

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/337408010>

Análise comportamental em juízes online para predição do desempenho final de alunos em disciplinas de computação

Conference Paper · November 2019

DOI: 10.5753/cbie.sbie.2019.1906

CITATIONS

2

READS

14

4 authors, including:



Arthur Alves

Federal University of Amazonas

2 PUBLICATIONS 2 CITATIONS

[SEE PROFILE](#)



Elaine Harada Teixeira de Oliveira

Federal University of Amazonas

124 PUBLICATIONS 732 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Synthetic routes of the fundamental building blocks of life [View project](#)



Predictive and Prescriptive Analytics in programming courses supported by online judges [View project](#)

Análise comportamental em juízes online para predição do desempenho final de alunos em disciplinas de computação

Arthur Bindá Alves, Leandro S. G. Carvalho, Elaine H. T. Oliveira,
David Fernandes de Oliveira

¹Instituto de Computação – Universidade Federal Amazonas (UFAM)

{aba,galvao,elaine,david}@icomp.ufam.edu.br

Abstract. *The prediction of the final performance of students is a topic studied by several researchers, and with the popularization of online judges new approaches can be adopted. This paper aims to propose and validate a machine learning method to infer the final performance of a computer student using only data related to his behavior within an online judge. For that, quantitative attributes were created to represent the programming profile and analyzed in the form of time series. As a result, we observed that the predictive model obtained 85% accuracy in the middle of the course, in line with conventional approaches.*

Resumo. *A predição do desempenho final de um aluno é um tema estudado por diversos pesquisadores, e com a popularização de juízes online novas estratégias podem ser adotadas para esse fim. Este trabalho tem por objetivo propor e validar um método baseado em aprendizagem de máquina para inferir o desempenho final de um estudante de computação utilizando apenas dados relacionados ao seu comportamento dentro de um juiz online. Para isso, atributos quantitativos foram criados para representar o perfil de programação e analisados na forma de séries temporais. Como resultado, nosso modelo obteve uma acurácia de 85% na metade da disciplina, equiparando-se a abordagens convencionais.*

1. Introdução

Disciplinas de Introdução à Programação (CS1) são consideradas pela literatura como de alta complexidade para alunos iniciantes, pois demandam uma série de habilidades cognitivas desde compreender o problema até escrever o código do plano de solução do problema. A consequência disso é que, em média, um terço dos alunos do mundo todo não conseguem nota para serem aprovados. Para contornar essa situação, muitas instituições vêm adotando a metodologia de aprendizagem híbrida. Nessa metodologia, faz-se uso de sistemas online aliados às aulas presenciais [Galvão et al. 2016]. Esses sistemas online são ambientes de correção automática de código (ACAC), também conhecidos como juízes online [Wasik et al. 2018].

Essas ferramentas disponibilizam atividades para serem resolvidas e submetidas à correção no próprio sistema. Assim que a questão é submetida, o estudante recebe o *feedback* imediato, informando se a solução proposta está correta ou errada. As vantagens que o ACAC apresenta aos alunos também abrem novas oportunidades de pesquisa na área de mineração de dados educacionais (EDM), pois mais interações dos alunos com o sistema se tornam passivas de coleta. A EDM é uma área de pesquisa que foca em

utilizar métodos de mineração de dados para analisar dados coletados nesses ambientes [Baker et al. 2011].

Dentro desse escopo, trabalhos como o de Dwan et al. [2017] e Gottardo et al. [2012] aproveitam os dados coletados durante as aulas para prever o desempenho final dos estudantes. Dwan et al. [2017], por exemplo, modelou um classificador que obteve 86% de acurácia baseando-se em dados referentes ao perfil de programação e passando como parâmetro também as notas das avaliações. Entretanto documentou-se que, ao retirar as notas das provas do conjunto de dados de entrada do classificador, a acurácia caiu em 10 pontos percentuais. A partir desse fato, este trabalho busca propor um método para inferir o desempenho final de um estudante utilizando apenas dados relacionados ao seu comportamento em um ambiente de correção automática de código.

