A Mineração de Dados Educacionais como Apoio na Análise e Compreensão do Processo de Aprendizagem

Jhonatan de Paula Candão
Universidade Estadual de Mato Grosso do Sul – UEMS
Nova Andradina, Brasil
jhonatan.candao@gmail.com

Eduardo Machado Real
Universidade Estadual de Mato Grosso do Sul – UEMS
Nova Andradina, Brasil
eduardomreal@uems.br

ABSTRACT
Often, an efficient analysis of student data can obtain important information on how the teaching and learning process of a course, or even specific course subjects, may be being conducted. In this case, the Educational Data Mining can direct managers and teachers to plan better actions that aim at good decision making. Among the methods that this application uses, are the algorithms of Machine Learning. In this context, this paper aims to present a study done out from the application of Educational Data Mining from distinct tasks of machine learning (classification, grouping and association rules), in real data sets of two undergraduate courses. As result, the adopted schemes and some analysis of the obtained results are shown.

RESUMO
Muitas vezes, uma análise eficiente sobre dados de estudantes pode obter importantes informações de como o processo de ensino e aprendizado de um curso, ou mesmo disciplinas específicas, pode estar sendo conduzido. Nisso, a Mineração de Dados Educacionais pode direcionar gestores e professores ao planejamento de melhores ações que visem boas tomadas de decisões. Dentre os métodos que tal aplicação utiliza, estão os algoritmos de Aprendizado de Máquina. Neste contexto, este trabalho tem por objetivo apresentar um estudo realizado a partir da aplicação da Mineração de Dados Educacionais a partir de tarefas de aprendizado de máquina distintos (classificação, agrupamento e regras de associação), sobre conjunto de dados reais de dois cursos de graduação. Como resultado, são mostrados os esquemas adotados e algumas análises dos resultados obtidos.

Author Keywords
Mineração de Dados, Aprendizado de Máquina, Mineração Dados Educacionais.

ACM Classification Keywords
H.2.8 [Database Applications] Data mining; J. [Computer Applications] Education.

INTRODUÇÃO

Os dados de estudantes podem ser gerados a partir de, por exemplo, sistemas de informações para administração escolar, sistemas de educação à distância, controles de disciplinas ao longo do ano, ambientes virtuais de aprendizagem etc. Assim, uma das dificuldades é a de como obter conhecimentos a partir desses dados, sabendo-se que muitas das informações existentes neles são importantes. E ainda, tais informações geralmente estão "ocultas" e apenas a análise "manual" de um especialista torna este processo custoso e pode não ser totalmente eficiente.

Para este domínio de dados é aplicada a Mineração de Dados Educacionais (MDE), que pode utilizar métodos de Aprendizado de Máquina (AM) que realizam, por exemplo, de tarefas de Classificação, Agrupamento e Regras de Associação nos dados. Com a MDE busca-se, por exemplo, avaliar a evasão do curso, avaliar as atividades realizadas pelos alunos, bem como a participação dos mesmos, recomendar conteúdo apropriado para o momento educacional vivenciado pelo estudante, antever o resultado de testes e de outras avaliações educacionais, com base na análise das atividades realizadas pelo estudante, agrupar estudantes, levantar o perfil dos estudantes, reavaliar o ensino-aprendizagem através do desempenho dos estudantes, entre outras.

Assim, este trabalho tem por objetivo principal mostrar a viabilidade da aplicação da MD sobre o domínio de dados de estudantes, a MDE. Para isso, foram utilizados três algoritmos de AM que realizam tarefas distintas, ou seja, de classificação (algoritmo J48, uma implementação do C4.5 [1]), de agrupamento (algoritmo k-means [2]) e de regras de associação (algoritmo Apriori [3]), aplicados sobre conjuntos de dados reais.

MINERAÇÃO DE DADOS

De maneira geral, o processo da Mineração de Dados (MD) está relacionado aos seguintes "pilares" gerais: (1) os dados (coleta e armazenamento), (2) a informação (dado analisado e com algum significado) e (3) o conhecimento (informação interpretada e aplicada).

