

As Tecnologias de Análise de Aprendizagem e os Desafios de Prever Desempenhos de Estudantes de Programação

Márcia G. de Oliveira¹

¹ Centro de Referência em Formação e EaD (Cefor)
Instituto Federal do Espírito Santo (IFES)
Vitória – ES – Brasil

clickmarcia@gmail.com

Abstract. *Predicting the future of the students' learning, anticipating the possibilities of failure to plan actions of reorientation of a learning process, is a challenge for computer science. Learning Analysis appears as a possibility to contemplate this challenge because it is an educational technique that aims to recognize profiles and learning trends from the collection and analysis of data in online environments. Knowing that the key processes of Learning Analysis are Selecting, Capturing, Adding and Reporting, Predicting, Using, Refining and Sharing, this work discusses the Predicting process, highlighting methodologies, technologies, practices, challenges and analysis of research paths of predictive learning analysis. Contemplating this discussion within the field of programming learning, this paper aims to present the art's state of predictive learning analysis, to propose a framework to predict programming performances and to point out ways to advance such research.*

Keywords: *Predictive Learning Analysis, Predicting of Performance, Programming Learning.*

Resumo. *Prever o futuro de aprendizagem de alunos, antecipando-se às possibilidades de fracasso escolar para planejar ações de reorientação do processo de aprendizagem, representa um desafio para as tecnologias do futuro. A Análise de Aprendizagem aparece como uma possibilidade de contemplar esse desafio, pois é uma técnica educacional que visa reconhecer perfis e tendências de aprendizagem a partir da coleta e da análise de dados de estudantes em ambientes online. Sabendo que os principais processos da Análise de Aprendizagem são Selecionar, Capturar, Agregar e Relatar, Prever, Usar, Refinar e Compartilhar, este trabalho traz à discussão o processo de Prever, destacando metodologias, tecnologias, práticas, desafios e caminhos de pesquisa da análise de aprendizagem preditiva. Contemplando essa discussão dentro do domínio da aprendizagem de programação, este trabalho tem como objetivos apresentar o estado da arte da análise de aprendizagem preditiva, propor um framework para previsão de desempenhos em programação e apontar caminhos para avançar nessas pesquisas.*

Palavras-chave: *Análise de Aprendizagem Preditiva, Previsão de Desempenhos, Aprendizagem de Programação.*

1. Introdução

A Análise de Aprendizagem é uma técnica educacional que visa reconhecer perfis e tendências de aprendizagem a partir da coleta e análise de dados estáticos e dinâmicos em ambientes *online* como, por exemplo, os ambientes virtuais de aprendizagem.

Os dados coletados e analisados a partir de interações em ambientes virtuais de aprendizagem possibilitam uma melhor compreensão e otimização de um processo de aprendizagem [Agudo-Peregrina et al. 2012]. Dessa forma, relacionando os dados em informações, a Análise de Aprendizagem busca, a partir destas, combinar um histórico de estados de aprendizagem de um estudante mapeados em desempenhos de atividades com seu atual estado de aprendizagem na tentativa de prever o que será útil para esse estudante no futuro [Lias and Elias 2011].

Atualmente a Análise de Aprendizagem tem despertado interesse em como os dados de interações de estudantes podem melhorar o ensino e a aprendizagem [Lias and Elias 2011]. No entanto, embora tenha havido um crescente interesse pela Análise de Aprendizagem e suas tecnologias adaptativas para melhorar o ensino e a aprendizagem, há poucos estudos que de fato detalhem resultados concretos [Johnson et al. 2016].

Os principais processos da Análise de Aprendizagem a partir de dados de interações em ambientes de aprendizagem são *Selecionar, Capturar, Agregar e Relatar, Prever, Usar, Refinar e Compartilhar* [Lias and Elias 2011]. Já as principais tecnologias utilizadas para o projeto de sistemas de Análise de Aprendizagem incluem mecanismos de Inteligência Artificial, *Data Mining*, Aprendizagem Adaptativa, *Inquiry*, Aprendizagem Baseada em Problemas, Modelagem Preditiva, *Clustering*, Mineração de Padrões, Mineração de Relacionamentos, Análise Sequencial, Predição de Desempenhos, Rastreamento de Conhecimento Bayesiano e Sequenciamento de Tarefas [Suchithra et al. 2015].