Para validar o método proposto neste trabalho, três perguntas de pesquisa foram estabelecidas a fim de comparar este modelo com o resultado final obtido por Dwan et al. [2017]. Essas perguntas são:

- É possível prever, com acurácia acima de 80%, a aprovação de um aluno levando em conta apenas atributos comportamentais?
- Qual o impacto que o atributo nota tem sobre o classificador?
- Como a importância dos atributos para o classificador varia ao longo da disciplina?

O presente artigo foi dividido em 5 seções. A Seção 2 apresenta os trabalhos relacionados que também realizaram predição de desempenho; a Seção 3 descreve a contextualização do método e dos experimentos apresentados neste trabalho; a Seção 4 apresenta a metodologia utilizada na condução dos experimentos; a Seção 5 apresenta os resultados alcançados; e, por fim, a Seção 6 apresenta as considerações finais e os trabalhos futuros.

2. Trabalhos Relacionados

Mineração de Dados é definida como o processo de descoberta de padrões que potencialmente podem ser úteis para a compreensão ou desenvolvimento de alguma atividade. Essa área da computação é extremamente abrangente e possui aplicações nas áreas da saúde, comércio e até mesmo na educação. A área de Mineração de Dados Educacionais, foca em compreender os dados coletados em contextos educacionais produzidos principalmente por estudantes e professores [Costa et al. 2013].

Os temas de pesquisa abordam problemas cotidianos do ensino como a detecção de alunos com chance de evasão e a previsão do desempenho final de alunos em cursos de educação a distância [Gottardo et al. 2012] ou em disciplinas de programação [Dwan et al. 2017].

Para alcançar esses resultados, os pesquisadores modificam e adotam métodos originários da mineração de dados convencional. Os principais são: Predição; Agrupamento e Mineração de relações [Baker et al. 2011]. Os trabalhos citados anteriormente aplicam majoritariamente um método de predição chamado classificação.

Um método pouco utilizado é a mineração de padrões sequenciais. Com foco em fazer associações temporais entre eventos [Baker 2010], essa técnica é utilizada para, por exemplo, determinar quais comportamentos de um estudante indicam um interesse no conteúdo apresentado. Esse método é comumente utilizado para identificar padrões de

navegação de consumidores em sites de compras [Yin et al. 2012] e pode ser usado para uma análise de estudantes em ambientes online.

Com a popularização do ensino pela internet, uma nova gama de atributos começou a ser coletada, como por exemplo: informações sobre navegação, cliques e tempo de permanência. Essa riqueza de dados ofereceu uma série de novas abordagens para a área de EDM, pois com dados mais granularizados, a precisão obtida pelos classificadores aumentou, aprofundando a compreensão sobre o problema.

Na área de predição de desempenho final, dois estudos se destacam. Dwan et al. [2017] extraiu 17 atributos da interação do aluno em um ACAC, uniu-os com as notas de avaliações e os introduziu em um classificador Random Forest. Esse trabalho obteve 86% de precisão, porém constatou-se que o desempenho do classificador estava diretamente dependente ao atributo nota e que seu resultado caía em 10% se a nota fosse desconsiderada [Dwan et al. 2017]. Gottardo et al. [2012] coletou 10 atributos comportamentais de alunos em um Ambiente Virtual de Aprendizagem (AVA). Unindo esses atributos com a nota das questões respondidas, ele alcançou a precisão de 77% no Random Forest. [Gottardo et al. 2012].

3. Contextualização

3.1. Juizes Online

Um juiz online, também chamado de ACAC, oferece problemas de programação para serem resolvidos pelos alunos através da submissão de códigos-fonte. Após a submissão do aluno, seu código-fonte é corrigido automaticamente através de casos de teste pré-cadastrados. Cada caso de teste possui um conjunto de entradas e saídas [Wasik et al. 2018]. Esse tipo de sistema vem se popularizando no ensino de disciplinas de Introdução à Programação (CS1) graças à correção automática de código, aliviando a carga de trabalho do professor e oferecendo um feedback instantâneo para os estudantes.