Os autores em [4] destacam que a Mineração de Dados tem atraído muita atenção na indústria da informação e na sociedade como um todo nos últimos anos, devido à ampla disponibilidade de enormes quantidades de dados e à necessidade iminente de transformar esses dados em informação e conhecimento úteis.

Existem diferentes metodologias para a aplicação da MD, tais como o KDD [5]. Essa metodologia é definida em fase. Essas fases, ou etapas, têm por proposta ajudar a implementação do processo da MD. A seguir são definidos

alguns conceitos do KDD, o qual talvez seja a metodologia mais conhecida.

O processo de KDD tem várias definições, em [5], diz que "KDD é um processo, não trivial, de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados". Na figura 1, são visualizadas as fases do KDD que descrevem todo o processo proposto.

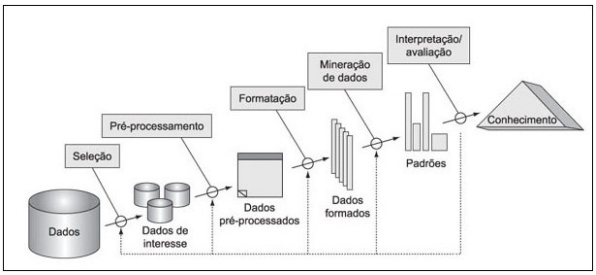

Figura 1. Fases do processo KDD. Fonte [5].

As fases são definidas como segue [5]:

- Seleção: é onde se faz a seleção dos dados. Esta fase afeta diretamente a qualidade do resultado final, pois é onde se define quais dados, com suas possíveis variáveis (atributos) farão parte desta seleção. É uma fase muito complexa, visto que esses dados podem vir de fontes e estruturas diferentes (arquivos-texto, banco de dados, relatórios, logs de acesso, transações etc).
- Pré-processamento (ou "limpeza"): É uma fase onde são tratados os dados que contêm algum tipo de problema, tais como os dados com valores ausentes ou discrepantes, e isso tem fator determinante na efetividade do algoritmo escolhido.
- Transformação (ou "formatação"): É a conversão dos dados em um formato comum, para a aplicação dos algoritmos. Se necessário, aqui pode se obter informações que faltam através da combinação ou transformação, que são os dados derivados, como, por exemplo, a idade de uma pessoa, que pode ser calculada através de sua data de nascimento.
- Mineração de Dados: São usadas várias estratégias para a visualização e diferentes técnicas de mineração, como a aplicação de algoritmos de Aprendizado de Máquina de classificação, agrupamento, regra de associação etc.
- Interpretação / Avaliação: São usadas variadas técnicas de interpretação e avaliação dos dados, isso depende do campo de pesquisa, mas todos com um intuito final: a informação.

A MD pode ser aplicada a diversos e diferentes domínios e contextos, tal como na Educação, utilizando-se de métodos de Aprendizado de Máquina (AM).

APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial (IA) que trata do desenvolvimento e a aplicação de métodos capazes de fazer a máquina predizer ou descrever situações a partir de experiências passadas, com exemplos, aprendidas através de conjuntos de dados.

De acordo com [6], o AM é definido como programas de computadores que aprimoraram um critério de desempenho usando informações passados, chamadas de exemplos ou simplesmente dados. O AM é um processo de aprendizagem que, a partir de um conjunto de instâncias fornecidas (que também podem ser nomeadas de dados, objetos, exemplos ou padrões), produz um conceito representado por essas instâncias.

Conforme [7], o AM fornece a base técnica da MD. Ele é usado para extrair informações dos dados brutos em conjuntos de dados - informações expressas de forma compreensível e que podem ser usadas para diversas finalidades. Assim, o AM é como a aquisição de descrições estruturais a partir de exemplos. O tipo de descrições encontrado pode ser usado para previsão, explicação e compreensão.