Neste trabalho, considerando o domínio da aprendizagem de programação, a ênfase será dada ao processo *Prever* e às tecnologias de predição de desempenhos como as Redes Neurais, as Árvores de Decisão, o *Clustering*, os Modelos de Regressão Linear e outras técnicas de *Data Mining* [Suchithra et al. 2015].

Embora haja algumas propostas de sistemas de previsão de desempenhos para o domínio da aprendizagem de programação [Watson et al. 2013], ainda há muitos desafios a serem vencidos, entre eles:

- Mapear perfis de aprendizagem de acordo com os desempenhos de estudantes em atividades de programação ao longo do tempo.
- Definir quais dados de entrada são altamente efetivos para sistemas de predição de desempenhos [Mat et al. 2013].
- Ter uma grande quantidade de informações de um histórico de desempenhos suficiente para prever com precisão desempenhos futuros.
- Tratar a variabilidade e a subjetividade da prática da programação que podem impactar em muitas variações de variáveis preditoras ao longo do tempo.
- Mapear perfis não apenas por conteúdos, mas por habilidades extraídas a partir de informações de códigos de programação desenvolvidos por alunos.

Com o propósito de apresentar o estado da arte da análise de aprendizagem preditiva, este trabalho destaca como as tecnologias dessa área evoluíram desde a década de

60 até a década atual. Além disso, são apontados os principais desafios e caminhos de pesquisa da Análise de Aprendizagem de Programação para os próximos anos.

Considerando esses desafios, este trabalho propõe um *framework* de previsão de desempenhos de estudantes de programação. Esse *framework* tem o objetivo de apresentar uma solução computacional que possibilite aos professores de programação ter uma visão dos estados de aprendizagem dos estudantes na prática da programação e realizar previsões de desempenhos futuros desses estudantes. Dessa forma, antecipando-se ao fracasso escolar, professores podem reorientar o ensino e realizar ações formativas para que os alunos melhorem seus desempenhos em programação.

Para apresentar as aplicações, os desafios, os caminhos de pesquisa e um *framework* da previsão de desempenhos de estudantes de programação, este trabalho está organizado conforme a ordem a seguir. Na Seção 2, é apresentada a análise de aprendizagem preditiva, a sua evolução e suas aplicações. Na Seção 3, é proposto um *framework* para a previsão de desempenhos de estudantes de programação. Na Seção 4, apontam-se os desafios da previsão de desempenhos no domínio da aprendizagem de programação e possíveis caminhos de pesquisa. Na Seção 5, conclui-se este trabalho com as considerações finais e propostas de trabalhos futuros.

2. A Análise de Aprendizagem Preditiva

A análise preditiva tem como objetivo descobrir padrões a partir de um histórico de dados para descobrir tendências e estimar valores que trazem à compreensão essas tendências. No caso da análise de aprendizagem preditiva, o objetivo é estimar valores que descrevem, a partir da análise de perfis de estudantes, um futuro de sucesso ou de fracasso escolar.

Uma estratégia específica de previsão de desempenhos de estudantes no domínio da programação é apresentada por [Watson et al. 2013]. Nesse trabalho, o método de predição de desempenhos de estudantes é baseado em dados de *logs* descrevendo vários aspectos de comportamentos de programação. Uma inovação desse método consiste em prever desempenhos baseando-se em como estudantes respondem a diferentes tipos de erros em comparação com seus pares [Watson et al. 2013].

A Tabela 1 apresenta a evolução de soluções propostas para análise de dados coletados e previsão de desempenhos de estudantes, especialmente no domínio da aprendizagem de programação, desde a década de 60 até a década de 90.

De acordo com a Tabela 1, as tendências de previsão de desempenhos em programação, na década de 60, estavam relacionadas a testes de aptidão que utilizavam técnicas de análise de regressão linear múltipla com os objetivos de identificar fatores relacionados a desempenhos, de selecionar profissionais e de reduzir custos de treinamento. Na década de 70, com o objetivo de avaliar competências específicas, houve uma maior preocupação de se analisar tanto as características dos programadores quanto dos programas utilizando, para isso, representações como as métricas de *software* [Curtis et al. 1979].