O sistema CodeBench¹ [Galvão et al. 2016], analisado neste trabalho, pertence à categoria de juizes educacionais [Wasik et al. 2018], cujo objetivo é auxiliar o processo de aprendizado. Para isso, juizes desta categoria fazem uso de abordagens como competição, gamificação e desafios para engajar os estudantes. O ambiente analisado neste trabalho utiliza mecânicas de gamificação para incentivar e recompensar comportamentos como resiliência e assiduidade.

Além disso, o CodeBench disponibiliza ao estudante um Ambiente Integrados de Desenvolvimento (IDE), composto por editor de códigos, compiladores e console de teste. Do ponto de vista pedagógico, essa característica beneficia os alunos, principalmente os *non-majors*, pois não precisam se preocupar em aprender comandos de terminal. Do ponto de vista da EDM, essa integração permite a coleta passiva de dados da interação do aluno com o juiz online, possibilitando conhecer com profundidade indicadores referentes a hábitos de estudo e aprendizado.

Entretanto, o hábito de estudo de um estudante de programação é dividido entre as atividades realizadas dentro e fora do juiz online. Por isso, os resultados desta pesquisa referem-se apenas a uma parte do hábito de estudo. Para comparar com fidelidade as análises feitas neste trabalho, é necessário que a plataforma analisada apresente os seguintes

¹<http://codebench.icomp.ufam.edu.br>

requisitos: i) encaixar-se na categoria de juízes online educacionais; ii) possuir IDE integrada ao ambiente de submissão de questões; iii) coletar dados durante a interação do estudante com a IDE.

3.2. Disciplina de Introdução à Programação

A estrutura da disciplina de CS1 gira em torno da solução e submissão de exercícios propostos pelo professor no juiz online. Existem dois tipos de exercícios: atividades de casa e exames. As atividades de casa consistem em dez questões relacionadas ao conteúdo apresentado durante a semana, com prazo de quinze dias para serem submetidas. Os exames são compostos por duas ou três questões, com duração fixa de duas horas para serem submetidas. Os exames são respondidos em laboratórios de informática com a fiscalização do professor.

A nota da disciplina é composta por uma média ponderada de sete exames e a média aritmética das atividades de casa. Os exames são programados para serem aplicados em intervalos fixos de 15 dias. Para obter aprovação, o aluno deve possuir uma média final maior ou igual a 5,0.

Os dados utilizados neste trabalho são oriundos de turmas de Introdução à Programação de 2017 e 2018 da Universidade Federal do Amazonas, abarcando 11 cursos de graduação presencial nas áreas de Ciências Exatas e Engenharia. Essa é a mesma base de dados utilizada no trabalho de Dwan et al. [2017]. A Tabela 1 apresenta e tipifica a base de dados analisada neste trabalho.

Tabela 1. Caracterização do dataset utilizado. Fonte: CodeBench

Atributos	Período letivo				Total
	2017-1	2017-2	2018-1	2018-2	
Nº de turmas	10	5	9	5	29
Nº de estudantes	463	177	465	180	1285
Nº de atividades de casa	1278	556	1550	893	4277
Nº de exames	163	103	182	107	555
Nº de códigos submetidos	38304	10942	47969	13458	110673
Nº de testes e submissões	419603	125848	455484	163821	1164756

4. Metodologia

O método utilizado nesta pesquisa coleta informações resultantes da interação dos alunos com o juiz online, e o seu processo é dividido em quatro etapas de mineração de dados: i) coleta de atributos; ii) engenharia de atributos; iii) classificação; iv) análise dos resultados.

