



Análise de Prazos de Entrega de Atividades no Moodle: um Estudo de Caso Utilizando Mineração de Dados

Fabieli De Conti

Discente do Programa de Pós-Graduação em Informática (PPGI) - Universidade Federal de Santa Maria (UFSM) - fabieli.conti@gmail.com

Andrea Schwertner Charão

Docente do Programa de Pós-Graduação em Informática (PPGI) - Universidade Federal de Santa Maria (UFSM) - andrea@inf.ufsm.br

Resumo

Este artigo descreve um estudo realizado sobre os dados gerados na interação com o Ambiente Virtual de Aprendizagem (AVA) Moodle em uma instituição de ensino, com foco na análise de prazos e datas efetivas de submissões de tarefas neste ambiente. O principal objetivo do trabalho é obter informações relevantes sobre a postagem de tarefas no ambiente, para subsidiar ações que possam auxiliar a reduzir o envio de trabalhos após o prazo estipulado ou muito próximo ao final do período de postagem. Para isso, são considerados o período em que a tarefa permaneceu aberta para postagem, o curso proveniente da tarefa e o período em que a postagem foi realizada. Esse estudo foi realizado seguindo as etapas do processo de descoberta de informação, com a utilização de algoritmos de mineração de dados da ferramenta Weka.

Palavras-chave: ambiente virtual de aprendizagem; moodle; kdd; mineração de dados

Delivery Time Analysis of Activities in Moodle: a Case Study using Data Mining

Abstract

This paper describes a study that was carried out on data generated through the interaction with the Virtual Learning Environment (VLE) Moodle in an educational institution, focusing on the analysis of deadlines and effective dates of assignment submissions in this environment. The main goal of this study is to obtain relevant information about assignment submissions, in order to support actions that would help to reduce submissions after or very near the deadline. The study considers the time the assignment remained open for submission, the course the task is intended for, and the date when the content was submitted. This study was carried out following the process of knowledge discovery in databases, using data mining algorithms from the Weka tool.

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



Introdução

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

O Modular Object-Oriented Dynamic Learning Environment (Moodle) é um AVA desenvolvido com base na política *Open Source*. Sua concepção foi iniciada nos anos 90, por Martin Dougiamas, com base na filosofia de aprendizagem do construcionismo social. O objetivo desse projeto era de suportar a criação e administração de cursos com enfoque no trabalho colaborativo em um ambiente de simples e intuitiva utilização.

Nos diferentes segmentos da sociedade, as instituições têm buscado na tecnologia recursos que agreguem valor aos seus negócios, seja agilizando operações, suportando ambientes ou viabilizando inovações (Silva, 2004). A inclusão desses sistemas de informação no cotidiano da sociedade gera um grande volume de dados provenientes das atividades rotineiras. Esses dados podem ser transformados em informações importantes para sucesso da organização, com a aplicação do processo de Descoberta de Conhecimento em Banco de Dados (KDD). Tal processo consiste na extração de conhecimento de alto nível a partir de dados de baixo nível disponíveis em grandes bancos de dados (Fayyad et al., 1996b).

A mineração de dados é uma etapa do processo de descoberta de conhecimento em banco de dados, da qual também fazem parte o pré-processamento e o pós-processamento (Boente, 2006). Técnicas de mineração de dados vêm sendo aplicadas para análise do grande volume de dados gerados em AVAs, com resultados promissores.

No presente trabalho, apresentamos um estudo de caso cujo foco é a análise de prazos de entrega de atividades no AVA Moodle, identificando padrões na postagem de atividades, contrapondo com o nível do curso a que tarefa pertencia e o período em que a tarefa permaneceu aberta para postagem.

Para a análise de prazos de entrega de atividades no AVA Moodle, extraímos um conjunto de dados provenientes do Instituto Federal Farroupilha - Campus de São Vicente do Sul. Na mineração dos dados, utilizamos a ferramenta Weka com o intuito de descobrir padrões que permitissem classificar as ocorrências em grupos. Os resultados permitiram interpretações relevantes para o caso em questão, confirmando a utilidade da abordagem. No restante deste artigo, apresenta-se o processo de descoberta de conhecimento, a descrição do estudo de caso e as considerações finais.