A partir da década de 80, a previsão de desempenhos de estudantes começa a ter uma perspectiva de reorientar o processo de aprendizagem de estudantes e, na década de 90, através do projeto de STIs (Sistemas Tutores Inteligentes) com funções de modelagem de estados de conhecimento, previsão de desempenhos e recomendação de sequências de atividades, visava-se monitorar e regular o processo de aprendizagem de estudantes para

Trabalhos de Previsão de Desempenhos de Estudantes - Características e Contribuições	
Década de 60	
[MCNAMARA and HUGHES 1961]	Apresenta uma revisão de pesquisas em seleção de programadores, de instrumentos de avaliação de performance e de pesquisa de variáveis predictoras mais relacionadas aos desempenhos de programadores.
[Badgley et al. 1962]	Relaciona atributos psicológicos, sociais e acadêmicos para prever sucesso ou fracasso de estudantes de medicina; Determina fatores úteis para modelos de predição; Analisa histórico de dados dos estudantes; Aplica técnica de análise de regressão linear; Identifica fatores de sucesso ou fracasso escolar; Desempenhos são preditos com 81% de acurácia.
[Bauer et al. 1968]	Aplica de testes para identificar aptidão em programação em processo seletivo; A finalidade é evitar desperdício de tempo e de custos em treinamentos; Aplica modelos de regressão linear múltipla com técnica <i>stepwise</i> para selecionar melhor combinação de variáveis para prever desempenhos; Compara a eficácia preditiva de vários testes; Os escores dos testes foram correlacionados com a nota final.
Década de 70	
[DeNELSKY and McKEE 1974]	Previsão de desempenhos com o objetivo de identificar competências específicas em programação; Criação de testes designados para medir aptidões específicas.
[Motley and Brooks 1977]	Previsão de taxas de erros de programação; Aplicação de análise de regressão linear múltipla a partir de características de programas e de variáveis de programadores.
[Curtis et al. 1979]	Investigação de características de software relacionadas à complexidade psicológica; Foco em validar o uso de métricas de complexidade de software para prever desempenhos de programadores; Reconhece como melhores preditores as métricas de <i>Halstead</i> , de <i>McCabe</i> e o número de linhas de código.
Década de 80	
[Hostetler 1983]	Previsão de aptidão em programação medindo certas habilidades cognitivas, traços de personalidade e histórico de desempenhos; O objetivo primário do estudo foi criar um modelo para prever sucesso em programação para reorientar estudantes de programação.
[Butcher and Muth 1985]	Previsão de desempenhos de calouros em cursos de Ciência da Computação a partir de informações sobre programas e históricos de desempenhos escolares junto com resultados de testes do <i>American College (Act)</i> .
[Werth 1986]	Estudos sobre a relação entre a nota do estudante no começo de um curso com sexo, idade, escola, performances acadêmicas anteriores, número de cursos de matemática e experiência de trabalho; Não há relação entre desempenhos com o tipo de personalidade; Relações mais significantes com nível de estudo, número de horas trabalhadas e número de aulas de matemática no ensino secundário.
Década de 90	
[Corbett and Anderson 1994]	Modelagem dos estados de conhecimento dos estudantes; O modelo de estudante ideal contém um conjunto de regras de produção ideal que forma o núcleo do Sistema Tutor Inteligente (STI); Esse modelo auxilia o tutor a monitorar o estado de conhecimento do aluno e a planejar sequências de atividades a serem recomendadas conforme necessidades do aprendiz; O modelo resultante prediz desempenhos com eficácia, prevê erros e, principalmente, possibilita a maioria dos estudantes alcançar altos níveis de desempenhos.

Tabela 1. A Evolução das Técnicas de Previsão de Desempenhos

que estes alcançassem êxitos de aprendizagem. Um exemplo de sistema tutor inteligente com essa perspectiva foi aplicado em um curso de programação introdutória com excelentes resultados de aprendizagem [Corbett and Anderson 1994].

A Tabela 2 apresenta as propostas mais recentes de previsão de desempenhos de estudantes, especialmente para o domínio da programação.

