

## Detecção automática de Estilos de Aprendizagem por meio de técnicas de clusterização e classificação

Rafael Miranda Abreu<sup>1</sup>, Cristiano Grijó Pitangui<sup>2</sup>, Alessandro Vivas Andrade<sup>1</sup>,  
Luciana Pereira de Assis<sup>1</sup>, Cristiano Maciel da Silva<sup>2</sup>

<sup>1</sup>Universidade Federal dos Vales do Jequitinhonha e Mucuri

<sup>2</sup>Universidade Federal de São João del-Rei

{rafael.abreu, alessandro.vivas, lpassis}@ufvjm.edu.br

{pitangui.cristiano, cristiano}@ufsj.edu.br

**Abstract.** *In recent years Learning Management Systems have been widely used to support Education. However, most of these systems provide the same content in the same way and formats to all students, which can be detrimental to the teaching/learning process. In this sense, this work, based on Felder-Silverman Learning Style model, presents an automatic approach to detect the students' Learning Styles so that the content offered is modeled according to their preferences. The proposed model, based on clustering and classification techniques, achieved an accuracy rate up to 90% proving to be a promising approach to automatically detect the students' learning preferences.*

**Resumo.** *Recentemente, os Sistemas de Gerenciamento de Aprendizagem têm sido muito utilizados no apoio à Educação. No entanto, a maioria destes sistemas fornece o mesmo conteúdo de iguais maneiras e formatos a todos os alunos, o que não favorece o processo de ensino-aprendizagem. Neste sentido, este trabalho, baseando-se no modelo de Estilos de Aprendizagem de Felder e Silverman, apresenta uma abordagem automática para a detecção de Estilos de Aprendizagem de estudantes para que o conteúdo ofertado seja modelado de acordo com as preferências dos mesmos. O modelo proposto, fundamentado em técnicas de clusterização e classificação, mostrou-se promissor, alcançando acurácia de detecção de até 90%.*

### 1. Introdução

Modalidade de ensino em constante crescimento, a Educação a Distância (EAD), além de permitir uma maior interação dos envolvidos, torna acessíveis seus conteúdos a todos usuários, mesmo quando estes estão distantes geograficamente. Entretanto, [Daniel et al. 2015], apontam desvantagens quanto à transmissão do ensino, como, por exemplo, a oferta de conteúdos idênticos aos estudantes, desconsiderando assim seus perfis diferenciados. Neste sentido, é desejável que Instituições Educacionais personalizem seus processos de acordo com as características de seus alunos [Kurilovas et al. 2014].

Tecnologias da Informação e Comunicação (TICs), auxiliam de diversas maneiras o processo de ensino-aprendizagem na EAD. Prova disso é a melhoria na comunicação por meio do uso de Ambientes Virtuais de Aprendizagem (AVAs). [Ribeiro et al. 2013]

descrevem esses ambientes como sistemas computacionais, que integram mídias, ordenam informações e desenvolvem interações entre pessoas e objetos do conhecimento.

Os AVAs utilizam-se de Objetos de Aprendizagem (OA), que de acordo com [Zaina et al. 2012], são entidades a serem utilizadas no processo de aprendizagem. São exemplos de OA: vídeos, imagens, textos, dentre outros. [Carvalho et al. 2017] definem OA como recursos de aprendizagem que têm como objetivo carregar informação a ser assimilada pelo aprendiz.

Grande parte das Instituições Educacionais utilizam AVAs que ofertam o conteúdo de igual maneira aos aprendizes, ignorando suas preferências de ensino, motivo pelo qual é crescente o avanço de estudos sobre recomendação personalizada de conteúdos [Daniel et al. 2015].

De acordo com [Graf et al. 2008], para associar o conteúdo mais adequado a um aluno, primeiramente o seu Estilo de Aprendizagem (EA) deve ser conhecido. Para [Borges and Stiubiener 2014], EA são as preferências do indivíduo em capturar, organizar e transformar a informação em conhecimento, de maneira a facilitar a compreensão do mesmo.

Existem basicamente duas formas de se identificar o EA de um aprendiz, a saber: de maneira colaborativa ou automática [Khribi et al. 2013]. A primeira necessita que o indivíduo forneça informações explicitamente, geralmente por preenchimento de questionários. Na segunda, as informações são obtidas automaticamente por meio do comportamento *online* do usuário.