1. Descoberta de conhecimento

Segundo Fayyad et al. (1996a), descoberta de conhecimento é “o processo, não trivial, de extração de informações implícitas, previamente desconhecidas e potencialmente úteis, a partir dos dados armazenados em um banco de dados”. Isto relaciona-se com as áreas de banco de dados, estatística e inteligência artificial, que juntas buscam extrair padrões, modelos em bases de dados que não são perceptíveis em uma análise simples.

Esse processo envolve as seguintes etapas:

1. Definir o tipo de conhecimento a descobrir: compreender o domínio da aplicação, e o tipo de decisão que tal conhecimento pode contribuir para melhorar.
2. Criar um conjunto de dados alvo (*Selection*): selecionar um conjunto de dados, ou focar num subconjunto, onde a descoberta deve ser realizada.
3. Limpar dados e pré-processar (*Preprocessing*): modelar ou estimar ruído, remover ruídos quando necessário, escolher estratégias para manipular campos de dados ausentes, formatar dados para a adequá-los à ferramenta de mineração.
4. Reduzir dados e projeção (*Transformation*): localizar características úteis para representar os dados dependendo do objetivo da tarefa, para reduzir o número de variáveis e/ou instâncias, e enriquecer a semântica das informações.
5. Minerar dados (*Data Mining*): selecionar os métodos a serem utilizados para localizar padrões nos dados, buscar por padrões de interesse, e buscar o melhor ajuste dos parâmetros para a tarefa.
6. Interpretar padrões minerados (*Interpretation/Evaluation*), com um possível retorno aos passos 1-6 para posterior iteração.
7. Implantar conhecimento descoberto (*Knowledge*): incorporar este conhecimento à performance do sistema, ou documentá-lo e reportá-lo às partes interessadas.

Somente é possível obter padrões relevantes se os dados são selecionados corretamente e transformados para formatos adequados de acordo com os algoritmos aplicados. Observando a Figura 1, conseguimos vislumbrar o encadeamento das atividades relacionadas com o pré-processamento para o sucesso do processo de KDD.

Os algoritmos devem ser escolhidos de acordo com os dados a serem trabalhados e o tipo de padrão desejado; já o conhecimento extraído na etapa de mineração deve ser avaliado e validado para então ser aplicado.

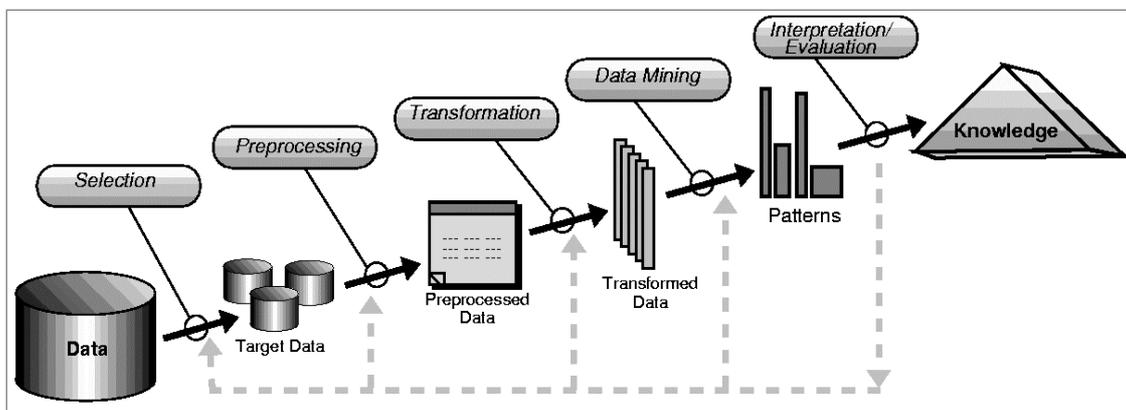


Figura 1. Etapas de Descoberta de Conhecimento em Banco de Dados (FAYYAD et al. 1996a)

2.1 Mineração de dados

Mineração de dados é definida como sendo “o uso de técnicas automáticas de exploração de grandes quantidades de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertos a olho nu pelo ser humano” (Carvalho, 2001).