Trabalhos mais recentes (a partir de 2000)

[Minaei-Bidgoli et al. 2003]	Tecnologias educacionais da web para coleta de dados de estudantes; Coleta rotineira de dados em grandes quantidades por sistemas web; Aplicação de técnicas de <i>Data mining</i> ; Abordagem para classificar alunos afim de prever sua nota final a partir de características reconhecidas nos dados coletados; Combinação de vários classificadores para melhorar a eficácia de classificação; Considera peso de características e aplica um algoritmo genético para melhorar precisão de previsão; Visa identificar alunos em risco e permitir o instrutor orientá-los em tempo hábil.
[Kotsiantis et al. 2010]	Combinação de classificadores utilizando métodos de votação e algoritmos de aprendizagem <i>online</i> para previsão de desempenhos de estudantes em curso de informática a distância.
[Romero and Ventura 2010]	Apresenta as principais propostas de previsão de desempenhos utilizando técnicas de <i>data mining</i> .
[Romero et al. 2013]	Especifica ferramenta de mineração de dados em <i>Moodle</i> ; Compara experimentalmente várias técnicas de mineração de dados para previsão de desempenhos; Apresenta exemplos de modelos descobertos e explica que modelos de classificadores são apropriados para ambientes educacionais, mais eficazes e mais compreensíveis para os professores.
[Sorour et al. 2015]	Previsão de desempenhos de estudantes pelo métodos LDA (<i>Latent Dirichlet Allocation</i>) e SVM (<i>Support Vector Machine</i>); Melhores resultados de previsão de desempenhos em relação aos modelos LSA (<i>Latent Semantic Analysis</i>) e PLSA (<i>Probabilistic Latent Semantic Analysis</i>).
[Naser et al. 2015]	Um modelo de Redes Neurais Artificiais é utilizado para prever desempenhos de estudantes dos cursos de Engenharia e Tecnologia da Informação; Tem como objetivos identificar fatores que afetam desempenhos, converter esses fatores em formas apropriadas para uma codificação de sistemas adaptativos e prever desempenhos de um estudante a partir de dados desse estudante. São considerados fatores como desempenhos anteriores, escores em disciplinas de Matemática e gênero; O modelo foi desenvolvido e treinado abrangendo cinco gerações de graduados; O Modelo prevê corretamente o desempenho de mais de 80% dos estudantes.
[Qiu et al. 2016]	Modelagem e previsão de comportamentos de estudantes em MOOCS (<i>Massive Open Online Courses</i>) através da análise profunda de dados demográficos, padrões de atividades de aprendizagem em fóruns, vídeos e exercícios; Proposta de <i>LadFG</i> (<i>Dynamic Factor Graph</i>) para incorporar esses dados de estudantes; Duas tarefas de previsão: previsão de desempenhos em exercícios e previsão de receber certificação; Modelo flexível que pode ser aplicado a várias configurações.
[Strang 2016]	A contribuição metodológica é a combinação de métodos onde a hipótese quantitativa foi testada e seguida pela coleta de dados qualitativos na análise de textos e mais análise quantitativa para descobrir padrões em textos <i>online</i> ; Foco na análise de aprendizagem <i>online</i> ; Vários atributos importantes dos estudantes e suas atividades são examinadas para identificar o que melhor favorece a previsão de escores mais altos; Tem o propósito de explorar relações entre notas de estudantes e fatores de aprendizagem utilizando um amplo conjunto de dados de um curso; Estudos atuais apontam para a habilidade de prever resultados de aprendizagem a partir da análise de <i>Big Data</i> , mas poucos indicadores para isso têm sido de fato encontrados; Uso de preditores quantitativos e dados qualitativos extraídos de textos de estudantes; Quatro preditores de interação <i>online</i> são utilizados para explicar modelo de regressão.

Tabela 2. Estado da Arte da Previsão de Desempenhos

Atualmente, a análise de aprendizagem preditiva tem como tendências a coleta de dados em sistemas *web*, a análise de dados em *Big Data* utilizando técnicas de *data mining*, a combinação de técnicas de classificação para melhorar eficácia de previsão e a utilização de modelos de regressão linear múltipla para a estimação de desempenhos.