4.1. Coleta de atributos

A coleta de atributos é a etapa precursora. Ela consiste na análise dos dados à procura de atributos que possam gerar ganho de informação para a etapa de classificação. Neste trabalho, a coleta foi feita em três arquivos principais. O primeiro é o arquivo de logs gerados pela ação do estudante com o ambiente de programação. Por ser composto por um editor de texto encapsulado por um compilador, o código submetido à avaliação pelo

aluno geralmente é escrito e testado nesse ambiente. Esse arquivo contém ações como: tecla pressionada, clique, cópia e colagem de texto.

O segundo é o arquivo do código submetido pelo aluno para a avaliação. A última versão avaliada pelo juiz online está disponível por completo, contendo ou não, importações de bibliotecas, linhas comentadas e nome das variáveis utilizadas pelo aluno.

O terceiro e último é o arquivo de execuções. Nele, podem ser encontradas as seguintes informações: erros de compilação, hora em que o código foi testado ou submetido à avaliação, resultados gerados pelo código e mensagens de resposta incompleta. Os atributos coletados nesta etapa foram:

- **num_submissões:** número de submissões à avaliação feitas em uma questão;
- **num_testes:** total de vezes que uma questão foi testada na IDE (compilada);
- **num_comentários:** total de linhas com comentário na questão submetida;
- **linhas_código:** número de linhas do código submetido;
- **linhas_log:** número de linhas de log geradas durante a interação do usuário com a IDE;
- **prop_paste:** proporção entre o número de caracteres colados (com CTRL+V) e número de caracteres digitados no editor de texto;
- **prop_delete:** proporção entre o número de caracteres deletados e número de caracteres digitados no editor de texto;
- **duração_atividade:** o tempo total (em minutos) que o aluno passa no ambiente solucionando uma questão.

4.2. Engenharia de atributos

A etapa de engenharia de atributos foi a mais complexa e demorada de todo o processo de mineração de dados. Durante essa etapa, os dados coletados são analisados a fim de identificar quais devem ser transformados, agrupados ou removidos.

Ao analisar-se a base coletada, percebem-se alguns problemas. A princípio, esta pesquisa utilizaria todos os exercícios disponíveis na base, porém aqueles do tipo atividade de casa possuem uma distribuição desbalanceada. Majoritariamente, os dados coletados das atividades de casa retratavam exercícios plagiados, incompletos ou não respondidos. Tentou-se contornar esse problema por meio da remoção de *outliers*, entretanto, notou-se que sua causa se encontrava na quantidade de dados inconclusivos, e não na existência de valores extremos.

Uma vez descartados os dados coletados a partir das atividades de casa, os dados oriundos dos exames se tornaram a única fonte de informação. Percebeu-se que esses dados possuíam uma distribuição mais homogênea, provavelmente devido a fatores como menor duração da atividade, alto impacto na média final e a presença do professor nos laboratórios de informática.

A etapa seguinte consistiu em inspecionar os atributos coletados a partir dos exames. Verificou-se que a quantidade de linhas de código (atributo **linhas_código**) aumentava no decorrer da disciplina devido à importações de bibliotecas da linguagem e não ao aumento da dificuldade das questões. Isso culminou na criação de um novo atributo, **linhas_código_filtrado**, que corresponde à média do número de linhas de código dos exercícios submetidos, desconsiderando importações de bibliotecas e comentários. O atributo **num_comentários** não mostrou correlação com o desempenho final e foi descartado, pois

os alunos *non-majors* da base não possuíam o hábito de escrever linhas de comentário em seus código.

