CBIE 2023
CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO
Uma escola para o futuro: Tecnologia e conectividade a serviço da educação

Previsão de Reprovações em Disciplinas Introdutórias de Programação: Um Estudo em um Ambiente de Correção Automática de Códigos

Carlos Eduardo Paulino Silva[1,2], **João Lucas Silva Solano**[1], **André Gustavo dos Santos**[1], **Julio Cesar Soares dos Reis**[1]

[1] Departamento de Informática - Universidade Federal de Viçosa (UFV)
Campus Universitário – 36.570-900 – Viçosa – MG – Brasil

[2]Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG)
Rua Afonso Sardinha, 90, Minas Talco – 36.494-018 – Ouro Branco – MG – Brasil

{carlos.e.silva,joao.l.solano,andresantos,jreis}@ufv.br

Abstract. *This paper describes the use of machine learning techniques applied to the educational context to predict, as soon as possible, students with a trend to fail in an introductory computer programming course in a Computer Science course. This prediction may allow the professor, through special attention to these students, to reverse this trend. To achieve this goal, only the interaction data of the students with an automatic code correction environment were used during the resolution of exercises in the course's practical classes. The best result model had an accuracy of 80.00% and a recall of 77.78% using data of 6 weeks of classes out of a total of 12.*

Resumo. *Este artigo descreve o uso de técnicas de aprendizagem de máquina aplicadas ao contexto educacional para prever, o quanto antes, os alunos com tendência de reprovação em uma disciplina introdutória de programação em um curso superior de Ciência da Computação. Essa previsão pode permitir ao professor, através de uma atenção especial para esses alunos, reverter essa tendência. Para atingir esse objetivo, foram utilizados dados de interação dos alunos com um ambiente de correção automática de códigos durante a resolução de exercícios nas aulas práticas da disciplina. A modelagem que obteve o melhor resultado teve uma acurácia de 80,00% e uma revocação de 77,78% com dados de 6 semanas de aulas de um total de 12.*

1. Introdução

Programação de computadores é uma das mais importantes disciplinas nos cursos de computação, pois, além do seu conteúdo ser usado por diversas outras disciplinas ao longo do curso, também é um requisito primordial para as vagas de emprego no mercado de trabalho de Tecnologia da Informação (TI) [Assyne et al. 2022]. Porém, normalmente é uma disciplina que possui uma alta taxa de reprovação [Bennedsen and Caspersen 2007]. Enquanto alguns alunos não possuem experiência alguma, outros possuem conhecimentos básicos e até mesmo avançados, adquiridos por curiosidade e auto-aprendizado ou em um curso formal, como por exemplo, o Técnico de Informática ou afins realizado durante o Ensino Médio. Nesse cenário, é crucial que o professor da disciplina tenha uma

metodologia de ensino diferenciada para cada um desses perfis de aluno, induzindo um aprendizado personalizado e significativo, independente da experiência prévia do aluno. Dessa forma, a disciplina atenderá as expectativas de aprendizado dos diversos perfis de alunos e as necessidades de ensino do professor. Entretanto, implantar essa metodologia na prática é um grande desafio.

De acordo com os estudos apresentados em [Medeiros et al. 2019] e [Becker and Quille 2019], aprender programação de computadores exige muita prática por parte do aluno, especialmente, através do desenvolvimento de uma grande quantidade de exercícios sobre os mais diversos tópicos envoltos na disciplina. Porém, a realização deste grande volume de exercícios pode ser entediante e/ou desmotivadora para o aluno se não houver nenhum retorno quanto a correção das atividades realizadas. Por parte do professor, a correção manual desses exercícios, além de ser impeditivo em algumas situações devido à quantidade de alunos, pode ser demorada e desgastante [Bez et al. 2013, Pereira et al. 2020]. Por outro lado, esta é uma tarefa essencial uma vez que permite ao professor acompanhar o desempenho dos alunos, percebendo assim quais alunos necessitam de uma atenção especial, propondo por exemplo, a participação desses alunos em monitorias e atividades de reforço de aprendizado usando outras metodologias de ensino. Através dessas ações, os alunos que possuem uma tendência de reprovação na disciplina podem ser identificados em tempo hábil para uma potencial reversão do cenário.