De acordo com a revisão do estado da arte da previsão de desempenhos no domínio da aprendizagem de programação, o *framework* proposto neste trabalho tem como objetivos específicos: analisar um histórico de códigos-fontes de programas desenvolvidos por estudantes em ambientes *online*, combinar características de códigos-fontes para modelar estados de aprendizagem ao longo de um curso gerando perfis, prever desempenhos de estudantes e realimentar estados de aprendizagem por recomendação de atividades de acordo com necessidades dos estudantes reconhecidas em seus perfis. O objetivo geral dessa proposta é, por conseguinte, alcançar êxitos coletivos de aprendizagem.

3. Um *Framework* para Previsão de Desempenhos em Programação

A proposta metodológica do *framework* de previsão de desempenhos deste trabalho aplica as ideias do STI de [Corbett and Anderson 1994] e do sistema de recomendação de atividades proposto por [Oliveira et al. 2013].

A Figura 1 apresenta o *framework* combinando as funcionalidades de mapeamento de perfis de estudantes, de previsão de desempenhos e de regulação da aprendizagem para o domínio da aprendizagem de programação.

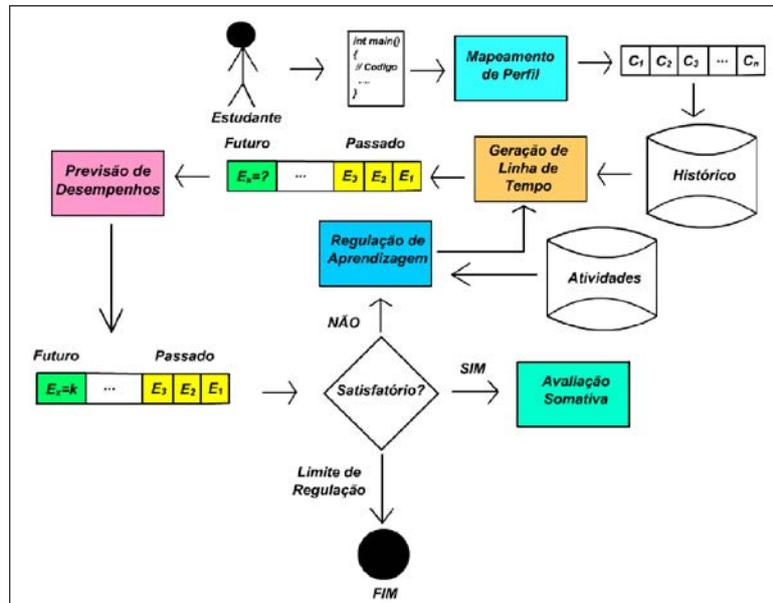


Figura 1. Framework de Previsão de Desempenhos

De acordo com a Figura 1, cada programa desenvolvido por um *Aluno* é mapeado em um perfil representado por um conjunto de componentes de habilidades que são variáveis de avaliação do domínio da programação [Oliveira et al. 2015b]. Em seguida, esse perfil é adicionado a um *histórico* de perfis do mesmo aluno ao longo de um curso. Para a *Geração de Linha do Tempo do Estudante*, cada perfil do histórico é mapeado em um estado de aprendizagem E_i , que pode ser o escore de cada exercício.

O estado de aprendizagem a ser predito é E_x . Para a *Previsão de Desempenhos*, é criado um modelo de regressão linear a partir dos perfis do *Histórico* e dos estados de aprendizagem E_i a eles associados. Em seguida, é predito um valor k para a variável independente E_x do modelo.

A partir do estado de aprendizagem predito, o processo de aprendizagem do aluno na prática da programação é reorientado. Se o estado for *satisfatório*, o aprendiz já poderá ser submetido a processos de *Avaliação Somativa* para qualificação de aprendizagem. Se não, ele é submetido a um processo de *Regulação da Aprendizagem* que consiste em ações de recomendação de atividades conforme dificuldades de aprendizagem. Uma nova linha do tempo é então criada a partir das novas atividades resolvidas e o estado de aprendizagem E_x é predito novamente.

