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

M. S. Guerra, H. Asseiss Neto 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 boas aliadas 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].

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

H. Asseiss Neto, 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

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 modelo 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 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-se 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 U*se 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].

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].

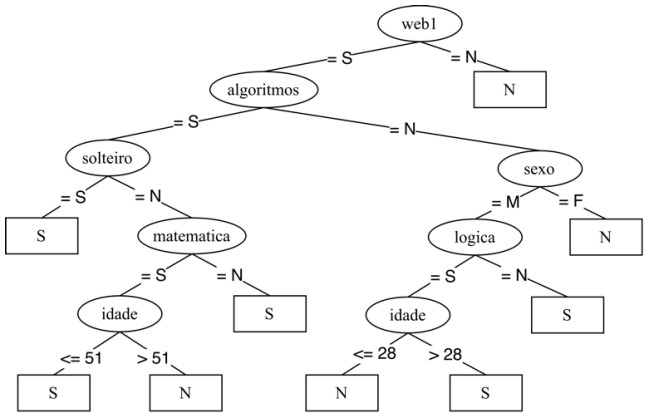


Figura 2. Representação gráfica da árvore de classificação (sem poda) utilizando a opção *Use Training Set*.

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 pala-

vras, 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].

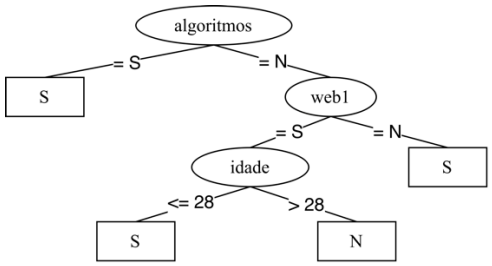


Figura 3. Representação gráfica da árvore de classificação (com poda e sem poda) utilizando a opção *Supplied Test Set*.

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`.

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 Pá-

ginas 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.

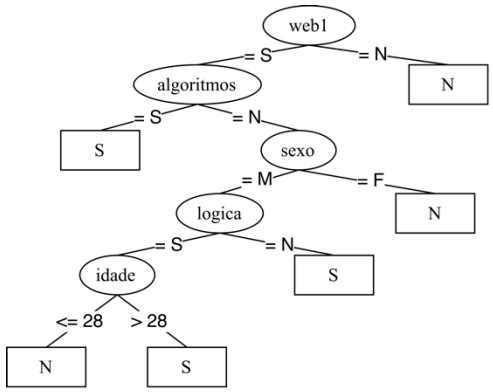


Figura 4. Representação gráfica da árvore de classificação (com poda) utilizando a opção *Cross-validation*.

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)

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.