Na prática, essa necessidade de atenção especial pode ser mensurada por meio de indícios que o próprio aluno fornece, mas que nem sempre são facilmente perceptíveis para o professor numa turma com muitos alunos, como por exemplo, várias tentativas para tentar resolver um exercício, exercícios sem tentativa de resolução, entre outros. É neste contexto que insere-se o objetivo deste trabalho. Especialmente, abordamos o problema de identificação prévia, o quanto antes, do aluno com tendência de reprovação baseado no desenvolvimento dos exercícios propostos pelo professor, em aulas práticas ao longo de uma disciplina introdutória de programação de um curso superior em computação.

Neste estudo, consideramos uma disciplina introdutória de programação de computadores que usa um Ambiente de Correção Automática de Códigos (ACACs), do qual podem ser extraídas diversas métricas de desempenho dos alunos. Os ACACs fazem correção automática de códigos submetidos para solução de exercícios previamente cadastrados. Eles podem ser classificados em dois tipos: *(i)* juízes *online* de propósito geral e *(ii)* juízes dedicados. Os juízes *online* de propósito geral possuem milhares de questões cadastradas, de diferentes níveis e origens, que podem ser usadas pelos próprios alunos como ambiente de treinamento extraclasse. Exemplos desse tipo de ACACs incluem o Beecrowd[1], o Online Judge[2], entre outros. Já os juízes dedicados podem ser instalados e gerenciados localmente pelos professores, com fácil cadastro de exercícios próprios e personalizados para a disciplina ministrada. Exemplos desse tipo de ACACs incluem o BOCA [de Campos and Ferreira 2004] e o CodeBench[3] [Galvão et al. 2016]. Os ACACs do tipo juízes dedicados vêm sendo largamente utilizados nas disciplinas de programação, favorecendo assim a escalabilidade do trabalho de correção de exercícios dos professores

[1] `https://www.beecrowd.com.br`
[2] `https://onlinejudge.org`
[3] `https://codebench.icomp.ufam.edu.br`

[Bez et al. 2013, Galvão et al. 2016, Francisco et al. 2018, Pereira et al. 2020].

Os ACACs contribuem bastante nas aulas práticas das disciplinas introdutórias de programação, pois os alunos com conhecimento prévio ou que aprendem mais rapidamente, conseguem terminar os exercícios e saber que estão corretos ou não sem (ou com pouca) necessidade de auxílio do professor. Isso também permite ao professor prestar um atendimento mais direcionado e por mais tempo aos alunos com mais dificuldade.

Como estratégia para identificar, o quanto antes, alunos com tendência de reprovação em uma disciplina introdutória de programação, propomos a utilização de técnicas de aprendizagem de máquina aplicadas ao contexto educacional, sobre os dados gerados exclusivamente pela interação dos alunos nos ACACs durante a resolução de exercícios propostos nas aulas práticas da disciplina. Tais dados envolvem por exemplo, a média de soluções corretas e incorretas submetidas pelos alunos, quantidade de tentativas de solução, entre outras. Em suma, os resultados revelam que as abordagens propostas apresentam um potencial interessante para suporte aos professores neste contexto.

Este artigo está organizado conforme detalhado a seguir. A Seção 2 descreve os principais trabalhos relacionados. A Seção 3 descreve a metodologia proposta para o desenvolvimento do trabalho e a Seção 4 a análise dos resultados alcançados. As conclusões e trabalhos futuros, além das ameaças à validade do trabalho, estão descritas na Seção 5.

2. Trabalhos relacionados

As técnicas de aprendizagem de máquina tem sido aplicadas aos dados, cada vez mais abundantes, oriundos de diversas abordagens educacionais, que demandam de forma crescente análises mais detalhadas visando um melhor planejamento e execução de ações na área da educação [Silva et al. 2017].Uma vez que nosso trabalho é focado no uso dos dados do ACACs, um tipo de abordagem educacional, utilizado em uma disciplina introdutória de programação de computadores para prever, o quanto antes, reprovações, na sequência serão relatados alguns trabalhos relacionados a essa temática e/ou contexto.

O trabalho apresentado em [Silva et al. 2020], tem como objetivo prever a evasão de alunos em disciplinas introdutórias de programação com base nos dados de utilização do ACACs CodeBench. Para isso, os autores utilizaram diferentes modelos de aprendizado de máquina sobre a base de dados do ACACs, analisando, a partir de 38 atributos, o comportamento e o desempenho em provas e exercícios de 2010 alunos dos anos 2016 a 2019. Por fim, após análise dos 5 algoritmos utilizados no experimento, o *Extra Tree Classifier*, baseado em árvores de decisão, teve o melhor desempenho médio dentre os analisados. O trabalho em questão se mostrou eficiente na previsão de evasão das disciplinas ao alcançar uma acurácia média de 91,96% nas seis primeiras semanas de aula.