O processo de *Regulação da Aprendizagem* se repete até que o estudante alcance um estado de aprendizagem *satisfatório*. No entanto, há um *limite de regulação* que in-

forma se a regulação possui possibilidades de êxito considerando os avanços em uma linha do tempo. Não havendo mais essas possibilidades depois de várias ações de regulação, o processo é encerrado e um relatório é gerado informando os estados de aprendizagem do aluno e as suas dificuldades para serem avaliadas individualmente pelo professor.

4. Desafios e Caminhos de Pesquisa para a Previsão de Desempenhos

Embora algumas soluções para previsão de desempenhos de estudantes já tenham sido desenvolvidas e possam ser estendidas para o domínio da programação, para um domínio tão complexo como este, há muitos desafios a serem vencidos entre os quais destacamos:

1. Assumindo uma representação vetorial do domínio a partir de códigos-fontes desenvolvidos por alunos, conforme [Oliveira et al. 2015b], como definir as variáveis preditoras, isto é, as dimensões ou *features*, que melhor representem um histórico de diferentes classes de exercícios de programação ao longo de um curso e sejam adequadas para formar um modelo de previsão de desempenhos?
2. Como tratar as variações de estilos nos códigos de programação de estudantes ao longo de uma linha do tempo?
3. Como estabelecer medições que garantam a confiabilidade dos sistemas de previsão de desempenhos de forma que motivem professores a utilizá-los? Será possível, a partir de um histórico de exercícios avaliados e mapeados em uma perspectiva multidimensional, prever o desempenho de um aluno na prova?
4. Como construir esse histórico de exercícios, uma vez que será necessária uma grande quantidade de amostras para cada aluno, sem demandar esforço em excesso de professores para corrigir todos esses exercícios?
5. No contexto de *MOOCs* (*Massive Open Online Courses - Cursos Online Abertos e Massivos*), como prever desempenhos de uma grande quantidade de alunos com a demanda de uma grande quantidade de exercícios pré-avaliados por professores?

Para vencer cada um desses desafios da previsão de desempenhos de estudantes de programação, apontamos os seguintes caminhos de pesquisa, respectivamente:

1. Na recomendação semi-automática de exercícios de programação para estudantes com dificuldades de aprendizagem, o trabalho de [Oliveira et al. 2013] representa o domínio da aprendizagem de programação a partir de informações de programas escritos em Linguagem C. Essas informações são mapeadas em vetores cujas dimensões são variáveis de avaliação de programação. Essa ideia pode ser, portanto, utilizada para representar perfis de alunos para prever seus desempenhos.
2. Uma proposta para tratar as variações em estilos de programação ao longo de um curso, é criar uma representação genérica dos exercícios de programação resolvidos pelos alunos através de métricas alternativas que quantifiquem padrões ou tendências de programação como esforço, complexidade, estilo e outras métricas [Hung et al. 1993].
3. Uma solução é identificar exercícios de programação resolvidos por alunos ao longo de um curso cujas soluções sejam estruturalmente semelhantes às soluções das questões de prova. Uma outra proposta seria comparar históricos de estudantes reformulando o problema de previsão em um problema de recomendação de filtragem colaborativa em que um aluno recebe a nota que outros alunos com históricos semelhantes ao seu receberiam. Ainda assim, para o domínio da programação, há altas variações no desenvolvimento de programas de aluno para aluno. Dessa forma, a confiabilidade de sistemas de previsão de desempenhos em programação ainda representa um grande desafio de pesquisa.