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

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

Nos tópicos seguintes abordaremos as tarefas do processo de mineração de dados:

- **Associação:** Essa tarefa consiste em descobrir atributos que ocorrem simultaneamente com grande frequência. Para analisar a qualidade das regras de associação, são utilizados parâmetros de confiança, obtidos da quantidade de vezes em que um item A e um item B aparecem juntos em uma mesma transação. Estes são relacionados com o total de vezes que o item A aparece no total das transações. O algoritmo Apriori é um exemplo baseado na associação.
- **Clusterização ou Agregação:** Com a aplicação dessa tarefa, os elementos com características semelhantes são agrupados em um mesmo cluster. Cada cluster apresenta internamente grande similaridade e grande diferença entre os outros cluster formados pelos conjuntos de dados. São exemplos dessa tarefa os algoritmos: K-Means, K-Modes, K-Prototypes, K-Medoids e EM.
- **Classificação:** Busca a descoberta de funções que mapeiem registros em classes pré-definidas. Essas funções após descobertas são usadas como sistema de apoio à decisão para prever a classe em que determinado conjunto de registros se enquadram. Os algoritmos dessa tarefa se utilizam de Redes Neurais, Algoritmos Genéticos e Lógica Indutiva.

2.2 Ferramenta Weka

Existem várias ferramentas que auxiliam na mineração de dados. Neste trabalho, utilizamos a ferramenta Weka (Weka, 2011), implementada pelos pesquisadores da Universidade de Waikato de Nova Zelândia, desenvolvida com linguagem de programação Java seguindo a abordagem de *framework*. Aliadas à característica de ter o seu código fonte aberto, essas características facilitam a adaptação, a inclusão de novas funcionalidades em algoritmos e a portabilidade entre diferentes sistemas operacionais.

A ferramenta possibilita o acesso aos dados, diretamente de bancos de dados via JDBC, através de URL, no formato csv ou no formato próprio, chamado Attribute-Relation File Format (ARFF). Nesse trabalho utiliza-se o modo explorer do Weka, que apresenta as seguintes interfaces: preprocess, classify, cluster, associate, select attribute e visualize.

2. Trabalhados relacionados

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

- atributos pessoais dos alunos, do curso, turma e disciplina;
- o tempo para que cada aluno demorou para se inscrever na turma;
- se o aluno realizou atividades em equipe;

- se o aluno utilizou agenda e anotações do sistema;
- o total de problemas resolvidos e dos pontos obtidos na realização dos mesmos;
- o nível de dificuldade dos exercícios resolvidos; e,
- a quantidade de acessos ao ambiente.

Baruque et al. (2008) analisaram as ferramentas do Moodle (chat, lição quiz, fórum, wiki, etc) que são preferidos pelos alunos de diferentes cursos, a fim de disponibilizar os recursos conforme os perfis. Este trabalho utilizou da ferramenta de mineração de dados *Magnum Opus*, que se baseia nas regras de associação para a descoberta de conhecimento.

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

Dentre os trabalhos relacionados, não encontramos análises com foco em prazos de entrega de atividades. No entanto, há indícios de que este tipo de análise é importante para apoiar decisões e formulação de estratégias para um adequado gerenciamento do tempo dedicado a atividades de uma ou mais disciplinas (Pilling-Cormick, 2002).

3. Estudo de Caso

Na realização deste trabalho foram trabalhados com a base de dados real, gerados pelo AVA Moodle versão 1.9.19, do Instituto Federal Farroupilha - Campus São Vicente do Sul. O ambiente contém 55 cursos cadastrados, sendo 15 de nível técnico, 21 de nível superior, 5 voltados para formação continuada dos profissionais de educação e 14 para curso no nível de pós-graduação. Dos 55 cursos, somente 23 utilizaram o recurso de tarefa em sua realização, onde 276 usuários interagiram em 63 atividades, totalizando 677 registros.