Em AM, as tarefas são divididas em modelos preditivos e descritivos. Nos preditivos os algoritmos induzem classificadores criados com base nos objetos (exemplos) os quais possuem a informação da classe. Para isso, os classificadores são construídos a partir de conjuntos de dados de treinamentos, que podem ser avaliados dando como entrada conjuntos de dados testes. Já nos descritivos, os objetos de um conjunto de dados não têm a informação da classe e o objetivo é então descrevê-los, como, por exemplo, agrupando-os conforme as semelhanças de cada um (similaridades ou dissimilaridades)

Nesse contexto, pode-se definir que os tipos de métodos/algoritmos que implementam AM estão relacionados ao nível de supervisão, a saber aqueles considerados neste trabalho: Aprendizado supervisionado (realiza tarefa preditiva) ou Aprendizado não-supervisionado (realiza tarefa descritiva).

A Figura 2 apresenta uma hierarquia de aprendizado de acordo com os tipos de tarefas de aprendizado e taxonomia do nível de supervisão dos métodos.

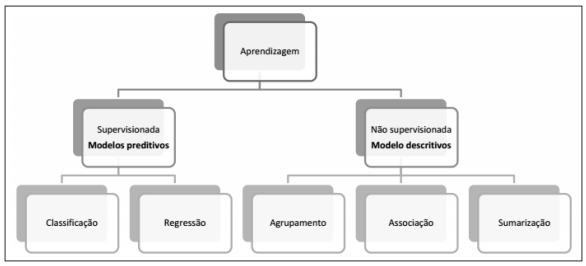

Figura 2. Tipos de métodos de AM. Fonte: [8].

Aprendizado de Máquina Supervisionado

Os autores em [9] definem que no AM supervisionado, um especialista fornece um rótulo de categoria ou custo para cada padrão em um conjunto de treinamento, e procura-se reduzir a soma dos custos para esses padrões.

De acordo como descrito em [10], os algoritmos de AM supervisionado fazem uso de uma informação extra, chamada classe (ou categoria), que faz parte da descrição de cada dado de treinamento. A classe de cada dado é, geralmente, fornecida por uma fonte externa ao processo de aprendizado (por exemplo, por um especialista humano na área de conhecimento em questão).

Aprendizado de Máquina Não-supervisionado

Em [11] está contextualizado que, como a informação da classe (rótulo) nem sempre está disponível, então um conjunto de dados é tido como entrada, cujo objetivo é desvendar os dados semelhantes para agrupamento. Isso é conhecido como reconhecimento de padrões não supervisionado ou aprendizado ou agrupamento não supervisionado.

De acordo com [10], os algoritmos de aprendizado não-supervisionado não fazem uso da informação dada pela classe e, por essa razão, são tipicamente usados para a inferência do conceito a partir de dados cuja descrição não incorpora a classe do conceito que representam. Algoritmos não supervisionados geralmente aprendem por meio da identificação de subconjuntos de dados que compartilham certas similaridades.

MINERAÇÃO DE DADOS EDUCACIONAIS

Em [12] é definido que, quando a MD é usada para a aplicação na área da educação, se dá a origem à linha de pesquisa de Mineração de Dados Educacionais (MDE). As informações armazenadas em ambientes educacionais constituem fontes riquíssimas de conhecimento que podem ser analisadas através de mineração de dados.

Neste sentido, os autores em [13], descrevem que tal processo de descoberta de conhecimento pode auxiliar professores a conduzirem melhor suas turmas, identificando dificuldades, compreendendo melhor o processo de aprendizagem dos estudantes e melhorando os métodos de ensino. Como sequência, os professores podem oferecer um feedback mais apropriado aos estudantes através de reflexões pertinentes a suas aprendizagens.

Na literatura existem diversos autores que definem a MDE e relatam os resultados de suas pesquisas, onde a MDE foi aplicada utilizando os mais diversos métodos de AM disponíveis, tais como em [12,13,14,15,16].