Já no trabalho apresentado em [Dwan et al. 2017], foi desenvolvido um modelo de previsão de aprovação em disciplinas introdutórias de programação utilizando técnicas de aprendizagem de máquina. Esse modelo obteve uma acurácia de 74,70% nas seis primeiras semanas de aula em uma base de dados balanceada. Nesse contexto, foram usados atributos relacionados à quantidade de submissões e erros que as soluções dos exercícios produziram quando submetidas ao ACACs CodeBench, além de métricas do código-fonte dessas soluções, do ambiente de desenvolvimento integrado (IDE), notas em atividades avaliativas e retorno dos alunos sobre o grau de dificuldade dos exercícios.

Também utilizando os dados de um ACACs, o trabalho desenvolvido em [Rico-Juan et al. 2023], realiza uma classificação do perfil dos alunos. No referente trabalho, diferentes modelos de aprendizagem foram comparados na previsão de desempenho em atividades acadêmicas, explorando, por exemplo, os seguintes atributos: *(i)* número de dias até o prazo final da tarefa, *(ii)* número de dias até o prazo final a partir da primeira submissão, *(iii)* número de submissões feitas, *(iv)* número de diferentes dias que houveram submissões e *(v)* se o aluno foi aprovado ou não na tarefa. A métrica da área sob a curva (AUC) é utilizada neste trabalho para mensurar o desempenho do modelo na distinção entre as classes positiva e negativa, chegando em um resultado final de 0,7, considerado satisfatório para a tarefa em questão.

Por fim, o trabalho desenvolvido em [Toledo and Martínez-López 2017] utiliza as interações dos alunos com os ACACs para identificar o esforço (falhas e acertos) ao tentarem resolver um determinado exercício e assim recomendar conteúdos, disponíveis em sistemas de *e-learning*, relacionados ao(s) tópico(s) da disciplina presente(s) nos exercícios que exigiam um maior esforço de um determinado aluno no desenvolvimento da sua solução.

Em resumo, nosso trabalho é complementar aos anteriores uma vez que estamos interessados na tarefa de previsão de reprovação de alunos em disciplinas introdutórias de programação. No entanto, exploramos dados genéricos e independentes que são fornecidos por qualquer ACACs (e.g., *porcentagem de soluções corretas e incorretas submetidas pelos alunos por aula prática para cada exercício*), ao contrário de [Dwan et al. 2017] e [Silva et al. 2020], o que favorece aspectos de generalização do estudo. Além disso, propusemos novos atributos, inspirados em trabalhos anteriores [Toledo and Martínez-López 2017, Rico-Juan et al. 2023], que nos permitem investigar o potencial de diferentes modelagens, testadas em conjunto ou separadamente, na tarefa de interesse.

3. Metodologia

Nesta seção apresentamos a metodologia adotada no desenvolvimento deste trabalho, detalhando os dados utilizados e, posteriormente, as modelagens realizadas.

3.1. Dados

Para este trabalho, foram utilizados os dados de interações dos alunos no ACACs BOCA, em uma disciplina introdutória de programação de um curso superior em computação. No total, foram usados os dados de quatro turmas, de anos diferentes, conforme detalhado na Tabela 1. O número de submissões de soluções realizadas em cada turma foi entre 4 e 5 por aluno por aula.

Tabela 1. Detalhamento das turmas.

Turma	Alunos	Aulas práticas	Submissões
1	55	10	2452
2	60	12	2886
3	59	13	3514
4	55	12	2900

Foram considerados apenas os dados de aulas práticas "regulares" aplicadas durante o período letivo. O termo "regulares" implica que as aulas tinham duração em torno de 110 minutos e possuíam, em média, pelo menos uma submissão por aluno. Nessas aulas o aluno é apresentado a uma lista de exercícios contendo, em média, 5 exercícios baseados nos conhecimentos teóricos lecionados previamente nas aulas teóricas da disciplina. Esses dados foram coletados diretamente do banco de dados do ACACs usando um *script* em Python e a linguagem SQL. Além disso, é importante mencionar que os dados dos alunos foram usados de forma anonimizada, além de não ter sido usado nenhum dado sensível, apenas a interação com o ACACs.