4. Nesse caso, sistemas de avaliação semi-automática de exercícios de programação poderiam ser utilizados em combinação com sistemas de previsão de desempenhos. Dessa forma, durante um curso, para cada exercício, apenas algumas amostras selecionadas seriam corrigidas por professores e as demais, por um sistema de avaliação semi-automática como o proposto por [Oliveira 2013] e melhorado por [Oliveira et al. 2015a]. Outras opções de avaliação automática de exercícios de programação são apresentadas por [Oliveira and Oliveira 2015].
5. No contexto de *MOOCs* [Pieterse 2013], com o apoio de sistemas de avaliação automática de exercícios de programação apontados no item anterior, a previsão de desempenhos pode tornar-se mais confiável uma vez que, havendo mais alunos, haverá mais históricos de exercícios e maior representação da diversidade de soluções desses exercícios. Nesse caso, o verdadeiro desafio a ser vencido é a redução de esforço de professores ao avaliar exemplos de soluções para cada um dos vários exercícios de programação aplicados com a finalidade de gerar o treino de sistemas de avaliação semi-automática. O ideal é, portanto, que, para cada exercício, o professor avalie o menor número possível de exemplos de soluções e que esses exemplos sejam os mais representativos de um conjunto com uma grande quantidade de amostras de soluções. O trabalho de [Oliveira et al. 2015a], por exemplo, é um passo inicial e relevante nesse caminho de pesquisa porque propõe uma estratégia de seleção de amostras e *features* de um conjunto de soluções de programação para formar o conjunto de treino de uma avaliador semi-automático.

5. Considerações Finais

Este trabalho apresentou um sumário do estado da arte da análise de aprendizagem preditiva, destacando a sua evolução, os métodos propostos, os desafios a serem vencidos e os caminhos de pesquisa a serem seguidos dentro do domínio da aprendizagem de programação. Além disso, propomos um *framework* para previsão de futuros desempenhos de estudantes a partir de um histórico de desempenhos em atividades de programação.

Para os próximos anos, os principais desafios da previsão de desempenhos em avaliações de programação são os seguintes: dispor de uma grande quantidade de informações a partir de um histórico de desempenhos suficiente para prever com precisão os desempenhos futuros e tratar a variabilidade e a subjetividade tão presentes no processo de construção de programas, o que pode impactar em ampla variabilidade nas variáveis preditoras de modelos de análise de aprendizagem preditiva.

A contribuição deste trabalho é promover a discussão da análise de aprendizagem preditiva na comunidade científica de interesse e incentivar, a partir deste estudo, o desenvolvimento de estratégias de previsão de desempenhos em programação com as finalidades de, antecipando-se a um futuro de fracasso escolar, reorientar práticas de ensino e remediar o processo de aprendizagem para que alunos melhorem seus desempenhos.

Referências

- Agudo-Peregrina, A., Hernández-García, A., and Iglesias-Pradas, S. (2012). Predicting academic performance with learning analytics in virtual learning environments: A comparative study of three interaction classifications. In *Computers in Education (SIIE), 2012 International Symposium on*, pages 1–6.
- Badgley, R. F., Hetherington, R. W., and Macleod, J. W. (1962). Social characteristics and prediction of academic performance of saskatchewan medical students. *Canadian Medical Association Journal*, 86(14):624.
- Bauer, R. et al. (1968). Predicting performance in a computer programming course.

- Butcher, D. F. and Muth, W. A. (1985). Predicting performance in an introductory computer science course. *Commun. ACM*, 28(3):263–268.
- Corbett, A. T. and Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278.
- Curtis, B., Sheppard, S. B., Milliman, P., Borst, M. A., and Love, T. (1979). Measuring the psychological complexity of software maintenance tasks with the halstead and mccabe metrics. *IEEE Trans. Softw. Eng.*, 5(2):96–104.
- DeNELSKY, G. Y. and McKEE, M. G. (1974). Prediction of computer programmer training and job performance using the aabp test1. *Personnel Psychology*, 27(1):129–137.
- Hostetler, T. R. (1983). Predicting student success in an introductory programming course. *SIGCSE Bull.*, 15(3):40–43.
- Hung, S.-l., Kwok, L.-f., and Chung, A. (1993). New metrics for automated programming assessment. In *Proceedings of the IFIP WG3.4/SEARCC (SRIG on Education and Training) Working Conference on Software Engineering Education*, pages 233–243, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.
- Johnson, L., Becker, S. A., Cummins, M., Estrada, V., Freeman, A., and Hall, C. (2016). *NMC Horizon Report: 2016. Higher Education Edition. Learning Analytics and Adaptive Learning: Time-to-Adoption Horizon: One Year or Less*. The New Media Consortium, Austin, Texas.
- Kotsiantis, S., Patriarcheas, K., and Xenos, M. (2010). A combinational incremental ensemble of classifiers as a technique for predicting students’ performance in distance education. *Knowledge-Based Systems*, 23(6):529–535.
- Lias, T. E. and Elias, T. (2011). Learning analytics: The definitions, the processes, and the potential.
- Mat, U. B., Buniyamin, N., Arsad, P. M., and Kassim, R. (2013). An overview of using academic analytics to predict and improve students’ achievement: A proposed proactive intelligent intervention. In *Engineering Education (ICEED), 2013 IEEE 5th Conference on*, pages 126–130.
- MCNAMARA, W. J. and HUGHES, J. L. (1961). A review of research on the selection of computer programmers. *Personnel Psychology*, 14(1):39–51.
- Minaei-Bidgoli, B., Kashy, D. A., Kortemeyer, G., and Punch, W. F. (2003). Predicting student performance: an application of data mining methods with an educational web-based system. In *Frontiers in Education, 2003. FIE 2003 33rd Annual*, volume 1, pages T2A–13.
- Motley, R. and Brooks, W. (1977). Statistical prediction of programming errors. Technical report, DTIC Document.
- Naser, S. A., Zaqout, I., Ghosh, M. A., Atallah, R., and Alajrami, E. (2015). Predicting student performance using artificial neural network: in the faculty of engineering and information technology. *International Journal of Hybrid Information Technology*, 8(2):221–228.
- Oliveira, M., Jiménez, N., Daher, P., and Oliveira, E. (2015a). Representação da diversidade de componentes latentes em exercícios de programação para classificação de