Exemplo de Aplicação

Para exemplificar, esta subseção apresenta dois experimentos a partir do conjunto de dados *Student Performance*[1], disponível pelo *UCI Repository*[2], usando o algoritmo de Classificação em árvore de decisão J48 [1] e o algoritmo de Regras de Associação Apriori [3], implementações disponíveis pelo software *Weka*[3] (desenvolvido pelo *Machine Learning Project* (*Department of CS of The University of Waikato*) e descrito em [17]). O conjunto de dados descreve o desempenho dos alunos no ensino secundário de duas escolas portuguesas. Os atributos incluem: notas de alunos, características demográficas, sociais e relacionadas à escola e foram coletados usando relatórios escolares e questionários. Dois conjuntos de dados são fornecidos sobre o desempenho, correspondentes a dois assuntos distintos, *student-mat.csv* e *student-por.csv*. No entanto, para este trabalho foi utilizado apenas o *student-mat.csv*, que possui 395 objetos, descritos por 32 atributos + a classe "G3" (em um intervalo [0, 20]).

Classifier output

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

Correctly Classified Instances         272               68.8608 %
Incorrectly Classified Instances       123               31.1392 %
Kappa statistic                          0.6586
Mean absolute error                      0.0407
Root mean squared error                  0.1427
Relative absolute error                 39.9852 %
Root relative squared error             63.2752 %
Total Number of Instances              395
```

Figura 3. Resultados gerados pelo J48.

A Figura 3 mostra os resultados gerados a partir do J48, configurado para a opção *Use training set*. Veja que 68,86% dos objetos foram classificados corretamente, e isso é um bom valor, considerando que o algoritmo tratou de um problema com 21 possíveis classes em apenas 395 instâncias (talvez, poderiam ser criados intervalos entre as 21 classes, ficando, assim, com 3, 4 ou 5 classes, por exemplo). Outra análise interessante seria visualizar a árvore de decisão criada (devido ao tamanho, não foi possível inserir neste texto), onde um modelo de predição é criado conforme os valores dos atributos.

```
Best rules found:

 1. failures=0 312 ==> higher=yes 305    conf:(0.98)
 2. nursery=yes 314 ==> higher=yes 300    conf:(0.96)
 3. internet=yes 329 ==> higher=yes 313    conf:(0.95)
 4. school=GP 349 ==> higher=yes 332    conf:(0.95)
 5. Pstatus=T 354 ==> higher=yes 335    conf:(0.95)
 6. schoolsup=no 344 ==> higher=yes 325    conf:(0.94)
 7. internet=yes 329 ==> Pstatus=T 298    conf:(0.91)
 8. internet=yes 329 ==> school=GP 297    conf:(0.9)
 9. schoolsup=no 344 ==> Pstatus=T 310    conf:(0.9)
10. higher=yes 375 ==> Pstatus=T 335    conf:(0.89)
```

Figura 4. Resultados gerados pelo Apriori.

A Figura 4 mostra um experimento a partir do Apriori (configurado com: suporte (*support*) de 75%, confiança (*confidence*) de 80% e geração de apenas 10 melhores regras). Observando as regras de associação geradas, todas elas são relativamente pequenas, mas isto não quer dizer

[1] https://archive.ics.uci.edu/ml/datasets/Student+Performance#

[2] UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php

[3] https://www.cs.waikato.ac.nz/ml/weka/

que são fracas. Neste caso, mais uma vez, seria necessário antes saber o que se quer, ou seja, formular algumas questões cujas respostas podem ser obtidas a partir das regras. Tomando como exemplo a regra 5, veja que "Pstatus=T" (status de coabitação dos pais - 'T' - morando juntos ou 'A' - separados) implica em "higher=yes", com 95% de confiança que isso acontece. O fato é que, no caso do Apriori, também é importante gerar uma quantidade maior de regras.

EXPERIMENTO E ANÁLISE DE RESULTADOS

Esta seção descreve os experimentos e análises realizados, os quais ocorreram sobre dois conjuntos de dados de estudantes, denominados *xxx-students-2017.arff* e *xxx-students-2018.arff*, originados de dois cursos de graduação cada um, curso 1 e curso 2)[4]. Foram utilizados três algoritmos, a saber: o de Classificação (árvore de decisão) J48 [1]; o de Agrupamento K-Means [2] e o de regras de Associação Apriori [3], a partirdas implementações disponíveis pelo software *Weka*.

O conjunto de dados *xxx-students-2017.arff* foi originado a partir de um questionário composto por 19 questões, aplicado a estudantes da 1ª série dos dois cursos durante o mês de março de 2017, sendo finalizado em fevereiro de 2018, quando foi possível o preenchimento da informação do atributo classe de cada objeto de estudante. São 58 objetos descritos por 19 atributos (equivalentes a cada questão do questionário aplicado, ou seja, q1, q2, ..., q19) + *target* (classe). O atributo da classe, que seria o 20º atributo, é aquele que indica se o aluno evadiu (1) ou não evadiu (0) do curso no qual ingressou. Nisso, contém 30 estudantes que não evadiram e 28 que evadiram (coincidentemente, um conjunto equilibrado).

O primeiro deles foi usando o algoritmo J48, aplicando-o a partir de três ambientes de execução (esquema na parte superior da Figura 5), veja:

(1) Com *Use Training set*: o conjunto de dados *xxx-students-2017.arff* foi usado completo para treino e teste, com opção *unpruned*, para criar o Modelo. Assim, um novo conjunto de dados, o *xxx-students-2018.arff*, foi aplicado ao Modelo criado para que fosse realizada a predição.

(2) Com *Cross validation*: o conjunto de dados *xxx-students-2017.arff* foi aplicado sobre 10-*folds cross validation* (método *k-folds cross validation*), com a proporção de 9/10 para treino e 1/10 para validação do Modelo criado, em um processo que se repetiu dez vezes nos subconjuntos exclusivos. Veja que um novo conjunto de dados, o *xxx-students-2018.arff*, foi aplicado ao Modelo criado para que fosse realizada a predição.