A partir dessa base de dados foram computados 16 atributos, sendo todos eles numéricos com exceção do atributo *identificador do aluno* que é categórico (i.e., código textual aleatório). Os atributos são descritos a seguir, sendo os 15 primeiros gerados a partir da coleta e agregação estatística dos dados de interação do aluno ao submeter sua solução para um determinado exercício de programação em um ACACs, e o último, alvo da predição do modelo, extraído do sistema acadêmico utilizado na universidade alvo do estudo.

- Identificador randomizado do aluno;
- Identificador da aula prática;
- Porcentagem de problemas aceitos por aula prática;
- Porcentagem de problemas que tiveram alguma tentativa por aula prática;
- Número de submissões em problemas aceitos por aula prática;
- Porcentagem de problemas sem submissões por aula prática;
- Porcentagem de problemas que tiveram poucas tentativas (menor que oito[4]) e não foram aceitos por aula prática;
- Porcentagem de problemas que tiveram muitas tentativas e não foram aceitos por aula prática;
- Porcentagem de problemas que tiveram poucas tentativas e foram aceitos por aula prática;
- Porcentagem de problemas que tiveram muitas tentativas e foram aceitos por aula prática;
- Porcentagem de submissões com resposta incorreta por aula prática;
- Porcentagem de submissões com erro de compilação por aula prática;
- Porcentagem de submissões com erro de formatação de saída por aula prática;
- Porcentagem de submissões com erro de execução por aula prática;
- Porcentagem de submissões com o tempo limite excedido[5] por aula prática;
- Reprovado: alvo da predição do modelo extraído manualmente pelo orientador do projeto do sistema acadêmico, já que os dados dos alunos foram anonimizados.

Os dados de todos os atributos, com exceção do *identificador do aluno*, *identificador da aula prática* e *reprovado*, foram normalizados usando o *Standard Scaler* da *Scikit-learn*[6], uma biblioteca de código aberto para a linguagem de programação Python, que, de modo geral, oferece uma ampla gama de ferramentas para tarefas de aprendizado de máquina.

[4]Segundo [Toledo and Martínez-López 2017]; assim, muitas tentativas correspondem 8 ou mais.

[5]Programa sem resposta durante tempo pré-estipulado; provavelmente entrou em *loop*.

[6]`https://scikit-learn.org/`

3.2. Modelagem dos Dados

Foram criadas 4 propostas para a modelagem dos dados (MD), descritas a seguir:

- **MD 1:** média final de todas as aulas práticas de cada atributo por aluno, com exceção do identificador da aula prática;
- **MD 2:** porcentagem por aula prática de cada atributo por aluno;
- **MD 3:** média acumulada por aula prática de cada atributo por aluno (ou seja, em cada aula prática *i* foram considerados os dados gerados das aulas 1, 2, ... *i*);
- **MD 4:** utilização da MD 3, porém com um modelo treinado e testado para cada aula prática, resultando assim em um conjunto de modelos, em vez de um único, como na MD 3.

Em todas as MDs foram usados os dados das turmas 1, 2 e 3 para a etapa de treinamento e na etapa de teste os dados da turma 4, que teve um total de 12 aulas práticas. Além disso, a base de dados utilizada (para treino) em todos os modelos foi balanceada, de forma randômica, diminuindo o conjunto de dados da classe majoritária (*under-sampling*[7]), nesse contexto os alunos com situação final "não reprovado" no atributo alvo do modelo.

No trabalho apresentado em [Dwan et al. 2017], ao estudar o problema de previsão de aprovação em contexto semelhante ao deste trabalho, foi concluído que os algoritmos baseados em árvores de decisão, também utilizados nos trabalhos de [Silva et al. 2020] e [Rico-Juan et al. 2023], mostram os melhores resultados em base de dados de ACACs. Portanto, no presente trabalho, o algoritmo utilizado em todos os modelos foi o *Random Forest Classifier*, que é um dos algoritmos baseados em árvores de decisão da biblioteca de aprendizado de máquina *Scikit-learn*. Os hiperparâmetros utilizados pelo algoritmo, listados na Tabela 2, foram definidos através do *Grid Search* com validação cruzada também da *Scikit-learn*, usando os dados das turmas 1, 2 e 3, de acordo com o intervalo de valores configurados. Os dados da turma 4 foram separados para avaliação do desempenho dos modelos gerados a partir das modelagens propostas.

Tabela 2. Hiperparâmetros utilizados pelo algoritmo *Random Forest Classifier*.