Por fim a base de dados neste instante consistia em linhas referentes à questões de todos os exames feitos durante um semestre letivo. Como este trabalho mira uma avaliação por mineração de padrões sequenciais, as linhas que antes representavam questões se tornaram a média aritmética simples desses atributos. O motivo dessa mudança será explicada na seção de classificação. Ao final desta etapa, resultaram os seguintes atributos:

- **média_submissões:** média de submissões à avaliação feitas em um exame;
- **média_testes:** média de testes realizados durante um exame;
- **média_linhas_código:** média de linhas dos códigos submetidos;
- **média_linhas_código_filtrado:** média de linhas dos códigos submetidos, desconsiderando linhas com comentários ou importações de bibliotecas;
- **média_linhas_log:** média das linhas de log geradas durante um exame;
- **média_prop_paste:** média da proporção de caracteres colados durante um exame;
- **média_prop_delete:** média da proporção de caracteres apagados durante um exame;
- **duração_atividade:** o tempo (em minutos) que o aluno passou realizando o exame.

4.3. Classificação

Para realizar a etapa de classificação se faz necessário a definição de um modelo de dados e a escolha do classificador mais adequado para o problema estudado. Esta técnica de Mineração de Dados consiste em inferir um aspecto particular dos dados (variável a ser prevista) através da combinação de outros dados pertencentes ao mesmo contexto (variáveis preditoras). No presente trabalho, a variável a ser prevista é o desempenho final dos estudantes, utilizando os atributos apresentados na seção de Metodologia.

A modelagem dos dados deste trabalho se baseia em séries de tempo. Como este estudo se baseia na análise do comportamento, é possível identificar uma linearidade ou mudança no comportamento comparando dados obtidos em exames diferentes. Ao longo da disciplina são realizados sete exames e é a partir dos dados destes exames que o modelo de predição será capaz de prever o desempenho final do aluno. Considerando que, com os dados referentes ao sétimo exame já é possível saber a média final do aluno, esse sétimo exame foi descartado por não interessar ao tema de pesquisa.

A maneira adotada para representar o intervalo de tempo entre exames foi pela média dos atributos dos exames realizados com o exame em análise. Então o classificador ao analisar os dados do terceiro exame, também analisa o padrão do aluno no primeiro e segundo exame. Isso permite ao classificador compreender e distinguir qual o comportamento de um estudante aprovado ou reprovado.

O classificador escolhido foi o Random Forest. Essa escolha baseou-se nas conclusões e resultados apresentados por [Dwan et al. 2017] e [Gottardo et al. 2012] que aplicaram dados de origem semelhante em três outros classificadores: SVM, AdaBoost e Árvore de Decisão e obtiveram o melhor resultado com o Random Forest. Esse modelo é validado pelo método de validação cruzada com 10 partições e a base de dados inserida foi balanceada.

Duas medidas são utilizadas para avaliar os resultados obtidos pelos classificadores: acurácia e MCC. Acurácia é a proporção entre o número de estudantes corretamente

classificados e o total de estudantes analisado. O MCC (*Matthews Correlation Coefficient*) é uma métrica para avaliar a qualidade de classificadores binários. Seu cálculo leva em conta verdadeiros e falsos positivos e retorna um valor entre -1 e 1 . Um coeficiente perto de $+1$ indica uma predição perfeita, 0 aponta uma classificação aleatória e -1 um conflito entre os valores reais e preditos.

5. Resultados

Foram realizados dois experimentos. No primeiro, a base de dados de um único semestre foi inserida de cada vez no classificador a fim de avaliar padrões de comportamento e especificidades de cada semestre de maneira isolada. No segundo experimento, a base de dados de todos os semestres (2017/1 e 2018/1) foi inserida para analisar a capacidade de generalização do classificador. Cada um desses experimentos foi executado com e sem a nota do exame. Para guiar a exposição, serão utilizadas as três perguntas de pesquisa apresentadas na Introdução.

5.1. É possível prever, com acurácia acima de 80%, a aprovação de um aluno levando em conta apenas atributos comportamentais?

O classificador Random Forest aplicado neste trabalho é capaz de inferir o desempenho final de um estudante com acurácia acima de 80% a partir de dados referentes aos três primeiros exames. No experimento 1 a maior acurácia foi de 91% com os dados dos seis exames, o experimento 2 obteve seu melhor resultado de 88% também no sexto exame. Esse fato já era esperado, já que o desempenho final do aluno está relacionado ao desempenho das médias dos exames.

