

Previsão de Reprovação e Desempenho em Disciplinas Introdutórias de Programação

João L. S. Solano¹, Barbara C. F. de Souza¹

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

{joao.l.solano, barbaracsouza}@ufv.br

Abstract. *This work focuses on providing support to teachers during practical programming classes in higher education by identifying students with a tendency to fail in the discipline. By monitoring students' individual performance throughout the academic term using the Automatic Code Correction Environment (ACACs) BOCA, we aim to assist teachers in identifying struggling students early on and providing personalized assistance. Leveraging data science techniques, this study aims to optimize the learning process and improve outcomes for programming students.*

Resumo. *Este trabalho tem como foco fornecer suporte aos professores durante as aulas práticas de programação no ensino superior, identificando os alunos com tendência a falhar na disciplina. Através do monitoramento do desempenho individual dos alunos ao longo do período letivo utilizando o Ambiente de Correção Automática de Códigos (ACACs) BOCA, buscamos auxiliar os professores a identificar precocemente os alunos com dificuldades e fornecer assistência personalizada. Utilizando técnicas de ciência de dados, este estudo tem como objetivo otimizar o processo de aprendizagem e melhorar os resultados dos estudantes de programação.*

1. Introdução

Uma das principais disciplinas dos cursos superiores de tecnologia é a programação. Uma vez que essa disciplina fundamenta a base de muitos conceitos e conhecimentos que serão utilizados ao longo de todo o curso, tanto na área acadêmica quanto no mercado de trabalho, ela deve receber atenção à altura de sua importância. Porém, a programação, é uma disciplina que muitas vezes é rotulada como difícil e desafiadora. Isso é explicado pelo fato de que, além de receber alunos de diversas realidades diferentes, exige conhecimentos e experiências que a grande maioria dos alunos não tiveram contato prévio antes de ingressar no Ensino Superior.

Como os alunos vem de realidades diferentes, muitos deles enfrentam grande dificuldade durante as aulas práticas de programação. Nesse cenário, é de grande importância identificar e fornecer auxílio especial à esses alunos, a fim de possam ter um desempenho e aprendizado satisfatório na disciplina. Diante disso, fica explícito que garantir um aprendizado de qualidade e eficaz aos alunos de programação nos cursos superiores é um grande desafio na educação.

Portanto, o objetivo principal desse trabalho é fornecer um auxílio aos professores durante as aulas práticas de programação, identificando, o quanto antes, alunos com

tendência de reprovação na disciplina, com base no desempenho individual de cada aluno nas aulas práticas ao longo do período letivo. Uma vez que o professor tem ciência de quem são aqueles que enfrentam mais dificuldade na disciplina pode fornecer um apoio especial ao estudante.

Nesse trabalho acompanharemos o desempenho dos alunos ao longo da disciplina pelo Ambiente de Correção Automática de Códigos (ACACs) BOCA (Online Contest Administrator), que é utilizado nas aulas práticas de Programação - INF110 (CS1). O ACACs BOCA é utilizado frequentemente nas competições de programação, porém sua habilidade de correção automática de exercícios de programação contribui bastante nas aulas práticas das disciplinas de programação, o que permite ao professor, por agilizar o processo de correção, fornecer um atendimento especial aos estudantes com mais dificuldade.

[Silva et al. 2017] comenta que a ciência de dados educacionais é definida como a exploração de dados com o intuito de aumentar a compreensão e a qualidade do processo de aprendizagem, combinando técnicas, de forma interdisciplinar, da estatística, da computação e da educação. Sendo assim, utilizando técnicas e algoritmos de ciências de dados, pretendemos otimizar o processo citado anteriormente, beneficiando todos os indivíduos ativos no processo de aprendizagem.