Hiperparâmetro	Valor definido	Valores configurados
n_estimators	20	20 a 200
min_samples_split	2	2, 5 e 10
min_samples_leaf	1	1, 2 e 4
max_depth	9	1 a 20
n_jobs	-1	Não se aplica

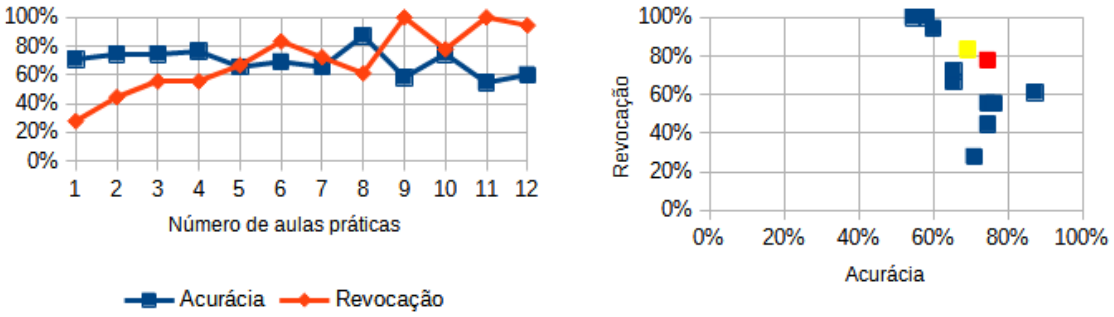
Para comparação das modelagens são utilizadas as métricas de acurácia e revocação. A acurácia é utilizada para verificar a porcentagem de alunos reprovados (classe positiva) que o modelo acertou e a revocação, o número de alunos reprovados não identificados. Esta métrica é utilizada devido ser bastante crítico o fato do aluno reprovado não ter sido previsto pelo modelo. Uma vez que o tempo é um fator importante neste contexto, verificamos também a quantidade de aulas práticas necessárias para alcançar os melhores valores de acurácia e revocação a tempo de ainda ser possível reverter a tendência de reprovação de um aluno. Portanto, definimos que cerca de 6 – 8

[7] https://imbalanced-learn.org/stable/under_sampling.html

semanas de aula é o limite máximo para a identificação desse aluno, uma vez que ainda restariam algumas semanas de aula (i.e., 4 – 6), para realização de alguma intervenção que porventura se torne necessária. Logo, tanto a acurácia quanto a revocação devem ter valores altos e com um bom compromisso entre elas o quanto antes e não após a 8ª semana de aulas práticas.

4. Análise dos Resultados

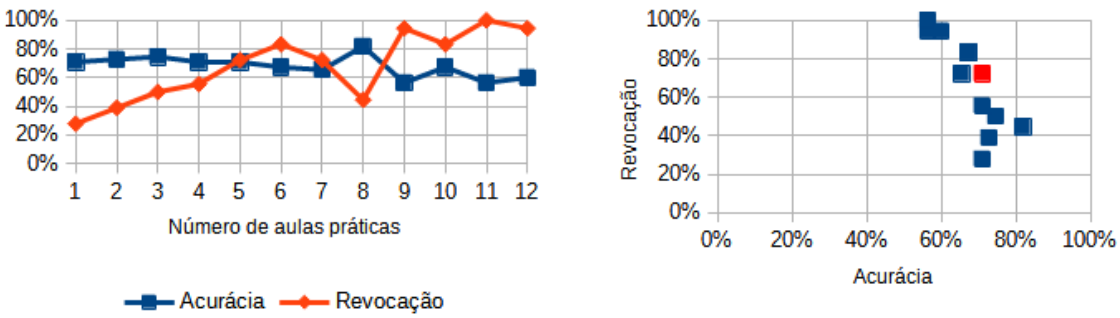
Os resultados referentes às métricas de acurácia e revocação de cada um dos modelos variam de acordo com as 12 aulas práticas da turma 4, correspondentes a 12 semanas de aula, como podem ser observados nas Figuras 1, 2, 3 e 4, respectivamente. Em cada figura são mostradas a variação das métricas ao longo das semanas (gráfico da esquerda, referenciado como `(a)` nas Figuras) e a combinação das métricas (gráfico da direita, referenciado como `(b)`) destacando as semanas com melhores resultados, no quadrante superior mais à direita.



(a) Variação das métricas **(b) Combinação das métricas**

Figura 1. Acurácia e revocação da MD 1.