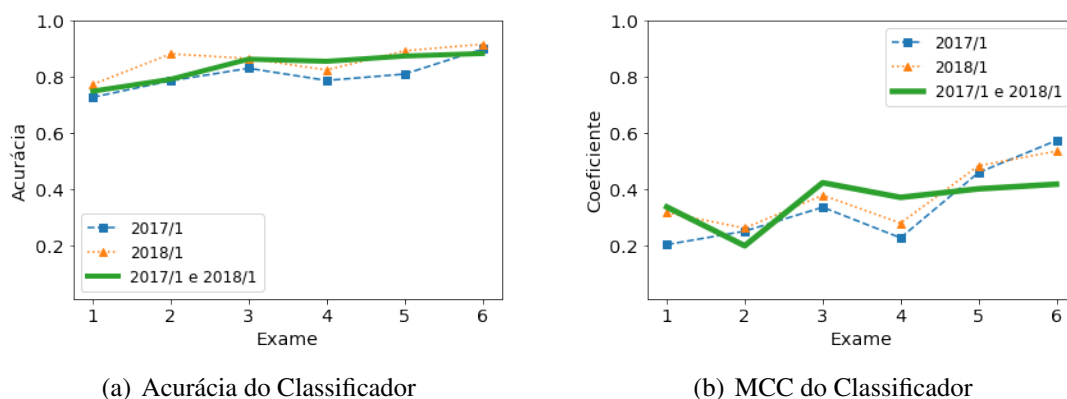


Figura 1. Comparativo de desempenho do classificadores nos dois experimentos

Comparando-se os gráficos da Figura 1, observou-se que:

- O motivo da variação na acurácia dos classificadores do experimento 1 provavelmente é em decorrência do distanciamento dos valores das variáveis preditoras. Esse tipo de variação está relacionada a quantidade de dados dos datasets pois durante o experimento 2 o mesmo problema não ocorreu;
- Tanto a acurácia quanto o MCC do classificador tem uma queda de desempenho no exame 4. Isso é devido a uma distribuição mais extensa dos atributos, provavelmente em decorrência da dificuldade do exame. Isso atrapalha a compreensão que o classificador tem sobre os dados;

- Por mais que possua uma acurácia final menor, o desempenho do classificador no experimento 2 é mais estável que o do experimento 1. Isso é um indicativo de maior capacidade de generalização.

5.2. Qual o impacto que o atributo nota tem sobre o classificador?

Ao integrar-se a nota ao conjunto de variáveis preditivas, o desempenho do classificador aumentou em até 12 pontos percentuais na acurácia e 0,20 no valor final do MCC. O atributo nota é composto pelo somatório de exames que um aluno tirou acima de 5,0 até o exame em análise. Foi-se testado o atributo nota como a média dos exames, entretanto, o resultado foi inferior ao obtido com o somatório.

O experimento 2 tem o classificador com maior capacidade de generalização, por esse motivo ele será utilizado para demonstrar o impacto que a nota tem sobre o classificador. A Figura 2 compara o desempenho desse classificador com e sem a nota como uma das variáveis predictoras. Essa comparação é realizada no mesmo subconjunto (*fold*) e seguindo os mesmos princípios apresentados na Metodologia.

