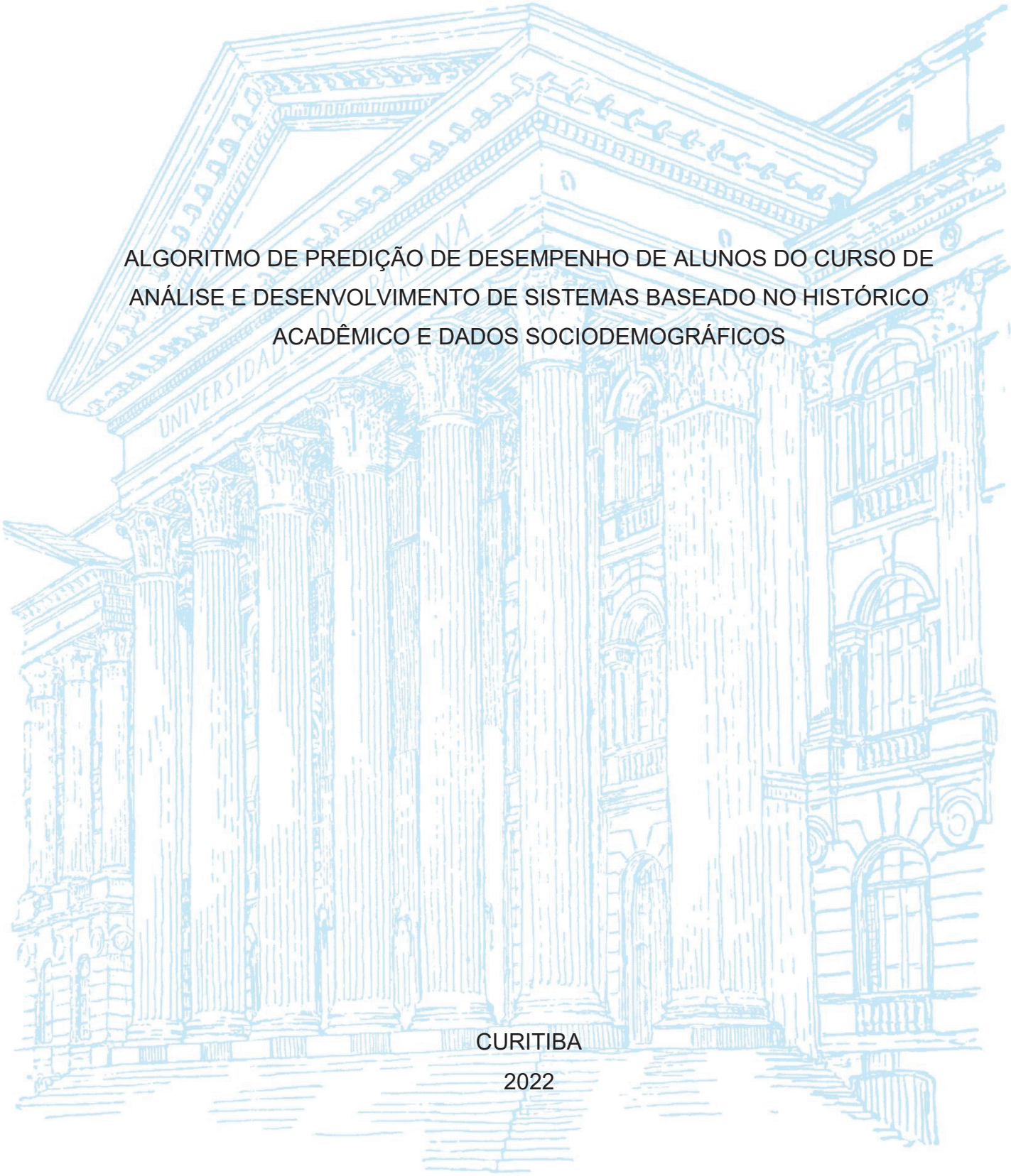
UNIVERSIDADE FEDERAL DO PARANÁ

CAMILA SCOTTI PINTO



ALGORITMO DE PREDIÇÃO DE DESEMPENHO DE ALUNOS DO CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS BASEADO NO HISTÓRICO ACADÊMICO E DADOS SOCIODEMOGRÁFICOS

CURITIBA

2022

CAMILA SCOTTI PINTO

ALGORITMO DE PREDIÇÃO DE DESEMPENHO DE ALUNOS DO CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS BASEADO NO HISTÓRICO ACADÊMICO E DADOS SOCIODEMOGRÁFICOS

Monografia apresentada ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Alexander Robert Kutzke

CURITIBA

2022

Algoritmo de Predição de Desempenho de Alunos do Curso de Análise e Desenvolvimento de Sistemas Baseado no Histórico Acadêmico e Dados Sociodemográficos

Camila Scotti Pinto
Especialização em Inteligência Artificial Aplicada
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil
camila.scotti@ufpr.br

Alexander Robert Kutzke
Especialização em Inteligência Artificial Aplicada
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil
alexander@ufpr.br

***Resumo*—Recentemente, tem havido um aumento no uso de ferramentas de *data mining* na extração de informações relevantes de bases de dado educacionais, sendo que tal processo recebe o nome de *Educational Data Mining*. Considerando a grande incidência de evasão nas universidades públicas brasileiras, existe uma grande importância na identificação precoce de alunos que possuíssem um risco de evasão. Tendo em vista o grande efeito de reprovações no aumento da probabilidade de evasão, este artigo traz a comparação de diferentes abordagens na identificação de alunos que possuem o maior risco de reprovar em determinada disciplina. Os seguintes métodos foram implementados em uma base real de alunos da UFPR: *Support Vector Machines*, Redes Neurais Profundas e *Random Forests*. Todos os algoritmos foram testados em uma base contendo unicamente os dados de histórico acadêmico e outra contendo os dados de histórico e dados socioeconômicos a fim de identificar as variáveis de maior relevância para o aumento no risco de reprovação. De todos os métodos mostrados, o *Random Forest* utilizando dados de histórico escolar e dados socioeconômicos mostrou a melhor performance para o problema apresentado.**

***Keywords*—educational data mining, machine learning, support vector machines, random forests, deep learning, redes neurais.**

***Abstract*—Recently, there has been a rise on the use of data mining to extract relevant information from educational data, such a process being called Educational Data Mining. Considering the current pace of college dropout in Brazilian public universities, there is a great importance on early identification of students with a risk of dropping out. Considering the great effect of class failure on increasing the probability of dropout, this paper compares different approaches in order to identify the students with a greater risk of failing a class. The proposed methods have been tested on a real world data set of UFPR students: Support Vector Machines, Deep Learning Neural Networks and Random Forests. All the algorithms have been tested with the grade history data and the socioeconomic data for a student in order to identify the most important variables for increasing the risk of class failure. Of all methods shown, the random forest using historical and socioeconomic data had the best performance for the task proposed.**

***Keywords*—educational data mining, machine learning, support vector machines, random forests, deep learning, neural networks.**

I. Introdução

A. Contextualização

A educação de forma geral e, especialmente, a de nível superior, se mostra essencial para o bom progresso de uma nação, uma vez que permite um maior desenvolvimento de ciência, tecnologia e inovação [1]. Além disso, a educação de nível superior aumenta a renda da população, uma vez que há um aumento de 258% na renda em comparação com aqueles que não concluíram o ensino médio. [2] Porém, o ensino superior brasileiro enfrenta diversos problemas atualmente, entre eles a escassez de docentes qualificados, falta de investimentos e altos níveis de evasão [1].

Estima-se que, entre 2019 e 2020, 35% dos alunos tenham desistido ou abandonado a faculdade. O que representaria um aumento de pouco mais de sete pontos percentuais em comparação com o ano anterior. Tal fenômeno pode culminar em um apagão de mão de obra qualificada nos próximos anos no país [3]. Além disso, leva a um desperdício de dinheiro público - quando se trata de universidades públicas - [4] e contribui para a perpetuação das desigualdades sociais [5].

B. Problema

Dado o contexto apresentado, existe uma grande importância em se identificar antecipadamente quais dos estudantes estão sob risco de abandonar o curso. Tal dado tornaria possível para os gestores acadêmicos (entre eles, reitores e coordenadores de curso) tomar atitudes para mitigar esse risco [6]. Além do histórico acadêmico, dados socioeconômicos aparentam ter uma grande influência sobre a permanência de um aluno na faculdade. Estudos mostram que entre os fatores que contribuem para a evasão estão reprovações, dificuldades para pagamentos de mensalidades, sexo e idade [7]. Portanto o presente trabalho assume que ao se detectar e atuar antecipadamente em uma reprovação será possível reverter um quadro futuro de evasão [7] [8].

C. Hipóteses

A principal hipótese na qual o presente trabalho se baseia é a de que é possível prever uma reprovação a partir das notas obtidas em matérias anteriores de um aluno. Uma hipótese adicional que será testada é a de que dados sociodemográficos possam contribuir para a identificação de estudantes sob risco de reprovação. Ou seja, de que não só as notas de um aluno, mas a sua situação pessoal e familiar possam significar que este aluno necessite de um apoio específico proveniente da equipe pedagógica.

D. Objetivo Geral

O presente trabalho possui como principal objetivo a comparação entre vários algoritmos de classificação na identificação precoce do risco de um aluno reprovar em determinada disciplina.

E. Objetivos Específicos

- Utilizar dados de histórico acadêmico para predizer a possibilidade de reprovação em uma disciplina antes desta ser cursada;
- Comparar três diferentes algoritmos de classificação - *Support Vector Machines*, Redes Neurais Profundas e *Random Forest* - considerando valores de acurácia, precisão e *recall*;
- Adicionar à mesma base dados socioeconômicos e verificar influência destes sobre a qualidade da predição de desempenho acadêmico em determinadas disciplinas.

F. Metodologia

Dados relativos à nota e frequência em disciplinas cursadas anteriormente serão utilizados a fim de realizar a predição de sucesso ao se cursar a disciplina. Após isso, serão adicionados dados socioeconômicos ao algoritmo com o objetivo de verificar a correlação entre a situação pessoal do aluno com a possibilidade de sucesso em determinadas disciplinas.

Foram testados os algoritmos de *Support Vector Machine*, Redes Neurais Profundas e *Random Forest* em conjunto com diferentes bases de dados a fim de se obter a maior acurácia de predição possível.

G. Resultados Esperados

Considerando a importância de um bom desempenho acadêmico para retenção de um aluno no curso universitário, acredita-se que uma predição apropriada da possibilidade de um aluno reprovar em dada disciplina de forma antecipada poderá desencadear em ações dos gestores acadêmicos a fim de impedir que tal fato ocorra. Isto, aliado a atitudes de apoio a estudantes em situações socioeconômicas de risco poderá auxiliar na diminuição das altas taxas de evasão universitária.