2. Trabalhos relacionados

No mesmo contexto, [Silva et al. 2017] utilizaram conceitos de mineração de dados para prever a evasão de estudantes em disciplinas introdutórias de programação baseados no desempenho do aluno capturados durante a utilização do ACACs CodeBench. Os autores aplicaram diferentes modelos de aprendizado de máquina sobre 38 atributos que caracterizam o desempenho de 2010 estudantes dos anos 2016 a 2019 da Universidade Federal do Amazonas (UFMA), analisando o comportamento e o desempenho em provas e exercícios de programação. Por fim, o trabalho em questão se mostrou eficiente na previsão de evasão da disciplina ao alcançar uma acurácia média de 91,96% nas seis primeiras semanas de aula, utilizando o algoritmo Extra Tree Classifier, baseado em árvores de decisão.

Um modelo de previsão de aprovação ou reprovação em disciplinas introdutórias de programação também foi desenvolvido por [Pereira et al. 2017], utilizando técnicas de aprendizagem de máquina. Nessa pesquisa, alcançou-se uma acurácia de 74,7% nas primeiras seis semanas de aula. O modelo em questão utiliza atributos como a quantidade de submissões e erros gerados pelos códigos dos exercícios quando submetidos ao ACACs, assim como métricas do código-fonte capturadas durante o desenvolvimento da solução, bem como notas em atividades e o feedback dos alunos quanto aos exercícios realizados.

3. Metodologia

No desenvolvimento dos modelos de aprendizagem de máquina abordados nesse trabalho foram utilizados os dados de interações dos alunos no ACACs BOCA durante a disciplina de Programação I - INF110 (CS1), disciplina obrigatório do curso de Ciência da Computação, durante os anos de 2017, 2018, 2019 e 2022. Uma vez que, durante os anos de 2021 e 2022, a respectiva disciplina foi ofertada de forma remota, os dados referentes

a esses anos não foram utilizadas durante essa pesquisa, por se passarem em um cenário diferente do habitual.

No conjunto selecionado, consideramos os dados apenas das aulas práticas da disciplina, que tem duração média de 110 minutos. Normalmente, a cada semana uma aula prática é ofertada referente ao conteúdo estudado no período de tempo desde a última aula prática, a fim de reforçar os conceitos vistos previamente em sala de aula.

3.1. Modelagem dos dados

Modelamos os dados das aulas práticas em 16 features que, a partir das interações dos alunos no BOCA, refletem o desempenho ao longo do semestre, por meio de uma média aritmética básica que leva em consideração as informações das aulas práticas anteriores do período. Os 13 atributos extraídos da base de dados do BOCA foram as seguintes:

- **matricula_hash**: matrícula do aluno anonimizada por uma função hash;
- **Porcentagem média de problemas aceitos por contest**;
- **Porcentagem média de problemas que tiveram alguma tentativa por contest**;
- **Média do número de submissões em problemas aceitos**;
- **Porcentagem média de problemas que tiveram poucas tentativas** (menor que 8¹.) e não foram aceitos por contest;
- **Porcentagem média de problemas que tiveram muitas tentativas e não foram aceitos por contest**;
- **Porcentagem média de problemas que tiveram poucas tentativas e foram aceitos por contest**;
- **Porcentagem média de problemas que tiveram muitas tentativas e foram aceitos por contest**;
- **Porcentagem média de submissões como 'Wrong Answer' por contest**: resposta incorreta de acordo com o caso de teste previamente cadastrado para determinado problema;
- **Porcentagem média de submissões como 'Compilation Error' por contest**: código-fonte com erro de compilação;
- **Porcentagem média de submissões como 'Presentation Error' por contest**: resposta com formatação incorreta de acordo com o caso de teste previamente cadastrado para determinado problema;
- **Porcentagem média de submissões como 'Runtime Error' por contest**: código-fonte com erro durante a execução;
- **Porcentagem média de submissões como 'Time Limit Exceeded' por contest**: código-fonte com tempo de execução maior que o previamente cadastrado para determinado problema;
- **Reprovado**: alvo da predição do modelo (0 - Aprovado, 1 - Reprovado).

