revista principia
Vol. 59, N. 3
REVISTA DE DIVULGAÇÃO CIENTÍFICA E TECNOLÓGICA DO INSTITUTO FEDERAL DE CIÊNCIA, EDUCAÇÃO E TECNOLOGIA DA PARAÍBA

DOI: http://dx.doi.org/10.18265/1517-0306a2021id5182
ARTIGO ORIGINAL

SUBMETIDO 06/04/2020
APROVADO 23/07/2021
PUBLICADO ON-LINE 15/08/2021
PUBLICADO 30/09/2022
EDITORA ASSOCIADA
Crishane Azevedo Freire

Predição precoce de problemas de desempenho de estudantes em modalidade de educação on-line: um estudo de caso no ensino médio integrado

João Paulo Lopes de Souza [1] *
Janderson Ferreira Dutra [2]
Damires Yluska de Souza Fernandes [3]

[1] joaopaulopbjp@gmail.com. *Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus Monteiro, Brasil*

[2] janderson.dutra@ifpb.edu.br. *Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus Cajazeiras, Brasil*

[3] damires@ifpb.edu.br. *Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus João Pessoa, Brasil*

*Autor para correpondência.

RESUMO: A necessidade de acompanhamento dos estágios de aprendizado discente e suas dificuldades de forma precoce tomou uma dimensão ainda maior nesses tempos recentes de pandemia de COVID-19 e de ensino on-line. Nesse panorama, este trabalho apresenta uma abordagem para predizer, de forma precoce, o desempenho de estudantes com probabilidade de reprovação superior a 50% em disciplinas específicas que ocorrem em ensino on-line. Para analisar e avaliar a viabilidade da abordagem proposta foi realizado um estudo de caso com dados do ensino técnico integrado do Campus Monteiro do IFPB a partir de dois cenários: (i) para turmas do primeiro ano e (ii) para turmas do segundo e terceiro anos. Foi construído um conjunto de dados com dados reais originados do Google Sala de Aula e do SUAP. Com base no conjunto de dados criado, foram construídos e avaliados modelos de classificação supervisionada usando os métodos Naive Bayes, KNN (K Nearest Neighbors), SVM (Support Vector Machine), Random Forest, Gradient Boosting e Extreme Gradient Boosting. Os métodos de classificação SVM e Gradient Boosting obtiveram especificidade superiores a 93% e 80%, respectivamente.

Palavras-chave: aprendizado de máquina; classificação supervisionada; educação on-line; mineração de dados educacionais; predição de desempenho de estudantes.

Early prediction of student performance problems in online education: a case study from the technical high school education

ABSTRACT: *In times of increasing online teaching, mainly due to COVID-19 pandemic, the need to monitor students' learning stages and provide early diagnosis on their difficulties has gained greater dimensions. In light of*

this overview, this paper presents an approach to early predicting the performance of students who are more than 50 percent likely to fail in specific subjects of online teaching. In order to analyze and evaluate the feasibility of the proposed approach, a case study using data from the integrated technical education at the Federal Institute of Paraíba – Campus Monteiro was carried out considering two scenarios: (i) classes in the first academic year and (ii) classes in the second and third academic years. To help matters, a dataset was built with real data from Google Classroom and SUAP. Based on this dataset, supervised classification models were created and evaluated using the following classification methods: Naive Bayes, KNN (K Nearest Neighbors), SVM (Support Vector Machine), Random Forest, Gradient Boosting and Extreme Gradient Boosting. The specificities of SVM and Gradient Boosting were higher than 93% and 80%, respectively.

Keywords: *Educational Data Mining; machine learning; online education; Student Performance Prediction; supervised classification.*

1 Introdução

A educação tem papel importante na nossa sociedade, tanto social quanto economicamente. Nesse sentido, são investidos muitos recursos para manutenção educacional que precisam ser usados de forma eficiente. Segundo o Portal[1] da Transparência do Governo Federal, foram investidos R$ 86 bilhões na educação pública em 2020.

Nesse panorama de investimentos, contudo, existe ainda um grande número de estudantes que apresentam problemas em seu desempenho acadêmico, seja com reprovações em disciplinas, ou seja, até mesmo com a evasão do curso. Acontece que os sinais desse desempenho indesejado só são geralmente percebidos de forma tardia e, nesse caso, talvez não exista tempo suficiente para uma intervenção que conduza a uma mudança positiva do desempenho.

No contexto do Campus Monteiro do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), conforme dados de 2020, da Plataforma[2] Nilo Peçanha, quase 30% dos estudantes do ensino médio integrado aos cursos técnicos que deveriam se formar em 2019, abandonaram o curso. Adicionalmente, quase 2% dos estudantes estão retidos em função de reprovações em algumas disciplinas.

Um caminho possível para tratar essa problemática é que, baseado em informações de predição de desempenho acadêmico, especialmente em estágios iniciais das disciplinas, os instrutores poderiam ter insumos reais para capturar o andamento dos discentes e aprimorar ou direcionar suas aulas a partir de ações precoces. A ideia é que essas ações possam prover o suporte necessário de apoio aos estudantes para que alcancem seus melhores desempenhos (CHANLEKHA; NIRAMITRANON, 2018). Assim, ao receber esse suporte, o docente pode identificar oportunidades de intervenção nos estágios iniciais

1 Portal da Transparência. **Investimento em educação pública em 2020**. Disponível em: http://www.portaltransparencia.gov.br/funcoes/12-educacao?ano=2020 . Acesso em: 13 out de 2020.

2 Plataforma Nilo Peçanha é um ambiente virtual de coleta, validação e disseminação das estatísticas oficiais da Rede Federal de Educação Profissional, Científica e Tecnológica (Rede Federal). Dados divulgados em 2020. Disponível em: http://plataformanilopecanha.mec.gov.br/2020.html . Acesso em: 13 out. 2020.

para ajudar estudantes em situação de desempenho acadêmico provavelmente insuficiente a voltar aos trilhos (CANO; LEONARD, 2019).

A necessidade de acompanhamento dos estágios de aprendizado discente e de suas dificuldades de forma precoce tomou uma dimensão ainda maior nesses tempos recentes de pandemia de COVID-19. O crescente aumento da modalidade de ensino a distância (EAD) e, mais recentemente, da educação on-line (PIMENTEL; ARAUJO, 2020) ocorreu devido aos esforços de diversos educadores ao redor do mundo para manter o processo de ensino e aprendizagem ativo em tempos de pandemia. É nesse contexto que o presente trabalho se situa.

Para Santos (2009), a educação on-line é tanto uma produção da cibercultura como uma evolução das gerações de EAD. A educação on-line é o conjunto de ações de ensino-aprendizagem ou atos de currículo mediados por interfaces digitais que potencializam práticas comunicacionais interativas e hipertextuais (SANTOS, 2009).

Considerando o cenário exposto, o problema alvo do presente trabalho é verificar a possibilidade de predizer os estudantes que têm probabilidade de reprovação superior a 50% em disciplinas considerando o período de até 3 semanas do início das aulas do bimestre. Considerando o contexto da modalidade de ensino médio integrado, toma-se por desempenho insuficiente aquele quando a média bimestral é menor que 70 pontos (de um total de até 100 pontos). Objetiva-se predizer essa possibilidade nos momentos iniciais das disciplinas (até a 3ª semana após seu início) que estão sendo ministradas em modalidade de educação on-line. A probabilidade de desempenho acadêmico insuficiente, caso não seja tratada com a devida intervenção, pode conduzir, em casos mais extremos, a uma reprovação da disciplina em questão ou até mesmo à evasão do curso do ensino médio integrado.

Assim, este trabalho apresenta uma abordagem para predizer, de forma precoce, os estudantes que possuem probabilidade de reprovação superior a 50% em disciplinas que ocorrem em ensino on-line. Para isso, faz uso de aprendizado de máquina supervisionado (AMP). A predição precoce só é possível para as disciplinas que adotam uma avaliação contínua, uma vez que elas permitem a análise das interações dos discentes, às avaliações e o acesso ao ambiente de aprendizagem on-line. Neste trabalho, o ambiente on-line de aprendizagem é o Google Sala de Aula.

