: cabeçalho com informações gerais, dados do modelo gerado com base no algoritmo executado e listagem de *clusters* gerados. Um exemplo de resultado de saída de um algoritmo de agrupamento é apresentado na Figura 10, Capítulo 3, Seção 3.1.3, página 40. No cabeçalho são apresentados o nome do conjunto de dados carregado, a quantidade de registros existentes no conjunto de dados e o nome de todos os atributos selecionados que farão parte do algoritmo de *data mining*. Na sequência, são exibidos os dados da execução do algoritmo, como nome do algoritmo escolhido, o número de iteração realizadas e os *clusters* gerados, juntamente com a quantidade de instâncias alocados em cada *cluster* e a informação de seus *centroids*. Por fim, é apresentado um resumo da quantidade de *clusters* gerados e a quantidade de registros que compõem cada *cluster*.

Os resultados obtidos através da tarefa de classificação podem ser divididos em cinco partes: cabeçalho, informações do algoritmo executado, avaliação no conjunto de treinamento, estimativa de desempenho e matriz de confusão. Na Figura 9, Capítulo 3, Seção 3.1.3, página 39 pode ser visto um exemplo de saída da tarefa de classificação. O cabeçalho é idêntico ao cabeçalho exibido no resultado da tarefa de agrupamento, contendo inicialmente o nome do conjunto de dados, a quantidade de registros disponíveis para análise e o nome de todos os atributos que formam o conjunto de dados. Em seguida, são mostradas informações específicas do algoritmo escolhido que podem variar de acordo com as particularidades de cada algoritmo. Na parte de avaliação do conjunto de treinamento são apresentadas informações estatísticas sobre o treinamento da classificação, como número de instâncias classificadas correta e incorretamente, valores relacionados aos erros de classificação e o número total de instâncias. Na quarta parte do resultado é exibida a acurácia detalhada para a classificação, fornecendo informações como verdadeiros positivos, falsos positivos, entre outros. No final, é apresentada uma matriz de confusão, onde podem ser vistas todas as classes e o número de instâncias classificadas em cada uma delas.

A interpretação e avaliação do resultado da *data mining* são atividades que requerem atenção de alguém que tenha conhecimento sobre os dados analisados. Não basta que o responsável pela análise e interpretação entenda o conjunto de dados avaliado, é necessário que ele entenda o mini mundo no qual o conjunto de dados se encontra. Isso é necessário, pois os resultados considerados satisfatórios tendem a trazer algo inesperado, previamente desconhecido, ou seja, um conhecimento novo. Porém, muitas vezes, para esse conhecimento novo ser reconhecido, é preciso a análise de um especialista no assunto do mini mundo. Além do especialista no mini mundo, também é preciso que o profissional responsável pela *data mining*, que conhece a estrutura de armazenamento do banco de dados, também esteja envolvido nesta atividade. Pois, se o resultado analisado for considerado insatisfatório, este profissional tem condições de criar novas estratégias para a execução do algoritmo, ou mesmo inserir ou retirar um atributo do conjunto de dados analisado, até que seja alcançado um resultado suficiente.

Para o resultado ser considerado satisfatório é imprescindível que um novo conhecimento tenha sido descoberto ou que novos padrões tenham sido criados e também que essas descobertas sejam úteis para o dententor do banco de dados. Caso a *data mining* realizada não apresente nenhuma informação nova ou a nova informação não seja útil, é necessário retornar à quarta etapa da metodologia proposta para executar o algoritmo novamente, utilizando opções diferentes, ou então, retornar ao banco de dados, reiniciando todo o processo.

Os detalhes de experimentos da metodologia proposta executados em dados reais, para dois estudos de caso distintos, são apresentados no capítulo seguinte.

5 APLICAÇÃO DA METODOLOGIA PROPOSTA E RESULTADOS OBTIDOS

Neste capítulo são apresentados dois estudos de caso distintos nos quais se aplicam a metodologia proposta neste trabalho para a descoberta de conhecimento em dados acadêmicos de duas instituições de ensino: Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul (IFMS) e Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP). Em ambos os estudos de caso, a metodologia foi aplicada voltada às tarefas de agrupamento e classificação. São apresentados os detalhes de execução em cada caso e os resultados obtidos são discutidos em detalhes. É possível observar que a metodologia proposta é capaz de obter dados interessantes e previamente desconhecidos em ambos os estudos de caso.

5.1 ESTUDO DE CASO: IFMS

Nesta seção apresenta-se um estudo de caso envolvendo os dados do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul (IFMS), mais especificamente, dados do Curso Superior de Tecnologia em Sistemas para Internet. A metodologia proposta neste trabalho foi aplicada a partir do banco de dados acadêmico do IFMS. Este banco de dados, do tipo relacional, é composto por 321 tabelas, aproximadamente 18 milhões de registros e um gigabyte de tamanho. Como a instituição teve origem em 2009, este banco de dados está sendo utilizado há nove anos como a principal fonte de informações diretamente ligadas ao ensino no IFMS. O banco de dados integra o sistema acadêmico utilizado por todos os *campi* da referida instituição e contempla dados da secretaria acadêmica, como matrícula, alunos, disciplinas, histórico, entre outros.

O estudo de caso se divide em duas partes, a primeira, que realiza a aplicação da tarefa de agrupamento e a segunda, em que se aplica a tarefa de classificação. Ambas as partes do estudo de caso são descritas nas Seções 5.1.1 e 5.1.2.

5.1.1 Aplicação da tarefa de agrupamento

Esta parte do presente estudo de caso consiste em executar as etapas da metodologia proposta, utilizando a tarefa de agrupamento, a fim de descobrir características em comum entre alunos que são aprovados ou reprovados em Algoritmos.

Seguindo o fluxograma apresentado na Figura 18, do Capítulo 4, página 50, a metodologia proposta, quando voltada à aplicação da tarefa de agrupamento, realiza as etapas de seleção, pré-processamento e transformação dos dados de maneira manual, isto é, sem a utilização da AADM. Essas etapas iniciais são descritas a seguir.

5.1.1.1 Etapa de seleção dos dados

Como definido na metodologia deste trabalho, o primeiro passo para a realização do agrupamento é a seleção de dados. Com o acesso ao banco de dados, inicia-se a busca por tabelas e seus relacionamentos para o teste. Em um primeiro momento são selecionadas tabelas que contém dados cadastrais dos alunos e tabelas com informações dos alunos na disciplina de Algoritmos. Mais especificamente, dez atributos de aluno foram selecionados, sendo eles: sexo, data de nascimento, estado civil, cidade em que reside, quantidade de filhos, se possui emprego, média salarial e o tipo de instituição de origem do aluno (pública ou privada). Também foram selecionados os atributos que armazenam a situação do aluno (aprovado ou reprovado) na disciplina Algoritmos e o percentual de faltas do aluno na mesma disciplina. Todos os atributos selecionados podem ser vistos na Figura 22. Ao final da etapa de seleção dos dados, obteve-se um total de 318 registros para a realização das etapas subsequentes.

Figura 22 - Atributos selecionados para a tarefa de agrupamento utilizando os dados do IFMS.

Sexo	Quantidade_filhos	Instituicao_o
Idade	Trabalha	Porcentagem_faltas
E_civil	Media_Salarial	
Cidade	Situacao	

Fonte: Autoria própria.

5.1.1.2 Etapa de pré-processamento dos dados

Depois de selecionados os dados a serem trabalhados, passou-se para a etapa de pré-processamento dos dados. Nesta etapa foram eliminados os registros de todos os alunos que iniciaram a disciplina de Algoritmos, mas não a concluíram (desistentes da disciplina), alunos que trancaram o curso e alunos que obtiveram aproveitamento desta matéria. Além desses, foram excluídos registros com dados faltantes ou incorretos, por exemplo, registros com o campo sexo nulo ou registros com a data de nascimento incompatível com público do curso. Foram eliminados também registros duplicados, contemplando os casos de alunos que estavam erroneamente matriculados duas vezes com os mesmos dados na disciplina de Algoritmos. Em situações como

essa, foi mantido apenas um registro de cada aluno.

5.1.1.3 Etapa de transformação dos dados

A etapa de pré-processamento foi finalizada com 238 registros restantes para a etapa subsequente, a de transformação dos dados. Nesta etapa, dentre os dados tranformados estão o campo `data de nascimento`, cujos valores foram transformados em idade, o campo `situação do aluno`, cujos valores eram aprovado ou reprovado e foram transformados em sim (`S`) ou não (`N`). O campo `possui trabalho`, que tinha seus valores compostos por verdadeiro ou falso, assumiu os valores sim (`S`) ou não (`N`). O campo `cidade`, que armazenava o nome da cidade onde o aluno reside, passou a ter o valor TL para alunos que moram em Três Lagoas e Fora para os alunos que moram em qualquer outra cidade. As demais colunas sofreram alterações menos significativas, como, `estado civil`, cujos valores davam feminilidade às palavras, e `solteiro(a)` foi transformado simplesmente para `solteiro`. Em outras palavras, esse campo foi generalizado para se obter somente a caracterísca relevante do atributo.

Todos os dados foram transformados no intuito de torná-los mais sólidos para que os testes fossem feitos apenas com informações relevantes. Outra questão importante é que a realização dessa etapa possibilita otimizar o tempo de processamento do algoritmo a ser utilizado na etapa de *data mining*, pois a complexidade dos atributos disponíveis para análise foi minimizada.

5.1.1.4 Etapa de data mining

A quarta etapa da metodologia para descoberta de conhecimento em bancos de dados acadêmicos é a etapa de *data mining*, onde é executada a técnica e o algoritmo definidos. Esta é a etapa na qual os dados são, de fato, minerados. Em outras palavras, é onde a informação é esculpida para que se possa obter um novo conhecimento, uma informação nova e útil.

Na aplicação da tarefa de agrupamento deste estudo de caso, optou-se por trabalhar com a técnica de particionamento. Esta técnica divide o conjunto de dados de forma que os dados mais semelhantes entre si são agrupados e, ao mesmo tempo, estes ficam separados dos demais. O objetivo é que, através dos agrupamentos, seja possível notar características em comum dos alunos que reprovam na disciplina de Algoritmos, ou mesmo, qual característica dos alunos que são aprovados se destaca mais, a ponto de ser considerada importante na formação de um *cluster*. Em outras palavras, espera-se que, por meio da tarefa de agrupamento, seja possível visualizar algo novo relacionado a fatores que contribuem para reprovação dos alunos em Algoritmos.

Com o ojetivo da etapa de *data mining* definido e os dados para análise selecionados e

transformados, deve-se definir o algoritmo que melhor se adequa nesta situação. Há várias opções de algoritmos para executar a técnica de particionamento. O *k-means* é o método de agrupamento mais utilizado na atualidade, tanto em aplicações científicas quanto comerciais (BERKHIN, 2006; WU et al., 2008), sendo denominada de *SimpleKMeans* a implementação em Java do algoritmo *k-means* na ferramenta Weka. É importante observar que o algoritmo *SimpleKMeans* utiliza, como medida de distância entre os atributos numéricos, a medida de distância euclidiana. Essa medida, por sua vez, aplica automaticamente a normalização dos atributos numéricos ao realizar o agrupamento.

A escolha do método *k-means* se dá pela sua simplicidade e eficiência, sendo que, em comparação com diversos outros algoritmos de agrupamento, o *k-means* é o mais eficiente em termos de tempo de execução (MAIMON; ROKACH, 2010). Além disso, apesar de ter sido proposto há mais de 50 anos, o *k-means* ainda é um dos algoritmos de agrupamento mais utilizados (JAIN, 2010).

Este algoritmo possui vários parâmetros de entrada, como pode ser visto na Tabela 3. Os parâmetros estão disponíveis na ferramenta Weka com valores pré-definidos (valores *default*), e, com exceção do parâmetro `numClusters`, os demais parâmetros tiveram seus valores mantidos. Em `displayStdDevs`, o valor foi definido como falso, pois não se deseja visualizar dados adicionais estatísticos de desvio padrão dos atributos; o parâmetro `distanceFunction` foi definido como a distância euclidiana para comparação entre os dados, que é o cálculo de distância mais simples baseado no plano cartesiano; `dontReplaceMissingValues` foi utilizado como falso, pois é o valor pré-definido, mas como não há valores ausentes nos conjuntos de dados, esse parâmetro não é relevante para o estudo de caso; o parâmetro `initializationMethod` foi definido como aleatório para a escolha randômica inicial dos *centroids*; em `maxIterations`, o valor de iterações máximo é definido como 500, pois é um limite superior para o número de iterações no processo de recalcular os *centroids*; o valor de `numClusters` foi definido como 4, pois este é o número de *clusters* gerados que trouxe o agrupamento mais interessante dos dados após inúmeros testes realizados; em `preserveInstancesOrder`, o valor falso foi utilizado para que não seja necessário analisar as instâncias na ordem de entrada; o parâmetro `seed` foi mantido em 10, pois é o valor pré-definido para a geração de números aleatórios no algoritmo.

Todos os testes foram executados utilizando apenas sete dos dez atributos selecionados na etapa de seleção dos dados. Os atributos *quantidade de filhos*, *média salarial* e *percentual de faltas* foram eliminados, visto que não influenciaram nos resultados. Manteve-se, então, os campos *sexo*, *data de nascimento*, *estado cívil*, *cidade em que reside*, *possui trabalho*, *instituição de origem* (pública ou privada) e *situação do aluno* (aprovado ou reprovado) em Algoritmos.

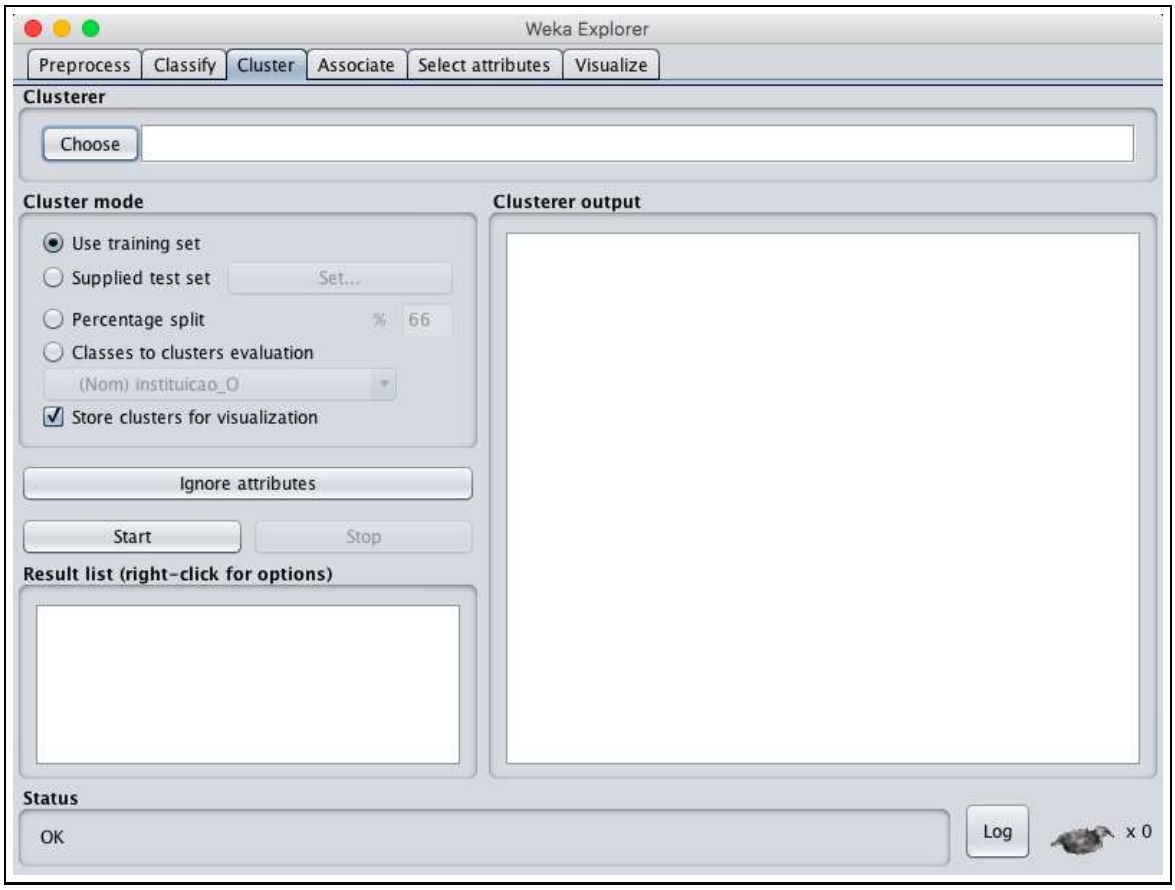
Tabela 3 - Principais parâmetros disponíveis para execução do algoritmo *SimpleKMeans*, suas descrições e valores utilizados durante o estudo de caso.

Parâmetro	Descrição	Valor utilizado
`displayStdDevs`	Exibe desvios padrão de atributos numéricos e a contagem de atributos nominais	`Falso`
`distanceFunction`	A função de distância a ser usada para comparar instâncias	Distância Euclidiana
`dontReplaceMissing-Values`	Substituir globalmente valores ausentes com a média/moda	`Falso`
`initializationMethod`	Define a forma de inicialização dos centros de cluster	`Aleatório`
`maxIterations`	Define o número máximo de iterações	`500`
`numClusters`	Define o número de clusters	`4`
`preserveInstances-Order`	Preserva a ordem das instâncias	`Falso`
`seed`	Valor aleatório de seed a ser usado	`10`

Fonte: Autoria própria.

A tela da ferramenta Weka para a geração de *cluster* é apresentada na Figura 23. Nesta figura pode ser observado que a definição do algoritmo é realizada através do botão `Choose` e também que é necessário escolher um modo de agrupamento dentre os disponíveis no painel `Cluster mode`. Cada modo de agrupamento é avaliado de maneira distinta e, para cada modo, é realizada a avaliação dos modelos de *clusters* gerados. A ferramenta Weka executa o algoritmo de *cluster* inicialmente para a geração dos *clusters* desejados. Cada instância do conjunto de dados é associada a um *cluster* de acordo com a proximidade de seus atributos com o centroide do *cluster*. Este processo é considerado o treinamento do algoritmo de *cluster*. Além disso, a ferramenta Weka também contabiliza a porcentagem de instâncias associadas a cada *cluster*. Este processo é considerado como o teste do agrupamento e é realizado de maneiras diferentes em cada modo de execução, descritos nesta seção.

Figura 23 - Interface inicial da execução da tarefa de agrupamento na ferramenta Weka.



Fonte: Autoria própria.

Em *Use training set*, o algoritmo de *cluster* é executado sobre o conjunto de dados carregado anteriormente e, então, as instâncias são associadas aos *clusters*, de acordo com o centroide de cada *cluster*, em um processo considerado treinamento do agrupamento. Posteriormente, a porcentagem de instâncias associadas a cada *cluster* é calculada como parte do teste do agrupamento. Em *Supplied test set*, a avaliação dos *clusters* é realizada de maneira semelhante ao modo anterior, mas é permitido que seja carregado um novo conjunto de dados para o teste do agrupamento. No modo *Percentage split*, a avaliação dos *clusters* é executada sobre o mesmo conjunto de dados de entrada, mas o treinamento e o teste do agrupamento são executados em porções diferentes do conjunto de dados. Por padrão, a ferramenta Weka define que a separação do conjunto de dados seja dois terços para treinamento e um terço para teste.

Um modo adicional oferecido pela ferramenta Weka para execução do agrupamento, denominado *Classes to clusters evaluation*, permite realizar testes considerando um atributo nominal específico dos registros de forma que, inicialmente, o algoritmo de *cluster* ignora este atributo e, assim, executa o agrupamento. A seguir, durante a fase de teste, são atribuídas classes para os

clusters com base na frequência em que o atributo especificado ocorre dentro de cada *cluster*. Por fim, é computado o erro de classificação de acordo com a atribuição anterior, e a matriz de confusão é mostrada. Esta é uma forma interessante de se obter um valor que define quão bem os *clusters* foram gerados de acordo com um atributo específico. Mas, neste caso, para se obter uma boa avaliação do agrupamento, é necessário que o conjunto de dados disponha de informações que classifiquem cada instância do conjunto de alguma forma.

Após realizadas todas as definições para a execução do algoritmo de *cluster*, pode-se dar início ao processo através do botão `Start` (Figura 23) e, então, é finalizada a etapa de *data mining*.