3.2. Modelos de Aprendizagem

Quanto aos algoritmos de aprendizagem utilizados, apoiamos nossa pesquisa na conclusão feita por [Pereira et al. 2017], quando afirma, baseado em sua análise dos modelos, que os algoritmos de árvores mostram os melhores resultados em base de dados de ACACs, como a utilizada no presente trabalho. Baseados nisso os algoritmos analisados

¹Segundo [Yera Toledo and Martinez 2017]

foram o *Random Forest Classifier* (RFC) e o *Decision Tree Classifier* (DTC), utilizando *Grid Search* para realizar o *hyperparameter tuning* dos modelos.

Os parâmetros submetidos ao Grid Search para realizar o *hyperparameter tuning* do RFC foram os seguintes: Os escolhidos pelo modelo estão em destaque.

- **n_estimators** (representa o número de árvores a serem construídas no modelo): **200**, 400, 600, 800, 1000;
- **max_depth** (representa a profundidade máxima permitida para uma árvore de decisão durante o treinamento.): **100**, 200, 300, 400, 500, Ilimitado.

De mesma forma realizamos a calibração do DTC e os resultados foram os seguintes:

- **max_depth** (representa a profundidade máxima permitida para uma árvore de decisão durante o treinamento.): **100**, 200, 300, 400, 500, 600, 700, 800, 900, Ilimitado.

Como apoio aos modelos anteriormente citados utilizamos um algoritmo de clusterização para identificar grupos de alunos baseados nas mesmas features anteriormente mencionadas. O algoritmo utilizado foi o *K-means*, que é um método de agrupamento que divide um conjunto de dados em *K-means* grupos, onde cada grupo é representado por seu centroide mais próximo.

O código-fonte com a utilização dos modelos, juntamente com o *dataset* utilizado em todas as etapas da pesquisa podem ser acessados no GitHub².

4. Resultados

Esta seção expõe os resultados alcançados em cada um dos experimentos. Foram feitas duas análises, envolvendo os dois modelos mencionados, a fim de mensurar o mais adequado para o cenário em foco, englobando os anos de 2017, 2018 e 2019 e realizando o teste de acurácia com os dados referentes a 2022.

4.1. Comparação dos Modelos de Aprendizagem

Tabela 1. Acurácia por aula prática

Aula	Modelo	Acurácia	Modelo	Acurácia
1	DTC	57.14%	RFC	64.29%
2	DTC	71.43%	RFC	71.43%
3	DTC	57.14%	RFC	71.43%
4	DTC	64.29%	RFC	78.57%
5	DTC	64.29%	RFC	85.71%
6	DTC	78.57%	RFC	71.43%
7	DTC	71.43%	RFC	85.71%
8	DTC	71.43%	RFC	78.57%
9	DTC	57.14%	RFC	78.57%
10	DTC	50.0%	RFC	85.71%
11	DTC	57.14%	RFC	92.86%
12	DTC	35.71%	RFC	78.57%

²<https://github.com/joaolucassolanoufv/fail-prediction-inf110>.

Observando os resultados alcançados percebemos que o modelo de aprendizagem Random Forest Classifier se saiu melhor em 11 dos 12 resultados em relação ao Decision Tree Classifier. Na abordagem proposta, tomando RFC como base, em comparação com os trabalhos relacionados citados anteriormente, tivemos uma acurácia de 71.43% na sexta aula prática, com uma média de 73.81% nas seis primeiras aulas. Já nas seis últimas aulas percebemos que o algoritmo melhora sua assertividade em relação a previsão das reprovações. Nesse período final tivemos uma acurácia média de 83.33%, apresentando, na penúltima aula prática, uma acurácia máxima de 92.86%. Uma possibilidade de justificativa para esse valor perceptivelmente destoante é que o número de alunos tende a diminuir nas últimas aulas do semestre.

4.2. Resultado da Clusterização dos Alunos

Nessa seção podemos analisar os resultados da clusterização dos alunos baseado no seu desempenho durante as aulas práticas.