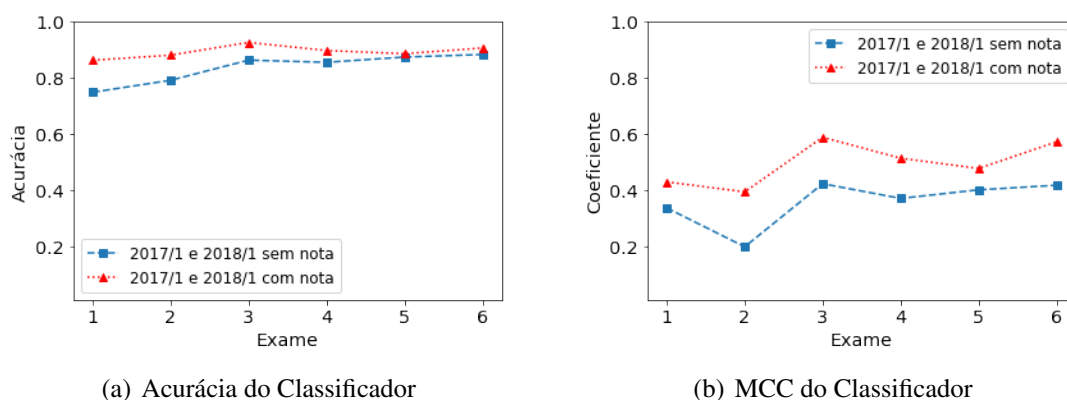


Figura 2. Comparativo de desempenho dos classificadores com e sem o atributo nota

Como o demonstrado por Dwan et al. [2017], o desempenho do classificador é superior quando a nota é usada como entrada. No intervalo dos três primeiros exames a diferença entre o desempenho dos classificadores é notória. Constatou-se que a importância (porcentagem associada a uma variável em função da sua capacidade de discretização) da variável nota durante todo o experimento 2, foi a maior entre todos os atributos. Esse fato garante uma vantagem ao classificador nos primeiros exames graças a baixa dispersão dos dados.

Entretanto, a partir do quarto exame, a diferença entre os classificadores diminui em decorrência dos falsos positivos que o atributo nota insere. Como a nota final é resultado de uma média ponderada, os últimos exames têm um peso maior e o atributo nota representa apenas o total de exames que o estudante foi aprovado, e não o quanto cada exame influencia no desempenho final. Como o classificador sem nota não recebe essa variável ele atribui maior importância para variáveis comportamentais que não o levam a cometer esse tipo de erro.

5.3. Como a importância dos atributos para o classificador varia ao longo da disciplina?

Como citado previamente, o classificador atribui a cada variável uma porcentagem chamada importância. A cada exame, a importância associada às variáveis preditivas muda proporcionalmente ao poder de discretização dos atributos durante aquela interação. Esse fato permite formular hipóteses a respeito do comportamento dos dados visando compreender com mais profundidade o comportamento dos dados.

A Figura 3 apresenta a variação da importância das variáveis **média_testes** e **média_linhas_log** durante o experimento 2. Estes atributos foram selecionados por serem os com maior diferença entre os valores com e sem nota ao longo das interações.

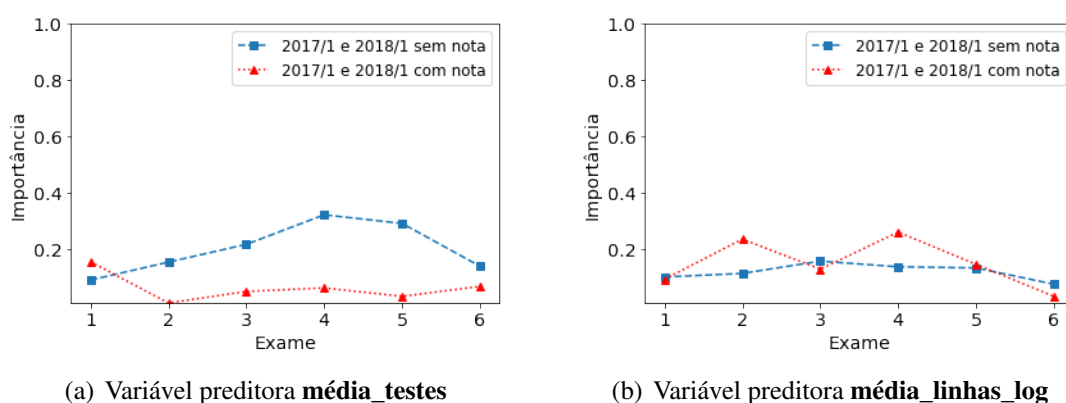


Figura 3. Comparativo da importância associada pelo classificador quando a nota está entre as variáveis preditoras