H. Organização do Artigo

O artigo foi dividido em quatro seções, incluindo esta seção introdutória. A segunda seção faz um levantamento teórico a cerca do processo de descobrimento em bases de dados, incluindo métodos de pré-processamento e diferentes algoritmos de classificação. A terceira sessão foca em descrever as bases de dados utilizadas e a aplicação das técnicas mostradas. Por fim, a quarta sessão apresenta os resultados dos experimentos e compara os resultados entre os mesmos.

II. Fundamentação Teórica

Essa seção tem como objetivo explorar os principais temas relacionados a este artigo e foi dividida em nove subseções. A subseção A é uma explicação geral sobre as diferentes etapas da extração de conhecimento em bases de dados. A subseção B discute especificamente uma das etapas da extração de conhecimento em bases de dados, que é o pré-processamento de dados. Já a subseção C traz um panorama geral sobre aprendizado de máquina, sendo que as subseções D, E e F explicam de forma aprofundada três algoritmos de classificação: redes neurais, *Support Vector Machines* e *Random Forests* respectivamente. A subseção G mostra diferentes métricas para aferição da qualidade dos resultados de um algoritmo de classificação. A subseção H traz um panoramara geral sobre os conceitos de *Educational Data Mining* e *Learning Analytics*. E, por fim, a subseção I traz o estado da arte atual para predição de desempenho estudantil utilizando *Educational Data Mining*.

A. Descoberta de conhecimento em Bases de Dados

A partir dos avanços da capacidade de armazenagem de grandes bases de dados, surgem diversas oportunidades de utilizar esse volume de dados em benefício das instituições. Porém, isso se torna inviável sem o auxílio de ferramentas computacionais adequadas. Logo, a partir disso, surgiu uma área denominada descoberta de conhecimento em base de dados ou KDD, sigla em inglês para *Knowledge Discovery in Databases* [9].

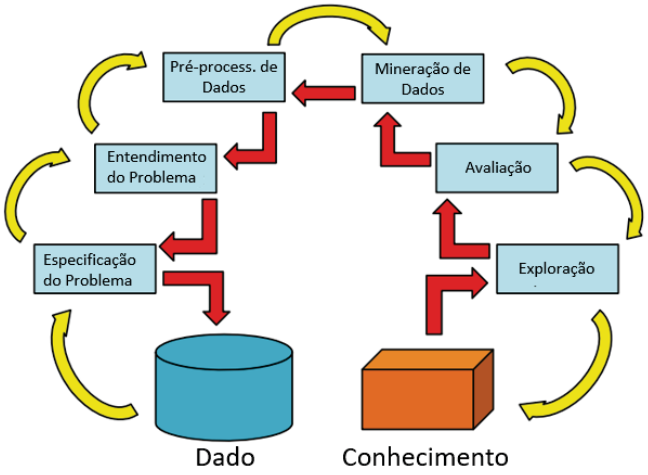
Segundo Garcia et al. [10], existem 6 passos no processo de KDD, sendo eles:

- Especificação do problema: Designação da área de aplicação, qual o conhecimento relevante já levantado por especialistas na área e objetivo final buscado pelo usuário;
- Entendimento do problema: Inclui o conhecimento tanto dos dados utilizados quanto o conhecimento especializado associado. Tal postura permite uma maior confiabilidade dos resultados obtidos posteriormente;
- Pré-processamento de dados: Inclui a limpeza, integração, transformação e redução dos dados;
- Mineração de dados: Nesta fase são extraidos os padrões de dados relevantes. Esta etapa também inclui a seleção do método de mineração de dados mais relevante, sendo as opções: classificação, regressão, clusterização e associação;
- Avaliação: Interpretação dos resultados a partir de medidas de interesse;
- Exploração dos resultados ou incorporação do conhecimento: Este estágio consiste na utilização direta do conhecimento adquirido. Pode-se incluir a utilização do conhecimento obtido e outro sistema para obtenção de

novos conhecimentos ou reporte deste através de ferramentas de visualização.

A Fig. 1 resume o processo de KDD a partir destes seis estágios. Deve-se frisar que não apenas todos os estágios estão interconectados mas que o processo reverso também é possível. [10]

Figura 1. Processo de KDD. Adaptado de GARCIA, S. et al [10]



B. Pré-Processamento de Dados

Para ser possível a retirada de dados relevantes de uma base de dados e garantir uma implementação bem sucedida das técnicas de *data mining* é necessário certificar que os dados utilizados estão corretos e íntegros. Em aplicações reais, existe uma dificuldade em garantir a integridade dos dados quando estes não estão completos ou possuem muitos dados que se diferenciam fortemente dos demais presentes na amostra (também conhecidos como *outliers*). [11] Tais tipos de dados podem ser chamados de dados ruins ou contaminados. [12]

A limpeza de dados é uma das partes principais do pré-processamento de dados. Esta é aplicada para remoção de ruídos, correção de inconsistências e de problemas relacionados a dados faltantes. Os dados faltantes podem ser originários de problemas nos equipamentos de medição, medidas realizadas incorretamente, esquecimento no momento do preenchimento dos dados, perda de informação e também por erros humanos. Em casos de questionários, a falta de dados pode-se dar pelo esquecimento de fazer uma pergunta ou registrar a resposta [13]. A forma mais comum de se lidar com esses dados faltantes é a deleção dos mesmos, omitindo as instâncias consideradas faltosas. Porém, tal método possui uma grande desvantagem ao reduzir a base de dados [14].

Em vez da remoção das variáveis ou das observações, é possível completar esses dados faltantes com valores plausíveis [13]. Entre as várias técnicas existentes, uma relativamente recente é a de *multiple imputation*, que leva em consideração a incerteza existente nas estimativas dos dados. Este método consiste em gerar múltiplos *datasets* dos quais os parâmetros de interesse podem ser estimados. Os parâmetros dos diferentes *datasets* são combinados, dando um estimativa do valor do parâmetro para os *datasets* como um todo. Logo, há uma menor chance de erro em comparação com quando os dados são estimados apenas uma vez [12] [15].

Outra situação que pode ser tratada no pré-processamento de dados é relativa às bases de dados desbalanceados. Estas ocorem quando existe uma quantidade muito maior de instâncias em uma classe do que nas demais. Especificamente, no caso de apenas duas classes, o problema acontece quando uma destas apresenta um grande percentual das instâncias (sendo chamada de classe majoritária) e a outra classe possui apenas um pequeno percentual dos dados totais (sendo chamada de classe minoritária). O *imbalance ratio*, traduzido como "taxa de desbalanceamento" é definido como a proporção entre a classe majoritária e a minoritária. [16]

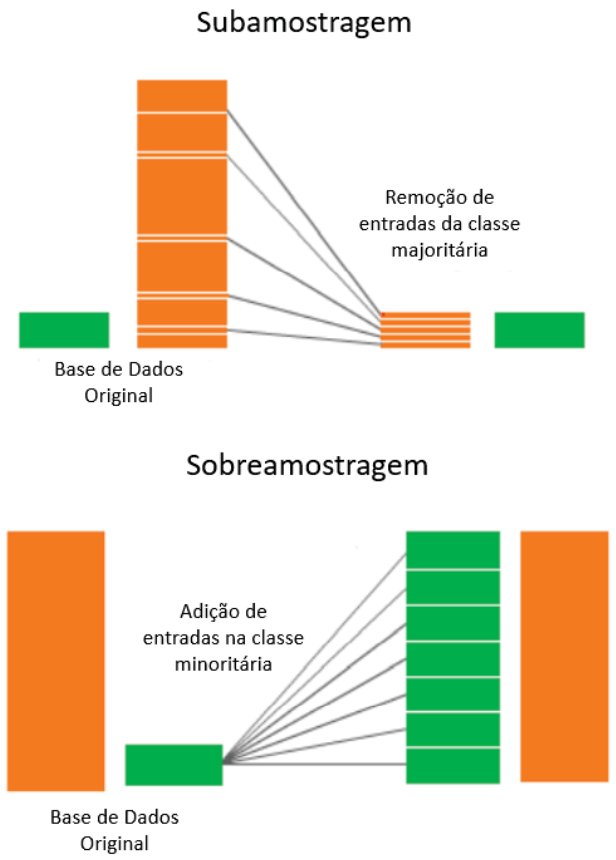
A grande maioria dos algoritmos de aprendizado de máquina tradicionais assumem que todas as classes apresentadas para o algoritmo terão uma quantidade similar de entradas, porém este fato não costuma ser verdadeiro. Entre diversas situações onde um desbalanceamento pode acontecer, tem-se como exemplo o diagnóstico de doenças e identificação de fraudes, casos em que se espera que a maior parte dos dados sejam pertencentes à uma única classe [17].

Com a influência da grande quantidade de dados da classe majoritária, os algoritmos de classificação tradicionais acabam falhando em encontrar instâncias pertencentes às classes de dados minoritárias [16]. Uma das formas mais comuns de se lidar com esse problema é com a utilização de técnicas de reamostragem. Esta técnica pode ser aplicada com subamostragem ou sobreamostragem. A subamostragem se trata de quando se diminui a quantidade de dados de uma das categorias, enquanto a sobreamostragem acontece quando se aumenta a quantidade de dados na classe minoritária. A Fig. 2 traz uma ilustração da diferença entre a subamostragem e sobreamostragem [17].

Ambos os métodos de subamostragem e sobreamostragem trazem uma maior exigência computacional, o que pode vir a ser significativo quando se trata de uma extensiva base de treino. A subamostragem, apesar de ter se apresentado como um bom método para aumentar a sensibilidade do preditor, é um método que pode acabar descartando dados potencialmente úteis que poderiam vir a ser importantes para o processo de treinamento [17]. A sobreamostragem, como pode-se ver na Fig. 2, trabalha aumentando quantidade de dados ao se criar entradas sintéticas baseadas nas entradas reais pré-existentes. Esta técnica tem mostrado resultados muito superiores aos obtidos pela técnica de subamostragem, uma vez que nenhum dado é perdido [18].

Grande parte das técnicas de predição em aprendizado de máquina se baseiam na utilização de variáveis quatitativas para aplicação dos métodos matemáticos. Porém, muitas vezes é necessário entender como diferentes etnias, gêneros e regiões influenciam em comportamentos sociais. Em vez de transformar estas categorias em uma escala de valores, são utilizadas variáveis *dummy* para representar cada uma das categorias e

Figura 2. Diferença entre Subamostragem e Sobreamostragem, adaptado de MOHAMED et al. [17]



incluir dados qualitativos nas predições [19].