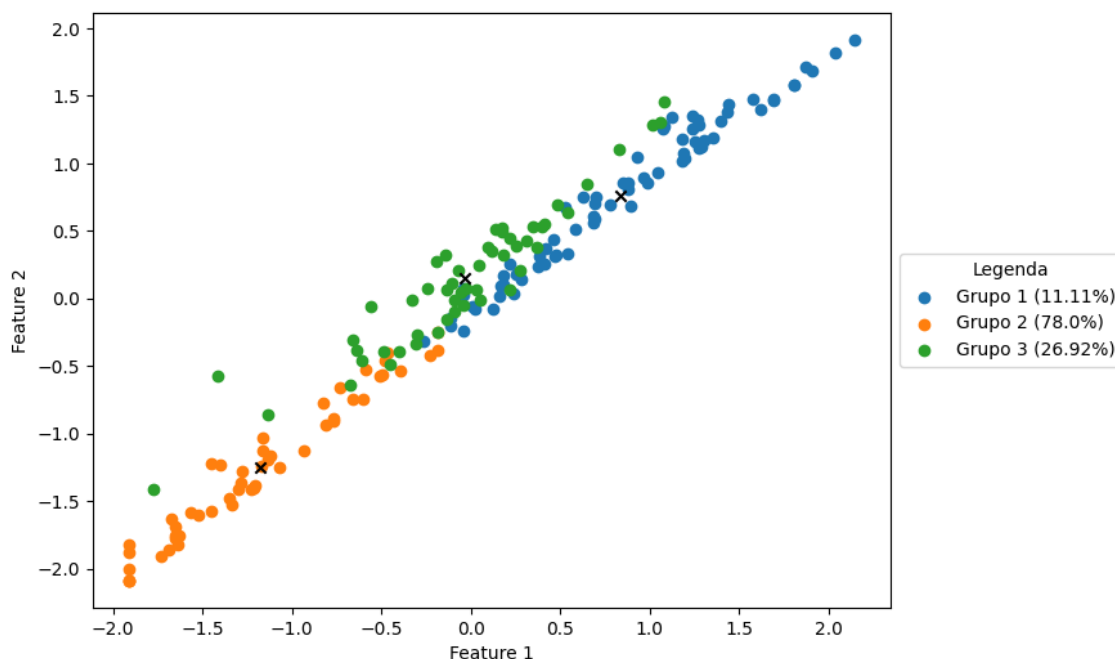


Figura 1. Resultado do K-means

Podemos perceber que, como resultado, tivemos 3 grupos de alunos bem definidos. É possível fazer o relacionamento entre a divisão dos grupos com o desempenho dos indivíduos presentes em cada um. Temos que no Grupo 1, 11.11% dos indivíduos reprovaram na disciplina. Seguimos com o Grupo 3, com 26.92% de reprovação e, por fim, com a maior porcentagem, o Grupo 2, com 78% de reprovados.

É possível deduzir, então, que os indivíduos do Grupo 2 são os que apresentam mais dificuldade na disciplina em questão. Esse dado é de grande importância para o professor dado que, além dos dados individuais gerados pelo modelo de classificação, ele pode ter uma visão comparativa da situação de um aluno em específico com a dos alunos que realizaram a disciplina anteriormente.

4.3. Experimentos

Como exemplificação de uso real das abordagens propostas podemos analisar as suas previsões para um determinado indivíduo real e descaracterizado ao longo do período.

4.3.1. Previsão por Aula Prática

Vamos analisar a situação de um aluno *X* ao longo do período e como ele transita entre os grupos do *K-means* em 4 momentos diferentes e como isso se relaciona com seu desempenho final.

Percebemos que o aluno inicia o período no **Grupo 1**, aquele que apresenta menor porcentagem de reprovados, com o algoritmo prevendo que será **aprovado**, como mantenha esse desempenho.