Para analisar e avaliar a viabilidade da abordagem proposta de previsão precoce do desempenho está sendo executado um estudo de caso com estudantes do ensino técnico integrado do campus Monteiro do IFPB. Os estudantes selecionados na amostra estão no primeiro, segundo e terceiro anos do curso. O estudo de caso engloba algumas disciplinas que realizam avaliação contínua em modalidade on-line (existem avaliações semanais) e concluíram o primeiro ou segundo bimestre no primeiro bloco das atividades não presenciais (período de 24/08/2020 a 25/10/2020). Destaque-se que a coleta dos dados ocorre no período entre novembro e dezembro de 2020. Assim, a efetividade de realizar a previsão antecipada é avaliada observando dados das primeiras 3 semanas de aulas em uma dada disciplina do ensino médio integrado.

Os modelos de AMP gerados foram avaliados conforme duas vertentes:

(I) para os discentes do primeiro ano, que ainda não possuem informações anteriores de notas no Instituto. Essa perspectiva é denominada neste trabalho de “modelo do primeiro ano”. Nessa situação, são utilizados dados cadastrais do Sistema Unificado da Administração Pública (SUAP), além das notas do Google Sala de Aula para as atividades não presenciais de 2020;

(II) para os estudantes do segundo e terceiro anos, que já cursaram outras disciplinas durante o primeiro ano e possuem informações anteriores de notas. Essa dimensão é referenciada como “modelo do segundo ano”. Nessa situação, além dos dados cadastrais do SUAP e das notas do Google Sala de Aula para as atividades não presenciais 2020, são utilizadas também as notas dos discentes nas disciplinas anteriores.

O artigo está organizado da seguinte forma: a Seção 2 introduz alguns conceitos; a Seção 3 comenta alguns trabalhos relacionados e compara-os aos diferenciais deste trabalho; a Seção 4 ilustra a abordagem desenvolvida e a criação do conjunto de dados; a Seção 5 apresenta os resultados obtidos com os modelos de AMP; a Seção 6 faz as considerações finais e aponta possibilidades de trabalhos futuros.

1 Fundamentação teórica

Esta seção aborda alguns fundamentos associados ao contexto deste trabalho.

1.1 Processo de mineração de dados educacionais e tarefa de classificação

O CRISP-DM descreve o ciclo de vida de um projeto de mineração de dados e contém seis fases (CHAPMAN *et al*., 2000): entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e implantação. Cada fase é composta por atividades específicas que contribuem para a construção ou refinamento do conjunto de dados a ser usado, assim como para a aplicação de algoritmos de AMP com vistas à geração dos modelos e sua avaliação.

Pode-se definir mineração de dados educacionais (do inglês *Educational Data Mining* – EDM) como a aplicação de técnicas de mineração de dados para os tipos específicos de conjuntos de dados originados de ambientes educacionais (ROMERO; VENTURA, 2020). Observa-se que a partir dos sistemas de apoio educacionais (no IFPB são o Moodle, Google Sala de Aula e SUAP, dentre outros), assim como dos professores, gestores educacionais e estudantes são produzidas matérias primas para tarefas de mineração de dados nessa área.

Uma das tarefas muito utilizadas em EDM é a de classificação supervisionada (RONALDO; PASSOS; BEZERRA, 2015). A tarefa de classificação consiste em criar um modelo de aprendizado com base em informações históricas, cujas instâncias já possuem um rótulo atribuído. A partir desse modelo, o objetivo é predizer a classe de um atributo-alvo de uma nova instância desconhecida.

Existem diversos métodos que realizam essa tarefa. Embasado nos trabalhos relacionados, alguns foram selecionados para estudo neste trabalho, a saber: Naive Bayes – MultinomialNB (NB), K Nearest Neighbors (KNN), *Support Vector Machine* (SVM) e *Random Forest* (RF). Além desses, foram alvo de experimentações também os algoritmos *Gradient Boosting* (GB) e *Extreme Gradient Boosting* (XGB), que possuem reconhecido bom desempenho em competições de mineração de dados (CHEN; GUESTRIN, 2016).

A seguir, é apresentada uma breve descrição a respeito dos métodos citados e usados.

***K-Nearest Neighbors* (KNN) –** Método de classificação que, apesar da simplicidade, é bastante efetivo em muitos casos. Para cada registro ou instância são analisados seus k vizinhos mais próximos. Dessa forma, o seu desempenho é fortemente influenciado pela escolha do valor de *k* (GUO *et al*., 2003). Ronaldo, Passos e Bezerra (2015) explicam que para cada registro do conjunto de dados são calculadas as distâncias para todos os demais registros, isso para identificar os *k* vizinhos mais próximos ou semelhantes. Exemplos de métricas usadas para definição das distâncias são a Euclidiana e a Manhattan (RONALDO; PASSOS; BEZERRA, 2015).

SVM (*Support Vector Machine*) – No SVM, a partir dos vetores de entrada são obtidos os vetores de suporte para cada classe de forma a maximizar a margem que separa esses dois suportes (CORTES; VAPNIK, 1995). Destaca-se que o SVM trabalha com dimensões maiores, usando o conceito de hiperplano que é uma generalização para mais de três dimensões. Outro ponto que merece destaque é o fato de o SVM lidar com dados não linearmente separáveis. Nesse caso, ele utiliza as chamadas funções *kernel* para mapear os dados para um espaço de dimensão maior que o original (RONALDO; PASSOS; BEZERRA, 2015).

Naive Bayes – Segundo John e Langley (2013), o Naive Bayes é um classificador que provê um método simples, com semântica clara, ancorado no conhecimento probabilístico. Os resultados dos testes do algoritmo mostraram bom desempenho para o domínio médico, também conforme John e Langley (2013). A base desse algoritmo é o teorema probabilístico de Bayes que trata do cálculo das probabilidades condicionais. O termo *naive* (ingênuo) refere-se à hipótese de que o método assume a independência da contribuição de cada atributo para a classe (RONALDO; PASSOS; BEZERRA, 2015).

***Random Forest* –** Algoritmos de AMP baseados em árvores são vastos e populares (ALBON, 2018). Sua base é a árvore de decisão, na qual uma série de regras de decisão são encadeadas até chegar nas folhas, que contém os rótulos de classes. A partir das árvores mais básicas são construídas estratégias mais complexas, como as florestas (ALBON, 2018). Breiman (2001) define *Random Forest* como uma coleção de classificadores, estruturados na forma de árvore, na qual cada uma delas irá votar na classe mais popular na entrada. Trata-se de um método do tipo *ensemble* que acrescenta maior resistência ao *overfitting* comparado a uma árvore simples. *Overfitting* é o fenômeno no qual o classificador se ajusta excessivamente ao conjunto de treinamento, tendo um bom desempenho no treino, mas atinge um baixo desempenho no conjunto de testes (RONALDO; PASSOS; BEZERRA, 2015). Os métodos de ensemble servem para agrupar esses modelos preditivos para melhorar a precisão e a estabilidade dos modelos.

No *Random Forest* muitas árvores são treinadas, mas cada árvore recebe apenas uma amostra aleatória das instâncias, e cada nó considera somente um subconjunto dos atributos quando determina a melhor forma de dividir a árvore. Feito isso, por votação majoritária, é determinada a classe (ALBON, 2018).

***Gradient Boosting* –** Friedman (2001) propôs as bases para o algoritmo *Gradient Boosting*. Ele derivou um método de *boosting* cuja otimização da função de perda é baseada no algoritmo do *Gradient Descent*. Esse novo método foi chamado de *Gradient Boosting Machine* (GBM). *Boosting* é um método que treina iterativamente uma série de preditores fracos ou modelos fracos. Frequentemente são árvores simples e, a cada iteração, tenta-se minimizar os erros de predição dos modelos anteriores (ALBON, 2018).

***Extreme Gradient Boosting* (XGBoost) –** O XGBoost é uma melhoria do *Gradient Boosting* que possibilita escalabilidade para execução em infraestrutura distribuída e otimização para usar com mais eficiência os núcleos do processador local. Além disso, ele lida melhor com dados esparsos (CHEN; GUESTRIN, 2016). Os autores citam diversas competições no qual o algoritmo apresentou o melhor desempenho. No site Kaggle[3], por exemplo, de 29 soluções que venceram as suas competições e foram publicadas nos blogs durante o ano de 2015, 17 delas usaram o XGBoost (CHEN; GUESTRIN, 2016).