(3) Com *Percentage Split*: o conjunto de dados *xxx-students-2017.arff* foi dividido (*%Split*) em dois, um conjunto de dados treino (65%) e um conjunto de dados teste (35%), com opção *unpruned*. A partir do conjunto de dados treino, um Modelo foi criado e o desempenho deste verificado usando o conjunto de dados teste. Após isso, um novo conjunto de dados, o *xxx-students-2018.arff*, foi aplicado ao Modelo para que fosse realizada a predição.

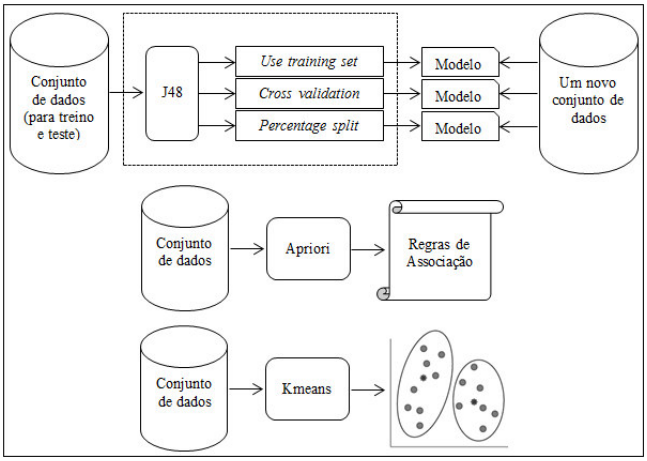

Figura 5. Ambientes de execução dos experimentos.

Este novo conjunto de dados, o *xxx-students-2018.arff* usado no último passo de cada ambiente de execução para o J48, foi gerado a partir da mesma estrutura do questionário referente ao *xxx-students-2017.arff*, porém aplicado aos estudantes em março de 2018. Contém 47 objetos e, obviamente, ainda sem a informação da classe (0 ou 1), cuja coleta deverá ser realizada apenas em fevereiro de 2019.

A Tabela 1 apresenta os resultados obtidos com o J48, isto é, a porcentagem de objetos que foram corretamente classificados com base na validação do conjunto de dados *xxx-students-2017.arff*, a qual ocorreu a partir dos resultados correspondentes à fase de criação de Modelo. Note que há diferenças de desempenhos, no entanto, diante da estrutura e características do conjunto de dados e considerando que não houve qualquer tipo de transformação ou pré-processamento nele, todos os ambientes tiveram um desempenho eficiente.

Por exemplo, o modelo construído a partir do *Use training set*, obteve a predição correta em 93.10% dos objetos, um bom valor obtido, levando em consideração que não houve qualquer tipo de "ajustes" nos dados, ou seja, por exemplo, não houve a retirada de nenhuma das 19 questões (há questões, talvez, com alta correlação e/ou que podem influenciar os resultados), mesmo considerando o conjunto de dados como pequeno.

Test options	% de objetos corretamente classificados
Use training set	93.10
Cross-validation	53.44
Percentage Split	55

Tabela 1. Resultados de % de instâncias corretamente classificadas em cada ambiente de execução com o J48.

[4] https://github.com/jhonatancandao/tcc_students_uems

Desta forma, de maneira simplificada, o objetivo neste primeiro experimento foi o de apresentar uma maneira de predizer a informação da classe de cada estudante do conjunto de dados de 2018 (*xxx-students-2018.arff*), a partir dos modelos treinados e criados pelo J48 sobre o conjunto de dados de 2017 (*xxx-students-2017.arff*). No entanto, obviamente, o ideal é explorar as diversas maneiras a partir de um tratamento mais sofisticado nos dados, com base nas técnicas descritas, por exemplo, em [4]. Assim, após o último passo de cada ambiente de execução, foi possível obter que, provavelmente, os estudantes (correspondentes aos objetos do conjunto de dados *xxx-students-2018.arff*) de números 3, 8, 17, 19, 21, 23, 26, 29, 31, 32, 34, 36, 37, 41, 43 e 44 podem evadir dos cursos 1 e 2 (34%).

O segundo experimento ocorreu usando o SimpleKmeans, implementação do algoritmo Kmeans (esquema na parte central da Figura 5). Neste caso, foram simuladas execuções considerando que o real número de grupos (classes 0 e 1) existentes no conjunto de dados *xxx-students-2017.arff* não fosse conhecido, com o intuito de, através do desempenho do algoritmo, determinar, pelo menos, o provável número de grupos (k).

Valores de k	% de dados agrupados incorretamente
2	46,55
3	62,06
4	67,24
5	70,68
6	72,41
7	72,41
8	77,58
9	77,58
10	81,03

Tabela 2. Desempenho dos agrupamentos para diferentes valores de k.

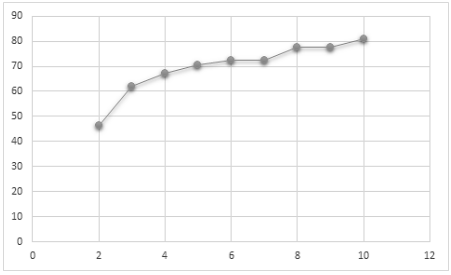

Figura 6. Gráfico da tendência de erros conforme o valor de k.

Para isso, foram utilizados diferentes valores de k (2, 3, 4, 5, 6, 7, 8, 9 e 10), onde foi possível verificar a % de erros que cada uma dessas entradas ocasionou para o algoritmo (veja a Tabela 2 e a Figura 6).