A respeito da Figura 3 observou-se que:

- A variável **média_testes** tem um alto nível de discretização, se tornando assim, um possível alvo de estudo a fim de compreender sua distribuição e derivar novos atributos com potencial ainda maior de discretização. Entretanto quando a variável **nota** compõe o conjunto de entradas, a sua importância inviabiliza o ganho de informação da **média_testes**;
- O comportamento da variável **média_linhas_log** é único entre todos os atributos. Este é, exclusivamente, o atributo que aumentou de importância quando a **nota** estava entre as variáveis de entrada. Isso aponta para uma variável que possui correlação positiva com o atributo **nota**, se mostrando valiosa para estudos que utilizem atributos comportamentais com nota para prever desempenho final.

6. Considerações Finais e Trabalhos Futuros

A análise de atributos comportamentais de estudantes em juízes online é uma linha promissora e que necessita ser mais explorada. Informações referentes ao tipo de comportamento ou hábito de estudo que levam um aluno a ser aprovado podem aprimorar ambientes que utilizam ferramentas de gamificação, pois fica evidente quais comportamentos devem ser encorajados ou desencorajados.

O classificador proposto neste trabalho, que utiliza apenas atributos comportamentais, obteve uma acurácia superior a 80% a partir do terceiro exame. Além disso, o fato

da diferença final entre os classificadores com nota e sem nota ser de apenas três pontos percentuais atesta o potencial que este tipo de análise apresenta. Do ponto de vista pedagógico, disponibilizar tanto para o professor quanto para o aluno inferências a respeito do desempenho final durante as etapas iniciais do curso pode auxiliar a reduzir os índices de reprovação em turmas de CS1.

Futuras pesquisas poderiam aplicar a metodologia utilizada para modelagem dos dados deste trabalho em juízes online semelhantes, a fim de compreender especificidades e as vantagens que atributos únicos de um ACAC poderiam apresentar ao classificador. Além disso espera-se que caso a nota final da disciplina seja uma média aritmética simples, o desempenho do classificador nesse contexto obtenha resultados melhores que os coletados neste trabalho.

Agradecimentos

Os autores agradecem ao apoio prestado pela Fundação de Amparo à Pesquisa do Estado do Amazonas - FAPEAM por meio do Edital N. 002/2018 - Universal Amazonas.

Referências

- Baker, R. (2010). Data mining for education. *International encyclopedia of education*, 7(3):112–118.
- Baker, R., Isotani, S., and Carvalho, A. (2011). Mineração de dados educacionais: Oportunidades para o brasil. *Brazilian Journal of Computers in Education*, 19(02):03.
- Costa, E., Baker, R. S., Amorim, L., Magalhães, J., and Marinho, T. (2013). Mineração de dados educacionais: conceitos, técnicas, ferramentas e aplicações. *Jornada de Atualização em Informática na Educação*, 1(1):1–29.
- Dwan, F., Oliveira, E., and Fernandes, D. (2017). Predição de zona de aprendizagem de alunos de introdução à programação em ambientes de correção automática de código. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 28, page 1507.
- Galvão, L., Fernandes, D., and Gadelha, B. (2016). Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 140.
- Gottardo, E., Kaestner, C., and Noronha, R. V. (2012). Previsão de desempenho de estudantes em cursos ead utilizando mineração de dados: uma estratégia baseada em séries temporais. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 23.
- Wasik, S., Antczak, M., Badura, J., Laskowski, A., and Sternal, T. (2018). A survey on online judge systems and their applications. *ACM Computing Surveys (CSUR)*, 51(1):3.
- Yin, J., Zheng, Z., and Cao, L. (2012). Uspan: an efficient algorithm for mining high utility sequential patterns. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 660–668. ACM.