Neste contexto, [Graf et al. 2008] ressaltam que por razões emocionais ou até mesmo por falta de compreensão, o questionário pode ser respondido equivocadamente, o que o torna um instrumento de menor confiabilidade. Por sua vez, de acordo com [Khribi et al. 2013], a abordagem automática é mais precisa e menos propensa a erros, já que analisa dados comportamentais reais.

Neste sentido, e com base no Modelo de Estilos de Aprendizagem de [Felder and Silverman 1988], este trabalho apresenta uma abordagem de detecção totalmente automática de EA de estudantes, utilizando-se para isso, técnicas de Mineração de Dados (MD).

Devido à baixa representação de alguns EA em bases de dados reais, desenvolveu-se um simulador de alunos que emula o comportamento dos mesmos em AVAs de acordo com seus EA.

Resultados obtidos apontam que a abordagem proposta é promissora na identificação automática de EA, uma vez que os algoritmos de clusterização e de classificação utilizados obtiveram acurácia de detecção de até 90%.

O restante do trabalho se organiza como segue. As seção 2 apresenta o referencial teórico e os principais trabalhos relacionados a esta pesquisa. A seção 3 apresenta a abordagem proposta para a detecção automática de EA. A metodologia e os resultados experimentais são apresentados na seção 4. Por fim, a seção 5 conclui o trabalho e aponta perspectivas de trabalhos futuros.

## 2. Referencial Teórico

Para um melhor entendimento do presente trabalho, esta seção apresenta conceitos e definições sobre temas que são de abordagem do presente estudo. Adicionalmente, são apresentados os principais trabalhos relacionados a esta pesquisa.

### 2.1. Estilos de Aprendizagem e o Modelo FSLSM

Como citado, [Borges and Stiubiener 2014] referem-se aos EA como sendo as preferências dos indivíduos em capturar, organizar e transformar informações em conhecimento de maneira a facilitar a compreensão do mesmo. O campo de EA é complexo e afetado por vários aspectos, levando a diferentes conceitos e visões. Existem muitos modelos de EA na literatura, cada um com suas próprias características, e propondo diferentes descrições e classificações de tipos de aprendizagem [Graf and Kinshuk 2009]. De forma geral, o modelo de Felder e Silverman (FSLSM - *Felder and Silverman Learning Style Model*), é o mais utilizado na área Educacional [Zaina et al. 2012].

A visão favorável sobre a utilização dos EAs no processo de ensino-aprendizagem não é unânime. Existem trabalhos críticos aos EAs, a exemplo [Kirschner and van Merriënboer 2013] afirmam que o EA é estático, o que faz com que o aluno seja classificado em um grupo específico. Por sua vez, [Rawson et al. 2017] evidenciam a baixa credibilidade dos métodos utilizados para detecção de EA. Entretanto, diversos outros estudos como [Bernard et al. 2017], [Carvalho et al. 2017] e [Kolekar et al. 2017] defendem a utilização de EAs e apresentam diferentes técnicas para sua identificação.

No FSLSM, as maneiras de como as informações são recebidas e processadas estão divididas em quatro dimensões. A dimensão de **processamento** é dividida em alunos reflexivos e ativos e procura apontar a melhor forma de processamento da informação pelo estudante. Alunos visuais e verbais pertencem à dimensão de **entrada**, que preocupa com o formato em que os OA são apresentados durante uma sessão de aprendizagem. Outra dimensão é a de **percepção**, que observa as preferências do aluno por algum tipo de OA, e é subdividida nas categorias sensitivo e intuitivo. Por fim, a dimensão de **organização**, na qual o estudante classifica-se como sequencial ou global, trata do ponto de vista de como o conteúdo é acessado.

A partir do FSLSM, [Felder and Soloman 1991], criaram o *Index of Learning Styles (ILS)*, um questionário para identificar o EA que contempla as dimensões contidas no modelo de *Felder e Silverman*.

### 2.2. Detecção de Estilos de Aprendizagem

Como apontado, estudos revelam que alunos podem assimilar mais facilmente os conteúdos apresentados, caso estes estejam de acordo com suas características individuais, e que uma das formas de realizar esta personalização é por meio da detecção do EA do aprendiz. Neste sentido, várias abordagens foram propostas.