- perfis. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 26, page 1177.
- Oliveira, M., Nogueira, M. A., and Oliveira, E. (2015b). Sistema de Apoio à Prática Assistida de Programação por Execução em Massa e Análise de Programas. In *XXIII Workshop sobre Educação em Computação (WEI) - CSBC 2015*, Recife, PE. SBC.
- Oliveira, M. G. (2013). *Núcleos de Avaliações Diagnóstica e Formativa para Regulação da Aprendizagem de Programação*. Tese de doutorado, Universidade Federal do Espírito Santo.
- Oliveira, M. G., Ciarelli, P. M., and Oliveira, E. (2013). Recommendation of programming activities by multi-label classification for a formative assessment of students. *Expert Systems with Applications*, 40(16):6641–6651.
- Oliveira, M. G. and Oliveira, E. (2015). Abordagens, práticas e desafios da avaliação automática de exercícios de programação. In *CSBC 2015 - DesafIE 2015*, Recife.
- Pieterse, V. (2013). Automated assessment of programming assignments. In *Proceedings of the 3rd Computer Science Education Research Conference on Computer Science Education Research*, CSERC '13, pages 4:45–4:56, Open Univ., Heerlen, The Netherlands, The Netherlands. Open Universiteit, Heerlen.
- Qiu, J., Tang, J., Liu, T. X., Gong, J., Zhang, C., Zhang, Q., and Xue, Y. (2016). Modeling and predicting learning behavior in moocs. *Science*, 45(50):55.
- Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., and Ventura, S. (2013). Web usage mining for predicting final marks of students that use moodle courses. *Computer Applications in Engineering Education*, 21(1):135–146.
- Romero, C. and Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618.
- Sorour, S. E., Goda, K., and Mine, T. (2015). Estimation of student performance by considering consecutive lessons. In *Advanced Applied Informatics (IIAI-AAI), 2015 IIAI 4th International Congress on*, pages 121–126. IEEE.
- Strang, K. D. (2016). Beyond engagement analytics: which online mixed-data factors predict student learning outcomes? *Education and Information Technologies*, pages 1–21.
- Suchithra, R., Vaidhehi, V., and Iyer, N. E. (2015). Survey of learning analytics based on purpose and techniques for improving student performance. *International Journal of Computer Applications*, 111(1).
- Watson, C., Li, F. W. B., and Godwin, J. L. (2013). Predicting performance in an introductory programming course by logging and analyzing student programming behavior. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 319–323.
- Werth, L. H. (1986). Predicting student performance in a beginning computer science class. In *Proceedings of the Seventeenth SIGCSE Technical Symposium on Computer Science Education*, SIGCSE86, pages 138–143, New York, NY, USA. ACM.