3 Disponível em: https://www.kaggle.com/. . Acesso em: 13 out. 2020.

1.2 Desbalanceamento de dados

No mundo real, existem vários conjuntos de dados desbalanceados. Isso significa que os dados possuem distorções severas de distribuição, o que dificulta a obtenção de resultados precisos de AMP e, consequentemente, a tomada de decisão (HE; MA, 2013). Levando em conta um conjunto de dados com duas categorias, uma das classes é dita minoritária quando apresenta uma menor quantidade de instâncias quando comparada com a outra classe. Realizar predição com base em conjuntos de dados desbalanceados pode gerar um viés para a classe majoritária (SANTOS *et al*., 2018).

O conjunto de dados obtido e usado neste trabalho apresenta um desbalanceamento de classes (25% – 75%), conforme mostrado na Seção 4.2. Sendo assim, foi necessário realizar um tratamento desse desequilíbrio entre as classes. Para isso existem algumas estratégias de tratamento, dentre elas, a *oversampling* e a *undersampling* (HARRINGTON, 2012).

A estratégia de *oversampling* objetiva replicar ou criar novas instâncias da classe minoritária. Essa estratégia pode gerar um modelo com *overfitting* (HE; MA, 2013). Por outro lado, a estratégia de *undersampling* consiste em remover instâncias da classe majoritária. Contudo, essa remoção pode gerar problemas como excluir dados importantes para o modelo (HE; MA, 2013).

1.3 Seleção de atributos

Em algumas situações, é necessário reduzir a dimensionalidade de um conjunto de dados obtido para uso em modelos de AMP. Uma das técnicas usadas para isso consiste em selecionar os atributos de mais alta qualidade, mais informativos e remover os demais menos úteis àquela tarefa de AMP em questão (ALBON, 2018).

Existem duas técnicas básicas que realizam esse tipo de seleção (RONALDO; PASSOS; BEZERRA, 2015): *filter e wrapper*. Este trabalho aborda apenas a primeira, pois sua execução independe do algoritmo de classificação utilizado e exige menos recursos computacionais. A técnica de *filter* tem por finalidade selecionar os atributos examinando suas propriedades estatísticas. Para dados categóricos pode ser usada a medida qui quadrado, que calcula a independência entre cada atributo e o atributo-alvo da classificação. Por outro lado, para dados quantitativos, pode ser usada a medida ANOVA (ALBON, 2018).

1.4 Avaliação de modelos

Criar modelos de AMP que sejam úteis e considerados de alta qualidade é uma tarefa desafiadora (ALBON, 2018). Segundo Ferrari e Castro (2016), uma das etapas do processo de classificação supervisionada é a separação dos dados em conjunto de treinamento e testes, na qual os dados de treino são usados para gerar o modelo, enquanto os dados de teste são usados para avaliar a qualidade do modelo gerado. Nesse sentido, Albon (2018) enumera duas estratégias principais de particionamento dos dados: *hold-out* e validação cruzada.

Uma vez realizado o treinamento e testes é o momento de avaliar o desempenho da predição do classificador. Para avaliar esse desempenho são usadas métricas baseadas em

algum cálculo de erro entre a saída fornecida pelo modelo e a saída desejada (FERRARI; CASTRO, 2016).

Para isso, algumas métricas de avaliação são normalmente empregadas (HE; MA, 2013):

- Acurácia geral (*overall accuracy*): mede o percentual de instâncias classificados corretamente e é calculado como (TP + TN) / (TP + FN + FP + TN). Onde: TP é *True Positive*; TN é *True Negative; FN é* False Negative e FP é *False Positive*;
- Sensitividade (*sensitivity*): verifica o percentual de positivos classificados corretamente. Calculado através da fórmula TP / (TP + FN);
- Especificidade (*specificity*): refere-se ao percentual de negativos corretamente identificados e pode ser obtido como TN / (TN + FP);
- Área sobre a curva ROC (AUC): faz uso da curva ROC (*Receiver Operating Characteristic*) para exibir o *trade-off* entre as taxas de classificação dos TP e FP.

Santos *et al*. (2018) alerta que a acurácia geral resulta algumas vezes em predições enviesadas para a classe majoritária no caso de *datasets* (conjunto de dados) desbalanceados e indica outras métricas adicionais a serem usadas, como a sensitividade (*sensitivity*), especificidade (*specificity*) e área sobre a curva ROC (AUC).

1.5 Transformação de dados categóricos para numéricos

Dados categóricos apresentam um conjunto limitado de valores possíveis. Normalmente, as categorias são do tipo texto (*string*). Alguns algoritmos têm restrições para trabalhar com dados categóricos diretamente. Para isso, pode-se realizar a transformação dessas categorias em valores numéricos. Manish (2020) apresenta duas técnicas com esse objetivo: *Label Encode* e *One Hot Encode*. Na primeira, segundo Manish (2020), é realizada uma codificação de cada categoria em um valor numérico, chamado de rótulo numérico. Esses rótulos são numerados entre 0 e a quantidade de categorias menos uma unidade. Enquanto na segunda, a estratégia é converter cada categoria em uma nova coluna que receberá os valores 1 ou 0 (verdadeiro/falso). Um grande benefício dessa técnica é evitar que os valores numéricos convertidos recebam pesos de forma imprópria pelo algoritmo de classificação (MANISH, 2020).

2 Trabalhos relacionados

Alguns trabalhos associados à predição de desempenho de estudantes em modalidades gerais de ensino são descritos a seguir.

Barros *et al*. (2020) analisou a viabilidade de predizer o desempenho de estudantes do ensino superior do curso de ciência e tecnologia em uma disciplina do segundo período (linguagem de programação) a partir de dados das notas nas disciplinas do primeiro período. Para alcançar esse objetivo, o autor utilizou uma tarefa de classificação binária e os métodos Naive Bayes, KNN, SVM e Árvore de decisão. O SVM apresentou a melhor acurácia, e a Árvore de decisão obteve a melhor especificidade para a classe de reprovado, 81%.

Chanlekha e Niramitranon (2018) realizaram experimentos com dados de turmas de sala de aula tradicional e criaram dois modelos (primeiro semestre e semestres posteriores) para tentar predizer, nos estágios iniciais das disciplinas, os estudantes que possuem alto risco de receber notas baixas. Para isso, foram utilizados os métodos de classificação Árvore de decisão, Naive Bayes, *Random Forest*, SVM e Redes Neurais. O método que apresentou os melhores valores de F2 Score para o primeiro semestre foi o *Random Forest* (55% para a disciplina Computador e programação e 78% para Matemática de engenharia I). Para os semestres posteriores os melhores métodos foram o SVM, com 66% de F2 Score para a disciplina Mecânica I, e Rede neural, com 69% de F2 Score para a disciplina de Mecânica II e 93% para Análise e projeto de algoritmos.

O trabalho de Rabelo *et al*. (2017), por sua vez, aplicou métodos de classificação em dados de cursos de graduação a distância que utilizam o ambiente virtual de aprendizagem (AVA) Moodle, para predizer o desempenho de estudantes em duas classes – sucesso ou insucesso – no decorrer das disciplinas. Foram utilizados os métodos de classificação ID3 e J48 (árvores de decisão). O J48 apresentou o melhor valor em termos de acurácia (96,5%).

Este trabalho se diferencia dos demais apresentados nesta seção em função de alguns aspectos tratados: (i) foi gerado um conjunto de dados a partir do Google Sala de Aula com informações do ensino médio integrado ao técnico, no contexto do IFPB, Campus Monteiro; (ii) na abordagem proposta, são utilizadas informações apenas das duas primeiras atividades avaliativas; (iii) a abordagem avalia adicionalmente os algoritmos de *Gradient Boosting* e *XGBoosting* para geração do modelo de predição; (iv) o trabalho realiza um tratamento para o desbalanceamento de classes existentes e analisa o impacto da seleção de atributos no resultado dos modelos de classificação obtidos.

3 Abordagem desenvolvida

A abordagem desenvolvida neste trabalho foi baseada nas etapas definidas pelo modelo de processo CRISP-DM (acrônimo para *Cross Industry Standard Process for Data Mining*).

A abordagem desenvolvida está pautada em dois pontos principais que são descritos a seguir.