C. Aprendizado de Máquina

O aprendizado de máquina, ou do inglês, *Machine Learning* pode ser descrito a partir da seguinte definição:

> Um programa de computador aprende pela experiência E com respeito a uma classe de tarefas T e medida de performance P se a sua performance nas tarefas T, conforme medido por P, melhora com a experiência E. [20]

Duas abordagens comuns de aprendizado de máquina são de aprendizado supervisionado e não supervisionado. No aprendizado supervisionado, o algoritmo recebe uma tupla contendo o valor de entrada e o resultado esperado. Após várias iterações, o algoritmo se torna capaz de predizer resultados a partir de novos dados de entrada. O aprendizado supervisionado pode ser aplicado em algoritmos de classificação com categorias pré-estabelecidas, além de reconhecimento de fala e de imagens [21].

No aprendizado não supervisionado não existe um resultado explicito pré-determinado que o algoritmo deve alcançar. Este método geralmente é utilizado quando há a necessidade de agrupar elementos de entrada em função de seus atributos. O modelo de aprendizado não supervisionado é importante por ser muito mais comum no cérebro que o modelo supervisionado. Entre as possíveis aplicações, tem-se segmentação de objetos e rotulagem automática [21] [22].

Os algoritmos de classificação, considerados como algoritmos de aprendizado supervisionado, são utilizados para resolver o problema de se assinalar uma observação x' em uma classe qualitativa denominada y', assumindo-se a existência de um número finito de classes. Tal classificação é feita a partir de uma base de dados denominada por $D = (x_1, y_1), ..., (x_n, y_n)$ para dados de treino x_i que possuem uma classificação y_i pré estabelecida. Na maior parte destes problemas não é possível determinar uma relação matemática entre x_i e y_i, por isso a relação tem que ser representada por uma distribuição probabilística. Estes algoritmos podem ser aplicados em diversas áreas, incluindo a área empresarial, de engenharia, medicina [23] [24] e educação.

Existem duas abordagens padrões para classificação de dados. Na primeira, na qual há distinções claras entre as classes, sendo que cada dado é categorizado entre elas. Já a segunda abordagem determina uma probabilidade para o dado pertencer a uma determinada classe. *Support Vector Machines* são o exemplo mais conhecido da primeira abordagem. Já para a segunda abordagem existem vários métodos populares como Redes Neurais, Regressão Logística, *k-nearest neighbors* e *Decision Trees*, sendo estes que tem diferenças consideráveis na forma como determinam a aproximação do modelo de probabilidade [24].

A partir da análise recente de diversos artigos publicados referentes especificamente a algoritmos de predição de desempenho estudantil, foi concluído que as *Support Vector Machines* são as mais amplamente utilizadas e as que trazem os melhores resultados. Além das *Support Vector Machines*, *Decision Trees*, *Naive Bayes* e *Random Forests* são propostas de algoritmos muito estudadas que têm gerado bons resultados [25].

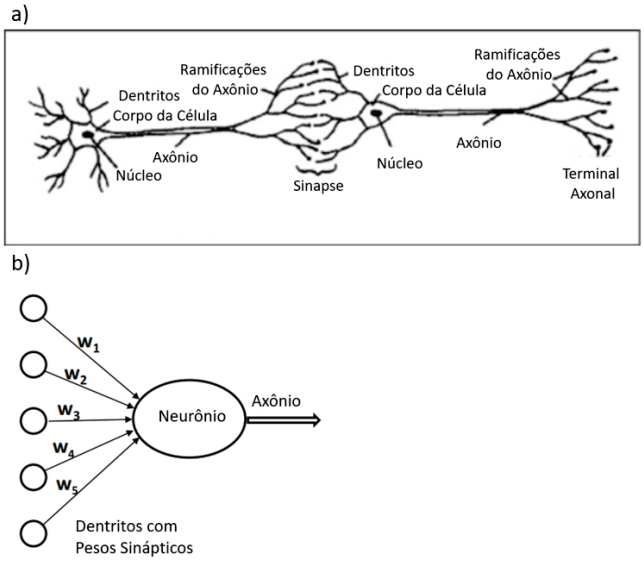
Devido à crescente complexidade do problema de classificação, vários novos algoritmos tem sido desenvolvidos recentemente, entre eles se destacam *Extreme Machine Learning*, Classificação baseada em representação esparsa e redes neurais profundas [23]. Neste artigo são utilizados e comparados os algoritmos de Redes Neurais, *Support Vector Machines* e *Random Forest* para a tarefa. Sendo que para Redes Neurais, foram utilizadas Redes Neurais Profundas.

D. Redes Neurais

As redes neurais artificiais (RNAs) foram inspiradas na anatomia neural dos seres vivos. As células do sistema nervoso humano são chamadas de neurônios e estes são conectados entre si através de axônios e dentritos. Quando estes são conectados ocorrem as sinapses. Nas RNAs, existem unidades computacionais que correspondem aos neurônios e conexões que correspondem às sinapses. Estas conexões possuem pesos, o que afeta a função computada naquela unidade. Na Fig. 3

é possível visualizar a estrutura biológica de um neurônio e a sua representação em uma RNA [26] [27].

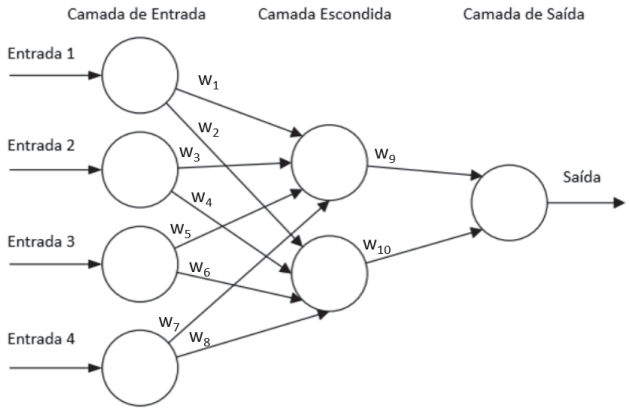
Figura 3. Representação das conexões sinápticas entre neurônios. a) Neurônio real b) Rede Neural Artificial. Adaptado de AGGARWAL, C. [27]



As RNAs são organizadas em camadas, sendo que cada camada possui uma ou mais unidades computacionais. O número de camadas em uma rede neural é referido como a profundidade desta e o número de unidades computacionais por camada é referido como a sua largura [28].

A estrutura básica das RNAs pode ser vista na Fig. 4. O dado de entrada geralmente consiste de um vetor multidimensional, cujos parâmetros são distribuidos para as camadas escondidas. A camada escondida faz decisões baseadas na camada anterior e pondera o quanto uma mudança estocástica nela irá impactar o resultado final, sendo este chamado de processo de aprendizado. [29].

Figura 4. Estrutura básica de uma RNA. Adaptado de SHEA et al. [29]



Considera-se que a camada de entrada possua n unidades computacionais que transmitam um número n de parâmetros $\overline{X} = [x_1, x_2, ..., x_n]$ com pesos $\overline{W} = [w_1, w_2, ..., w_n]$ para a camada seguinte. A cada camada é calculada função linear $\overline{W} \cdot \overline{X} = \Sigma_{i=1}^{d} w_i x_i$ e aplicada uma função de ativação para se ter um resultado de saída que será utilizado como entrada para a camada seguinte. O valor de saída, denominado por $\overline{y}$, é o resultado entregue pela última camada do modelo [27] [30].

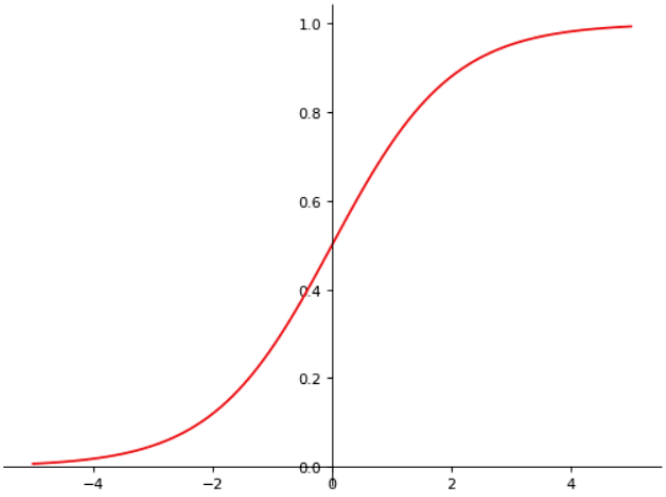
Para se treinar um modelo de RNA, é fornecido um conjunto de instâncias no formato $(\overline{X}, y)$ sendo que y é um valor entre -1 e 1, correspondente ao resultado esperado. A diferença entre o valor predito e o real pode ser denominado por $E(\overline{X}) = y - \overline{y}$ e a cada iteração do treinamento os valores dos pesos são ajustados para que esta diferença se torne a menor possível [27].

A função $E(\overline{X})$ recebe o nome de função de perda e dependendo do caso e da natureza do resultado esperado, esta pode ser caculada de diferentes formas. A função de perda *binary cross-entrophy* mede a performance dos modelos de classificação quado há apenas duas possibilidades de saída (resposta binária) e esta pode ser representada em probabilidades entre 0 e 1. Já a função de perda *cross-entrophy* é utilizada quando existem múltiplas classes [28].

Três das principais funções de ativação para uma unidade computacional em uma rede neural são: Sigmóide, Unidade Linear Retificada (ReLU) e *Softmax*. A função de ativação sigmóide é a mais amplamente utilizada para problemas de classificação com duas classes, e pode ser representada pela Equação 1 e Fig. 5. [28] [30]

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

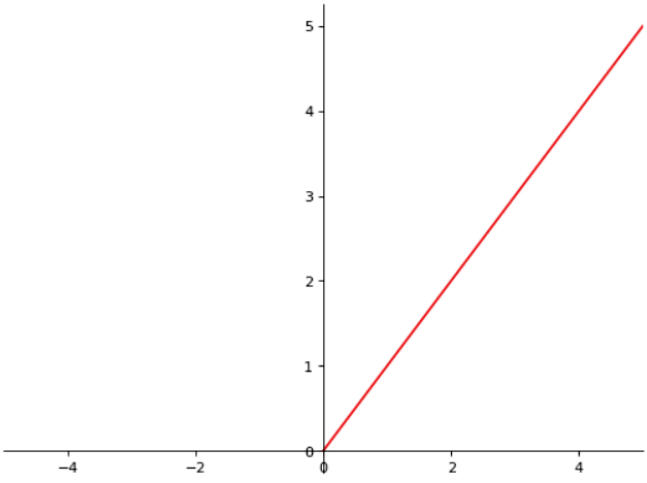
Figura 5. Função de ativação Sigmóide



A função ReLU costuma ser mais amplamente utilizada nas camadas intermediárias [28]. A sua vantagem é de que nem todos as unidades computacionais são ativadas ao mesmo tempo. Esta pode ser representada pela Equação 2 e pela Fig. 6 [28] [30].