5.1.1.5 Etapa de interpretação e avaliação dos resultados

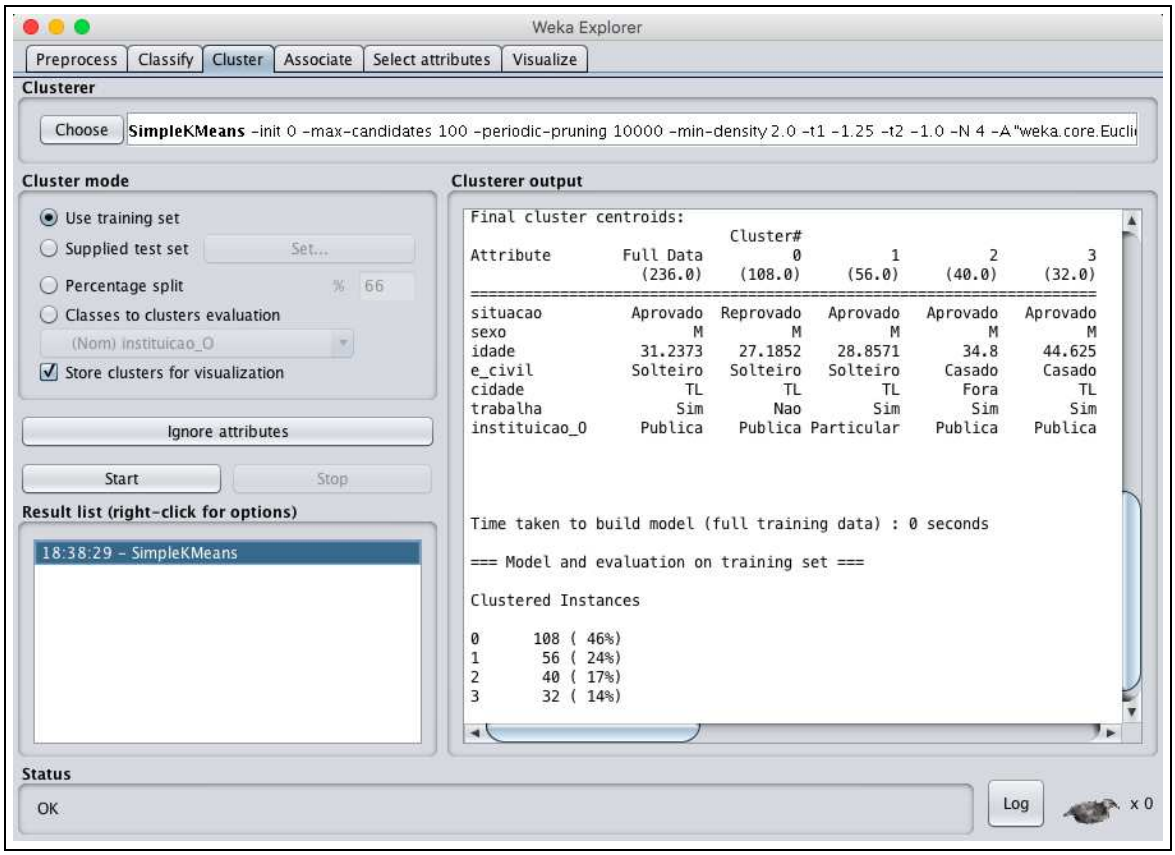
Na última etapa da metodologia verifica-se a possibilidade de interpretar o resultado obtido. Em caso positivo, é feita uma avaliação dos dados a fim de validar se o resultado é uma informação nova e se essa informação pode ser útil. Caso não seja possível interpretar o resultado ou o resultado não tenha gerado um conhecimento novo e útil, é necessário refazer a etapa de *data mining*, ou, então, refazer todas as etapas desde o início, até que seja encontrado um resultado satisfatório.

Neste estudo de caso foi necessário refazer todas as etapas inúmeras vezes e de diversas formas, já que os resultados, apesar de sempre serem interpretáveis, não eram satisfatórios, pois, na maioria das vezes eram apresentadas informações previamente conhecidas. Para que o resultado seja considerado satisfatório, é preciso que se obtenha uma informação anteriormente desconhecida. A seguir são analisados os melhores resultados de cada um dos quatros modos de agrupamento disponíveis na ferramenta Weka.

O primeiro resultado foi obtido através da opção *Use training set*, que pode ser visto na Figura 24. Na figura, pode-se observar que foram gerados quatro *clusters*. O *cluster* 0 é o mais populoso, com 46% dos registros. Para este *cluster* pode ser dado o nome de Desinteressados, pois a maioria dos alunos que o compõe são solteiros, não trabalham, são os mais novos, moram em Três Lagoas (não viajam diariamente) e, ainda assim, foram reprovados na disciplina de Algoritmos. O *cluster* 1, com 24% dos registros, pode ser chamado de Esperado, já que podem ser considerados novos, são solteiros, moram na cidade, foram aprovados em Algoritmos, trabalham e são os únicos que vieram de instituições de ensino particulares. O *cluster* 2, com 17% dos registros, pode ser chamado de Normal, pois grande parte dos alunos que o formam tem idade média, são casados, moram fora, trabalham e foram aprovados em Algoritmos. O último *cluster*, que conta com 14% dos registros, pode ser nomeado como Guerreiros, pois são

os mais velhos, casados, trabalham e foram aprovados em Algoritmos.

Figura 24 - Interface do resultado do *cluster* utilizando a opção `Use training set`.

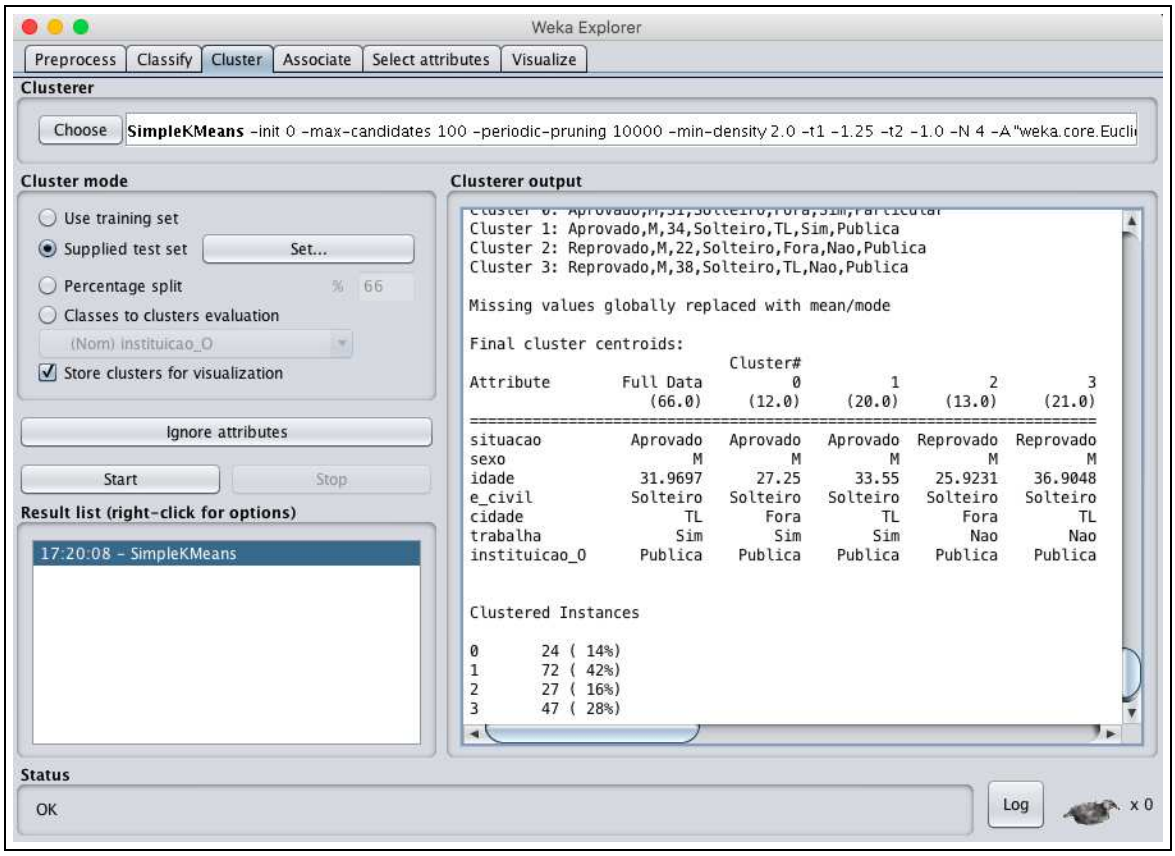


Fonte: Autoria própria.

Os resultados a partir da opção *Use training set* podem ser considerados interessantes, pois revelam que grande parte dos alunos que reprovam na disciplina de Algoritmos são jovens, solteiros, não trabalham e não viajam diariamente para estudar. Porém, o modo de agrupamento *Use training set* utiliza os mesmos dados para treinamento e teste, o que torna o resultado menos realista (WITTEN; FRANK; HALL, 2011).

O segundo modo de agrupamento utilizado foi o `Supplied test set`. Para realizar o teste nesse modo foi necessário dividir o conjunto de dados em dois, pois, a opção `Supplied test set` exige um conjunto de dados para treinamento e outro conjunto para o teste. Sendo assim, o conjunto de dados que contempla dados de alunos que passaram pelo IFMS entre os anos de 2011 e 2017 foi divido em dois subconjuntos, sendo o primeiro conjunto com dados de 2011 a 2013, com 66 registros, e o segundo conjunto com dados de 2014 a 2017, com 170 registros. O resultado gerado por meio da opção `Supplied test set` pode ser visto na Figura 25.

Figura 25 - Interface do resultado do *cluster* utilizando a opção `Supplied test set`.



Fonte: Autoria própria.

Neste modo de execução, primeiramente foi realizado o treinamento, para depois ser executado o teste do agrupamento. Pode-se observar na figura, na formação dos *clusters*, que na coluna `Full Data`, tem-se o valor 66, compreendendo a soma das quantidades dos *clusters* 0, 1, 2 e 3. Esta é a quantidade exata de registros que compõem o primeiro conjunto de dados (2011 a 2013) e este primeiro conjunto de dados foi utilizado para o treinamento do algoritmo. Para a execução do teste foi utlizado o segundo conjunto de dados (2014 a 2017), que compreende 170 registros. Na seção `Clustered Instances`, os registros do segundo conjunto de dados são apresentados com sua distribuição em cada *cluster*, bem como a porcentagem de cada grupo em relação ao número total de registros.

Ainda na Figura 25, pode se notar que os *clusters* formados são muito semelhantes. Aos *clusters* 0 e 1, que compreendem 56% dos registros testados, pode-se dar o nome de Normal, já que a maioria dos alunos que os compõem são solteiros, tem idade próxima da média, trabalham, são da cidade e foram aprovados na disciplina de Algoritmos. Os *clusters* 2 e 3, que somam 44% dos registros testados, podem ser chamados de Desinteressados, pois os registros que os compõem são, na maioria, alunos solteiros, de idade próxima da média, não trabalham,

são da cidade e não foram aprovados em Algoritmos.

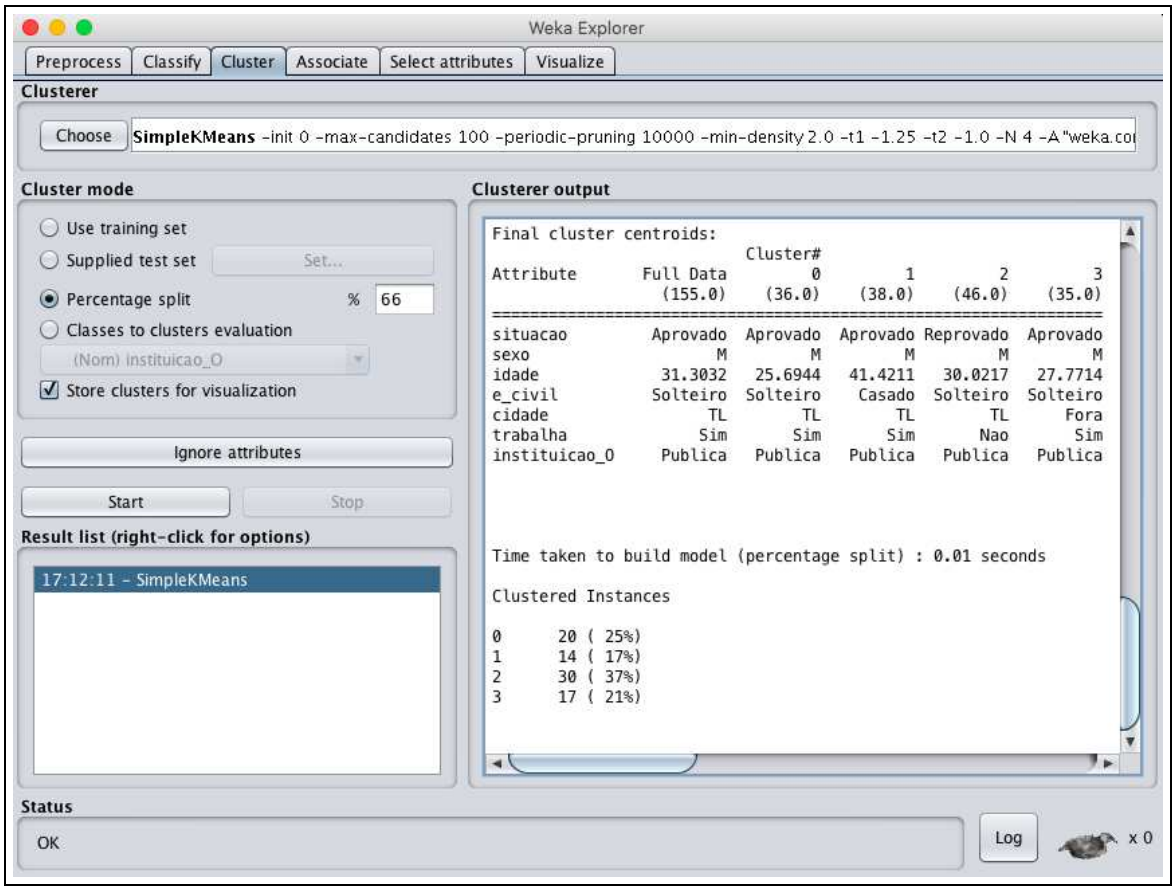
O resultado obtido utilizando a opção `Supplied test set`, pode ser facilmente interpretável. Porém não foi obtido um novo conhecimento, já que é esperado o fato de que alunos de média idade, que trabalham e são solteiros sejam, na maioria das vezes, aprovados em Algoritmos. Também não é novidade que os alunos mais velhos reprovem em Algoritmos. O fato mais interessante deste resultado está no *cluster* 2, pois este sugere que os alunos mais jovens, que não trabalham e que são de outra cidade acabam reprovando na disciplina de Algoritmo.

O próximo teste realizado foi utilizando o modo de agrupamento *Percentage split*, mantendo sua porcentagem para treinamento em 66%, que é o valor padrão do algoritmo. Alterações significativas neste valor não são interessantes por mudar demais a quantidade de dados para treinamento e para teste. Foi observado durante os testes que mudanças sutis neste valor produziram os mesmos resultados.

Na Figura 26, no painel `Clusters output` é possível observar que primeiramente é apresentada uma tabela com os resultados obtidos do treinamento do algoritmo, ou seja, o resultado da execução do treinamento do algoritmo utilizando apenas 66% dos dados. Durante a análise desses 2/3 dos dados, foram criados quatro *clusters* que representam instâncias com características específicas para que, em um próximo passo, o algoritmo analise o 1/3 dos dados restantes e agrupe-os em um dos quatro *clusters* disponíveis. Esta última execução é o processo de teste propriamente dito.

Ainda na Figura 26, pode ser visto o resultado do teste, apresentado em duas colunas na região denominada `Clustered Instances`. Nas colunas são apresentados, à esquerda, os diferentes *clusters* gerados e, à direita, a quantidade de instâncias agrupadas em cada *cluster*, juntamente com sua porcentagem em relação ao total de registros agrupados.

Figura 26 - Interface do resultado do *cluster* utilizando a opção `Percentage split`.



Fonte: Autoria própria.

O resutado gerado nesta execução de *data mining* pode ser interpretado da seguinte maneira: foram gerados quatro *clusters*, sendo o *cluster* 0 com 25% dos registros, o *cluster* 1 com 17% dos registros, o *cluster* 2 com 37% dos registros e o *cluster* 3 com 21% dos registros. O *cluster* 0 pode ser chamado de Esperado, pois são solteiros, da cidade, jovens, trabalham e foram aprovados em Algoritmos. O *cluster* 1 pode ser nomeado de Guerreiros, pois neste grupo estão os alunos mais velhos, que trabalham, que moram na cidade, que são casados e que foram aprovados em Algoritmos. O *cluster* 2 pode ser chamado de Desinteressados, já que são solteiros, moram na cidade, não trabalham, têm idade próxima da média e mesmo assim não foram aprovados em Algoritmos. E, por último, o *cluster* 3, que pode ser chamado de Normal, pois neste grupo estão os alunos de idade próxima da média, que trabalham, são solteiros, moram fora (viajam todos os dias para estudar) e que foram aprovados em Algoritmos. Pode ser observado neste resultado que os campos sexo e instituição de origem não foram decisivos em momento algum. Ou seja, estas duas características, dentre as sete analisadas, são as que menos influenciam no desempenho do aluno na disciplina de Algoritmos.

Este resultado foi considerado satisfatório, revelando um novo conhecimento, uma vez que era esperado que alunos de idade mais avançada, que são casados e trabalham, fossem os responsáveis pelo alto número de reprovação na disciplina de Algoritmos. Ao mesmo tempo, era esperado que alunos que residem na cidade, que não trabalham, que são solteiro e que têm idade média fossem, em sua maioria, aprovados em Algoritmos. Ou seja, estes resultados mostram exatamente o contrário do que era esperado. Pode-se concluir que os alunos mais velhos, apesar de terem menos tempo para estudar devido à família e ao trabalho, são os que mais se dedicam à disciplina. Já os alunos que têm mais tempo livre para os estudos e têm idade média não estão muito precupados com seu desempenho na disciplina de Algoritmos.

O resultado obtido, além de ser considerado um novo conhecimento, também pode ser considerado útil, pois a informação permite que a instituicão possa promover atividades de motivação para os alunos que foram agrupados no *cluster* 2. Além disso, pode-se ofertar aulas de reforço no contraturno para esse alunos, já que eles não trabalham. Por fim, de posse das características obtidas, outras estratégias podem ser criadas e direcionadas a grupos específicos pelo coordenador do curso em questão.

O quarto modo de agrupamento para execução do teste é denominado `Classes to clusters evaluation`. Este modo permite que um atributo determinado do conjunto de dados de entrada seja utilizado para avaliar o resultado do agrupamento, de modo que as instâncias são agrupadas sem que seja considerado o atributo escolhido. Assim, cada instância alocada em um *cluster* pode ser classificada de forma correta ou incorreta, de acordo com o valor do atributo de cada instância. No entanto, o conjunto de dados em estudo não possui previamente informações que permitam categorizar cada instância de forma explícita, isto é, não existe a informação, para cada aluno, de qual é a classe determinada por suas características (desinteressado, guerreiro, etc). Assim, o modo `Classes to clusters evaluation` não foi considerado neste estudo de caso.

5.1.2 Aplicação da tarefa de classificação

Durante um estudo aprofundado do banco de dados do IFMS, observou-se que a disciplina Linguagem de Programação 1 (LP1), que é ofertada para as turmas de segundo semestre do Curso de Tecnologia em Sistemas para Internet, acusa muitas reprovações. Além disso, é sabido que esta disciplina é extremamente importante para o curso e que, quando o aluno reprova na disciplina, a chance de ser necessário prorrogar sua formatura é muito grande. O fato é agravado quando o aluno reprova mais de uma vez na mesma disciplina e acaba evadindo-se do curso. Com base nessa informação, escolheu-se utilizar a tarefa de classificação, para que, de alguma

forma, seja possível estimar quando um novo aluno reprovará nesta disciplina, para que uma atitude preventiva possa ser tomada antes da reprovação.

Esta segunda parte do presente estudo de caso consiste em executar as etapas da metodologia para descoberta de conhecimento em bancos de dados acadêmicos, utilizando a tarefa de classificação, com o objetivo de predizer alunos propensos a reprovar na disciplina de LP1, com base na aprovação ou não desses alunos em algumas disciplinas chaves ofertadas no mesmo curso. A classificação toma como base, ainda, atributos adicionais relacionados aos dados seus pessoais, como idade, sexo e estado civil.

5.1.2.1 Etapa de aplicação da AADM

Para a tarefa de classificação, a primeira etapa da metodologia proposta consiste na aplicação da AADM, que se inicia na coleta de dados a partir do banco de dados acadêmico. Utilizando a AADM, foram selecionadas várias tabelas referentes ao dados pessoais dos alunos, tabelas com informações das disciplinas e tabelas que contêm os dados necessários para compor um histórico escolar. Depois de realizadas essas seleções, foram separados, em cada uma das tabelas, os atributos considerados interessantes para este estudo.

Especificamente, foram selecionadas disciplinas que influenciam, segundo o Projeto Pedagógico do Curso, no desempenho dos alunos na disciplina de LP1, sendo elas: Fundamentos Matemáticos, Lógica Digital, Desenvolvimento Web 1 e Algoritmos. Estes atributos armazenam o resultado do aluno na disciplina (aprovado ou reprovado). Também foram selecionados os atributos `data de nascimento`, `estado civil` e `sexo` de cada aluno. A Figura 27 apresenta todos os atributos selecionados.

Figura 27 - Atributos selecionados para a tarefa de classificação utilizando os dados do IFMS.

```
Programacao      Logica           Idade
Web1             Algoritmos       Solteiro
Matematica       Sexo
```

Fonte: Autoria própria.

Na sequência, a aplicação AADM realiza correções e ajustes, de forma automática, nos dados que apresentam alguma anomalia. Por exemplo, casos em que um mesmo aluno se encontra matriculado duas vezes na mesma disciplina, no mesmo semestre, além de dados faltantes como alunos matriculados, mas não relacionados a nenhuma disciplina.