2.1 Conjunto de dados

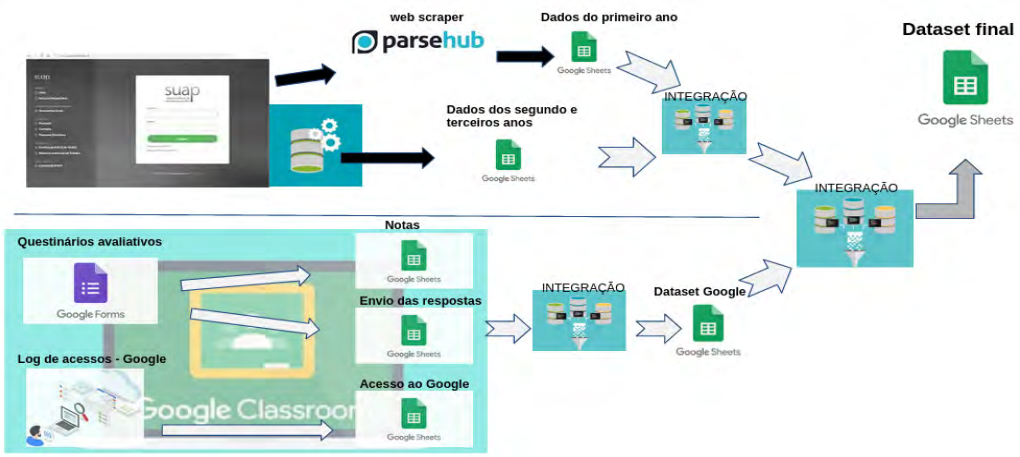
Os dados necessários para o trabalho se encontravam em fontes de dados diferentes. A Figura 1 ilustra o processo de coleta, extração e pré-processamento do conjunto de dados.

Na coleta inicial, o *dataset* foi composto de 15 atributos oriundos do SUAP referentes a 492 instâncias de estudantes e 8 atributos provenientes do Google Sala de Aula relativos ao total de 488 instâncias de estudantes (existem alguns estudantes que estão cadastrados no SUAP, mas não estão matriculados nas disciplinas ministradas no Google Sala de Aula). As instâncias da sala de aula do Google estão distribuídas da seguinte forma, entre primeiro, segundo e terceiro ano:

- No primeiro ano, tem-se um total de 259 instâncias, sendo 134 instâncias da disciplina Filosofia I e 125 instâncias da disciplina Informática Básica;
- No segundo ano, o total é de 128 instâncias da disciplina Filosofia II;
- No terceiro ano, totalizam-se 101 instâncias da disciplina Filosofia III.

Os dados necessários para construção do *dataset* foram obtidos em etapas e de diversas fontes e formatos, conforme ilustrado na Figura 1.

Figura 1 ▶
Pipeline de coleta, integração e geração do *dataset*.
Fonte: elaborada pelos autores



Como visto na Figura 1, uma parte importante da coleta de dados foi realizada a partir do sistema SUAP, que apoia a gestão acadêmica no IFPB. A maioria dos dados coletados a partir do SUAP foi extraída e cedida pela Diretoria de Ensino do Campus Monteiro do IFPB. Contudo, os dados para os estudantes do primeiro ano não foram extraídos em plenitude. Nesse caso específico, para obtenção dos dados desses estudantes, que ainda não concluíram nenhuma disciplina, foi utilizada a versão gratuita do *web scraper* parsehub[4] para extrair os dados deste subgrupo de estudantes usando o acesso de professor no SUAP.

Um fragmento desse *dataset* parcial com dados do SUAP é mostrado na Figura 2.

Outra parte relevante dos dados foi obtida diretamente do Google Sala de Aula (*Classroom*). Nesse sentido, a coleta foi bem mais difícil, pois não existe extração automatizada e centralizada. Isso é verdade, pelo menos, para a realidade do IFPB, que possui uma licença simplificada do Google. Com o apoio da Direção de Ensino, que sensibilizou alguns professores para permitir a coleta dos dados diretamente nas salas de aula virtuais de suas disciplinas, foi possível coletá-los.

Figura 2 ▶
Fragmento do *dataset* parcial do SUAP.
Fonte: dados da pesquisa

Matricula	Curso	Ano de Ingresso	Cor/Raca	Cota MEC	Cota SISTEC	Data de Nascimento	Nome da Escola de Origem	Sexo	Zona Residencial do Aluno
	MSI	2020	Nao declarada	publica	Escola Publica	06/04/2004	E.M.E.F II PROF.A MARIA DO SOCORRO ARAGAO LIBERAL	M	Rural
	MSI	2020	Branca	publica	Escola Publica	07/08/2004	EE. EUGENIA FERRAREZI NUNES	M	Urbana
	MSI	2020	Parda	publica_PPI	Escola Publica	13/08/2004	E.M.E.F II PROF.A MARIA DO SOCORRO ARAGAO LIBERAL	M	Urbana
	MSI	2020	Parda	publica_PPI	Escola Publica	07/05/2005	E.M.E.F II PROF.A MARIA DO SOCORRO ARAGAO LIBERAL	F	Urbana
	MSI	2020	Parda	publica_PPI	Escola Publica	30/03/2005	E.M.E.F. TOBIAS REMIGIO GOMES	F	Rural

Para a extração dos dados do Google Sala de Aula foi preciso gerar uma planilha contendo os dados com as notas dos estudantes da turma (para extrair as notas em cada atividade avaliativa) e, para cada atividade avaliativa do tipo formulário, foi gerada uma planilha com as respostas, os e-mails e dados do momento do envio (para extrair

4 Disponível em: https://www.parsehub.com/. . Acesso em: 13 out. 2020.

informações do momento do envio). Esses dois tipos de planilhas são gerados de forma simplificada pela plataforma de ensino.

A partir dessas planilhas são obtidas informações das avaliações contínuas semanais para algumas disciplinas que estão no primeiro bloco (período de 24/08 a 25/10) das atividades não presenciais de 2020 (período de pandemia de COVID-19). Essas informações são notas das duas primeiras atividades avaliativas da disciplina (numérico – 0 a 100), tempo em dias entre a data de criação do arquivo da primeira atividade avaliativa e a data do envio das respostas (numérico – 0 a 99, sendo o valor 0 para o envio das respostas no mesmo dia de disponibilização da atividade e o valor 99 quando não foi enviada resposta), nota final do bimestre na disciplina e a situação do estudante ao final do bimestre (categórico – Desempenho suficiente e Desempenho insuficiente).

Além dos dois tipos de planilhas citados, também foram obtidas informações do relatório padrão do Google Sala de Aula através de vários chamados abertos pelo SUAP e direcionados para a Diretoria de Gestão de Tecnologia da Informação (DGTI). Cada um desses chamados solicitava os dados do último acesso dos usuários do Campus Monteiro para cada semana inicial de aula. O objetivo era extrair o momento do último acesso dos estudantes em cada semana de aula para calcular o tempo em dias, entre a data do início da primeira semana e a data do último acesso na semana ao ambiente AVA – Google Sala de Aula (numérico – 0 a 99, sendo o valor 0 quando o discente acessou pela última vez no primeiro dia da semana e o valor 99 quando não houve acesso na semana).

Para finalizar o *dataset* originado da fonte Google Sala de Aula foi necessário integrar os dados dos três tipos de planilhas indicadas. Para essa integração é utilizado o e-mail acadêmico do discente que é comum às três planilhas. Um fragmento desse *dataset* parcial com dados do Google Sala de Aula é apresentado na Figura 3.

Além dessa integração, fez-se necessário outra etapa de integração de dados entre os conjuntos de dados oriundos do SUAP e do Google Sala de Aula. Para isso, foi realizada uma junção entre esses *datasets* com base no e-mail acadêmico.

Figura 3 ▶
Fragmento do *dataset* parcial do Google Sala de Aula.
Fonte: dados da pesquisa

E-mail	Q1 - Nota	Q2 - Nota	Media Bimestre I	Situacao final bimestre	Codigo Situacao final bimestre	Q1 - Tempo envio	Q2 - Tempo envio	Nome	Matricula	S1 - Ultimo acesso	S2 - Ultimo acesso
	88.888889	95.0	94.629630	Desempenho bom	1	10.0	3.0			Aug 29, 2020 9:21:04 PM	Sep 5, 2020 4:37:04 PM
	26.666667	50.0	25.555556	Desempenho insuficiente	0	12.0	8.0			Aug 25, 2020 10:17:57 AM	Aug 31, 2020 6:30:49 PM
	80.000000	70.0	77.777778	Desempenho bom	1	13.0	15.0			NaN	Aug 31, 2020 11:21:50 AM
	88.888889	0.0	29.629630	Desempenho insuficiente	0	12.0	NaN			Aug 27, 2020 9:32:10 PM	Aug 31, 2020 2:45:01 PM
	88.888889	100.0	96.296296	Desempenho bom	1	10.0	5.0			Aug 27, 2020 3:45:13 PM	Sep 5, 2020 5:09:53 PM

A maioria dos atributos originados do SUAP são categóricos, tais como cor/raça (categórico), data de nascimento (data), sexo (categórico), dentre outros. Enquanto aqueles provenientes do Google Sala de Aula são, na maior parte, valores numéricos. Para esses dados categóricos foi realizada uma transformação para valores numéricos.