4.1 Seleção dos Dados

A estrutura da base de dados do Moodle é composta por 203 tabelas em SQL, que armazenam todas as informações sobre os cursos, usuários cadastrados, acessos, atividades e avaliações. O conjunto de dados trabalhados no estudo é proveniente das seguintes tabelas: *mdl_assignment*, que armazena informações sobre cada tarefa, e *mdl_assignment_submissions*, que armazena informações sobre cada tarefa submetida.

Para a seleção dos dados, foram realizadas consultas no banco de dados utilizando a linguagem SQL. Da tabela *Assignment* (atividade) selecionamos: identificação da atividade; curso; prazo final para a postagem; data de abertura da atividade; identificação do usuário. Já da tabela *Assignment Submissions* (atividade submetida) buscamos a data da postagem e o usuário. Para gerar o arquivo ARFF, o conjunto dos dados selecionados passou pelas etapas dos tópicos seguintes.

4.2 Limpeza e transformação do dados

Nessa etapa removemos os atributos que não agregam informações para a análise dos dados, restando o conjunto de dados formado pelos atributos, identificação do nível do curso, data de início, data de término para postagem e a data em que foi efetivada a postagem. Com esses dados, geramos o arquivo ARFF. Depois de alguns testes observamos a dificuldade de interpretar os dados expressos na forma de datas, por isso resolvemos adotar outros parâmetros para esses atributos. Para tal, criamos os seguintes atributos:

- Período de postagem: que é calculado diminuindo a data inicial da postagem data de final;
- Entrega realizada: data final da postagem menos a data em que foi realizada a postagem;
- Porcentagem Diferença entre postagem: porcentagem do período de postagem menos período da entrega realizada.

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

A fim de facilitar a legibilidade dos resultados, transformamos o período expresso em Timestamp, para o formato dias e horas, utilizando a seguinte fórmula: data do período dividida por 3600, o resultado inteiro representa as horas e o que segue após a vírgula é multiplicado por 60 e dividido por 100, representando os minutos. Juntando os dois números temos o período expresso em dias e horas.

Também acrescentamos um novo atributo que armazena a situação da postagem, a qual pode ser classificada em 6 classes:

- Classe 1: quando postado logo na abertura da atividade, ou seja, a postagem ocorreu entre o intervalo dos 25% iniciais do tempo total para a postagem;
- Classe 2: quando a postagem ocorreu entre o intervalo superior ou igual a 25% do tempo total da postagem e inferior a 50% para a postagem;
- Classe 3: quando postado entre o intervalo superior ou igual a 50% do tempo total da postagem e inferior ou igual 75% do tempo;
- Classe 4: quando postado entre o intervalo superior ou igual a 75% do tempo total da postagem e inferior ou igual 95% do tempo;
- Classe 5: quando postado no intervalo de tempo superior a 95% do período em que a atividade está aberta para postagem; e,
- Classe 6: quando postado após o término do período de postagem da atividade. Para calcular a porcentagem dos intervalos de postagem utilizamos a seguinte fórmula: período de postagem multiplicado por 100, dividido pelo período total de postagem.

Na tabela 1 podemos visualizar a relação das situações de postagem com a quantidade de ocorrência das mesmas.

Tabela 1. Relação das classes e a quantidade de ocorrência de cada uma.

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

Com o intuito de obter uma identificação mais simplificada do tempo em que as tarefas ficaram abertas para a postagem, realizamos a classificação das mesmas, em 5 classes. Para tal divisão nos baseamos na quantidade de horas que permaneceram abertas.

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

A distribuição nas diferentes Classes de períodos de entrega relacionadas com o Nível de Ensino, podem ser observadas no gráfico apresentado na figura 2.