A partir desta simulação, foi possível notar uma tendência ao melhor valor para k, ou seja, k = 2, o qual gerou a menor porcentagem de objetos agrupados incorretamente (46,55%).

O terceiro experimento ocorreu a partir da implementação do algoritmo Apriori (esquema na parte inferior da Figura 5), usando como configurações básicas o suporte (*support*) com valor de 0.2, confiança (*confidence*) com valor 0.8 e com limite de 200 regras (uma escolha de limite aleatória).

Uma das regras geradas pelo questionário *xxx-students-2017.arff* foi:

q17=1 target= 0 19 ==> q10 = 1 17 <conf: (0.89)>

onde, "A" (antecedente) é q17=1 target= 0 e "B" (consequente) q10 = 1.

Essa regra nos mostra que quem iniciou os estudos no primeiro dia de aula e não evadiu do curso, implica que estudou todo o ensino médio em escola pública. A parte "A" da regra representa o suporte, que nesse exemplo tem um *support* de 32,75%, pois aparece em 19 vezes das 58 totais do questionário. A parte "B" da regra representa a confiança que tem um valor de 89%, pois das 19 vezes que aparecem juntas na parte "A" elas implicaram em 17 vezes na parte "B".

Em outra regra gerada, verifica-se qual o interesse do estudante após concluir o curso de Matemática:

q1=2 q10=1 q14=1 q16=1 q17=1 13 ==> q15=2 <conf:(0.92)>

onde, "'A"': q1=2 q10=1 q14=1 q16=1 q17=1 e "'B"': q15=2.

Essa regra nos mostra que os estudantes que frequentam o curso1 (q1), cursou todo o ensino médio em escola pública (q10), ao ingressar no curso tinha informação sobre o projeto pedagógico (q14), ingressou na universidade pelo Sisu (q16) e iniciou os estudos no primeiro dia de aula (q17), e isso implicou que o estudante tem o objetivo de ao termino do curso, realizar concurso público (parte "'B"'(q15)).

Na tentativa de obter o atributo *target* que mostra se o aluno evadiu ou não evadiu (parte "B", ou seja, no que implica), a quantidade de regras foi sendo aumentada até 1300, quando aí foi possível encontrar algumas dessas regras. Como exemplo, são quatro das regras geradas, onde na parte "B" tem-se o valor de *target*=0, que significa que o aluno não evadiu. Pode-se observar que em todos os conjuntos de regras aparece q12=1, a qual mostra por qual razão o aluno escolheu o seu curso, que nesse caso foi pela adequação à área que tem mais facilidade. Veja:

q12=1 q14=1 17 ==> target=0 14 <conf:(0.82)>
q12=1 q14=1 q19=4 14 ==> target=0 12 <conf:(0.86)>
q4=1 q12=1 14 ==> target=0 12 <conf:(0.86)>
q4=1 q12=1 q16=1 14 ==> target=0 12 conf:(0.86)

Assim, nestas quatro regras, nota-se que a facilidade com a área tem forte relação para o estudante não evadir no curso. Por vez, utilizando o conjunto de dados *xxx-students-2018.arff* e usando as mesmas configurações do algoritmo

Apriori utilizadas para o conjunto de dados *xxx-students-2017.arff*, foi possível identificar uma regra similar a do interesse do estudante após concluir o curso 1:

q1=2 q10=1 q16=1 q17=1 18 ==> q15=2 16 conf:(0.86)

Nisso, verifica-se que o grande estímulo para conclusão do curso 1, tanto para os alunos de 2017 quanto para os de 2018, é prestar um concurso público.

CONSIDERAÇÕES

Neste Trabalho foi apresentado um estudo acerca dos conceitos e técnicas da Mineração de Dados (MD), especificamente direcionados à aplicação em conjunto de dados de estudantes, denominada Mineração de Dados Educacionais (MDE), a partir do uso de três métodos de Aprendizado de Máquina (AM). Os métodos utilizados correspondem a três distintas tarefas de aprendizado (classificação, agrupamentos de dados e regras de associação). O objetivo principal foi o de mostrar como a MD e os métodos de AM podem na análise de informações de dados de estudantes. Para isso, os experimentos e análise dos resultados acorreram a partir de conjuntos de dados reais de estudantes. Como trabalhos futuros, a pesquisa deverá incrementar estudos com conjuntos de dados reais mais volumosos, incluindo cursos EAD e dados do ENADE, com pré-processamento de dados.