A fim de melhorar a precisão de abordagens automáticas para detecção de EA, [Bernard et al. 2017] apresentam quatro técnicas, cada uma baseada em um algoritmo, a saber: Rede Neural, Algoritmo Genético, Colônias de Formigas, e Otimização por Enxame de Partículas. Testes foram realizados com dados extraídos do *Moodle* e os resulta-

dos obtidos mostraram-se promissores. No entanto, mesmo sendo abordagens automáticas, o questionário ILS foi aplicado para coletar dados e validar resultados.

A partir de dados capturados por meio dos *logs* em um portal *e-learning*, [Kolekar et al. 2017] propuseram uma abordagem que utiliza o algoritmo de clusterização (*Fuzzy C Means*) (FCM) para agrupar os dados comportamentais capturados nas categorias FSLSM, e posteriormente o algoritmo *Gravitational Search based Back Propagation Neural Network* prediz o EA do aluno.

[Pitigala Liyanage et al. 2016] extraíram e pré-processaram dados de um AVA *Moodle*, para então, por meio de técnicas de MD, preverem o EA do aluno considerando o modelo de FSLSM. Os dados foram trabalhados no *framework* WEKA, que implementa os algoritmos utilizados: *J48*, *Bayesian Network*, *Naive Bayes* e *Random Forest*. O questionário ILS foi aplicado para efeitos de validação dos resultados.

Segundo [Khribi et al. 2013], existem basicamente duas maneiras de descobrir o EA de um aluno. A primeira é de forma colaborativa, que requer o fornecimento explícito de informações por parte do usuário. Na segunda, as informações do usuário são coletadas de forma automática de acordo com seu comportamento e atividades *online* em um AVA.

Conforme trabalhos apresentados, a maioria das pesquisas sobre descoberta de EA utiliza a abordagem colaborativa, a qual necessita de informações explicitamente fornecidas por parte do aluno. Como já apontado, métodos que optam por esse tipo de abordagem estão mais sujeitos a falhas, o que torna as abordagens automáticas mais confiáveis.

Apesar do exposto, a maioria das abordagens automáticas existentes, em algum momento, utilizam-se de dados oriundos de questionários, seja para alimentar e treinar seus algoritmos, ou para validar seus resultados. Possivelmente, devido a dificuldade em realizar essa validação sem uso de questionários, pesquisas de abordagens totalmente automáticas para detecção de EA ainda são escassas, o que torna a presente proposta diferenciada.

### 3. Abordagem Proposta

A fim de detectar o EA de estudantes, sem que estes forneçam qualquer informação (mesmo para efeitos de validação), elaborou-se a abordagem apresentada na Figura 1.

Baseando-se no modelo FSLSM, e no trabalho [Graf and Kinshuk 2009], inicialmente realiza-se a categorização das ferramentas e OA do *Moodle*, como por exemplo: alunos ativos preferem postar em fóruns com mais frequência, aprendizes verbais visitam material de leitura com maior frequência, e assim por diante. Em seguida, realiza-se a extração de dados dos alunos que já estão registrados na base de dados do *Moodle* por meio de *logs*.

De posse das informações obtidas na categorização das ferramentas e OA, e dos dados pré-processados extraídos do *Moodle*, realiza-se uma clusterização, onde os alunos são agrupados de acordo com seu comportamento *online*: alunos que se comportam de maneira semelhante serão agrupados em um mesmo *cluster*.

Optou-se pela clusterização em 16 grupos pelo FSLSM possuir 8 categorias, sendo 2 em cada dimensão. Dessa forma, o aluno tem a possibilidade de se enquadrar em uma das 16 possíveis sequências de EA. Após a clusterização, o próximo passo é a utilização

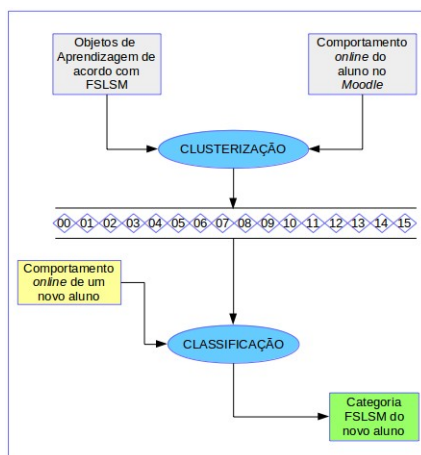


Figura 1. Abordagem Proposta - Fonte: Próprio Autor.

de classificadores para prever o EA de novos alunos. Neste sentido, as regras geradas pelos classificadores associarão o novo estudante a um dos 16 *clusters*.