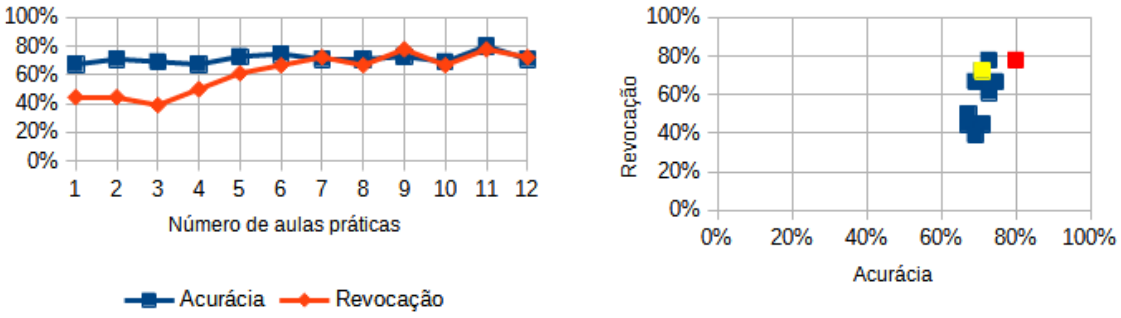
Na Figura 1a, é possível observar que o melhor valor de acurácia (87,27%) da MD 1, referente a média final de todas as aulas práticas de cada atributo por aluno, acontece na 8ª semana de aulas. Porém, nessa mesma semana ocorre um baixo valor de revocação (61,11%). O melhor resultado combinado acontece na 10ª semana de aulas (ponto vermelho da Figura 1b). No entanto, perceba que este resultado está acima do limite máximo de semanas de aulas definido, que são 8 semanas. Portanto, podemos observar que o melhor resultado prático dessa modelagem é alcançado na 6ª semana (ponto amarelo da Figura 1b), que é o próximo melhor valor combinado de acurácia (69,09%) e revocação (83,33%).



(a) Variação das métricas **(b) Combinação das métricas**

Figura 2. Acurácia e revocação da MD 2.

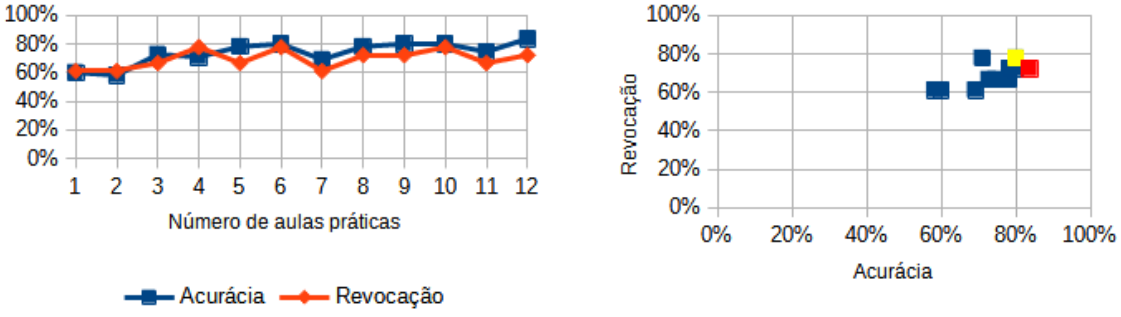
Na Figura 2a, é possível observar que o melhor valor de acurácia (81,82%) da MD 2, referente a porcentagem por aula prática de cada atributo por aluno, acontece na 8ª semana de aulas, porém, nessa mesma semana ocorre um dos menores valores de revocação (44,44%). Esse fato pode ser explicado por uma alta taxa de ausência de alunos na aula dessa semana comprometendo assim a agregação estatística dos dados. O melhor resultado combinado de acurácia (70,91%) e revocação (72,22%) acontece na 5ª semana de aulas, destacada em vermelho na Figura 2b.



(a) Variação das métricas **(b) Combinação das métricas**

Figura 3. Acurácia e revocação da MD 3.

Na Figura 3a é possível observar que o melhor valor combinado de acurácia (80,00%) e revocação (77,78%) para a MD 3, referente a média acumulada por aula prática de cada atributo por aluno, acontece na 11ª semana de aulas (ponto vermelho na Figura 3b). Entretanto, considerando a necessidade de resposta antecipada dentro do nosso limite máximo de 8 semanas de aulas, o melhor combinado de acurácia (70,91%) e revocação (72,22%) acontece na 7ª semana de aulas (ponto amarelo na Figura 3b). Podemos observar também que entre o ponto amarelo e o ponto vermelho na Figura 3b existe um ponto azul com 72,73% de acurácia e 77,78% de revocação, porém, esses valores ocorrem na 9ª semana de aulas, por isso, não foi escolhido.