$$f(x) = max(0, wx + b) \tag{2}$$

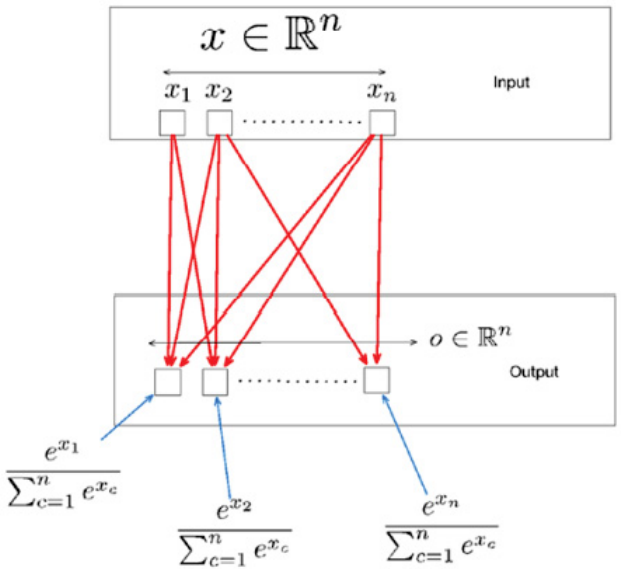
Figura 6. Função de ativação ReLU



Já a função de ativação *Softmax* costuma ser utilizada para a camada de saída e a normaliza com a função de perda de *cross-entropy*. A função retorna uma probabilidade conforme indicado na Equação 3. A Fig. 7 consegue exemplificar a equação, sendo que nesta o valor de j corresponde a cada uma das diferentes classes do modelo [28] [30].

$$\sigma(z)_i = \frac{e^{z_j}}{\Sigma_{c=1}^{n} e^{z_c}} \; para \; j = 1, ..., n \tag{3}$$

Figura 7. Função de ativação *Softmax*. Extraido de SKETKAR, N. et al. [28]



Por fim, uma camada muito comum em redes neurais é a camada de *dropout*. Esta retira aleatoriamente unidades computacionais da camada anterior, tendo como objetivo impedir o *overfitting* [28] [30]. O fenômeno de *overfitting* pode ser determinado quando o algoritmo se ajusta levando em consideração os ruídos e peculiaridades dos dados de treinamento, em vez de encontrar uma regra de predição geral [31].

De forma geral, uma rede neural funciona mapeando a Equação 4, sendo que o mapeamento da função G(x) é feito durante a fase de treinamento, na qual a rede aprende a associar corretamente os dados de entrada u com os dados de saída x. [32]

$$u = G(x) \tag{4}$$

Uma RNA é considerada uma Rede Neural Profunda, do inglês *Deep Neural Network* (DNNs) quando existem ao menos 4 camadas de unidades computacionais, incluido as camadas de entrada e saída [33]. Estas têm mostrado um grande poder de classificação devido às suas características não lineares, estruturas flexíveis e capacidades adaptativas. São consideradas como estado-da-arte e muitas vezes têm se mostrado capazes de obter uma resposta quase humana em diversas tarefas de reconhecimento de padrões, como reconhecimento de imagem e de fala [33] [34] [35].

E. Support Vector Machines

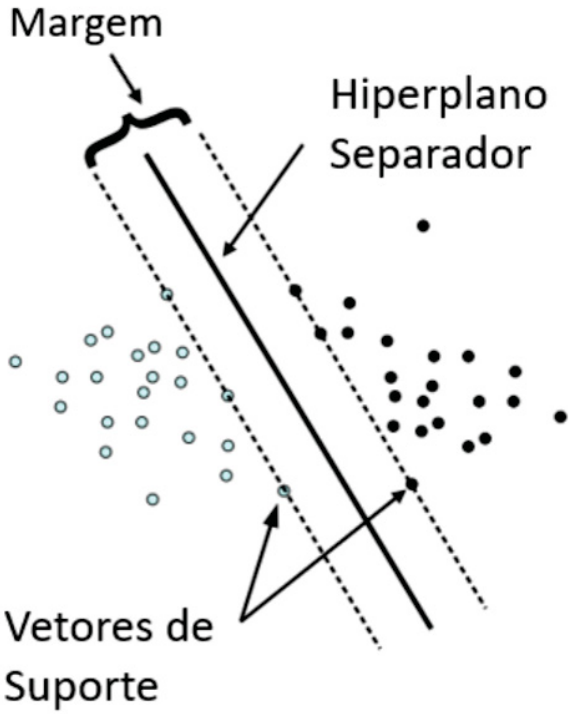
As *Support Vector Machines* ou SVMs foram primeiramente desenvolvidas em 1995 para classificação binária. Entre as suas diversas aplicações está a área médica [36], recuperação de imagens [37] e documentos [38] e sistemas de segurança [39]. Este pode ser considerado como um algoritmo na intersecção entre o teórico e o prático, já que constrói modelos que são complexos para reconhecimento de padrões não lineares, regressão e extração de conhecimento, porém simples o bastante para serem analisados matematicamente como se fossem simples algoritmos lineares [40].

Seu funcionamento básico consiste em procurar uma separação ótima do hiperplano em duas classes distintas. Essa separação ótima é obtida ao se maximizar a distância entre os pontos mais próximos das duas classes. A localização destes pontos mais próximos é onde estariam situados os vetores de suporte, origem do nome do método [41].

As SVMs podem possuir margens rígidas ou margens suaves. As SVMs de margens rígidas definem limites lineares considerando que os dados sejam linearmente separáveis. Porém dado o fato que este é um caso raro em aplicações reais, nas SVMs de margens suaves é permitido que alguns dados possam violar a condição da fronteira [42]. O funcionamento de uma SVM de margens rígidas pode ser visualizado na Fig. 8.

No caso de dados que se encontram no "lado errado" da margem de separação das classes, há uma redução do seu peso para assim reduzir a sua influência, quando isso acontece o algortimo recebe o nome de SVM com margens suaves. Outra particularidade é quando não é possível encontrar um separador linear. Neste caso os dados são projetados em

Figura 8. Ilustração de uma SVM com margens rígidas. Adaptado de MEYER, D. [41]



um plano de dimensão superior até que os dados sejam efetivamente linearmente separáveis. [41]

Para o caso de SVMs com margens rígidas, considera-se T como um conjunto de treinamento com n dados $x_i \in X$ e seus rótulos $y_i \in Y$, sendo que X é o espaço de estado e $Y = -1, +1$ [42].

A partir disso, a função do hiperplano se dará pela Equação 5. Esta expressão deverá classificar posteriormente de forma correta, novos dados de entrada X, nos quais os valores de y são desconhecidos. O hiperplano canônico com relação ao conjunto T é aquele em que os valores de w e b são definidos de forma que exemplos mais próximos da Equação 5 satisfaçam a equação 6. Tal forma irá implicar na expressão 7 [40] [42].

$$f(x) = w \cdot x + b = 0 \tag{5}$$

$$\mid w \cdot x_i + b \mid = 1 \tag{6}$$

$$y_i(w \cdot x_i + b) - 1 \geqslant 0, \forall (x_i, y_i) \in T \tag{7}$$

Para que se tenha a maximização da margem, se dará como necessário recorrer em um problema de otimização representado pela a expressão 8 [42].

$$\underset{w\,b}{\text{minimizar}} \frac{1}{2} \mid\mid w \mid\mid^2 \tag{8}$$

F. Random Forest

Random Forest é um método de classificação inicialmente introduzido em 2001 [43]. As *Random Forests* podem ser utilizadas tanto para tarefas de classificação quanto para regressão, da mesma forma, os dados de entrada podem ser categóricos e contínuos, o que não acontece na maioria dos demais algoritmos de classificação. Na sua época de criação, o método se mostrou favorável quando comparado a outras opções do mesmo período e apresentando um ruído menor. Do inglês, seu nome pode ser traduzido para "árvores aleatórias" e, como o nome sugere, este algoritmo se dá por um conjunto de árvores de regressão (ou decisão) dependentes de um vetor de variáveis aleatórias [44].

As árvores de regressão utilizadas no modelo *Random Forest* separam o espaço de predição com uma sequência de partições binárias, cada uma ligada a uma variável específica. O nó raiz da árvore consiste da totalidade do espaço de estados e os nós que não possuem bipartições são chamados de "nós terminais". Cada nó não terminal se biparticiona em dois nós (visualmente, tendo um para a esquerda e um para a direita), de acordo com o valor de uma das variáveis de predição. Quando uma das variáveis de predição é contínua, é determinado um ponto específico de bipartição, dependendo se o valor da medição for maior ou menor que o ponto de bipartição, este irá para o nó da esquerda ou da direita. Cada bipartição é determinada ao se considerar todas as possibilidades para aquele nó e escolhendo a melhor opção seguindo um certo critério pré-estabelecido. [44]

Um exemplo de seu funcionamento pode ser visualizado na Fig. 9, na qual o conjunto de amostras foi particionado em 4 subgrupos para a classificação de 3 grupos [45]. Neste, o nó raiz é o que depende do valor de X_2, os dois pontos dependentes do valor de X_1 são nós não terminais e o conjunto possui no total 4 nós terminais. Percebe-se também que os pontos de bipartição neste caso são determinados por variáveis contínuas.

As *Random Forests* se mostram especialmente adequadas para problemas que dependem de um grande número de varáveis. Tal situação está cada vez mais comum, considerando a crescente capacidade de coletar e armazenar grandes volumes de dados. Para estes tipos de problemas, os metódos classicos de classificação costumam ficar sobrecarregados pelo grande volume de variáveis e isto acerreta em uma perda de performance [46].

G. Avaliação da Performance dos Algoritmos de Classificação

Após o treinamento do modelo, se faz necessária a utilização de dados com rótulos pré-conhecidos para validação deste e também para uma comparação com outros modelos. Para um problema binário, como o apresentado neste artigo, existem duas possibilidades tanto para a classificação gerada pelo algoritmo quanto para a classificação real: 0 ou 1 (ou positivo e negativo). [47] [48].