Para isso, foram utilizadas duas técnicas (MANISH, 2020): *Label Encode* e *One Hot Encode*. O critério na escolha da técnica foi baseado na quantidade de categorias em cada atributo.

Para os atributos que possuem poucas categorias (menor ou igual a oito categorias) – Curso, Cor/Raça, Cota MEC, Cota SISTEC, Faixa de Renda (SISTEC), Sexo, Tipo Escola de Origem e Zona Residencial do Estudante – foi utilizada a técnica *One Hot Encode* para que o modelo atribua pesos iguais a cada uma categoria.

Apenas em um atributo – Escola de Origem – que possuía muitas categorias (66 categorias), foi utilizada a técnica de *Label Encode*.

Foi necessário também derivar um novo campo – idade – a partir da data de nascimento.

Ao final do pré-processamento, seguem os atributos dos conjuntos de dados para o modelo do segundo ano que é semelhante ao do primeiro, mas inclui as notas de algumas disciplinas cursadas anteriormente:

- "Q1 - Nota", "Q2 - Nota", "Q1 - Tempo envio", "Q2 - Tempo envio", "S1 - Tempo ultimo acesso", "S2 - Tempo ultimo acesso", "Ano de Ingresso", "nota_Biologia_I", "nota_Educacao_Fisica_I", "nota_Filosofia_I", "nota_Fisica_I", "nota_Geografia_I", "nota_Historia_I", "nota_Lingua_Portuguesa_I", "nota_Matematica_I", "nota_Quimica_I", "nota_Sociologia I", "Idade", "curso_integrado_MSI", "curso_integrado_MUSICA", "curso_integrado_TED", "sexo_F", "sexo_M", "cor_raca_Branca", "cor_raca_Indigena", "cor_raca_Nao declarada", "cor_raca_Parda", "cor_raca_Preta", "tipo_escola_origem_Privada", "tipo_escola_origem_Publica", "zona_residencial_Rural", "zona_residencial_Urbana", "cota_SISTEC_Cor/Raca", "cota_SISTEC_Escola Publica", "cota_SISTEC_Nao se aplica", "cota_SISTEC_Necessidades Especiais", "cota_MEC_Nao se aplica", "cota_MEC_publica", "cota_MEC_publica_PCD", "cota_MEC_publica_PPI", "cota_MEC_publica_renda_menor_1,5SM", "cota_MEC_publica_renda_menor_1,5SM_PCD", "cota_MEC_publica_renda_menor_1,5SM_PPI", "cota_MEC_publica_renda_menor_1,5SM_PPI_PCD", "Escola de Origem", "Codigo Situacao final bimestre".

Um fragmento do conjunto de dados final para o modelo do segundo ano é mostrado na Figura 4.

Figura 4 ▶
Amostra dos dados do modelo do segundo ano – *dataset* final.
Fonte: dados da pesquisa

	Q1 - Nota	Q2 - Nota	Q1 - Tempo envio	Q2 - Tempo envio	S1 - Tempo ultimo acesso	S2 - Tempo ultimo acesso	Ano de Ingresso	('Nota', 'Biologia I')	('Nota', 'Educacao Fisica I')	('Nota', 'Filosofia I')	...	cota_SISTEC_Necessidades Especiais	cota_MEC_Nao se aplica
0	0.0	0	99.0	99.0	99.0	99.0	2016.0	65.0	83.0	54.0	...	0.0	0.0
1	75.0	80	5.0	5.0	99.0	99.0	2016.0	52.0	81.0	57.0	...	0.0	0.0
2	75.0	80	1.0	5.0	2.0	5.0	2017.0	73.0	88.5	68.0	...	0.0	1.0
3	87.5	90	5.0	5.0	99.0	1.0	2017.0	69.0	96.0	72.0	...	0.0	0.0
4	62.5	0	6.0	99.0	99.0	2.0	2017.0	72.0	84.0	66.0	...	0.0	1.0

2.2 Modelagem e avaliação

A abordagem desenvolvida está centrada em uma tarefa de classificação binária que deve indicar se o desempenho do estudante será bom ou insuficiente. A meta é predizer de forma precoce quais estudantes possuem um elevado risco de reprovação. Assim, o interesse maior é encontrar os estudantes classificados na categoria "Desempenho insuficiente".

Para a realização da modelagem e avaliação experimental, foram definidos dois cenários de experimentação, descritos a seguir:

- **Cenário 1:** considerado o *baseline*, neste cenário foram gerados modelos iniciais a partir da divisão simples dos dados de treino e teste (*hold out*) de forma aleatória.
- **Cenário 2:** nesse cenário, foi utilizada a técnica de validação cruzada *k-fold* estratificada pela variável alvo com 10 *folds*. Adicionalmente, foram realizadas diversas combinações da validação cruzada com: (i) tratamento do desbalanceamento de classes e (ii) análise de *features* (todos os atributos, os 10 atributos mais importantes e os 6 atributos mais importantes).

O Cenário 2 foi subdividido em Cenário 2.1, voltado à geração do modelo do primeiro ano, e Cenário 2.2, que buscou a obtenção do modelo do segundo e terceiro anos.

Em todos os cenários, os algoritmos utilizados foram: Naive Bayes – MultinomialNB (NB), *K Nearest Neighbors* (KNN), *Support Vector Machine* (SVM) e *Random Forest* (RF), *Gradient Boosting* (GB) e *Extreme Gradient Boosting* (XGB). Uma descrição breve acerca dos algoritmos foi apresentada na Seção 2.1.

Para o desenvolvimento, documentação e testes, foram usados o Jupyter Notebook e a linguagem Python. As bibliotecas específicas do Python – como scikit-learn[5], pandas[6], numpy[7], imbalanced-learn[8], xgboost[9] e matplotlib[10] – foram usadas. Todos os experimentos ocorreram em máquina local – Notebook Dell Inspiron 14. Ele possui 16 Gb de RAM, Processador Intel Core i7-8565U com 8 núcleos e velocidade de 1.80 GHz e Disco SSD.

Para a avaliação dos modelos, algumas métricas foram definidas tendo em vista o foco da abordagem proposta. O foco é encontrar os estudantes classificados na categoria "Desempenho insuficiente", ou seja, rótulo negativo, que é a classe minoritária, conforme analisado anteriormente. Dessa forma, baseado em Santos *et al*. (2018) e He e Ma (2013), a métrica mais indicada para essa avaliação da classe dos verdadeiros negativos (TN) é a especificidade.

Além da especificidade, foram incluídas a métrica que identifica a área sob a curva ROC (AUC) e a acurácia geral (*overall accuracy*). Essas métricas são amplamente utilizadas nos trabalhos relacionados e podem ser usadas para fins de comparação.

Com relação ao aspecto de desbalanceamento dos dados, uma vez que existem mais estudantes com desempenho acadêmico suficiente comparados com aqueles que apresentam desempenho insuficiente, algumas questões foram contempladas.

Para o *dataset* utilizado na construção do modelo do primeiro ano, o qual é composto pelas disciplinas Filosofia I e Informática Básica, pode-se verificar que a classe majoritária possui 194 instâncias (75%), enquanto a minoritária, só 65 instâncias (25%).

Por outro lado, para o *dataset* usado na construção do modelo do segundo ano, o qual é composto pelas disciplinas Filosofia II e Filosofia III, pode-se verificar que a classe majoritária possui 177 instâncias (78%), enquanto a minoritária, somente 52 (22%).