(a) Variação das métricas **(b) Combinação das métricas**

Figura 4. Acurácia e revocação da MD 4.

Por fim, na Figura 4a, é possível observar que os melhores valores combinados de acurácia (80,00%) e revocação (77,78%) para a MD 4, referente a média acumulada por aula prática de cada atributo por aluno com o treino e teste do modelo sendo realizado para cada aula prática, acontecem na 6ª e 10ª semanas de aulas (sobrepostas no ponto amarelo na Figura 4b). A 12ª semana (ponto vermelho na Figura 4b) tem alta acurácia mas menor revocação. Priorizando resposta antecipada dentro do limite máximo de 8 semanas de aulas, consideramos que o melhor resultado desta modelagem acontece na 6ª semana de aulas.

Interessante observar que, nas modelagens 1 e 2, os valores de acurácia e revocação são muito discrepantes e oscilantes ao longo das semanas de aulas. Esse cenário ocorre na modelagem 3 de forma mais localizada, apenas na primeira metade das semanas de aulas. Já na modelagem 4 essa discrepância é praticamente inexistente. Isso pode ser explicado pela diminuição de frequência dos alunos nas últimas aulas práticas, o que prejudica o cálculo da porcentagem dos atributos, em especial nas modelagens 1 e 2, onde é usada a média final de todas as aulas práticas e de cada aula prática, respectivamente, ao contrário das modelagens 3 e 4, onde é usada a média parcial por aula prática. Além disso, como na modelagem 4 é criado um modelo para cada aula prática, os modelos são muito bem ajustados para cada contexto de aula, tornando assim significamente menor a discrepância entre os valores de acurácia e revocação. Em resumo, o melhor resultado gerado por cada modelagem, considerando a acurácia, a revocação e o número de aulas práticas (1 aula prática por semana) necessárias, estão descritos na Tabela 3.

Tabela 3. Melhores resultados das modelagens.

Modelagem	Acurácia	Revocação	Aulas práticas
1	69,09%	83,33%	6
2	70,91%	72,22%	5
3	70,91%	72,22%	7
4	**80,00%**	**77,78%**	**6**

Analisando os melhores resultados das modelagens na Tabela 3, constatamos que a modelagem 4 possui resultados promissores, pois, na metade das aulas práticas totais da disciplina, consegue obter uma sinalização dos alunos com tendência de reprovação com uma acurácia de 80,00%, em um tempo semelhante de aulas ao obtido no trabalho apresentado em [Dwan et al. 2017], 6 semanas. Em relação ao trabalho apresentado em [Dwan et al. 2017], é importante ressaltar que a limitação da modelagem 4 é que são necessários 12 modelos, 1 para cada aula prática. Entretanto, são utilizados exclusivamente dados do ACACs, sem nenhum dado extra, o que pode permitir uma maior generalização dos modelos criados para serem aplicados em diversos contextos.

Após a escolha da modelagem 4, os modelos gerados foram persistidos e consumidos através da *interface* do ACACs para sinalizar ao professor quais são os alunos com tendência de reprovação (marcados com !), conforme pode ser observado na Figura 5, a qual utiliza dados fictícios. Em uma análise preliminar, constatou-se que os alunos sinalizados com tendência de reprovação normalmente apresentam notas baixas em atividades avaliativas individuais da disciplina, o que é um indício forte da concretização da sua tendência de reprovação no final da disciplina.

5. Conclusão e Trabalhos Futuros

Neste trabalho tivemos como objetivo, com o auxílio de técnicas de aprendizagem de máquina aplicadas ao contexto educacional, identificar, o quanto antes, os alunos que apresentam tendência de reprovação em disciplinas introdutórias de programação que utilizam qualquer ACACs, como uma abordagem educacional, para auxiliar no processo de ensino-aprendizagem. Os resultados alcançados demonstram que a hipótese de utilização dos dados gerados exclusivamente pela interação dos alunos no ACACs, durante a resolução de exercícios propostos nas aulas práticas de uma disciplina introdutória de programação, para uma previsão antecipada e automática de alunos com tendência de

Figura 5. *Interface* do professor no ACACs BOCA com a sinalização dos alunos com tendência de reprovação.

reprovação é verdadeira. Portanto, a partir dessa proposta o professor sabe, de forma rápida e objetiva, quais alunos precisam de uma atenção especial ou até mesmo de uma metodologia de ensino diferenciada para reverter a tendência de reprovação na disciplina.