A matriz de confusão é um dos métodos mais comuns de verificação da qualidade da classificação do modelo. As quatro possibilidades de combinação para a matriz de confusão em

Figura 9. Ilustração de Árvore de Decisão. Partições do espaço (a) e estrutura da árvore de decisão (b).Extraído de LOH, W [45]

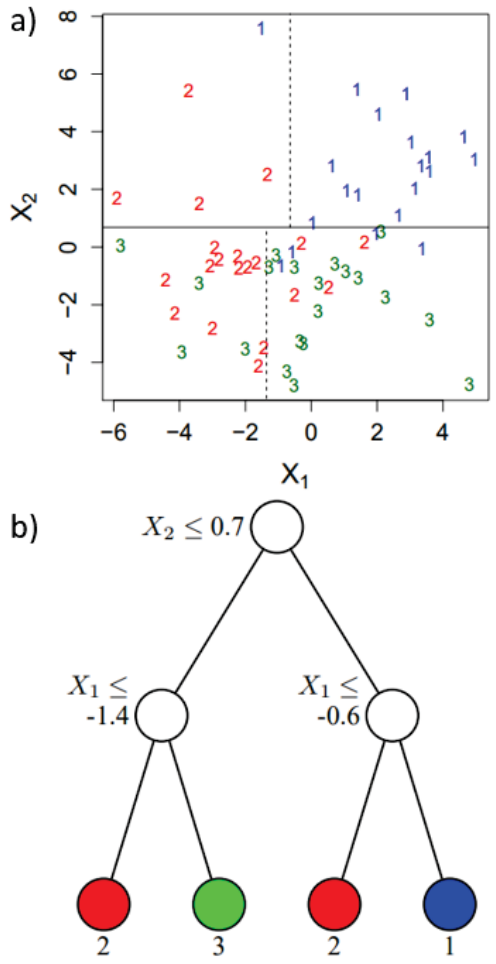
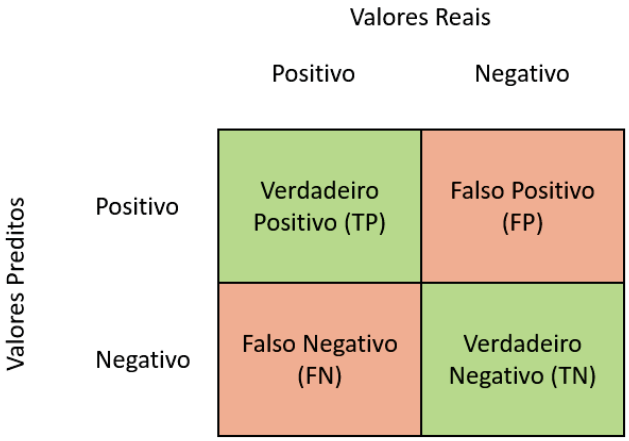


Figura 10. Contagem de Registros Tipo e Status da Matéria no Histórico, adaptado de THARWAT, A. [47]



um problema binário podem ser vistas na Fig. 10. Caso o resultado da classificação e o valor real do dado sejam positivos, este será considerado como um verdadeiro positivo ou TP. Caso ambos sejam negativos, este será considerado como um verdadeiro negativo ou TN. As classificações errôneas possíveis seriam um valor negativo classificado como positivo (erro tipo I ou alarme falso), recebendo a nomenclatura de FP, e um valor positivo classificado como negativo (erro tipo II) que recebe o nome de FN [47] [48].

O método empírico mais comum para se calcular a qualidade de um modelo é a acurácia. Este não foca em nenhuma classe em específico e pode ser considerada uma forma generalista de comparar algoritmos. O seu cáclulo pode ser encontrado na Equação 9. Em situações de classificações em que o resultado correto da predição de uma classe é de maior interesse que o resultado das demais, costuma se utilizar como métrica a precisão, *recall* e F-medida. Os cálculos para estas métricas estão respectivamente representados nas Equações 10, 11 e 12 [48]. Nas equações apresentadas, a classe de maior interesse para a classificação foi a positiva, caso o interesse esteja em se predizer a classe negativa, o valor de TP deverá ser substituido por TN.

$$acurácia = \frac{TP + TN}{TP + FP + FN + TN} \tag{9}$$

$$precisão = \frac{TP}{TP + FP} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

$$F - medida = \frac{(\beta^2 + 1) * precisão * recall}{\beta^2 * precisão + recall} \tag{12}$$

H. Educational Data Mining e Learning Analytics

Com o crescimento da utilização de recursos de *e-learning* e da manutenção de bases de dados virtuais com informações dos alunos, tem havido um aumento no volume de dados relativos à educação. Esta situação traz a necessidade de utilização de ferramentas para o tratamento destes dados e geração de conhecimento a partir destes que possam beneficiar os alunos [49].

A partir disto, tem-se visto recentemente um crescimento na adoção de processos de *Data Mining* para extração de informações relevantes a partir de dados educacionais. Tal processo recebe o nome de *Educational Data Mining* ou EDM. De acordo com o site da comunidade de EDM [50], esta é uma:

> disciplina emergente, preocupada em desenvolver métodos para explorar os dados únicos e cada vez em maior escala que vêm de ambientes educacionais e usar esses métodos para compreender melhor os alunos e os ambientes em que eles aprendem. [50]

Entre as diversas técnicas de EDM utilizadas, se destacam a associação, classificação, clusterização, busca de padrões e

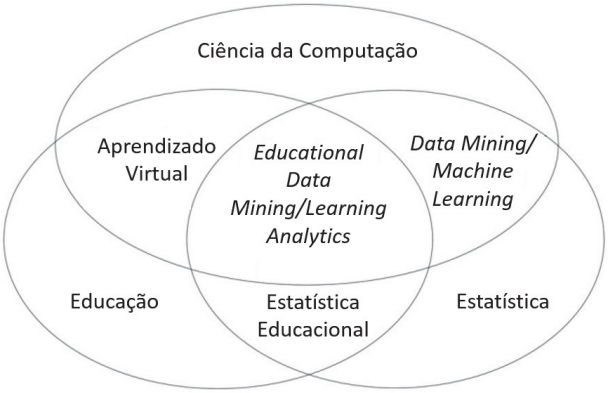
regressões. A utilização de tais técnicas trouxe uma alteração na forma como pesquisadores educacionais e instituições de ensino tem tomado decisões. Entre as diversas atitudes que a instituição de ensino pode tomar com base nesses dados inclui-se oferecer um suporte individual para o aluno que está com dificuldade [51] ou rever curriculo e ferramentas de ensino utilizadas para melhorar a atenção e engajamento dos estudantes.

Uma área correlacionada que têm visto o mesmo crescimento é a de *Learning Analytics* (LA), a qual pode ser definida como:

> Uma variedade de ferramentas coletoras de dados e técnicas analíticas para estudar o engajamento, desempenho e progresso dos alunos com o objetivo de usar o que é aprendido para revisar currículo, ensinamento e avaliação em tempo real [52].

Ambas as áreas de EDM e LA podem ser definidas como a combinação de 3 áreas, conforme mostrado na Fig. 11: ciência da computação, educação e estatística [49].

Figura 11. Principais áreas de conhecimento relacionadas a EDM e LA. Adaptado de ROMERO, C. [49]



I. Estado da Arte

Em 2020, Rastrollo-Guerrero et al. [25] fizeram um estudo extensivo utilizando 64 artigos diferentes de algoritmos de *Machine Learning* para predição de comportamento de alunos. Foram considerados majoritariamente artigos recentes, sendo que 90% dos que foram selecionados foram publicados nos 6 anos precedentes a escrita do mesmo. Foi relatado que há uma forte tendência na predição da performance de alunos no nível universitário, já que 70% dos artigos estudados se focam nesta etapa estudantil. Também é relatado que a técnica mais comum para predição de performance estudantil é a de aprendizado supervisionado, uma vez que as técnicas de aprendizado não supervisionado não tem mostrado bons resultados no setor de EDM. Entre as técnicas de aprendizado supervisionado, o SVM é algoritmo mais utilizado, seguido de *Decision Trees*, *Naive Bayes* e *Random Forests*. As redes Neurais Artificiais foram menos utilizadas, porém apresentaram uma grande precisão em suas predições.

Em 2017, Coelho et al. [53] publicaram um estudo a cerca da utilização de *Deep Neural Networks* em EDM, alegando a falta de estudos realizados neste tema na última década. Acredita-se que tal fenômono se deve à preferência da comunidade de *Educational Data Mining* por algoritmos mais conservadores [54]. Outro ponto negativo levantado pelos autores foi a dificuldade na interpretação dos resultados. Porém, os surpreendentes resultados positivos dos modelos de DNNs parecem compensar essa desvantagem.

Por fim, em 2020, Alyahian et. al [55] fizeram um estudo relativo às melhores praticas encontradas em artigos recentes sobre o tema. O escopo da análise foi de artigos publicados nos últimos 5 anos, devido à grande popularidade do tema de EDM. Entre as melhores práticas estão a remoção de *outliers*, o preenchimento ou deleção de dados faltantes, balanceamento de classes e seleção de atributos. O artigo cita que, quando não tratado, o problema de dados faltantes costuma impactar negativamente algoritmos de SVM e Redes Neurais, porém não costuma ter o mesmo resultado negativo para *Random Forests*.

O trabalho também cita os atributos mais populares que tiveram a maior taxa na determinação do sucesso de um estudantes. Entre os atributos que foram listados como sendo os mais relevantes para a predição de sucesso estudantil estão: gênero, matéria, período e tipo de programa. Muitos algoritmos têm usado também de dados coletados por plataformas online de aprendizado, como o tempo total dispendido na visualização de material e número de acessos ao sistema virtual de ensino disponibilizado pela instituição. [55]

III. Materiais e Métodos

Nesta seção há a descrição do método utilizado para detecção preliminar de reprovações de alunos. Na seção A são descritas as bases de dados utilizadas. Nas seções B e C, são discutidos os pré-processamentos realizados nas bases de alunos e de notas, respectivamente. Na seção D é mostrado como os dados faltantes foram tratados e na seção E como foi lidado com o desbalanceamento entre as classes. A seção F descreve os primeiros modelos de classificação testados, utilizando apenas dados de nota e frequência Já a seção G descreve o segundo conjunto de modelos utilizados, os quais fizeram uso dos dados socioeconômicos dos alunos. Por fim, a seção H descreve os métodos utilizados para avaliação e comparação entre os modelos.

A. Descrição das Bases de Dados

Para o presente trabalho foram utilizadas bases de dados fornecidas pela Universidade Federal do Paraná (UFPR) contendo dados anonimizados de alunos do curso de Tecnologia em Análise e Desenvolvimento de Sistemas. A descrição de cada uma das bases pode ser encontrada na Tabela I, sendo que todos podem ser conectadas por uma chave em comum denominada ID Aluno. As bases contam com dados de 583 alunos que cursaram 127 disciplinas diferentes, obtendo

18.857 resultados finais diferentes. Tais dados foram coletados entre os anos de 2012 e 2020. As Tabelas II, III e IV mostram com detalhes os atributos presentes em cada uma das bases fornecidas, bem como os tipos destes atributos.

Tabela I
BASES DE DADOS UTILIZADAS

Nº	Tabela	Nº de Dados
1	Alunos	583
2	Histórico	18.857
3	Integralização	583

B. Pré-processamento da Base de Notas

Da base de dados de histórico, as colunas 4 e 6 são redundantes, portanto optou-se em retirar a coluna 4 da Tabela III.