É importante notar que, atualmente, bases de dados reais dificilmente contemplam as 16 possíveis sequências de EA de acordo com o FSLSM, uma vez que isto exigiria a utilização dos mais variados e diversos OA presentes no *Moodle*. Por este motivo, este trabalho propõe um simulador de alunos, que possibilita a "geração" de alunos pertencentes a qualquer EA no FSLSM.

### 3.1. Simulador de Alunos

A ferramenta implementada tem a função de gerar bases de dados artificiais que simulam bases de dados reais. Seu desenvolvimento baseou-se nos estudos realizados por [Graf and Kinshuk 2009], que definiram padrões relevantes de comportamento do estudante para cada categoria de cada dimensão do FSLSM.

Cada aluno gerado pelo simulador é representado por valores percentuais que retratam interações com os diversos OA disponíveis no *Moodle*, tais como: número de visitas e postagens em fóruns, tempo utilizado na leitura de resumos, número de revisões nos testes, desempenho em questões específicas e gerais, número de OA ignorados, dentre outros. Foram considerados 32 atributos, e os valores a eles atribuídos, que representam interações com o AVA, estão em conformidade com os limites especificados em [Graf and Kinshuk 2009].

É relevante o fato de que alunos de cursos EAD não interagem apenas com atributos relacionados ao seu EA. Isso deve-se a diversos fatores, tais como, a obrigatoriedade no uso de certas ferramentas, ou até mesmo o oferecimento de um conteúdo utilizando apenas um OA.

Por este motivo, foram adicionados ruídos à geração das interações dos alunos. Assim, ao se gerar uma determinada interação de um aluno de um determinado EA, algumas de suas interações são geradas referentes a outro EA. Dessa forma, atualmente o simulador permite que sejam geradas bases de dados com ruídos de 12,5%, 25% e 37,5%.

### 3.2. Clusterização e Classificação

Utilizou-se o *framework* WEKA para realização das etapas de clusterização e classificação. O WEKA é uma ferramenta desenvolvida em Java que reúne um conjunto de algoritmos de Aprendizado de Máquina para tarefas de Mineração de Dados [Frank et al. 2016].

Para a clusterização, foram testados quatro algoritmos: *K-Means*, *Hierarchical Clusterer*, *FarthestFirst* e *Canopy*. Após a realização dos testes, constatou-se que o algoritmo que saiu-se melhor, i.e., apresentou a menor quantidade de erros, foi o *FarthestFirst*.

Durante a etapa de classificação, optou-se pela utilização de três algoritmos, a saber, o *jRip*, o *J48* e o *RandomForest*. Optou-se por estes algoritmos uma vez que seus modelos de classificação são constituídos por regras inteligíveis, i.e., regras que podem ser facilmente entendidas e interpretadas.

## 4. Resultados

Esta seção apresenta a metodologia utilizada na pesquisa, bem como apresenta e analisa os resultados obtidos através da aplicação da abordagem proposta.

### 4.1. Metodologia Experimental

A primeira etapa de análise utilizou o simulador de alunos desenvolvido para geração de três bases de dados de interações de alunos com 800 registros em cada base, sendo a primeira base com 12,5% de ruído, a segunda com 25% e uma última com 37,5%. Cada base de dados é composta por 50 interações de estudantes para cada uma das 16 classes de EA de acordo com o FSLSM.

Estes percentuais de ruído foram definidos de acordo com o número de atributos alterados em cada dimensão do FSLSM, ou seja, quando o ruído é de 12,5%, é alterado 1 atributo em cada dimensão, o que totaliza 4 atributos modificados. De forma análoga, quando o ruído é de 25%, são alterados 2 atributos em cada dimensão, o que totaliza 8 atributos modificados. Finalmente, para as bases com ruído de 37%, são modificados 12 atributos, 3 em cada dimensão, dos 32 considerados.