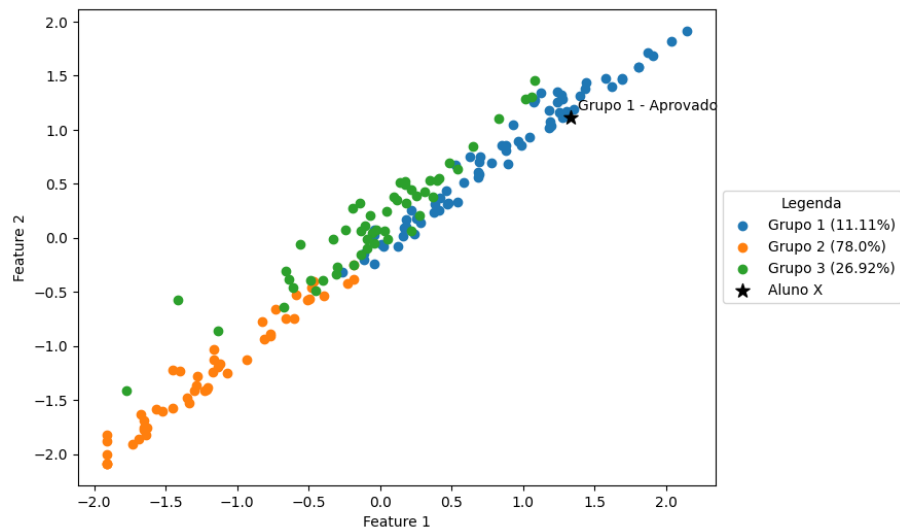


Figura 2. Aula Prática 1

Já nesse momento o algoritmo de classificação já alerta sobre sua **reprovação**, porém ainda se mantém no **Grupo 1**.

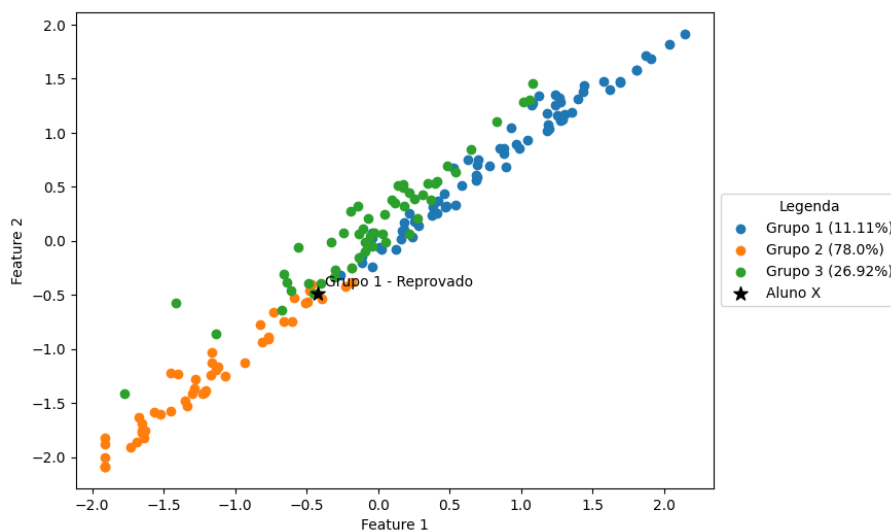


Figura 3. Aula Prática 6

Na aula 9 o indivíduo já pertence ao **Grupo 2**, com percentual mediano de reprovação dentre os grupos.