No primeiro modelo criado, foi utilizado apenas o histórico do aluno para prever uma possível reprovação. A coluna 13, denominada "Tipo" se refere a forma como a disciplina foi cursada. Na Fig. 12 é possível ver a distribuição dos dados presentes nesta coluna. Um aluno cursando uma matéria de forma regular (denominado na base como tipo "Turma") se matricula para cursar a disciplina na faculdade na qual está matriculado e no semestre esperado e pode ter quatro possíveis resultados. O resultado positivo esperado é que o aluno seja aprovado, o que acontece com 63,0% dos alunos que cursam uma disciplina pelo tipo "Turma". Como resultado negativo, existem 3 opções possíveis: o aluno pode cancelar a sua matricula na máteria, o aluno pode ser repovado por nota ou ele pode ser reprovado por frequência ao não comparecer às aulas da disciplina.

Situações de equivalência (o aluno cursou outra disciplina que possui um curriculo semelhante), adiantamento (o aluno realiza uma prova para comprovar conhecimento prévio na disciplina, seja através do trabalho ou outros cursos) e aproveitamento (o aluno realiza uma prova para comprovar conhecimento na disciplina, sendo que este já havia sido reprovado nela anteriormente) podem gerar resultados positivos, de aprovação, ou negativos, de reprovação. Tais possíveis resultados já se encontram discriminados na base utilizada. Por fim, um aluno que realiza um trancamento, decide por suspender a sua matricula por prazo pré-determinado dentro da sua universidade. Tal situação caracteriza-se como um resultado negativo e este deve ser igualmente predito.

Já na Fig. 13 é possível ver as médias de nota e frequência, a fim de se identificar possíveis inconsistências nos dados.

Pode-se notar que 96,1% dos dados registrados são de alunos que cursaram a matéria de forma regular. Entre os demais, é possível perceber que as matérias cursadas como adiantamento possuem valores de frequência inconsistentes, o mesmo se dá para aproveitamento, mobilidade, sem turma e trancamento. A partir disso, considera-se a eliminação dos dados dessas categorias.

Além disso, foi realizada uma simplificação das classes presentes na coluna "Status". Reprovado por nota, reprovado por frequência e cancelado foram agrupados na classe reprovado, juntamente com os dados de trancamento.

Tabela II
ATRIBUTOS DA BASE DE DADO DE ALUNOS

Nº	Atributo	Tipo
1	ID Aluno	Numérico
2	Altas habilidade/Superdotação (Sim/Não)	Categórico
3	Ano conclusão (Ensino médio)	Numérico
4	Ano Evasão	Numérico
5	Ano Ingresso Currículo	Numérico
6	Ano Ingresso Curso	Numérico
7	Ano Versão Currículo (2009 ou 2017)	Categórico
8	CH Currículo - Total	Numérico
9	CH Integralizada + Matriculada	Numérico
10	CH Integralizada - AF	Numérico
11	CH Integralizada - Obrigatória	Numérico
12	CH Integralizada - Optativa	Numérico
13	CH Integralizada - Total	Numérico
14	CH Integralizada - Total (%)	Numérico
15	CH Matriculada	Numérico
16	Cidade Nascimento	Categórico
17	Comunidade Indígena	Categórico
18	Comunidade quilombola e estado	Categórico
19	Cota	Categórico
20	Curso	Categórico
21	Data Conclusão	Data
22	Data de Colação	Data
23	Data Exp. Diploma	Data
24	Data Matrícula	Data
25	Distúrbio ou disfunção de aprendizagem	Categórico
26	Enade (Regular/Não Regular)	Categórico
27	Estado Nascimento	Categórico
28	Forma de evasão	Categórico
29	Forma de Ingresso	Categórico
30	Grau	Categórico
31	Habilitação	Categórico
32	Indígena (Sim/Não)	Categórico
33	IRA	Numérico
34	Matriculado (Sim/Não)	Numérico
35	Migrante, Refugiado, Apátrida ou Portador de visto humanitário (Sim/Não)	Categórico
36	NSC	Numérico
37	País Nascimento	Categórico
38	Período atual	Numérico
39	Período evasão	Numérico
40	Período Ingresso	Numérico
41	Quilombola (Sim/Não)	Categórico
42	Raça/Cor (Preta/Parda/Amarela/Branca/Não Informado)	Categórico
43	Reprovações por frequência	Numérico
44	Reprovações por nota	Numérico
45	Situação (Registro Ativo /Conclusão Formatura /Trancamento /Evasão)	Categórico
46	Solicitou Matrícula (Sim/Não)	Categórico
47	Surdo ou surda - usuário de Libras (Sim/Não)	Categórico
48	Tipo de Deficiência	Categórico
49	Transtorno do Espectro Autista	Categórico
50	Turno (Noturno/Vespertino)	Categórico
51	Versão do Currículo	Categórico

C. Pré-processamento da base de alunos

Dos dados recebidos, todos os alunos cursam o grau tecnológico de "Análise e Desenvolvimento de Sistemas". Além disso, não foi identificado nenhum aluno proveniente de comunidade indigena ou quilombola, como também, durante o período analisado, nenhum aluno apresentava nenhum tipo de deficência. Dado estes fatos, as colunas 17, 18, 20, 30, 31, 41, 47, 48 e 49 foram retiradas da base de dados de alunos.

Tabela III
ATRIBUTOS DA BASE DE DADOS DE HISTÓRICO

Nº	Atributo	Tipo
1	Id Aluno	Numérico
2	Ano	Numérico
3	CH	Numérico
4	Código	Categórico
5	Currículo (Sim/Não)	Categórico
6	Disciplina	Categórico
7	Frequência	Numérico
8	Natureza (Obrigatória/Optativa)	Categórico
9	Nota	Numérico
10	Observação	Categórico
11	Período	Categórico
12	Status	Categórico
13	Tipo	Categórico

Tabela IV
ATRIBUTOS DA BASE DE DADOS DE INTEGRALIZAÇÃO

Nº	Atributo	Tipo
1	Id Aluno	Numérico
2	Código SIE	Categórico
3	Currículo	Categórico
4	Integralizado (Sim/Não)	Categórico
5	IRA	Numérico
6	Jubilamento	Categórico
7	Matriculado	Categórico
8	NSC	Numérico
9	Período atual	Numérico
10	Status	Categórico
11	Tempo universidade	Numérico
12	Turno (Noturno/Vespertino)	Categórico

Figura 12. Contagem de Registros Tipo e Status da Matéria no Histórico

Tipo	Status	Registros
TURMA	Aprovado	9517
	Matriculado	3018
	Cancelado	2119
	Reprovado por nota	1811
	Reprovado por frequência	1656
EQUIVALENCIA	Aprovado	435
TRANCAMENTO	-	111
ADIANTAMENTO	Aprovado	70
APROVEITAMENTO	Reprovado	66
	Aprovado	43
SEM_TURMA	Aprovado	6
ADIANTAMENTO	Reprovado	4
MOBILIDADE	-	1

Além disso, entende-se que os dados das colunas 9 e 13 podem

Figura 13. Registros de Nota e Frequência por Tipo e Status da Matéria no Histórico

Tipo	Status	Frequência	Nota
ADIANTAMENTO	Aprovado	4.285714	80.385712
	Reprovado	0.000000	18.750000
APROVEITAMENTO	Aprovado	0.000000	62.302326
	Reprovado	0.000000	18.545454
EQUIVALENCIA	Aprovado	0.689655	0.822989
MOBILIDADE	-	NaN	NaN
SEM_TURMA	Aprovado	98.833336	18.333334
TRANCAMENTO	-	NaN	NaN
TURMA	Aprovado	93.286644	79.480614
	Cancelado	13.784804	0.371874
	Matriculado	1.346587	0.172962
	Reprovado por frequência	31.068237	5.959541
	Reprovado por nota	89.393707	20.765324

ser inferidos a partir das demais e, para evitar redundâncias, estas foram removidas da base também. Por fim, a Tabela IV contém uma breve descrição dos dados de integralização. Nesta, como nenhum dos alunos apresentou jubilamento no periodo analisado, a coluna 6 foi retirada.

Deve-se ressaltar que houve uma mudança na grade curricular em 2017. As colunas 7 e 51 da Tabela II e 3 da Tabela IV identificam os alunos que cursam a versão curricular de 2017 ou a anterior, de 2009. Alguns acrônimos específicos utilizados nestas bases foram: CH, que significa Carga Horária, IRA, que significa Índice de Rendimento Acadêmico e NSC que significa Número de Semestres Cursados.

Foi feito um pré-processamento da base de alunos, verificando os dados presentes em algumas das colunas para entender a sua relevância e distribuição.

As primeiras colunas a serem exploradas foram as de país, estado e cidade de nascimento. O objetivo nesse caso foi verificar a quantidade e representatividade das categorias existentes, a fim de averiguar uma possível simplicação das mesmas.

Na Fig. 14 é possível visualizar a quantidade de alunos por país de nascimento. Com isso, percebe-se que cerca de 89,7% dos alunos se declararam como brasileiros e que cerca de 9,6% dos alunos não possuem essa informação preenchida. Como uma quantidade inferior a 1% dos alunos é nascido fora do Brasil, esses dados foram agrupados em uma única categoria intitulada de ”Outros”.

Na Fig. 15 é possível visualizar a quantidade de alunos por estado de nascimento. 68,8% dos alunos se declararam nascidos no estado do Paraná, 10,3% estão com dados faltantes e os outros 20,9% estão distribuidos entre 18 estados. A fim de simplificar os dados e trazer análises relevantes, os estados foram agrupados nas regiões Norte, Nordeste, Sul, Sudeste

Figura 14. Registros de Alunos por País de Nascimento

País Nascimento	Registros
BRASIL	523
-	56
ANGOLA	1
COLOMBIA	1
HAITI	1
VENEZUELA	1

Figura 15. Registros de Alunos por Estado de Nascimento

Estado Nascimento	Registros
Paraná	401
-	60
São Paulo	58
Santa Catarina	22
Rio de Janeiro	7
Rio Grande do Sul	7
Rondônia	5
Mato Grosso do Sul	5
Bahia	2
Pernambuco	2
Paraíba	2
Minas Gerais	2
Mato Grosso	2
Distrito Federal	2
Ceará	1
Amazonas	1
Rio Grande do Norte	1
Maranhão	1
Espírito Santo	1
Tocantins	1

e Centro-Oeste, além do estado do Paraná, que foi mantido separado.