---

5 Disponível em: https://scikit-learn.org/stable/. . Acesso em: 13 out. 2020.

6 Disponível em: https://pandas.pydata.org/. . Acesso em: 13 out. 2020.

7 Disponível em: https://numpy.org/. . Acesso em: 13 out. 2020.

8 Disponível em: https://imbalanced-learn.readthedocs.io/en/stable/install.html. . Acesso em: 13 out. 2020.

9 Disponível em: https://xgboost.readthedocs.io/en/latest/python/python_intro.html. . Acesso em: 13 out. 2020.

10 Disponível em: https://matplotlib.org/. . Acesso em: 13 out. 2020.

Os percentuais indicados para as duas vertentes exibem valores que carecem de tratamento do desequilíbrio entre as duas classes. Para tratar esse desbalanceamento, foram utilizadas as técnicas de *oversampling* – *RandomOverSampler* (ROS) e SMOTE (SMO) – e *undersampling* – *RandomUnderSampler* (RUS).

Outrossim, ao lidar com esse tipo de tratamento, outro ponto importante a destacar é o momento no qual é aplicado o tratamento citado. Nesse sentido, Santos *et al*. (2018) e Mohan (2019) fazem algumas análises que demonstram a aplicação da técnica de rebalanceamento antes da validação cruzada e durante a validação para cada *fold*. Os autores indicam que o tratamento deve ser feito durante a validação, para cada cada *fold*, somente para o conjunto de treinamento. Este trabalho segue a diretriz de Santos *et al*. (2018) e Mohan (2019).

Para a seleção de *features*, buscando analisar a influência dos diferentes atributos do *dataset* para a geração dos modelos de classificação, foi aplicada a técnica de seleção de atributos do tipo *Filter*, *SelectKBest*. Para isso, foi usada a medida estatística Chi quadrado para selecionar as melhores 15 *features*. Foram gerados três subconjuntos de atributos para análise: todas as *features* (Todas), os 10 (Top_10) e os 6 (Top_6) atributos mais importantes.

3 Resultados

Nesta seção, são apresentados os resultados obtidos relativos à avaliação experimental dos métodos de classificação, tratamento de desbalanceamento e seleção de atributos, conforme os cenários descritos na Seção 4.2. Os resultados relativos às métricas de desempenho dos modelos de classificação gerados foram ordenados de forma decrescente dos valores relativos à especificidade, AUC e acurácia geral.

Para as tabelas com os resultados exibidas nesta seção utiliza-se o padrão de abreviação dos títulos das colunas da seguinte maneira:

- **Cla**: método classificador;
- **Fea**: subconjunto de atributos (*features*);
- **Res**: técnica de tratamento de desbalanceamento (*reasampling*);
- **Esp**: métrica especificidade;
- **Auc**: métrica área sobre a curva ROC;
- **Acc**: métrica acurácia geral (*overall accuracy*);
- **Tt**: tempo de treinamento (segundos).

3.1 Cenário 1 – ***baseline***

Na Tabela 1 são exibidas as métricas para o modelo do primeiro ano seguindo o Cenário 1:

Tabela 1 ▶
Métricas dos modelos do Cenário 1 – primeiro ano.
Fonte: dados da pesquisa

Classificador	Esp	AUC	Acc	Tt
NB	0,35	0,666	0,820	0,008
KNN	0,5	0,741	0,858	0,004
SVM	0,75	0,866	0,923	0,192
RF	0,7	0,824	0,884	0,152
GB	0,7	0,815	0,871	0,056
XGB	0,7	0,832	0,897	28,97

Na Tabela 2 são exibidas as métricas alcançadas para o modelo do segundo ano também dentro do contexto do Cenário 1:

Tabela 2 ▶
Métricas dos modelos do Cenário 1 – segundo ano.
Fonte: dados da pesquisa

Cla	Esp	Auc	Acc	Tt
NB	0,562	0,724	0,811	0,004
KNN	0,375	0,687	0,855	0,002
SVM	0,687	0,787	0,840	0,25
RF	0,5	0,740	0,869	0,153
GB	0,687	0,824	0,898	0,09
XGB	0,687	0,815	0,884	44,85

3.2 Cenário 2.1 para o modelo do primeiro ano

Para o modelo das disciplinas do primeiro ano seguindo o Cenário 2.1, foram alcançados os resultados conforme mostra a Tabela 3:

Tabela 3 ▶
Métricas para o Cenário 2.1 – primeiro ano.
Fonte: dados da pesquisa

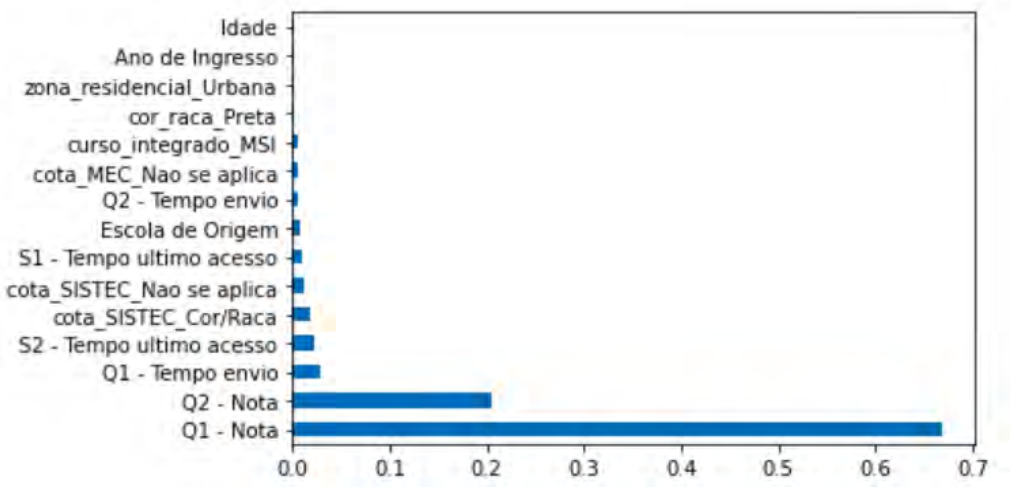
Cla	Fea	Res	Esp	Auc	Acc	Tt
GB	Todas	SMO	0,8	0,891	0,935	0,908
XGB	Top_10	RUS	0,8	0,865	0,897	32,60
RF	Top_10	RUS	0,8	0,865	0,897	1,602
XGB	Top_6	RUS	0,8	0,865	0,897	105,71
RF	Top_6	RUS	0,8	0,856	0,884	1,576
KNN	Top_6	ROS	0,8	0,856	0,884	0,076

Da análise da Tabela 3, destaca-se que os cinco melhores modelos utilizaram algoritmos do tipo *ensembles*, e a maioria utilizou a técnica de *undersampling* para balanceamento, à exceção do modelo com *Gradient Boosting* que usou todos os atributos e a técnica SMOTE.

Em se tratando do modelo do primeiro ano, que não conta com informações de notas do ano anterior, vale ressaltar que a maioria dos algoritmos conseguiu bons resultados com apenas 6 ou 10 atributos.

Na sequência, faz-se uma avaliação dos atributos que mais contribuíram para o melhor modelo – *Gradient Boosting Classifier* – Todas – SMO (classificador *Gradient Boosting*, com todas as *features* e técnica de tratamento de desbalanceamento SMOTE), Figura 5.

Figura 5 ▶
Importância de cada *feature* para o melhor modelo – Cenário 2.1.
Fonte:dados da pesquisa



Desses resultados, percebe-se que o método *Gradient Boosting* atingiu 80% de acerto para a classe minoritária. Nesse sentido, o *Gradient Boosting* com todos os atributos atingiu melhor desempenho e, para isso, usou atributos como cotas, o curso e ano de ingresso além das notas das duas primeiras atividades, seus tempos de envio e o acesso na primeira semana e na segunda.