Posteriormente, a aplicação AADM realiza o passo em que atributos são ajustados em sua

forma de exibição, mas não em seu valor significativo. Neste caso, os atributos referentes à situação do aluno nas disciplinas (aprovado ou reprovado) tiveram seus valores transformados em `S` (Sim) ou `N` (Não). O atributo estado civil foi nomeado como `solteiro` e também teve seus valores transformados `S` (Sim) ou `N` (Não) e o atributo sexo teve seus valores transformados em `M` (Masculino) e `F` (Feminino). O atributo data de nascimento foi renomeado para `idade` e seus valores deixaram de ser a data de nascimento do aluno e passaram a corresponder à idade de cada aluno.

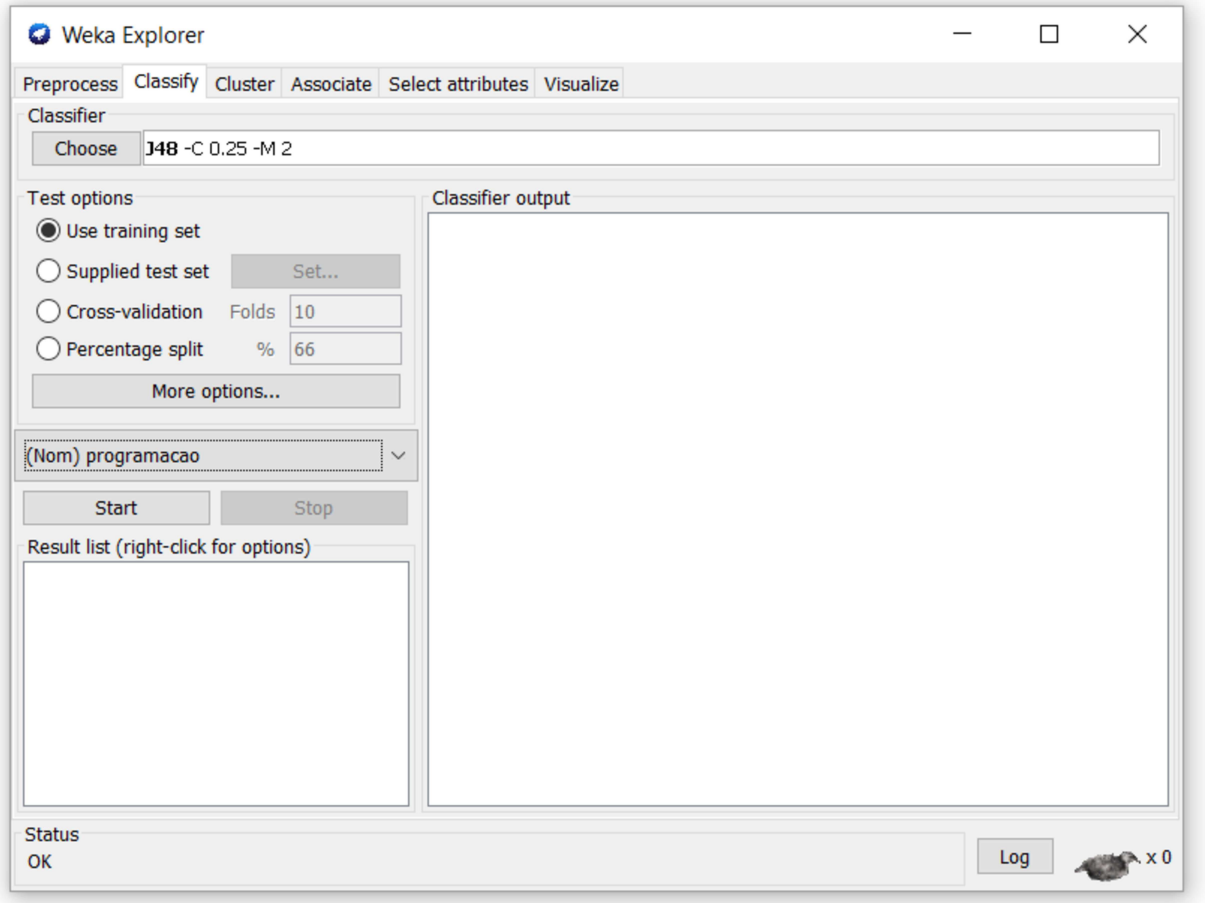
Ao final da execução, aplicação AADM gerou um conjunto de dados que corresponde a 234 registros de alunos que passaram pelo curso em questão entre 2011 e 2017. A aplicação garante que o conjunto de dados obtido consista apenas de alunos que cursaram, até o final, todas as disciplinas aqui analisadas.

5.1.2.2 Etapa de data mining

A partir do conjunto de dados obtido no passo anterior, a sequência da metodologia é a aplicação da *data mining*, mais especificamente, da tarefa de classificação, cujo objetivo é encontrar uma relação entre os atributos e a classe em que se deseja classificar os dados, para que, no futuro, esta relação possa ser usada para classificar (estimar) dados novos e desconhecidos (GIUDICI, 2005). Foi definida a utilização da técnica de árvore de decisão para a execução da tarefa de classificação nesta etapa, pois, segundo Witten, Frank e Hall (2011), este é um dos métodos recomendados para a tarefa de classificação. Além disso, a árvore de decisão apresenta a vantagem de ser um método do qual se pode extrair e interpretar as características que levaram à construção da árvore, o que pode ser considerado importante para o processo de descoberta de conhecimento de forma geral.

Na Figura 28 é apresentada a tela de execução da classificação pelo Weka. Na figura pode ser visto que o algoritmo escolhido foi o algoritmo de árvore de decisão denominado J4.8, que é a implementação *open-source* em Java do algoritmo C4.5. Esta implementação, desenvolvida pelos próprios autores do Weka, oferece uma variedade de parâmetros configuráveis, relacionados ao comportamento da árvore de decisão gerada (KAUR; CHHABRA, 2014). Os principais parâmetros do algoritmo são apresentados e detalhados na Tabela 4.

Figura 28 - Interface do *Explorer* para a tarefa de classificação.



Fonte: Autoria própria.

Na Figura 28, são apresentados os modos de execução `Use training set`, `Supplied test set`, `Cross-validation` e `Percentage split`. Neste estudo os testes foram realizados com todos os modos de execução aqui citadas. Ainda na Figura 28, também é apresentada a opção de seleção do atributo alvo, que, no caso, é o atributo `programacao`. Isso significa que o que se deseja classificar é se o aluno será aprovado ou não na disciplina de Linguagem de Programação (LP1). Finalmente, na figura, ainda pode ser visto o quadro denominado `Classifier output`, onde são exibidos os resultados das execuções realizadas.

Tabela 4 - Principais parâmetros disponíveis para execução do algoritmo J4.8, suas descrições e valores utilizados durante o estudo de caso.

Parâmetro	Descrição	Valor utilizado
`confidenceFactor`	Fator de confiança usado para a poda (valores menores resultam em mais poda)	`0,25`
`debug`	Se verdadeiro, o classificador pode retornar saídas adicionais ao console	`Falso`
`minNumObj`	Número mínimo de instâncias por folha	`2`
`numDecimalPlaces`	O número de casas decimais a ser usado nos números utilizados na saída do resultado	`2`
`saveInstanceData`	Define se salvará os dados de treinamento para visualização posterior	`Falso`
`unpruned`	Determina se executará poda	`Falso`, `Verdadeiro`

Fonte: Autoria própria.

Na Tabela 4 é possível observar que o valor utilizado para o parâmetro `unpruned` é apresentado como verdadeiro e falso. Conforme a descrição apresentada na tabela, este parâmetro determina se o algoritmo de construção da árvore de decisão executará ou não a poda na árvore. Nesta parte do estudo de caso, todas as execuções do algoritmo de árvore de decisão foram realizadas com ambos os tipos de árvore (com poda e sem poda). Os demais parâmetros são mantidos com os valores pré-definidos trazidos pela ferramenta Weka. O parâmetro `confidenceFactor` é mantido em 0,25 pois é um valor relativamente baixo e que gera podas com capacidades de generalização; `debug` foi mantido como falso, pois não se deseja visualizar etapas intermediárias na saída da execução do algoritmo; `minNumObj` foi mantido em 2 para que não fossem geradas árvores cujas folhas tivessem uma instância apenas; `numDecimalPlaces` define em 2 o número de casas decimais para facilitar a visualização dos resultados; `saveInstanceData` foi utilizado como falso pois não era necessário armazenar os resultados para visualização posterior; e `unpruned` foi utilizado ora como verdadeiro, ora como falso, e ambos os casos foram descritos neste estudo de caso.

Os detalhes da classificação realizada em todos os modos de execução e os resultados dos testes são apresentados na próxima seção.

5.1.2.3 Etapa de interpretação e avaliação dos resultados

Definidos a tarefa, a técnica e o algoritmo para a *data mining*, há ainda várias combinações possíveis para executar o algoritmo de classificação na ferramenta Weka. Todas as opções de execução disponíveis na ferramenta foram utilizadas: `Use Training Set`, `Supplied Test Set`, `Cross-validation` e `Percentage split`. A opção `Use Training Set` permite que o conjunto de dados carregado seja utilizado para treinar o algoritmo e gerar um modelo de classificação. `Supplied Test Set` permite que um novo conjunto de dados seja carregado para testar a classificação com o modelo gerado. A opção `Cross-validation` divide o conjunto de dados carregado de forma aleatória para realizar o treinamento e executar o teste da classificação. Por fim, a opção `Percentage split` divide o conjunto de dados em duas partes com uma proporção escolhida, sendo que uma parte é utilizada para treinamento e a outra parte é utilizada para o teste da classificação.

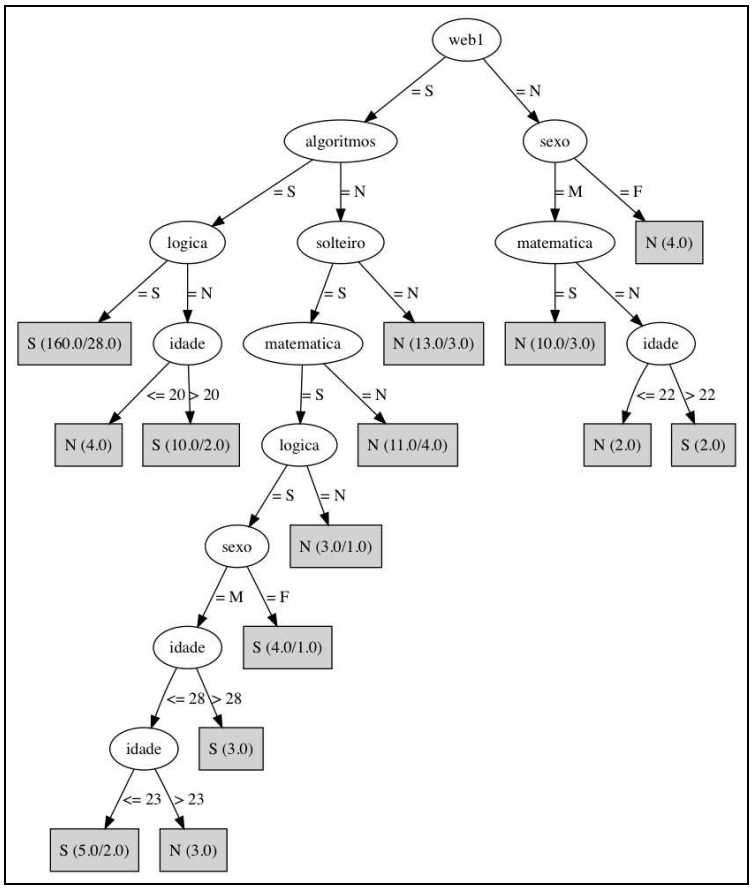
Foram realizadas duas execuções do algoritmo J4.8 para cada opção descrita anteriormente, totalizando oito execuções. Em todas as opções o atributo alvo foi sempre o mesmo: `programacao` e os atributos base também não mudaram, foram eles: `algoritmos`, `web1`, `matematica`, `logica`, `solteiro`, `idade` e `sexo`. Para cada opção, o algoritmo teve seu parâmetro `prune` alterado entre `verdadeiro` e `falso`, isto é, o algoritmo de classificação gerou versões da árvore de decisão com poda e sem poda. A poda em uma árvore de decisão tem o objetivo de gerar uma árvore mais simples e compacta, com precisão próxima da árvore original. Além disso, a poda permite aumentar a capacidade de generalização da árvore de decisão (MAIMON; ROKACH, 2010).

Na execução `Use Training Set`, foram geradas árvores de decisão com poda e sem poda. Neste caso o sucesso foi maior na versão da árvore sem poda, que é apresentada na Figura 29, em que todos os atributos foram considerados. A matriz de confusão da versão sem poda apresenta 81,2% de acerto, enquanto a matriz de confusão da árvore com poda apresenta 78,6% de acerto, como apresentado na Tabela 5.

Na matriz de confusão da árvore sem poda, apresentada na Tabela 5 (1), é possível observar que 151 alunos foram classificados como aprovados e 39 como reprovados. Ainda é possível observar que 11 alunos foram classificados como reprovado mas, no conjunto de dados estes registros constam como aprovado, e que 33 alunos foram classificados como aprovados, porém, são reprovados no conjunto de dados. Em outras palavras, o algoritmo classificou 190 instâncias corretamente, ou seja, uma taxa de acerto de 81,2%. Na matriz de confusão da árvore com poda, apresentada na Tabela 5 (2), é possível observar que a taxa de acerto é menor, com valor de 78,6%.

Ambas as taxas de acerto são consideradas satisfatórias, mas o uso da opção `Use Training Set` indica que os mesmos dados utilizados para realizar o treinamento são utilizados para realizar a tarefa de classificação, gerando-se assim um resultado otimista para a classificação (WITTEN; FRANK; HALL, 2011). Isto é, o teste realizado sobre os mesmos dados utilizados durante o treinamento da classificação não garante uma representação realista do resultado.

Figura 29 - Representação gráfica da árvore de classificação (sem poda) utilizando a opção `Use Training Set`.



Fonte: Autoria própria.

Tabela 5 - Resultados das execuções `Use Training Set` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Use Training Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	190	81,2%
Instâncias Classificadas Incorretamente	44	18,8%

Matriz de confusão		
a	**b**	<-- Classificado como
151	11	a = S (aprovado)
33	39	b = N (reprovado)

(1)

Use Training Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	184	78,6%
Instâncias Classificadas Incorretamente	50	21,4%

Matriz de confusão		
a	**b**	<-- Classificado como
142	20	a = S (aprovado)
30	42	b = N (reprovado)

(2)

Fonte: Autoria própria.

Para a realização dos testes utilizando a opção `Supplied Test Set`, é necessário separar o conjunto de dados em duas partes, sendo que a primeira parte é utilizada para o treinamento do algoritmo e para a construção da árvore e a segunda parte dos dados é utilizada para realizar o teste da classificação. Para a primeira parte, foi utilizado um subconjunto dos dados referentes às matrículas entre 2011 e 2013 e, para a segunda parte, um subconjunto referente às matrículas de 2014 a 2017.

Neste caso, também foram geradas as versões das árvores com poda e sem poda no algoritmo J4.8. Os resultados das classificações são apresentados na Tabela 6, onde pode ser visto que a versão da árvore com poda obteve um melhor resultado, com uma taxa de acerto de 73,1%. Na matriz de confusão pode ser visto que 104 alunos foram classificados corretamente como aprovados e 21 alunos foram classificados corretamente como reprovados, restando 46 alunos classificados incorretamente.

A versão da árvore de decisão gerada com poda, apresentada na Figura 30, levou em consideração apenas o atributo `algoritmos` para a tomada de decisão da classificação. Em outras palavras, o modelo de classificação considera apenas que, se o aluno for aprovado em Algoritmos, ele também será aprovado em LP1.

Os resultados obtidos na execução da opção `Supplied Test Set` podem ser considerados bons, uma vez que a divisão do conjunto de dados, para treinamento e teste, oferece um resultado mais fiel à realidade.

Tabela 6 - Resultados das execuções `Supplied Test Set` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Supplied Test Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	122	71,3%
Instâncias Classificadas Incorretamente	49	28,7%

Matriz de confusão		
a	**b**	<-- Classificado como
111	5	a = S (aprovado)
44	11	b = N (reprovado)

(1)

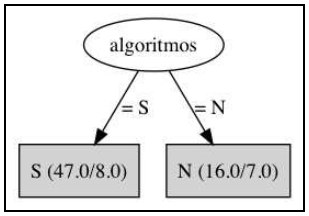
Supplied Test Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	125	73,1%
Instâncias Classificadas Incorretamente	46	26,9%

Matriz de confusão		
a	**b**	<-- Classificado como
104	12	a = S (aprovado)
34	21	b = N (reprovado)

(2)

Fonte: Autoria própria.

Figura 30 - Representação gráfica da árvore de classificação (com poda) utilizando a opção `Supplied Test Set`.



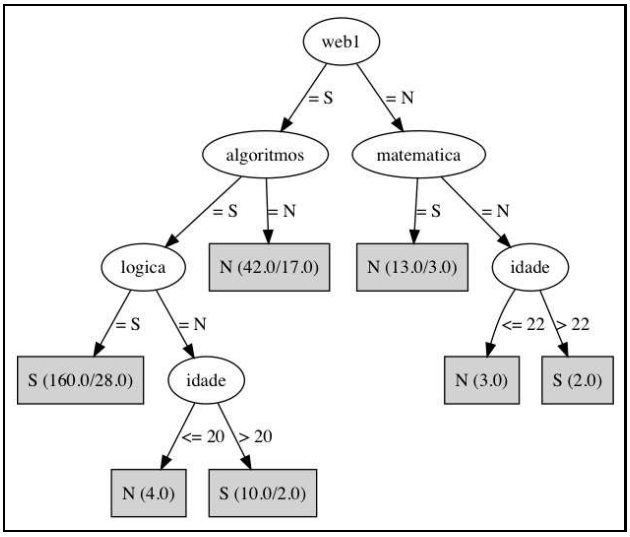
Fonte: Autoria própria.

Além dos testes ulilizando as opções `Use Training Set` e `Supplied Test Set`, também foram feitos testes com a opção `Cross-validation`, que realiza o treinamento e o teste de classificação em um mesmo conjunto de dados. Neste caso foi configurado o parâmetro `Fold` em 10, o que significa que o algoritmo de classificação divide o conjunto de dados em dez partes, utilizando, inicialmente, nove partes para realizar o treinamento e uma parte para executar o teste. Em seguida, é utilizada uma parte diferente para teste e um conjunto diferente de nove partes para treinamento. O processo é repetido, substituindo as partes entre treinamento e teste, até que todas as dez partes tenham sido utilizadas para teste.

Como não foram fixados subconjuntos de treinamento e teste específicos, o método `Cross-validation` fornece estimativas de erro mais realistas para a construção da árvore de classificação. Neste teste foi utilizado o conjunto de dados completo e, como nos demais testes, também foram verificadas as opções com poda e sem poda da árvore. Na Tabela 7 podem ser vistos os resultados das classificações nas versões sem poda (1) e com poda (2). Pelas características de generalização da árvore com poda, esta versão gerou melhor resultado, obtendo uma taxa de acerto de 73,5% na classificação, ou seja, 172 dos 234 registros foram classificados

corretamente, como pode ser visto na Tabela 7 (2). A árvore com poda possui seis nós internos e pode ser vista na Figura 31, em que os atributos considerados foram: `web1`, `algoritmos`, `matematica`, `logica` e `idade`.

Figura 31 - Representação gráfica da árvore de classificação (com poda) utilizando a opção `Cross-validation`.



Fonte: Autoria própria.

Tabela 7 - Resultados das execuções `Cross-validation` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão motram a distribuição de instâncias classificadas correta e incorretamente.

Cross-validation: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	166	70,9%
Instâncias Classificadas Incorretamente	68	29,1%

Matriz de confusão		
a	**b**	<-- Classificado como
142	20	a = S (aprovado)
48	24	b = N (reprovado)

(1)

Cross-validation: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	172	73,5%
Instâncias Classificadas Incorretamente	62	26,5%

Matriz de confusão		
a	**b**	<-- Classificado como
141	21	a = S (aprovado)
41	31	b = N (reprovado)

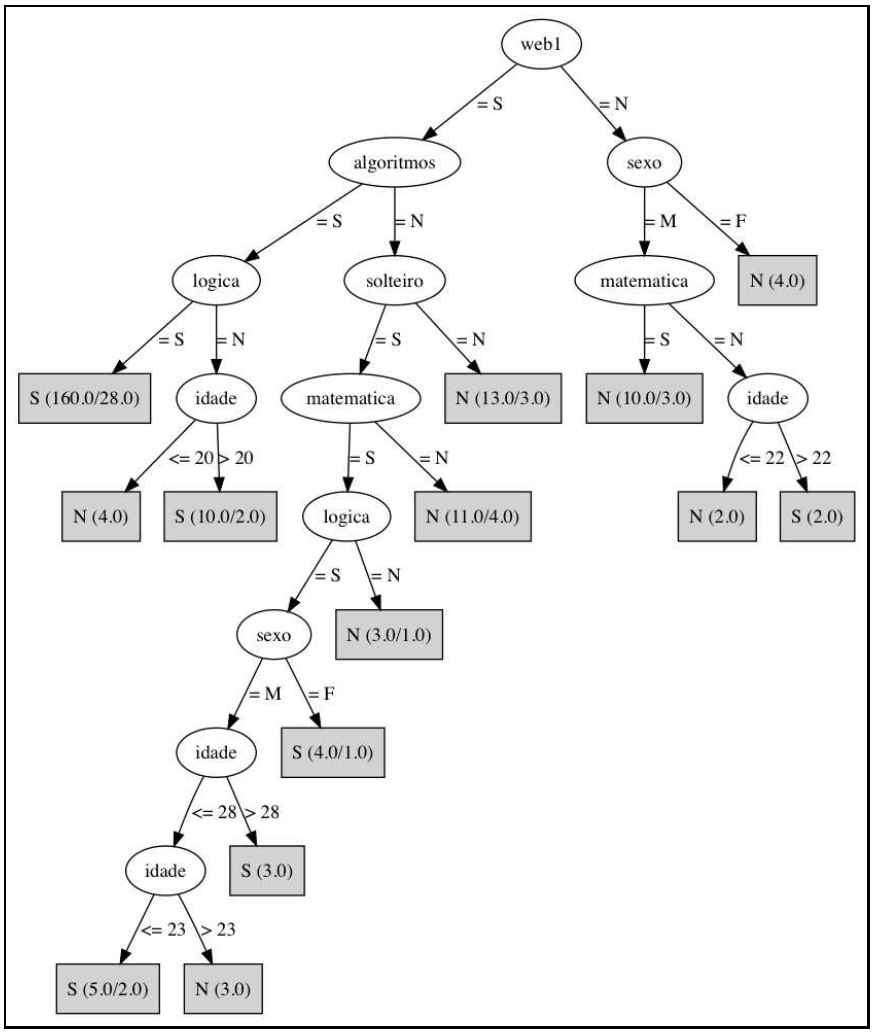
(2)

Fonte: Autoria própria.