Como trabalhos futuros pretendemos enriquecer os dados utilizados nos modelos gerados agregando os dados do perfil do aluno, como por exemplo, se já teve contato com programação antes de iniciar o curso, participou de olimpíadas de conhecimento e, também, dados do Censo do Instituto Brasileiro de Geografia e Estatística (IBGE) referentes a cidade/região de origem do aluno, por exemplo. Além disso, pretendemos realizar experimentos mais exaustivos, utilizando inclusive, outros algoritmos de aprendizado de máquina e/ou outros valores de hiperparâmetros para uma comparação mais abrangente dos resultados.

Identificamos como ameaça a validade do trabalho, a necessidade de entender melhor o contexto dos dados utilizados, pois, seria importante que a quantidade de aulas práticas em todas as turmas fossem iguais, além dos tópicos do conteúdo da disciplina terem a mesma ordem cronológica e as aulas práticas terem a presença regular dos alunos. Essa ameaça fica explícita ao analisarmos os resultados alcançados pelos modelos propostos, onde é possível constatar que a melhoria dos valores de acurácia e revocação não é algo sempre crescente com o passar das aulas práticas, como era de se esperar; o que pode ser explicado pela ausência intencional de alunos já aprovados ou já reprovados nas últimas aulas práticas, por exemplo, enviesando os dados utilizados para geração dos modelos.

Agradecimentos

Os autores agradecem ao Sicoob-UFVCredi, ao Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG) e a Universidade Federal de Viçosa (UFV) pelos recursos disponibilizados para a execução do projeto que gerou esse artigo e a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que possibilitou a sua apresentação e publicação através de recursos do Programa de Apoio à Pós-Graduação (PROAP).

Referências

Assyne, N., Ghanbari, H., and Pulkkinen, M. (2022). The state of research on software engineering competencies: A systematic mapping study. *Journal of Systems and Soft-*

ware, 185:111183.

Becker, B. A. and Quille, K. (2019). 50 years of cs1 at sigcse: A review of the evolution of introductory programming education research. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 338–344, New York, NY, USA. Association for Computing Machinery.

Bennedsen, J. and Caspersen, M. (2007). Failure rates in introductory programming. *SIGCSE Bulletin*, 39:32–36.

Bez, J. L., Ferreira, C. E., and Tonin, N. (2013). Uri online judge academic: A tool for professors. In *Proceedings of the 2013 International Conference on Advanced ICT and Education*, pages 744–747. Atlantis Press.

de Campos, C. and Ferreira, C. (2004). Boca: um sistema de apoio a competições de programação. In *Workshop de Educação em Computação*. Sociedade Brasileira de Computação.

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. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 28(1):1507.

Francisco, R., Ambrósio, A. P., Junior, C. X., and Fernandes, M. (2018). Juiz online no ensino de cs1 - lições aprendidas e proposta de uma ferramenta. *Revista Brasileira de Informática na Educação*, 26(03):163.

Galvão, L., Fernandes, D., and Gadelha, B. (2016). Juiz online como ferramenta de apoio a uma metodologia de ensino híbrido em programação. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 27(1):140.

Medeiros, R. P., Ramalho, G. L., and Falcão, T. P. (2019). A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2):77–90.

Pereira, F. D., Oliveira, E. H. T., Oliveira, D. B. F., Cristea, A. I., Carvalho, L. S. G., Fonseca, S. C., Toda, A., and Isotani, S. (2020). Using learning analytics in the amazonas: understanding students' behaviour in introductory programming. *British Journal of Educational Technology*, 51(4):955–972.

Rico-Juan, J. R., Sánchez-Cartagena, V. M., Valero-Mas, J. J., and Gallego, A. J. (2023). Identifying student profiles within online judge systems using explainable artificial intelligence. *IEEE Transactions on Learning Technologies*, pages 1–14.

Silva, D., Tamayo, S., Pessoa, M., Pires, F., Oliveira, D., Oliveira, E., and Carvalho, L. (2020). Minerando dados de um juiz on-line para prever a evasão de estudantes em disciplinas introdutórias de programação. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1343–1352, Porto Alegre, RS, Brasil. SBC.

Silva, L., Silveira, I., Silva, L., Rodrigues, R., and Ramos, J. (2017). Ciência de dados educacionais: definições e convergências entre as áreas de pesquisa. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 6(1):764.

Toledo, R. Y. and Martínez-López, L. (2017). A recommendation approach for programming online judges supported by data preprocessing techniques. *Appl. Intell.*, 47:277–290.