Na Fig. 16 é possível visualizar a quantidade de alunos distribuídos por cidade de nascimento. Neste caso, tem-se 53,0% dos alunos nascidos em Curitiba, 8,6% de dados faltantes e o 38,4% restante distribuido em 137 cidades diferentes. A fim de simplicação e facilitação da análise, os dados foram agrupados em capitais e cidades do interior dos estados. As exceções foram Curitiba e a região metropolitana de Curitiba (RMC) que foram separadas devido à sua grande quantidade de registros.

Além dos dados geográficos, foi explorada a coluna 29, relativa à forma de ingresso. Na Fig. 17 é possível ver as diferentes formas de ingresso dos alunos. Uma vez que 89,9% dos alunos ingressaram no curso através do vestibular ou do ENEM, optou-se por reunir as demais opções em uma única categoria intitulada ”Outros”.

D. Limpeza de Dados Faltantes

Como foi possível visualizar nas Fig. 14, Fig. 15 e Fig. 16, existe uma quantidade considerável de dados faltantes no base. Entre outras colunas, uma que chama a atenção pela quantidade dados ausentes é a 42, relativa a Raça/Cor dos estudantes. Na Fig. 18, pode ser visualizado que cerca de 51,1% dos alunos não tem a etnia declarada.

A função MICE da biblioteca Fancyimput[1] consegue lidar apenas com dados numéricos faltantes. Como os principais dados faltantes se encontravam nas variáveis categóricas, a falta de resposta foi mantida como uma categoria separada das demais.

E. Balanceamento de Classes

Na Fig. 19 tem-se a distribuição dos dados de histórico pela classe agrupada. É possível perceber uma maior incidência de alunos aprovados do que reprovados. Por esse motivo, foi necessária uma sobreamostragem dos dados de alunos reprovados utilizando a função SMOTE presente na biblioteca imblearn[2]. do Python.

Na Fig. 20 é possível visualizar a mudança na distribuição dos dados após a aplicação da função SMOTE. Houve um aumento de 78% na quantidade de entradas para reprovações.

F. Análise por Nota e Frequência

A primeira análise de classificação foi feita utilizando somente os dados de histórico do aluno, contidos na Tabela III. A cada uma das linhas de dados foram adicionadas as disciplinas anteriores cursadas pelo aluno e as respectivas notas e frequências de cada uma. As variáveis categóricas “Disciplina”, “Natureza”, “Periodo” e “Observação” foram transformadas em variáveis *dummy*. Os valores de nota e

[1]Biblioteca que conta com variedade de diferentes algoritmos de imputação. Mais detalhes e documentação disponíveis em https://pypi.org/project/fancyimpute/

[2]Biblioteca focada em lidar com problemas de desbalanceamento de classes. Mais detalhes e documentação disponíveis em https://imbalanced-learn.org/stable/index.html

Figura 16. Registros de Alunos por Cidade de Nascimento

Cidade Nascimento	Registros
Curitiba	309
-	50
São Paulo	20
São José dos Pinhais	11
Registro	10
...	...
Guajará-Mirim	1
Goioerê	1
Garça	1
Fraiburgo	1
haiti	1

139 rows × 1 columns

Figura 17. Registros de Alunos por Forma de Ingresso

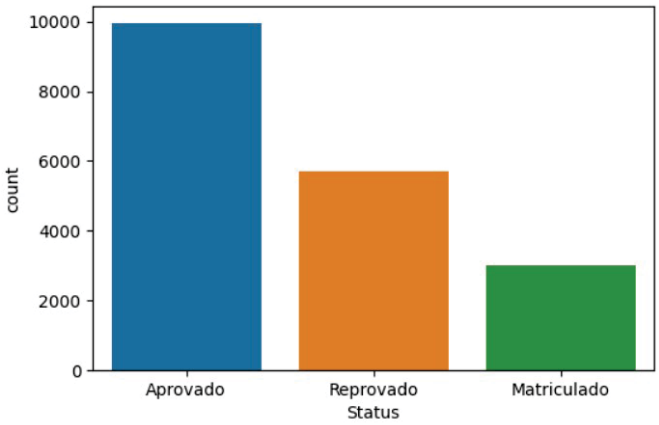
Forma de Ingresso	Registros
Vestibular	441
ENEM	83
Reopção	32
Mudança de Turno	10
Aproveitamento Curso Superior	7
Refugiado	3
Transferência Provar	3
Mudança de Currículo	1
Processo Seletivo	1
Reintegração	1
Transferência Ex-Ofício	1

de frequência da disciplina vigente foram retirados pois entende-se que tais dados não estarão disponíveis no momento da aplicação do algoritmo. Por fim, foi retirada a coluna "id_aluno" para que esta não interfira na relevância dos demais parâmetros. A partir disso, foi realizada uma classificação entre aprovados e reprovados pelas técnicas de SVM , *Deep Learning* e *Random Forest*.

Figura 18. Registros de Alunos por Raça/Cor Autodeclarada

Raça/Cor	Registros
Não Informado	286
Branca	215
Parda	49
Preta	13
0	12
Amarela	8

Figura 19. Balanceamento dos Status dos Alunos



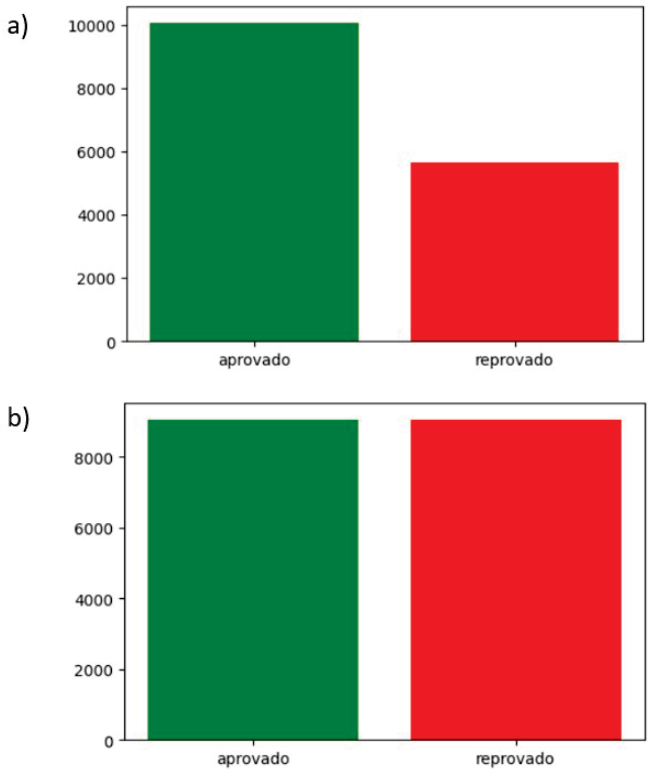
O SVM foi aplicado através do classificador previamente disponibilizado na bibliteca *sklearn*[3]. Como parâmetros para o odelo, foi o utilizado o *kernel Radial Basis Function* (RBF) e *random state* igual a 1.

Já o *Deep Learning* foi implementado utilizando os códigos presentes na biblioteca *tensorflow*[4]. Este contou com oito camadas, sendo a primeira camada de *input*, que recebe um dado de entrada com 431 dimensões. A segunda camada, de *Flatten* foi posta para transformar o dado de entrada em uma *arrray* bidimensional. A terceira camada densa contou com a função de ativação sigmóide. A quinta e sexta camadas densas contaram com a função de ativação ReLU A oitava camada tem a função de ativação *softmax*. A quarta e sétima camadas são de *dropout*, para evitar o *overfitting*. O modelo contou com o otimizador *Adam*, função de perda de *sparse categorical*

[3]Biblioteca que conta com dezenas de algoritmos e modelos de *Machine Learning*. Mais detalhes e documentação disponíveis em https://scikit-learn.org/stable/

[4]Biblioteca de código aberto para *Machine Learning*. Mais detalhes e documentação disponíveis em https://www.tensorflow.org/?hl=pt-br

Figura 20. Distribuição do status dos dados utilizados no modelo. a) Totalidade dos dados de teste e de treino antes da aplicação do balanceamento de classes. b) Distribuição dos dados de treino após a aplicação do balanceamento entre as classes,



crossentropy e foi treinada em 30 épocas. O sumário do modelo pode ser encontrado na Fig. 21

Figura 21. Modelo *Deep Learning*

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 431)]	0
flatten (Flatten)	(None, 431)	0
dense (Dense)	(None, 256)	110592
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dense_2 (Dense)	(None, 128)	16512
dropout_1 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 64)	8256

Total params: 168,256
Trainable params: 168,256
Non-trainable params: 0

Por fim, o modelo de *Random Forest* foi aplicado através do classificador presente na biblioteca *sklearn*. O número *n_estimators*, relativo ao número de árvore, foi 100.

G. Análise por Nota, Frequência e Dados Sociodemográficos

A segunda análise foi feita com a adição de dados sociodemográficos, conforme presentes na Tabela II. Nesta análise, além das colunas já supracitadas, foram retiradas também as colunas "Ano evasão", "Data Conclusão", "Data de Colação", "Data Exp. Diploma", "Data Matrícula", "Forma de evasão" e "Período evasão", uma vezes que estas não seriam relevantes para uma análise relativa a uma eventual reprovação ou não em determinada disciplina. Após a adição destes dados, foram executados os algoritmos de classificação por SVM, *Deep Learning* e *Random Forest* exatamente como na primeira análise.

O modelo de *Deep Learning* foi mantido semelhante ao da primeira análise, sendo que as dimensões dos dados de entrada e o número total de parâmetros foram atualizados de acordo. Este encontra-se na Fig. 22

Figura 22. Modelo *Deep Learning* para a Análise por Nota, Frequência e Dados Sociodemográficos

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 510)]	0
flatten_2 (Flatten)	(None, 510)	0
dense_6 (Dense)	(None, 256)	130816
dropout_4 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 128)	32896
dense_8 (Dense)	(None, 128)	16512
dropout_5 (Dropout)	(None, 128)	0
dense_9 (Dense)	(None, 64)	8256

Total params: 188,480
Trainable params: 188,480
Non-trainable params: 0

H. Avaliação dos Modelos

Para a avaliação da qualidade dos modelos, foi utilizada a matriz de confusão e dados de acurácia, precisão, *recall* e *f1-score*. Para a criação da matriz de confusão foi utilizada a biblioteca *seaborn*[5] em conjunto com *sklearn* e *matplotlib*[6].

I. Análises dos Falsos Positivos e Falsos Negativos

Após ser determinado qual o melhor modelo para a tarefa em questão, foi feita uma análise mais minuciosa a cerca quais foram os parâmetros mais comuns que leveram a algoritmo a se equivocar no resultado final. A partir disso, é possível

[5]Biblioteca focada em visulização de dados estatísticos. Mais detalhes e documentação disponíveis em https://seaborn.pydata.org/

[6]Biblioteca focado na criação de imagens estáticas, animadas ou interativas. Mais detalhes e documentação disponíveis em https://matplotlib.org/

supor quais são informações que mais contribuem para a classificação de um aluno como possível reprovação em uma matéria.