Por fim, foram feitos testes com a opção `Percentage Split`. Este modo de execução, como mencionado anteriormente, separa uma parte do conjunto de dados para treinamento e executa o teste com a parte restante dos dados. Por padrão, é sugerido a realização do treinamento com 66% dos dados, sendo que este valor pode ser alterado. Porém, neste estudo o melhor resultado foi alcançado mantendo a sugestão inicial da ferramenta Weka. No caso, o algoritmo também foi executado nas versões com e sem a poda da árvore e o melhor resultado

foi obtido com a versão sem poda, que classificou 75% dos registros testados corretamente, enquanto a versão da árvore com pada teve uma taxa de acerto de 73,8%. Esses dados podem ser vistos na Tabela 8. A árvore sem poda contempla todos os atributos, como pode ser visto na Figura 32. É possível verificar, ainda, que a árvore gerada nesta execução é a mesma gerada no modo `Use training set`, na Figura 29. Este fato ocorre pois a ferramenta Weka sempre disponibiliza a árvore gerada a partir do conjunto de dados completo carregado inicialmente, não contabilizando as divisões para treinamento e teste ocorridas nos diferentes modos de execução.

Figura 32 - Representação gráfica da árvore de classificação (sem poda) utilizando a opção `Percentage split`.



Fonte: Autoria própria.

Tabela 8 - Resultados das execuções `Percentage split` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão motram a distribuição de instâncias classificadas correta e incorretamente.

Percentage Split: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	60	75,0%
Instâncias Classificadas Incorretamente	20	25,0%

Matriz de confusão		
a	**b**	<-- Classificado como
46	5	a = S (aprovado)
15	14	b = N (reprovado)

(1)

Percentage Split: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	59	73,8%
Instâncias Classificadas Incorretamente	21	26,2%

Matriz de confusão		
a	**b**	<-- Classificado como
46	5	a = S (aprovado)
16	13	b = N (reprovado)

(2)

Fonte: Autoria própria.

Os resultados mostram que a aplicação da *data mining* nos dados acadêmicos estudados pode obter bastante sucesso, possibilitando a descoberta de conhecimento em dados que eram simplesmente armazenados. Dentre as informações extraídas, verificou-se que o desempenho do aluno na disciplina de Páginas Web 1 diz mais sobre o resultado que ele terá em Linguagem de Programação 1 do que a própria disciplina de Algoritmos, o que não era esperado. Isto pode ser verificado analisando as Figuras 29, 30 e 31, em que o atributo `web1` aparecem mais vezes como raiz do que o atributo `algoritmos`, isto é, possui maior influência sobre o atributo alvo.

Ao final das execuções dos testes realizados com a tarefa de agrupamento utilizando os dados do IFMS conclui-se que o melhor resultado foi obtido através do modo de agrupamento `Percentage split`, como pode ser visto na Tabela 9.

Tabela 9 - Resumo dos desempenhos das opções de teste de classificação com dados do IFMS.

Opção de teste	**% de classificação correta**	**% de classificação incorreta**	**Árvore**
Use training set	81,2	18,8	Sem poda
Supplied test set	73,1	26,9	Com poda
Cross-validation	73,5	26,5	Com poda
Percentage split	75,0	25,0	Sem poda

Fonte: Autoria própria.

5.2 ESTUDO DE CASO: UNESP

Nesta seção apresenta-se um segundo estudo de caso, onde também são aplicadas as tarefas de agrupamento e classificação em dados acadêmicos. A metodologia proposta neste trabalho para a descoberta de conhecimento em bancos de dados acadêmicos foi aplicada aos dados da

Universidade Estadual Paulista "Júlio de Mesquita Filho" (UNESP), mais especificamente, aos dados dos alunos do Curso de Graduação em Engenharia Elétrica do Câmpus de Ilha Solteira. Neste estudo, as análises realizadas podem ser consideradas socioeconômicas, já que envolvem majoritariamente dados tanto da ordem econômica como social.

No presente estudo, os dados foram obtidos de forma distinta em relação ao estudo de caso anterior, sem acesso direto ao banco de dados da instituição. Os dados utilizados foram extraídos por usuário final através do sistema que gerencia os dados acadêmicos da UNESP. Os dados utilizados correspondem ao período de 2006 a 2016, e compreendem os alunos matriculados em todos os semestres do curso. Foram gerados vários relatórios com dados referentes aos alunos, contendo nome, coeficiente de rendimento, raça, data de matrícula, quantidade de semestres cursados, porcentagem de aproveitamento de disciplinas, entre outros. Os relatórios foram exportados pelo sistema no formato de arquivo CSV e posteriormente foram transformados nos conjuntos de dados utilizados para aplicação da metodologia.

O estudo de caso se divide em duas partes, a primeira realiza a aplicação da tarefa de agrupamento e, na segunda parte, é aplicada a tarefa de classificação. Ambas as partes do presente estudo de caso são descritas nas Seções 5.2.1 e 5.2.2, a seguir.

5.2.1 Aplicação da tarefa de agrupamento

Esta primeira parte do estudo que envolve os dados da UNESP apresenta a execução das etapas do processo da metodologia apresentada no Capítulo 4, aplicando a tarefa de agrupamento. O objetivo aqui é descobrir características comuns entre alunos que têm bom rendimento acadêmico e, ao mesmo tempo, revelar quais características mais influenciam no momento de separar os alunos que têm melhor rendimento daqueles cujo desempenho é considerado instatisfatório.

5.2.1.1 Etapa de seleção dos dados

Na primeira etapa da metodologia espera-se uma análise aprofundada da estrutura de relacionamentos do banco de dados, considerando que a interpretação dos dados possa permitir a execução das etapas do processo de forma estratégica. Porém, neste caso, como já mencionado, o acesso aos dados se deu por meio de relatórios do sistema acadêmico, ocasionando assim uma análise relativamente mais restrita dos dados.

A seleção de dados foi feita explorando os arquivos CSV gerados pelo sistema acadêmico da UNESP. Salienta-se que, para a obtenção do conjunto de dados em arquivo CSV, foi necessário executar diversos processos manualmente para a união de informações dispostas em múltiplos

arquivos, uma vez que diferentes arquivos referentes a relatórios distintos foram obtidos. A junção de todos os dados para um mesmo arquivo necessita ser realizada de forma cautelosa para garantir que dados consistentes e confiáveis pudessem estar disponíveis no conjunto de dados final. Em outras palavras, foi preciso garantir que a tarefa de união de dados distribuídos em diversos arquivos, mas referentes aos mesmos alunos, resultasse em um conjunto onde a integridade dos dados era preservada.

De posse do conjunto de dados selecionado, foram definidos nove campos para serem utilizados na tarefa de agrupamento: `Classificacao`, que é a posição de entrada do aluno no vestibular; `CR`, que é o campo que informa o coeficiente de rendimento do aluno; `DiscAproveitada`, campo que armazena a porcentagem de aproveitamento das disciplinas cursadas pelo aluno; `Presenca`, que refere-se à média da porcentagem de presença do aluno nas disciplinas; `Sexo`, com o sexo do aluno; `Cor`, referindo-se à etnia do aluno; `Escola`, campo onde é registrado se o aluno cursou o ensino médio em escola pública ou privada; `Cota`, que é o campo que armazena se o aluno entrou por cotas ou não; e, finalmente, `Idade`, com a idade do aluno. Todos os atributos selecionados podem ser vistos na Figura 33.

Figura 33 - Atributos selecionados para a tarefa de agrupamento utilizando os dados da UNESP.

`Classificacao`	`Presenca`	`Escola`
`CR`	`Sexo`	`Cota`
`DiscAproveitada`	`Cor`	`Idade`

Fonte: Autoria própria.

5.2.1.2 Etapa de pré-processamento dos dados

A etapa de pré-processamento é a segunda etapa da metodologia proposta. Nesta etapa realiza-se uma limpeza no conjunto de dados em que será aplicada a *data mining*. Esta limpeza se propõe a eliminar dados duplicados e sem relevância, além de corrigir dados incompletos ou faltantes.

Neste estudo, como os dados analisados foram gerados a partir de relatórios, a quantidade de problemas nesta etapa foi menor quando comparada à mesma etapa no estudo de caso do IFMS. No entanto, alguns detalhes puderam ser melhorados, como a exclusão de registros que continham algum dado faltante e a eliminação de dados duplicados.

Os registros de alunos que tinham cursado apenas o primeiro ano foram eliminados, já que o campo conceito foi atribuido a partir da porcentagem de aproveitamento das disciplinas de

cada aluno. A eliminação desses registros se deu com o intuito de tornar o estudo mais realista, considerando que os alunos podem reprovar mais no primeiro semestre, por não estarem acostumados com o sistema de ensino-aprendizagem apresentado no curso. Ao final desta etapa foram obtidos 378 registros para serem analisados.

5.2.1.3 Etapa de transformação dos dados

A etapa seguinte da metodologia, a etapa de transformação, tem o objetivo de minimizar a complexidade dos dados que serão analisados. Alguns dados selecionados neste estudos foram transformados. O campo `CR`, originalmente com quatro casas decimais após a vírgula, passou a ter apenas uma casa decimal depois da vírgula e, portanto, teve seu valor arrendondado. Os campos `DiscAproveitada` e `Presenca` no início eram valores decimais com duas casas após a vírgula e passaram a ser números inteiros, também arrendodados. O campo `Sexo`, que continha os valores *masculino* ou *feminino* recebeu os valores *M* ou *F*, respectivamente. O campo `Cota` inicialmente tinha descrito por qual tipo de cota o aluno ingressou na instituição, caso ele tivesse usado o sistema de cotas. Este campo foi alterado apenas para *sim* ou *não*, indicando se o aluno fez uso do sistema de cotas ou não. O campo `Idade` também foi transformado, pois originalmente no arquivo CSV tinha-se o campo data de nascimento.

5.2.1.4 Etapa de data mining

A etapa de *data mining* é considerada a principal etapa do processo de descoberta de conhecimento por ser a etapa onde a técnica de mineração é aplicada. Nesta parte do estudo de caso, onde o objetivo é realizar a tarefa de agrupamento, a técnica escolhida foi a de particionamento. Assim como na etapa de *data mining* realizada no estudo de caso do IFMS, apresentado no item 5.1.1.4, o objetivo desse agrupamento era descobrir características em comum entre os alunos que têm alto rendimento acadêmico, identificar caracteríscas compartilhadas entre os alunos que têm baixo rendimento acadêmico e encontrar características capazes de distinguir se o aluno terá ou não um bom desempenho acadêmico.

O algoritmo de *data mining* escolhido foi o *SimpleKMeans*, que, como já discutido neste trabalho, é uma implementação em Java do algoritmo *k-means* disponível na ferramenta Weka. O *SimpleKMeans* foi adotado neste estudo por sua simplicidade e por ser uma das opções mais utilizadas dentre os algoritmos de agrupamento (JAIN, 2010; WU et al., 2008).

Os principais parâmetros disponíveis para execução do algoritmo *SimpleKMeans*, juntamente com sua descrição e o valor utilizado, podem ser vistos na Tabela 3, página 63. Dentre

os parâmetros do algoritmo, o único que sofreu alteração foi o `numClusters`, definido como 3. Este parâmetro define o número de *clusters* que serão gerados em cada teste e, depois de tentativas de execução com diferentes números de *clusters*, o valor 3 foi o que gerou os agrupamentos que oferecem interpretações mais interessantes. Com os demais parâmetros, foram realizados experimentos com variações diferentes em seus valores, mas que não resultaram em mudanças significativas e, portanto, os demais parâmetros foram mantidos com seus valores padrão, apresentados na tabela.

A tela da ferramenta Weka onde é executada a *data mining* foi apresentada anteriormente na Figura 23, página 64. É nesta mesma tela onde são realizados os testes de agrupamento com os dados da UNESP. Todos os modos de agrupamento disponíveis para testes foram realizados nesta etapa: `Use training set`, `Supplied test set`, `Percentage split` e `Classes to clusters evaluation`.

5.2.1.5 Etapa de interpretação e avaliação dos resultados

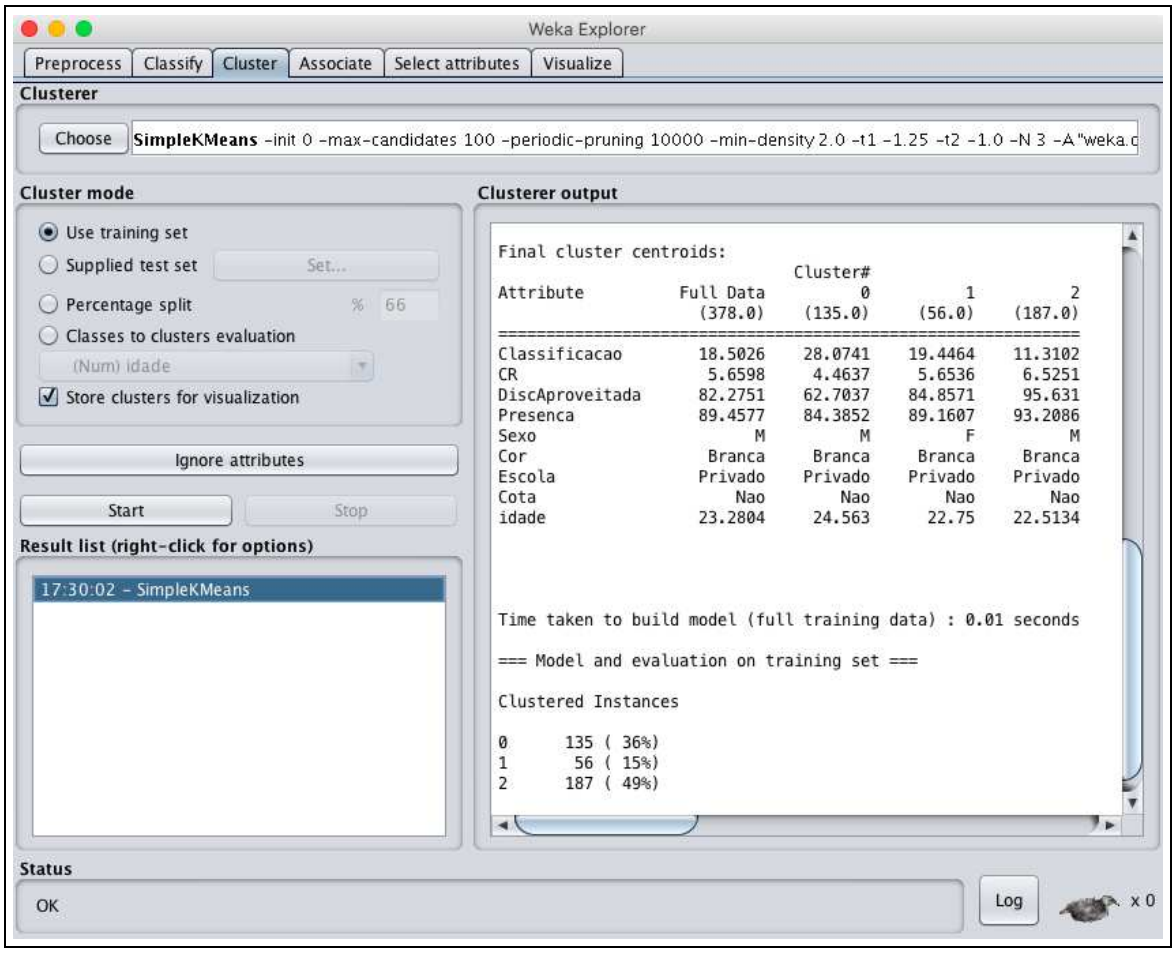
É nesta etapa que os resultados gerados pela etapa de *data mining* são interpretados e avaliados. O objetivo desta etapa é analisar os resultados, verificar se é possível entender a resposta do algoritmo executado e, posteriormente, avaliar se as respostas geradas são satisfatórias.

Como já mencionado, existem quatro modos de agrupamento diferentes disponíveis na ferramenta Weka, a seguir serão detalhados os resultados obtidos nas execuções de cada modo.

O primeiro modo testado foi o `Use training set`. Este modo de execução utiliza o conjunto de dados carregado para realizar tanto o treinamento quanto o teste do algoritmo. O resultado gerado a partir deste modo de execução pode ser visto na Figura 34. Observa-se, na figura, que foram gerados três *clusters* e que os atributos `Cor`, `Escola` e `Cota` não foram decisivos na hora da formação dos *clusters*, pois aparecem com o mesmo valor nos três agrupamentos. O *cluster* 0 reúne 36% dos alunos e poderia ser nomeado de “Baixo Desempenho”, pois a maioria dos alunos que foram alocados neste grupo foram os piores classificados no vestibular, seus coeficientes de rendimento foram os mais baixos, a porcentagem de aproveitamento de disciplinas também foi menor e a frequência desses alunos nas aulas também é baixa. O *cluster* 1 é o menor, com apenas com 15% dos alunos, e é o único grupo composto em sua maioria por mulheres. A esse *cluster* pode-se dar o nome de “Médio Desempenho”, pois os atributos `Classificacao`, `CR`, `DiscAproveitada` e `Presenca` tiveram seus valores médios comparados com os demais agrupamentos. O terceiro *cluster* é o maior, com 49% dos alunos, e pode ser denominado “Alto Desempenho”, pois a maioria dos alunos ficaram bem classificados no vestibular, têm os melhores coeficientes de rendimento, têm aproveitamento de disciplina maior que

95% e ainda são os mais assíduos às aulas.

Figura 34 - Interface do resultado do *cluster* utilizando a opção `Use training set` no estudo de caso da UNESP.



Fonte: Autoria própria.

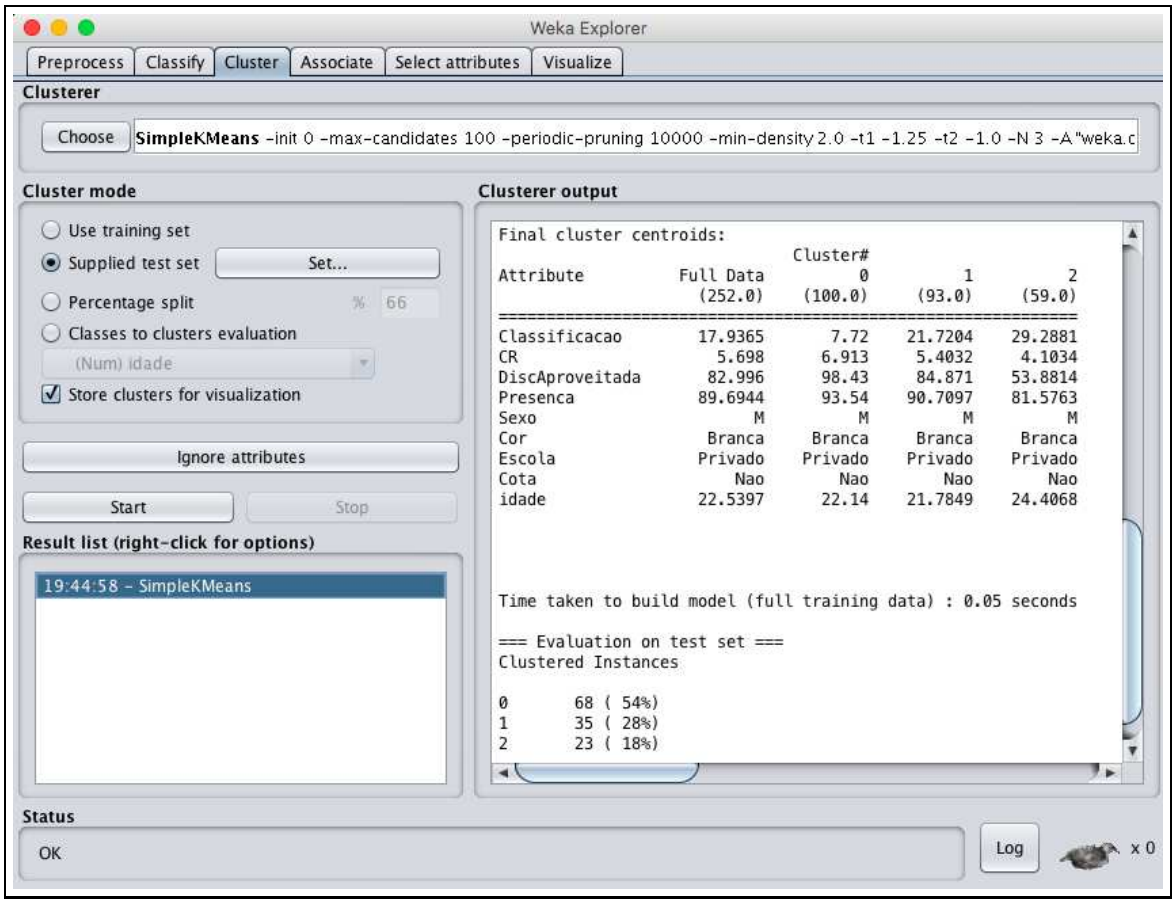
O ponto que pode ser considerado interessante nos resultados do modo `Use training set` é que, apesar de o curso de Engenharia Elétrica contar com muito mais alunos do que alunas, as alunas conseguem ter um bom rendimento no curso. No entanto, como já mencionado, o modo de agrupamento `Use training set` não é considerada uma execução realista.