3.3 Cenário 2.2 para o modelo do segundo ano

Para o modelo do segundo ano no contexto do Cenário 2.2, os resultados obtidos encontram-se na Tabela 4:

Tabela 4 ▶
Métricas para o Cenário 2.2 – segundo ano.
Fonte:dados da pesquisa

Cla	Fea	Res	Esp	Auc	Acc	Tt
SVM	Top_10	RUS	0,937	0,874	0,840	0,461
SVM	Top_6	RUS	0,937	0,864	0,826	0,166
SVM	Top_10	ROS	0,875	0,862	0,855	3,532
SVM	Top_6	ROS	0,875	0,862	0,855	0,546
SVM	Top_6	SMO	0,875	0,852	0,840	0,511
KNN	Top_6	SMO	0,875	0,824	0,797	0,08
KNN	Top_6	RUS	0,875	0,805	0,768	0,076
GB	Todas	ROS	0,812	0,859	0,884	1,44

Desses valores, observa-se que o SVM com o *kernel* linear obteve os valores mais interessantes e equilibrados na classificação das duas classes, conforme destacam os valores de AUC. Esse algoritmo se adaptou bem às várias técnicas de tratamento de

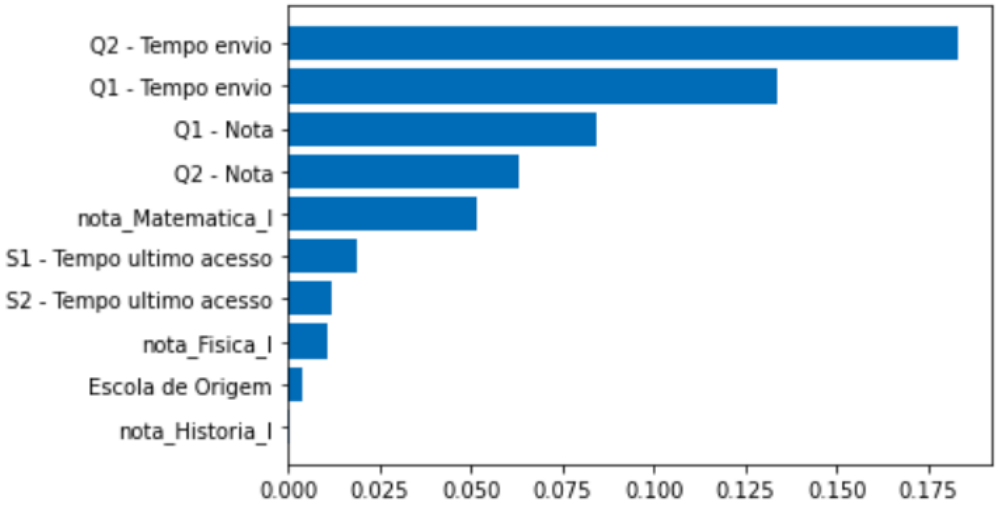
desbalanceamento dos dados, sendo que o *under sampling* conseguiu os melhores resultados.

Vale destacar também o desempenho do *Gradient Boosting* que apresenta um ótimo resultado para a AUC.

Em relação à importância dos atributos, destaca-se que quase todos os algoritmos apresentam valores ótimos para os subconjuntos de 6 ou 10 *features* mais importantes. Sendo assim, os modelos ficam mais simples e, possivelmente, mais generalistas e imunes ao *overfitting*.

Faz-se uma análise dos atributos que mais contribuíram para o melhor modelo – SVM – Top 10 – RUS (classificador SVM, com as dez *features* mais importantes e técnica de tratamento de desbalanceamento RUS), Figura 6.

Figura 6 ▶
Importância de cada *feature* para o melhor modelo – Cenário 2.2.
Fonte: dados da pesquisa



Esses resultados demonstram que o modelo atribuiu maior peso para as notas das atividades avaliativas das duas primeiras semanas – "Q1 - Nota" e "Q2 - Nota" – e para o tempo de envio dessas respostas – "Q1 - Tempo envio" e "Q2 - Tempo envio". Só então ele utilizou as notas das disciplinas de Matemática I, Física I e História I e a informação da escola de origem.

3.4 Discussão

Uma questão relevante demonstrada nos resultados é que foi possível obter um ótimo desempenho de classificação da classe minoritária, com percentual de acerto superior a 80%, utilizando apenas informações das duas primeiras semanas de aula – notas das avaliações contínuas e envio das suas respostas, além das informações de acesso ao ambiente de aprendizagem por parte do estudante nesse período.

Fazendo uma comparação dos resultados dos modelos do primeiro ano e do segundo ano com o *baseline* – Cenário 1, confirma-se uma pequena melhoria nos valores de especificidade e AUC no primeiro caso e um significativo incremento desses valores para o segundo caso. Para o primeiro ano, a pequena melhora se deu, possivelmente, em função da pequena quantidade de atributos disponíveis.

Outra análise importante dos resultados diz respeito ao tempo de treinamento dos métodos de classificação. Nesse sentido, verifica-se que o único método que apresentou um tempo de treino muito maior que os demais foi o *Extreme Gradient Boosting* (XGB) – com tempo de treinamento que foi entre 20 a 179 vezes maior que o segundo método

mais lento no treino. Já o *Gradient Boosting* (GB), outro método do tipo *ensemble*, exibiu ótimos tempos nos resultados gerais, demonstrando tempo de treino mais lento apenas comparado com os métodos KNN e o *Naive Bayes* (NB). A única exceção desse desempenho do GB foi no Cenário 2.2, no qual apresentou tempo de treino mais lento que o SVM, além do KNN.

Seleção de atributos. Para o modelo do primeiro ano que não possui informações de notas anteriores, somente dados cadastrais e os atributos das notas e comportamentos oriundos do Sala de Aula, segue a lista na Tabela 5.

Tabela 5 ▶
Atributos mais importantes – modelo primeiro ano, segundo SelectKBest e chi quadrado.
Fonte: dados da pesquisa

Atributo	***Score***
S2 - Tempo ultimo acesso	4505,66
Q2 - Tempo envio	3410,32
S1 - Tempo ultimo acesso	3360,77
Q1 - Tempo envio	1520,14
Q2 - Nota	1472,42
Q1 - Nota	1384,50
Escola de Origem	65,25
curso_integrado_MUSICA	14,19
cota_SISTEC_Necessidades Especiais	13,28
cota_SISTEC_Nao se aplica	10,89
cota_MEC_Nao se aplica	9,02
curso_integrado_MSI	8,19
cota_SISTEC_Cor/Raca	7,68
cota_MEC_publica_PPI	7,59

Destacam-se os atributos oriundos do Sala de Aula – "Q2 - Tempo envio", "S1 - Tempo ultimo acesso", "S2 - Tempo ultimo acesso", "Q1 - Tempo envio", "Q2 - Nota" e "Q1 - Nota" –, com pontuação bem maior que os demais atributos cadastrais e de nota oriundos do SUAP. Na sequência, tem-se a informação da escola de origem do estudante com grande peso e diversas notas das disciplinas do ano anterior com pesos menores.

Para o modelo do segundo ano que possui atributos de notas do ano anterior, além dos atributos das notas e interação oriundos do Sala de Aula, segue a lista na Tabela 6.

Tabela 6 ▶
Atributos mais importantes – modelo segundo ano, segundo SelectKBest e chi quadrado.
Fonte: dados da pesquisa

Atributo	***Score***
Q2 - Tempo envio	2694,07
S1 - Tempo ultimo acesso	2001,40
Q1 - Tempo envio	1310,48
Q2 - Nota	655,60
Q1 - Nota	532,74
Escola de Origem	93,51
nota_Fisica_I	88,33

nota_Historia_I	77,12
nota_Matematica_I	72,84
nota_Lingua_Portuguesa_I	64,49
nota_Quimica_I	47,87
nota_Filosofia_I	44,48
nota_Biologia_I	28,79
nota_Geografia_I	19,48

Analisando os atributos verifica-se que os atributos oriundos do Sala de Aula – "Q2 - Tempo envio", "S1 - Tempo ultimo acesso", "S2 - Tempo ultimo acesso", "Q1 - Tempo envio", "Q2 - Nota" e "Q1 - Nota" – continuam com pontuação bem maior que os demais atributos cadastrais. Em seguida, tem-se a informação da escola de origem do estudante com grande peso, semelhante ao modelo anterior. Por fim, com menores pesos, observam-se informações do curso do estudante, informações sobre cotas e o tipo de escola de origem.

4 Considerações finais

Este trabalho propôs uma abordagem supervisionada para predizer de forma precoce os estudantes que terão desempenho acadêmico insuficiente em disciplina realizada em modalidade de educação on-line.

Para tal, foi construído um *dataset* com dados reais originados do Google Sala de Aula (não existe *dataset* com essa origem nos trabalhos relacionados) e do SUAP para o ensino médio integrado ao técnico, uma modalidade de ensino não estudada na literatura correlata.

Uma vez finalizado o conjunto de dados, o qual é considerado desbalanceado, foram construídos e avaliados modelos de aprendizado supervisionado usando seis diferentes algoritmos em cenários sem balanceamento e com seu tratamento. Os artefatos associados aos notebooks do Jupyter do trabalho encontram-se disponíveis de forma pública no Github[11] do autor.

Também foi objeto de análise a influência da seleção de atributos no desempenho final dos melhores preditores.