Para tornar os testes ainda mais diversos, foi gerada uma quarta base de dados de interações de estudantes com os mesmos 800 registros e com ruído de 25%, onde foram alterados 8 atributos. No entanto, em vez de serem alterados 2 atributos em cada dimensão, como a base anterior de 25% de ruído, a escolha dos atributos modificados foi realizada de maneira aleatória.

Para a realização dos testes, utilizou-se a metodologia *10-fold cross validation*, i.e., as bases de dados foram subdivididas em 10 *folds*, onde cada *fold* possui um conjunto de treinamento com 720 registros, e um conjunto de teste com 80 registros (90% e 10%, respectivamente). Os algoritmos foram parametrizados utilizando os valores *default* sugeridos no *framework* Weka. A etapa seguinte constituiu-se na clusterização das bases de treinamento e aplicação dos classificadores.

Cada *cluster* é associado a determinada classe, no entanto, não é informado a qual dos 16 EA esta classe está ligada, sendo assim, antes de aplicar a classificação, criou-se um algoritmo com o intuito de identificar qual EA cada classe esta relacionada. Para realizar esta identificação, o algoritmo utiliza como parâmetros os centroides gerados pelo clusterizador.

A ideia geral deste procedimento é utilizar como parâmetro de entrada os valores de centroides, e após a realização de cálculos de distância entre os centroides e os valores de interação dos alunos, o algoritmo retorna a qual das 16 sequências de EA cada *cluster* está associado. Para desenvolvimento deste algoritmo, identificou-se quais atributos eram relevantes para cada dimensão e cada categoria do FSLSM. Os valores dos limites de cada atributo utilizados na implementação são descritos em [Graf and Kinshuk 2009].

Por fim, após a geração dos resultados, utilizou-se o Teste t com p-valor de 5% para fins de comparação das acurácias obtidas pelos modelos.

## 4.2. Resultados Experimentais

A Tabela 1 apresenta a performance do *FarthestFirst* na fase de clusterização, no que tange o número de erros, a acurácia, e classes não identificadas (*no class*), em cada conjunto de treinamento de cada *fold*, nas categorias de ruídos consideradas.

**Tabela 1. Resultados da clusterização pelo *FarthestFirst*.**

FOLDS	RUIÍDO 12,5%			RUIÍDO 25%			RUIÍDO 37,5%			RUIÍDO ALEATÓRIO DE 25% <sup>1</sup>		
	ERROS	ACURÁCIA	NO CLASS	ERROS	ACURÁCIA	NO CLASS	ERROS	ACURÁCIA	NO CLASS	ERROS	ACURÁCIA	NO CLASS
01	7	99,03%	0	105	85,42%	1	203	71,81%	1	26	96,39%	0
02	9	98,75%	0	53	92,64%	0	153	78,75%	1	40	94,44%	0
03	8	98,89%	0	68	90,56%	0	173	75,97%	0	90	87,50%	1
04	10	98,61%	0	48	93,33%	0	132	81,67%	0	160	77,78%	2
05	9	98,75%	0	70	90,28%	0	126	82,50%	0	92	87,22%	1
06	11	98,47%	0	58	91,94%	0	197	72,64%	0	89	87,64%	1
07	76	89,44%	1	103	85,69%	1	127	82,36%	1	82	88,61%	1
08	11	98,47%	0	51	92,92%	0	181	74,86%	0	157	78,17%	2
09	10	98,61%	0	46	93,61%	0	126	82,50%	0	154	78,61%	2
10	13	98,19%	0	99	86,25%	1	165	77,08%	1	40	94,44%	0
MÉDIA	16,4	97,72%	0,1	70,1	90,26%	0,3	158,3	78,01%	0,4	93	87,08%	1

A figura 2 apresenta os resultados dos testes de classificação também realizados em todos os 10  *folds*, nas quatro categorias de ruídos considerados. Foram utilizadas as bases de dados de treinamento clusterizadas na fase anterior, bem como os conjuntos de testes referentes a cada *fold*. Os três classificadores testados foram avaliados quanto ao número de erros e acertos, acurácia, número de regras geradas e a quantidade de literais nas regras (número de vezes que algum atributo foi utilizado na da regra).