O segundo modo de agrupamento utilizado para teste foi `Supplied test set`. Esta opção precisa de dois conjuntos de dados para sua execução. O primeiro conjunto é utilizado para treinamento do algoritmo e o segundo conjunto é utilizado para o teste. Sendo assim, o conjunto de dados principal de 378 registros foi dividido de forma aleatória, onde o primeiro conjunto ficou com 252 registros e o segundo conjunto ficou com 126 registros para teste.

O resultado obtido por meio da opção `Supplied test set` é apresentado na Figura 35. Pode-se obervar que foram formados três *clusters* com o primeiro conjundo de dados, já que

a coluna `Full Data` mostra o total de 252 registros. Ainda na Figura 35, abaixo na tabela `Clustered Instances` é exibido o resultado do teste com o segundo conjunto de dados (126 registros). No geral, neste resultado pode-se dizer que os grupos novamente foram divididos em Baixo, Médio e Alto Desempenho. O primeiro *cluster*, com 54% dos registros é o denominado "Alto Desempenho", onde estão concentrado os melhores alunos. O segundo *cluster*, com 35% dos registros, pode ser chamado de "Médio Desempenho", pois de acordo com o agrupamento a maioria dos alunos com desempenho acadêmico medianos estão neste grupo. O último *cluster*, denominado "Baixo Desempenho" conta com 23% dos registros. A maioria dos alunos que compõem esse *cluster* ficaram mal classificados no vestibular, têm um aproveitamento de disciplinas pouco maior que 50%, têm baixo coeficiente de rendimento e faltam mais às aulas. Novamente os campos `Cor`, `Escola` e `Cota` não foram decisivos na hora de montar os *clusters*. O campo `Idade` indica que os alunos com "Baixo Desempenho" são os mais velhos.

Figura 35 - Interface do resultado do *cluster* utilizando a opção `Supplied test set` no estudo de caso da UNESP.



Fonte: Autoria própria.

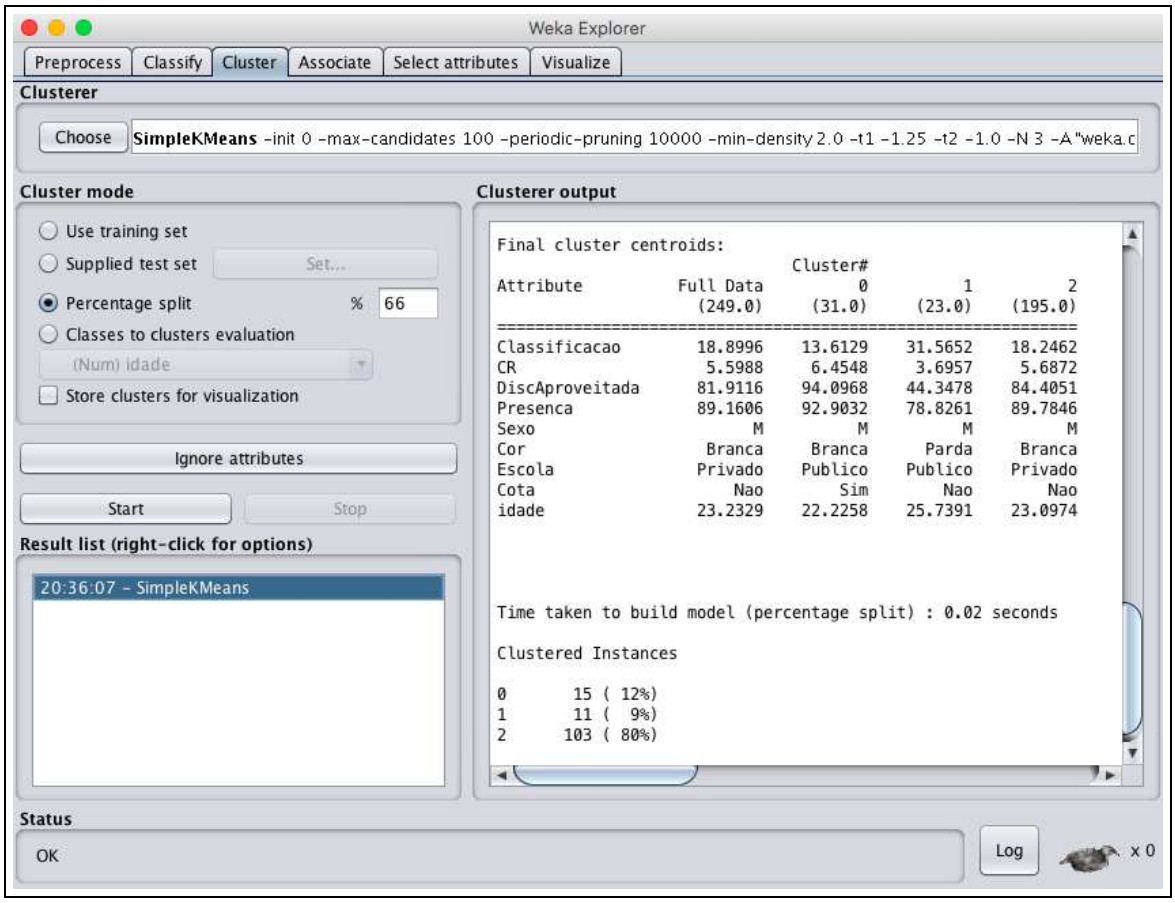
O resultado do `Supplied test set` revelou que a UNESP tem aluno de baixo, médio

e alto desempenho. Essa informação já era conhecida, logo o processo de *data mining* não revelou um novo conhecimento. Embora tenham sido realizados vários testes com o modo de agrupamento `Supplied test set`, os resultados obtidos não foram satisfatórios.

O `Percentage Split` foi o terceiro modo de agrupamento testado. Este modo faz o treinamento e o teste com partes diferentes do conjunto de dados, mas não é necessário utilizar separadamente dois conjuntos de dados. É carregado o conjunto de dados principal e a ferramenta Weka se encarrega de dividir os dados entre treinamento e teste. A quantidade de dados separada para o treinamento é de acordo com um parâmetro `%`, ao lado do modo de agrupamento `Percentage Split`, como pode ser visto na Figura 36. O valor padrão desse parâmetro é 66%, o que significa que 66% dos dados serão utilizados para treinamento e os outros 34% serão utilizados no teste. Nesse estudo foi mantido o valor padrão, pois foi o que gerou resultados mais interessantes.

Como nos demais testes de agrupamento, foram organizados três *clusters*. O *cluster* 0 pode receber o nome de "Alto Desempenho", pois tem a média de classificação no vestibular bem melhor que os outros agrupamentos, o coeficiente de rendimento dos alunos alocados neste *cluster* é alto e a porcentagem de aproveitamento das disciplinas é bem expressiva. Os alunos deste *cluster* faltam pouco às aulas, mas, o mais interessante é que os alunos integrantes desse *cluster* estudaram em escolas públicas e são cotistas. O *cluster* 1 é o menor entre os três, ele pode ser nomeado de "Baixo Desempenho". Os piores números estão neste *cluster*, a média do campo `Classificacao` foi 31, aproximadamente. O coeficiente de rendimento, assim como a porcentagem de aproveitamento nas disciplinas cursadas, são consideravelmente baixas. Até a frequência desse alunos nas salas de aula é baixa. Este *cluster* tem um valor diferente até então, a cor parda, segundo o agrupamento, é uma característica que une alunos de baixo desempenho. O terceiro *cluster*, muito maior que os demais, é composto pelo alunos que têm desempenho médio. Logo, o nome para este *cluster* é "Médio Desempenho", dado que os atributos `Classificacao`, `CR` e `DiscAproveitada` têm valores medianos. Os atributos `sexo` e `idade` não se destacaram neste teste.

Figura 36 - Interface do resultado do *cluster* utilizando a opção `Percentage Split` no estudo de caso da UNESP.



Fonte: Autoria própria.

O resultado alcançado por meio do `Percentage Split` pode ser considerado interessante, pois revelou que alunos oriundos de escolas públicas e que são cotistas parecem ter um maior comprometimento com o curso, pois seus resultados estão entre os melhores. Esse modo de agrupamento revelou também que alunos de escolas públicas, que não são cotistas, que têm em média 25 anos, que são pardos e foram mal classificados no vestibular, tendem a ter um baixo desempenho acadêmico.

O quarto modo de agrupamento é denominado `Classes to clusters evaluation`. Este modo de execução permite que seja escolhido um atributo nominal para verificar se os registros foram alocados no `cluster` certo. O atributo escolhido é ignorado no momento da alocação das instâncias e, posteriormente, o algoritmo usa o valor do atributo separado para conferir se a instância foi alocada corretamente. No entanto, o conjunto de dados aqui disponível para teste não possui um atributo nominal adequado para este processo, pois os campos `Sexo`, `Cor` `Escola` e `Cota` não são decisivos a ponto de serem usados para verificar se uma instância foi ou

não alocada corretamente. Isto é, não foi possível realizar o teste com o modo de agrupamento `Classes to clusters evaluation`.

Depois dos testes realizados, pode-se dizer que o modo de agrupamento que gerou melhor resultado foi o `Percentage Split`, pois seu resultado mostrou que a maioria dos alunos cotistas oriundos de escolas públicas têm muito bom desempenho no curso de Engenharia Elétrica e esse dado pode ser considerado interessante, já que é comum pensar que alunos vindos de escolas particulares têm melhor rendimento que alunos de escolas públicas. O resultado, a partir do modo de agrupamento `Supplied test set`, não atingiu o objetivo da *data mining*, pois a informação gerada, é compreensiva, mas não é nova. Ainda assim, mesmo este modo de agrupamento tendo sido testado várias vezes, o melhor resultado encontrado não gerou um novo conhecimento. O modo `Use training set`, não oferece os resultados mais realistas (WITTEN; FRANK; HALL, 2011), mas indicou que as mulheres, mesmo sendo minoria, também têm bons resultados no curso de Engenharia Elétrica analisado.

5.2.2 Aplicação da tarefa de classificação

Esta parte do presente estudo compreende a execução das etapas da metodologia para descoberta de conhecimento em banco de dados acadêmico, utilizando a tarefa de classificação, com o objetivo de descobrir novas informações que possam ser úteis para a gestão do curso de Engenharia Elétrica. Mais especificamente, o objetivo deste estudo é classificar os alunos em conceitos A, B, C ou D, a partir de dados socioeconômicos dos alunos.

Neste estudo as três primeiras etapas do processo proposto pela metodologia não puderam ser executadas utilizando a aplicação AADM, uma vez que a aplicação acessa o banco de dados diretamente para realizar essas atividades. Logo, neste caso, todas as etapas foram realizadas de forma manual, seguindo a metodologia proposta, baseada sempre no conjunto de dados obtido dos relatórios do sistema acadêmico da instituição.

5.2.2.1 Etapa de seleção dos dados

Nesta etapa, a seleção dos dados foi realizada a partir de arquivos CSV exportados do sistema acadêmico da instituição. Os dados selecionados, em sua maioria, podem ser considerados da ordem social ou econômica. Os campos selecionados foram a classificação do aluno no vestibular, denominado `Classificacao`, a média da porcentagem de presença do aluno nas disciplinas, denominado `Presenca`, o `Sexo`, a `Cor`, a `Idade` do aluno, o campo que armazena se o aluno entrou por cota ou não, denominado `Cota` e o campo que mantém registrado se o

aluno cursou o ensino médio em escola pública ou privada, denominado `Escola`. Um campo adicional para o aluno, denominado `Conceito`, não existia no arquivo original e foi criado especificamente para este estudo. Todos os atributos selecionados podem ser visto na Figura 37.

Figura 37 - Atributos selecionados para a tarefa de classificação utilizando os dados da UNESP.

```
Classificacao       Cor                 Idade
Presenca            Escola              Conceito
Sexo                Cota
```

Fonte: Autoria própria.

O `Conceito` foi populado com base na porcentagem de aproveitamento das disciplinas de cada aluno. Se o aluno teve um aproveitamento de 90% ou mais nas disciplinas, então ele teve o conceito igual a A. Se o aluno teve um aproveitamento de disciplinas entre 70 e 89% ele teve para o campo conceito o valor B. Quando a aluno teve aproveitamento de disciplinas de 50 a 69% o valor para o campo conceito foi C e, para o aluno que teve o aproveitamento menor que 49%, o conceito foi definido como D. Todos esses dados podem ser vistos de maneira simplificada na Tabela 10. Na Tabela 11 são apresentadas as quantidades de alunos em cada conceito.

Tabela 10 - Atribuição de conceitos para porcentagens de aproveitamento de disciplinas.

% Aproveitamento de disciplinas	**Conceito**
≥ 90	A
< 90 e ≥ 70	B
< 70 e ≥ 50	C
< 50	D

Fonte: Autoria própria.

Tabela 11 - Quantidade de alunos em cada um dos conceitos A, B, C e D.

Conceito	**Quantidade de alunos**
A	187
B	103
C	63
D	25

Fonte: Autoria própria.

5.2.2.2 Etapa de pré-processamento dos dados

Para a realização do pré-processamento, foi verificado que, mesmo que o acesso aos dados tenha sido por meio da união de diferentes relatórios, em vez de consultas ao banco de dados, muitos campos encontravam-se vazios e muitos alunos que constavam em um relatório não constavam em outros. Então, durante o pré-processamento, foram eliminados muitos registros, mantendo-se apenas os registros que estavam com todos os campos preenchidos e que eram consistentes dentre todos os relatórios gerados.

Assim como a etapa de pré-processamento realizada na aplicação da tarefa de agrupamento, nesta parte do estudo de caso, também foram eliminados os alunos que cursaram apenas o primeiro ano do curso de Engenharia Elétrica. Após a exclusão de todos os registros nesta etapa, restaram 378 registros considerados aptos para as próximas etapas da metodologia.

5.2.2.3 Etapa de transformação dos dados

Na etapa seguinte da metodologia, quatro campos tiveram seus valores transformados. O campo `Presenca` deixou de ser um campo decimal e passou a ser um campo inteiro, tendo seus valores arrendodados. O campo `Sexo`, que continha os valores *masculino* ou *feminino* recebeu os valores *M* ou *F*, respectivamente. O campo `Cota` tinha o valor *não* para o aluno não cotista e, para o aluno cotista, tinha especificado se era cota racial ou social. Neste caso a alteração foi feita apenas para alunos cotistas, transformando seus valores para *sim*. Logo, os valores para o campo `Cota` foram definidos como *Sim* ou *Não*. O campo `Idade` também foi transformado, pois originalmente os dados apresentavam apenas a data de nascimento.

5.2.2.4 Etapa de data mining

Na etapa de *data mining* o algoritmo escolhido, de acordo com a tarefa e técnica definidas, é executado. A execução da *data mining* tem o objetivo de descobrir novos conhecimentos, bem como produzir padrões para um determinado conjunto de dados (FAYYAD; PIATETSKY--SHAPIRO; SMYTH, 1996).

A execução desta etapa segue os moldes da execução aplicada no estudo de caso que envolveu dados do IFMS, apresentado neste mesmo Capítulo, no item 5.1.2.2, página 72. Na *data mining* realizada nesta parte do estudo de caso também foi utilizada a tarefa de classificação, mas com o objetivo de classificar os alunos com conceitos A, B, C ou D. O algoritmo utilizado também foi o algoritmo o J4.8, pois é um dos algoritmos recomendados para este tipo de classificação (WITTEN; FRANK; HALL, 2011).

Assim como na execução da etapa de *data mining* apresentada o item 5.1.2.2, nesta parte do estudo de caso também foram produzidos testes com os modos de execução `Use training set`, `Supplied test set`, `Cross-validation` e `Percentage split`. Todos os modos de execução também foram testados com a poda e sem a poda da árvore de decisão. A diferença entre a execução da tarefa de classificação realizada sobre os dados do IFMS e os dados da UNESP é que, no primeiro o atributo alvo era `Programação` e, no segundo o atributo alvo é o `Conceito`. O motivo é que se deseja classificar se o aluno tem conceito A, B, C ou D.

5.2.2.5 Etapa de interpretação e avaliação dos resultados

Nesta última etapa da metodologia, são apresentados as interpretações e avaliações dos resultados obtidos em cada um dos testes realizados: quatro modos de execução, `Use training set`, `Supplied test set`, `Cross-validation` e `Percentage split`; e em cada modo foi executado o algoritmo de árvore de decisão com pode e sem poda da árvore.

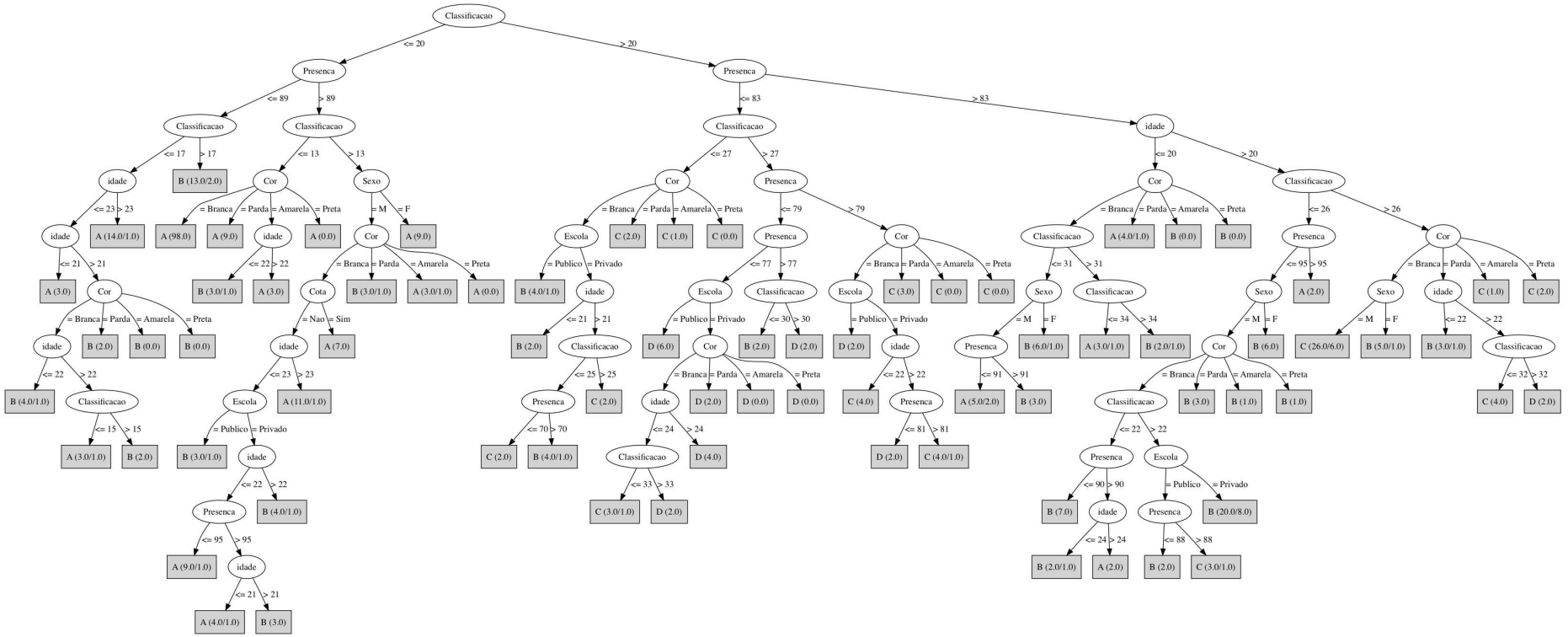
Em todos os testes realizados o atributo alvo foi sempre o campo `Conceito` e os demais campos utilizados também permaneceram os mesmos, sendo eles `Classificacao`, `Presenca`, `Sexo`, `Cor`, `Idade`, `Cota` e `Escola`.

Na execução do modo `Use training set`, a porcentagem de instâncias classificadas corretamente foi consideravelmente alta, sendo 85% de acertos no teste com a árvore podada e 89,2% de acertos no teste realizado sem a poda da árvore. A árvore que gerou o melhor resultado pode ser vista na Figura 38, esta árvore pode ser considerada grande, um dos fatos que contribuiram para seu tamanho é que todos os sete atibutos foram utilizados na construção da árvore e, além disso, o atributo `Cor` tem quatro valores possíveis (branca, amarela, parda e preta), o que aumenta o número de ramificações.