REFERÊNCIAS

1. J. R. Quinlan. 1993. C4.5: Programs for machine learning, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. https://dl.acm.org/citation.cfm?id=152181
2. A. K. Jain, M. N. Murty e P. J. Flynn. 1999. Data clustering: A review. ACM Comput. Surv., v. 31, n. 3, p. 264–323. http://doi.acm.org/10.1145/331499.331504
3. R. Agrawal e R. Srikant. 1994. Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., p. 487–499. https://dl.acm.org/citation.cfm?id=672836
4. J. Han e M. Kamber. 2006. Data mining: Concepts and techniques, 2ed, 500 Sansome Street, Suite 400, San Francisco, CA 94111: Morgan Kaufmann Publishers.
5. U. Fayyad, G. Piatetsky-Shapiro e P. Smyth. 1996. From data mining to knowledge discovery: An overview. In: Advances in knowledge discovery and data mining. AI Magazine, p. 1–34. https://dl.acm.org/citation.cfm?id=257942
6. E. Alpaydin. 2010. Introduction to machine learning. 2ed. The MIT Press.
7. I. H. Witten e E. Frank. 2005. Data mining: Practical machine learning tools and techniques. Morgan Kaufmann Series in Data Management Systems, 2ed. Morgan Kaufmann.
8. J. Vilar. 2017. Fundamentos de data science machine learning (Parte 1). Acesso em: ago-2017. Disponível em: https://jvilar.files.wordpress.com/2017/01/dinoml1.png
9. R. O. Duda, P. E. Hart e D. G. Stork. 2001. Pattern classification, 2ed, New York, NY, USA: Wiley-Interscience.
10. E. M. Real, M. C. Nicoletti e O. L. Oliveira. 2014. A closer look into sequential clustering algorithms and associated post-processing refinement strategies. In International Journal Innovative Computing and Applications , v.6, n.1, p.1-12. https://doi.org/10.1504/IJICA.2014.064214
11. Theodoridis, S. e Koutroumbas, K. 2009. Pattern recognition, fourth edition. Elsevier.
12. C. G. Webber, D. Zat e M. F. W. P. Lima. 2013. Utilização de algoritmos de agrupamento na mineração de dados educacionais. Anais da Revista Novas Tecnologias na Educação (RENOTE), v.11, n.1, p.1-10 http://dx.doi.org/10.22456/1679-1916.41639
13. C. Romero e S. Ventura. 2010. Educational data mining: A review of the state of the art. In: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 40, n. 6, p. 601–618. https://doi.org/10.1109/TSMCC.2010.2053532
14. V. C. G. Coelho, J. P. C. L. Costa, D. A. Silva, R. T. Sousa Junior, D. C. R. Sousa, E. D. Canedo, D. Guerreiro e Silva e R. T. Sousa Junior. 2015. Mineração de dados educacionais para identificação de barreiras na utilização da educação a distância. Anais do XXI Congresso Internacional ABED de Educação a Distância (CIAED), p.1-11. DOI: 10.17143/abed1995
15. M. Marques e M. C. Nicolleti. 2015. Mineração de dados educacionais para a predição de desempenho acadêmico em cursos universitários. Anais do XI Workshop de Computação da FACCAMP (WCF). p.1-5. Acesso em: ago-2017. Disponível em: http://www.cc.faccamp.br/anaisdowcf/edicoes_anteriores/wcf2015/
16. R. S. França e H. J. C. Amaral. 2013. Aplicação de técnicas de mineração de dados para o mapeamento do conhecimento na aprendizagem de programação: Uma estratégia baseada na taxonomia de Bloom. Anais dos Workshpos do II Congresso Brasileiro de Informática na Educação, p.122-131.
17. R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald e Scuse, D. 2012.Weka manual for version 3-6-8. Acesso em: jul-2016. Disponível em: http://www.nilc.icmc.usp.br/ elc-ebralc2012/minicursos/WekaManual-3-6-8.pdf.