	RUIÍDO 12,5%						RUIÍDO 25%				
	ACERTOS	ERROS	ACURÁCIA	REGRAS	LITERAIS		ACERTOS	ERROS	ACURÁCIA	REGRAS	LITERAIS
<i>jRip</i>	63	17	79,00%	25	74	<i>jRip</i>	61	19	76,00%	27	77
<i>J48</i>	78	2	98,00%	20	39	<i>J48</i>	71	9	88,00%	36	72
<i>RandomForest</i>	79	1	99,00%	2971	5742	<i>RandomForest</i>	78	2	97,00%	5979	11799
	RUIÍDO 37,5%						RUIÍDO 25% <sup>1</sup>				
	ACERTOS	ERROS	ACURÁCIA	REGRAS	LITERAIS		ACERTOS	ERROS	ACURÁCIA	REGRAS	LITERAIS
<i>jRip</i>	50	30	63,00%	31	95	<i>jRip</i>	60	20	75,00%	26	73
<i>J48</i>	60	20	76,00%	62	122	<i>J48</i>	70	10	88,00%	35	68
<i>RandomForest</i>	73	7	92,00%	9329	18458	<i>RandomForest</i>	73	7	91,00%	5370	10542

**Figura 2. Acurácias dos modelos utilizados - Fonte: Próprio Autor.**

## 4.3. Análise e Discussão

Levando em consideração a divisão das bases em  *folds* para a criação dos conjuntos de treinamento e teste, o *FarthestFirst* foi utilizado por 10 vezes em cada base, ou seja, em cada um dos conjuntos de treinamento. Assim, o *FarthestFirst* obteve na média uma acurácia de 97,72% na base de 12,5%, 90,26% na base de 25%, e na base de 37,5%, a acurácia obtida foi de 78,01%. Na base de dados de 25% de ruído aleatório, a acurácia foi de 87,08%.

<sup>1</sup>Base de dados com 25% de ruído aleatório entre os atributos.

Após as bases de dados serem clusterizadas, elas serão utilizadas para a construção de classificadores para que estes gerem as regras de identificação dos EA dos estudantes. Através do número de erros e acertos nos conjuntos de teste, pôde-se calcular a acurácia de cada algoritmo, e ao se visualizar as regras geradas, é possível identificar a complexidade de cada modelo (número de regras geradas e número de literais usados pelas regras).

Assim como na fase de clusterização, a classificação foi realizada em todos os 10 folds nos quatro tipos de ruídos. Analisando apenas o quesito acurácia, na maioria dos folds, dos três classificadores testados, o que apresentou os melhores resultados foi o *RandomForest*, seguido pelo *J48*, e em último o *jRip*. Entretanto, outras variáveis, como número de regras geradas e quantidade de literais nas regras também devem ser levados em consideração. Neste caso, o *RandomForest*, que obteve a melhor acurácia, apresenta aqui o pior resultado, gerando milhares de regras e literais, enquanto o *jRip* e o *J48* apresentam uma quantidade bem reduzida nestes quesitos.

Os resultados do Teste t confirmaram que, quanto à acurácia, o *RandomForest* é realmente o melhor algoritmo comparado aos demais na maioria dos experimentos realizados. Seus resultados são estatisticamente equivalentes apenas aos do *J48* na base de dados de 12,5%, quando o valor retornado do p-valor foi de 0.6341. Adicionalmente, pôde-se constatar que o *jRip* é inferior aos demais classificadores considerando os resultados do Teste t.

Observa-se, pelos resultados, que o *FarthestFirst* apresenta-se como um algoritmo promissor na identificação de *clusters* de alunos que possuem o mesmo EA. Observa-se também que dos três classificadores analisados, o *J48* tende a ser o melhor para ser utilizado na abordagem proposta, uma vez que, o *RandomForest*, mesmo obtendo acurácias mais altas, possui uma maior complexidade de regras. Por sua vez, o *jRip*, apesar de possuir uma complexidade de regras semelhantes ao *J48*, apresenta acurácia inferior.