Na matriz de confusão da árvore sem poda, apresentada na Tabela 12 (1), é possível observar que a grande maioria dos registros foram classificados corretamente, como mostra a diagonal principal da matriz. Pode-se observar, na Tabela 12 (2), que a quantidade de registros classificados corretamente a partir da árvore com poda também é considerável, porém, ainda é inferior à quantidade de acertos gerado pela árvore anterior.

Os dois resultados poderiam ser considerados bons, no entanto, a opção `Use training set` não é considerada uma boa opção para testes, pois o algoritmo usa os mesmo dados para treinamento e depois para teste o que torna o resultado bastante otimista, sem considerar a existência de dados novos e desconhecidos (WITTEN; FRANK; HALL, 2011).

Figura 38 - Representação gráfica da árvore de classificação (sem poda) utilizando a opção `Use Training Set`.



Fonte: Autoria própria.

Tabela 12 - Resultados das execuções `Use Training Set` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Use Training Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	337	89,2%
Instâncias Classificadas Incorretamente	41	10,8%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
179	8	0	0	a = A (conceito A)
8	88	7	0	b = B (conceito B)
1	14	48	0	c = C (conceito C)
1	0	2	22	d = D (conceito D)

(1)

Use Training Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	321	85,0%
Instâncias Classificadas Incorretamente	57	15,0%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
176	11	0	0	a = A (conceito A)
17	79	7	0	b = B (conceito B)
1	16	43	3	c = C (conceito C)
1	0	1	23	d = D (conceito D)

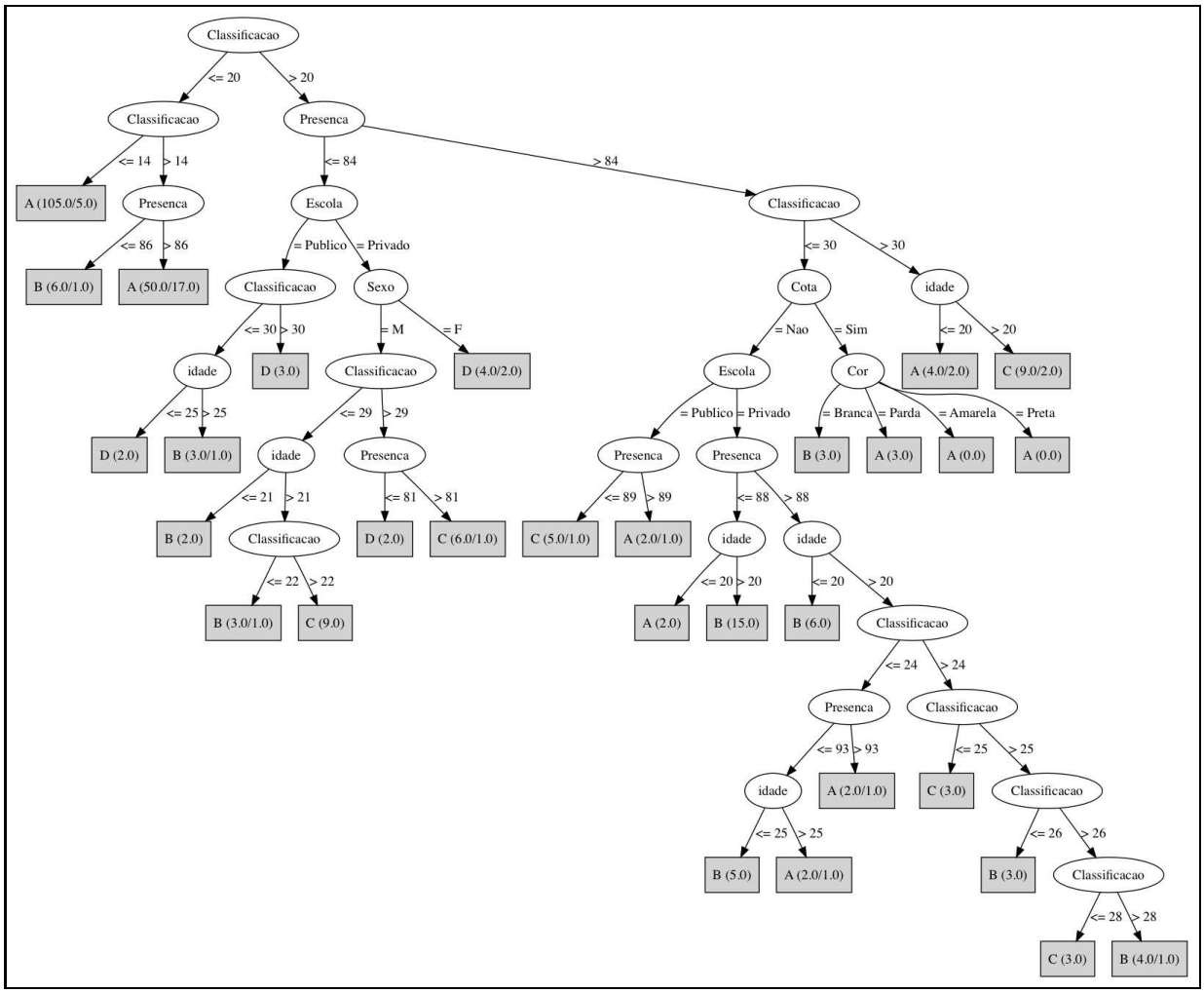
(2)

Fonte: Autoria própria.

O segundo modo de execução testado foi o `Supplied test set`. Para esse modo foi necessário dividir o conjunto de dados em duas partes, pois essa opção exige que seja carregado um conjunto de dados para treinamento e em seguida que seja carregado um segundo conjunto de dados para o teste. Então, o conjunto de dados principal foi separado, sendo que no primeiro conjunto utilizado para treinamento manteve-se 2/3 dos dados e, no segundo conjunto de dados, ficaram 1/3 dos dados restantes.

No modo `Supplied test set` o melhor resultado foi obtido com a opção da árvore com poda, onde a porcentagem de instâncias classificadas corretamente foi de 85,8%, enquanto que com a árvore sem poda o número de instâncias classificadas corretamente foi de 83,5%. A árvore com poda considerou todos os atributos em sua formação, como pode ser visto na Figura 39.

Figura 39 - Representação gráfica da árvore de classificação (com poda) utilizando a opção `Supplied test set`.



Fonte: Autoria própria.

Os detalhes da matriz de confusão da árvore com poda e sem poda, bem como as porcentagens de instâncias classificadas corretas e incorretamentes podem ser vistos na Tabela 13 (1) e na Tabela 13 (2).

Tabela 13 - Resultados das execuções `Supplied test set` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Supplied Test Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	106	83,5%
Instâncias Classificadas Incorretamente	21	16,5%

Matriz de confusão

a	**b**	**c**	**d**	<-- Classificado como
53	0	0	0	a = A (conceito A)
3	15	1	0	b = B (conceito B)
3	5	28	3	c = C (conceito C)
0	3	3	10	d = D (conceito D)

(1)

Supplied Test Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	109	85,8%
Instâncias Classificadas Incorretamente	18	14,2%

Matriz de confusão

a	**b**	**c**	**d**	<-- Classificado como
53	0	0	0	a = A (conceito A)
3	15	1	0	b = B (conceito B)
3	4	28	4	c = C (conceito C)
1	0	2	13	d = D (conceito D)

(2)

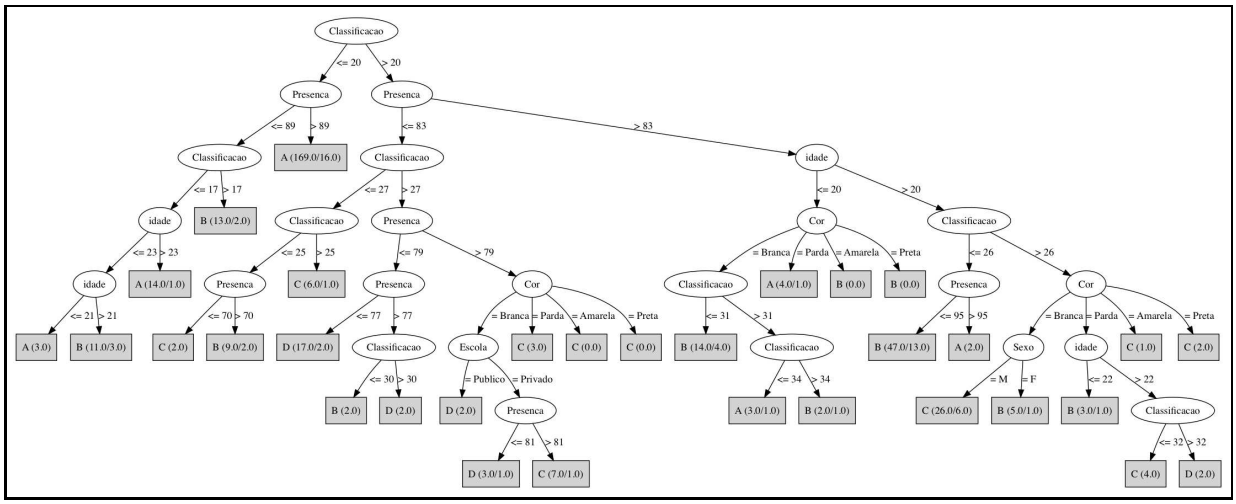
Fonte: Autoria própria.

A terceira opção testada foi a `Cross-validation`. Esta opção requer um único conjunto de dados para sua execução, porém este conjunto de dados é dividido em uma determinada quantidade de partes para treinamento e teste. Neste estudo seguiu-se a sugestão da ferramenta Weka em manter o parâmetro `Fold` com seu valor padrão 10. Ou seja, optou-se por dividir o conjunto de dados em dez partes, já que foi com esse valor que a opção `Cross-validation` gerou o melhor resultado. Vale lembrar que todas as dez partes são utilizadas para treinamento e teste, em momentos diferentes.

O melhor resultado alcançado por meio da opção `Cross-validation` também se deu quando o valor do parâmetro `unprune` era falso, ou seja, quando a árvore foi gerada com poda. A árvore podada pode ser vista na Figura 40, onde pode-se notar que o campo `Cota` não foi considerado para a composição da árvore. Também pode ser notado na mesma figura que a árvore gerada no modo de execução `Cross-validation` é menor que as árvores geradas com os modos de execução `Use training set` e `Supplied test set`.

Na Tabela 14 pode ser visto que a árvore com poda teve 270 registros classificados corretamente, ou seja, uma taxa de 71,4% de acerto, enquanto que com a árvore sem poda a taxa de acerto caiu para 67,7%. Nesta tabela também são apresentadas as matrizes de confusão das árvores sem e com poda.

Figura 40 - Representação gráfica da árvore de classificação (com poda) utilizando a opção `Cross-validation`.



Fonte: Autoria própria.

Tabela 14 - Resultados das execuções `Cross-validation` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Cross-validation: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	256	67,7%
Instâncias Classificadas Incorretamente	122	32,3%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
161	24	2	0	a = A (conceito A)
28	51	22	2	b = B (conceito B)
1	25	31	6	c = C (conceito C)
1	1	10	13	d = D (conceito D)

(1)

Cross-validation: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	270	71,4%
Instâncias Classificadas Incorretamente	108	28,6%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
168	17	2	0	a = A (conceito A)
28	53	21	1	b = B (conceito B)
1	21	37	4	c = C (conceito C)
1	0	12	12	d = D (conceito D)

(2)

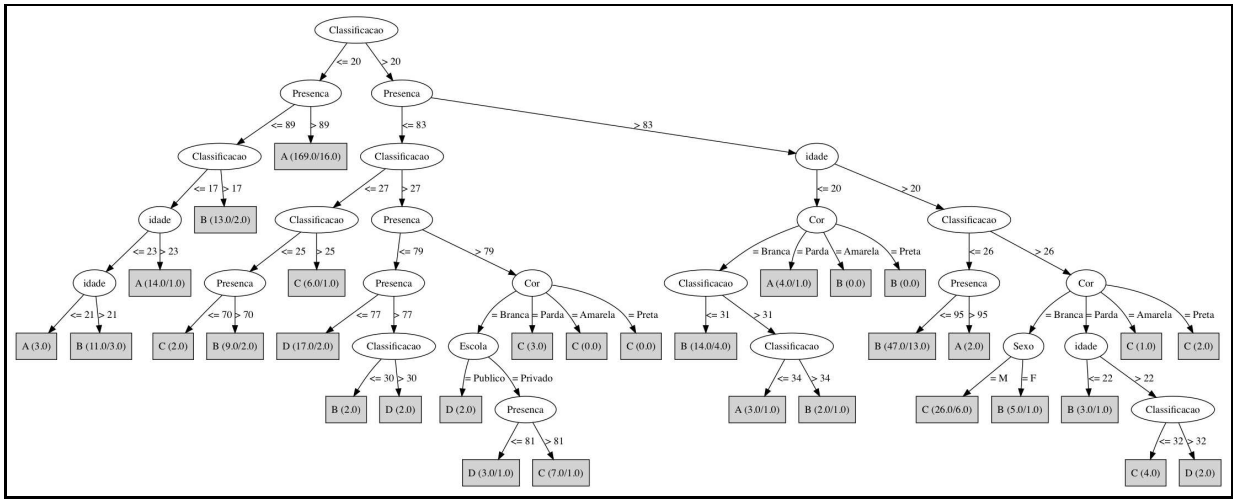
Fonte: Autoria própria.

O último modo de execução da classificação realizado foi o `Percentage split`. Este modo também utiliza o mesmo conjunto de dados para treinamento e teste, mas sua forma de fucionamento é diferente do modo `Cross-validation`. Neste caso o conjunto de dados é separado em dois, sendo uma parte para treinamento e a outra parte para teste. Seguiu-se o padrão e dividiu-se o conjunto de dados em 2/3 para treinamento e 1/3 para teste e, no modo `Percentage split`, esse processo acontece apenas uma vez, ao contrário do que ocorre no modo `Cross-validation`.

No modo `Percentage split` o melhor resultado também foi obtido com a opção da árvore

com poda, esta árvore pode ser vista na Figura 41. Pode-se observar que esta árvore é idêntica a árvore gerada na opção `Cross-validation`, apresentada na Figura 40 e nota-se que o campo `Cota` também não foi considerado no desenvolvimento da árvore.

Figura 41 - Representação gráfica da árvore de classificação (com poda) utilizando a opção `Percentage split`.



Fonte: Autoria própria.

Na Tabela 15 são apresentados os resultados obtidos por meio do modo `Percentage split`. Observa-se que, com a árvore podada, a taxa de instâncias classificadas corretamente foi de 78,3%, enquanto que a taxa de acerto com a árvore sem poda foi de 75,2%. Mais uma vez, a árvore podada gerou o melhor resultado pelas suas características de generalização na classificação de instâncias desconhecidas. Ainda na Tabela 15 podem ser vistos os detalhes das matrizes de confusão das árvores geradas sem e com poda, bem como as quantidades de instâncias classificadas correta e incorretamente.

Tabela 15 - Resultados das execuções `Percentage split` do algoritmo J4.8 sem poda (1) e com poda (2). As matrizes de confusão mostram a distribuição de instâncias classificadas correta e incorretamente.

Percentage Split: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamente	97	75,2%
Instâncias Classificadas Incorretamente	32	24,8%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
63	4	0	0	a = A (conceito A)
10	10	8	1	b = B (conceito B)
1	4	20	1	c = C (conceito C)
0	0	3	4	d = D (conceito D)

(1)

Percentage Split: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamente	101	78,3%
Instâncias Classificadas Incorretamente	28	21,7%

Matriz de confusão				
a	**b**	**c**	**d**	<-- Classificado como
63	4	0	0	a = A (conceito A)
7	13	9	0	b = B (conceito B)
0	4	21	1	c = C (conceito C)
0	0	3	4	d = D (conceito D)

(2)

Fonte: Autoria própria.

Depois de realizados todos os teste disponíveis na tarefa de classificação utilizando dados da UNEP, conclui-se que os melhores resultados foram obtidos através do modo de agrupamento `Supplied test set` e `Percentage split`, como pode ser visto na Tabela 16.

Tabela 16 - Resumo dos desempenhos das opções de teste de classificação com dados da UNESP.

Opção de teste	**% de classificação correta**	**% de classificação incorreta**	**Árvore**
Use training set	89,2	10,8	Sem poda
Supplied test set	85,8	14,2	Com poda
Cross-validation	71,4	28,6	Com poda
Percentage split	78,3	21,7	Com poda

Fonte: Autoria própria.

6 CONCLUSÕES

Este trabalho apresentou uma proposta de metodologia, baseada no KDD, com o objetivo de simplificar o processo de descoberta de novos conhecimentos em bancos de dados acadêmicos para as tarefas de classificação e agrupamento. A metodologia proposta se difere do KDD no sentido em que, uma vez que o resultado da *data mining* é avaliado e, por qualquer motivo não é satisfatório, retornar a qualquer etapa pode gerar maior demanda de tempo para a conclusão do processo. Um modo de diminuir o tempo gasto no processo é retornar à etapa de *data mining* ou, se necessário, retornar ao banco de dados e estudá-lo novamente.

Ademais, conclui-se que a metodologia proposta simplifica e agiliza consideravelmente o processo de decoberta de novos conhecimentos em bancos de dados acadêmicos quando a tarefa executada é a de classificação, uma vez que a AADM, em uma única etapa, acessa o banco de dados, seleciona os dados especificados, realiza o pré-processamento, eliminando todos os registros incompletos ou inconsistentes, e, por fim, transforma os dados conforme necessário. Além disso, para finalizar seu processo, a AADM gera um arquivo do tipo ARFF pronto para ser importado no sistema Weka. A agilidade oferecida na execução da metodologia é ainda maior, uma vez que são restritas as possibilidades de retorno a etapas anteriores, nos casos em que são obtidos resultados insatisfatórios.

Concluiu-se também, neste estudo, que a Weka é uma ferramenta importante para a realização da etapa de *data mining*. A Weka realiza a importação dos conjuntos de dados a serem trabalhados, implementa e executa os algoritmos de *data mining* utilizados tanto na tarefa de classificação quanto na tarefa de agrupamento, de forma simples e flexível, mostrando-se uma ferramenta adequada para a metodologia de descoberta de conhecimento proposta.

Por fim, com base nos estudos de casos realizados, conclui-se que o modo de agrupamento `Percentage split` é bastante interessante para estudos que envolvem dados acadêmicos. Pois, tanto na sua execução com dados do IFMS quanto com dados da UNESP foram descobertos novos conhecimentos. No caso do IFMS o resultado gerado a partir do modo `Percentage split` demonstrou que alunos de idade mais avançada, casados e trabalhadores são, na maioria das vezes, aprovados na disciplina de Algoritmos, enquanto que alunos com menos idade, solteiros e que não trabalham, em sua maioria, são reprovados. No caso da UNESP, o agrupamento realizado por meio do modo `Percentage split` revelou que alunos cotistas, brancos e prove-

nientes de escola pública têm alto desempenho no curso de Engenharia Elétrica. Os três fatos descobertos podem ser considerados interessantes, já que são diferentes do esperado.

Ainda com base nos estudos de casos realizados, mas agora com a tarefa de classificação, conclui-se que é possível estimar dados acadêmicos com taxas de acertos de 75% ou mais, como foi o caso das opções de teste `Percentage split` e `Supplied test set` aplicadas aos dados do IFMS e da UNESP, respectivamente. Acertos igual ou superior a 75% são consideráveis já que é uma predição e a quantidade de dados disponíveis para teste não era muito grande. No caso do IFMS, a classificação demonstrou que é possível predizer a aprovação dos alunos na disciplina de Linguagem de Programação 1, com base nos resultados das disciplinas de Páginas Web 1 e Algoritmos. No caso da UNESP, com base nas informações de classificação no vestibular, presença dos alunos nas aulas, além de idade e cor, foi possível classificar os alunos nos conceitos A, B, C ou D com uma taxa de acerto considerável.

6.1 TRABALHOS FUTUROS

Tem-se a possibilidade, como trabalho futuro, de incorporar mais funcinalidades à Aplicação para Apoio à *Data Mining* (AADM) a fim de transformar essa aplicação em uma ferramenta que execute todos os passos da metodologia proposta. Até o momento, a ferramenta executa as três primeiras etapas da metodologia, sendo elas a etapa de seleção de dados, a etapa de pré-processamento dos dados e a etapa de transformação dos dados. O objetivo é que sejam implementados, em trabalhos futuros, algoritmos de agrupamento e classificação em uma ferramenta mais direcionada à análise de bancos de dados acadêmicos. Além disso, tem-se a possibilidade de visualização dos resultados das execuções dos algoritmos na própria ferramenta desenvolvida, de maneira semelhante ao que se obtém na ferramenta Weka.

Espera-se, ainda, em trabalhos futuros, a utilização de dados acadêmicos mais abrangentes do banco de dados da UNESP para a execução das tarefas utilizadas neste trabalho. Espera-se que possam ser utilizados dados como notas obtidas em todas as disciplinas cursadas e que esses dados possam ser usados para a obtenção de informações e conhecimentos mais detalhados do Curso de Engenharia Elétrica da instituição.