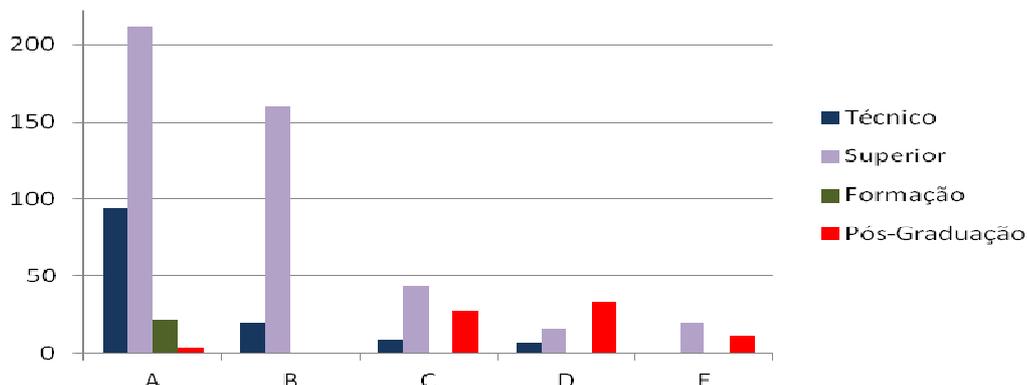


Figura 2. Atividade classificadas por tempo de postagem e Nível de Ensino

4.3 Mineração

A primeira atividade realizada foi de agrupamento, na qual buscamos encontrar grupos que tenham semelhança entre eles, relacionando os atributos período de entrega, porcentagem diferença entre postagem e situação. Para tal, executamos os algoritmos: SimpleKmeans, Cobweb, DBScan e o Expectativa (EM), sendo que esse último apresentou padrões mais adequados às regras que estávamos buscando.

Outro questionamento que tínhamos era quanto a relação do tempo para a postagem e o período na qual ocorreu a mesma, relacionando com os níveis de ensino. Para encontrar essas regras utilizamos os algoritmos baseados na atividade de classificação, na qual é possível com base em um subconjunto de dados determinar o valor de um atributo.

Executamos os algoritmos: Id3, J48, ADTree, UserClassifier, PredictionNode, ClassifierTree, Prism, Part, Naive Bayes. O algoritmo selecionado foi o J48, por apresentar resultados de classificação com maior margem de acertos.

4.4 Resultados

Executamos o algoritmo de agrupamento EM, que busca identificar semelhança máxima entre parâmetros de modelos estatísticos, e obtivemos a divisão dos dados em cinco clusters, apresentados na figura 3.

Attribute	Cluster				
	0 (0.11)	1 (0.04)	2 (0.29)	3 (0.45)	4 (0.11)

classe_periodo_entrega					
A	1.5033	5.0634	155.3759	101.1946	71.8627
B	1.0616	5.0243	10.4837	165.0472	1.3832
C	29.373	1.3776	29.3687	24.3782	1.5025
D	35.5449	1.4953	3.3756	15.9168	5.6674
E	11.4712	18.9244	1.3718	3.1312	1.1015
[total]	78.954	31.8851	199.9756	309.668	81.5173
situação					
Classe_1	25.0321	5.5348	61.6085	226.4639	4.3606
Classe_2	13.5541	1.4807	62.0934	32.0077	1.8641
Classe_3	8.2855	1.4333	41.4448	8.5726	18.2638
Classe_4	25.9326	1.4574	10.9258	12.5374	2.1467
Classe_5	1.904	20.983	3.3362	4.6311	44.1457
Classe_6	5.2457	1.9957	21.567	26.4553	11.7363
[total]	79.954	32.8851	200.9756	310.668	82.5173
nivel_curso					
Superior	3.3218	26.5467	149.6412	273.1865	5.3038
Pós	70.772	1.7896	2.4005	1.3667	2.6712
Técnico	2.8586	1.3221	45.8417	33.0845	50.8931
Formação	1.0017	1.2266	1.0922	1.0304	21.6491
[total]	77.954	30.8851	198.9756	308.668	80.5173
Clustered Instances					
0	71 (10%)				
1	25 (4%)				
2	176 (26%)				
3	312 (46%)				
4	93 (14%)				

Figura 3. Agrupamento utilizando o algoritmo EM.