A Figura 3 ilustra as regras geradas em um dos folds pelo *J48*. Neste caso, o algoritmo gerou uma árvore com 21 regras, utilizando 41 literais.

```
=== Classifier model (full training set) ===
J48 pruned tree
-----
exerNv <= 0.25
|
| tesNr <= 0.2
| |
| | pagNv <= 0.1
| | |
| | | forNp <= 1: classe12 (43.0)
| | | forNp > 1: classe14 (45.0)
| | |
| | | pagNv > 0.1
| | | |
| | | | forNp <= 1: classe13 (42.0)
| | | | forNp > 1: classe15 (45.0/1.0)
| | |
| | | tesNr > 0.2
| | | |
| | | | forNp <= 1
| | | | |
| | | | | pagNv <= 0.1: classe08 (42.0)
| | | | | pagNv > 0.1: classe09 (47.0)
| | | |
| | | | forNp > 1
| | | | |
| | | | | pagNv <= 0.1: classe10 (44.0)
| | | | | pagNv > 0.1: classe11 (46.0)
| | |
| | | exerNv > 0.25
| | | |
| | | | exerNr <= 0.2
| | | | |
| | | | | pagNv <= 0.1
| | | | | |
| | | | | | forNp <= 1: classe04 (46.0)
| | | | | | forNp > 1: classe06 (45.0/1.0)
| | | | |
| | | | | pagNv > 0.1
| | | | | |
| | | | | | forNp <= 1: classe05 (47.0)
| | | | | | forNp > 1
| | | | | | |
| | | | | | | exerDqe <= 0.03
| | | | | | | |
| | | | | | | | exerDe <= 0.89: classe07 (4.0)
| | | | | | | | exerDe > 0.89: classe05 (2.0)
| | | | | | | | exerDqe > 0.03: classe07 (41.0)
| | | | |
| | | | | exerNr > 0.2
| | | | | |
| | | | | | exerDag <= 0.5
| | | | | | |
| | | | | | | objTexNv <= 0.49
| | | | | | | |
| | | | | | | | objTexTg <= 0.43: classe00 (39.0)
| | | | | | | | objTexTg > 0.43
| | | | | | | | |
| | | | | | | | | exerTg <= 0.85: classe02 (2.0)
| | | | | | | | | exerTg > 0.85: classe08 (3.0)
| | | | | | | |
| | | | | | | | objTexNv > 0.49: classe02 (44.0)
| | | | | | |
| | | | | | | exerDag > 0.5
| | | | | | | |
| | | | | | | | forNp <= 1: classe01 (46.0)
| | | | | | | | forNp > 1
| | | | | | | | |
| | | | | | | | | resTg <= 0.15: classe02 (3.0/1.0)
| | | | | | | | | resTg > 0.15: classe03 (44.0)
| | | | |
| | |
| |
|
Number of Leaves :    21
Size of the tree :    41
```

Figura 3. Regras J48 - Fonte: Próprio Autor.



Na árvore apresentada, o *j48* inicia analisando o atributo **exerNv**, que representa o número de visitas a exercícios. Se este número for menor ou igual a 25%, o algoritmo analisa o atributo **tesNr** (Nº de revisões em testes de auto-avaliação), caso contrário, o atributo a ser analisado é o **exerNR** (Nº de revisões em exercícios). Neste sentido, se o **tesNr** for menor ou igual 20%, será analisado o atributo **pagNv**, que representa o número de visitas em páginas, e se for maior que 20%, será analisado o atributo **forNp** (número de postagens em fóruns).

Com esta mesma lógica todos os níveis da árvore de decisão são interpretados, até que a regra de decisão seja gerada por completo (atingimento de um nó folha), e identifique-se em qual das 16 classes determinado comportamento pode se encaixar. Neste exemplo, a 1ª regra gerada é: **Se**  $\text{exerNv} \leq 0.25$  e  $\text{tesNr} \leq 0.2$  e  $\text{pagNv} \leq 0.1$  e  $\text{forNp} \leq 1$  **então** classe12).

## 5. Considerações Finais

Conforme apresentado, a maioria das pesquisas no âmbito de detecção de EA utilizam a abordagem colaborativa, a qual necessita de informações explicitamente fornecidas por parte do estudante. Considerando que métodos que optam por esse tipo de abordagem estão sujeitos a falhas, esta pesquisa optou por utilizar a abordagem automática.

Neste sentido, propôs-se a aplicação das técnicas de clusterização e classificação, e devido à baixa representação de alguns EA em bases de dados reais, outra contribuição desta pesquisa foi a introdução a um simulador de alunos que emula o comportamento dos mesmos de acordo com seus EA. Este simulador foi utilizado para gerar o comportamento de alunos distribuídos nos 16 possíveis EA representados no FSLSM.