REFERÊNCIAS

ARAUJO, F. H. D.; SANTANA, A. M.; SANTOS NETO, P. A. Evaluation of classifiers based on decision tree for learning medical claim process. *IEEE Latin America Transactions*, Piscataway, v. 13, n. 1, p. 299–306, Jan 2015.

BARADWAJ, B. K.; PAL, S. Mining educational data to analyze students' performance. *International Journal of Advanced Computer Science and Applications*, West Yorkshire, v. 2, n. 6, p. 63–69, 2011.

BERKHIN, P. A survey of clustering data mining techniques. In: ____. *Grouping multidimensional data*: recent advances in clustering. Berlin: Springer Berlin Heidelberg, 2006. p. 25–71.

BOENTE, A. N. P.; GOLDSCHMIDT, R. R.; ESTRELA, V. V. Uma metodologia de suporte ao processo de descoberta de conhecimento em bases de dados. In: SIMPÓSIO DE EXCELÊNCIA EM GESTÃO E TECNOLOGIA, 5., 2008, Rio de Janeiro. *Anais...* Rio de Janeiro: Associação Educacional Dom Bosco, 2008. p. 4–5.

BREIMAN, L.; FRIEDMAN, J.; STONE, C.; OLSHEN, R. *Classification and regression trees*. Wadsworth: Taylor & Francis, 1984. (The Wadsworth and Brooks-Cole Statistics-Probability Series).

CABENA, P.; HADJNIAN, P.; STADLER, R.; ZANASI, S. *Discovering data mining*: from concept to implementation. New York: Prentice Hall, 1997.

CEREDA, P. R. M.; JOSE, J. Adaptive data mining: preliminary studies. *IEEE Latin America Transactions*, Piscataway, v. 12, n. 7, p. 1258–1270, Oct 2014.

CHEN, M.-S.; HAN, J.; YU, P. S. Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, Los Alamitos, v. 8, n. 6, p. 866–883, 1996.

CIOS, K. J.; PEDRYCZ, W.; SWINIARSKI, R. W.; KURGAN, L. *Data mining*: a knowledge discovery approach. [S.l.]: Springer Science & Business Media, 2007.

DATE, C. J. *Introdução a sistemas de bancos de dados.* São Paulo: Campus, 1991.

DEEBA, K.; AMUTHA, B. Classification algorithms of data mining. *Indian Journal of Science and Technology*, Bangalore, v. 9, n. 39, 2016.

DINIZ, C. A.; LOUZADA-NETO, F. *Data mining*: uma introdução. São Paulo: ABE, 2000.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de banco de dados*. São Paulo: Pearson, 2011.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *AI Magazine*, Palo Alto, v. 17, n. 3, p. 37–53, 1996.

FAYYAD, U.; UTHURUSAMY, R. Evolving data mining into solutions for insights. *Communications of the ACM*, New York, v. 45, n. 8, p. 28–31, 2002.

FERREIRA, E. L.; RAUSCH, H.; CAMPOS, S.; FARIA-CAMPOS, A.; PIETRA, E.; SANTOS, L. da S. Medical data mining: a case study of a paracoccidioidomycosis patient's database. In: INTERNATIONAL CONFERENCE ON E-HEALTH NETWORKING, APPLICATIONS AND SERVICES (HEALTHCOM), 16., 2014, Natal. *Anais...* Natal: IEEE, 2014. p. 275–280.

GHEWARE, S.; KEJKAR, A.; TONDARE, S. Data mining: task, tools, techniques and applications. *International Journal of Advanced Research in Computer and Communication Engineering*, Thanjavur, v. 3, n. 10, p. 8095–8098, October 2014.

GIUDICI, P. *Applied data mining*: statistical methods for business and industry. Chichester: Wiley & Sons, 2005.

GOLDSCHMIDT, R. R.; PASSOS, E. P. L. *Data mining*: um guia prático - conceitos, técnicas, ferramentas, orientações e aplicações. Rio de Janeiro: Editora Campus, 2005.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The Weka data mining software: an update. *SIGKDD Explor. Newsl.*, New York, v. 11, n. 1, p. 10–18, nov 2009.

HAN, J.; HAIHONG, E.; LE, G.; DU, J. Survey on NoSQL database. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND APPLICATIONS- ICPCA, 6., 2011, Port Elizabeth. *Anais...* Port Elizabeth: IEEE, 2011. p. 363–366.

HAN, J.; KAMBER, M. *Data mining*: concepts and techniques. 3. ed. Boston: Elsevier Science, 2006.

HAND, D.; MANNILA, H.; SMYTH, P. *Principles of data mining*. Cambridge: MIT Press, 2001.

HEREDIA, D.; AMAYA, Y.; BARRIENTOS, E. Student dropout predictive model using data mining techniques. *IEEE Latin America Transactions*, Piscataway, v. 13, n. 9, p. 3127–3134, Sept 2015.

HOLMES, G.; DONKIN, A.; WITTEN, I. H. Weka: a machine learning workbench. In: INTELLIGENT INFORMATION SYSTEMS, 2., 1994, Brisbane. *Anais...* Brisbane: IEEE, 1994. p. 357–361.

JAIN, A. K. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, Amsterdam, v. 31, n. 8, p. 651 – 666, June 2010.

JIANG, M.-F.; TSENG, S.-S.; LIAO, S.-Y. Data types generalization for data mining algorithms. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS, MAN, AND CYBERNETICS- SMC, 12., 1999, Tokyo. *Anais...* Tokyo: IEEE, 1999. v. 5, p. 928–933.

KAUR, G.; CHHABRA, A. Improved J48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications*, New York, v. 98, n. 22, p. 13–17, July 2014.

KESAVARAJ, G.; SUKUMARAN, S. A study on classification techniques in data mining. In: INTERNATIONAL CONFERENCE ON COMPUTING, COMMUNICATIONS AND NETWORKING TECHNOLOGIES- ICCCNT, 4., 2013, Tiruchengode. *Anais...* Tiruchengode: IEEE, 2013. p. 1–7.

KTONA, A.; XHAJA, D.; NINKA, I. Extracting relationships between students' academic performance and their area of interest using data mining techniques. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE, COMMUNICATION SYSTEMS AND NETWORKS- CICSYN, 6., 2014, Tetovo. *Anais...* Tetovo: IEEE, 2014. p. 275–280.

MAIMON, O.; ROKACH, L. E. *Data mining and knowledge discovery handbook*. 2. ed. New York: Springer, 2010.

MARQUEZ-VERA, C.; CANO, A.; ROMERO, C.; NOAMAN, A. Y. M.; FARDOUN, H. M.; VENTURA, S. Early dropout prediction using data mining: a case study with high school students. *Expert Systems*, New Jersey, v. 33, n. 1, p. 107–124, 2016.

MORLEY, D.; PARKER, C. *Understanding computers*: today and tomorrow, comprehensive. [S.l.]: Cengage Learning, 2014.

NANDESHWAR, A.; MENZIES, T.; NELSON, A. Learning patterns of university student retention. *Expert Systems with Applications*, Kidlington, v. 38, n. 12, p. 14984 – 14996, 2011.

NAYAK, A.; PORIYA, A.; POOJARY, D. Type of NoSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems*, Foundation of Computer Science, New York, v. 5, n. 4, p. 16–19, March 2013.

ORGANIZAÇÃO DAS NAÇÕES UNIDAS- ONU. *A ONU e a população mundial.* 2014. [S.l.: s.n.]. Disponível em: <http://www.onu.org.br/a-onu-em-acao/a-onu-em-acao/a-onu-e-a-populacao-mundial/>. Acesso em: 24 set. 2014.

PAN, S.; MORRIS, T.; ADHIKARI, U. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, Piscataway, v. 6, n. 6, p. 3104–3113, Nov 2015.

QUINLAN, J. Induction of decision trees. *Machine Learning*, Boston, v. 1, n. 1, p. 81–106, 1986.

QUINLAN, J. R. *C4.5*: programs for machine learning. San Francisco: Morgan Kaufmann Publishers, 1993.

REN, Y.; ZHANG, L.; SUGANTHAN, P. N. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational Intelligence Magazine*, Piscataway, v. 11, n. 1, p. 41–53, Feb 2016.

SFERRA, H. H.; CORREA, A. M. J. Conceitos e aplicações de data mining. *Revista Ciência & Tecnologia*, Piracicaba, v. 11, n. 22, p. 19–34, 2003.

SILVA, L. A. E. A data mining approach for standardization of collectors names in herbarium database. *IEEE Latin America Transactions*, Piscataway, v. 14, n. 2, p. 805–810, Fev 2016.

TAN, P.; STEINBACH, M.; KUMAR, V. *Introdução ao data mining*. Rio de Janeiro: Ciência

Moderna, 2009.

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

WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; YU, P. S.; ZHOU, Z.-H.; STEINBACH, M.; HAND, D. J.; STEINBERG, D. Top 10 algorithms in data mining. *Knowledge and Information Systems*, London, v. 14, n. 1, p. 1–37, 2008.

APÊNDICE A - TRABALHO SUBMETIDO E ACEITO

GUERRA, M. S.; ASSEISS, H.; OLIVEIRA, S. A. A Case Study of Applying the Classification Task for Students' Performance Prediction. *IEEE Latin America Transactions*. ISSN: 1548-0992.

Estado do artigo

Paper ID: 4559

Title: Student Performance Prediction Using Data Mining and the Classification Task

Submission data: 2016-10-17

Last upadate: 2017-02-07

Authors: Maraísa Silva Guerra
Habib Asseiss
Sérgio Azevedo Oliveira

Paper in Review.

Reviewer ID:	Reviewer assign:	Recommendation:	Editor Comments:	Editor Decision:
1118	2016-10-19	Accept	07/02/2017 Modify. Os autores atenderam a todas as sugestões efetuadas e aumentaram muito a qualidade do artigo. Recomendo a publicação. Algumas sugestões de modificações para a versão final: 1. "Fig. X" ao invés de "Figura X" em todas chamadas das figuras no texto, conforme exige a formatação da IEEE; 2. "extrair informações e relacionamentos entre informações" ao invés de "extrair informações e relacionamentos de informação"; 3. Não colocaria artigo antes de "Data Mining". Simplesmente usaria "Data Mining"; 4. "Internet" ao invés de "internet"; 5. "em banco de dados" ao invés de "em dados armazenados em bancos de dados"	Awaiting

A Case Study of Applying the Classification Task for Students' Performance Prediction

M. S. Guerra, H. Asseiss and S. A. Oliveira

Abstract— **This paper presents a study involving the application of data mining techniques for extracting knowledge from the academic database of the Federal Institute of Mato Grosso do Sul (IFMS). The main goal is the prediction of students' performance on specific classes of the Internet Systems course. Extra students' information such as age and gender are also considered. Knowledge Discovery in Databases (KDD) is described and its steps are applied in this study. The classification task is used to generate decision trees that are tested on different datasets. The results show a success rate of 75.8% on the classification of new and unknown students based on the decision trees models generated.**

Keywords— **Data mining, classification task, academic database.**

I. INTRODUÇÃO

DATA mining é uma área interdisciplinar que desperta o interesse de pesquisadores em diversas áreas, que procuram aplicar algoritmos e técnicas específicas com o objetivo de extrair informações e relacionamentos entre informações em banco de dados. Trata-se do principal passo do processo denominado *knowledge discovery in database* (KDD). O KDD é um processo composto por cinco etapas: seleção de dados, pré-processamento de dados, transformação de dados, *data mining* e interpretação dos dados [1], [7].

Data mining, ou mineração de dados, vem ganhando muito espaço na área educacional, pois os bancos de dados escolares de instituições públicas ou privadas recebem todos os dias novos dados, que muitas vezes são subutilizados. Além disso, há muita informação ainda a ser descoberta sobre a educação no Brasil e técnicas de *data mining* podem ser bons aliados na busca por novos conhecimentos.

Em cursos da área da Computação a evasão de estudantes sempre foi muito alta. Segundo pesquisa recente publicada na 5ª edição do Índice Brasscom de Convergência Digital (IBCD), apresentada pela Associação Brasileira das Empresas de Tecnologia da Informação e Comunicação (Brasscom), a evasão de alunos nos cursos da área de Computação é bastante preocupante, pois apenas 18% dos ingressantes concluem o curso. Este fato é percebido fortemente no início do curso, onde os alunos apresentam maiores dificuldades na aprendizagem, o que gera reprovação e, consequentemente, a evasão [2].

Neste estudo foi utilizado o banco de dados do sistema acadêmico do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul (IFMS). No total, o banco de dados tem um tamanho aproximado de 76 MB, 294 tabelas e 9.692.875 registros, correspondendo a dados acadêmicos entre janeiro de 2012 e dezembro de 2015.

Entre as várias ferramentas disponíveis para execução de técnicas de *data mining*, destaca-se a Weka, uma ferramenta livre, de código aberto, desenvolvida em Java, amplamente testada em diversas plataformas e muito popular em ambientes acadêmicos [3], [4]. Dentre as tarefas de *data mining* que podem ser executadas na Weka está a tarefa de classificação, uma função de aprendizado que mapeia dados de entrada, ou um conjunto de dados de entrada, em um número finito de categorias pré-definidas, denominadas classes [3], [5]. Uma vez determinada, tal função pode ser aplicada a novos registros de forma a estimar a classe em que tais registros se enquadram.

Dentre técnicas bastante exploradas para executar a tarefa de classificação, destacam-se as árvores de decisão, consideradas uma das abordagens mais populares [17]. Uma característica importante das árvores de decisão é que, quando possuem um número razoável de folhas, as árvores podem ser convertidas em um conjunto de regras a fim de torná-las facilmente compreensíveis [20]. Diferentes algoritmos podem ser aplicados para a construção de árvores de decisão a partir de um conjunto de dados, entre eles encontra-se o C4.5 [6] e variantes, como o J4.8 [3].

Este artigo descreve a condução de testes e apresenta resultados de um estudo de caso onde se aplica a tarefa de classificação por árvores de decisão em dados reais. Os testes executados têm como objetivo classificar estudantes com base em seu desempenho escolar e são conduzidos de forma a verificar diferentes execuções do algoritmo de árvores de decisão. São gerados diferentes modelos de treinamento da classificação e, sequencialmente, são executadas as classificações com base nos modelos gerados. Por fim, são comparados os diferentes resultados e suas taxas de acerto são analisadas.

II. TRABALHOS RELACIONADOS

Recentes estudos demonstram a viabilidade da aplicação de técnicas de *data mining* em diversas áreas da ciência, como educação [5], medicina [7], [8], biologia [9] e engenharia [10]. Destaca-se uma forte tendência em estudos da área da educação, em que se aplicam técnicas de *data mining* com objetivos de oferecer predições envolvendo dados acadêmicos. Amaya et al. [11] demonstram a construção e a avaliação de um mo-

M. S. Guerra, Instituto Federal de Mato Grosso do Sul, Três Lagoas, MS, Brasil, maraisa.guerra@ifms.edu.br

H. Asseiss, Instituto Federal de Mato Grosso do Sul, Três Lagoas, MS, Brasil, habib.asseiss@ifms.edu.br

S. A. Oliveira, Universidade Estadual Paulista "Júlio de Mesquita Filho", Ilha Solteira, SP, Brasil, grilo@dee.feis.unesp.br

delo de predição da probabilidade de evasão escolar utilizando classificação através de árvores de decisão.

Nandeshwar et al. [12] utilizam as tarefas de árvores de decisão e redes neurais artificiais para predizer a retenção de estudantes nos três primeiros anos de um curso de graduação, detectando, como resultado, influências importantes para a retenção, como o estado socioeconômico familiar do aluno e sua média de notas no ensino médio.

O trabalho de Ktona et al. [13] consiste na aplicação da classificação para extrair regras de associação entre o desempenho acadêmico de estudantes e o programa de mestrado em que desejam ingressar no futuro. Os mesmos autores ainda utilizam agrupamento para particionar os estudantes de acordo com suas caraterísticas.

Márquez-Vera et al. [14] propõem a criação de uma metodologia de classificação específica para gerar modelos de predição da evasão escolar utilizando algoritmos genéticos. A metodologia proposta é capaz de gerar predições confiáveis da evasão de estudantes com até 6 semanas de curso.

Ramos et al. [22] realizam um estudo comparativo de métodos de agrupamento em dados provenientes de um sistema de educação a distância, obtendo a formação de diferentes grupos, cujos registros possuem características semelhantes. Os resultados mostram que é possível utilizar as informações específicas dos grupos para que sejam tomadas ações diferenciadas e, consequentemente, melhorar a qualidade dos cursos.

III. BANCO DE DADOS E KDD

Bancos de dados podem conter informações desconhecidas que podem ser úteis para entender alguns fenômenos ou tendências. Com o objetivo de revelar informações, utiliza-se o processo chamado de descoberta de conhecimento em banco de dados (KDD), que pode ser definido como um processo não trivial de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados [1].

O processo de KDD é composto por um conjunto de etapas com a finalidade de obter novos conhecimentos a respeito de um determinado domínio, a partir de uma base dados em estado bruto [8]. Pode-se dizer que o processo de KDD compreende todo o ciclo que o dado percorre até que este transforme-se em conhecimento [15], [16].

Para se iniciar o processo de KDD, é necessário conhecer o domínio em que os dados estão inseridos e levantar conhecimentos prévios, estabelecendo os objetivos da realização de *data mining*. A primeira etapa do processo é a seleção de dados, em que um conjunto de dados é preparado selecionando tabelas e atributos específicos que serão trabalhados. É possível que a seleção de dados tenha como fonte tabelas diferentes de múltiplos bancos de dados. A segunda tarefa é o pré-processamento, em que se executa operações como remoção de ruídos e dados sem relevância e a manipulação de campos com valores omissos [9]. O pré-processamento busca consolidar os dados, reduzindo sua complexidade. A terceira tarefa é a transformação dos dados, que pode envolver a seleção de atributos, discretização de atributos numéricos, projeção e amostragem de dados, com o objetivo de otimizar a execução de *data mining* [7]. A quarta etapa consiste na execução de algoritmos de aprendizagem para realizar extração de novos conhecimentos e padrões, processo este denominado *data mining*. Nesta etapa, podem ser aplicados algoritmos de classificação, agrupamento, regressão, entre outros [1]. Por fim, a última etapa é a avaliação do resultado, onde se interpreta e valida os padrões minerados [17], [3].

IV. ESTUDO DE CASO

Neste estudo o processo de KDD foi aplicado no banco de dados do sistema acadêmico do IFMS, com foco nos dados do Curso Superior de Tecnologia em Sistemas para Internet. Na primeira etapa do processo, a seleção de dados, foram definidas as tabelas do banco de dados necessárias para a obtenção dos atributos desejados para a geração de dois conjuntos de dados. O primeiro, denominado Conjunto de Dados 1, compreende os registros de alunos ingressantes entre 2011 e 2013. O segundo, denominado Conjunto de Dados 2, é composto de todos os alunos ingressantes entre 2014 e 2015. Ambos os conjuntos são compostos por atributos que descrevem a situação do aluno em disciplinas específicas do primeiro semestre do curso, bem como idade, estado civil e sexo de cada aluno. O atributo alvo deste processo é a situação do aluno na disciplina de Linguagem de Programação 1, que é ofertada no segundo semestre do curso. De acordo com o projeto pedagógico do curso, a disciplina de Linguagem de Programação 1 é considerada pré-requisito para diversas outras disciplinas posteriores, pois sua ementa abrange conhecimentos básicos de programação. Além disso, o objetivo geral do curso é formar profissionais com competências em desenvolvimento de sistemas e páginas de Internet. Logo, essa disciplina pode ser considerada chave nesse curso.

Posteriormente, como segunda etapa do processo, foi executado o pré-processamento, que verificou a qualidade dos dados e a existência de atributos nulos. Neste caso diversos registros foram excluídos, sendo mantidos apenas os alunos que concluíram todas as disciplinas analisadas: Algoritmos, Lógica Digital, Matemática, Páginas Web 1, e Linguagem de Programação 1. Na sequência, a transformação dos dados foi aplicada, onde os atributos nominais foram ajustados em sua forma de exibição, mas não em seu valor significativo. Neste caso, os atributos referentes às disciplinas, bem como o atributo `solteiro`, tiveram seus valores transformados em `S` (Sim) ou `N` (Não) e o atribuo `sexo` foi transformado em `M` (Masculino) ou `F` (Feminino). Após a execução dessas etapas o Conjunto de Dados 1 e o Conjunto de Dados 2 compreenderam, respectivamente, 26 e 99 registros.

Neste trabalho foi desenvolvida uma aplicação Python para automatizar a execução das etapas do processo de KDD descritas anteriormente: seleção de dados, pré-processamento de dados e transformação de dados. Esta aplicação permite gerar conjuntos de dados diretamente a partir do banco de dados acadêmico, tomando como entrada a data de matrícula dos alunos, as disciplinas que se deseja filtrar e o *campus* da instituição onde o curso é ofertado. A aplicação gera saída em

formato adequado para a execução dos passos posteriores, onde se utiliza a ferramenta Weka, cujo formato de arquivo é denominado ARFF e dispõe os atributos e seus possíveis valores no trecho `@relation` do arquivo, seguido pelo trecho de dados `@data`, onde se encontram os registros. Uma porção dos dados do Conjunto de Dados 1, no formato ARFF, é apresentada na Fig. 1.

```
@relation Aluno

@attribute programacao {S, N}
@attribute web1 {S, N}
@attribute matematica {S, N}
@attribute logica {S, N}
@attribute algoritmos {S, N}
@attribute sexo {M, F}
@attribute idade real
@attribute solteiro {S, N}

@data

N,S,S,S,S,M,34,S
N,S,S,S,N,M,37,N
S,S,S,S,S,M,25,S
N,S,N,S,N,F,30,S
S,S,S,S,S,M,30,S
S,S,S,S,S,M,48,N
S,S,N,N,S,M,20,S
N,S,N,N,N,M,30,S
S,S,S,S,S,M,35,S
S,S,N,N,N,F,22,S
...
```

Figura 1. Conjunto de Dados 1 no formato ARFF, gerado pela aplicação Python, contendo os atributos e os dados para execução da classificação.