Os resultados demonstraram que o método de classificação SVM obteve valores para a especificidade superiores a 93% e AUC maiores que 87% para o modelo do segundo ano, o qual possui informações das notas de disciplinas cursadas no ano anterior, além das informações cadastrais do SUAP e de notas e interação no Google Sala de Aula. Já para o modelo do primeiro ano, o método *Gradient Boosting* atingiu valores de especificidade de 80% e AUC de 89%. Nesse último caso, não existe histórico de notas, uma vez que os estudantes estão cursando as disciplinas iniciais do ensino médio.

Ressalte-se que a obtenção desses resultados foi possível utilizando apenas informações das duas primeiras semanas de aula – notas das avaliações contínuas e envio das suas respostas, além das informações de acesso ao ambiente de aprendizagem por parte do

11 Disponível em: https://github.com/joaopaulopbjp/predicao_precoce_de_desempenho_academico . Acesso em: 14 out. 2020.

estudante nesse período. Assim sendo, torna-se possível uma predição extremamente precoce dos estudantes que poderão ter desempenho insuficiente com apenas duas semanas de aula e duas atividades avaliativas. Isso possibilita uma oportunidade relevante de atuação dos professores das disciplinas, das equipes pedagógicas e de gestão acadêmica para apoiar os estudantes com provável desempenho insuficiente por meio do monitoramento da evolução da sua aprendizagem e de estratégias específicas de apoio.

Outro destaque para os modelos construídos foi a sua simplicidade, tanto para o primeiro quanto para o segundo ano. Ambos apresentam modelos com bom desempenho preditivo com apenas 6 ou 10 atributos.

Os modelos construídos neste trabalho podem ser utilizados também em ambientes de sala de aula tradicional que usam a plataforma de sala de aula virtual do Google como apoio das atividades da disciplina.

Como trabalhos futuros, o conjunto de dados será estendido contemplando dados de outros *campi* do IFPB, bem como será analisada uma maior quantidade de atividades avaliativas, envolvendo um bimestre inteiro. Isso promoverá a verificação mais incisiva da capacidade de generalização do modelo. Pode ser incluída também a métrica F2 Score para possibilitar a comparação dos modelos apresentados com aqueles dos trabalhos relacionados. Adicionalmente, outras estratégias de tratamento de desbalanceamento poderão ser verificadas. Por fim, a análise dos hiperparâmetros dos métodos de classificação pode promover uma melhora nos resultados.

Referências

ALBON, C. **Machine Learning with Python cookbook**: practical solutions from preprocessing to Deep Learning. 1. ed. Sebastopol: O'Reilly Media, 2018.

BARROS, R. P.; SANTANA JUNIOR, O. V.; SILVA, I. R. M.; SANTOS, L. F.; CÂMARA NETO, V. R. Predição do rendimento dos alunos em lógica de programação com base no desempenho das disciplinas do primeiro período do curso de ciências e tecnologia utilizando técnicas de mineração de dados. **Brazilian Journal of Development**, v. 6, n. 1, p. 2523-2534, 16 jan. 2020. DOI: https://doi.org/10.34117/bjdv6n1-186.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5-32, 2001. DOI: https://doi.org/10.1023/A:1010933404324.

CANO, A.; LEONARD, J. D. Interpretable Multiview Early Warning System Adapted to Underrepresented Student Populations. **IEEE Transactions on Learning Technologies**, v. 12, n. 2, p. 198-211, 1 abr. 2019. DOI: https://doi.org/10.1109/TLT.2019.2911079.

CHANLEKHA, H.; NIRAMITRANON, J. Student performance prediction model for early-identification of at-risk students in traditional classroom settings. *In*: INTERNATIONAL CONFERENCE ON MANAGEMENT OF DIGITAL ECOSYSTEMS, 10., 2018, Tokyo. **Proceedings** (...). New York: Association for Computing Machinery, 2018. DOI: https://doi.org/10.1145/3281375.3281403.

CHAPMAN, P.; CLINTON, J.; KERBER, R.; KHABAZA, T.; REINARTZ, T.P.; SHEARER, C.; WIRTH, R. **CRISP-DM 1.0**: step-by-step data mining guide. [*S.l.*]: CRISP-DM Consortium, 2000. Disponível em: https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP-0800.pdf. Acesso em: 30 jan. 2021.

CHEN, T.; GUESTRIN, C. **XGBoost:** A Scalable Tree Boosting System. *In*: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 22., 2016, San Francisco. **Proceedings** (...). New York: ACM, 2016. DOI: https://doi.org/10.1145/2939672.2939785.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273-297, set. 1995. DOI: https://doi.org/10.1007/BF00994018.

FERRARI, D.; CASTRO, L. **Introdução à mineração de dados**: conceitos básicos, algoritmos e aplicações. 1. ed. São Paulo: Saraiva, 2016.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, v. 29, n. 5, p. 1189-1232, 2001. DOI: https://doi.org/10.1214/aos/1013203451.

GUO, G.;WANG, H.; BELL, D.; BI, Y.; GREER, K. KNN model-based approach in classification. *In*: MEERSMAN, R.; TARI, Z.; SCHMIDT, D. C. (ed.). **On The Move to Meaningful Internet Systems 2003**: CoopIS, DOA, and ODBASE. OTM 2003. Berlin: Springer, 2003. (Lecture Notes in Computer Science, v. 2888). DOI: https://doi.org/10.1007/978-3-540-39964-3_62.

HARRINGTON, P. **Machine Learning in action**. Shelter Island: Manning Publications, 2012.

HE, H.; MA, Y. **Imbalanced learning**: foundations, algorithms, and applications. IEEE. New Jersey: John Wiley & Sons, 2013.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. **arXiv preprint**, arXiv:1302.4964, 2013. DOI: https://doi.org/10.48550/arXiv.1302.4964.

MANISH, P. Handling categorical data in Python tutorial. **Datacamp**, 5 jan. 2020. Disponível em: https://www.datacamp.com/community/tutorials/categorical-data. Acesso em: 30 jan. 2021.

MOHAN, A. Cross-validation for imbalanced datasets. **Lumiata**, 5 mar. 2019. Disponível em: https://medium.com/lumiata/cross-validation-for-imbalanced-datasets-9d203ba47e8. Acesso em: 12 dez. 2020.

PIMENTEL, M.; ARAUJO, R. #FiqueEmCasa, mas se mantenha ensinando-aprendendo: algumas questões educacionais em tempos de pandemia. **SBC Horizontes**, 30 mar. 2020. Disponível em: http://horizontes.sbc.org.br/index.php/2020/03/fiqueemcasa/. Acesso em: 3 set. 2020.

RABELO, H.; BURLAMAQUI, A.; VALENTIM, R.; RABELO, D. S. S.; MEDEIROS, S. Utilização de técnicas de mineração de dados educacionais para predição de desempenho de alunos de EaD em ambientes virtuais de aprendizagem. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)**, Anais do SBIE 2017 (Proceedings of the SBIE 2017), p. 1527-1536, out. 2017. DOI: http://dx.doi.org/10.5753/cbie.sbie.2017.1527.

ROMERO, C.; VENTURA, S. Educational data mining and learning analytics: an updated survey. **WIREs Data Mining and Knowledge Discovery**, v. 10, n. 3, e1355, 2020. DOI: https://doi.org/10.1002/widm.1355.

RONALDO, G.; PASSOS, E.; BEZERRA, E. **Data Mining**: conceitos, técnicas, algoritmos, orientações e aplicações. 2. ed. Rio de Janeiro: Elsevier, 2015.

SANTOS, E. **Educação online para além da EAD:** um fenômeno da cibercultura. *In*: CONGRESSO INTERNACIONAL GALEGO-PORTUGUÊS DE PSICOPEDAGOGIA, 10., 2009, Braga, Portugal. **Actas** (...). Braga: Universidade do Minho, 2009. p. 5658-5671. Disponível em: https://www.educacion.udc.es/grupos/gipdae/documentos/congreso/xcongreso/pdfs/t12/t12c427.pdf. Acesso em: 11 dez. 2020.

SANTOS, M. S.; SOARES, J. P.; ABREU, P. H.; ARAUJO, H.; SANTOS, J. Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [Research Frontier]. **IEEE Computational Intelligence Magazine**, v. 13, n. 4, p. 59-76, 1 nov. 2018. DOI: https://doi.org/10.1109/MCI.2018.2866730.