A implementação da abordagem proposta utilizou o algoritmo *FarthestFirst* na fase da clusterização e os algoritmos *jRip*, *J48* e *RandomForest* na etapa de classificação. Adicionalmente, desenvolveu-se um algoritmo que por meio dos centróides gerados pelo clusterizador identifica qual *cluster* está associado ao EA do aluno. Constatou-se que algoritmo de clusterização *FarthestFirst* é promissor na identificação de *clusters* de alunos que pertencem ao mesmo EA, e que o algoritmo *J48* foi o que melhor se adequou à abordagem proposta.

Como trabalho futuro, pretende-se aprofundar os estudos sobre as regras extraídas deste trabalho e utilizá-las na classificação de alunos em bases de dados reais do *Moodle*. Adicionalmente, realizar-se-á um estudo dos parâmetros das diversas técnicas adotadas para a verificação de possíveis relações entre seus valores e os resultados obtidos.

## Referências

- Bernard, J., Chang, T. W., Popescu, E., and Graf, S. (2017). Learning style Identifier: Improving the precision of learning style identification through computational intelligence algorithms. *Expert Systems with Applications*, 75:94–108.
- Borges, G. and Stiubiener, I. (2014). Recommending learning objects based on utility and learning style. *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, 2015-Febru(February):1–9.

- Carvalho, V., Araújo, R. D., Ferreira, H., Cattelan, R., and Dorça, F. (2017). OntaES: Uma Ontologia para Sistemas Adaptativos Educacionais Baseado em Objetos de Aprendizagem e Estilos de Aprendizagem. page 1307.
- Daniel, R., Dorça, F. A., and Resende, D. T. (2015). Recomendação de conteúdo personalizada com base em estilos de aprendizagem: uma abordagem prática. *Revista Brasileira de Informática na Educação*, 23(03):12.
- Felder, R. and Silverman, L. (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(June):674–681.
- Felder, R. and Soloman, B. (1991). Learning Styles and Stratgies. *Strategies*, pages 10–12.
- Frank, E., Hall, M. A., and Witten, I. H. (2016). The WEKA Workbench. Online Appendix for Data Mining: Practical Machine Learning Tools and Techniques.
- Graf, S. and Kinshuk (2009). Advanced Adaptivity in Learning Management Systems by Considering Learning Styles. In *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 235–238. IEEE.
- Graf, S., Kinshuk, and Liu, T. C. (2008). Identifying learning styles in learning management systems by using indications from students' behaviour. *The 8th IEEE International Conference on Advanced Learning Technologies*, pages 482–486.
- Khribi, M. K., Jemni, M., Nasraoui, O., Graf, S., and Kinshuk (2013). Toward a Fully Automatic Learner Modeling Based on Web Usage Mining with Respect to Educational Preferences and Learning Styles. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 403–407. IEEE.
- Kirschner, P. A. and van Merriënboer, J. J. (2013). Do Learners Really Know Best? Urban Legends in Education. *Educational Psychologist*, 48(3):169–183.
- Kolekar, S. V., Pai, R. M., and Pai, M. (2017). Prediction of Learner's Profile based on learning styles in Adaptive E-learning System. *iJET*, 12(6):31–51.
- Kurilovas, E., Zilinskiene, I., and Dagiene, V. (2014). Recommending suitable learning scenarios according to learners' preferences: An improved swarm based approach. *Computers in Human Behavior*, 30:550–557.
- Pitigala Liyanage, M. P., Gunawardena, K. L., and Hirakawa, M. (2016). Detecting Learning Styles in Learning Management Systems Using Data Mining. *IPSJ Transactions on Computers and Education*, 2(1):1–10.
- Rawson, K., Stahovich, T. F., and Mayer, R. E. (2017). Homework and achievement: Using smartpen technology to find the connection. *Journal of Educational Psychology*, 109(2):208–219.
- Ribeiro, F. A. A., Fonseca, L. C. C., and Freitas, M. D. S. (2013). Recomendando Objetos de Aprendizagem a partir das hashtags postadas no Moodle. *XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013)*, 25(Cbie):82–91.
- Zaina, L. A. M., Bressan, G., Cardieri, M. A. A. C., and Rodrigues Júnior, J. F. (2012). e-LORS: Uma Abordagem para Recomendação de Objetos de Aprendizagem. *Revista Brasileira de Informática na Educação*, 20(1):04.