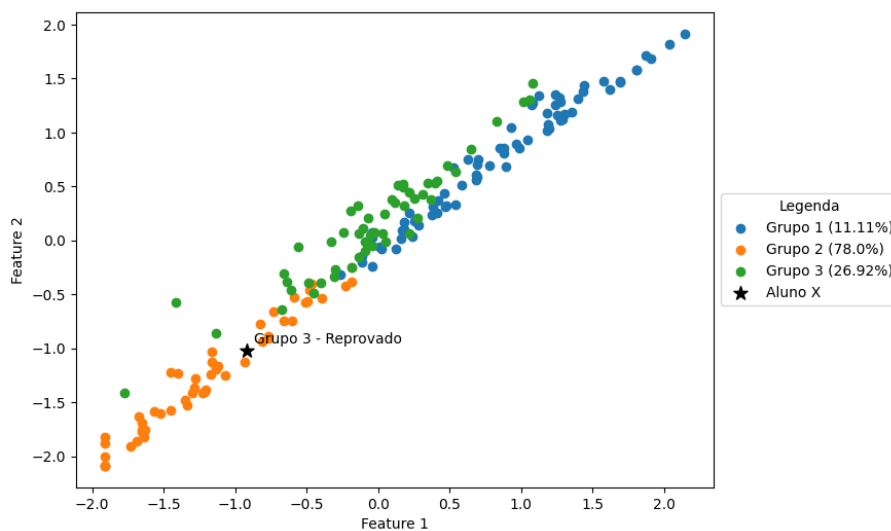


Figura 4. Aula Prática 9

Agora, na aula 11, ao fim do período letivo, o indivíduo se encontra no **Grupo 3**, grupo que apresenta maior percentual de reprovados, indicando que seria o grupo que tem mais dificuldade na disciplina de programação.

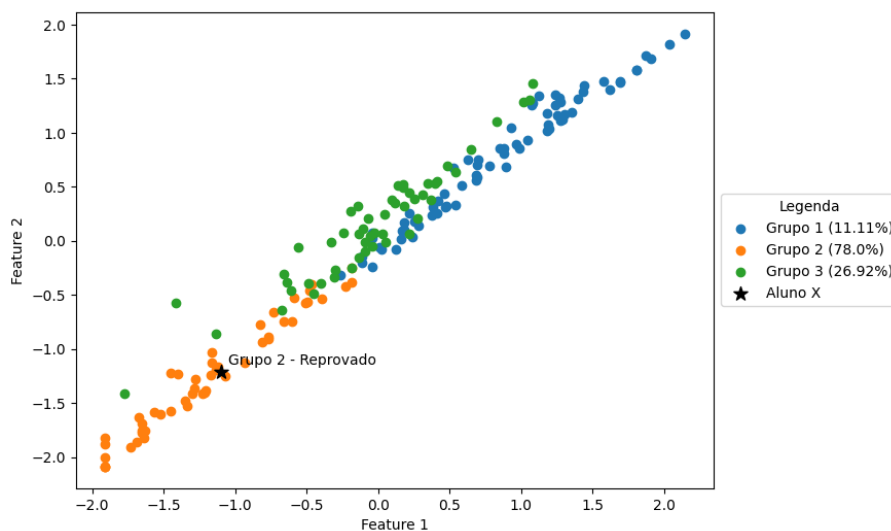


Figura 5. Aula Prática 11

Pelo que analisamos, os algoritmos apontaram que o aluno em questão teve um bom início na disciplina porém, ao decorrer do período, seu desempenho foi decaindo até se encontrar no grupo com mais reprovações e com a previsão de reprovação. De fato, esse aluno em 2022 reprovou na disciplina de programação, exemplificando com clareza o uso da abordagem proposta em um contexto real.

5. Conclusão

Os resultados obtidos confirmam a viabilidade da utilização dos dados gerados pela interação dos alunos no ACACs BOCA, durante a resolução de exercícios propostos na disciplina introdutória de programação, para a previsão rápida e precisa de alunos com tendência de reprovação. Essa abordagem permite ao professor identificar de maneira ágil e objetiva quais alunos necessitam de atenção especial ou de uma metodologia de ensino diferenciada para reverter a tendência de reprovação na disciplina. Com base nessa proposta, é possível tomar medidas adequadas para auxiliar os estudantes, contribuindo assim para o aumento do sucesso acadêmico e aprimoramento do processo de ensino-aprendizagem.

Referências

- Pereira, F., Teixeira de Oliveira, E., and Fernandes, D. (2017). Predição de zona de aprendizagem de alunos de introdução à programação em ambientes de correção automática de código. page 1507.
- Silva, L., Silveira, I., Silva, L., Rodrigues, R., and Ramos, J. (2017). Ciência de dados educacionais definições e convergências entre as áreas de pesquisa.
- Yera Toledo, R. and Martinez, L. (2017). A recommendation approach for programming online judges supported by data preprocessing techniques. volume 47.