Na execução de *data mining*, quarta etapa processo de KDD, foi realizada a aplicação da tarefa de classificação nos conjuntos de dados. Neste estudo de caso o método foi aplicado com objetivo de classificar alunos como aprovados ou reprovados em um atributo alvo, a disciplina Linguagem de Programação 1. Esta etapa considera que existem duas classes pré-existentes em que os alunos podem ser classificados: `S` (Sim), para o caso de aluno aprovado, e `N` (Não), para o caso de aluno reprovado.

O objetivo da classificação é encontrar relacionamentos entre os atributos e uma classe, de modo que o processo de classificação possa usar esse relacionamento para predizer a classe de um registro novo e desconhecido [18], [19]. Devido às suas características de simplicidade e compreensão, além de sua adequação aos objetivos deste estudo, optou-se por utilizar a técnica de árvore de decisão, que constrói uma árvore cujos nós representam atributos e as arestas representam os valores desses atributos. Ao classificar um registro do conjunto de dados, percorre-se um caminho na árvore que se inicia a partir do nó raiz, onde é realizada a condição de teste do nó, seguindo, posteriormente, para a ramificação apropriada com base no resultado do teste [15].

A ferramenta Weka foi utilizada para executar a classificação. Esta ferramenta possibilita a execução de diferentes algoritmos de *data mining* e, neste caso, o algoritmo J4.8 foi utilizado para a construção da árvore de decisão. Este algoritmo foi escolhido por suas características de versatilidade, que permitem classificar registros sem a necessidade de discretizar valores numéricos, como `idade`, nos conjuntos de dados gerados. O algoritmo J4.8 é uma implementação *open-source* em Java do algoritmo C4.5, que se baseia em uma estratégia que expande uma árvore de decisão escolhendo localmente quais atributos usar para particionar os dados [6].

V. TESTES E RESULTADOS

Definidos a tarefa, a técnica e o algoritmo para *data mining*, há ainda várias combinações possíveis para executar o algoritmo de classificação na ferramenta Weka. Três diferentes opções de execução foram utilizadas: `Use Training Set`, `Supplied Test Set` e `Cross-validation`. A opção `Use Training Set` permite que o conjunto de dados carregado seja utilizado para treinar o algoritmo e gerar um modelo de classificação. A opção `Supplied Test Set` permite que um novo conjunto de dados seja carregado para testar a classificação com o modelo gerado. A opção `Cross-validation` divide o conjunto de dados carregado de forma aleatória para realizar o treinamento e executar o teste da classificação.

Foram realizadas duas execuções do algoritmo J4.8 para cada opção descrita anteriormente, totalizando seis execuções. Em todas as opções o atributo alvo foi sempre o mesmo, `programacao`, e os atributos base também foram mantidos, sendo eles: `algoritmos`, `web1`, `matematica`, `logica`, `solteiro`, `idade` e `sexo`. Para cada opção, o algoritmo teve seu parâmetro `prune` alterado entre `verdadeiro` e `falso`, isto é, o algoritmo de classificação gerou versões da árvore de decisão com poda e sem poda. A poda em uma árvore de decisão tem o objetivo de gerar uma árvore mais simples e compacta, com precisão próxima da árvore original. Além disso, a poda permite aumentar a capacidade de generalização da árvore de decisão [3].

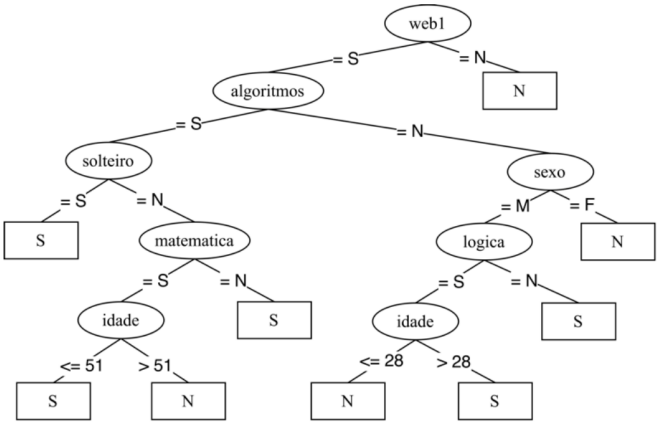


Figura 2. Representação gráfica da árvore de classificação (sem poda) utilizando a opção `Use Training Set`.

Na execução `Use Training Set`, foi utilizado o Conjunto de Dados 2 e foram geradas árvores de decisão com poda e sem poda. Neste caso o sucesso foi maior na versão da árvore sem poda, que é apresentada na Fig. 2, em que todos os atributos foram considerados. A matriz de confusão da versão sem poda apresenta 84,8% de acerto, enquanto a matriz de confusão da árvore com poda apresenta 82,8% de acerto, como

apresentado na Tabela I. Ambas as taxas de acerto são consideradas satisfatórias, mas o uso da opção `Use Training Set` indica que os mesmos dados utilizados para realizar o treinamento são utilizados para realizar a tarefa de classificação, gerando-se assim um resultado otimista para a classificação [3].

Na matriz de confusão da árvore sem poda, apresentada na Tabela I (1), é possível observar que 70 alunos foram classificados como aprovados e 14 como reprovados. Ainda é possível observar que um aluno foi classificado como reprovado, mas, no conjunto de dados este registro consta como aprovado, e que 14 alunos foram classificados como aprovados, porém, são reprovados no conjunto de dados. Em outras palavras, o algoritmo classificou 84 instâncias corretamente, ou seja, uma taxa de acerto de 84,8%. Na matriz de confusão da árvore com poda, apresentada na Tabela I (2), é possível observar que a taxa de acerto é menor, com valor de 82,8%.

TABELA I
SAÍDAS DAS EXECUÇÕES USE TRAINING SET DO ALGORITMO J4.8 SEM PODA (1) E COM PODA (2). AS MATRIZES DE CONFUSÃO MOSTRAM A DISTRIBUIÇÃO DE INSTÂNCIAS CLASSIFICADAS CORRETA E INCORRETAMENTE.

Use Training Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamentes	84	84,8%
Instâncias Classificadas Incorretamentes	15	15,2%

Matriz de confusão		
a	**b**	<-- Classificado como
70	1	a = S (aprovado)
14	14	b = N (reprovado)

(1)

Use Training Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamentes	82	82,8%
Instâncias Classificadas Incorretamentes	17	17,2%

Matriz de confusão		
a	**b**	<-- Classificado como
70	1	a = S (aprovado)
16	12	b = N (reprovado)

(2)

Os testes realizados com a opção `Supplied Test Set` utilizaram o Conjunto de Dados 1 para treinar o algoritmo de classificação e o Conjunto de Dados 2 para realizar o teste da classificação. Neste caso, também foram geradas as versões das árvores com poda e sem poda no algoritmo J4.8. Os resultados das classificações são apresentados na Tabela II, onde pode ser visto que os resultados com poda e sem poda foram os mesmos, com taxa de acerto de 72,7%. Na matriz de confusão pode ser visto que 67 alunos foram classificados corretamente como aprovado e 5 alunos foram classificados corretamente como reprovados, restando 27 alunos classificados incorretamente.

Ambas as árvores de decisão geradas, com poda e sem poda, se mantiveram com três nós internos, como pode ser visto na Fig. 3, e somente os atributos `algoritmos`, `web1` e `idade` foram considerados.

Além dos testes utilizando as opções `Use Training Set` e `Supplied Test Set`, também foram feitos testes com a opção `Cross-validation`, que realiza o treinamento e o teste de classificação em um mesmo conjunto de dados. Neste caso foi configurado o parâmetro `Fold` em 10, o que significa que o algoritmo de classificação divide o conjunto de dados em dez partes, utilizando, inicialmente, nove partes para realizar o aprendizado e uma parte para executar o teste. Em seguida é utilizada uma parte diferente para teste e um conjunto diferente de nove partes para treinamento. O processo é repetido, substituindo as partes entre aprendizado e teste, até que todas as dez partes tenham sido utilizadas para teste.

A escolha do valor 10 do parâmetro `Fold` decorre de razões empíricas: testes extensivos em diferentes bases de dados, utilizando técnicas de aprendizado diferentes mostram que 10 é o número ideal para se obter a melhor estimativa de erro [3]. Não há, de fato, argumentos teóricos que caracterizem o parâmetro `Fold` de forma conclusiva, mas a escolha do valor 10 tem se tornado um padrão em termos práticos [3].

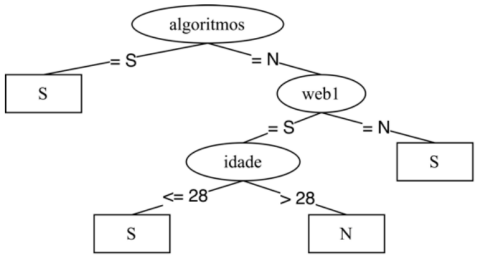


Figura 3. Representação gráfica da árvore de classificação (com poda e sem poda) utilizando a opção `Supplied Test Set`.

TABELA II
SAÍDAS DAS EXECUÇÕES SUPPLIED TEST SET DO ALGORITMO J4.8 SEM PODA (1) E COM PODA (2). AS MATRIZES DE CONFUSÃO MOSTRAM A DISTRIBUIÇÃO DE INSTÂNCIAS CLASSIFICADAS CORRETA E INCORRETAMENTE.

Supplied Test Set: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamentes	72	72,7%
Instâncias Classificadas Incorretamentes	27	27,3%

Matriz de confusão		
a	**b**	<-- Classificado como
67	4	a = S (aprovado)
23	5	b = N (reprovado)

(1)

Supplied Test Set: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamentes	72	72,7%
Instâncias Classificadas Incorretamentes	27	27,3%

Matriz de confusão		
a	**b**	<-- Classificado como
67	4	a = S (aprovado)
23	5	b = N (reprovado)

(2)

Como não foram fixados subconjuntos de treinamento e teste específicos, o método `Cross-validation` fornece

estimativas de erro mais realistas para a construção da árvore de classificação. Neste teste foi utilizado o Conjunto de Dados 2 e, como nos demais testes, também foram verificadas as opções com poda e sem poda da árvore. Na Tabela III podem ser vistos os resultados das classificações nas versões sem poda (1) e com poda (2). Pelas características de generalização da árvore com poda, esta versão gerou melhor resultado, obtendo uma taxa de acerto de 75,8% na classificação, ou seja, 75 dos 99 registros foram classificados corretamente, como pode ser visto na Tabela III (2). A árvore com poda possui cinco nós internos e pode ser vista na Fig. 4, em que os atributos considerados foram: `web1`, `algoritmos`, `sexo`, `logica` e `idade`.

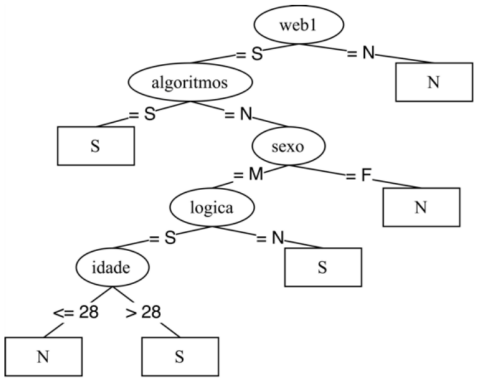


Figura 4. Representação gráfica da árvore de classificação (com poda) utilizando a opção `Cross-validation`.

TABELA III
SAÍDAS DAS EXECUÇÕES CROSS-VALIDATION DO ALGORITMO J4.8 SEM PODA (1) E COM PODA (2). AS MATRIZES DE CONFUSÃO MOTRAM A DISTRIBUIÇÃO DE INSTÂNCIAS CLASSIFICADAS CORRETA E INCORRETAMENTE.

Cross-validation: `unpruned=True` (sem poda)		
Instâncias Classificadas Corretamentes	74	74,7%
Instâncias Classificadas Incorretamentes	25	25,3%

Matriz de confusão		
a	**b**	<-- Classificado como
65	6	a = S (aprovado)
19	9	b = N (reprovado)

(1)

Cross-validation: `unpruned=False` (com poda)		
Instâncias Classificadas Corretamentes	75	75,8%
Instâncias Classificadas Incorretamentes	24	24,2%

Matriz de confusão		
a	**b**	<-- Classificado como
66	5	a = S (aprovado)
19	9	b = N (reprovado)

(2)

Os resultados mostram ainda que a aplicação de *data mining* em dados acadêmicos pode obter bastante sucesso, possibilitando a descoberta de conhecimento em dados que eram simplesmente armazenados. Dentre as informações extraídas, verificou-se que o desempenho do aluno na disciplina de Páginas Web 1 diz mais sobre o resultado que ele terá em Linguagem de Programação 1 do que a própria disciplina de Algoritmos, o que não era esperado.

Como árvores de decisão descartam nós cujos valores são menos significativos para a classificação [21], pode ser observado pela análise das árvores geradas que a idade do aluno foi considerada mais significativa que o desempenho do aluno em disciplinas como Lógica Digital e Matemática, uma vez que o atributo `idade` é uma condição testada em todas as árvores geradas, enquanto `logica` é um atributo presente nas árvores `Use Training Set` e `Cross-validation` e `matematica` é um atributo testado somente na árvore `Use Training Set`.

VI. CONCLUSÃO

Com os testes realizados, foi possível confirmar que a execução `Use Training Set` aponta a melhor taxa de acerto na tarefa de classificação. No entanto, por utilizar o mesmo conjunto de dados para realizar o treinamento e o teste da classificação, este resultado não é considerado realista. Os resultados com a opção `Cross-validation` ocupam o segundo lugar na lista de maior taxa de acertos do algoritmo e é um bom resultado, já que esta opção utiliza partes diferentes do conjunto de dados para treinamento e teste. Entre todos os testes realizados, a opção com menor percentual de acerto é a `Supplied Test Set`, pois utiliza os Conjuntos de Dados 1 e 2, relativos as turmas e anos diferentes do curso, para o treinamento e a classificação.

Neste estudo concluiu-se que a melhor opção para classificação utilizando o algoritmo J4.8 é a `Cross-validation`, mesmo que seu percentual de classificações corretas não seja o maior. Esta opção é a mais próxima da realidade, pois o treinamento e o teste são realizados diversas vezes com diferentes partes do conjunto de dados. Além disso, dentre as versões com poda e sem poda da árvore de decisão gerada, a que garantiu maior taxa de acerto foi a versão com poda, uma vez que ela possui uma maior capacidade de generalização na classificação. Assim, considera-se o melhor resultado a execução de `Cross-validation` com poda, que obteve uma taxa de acerto de 75,8%.

Com os resultados obtidos, conclui-se que é possível prever o desempenho dos alunos em casos específicos e, com isso, tomar atitudes antecipadas para tentar diminuir a reprovação de alguns alunos e consequentemente diminuir a evasão por reprovação no curso.

Apesar da complexidade do banco de dados estudado e da grande incidência de dados incompletos, a descoberta de informações relevantes para o contexto escolar foi viável. Espera-se, como trabalhos de pesquisas futuras, o desenvolvimento de um módulo para o sistema acadêmico do IFMS que possibilite a previsão e prevenção de reprovações e, consequentemente, a diminuição da evasão escolar.

REFERÊNCIAS

[1] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, “From data mining to knowledge discovery databases.,” AI Magazine, pp. 37-53, 1996.

[2] “Índice Brasscom de Convergência Digital,” 2015. [Online]. Available: www.brasscom.org.br. [Accessed: 14-May-2016].

[3] H. Witten, E. Frank and M. A. Hall, Data mining: Practical Machine Learning Tools and Techniques. Burlington: Morgan Kaufmann, 2011.

[4] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," in *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10-18, Jun. 2009.

[5] B. K. Baradwaj and S. Pal, "Mining Educational Data to Analyze Students' Performance," in *International Journal of Advanced Computer Science and Applications* – IJACSA, vol. 2, no. 6, pp. 63-69, 2011.

[6] J. R. Quinlan, C4.5, San Francisco: Morgan Kaufmann Publishers Inc., 1993.

[7] E. L. Ferreira, H. Rausch, S. Campos, A. Faria-Campos, E. Pietra and L. da Silva Santos, "Medical data mining: A case study of a Paracoccidioidomycosis patient's database," *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, Natal, 2014, pp. 275-280.

[8] F. H. D. Araujo, A. M. Santana and P. A. S. Neto, "Evaluation of Classifiers Based on Decision Tree for Learning Medical Claim Process," in *IEEE Latin America Transactions*, vol. 13, no. 1, pp. 299-306, Jan. 2015.

[9] L. A. E. Silva, "A Data Mining Approach for Standardization of Collectors Names in Herbarium Database," in *IEEE Latin America Transactions*, vol. 14, no. 2, pp. 805-810, Feb. 2016.

[10] S. Pan, T. Morris and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," in *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104-3113, Nov. 2015.

[11] Y. Amaya, D. Heredia and E. Barrientos, "Student dropout predictive model using data mining techniques," in *IEEE Latin America Transactions*, vol. 13, no. 9, pp. 3127-3134, Sept. 2015.

[12] Nandeshwar, T. Menzies and A. Nelson, "Learning patterns of university student retention," in *Expert Systems with Applications*, vol. 38, no. 12, pp. 14984-14996, Nov-Dec. 2011.

[13] Ktona, D. Xhaja and I. Ninka, "Extracting Relationships between Students' Academic Performance and Their Area of Interest Using Data Mining Techniques," *Computational Intelligence, Communication Systems and Networks (CICSyN), 2014 Sixth International Conference on*, Tetova, 2014, pp. 6-11.

[14] C. Márquez-Vera, A. Cano, C. Romero, A. Y. M. Noaman, H. M. Fardoun and S. Ventura, "Early dropout prediction using data mining: a case study with high school students," in *Expert Systems*, vol. 33, no. 1, pp. 107-124, Feb. 2016.

[15] U. Fayyad and R. Uthurusamy, “Evolving data mining into solutions for insights.,” Communications of the ACM, vol. 45, pp. 28-31, 2002.

[16] P. R. M. Cereda and J. Jose, "Adaptive data mining: preliminary studies," in *IEEE Latin America Transactions*, vol. 12, no. 7, pp. 1258-1270, Oct. 2014.

[17] N. Boente, R. R. Goldschmidt and V. V. Estrela, " Uma metodologia de suporte ao processo de descoberta de conhecimento em bases de dados," in *Anais do Simpósio de Excelência em Gestão e Tecnologia* – SEGeT 2008, Rio de Janeiro, Brasil, Outubro 2008.

[18] G. Kesavaraj and S. Sukumaran, "A study on classification techniques in data mining," *Computing, Communications and Networking Technologies (ICCCNT 2013) Fourth International Conference on*, Tiruchengode, 2013, pp. 1-7.

[19] Y. Ren, L. Zhang and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions [Review Article]," in *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 41-53, Feb. 2016.

[20] M. Oded, and L. Rokach, eds. Data mining and knowledge discovery handbook. Vol. 2. New York: Springer, 2010.

[21] P. N. Tan, M. Steinbach and V. Kumar. Introduction to Data Mining, (First Edition). Boston: Addison-Wesley, 2005.

[22] J. L. C. Ramos, R. E. D. Silva, R. L. Rodrigues, J. C. S. Silva and A. S. Gomes, "A Comparative Study between Clustering Methods in Educational Data Mining," in *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3755-3761, Aug. 2016.

Maraísa da Silva Guerra possui graduação em Sistemas de Informação pela Universidade Estadual do Norte do Paraná (2007), especialização pela Universidade do Norte do Paraná (2009). Tem experiência em desenvolvimento de software e banco de dados. Atualmente é professora efetiva do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul e aluna de mestrado na Universidade Estadual Paulista “Júlio de Mesquita Filho”.

Habib Asseiss Neto possui graduação em Ciência da Computação pela Universidade Federal de Mato Grosso do Sul (2008), mestrado pela Universidade Federal de Mato Grosso do Sul (2012). Tem experiência em bioinformática, mineração de dados e desenvolvimento de aplicações web. Atualmente é professor efetivo do Instituto Federal de Educação, Ciência e Tecnologia de Mato Grosso do Sul e aluno de doutorado na Universidade Federal de Minas Gerais.

Sérgio Azevedo de Oliveira possui graduação em Engenharia Elétrica pela Universidade Estadual Paulista “Júlio de Mesquita Filho” (1981), mestrado pela Universidade Federal de Santa Catarina (1989), doutorado pela Universidade Estadual de Campinas (2004) e pós-doutorado pela Universidad de Castilla-La Mancha - Espanha (2011). Atualmente é professor assistente doutor da Universidade Estadual Paulista “Júlio de Mesquita Filho”. Tem experiência na área de Engenharia Elétrica atuando principalmente nos seguintes temas: software educacional, Linux, processamento paralelo e distribuído, planejamento da expansão da transmissão, meta-heurísticas combinatórias, otimização ordinal e vulnerabilidade dos sistemas de energia elétrica.