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

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

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

No cluster 3: Nesse grupo também temos a maior incidência de cursos do nível e técnico e mais da metade de todas as atividades do nível superior estão neste agrupamento, com o prazo de postagem concentrado em até 15 dias, sendo que as atividades foram postadas com maior incidência nos 25% iniciais do período para a postagem.

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

Nas figuras 4 e 5, podemos observar os resultados obtidos com a execução do algoritmo J48. Na figura 4 é apresentada a árvore de classificação obtida pela execução do algoritmo, onde verificamos que os cursos de nível superior têm suas atividades criadas com o tempo para postagem em sua grande maioria de até 30 dias e a postagem ocorre geralmente nos primeiros 25% do período total para postagem.

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

Figura 4. Classificação Algoritmo J48

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

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

=== Detailed Accuracy By Class ===

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

=== Confusion Matrix ===
  a  b  c  d  <-- classified as
420  2  31  0 |  a = Superior
 45  27  2  0 |  b = Pós
 77  1  51  0 |  c = Técnico
  2  0  19  0 |  d = Formação
```

Figura 5. Classificação Algoritmo J48

5. Considerações finais

Após o estudo de caso realizado, reforçou-se o entendimento de que o processo de KDD é uma atividade que exige domínio da área de negócio e conhecimento das técnicas e etapas, e como resultado desse processo podemos visualizar de maneira mais clara os relacionamentos entre os dados gerados pelas atividades operacionais.

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

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

Com o estudo realizado, concluímos que nesse contexto é mais viável a realização de atividades com um prazo menor, pois além do índice de postagem ser maior logo na abertura para postagem o professor consegue um *feedback* mais rápido do processo de aprendizagem do aluno, podendo esse tomar atitudes corretivas afim de evitar o insucesso ou desistência do aluno.

Pretendemos continuar realizando estudos com o conjunto de dados e também realizar a implementação de um aplicativo de mineração de dados do Moodle, com uma interface mais simplificada para facilitar a utilização por pessoas que não tenham domínio das ferramentas de mineração.

Referências

- BARUQUE, C. B. et al, 2007. Analysing users' access logs in Moodle to improve e learning. Euro American Conference on Telematics and information Systems. Faro, Portugal.
- BOENTE, A.N.P. Descoberta de Conhecimento em Bases de Dados. Iowa, Tese de Doutorado – Departamento de Informática, AWU – American World University, 2006.
- CARVALHO, L. A. V., 2001. Datamining: A Mineiração de Dados no Marketing, Medicina, Economia, Engenharia e Administração. Editora Érica Ltda, São Paulo, Brasil.
- DIAS, M. M. et al, 2008. Aplicação de Técnicas de Mineração de dados no Processo de Aprendizagem na Educação a Distância. Anais do XIX Simpósio Brasileiro de Informática na Educação. SBC, Fortaleza, Brasil.
- FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., 1996a. From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining*, AAAI Press/The MIT Press. England.
- _____. Knowledge Discovery and Data Mining: Towards a Unifying Framework. Second International Conference on KD & DM. Portland, Oregon, 1996b.
- HARRISON, T.H., 1998. Intranet data warehouse. Editora Berkeley, São Paulo, Brasil.
- PILLING-CORMICK, J. Transformative and Self-Directed Learning in Practice. *New Directions for Adult and Continuing Education*, v. 1997, n. 74, p. 69–77, 1997.
- ROMERO, C., VENTURA, S., GARCÍA, E., 2007. Data mining in course management systems: Moodle case study and tutorial. Elsevier Science. Oxford, Inglaterra.
- SILVA, Marcelino Pereira dos Santos. Mineração de Dados - Conceitos, Aplicações e Experimentos com Weka. IV Escola Regional de Informática RJ/ES (IV ERI RJ/ES) Nov 2004. Sociedade Brasileira de Computação – SBC. Disponível em: www.sbc.org.br/bibliotecadigital/download.php?paper=35. Acessado: 20 set. 2011.
- WEKA, 2011. *Data Mining Software in Java*. Disponível em <http://www.cs.waikato.ac.nz/ml/weka>.