Dada a característica social do problema, tal análise se mostra de grande relevâcia para que se possa entender quais seriam as possíveis causas raízes que levariam um aluno a ter um risco maior de reprovação que outro. A partir da descoberta destas possíveis causas raízes, seria então possível encontrar formas de se trabalhar e mitigar o problema.

IV. Resultados e Discussões

Para que se tornasse possível uma predição de uma eventual reprovação por parte de um aluno, foram realizados alguns experiementos utilizando diferentes algoritmos. Com o auxílio da linguagem de programação *Python* e do ambiente de desenvolvimento *Google Colab*, foram testados os algoritmos de SVM, *Deep Learning* e *Random Forest* em dois contextos diferentes: considerando apenas os dados de histórico (como nota e frequência de disclinas cursadas em semestres anteriores) e utilizando estas mesmas informações em conjunto com dados socioeconômicos dos alunos. Os algoritmos utilizados foram nomeados como:

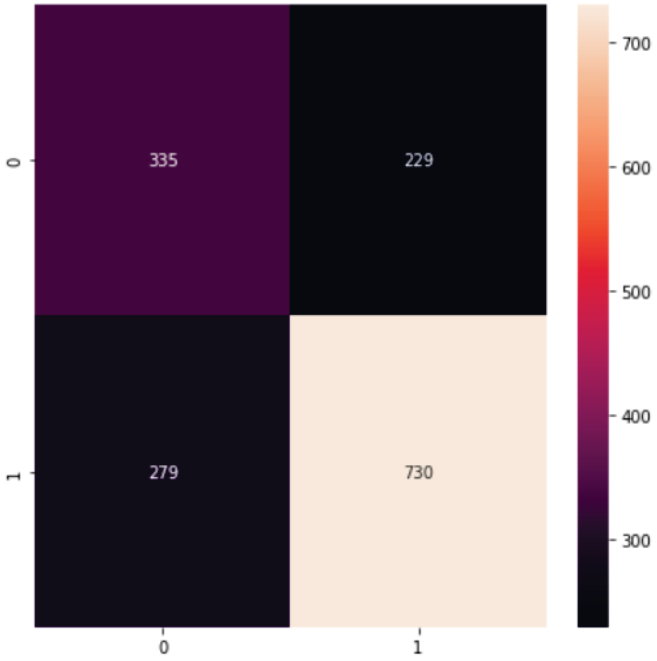
- SVM_hist: predição de aprovação baseado em dados de histórico com a utilização de SVM;
- DL_hist: predição de aprovação baseado em dados de histórico com a utilização de *Deep Learning*;
- RF_hist: predição de aprovação baseado em dados de histórico com a utilização de *Random Forest*;
- SVM_soc: predição de aprovação baseado em dados de histórico e dados socioeconômicos com a utilização de SVM;
- DL_soc: predição de aprovação baseado em dados de histórico e dados socioeconômicos com a utilização de *Deep Learning*;
- RF_soc: predição de aprovação baseado em dados de histórico e dados socioeconômicos com a utilização de *Random Forest*.

A. Análise por Nota e Frequência

A base original foi dividida em 90% para treino e 10% para testes. A base de dados de teste contava com 1.573 entradas enquanto a base de treino tinha 14.154 entradas. Após da aplicação de *oversampling*, a base de treino passou a ter 18.124 entradas. A partir das transformações comentadas anteriormente, a base passou a ter 431 colunas de dados de histórico de um aluno em disciplinas anteriores a qual foi realizada a predição.

A matriz de confusão para o modelo SVM_hist pode ser visualizada na Fig. 23. Em todas as matrizes de confusão apresentadas neste trabalho, o eixo das ordenadas representa os valores reais, sendo 0 reprovação e 1 aprovação e o eixo das abcissas representa os valores preditos. Neste caso, o modelo já conta com uma capacidade de predição razoável, uma vez que há uma incidência maior de valores classificados de forma correta do que de forma errônea. No modelo é possível perceber uma melhor precisão e *recall* para aprovações do que para reprovações.

Figura 23. Matriz de confusão para o preditor de nota com SVM



Na Tabela V é possível ver os valores de acurácia, precisão, *recall* e *f1-score* para um melhor entendimento dos detalhes do modelo.

Tabela V
Resultados do preditor de nota com SVM

	precisão	recall	f1-score	suporte
0	0.55	0.59	0.57	564
1	0.76	0.72	0.74	1009
acurácia			0.68	1573
média macro	0.65	0.66	0.66	1573
média ponderada	0.68	0.68	0.68	1573

A classificação com *Deep Learning*, chamada de modelo DL_hist, demonstrou uma tendência menor de acertar o resultado da predição e também apresentou uma incidência maior de falsos negativos. Por outro lado, o algoritmo teve uma tendência maior de acertar uma reprovação, dado o baixo percentual de falsos positivos. O resultado da matriz de confusão pode ser visualizado na Fig. 24 e a nomenclatura é a mesma da matriz de confusão mostrada anteriormente.

Na Tabela VI é possível ver os valores de precisão, recall e f1-score.

Por fim, na análise com utilização de *Random Forest*, entitulada RF_hist, foi possível notar uma melhora consistente na predição, tendo uma redução nos valores preditos erroneamente (falsos positivos e falsos negativos) e um aumento das predições corretas (verdadeiros positivos e verdadeiros

Figura 24. Matriz de confusão para o preditor de nota com *Deep Learning*

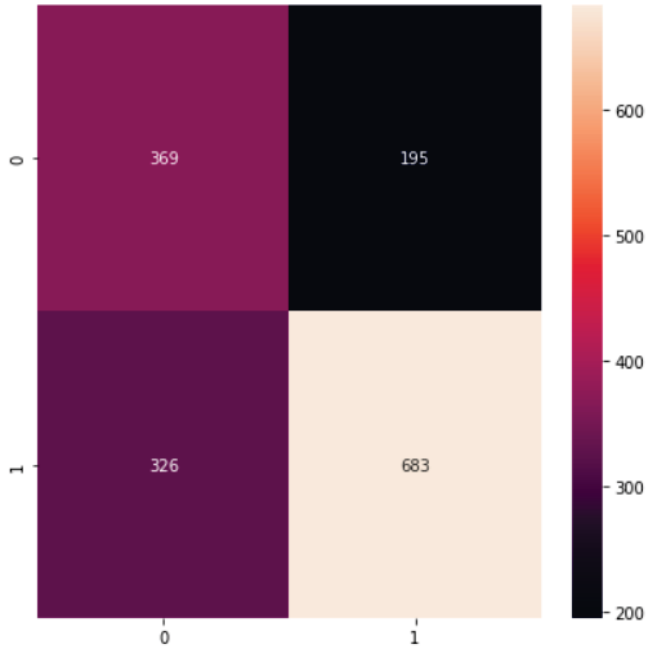


Figura 25. Matriz de confusão para o preditor de nota com *Random Forest*

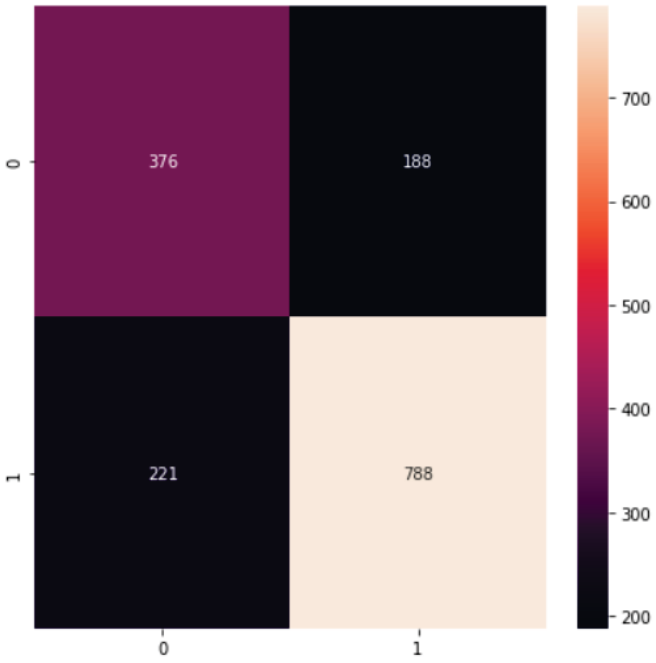


Tabela VI
RESULTADOS DO PREDITOR DE NOTA COM *Deep Learning*

	precisão	recall	f1-score	suporte
0	0.53	0.65	0.59	564
1	0.78	0.68	0.72	1009
acurácia			0.67	1573
média macro	0.65	0.67	0.66	1573
média ponderada	0.69	0.67	0.67	1573

negativos). Sua matriz de confusão pode ser visualizada na Fig. 25

Na Tabela VII é possível ver os valores de precisão, *recall* e *f1-score*. Como dito anteriormente, este algoritmo atinge melhores resultados que o SVM_hist e o DL_hist em todas as instâncias.

Tabela VII
RESULTADOS DO PREDITOR DE NOTA COM *Random Forest*

	precisão	recall	f1-score	suporte
0	0.63	0.67	0.65	564
1	0.81	0.78	0.79	1009
acurácia			0.74	1573
média macro	0.72	0.72	0.72	1573
média ponderada	0.74	0.74	0.74	1573

B. Análise por Nota, Frequência e Dados Sociodemográficos

A base foi dividida da mesma forma que a base anterior, sendo que a principal diferença é que neste caso ela contava com 89 colunas adicionais de dados sociodemográficos, totalizando 510 colunas. A classificação foi feita utilizando-se de todos os dados presentes nas bases de histórico e de alunos, com exceção daqueles que foram retirados conforme discutido na seção III.

Ao se executar o novo modelo de dados com o algoritmo de SVM, foi possível perceber que os resultados se mostraram muito semelhantes aos alcançados pelo SVM_hist, sendo que o modelo com mais colunas apresentou uma pequena redução em sua capacidade de predição. Isso indica que provavelmente os dados de histórico são os mais relevantes para a realização da predição para este modelo. O resultado da matriz de confusão para o o modelo SVM_soc pode ser encontrado na Fig. 26.

Na Tabela VIII é possível ver os valores de acurácia, precisão, *recall* e *f1-score*.

Tabela VIII
RESULTADOS DO PREDITOR DE NOTA COM DADOS SOCIODEMOGRÁFICOS USANDO SVM

	precisão	recall	f1-score	suporte
0	0.54	0.59	0.57	564
1	0.76	0.72	0.74	1009
acurácia			0.67	1573
média macro	0.65	0.66	0.65	1573
média ponderada	0.68	0.67	0.68	1573

Foi então realizada a mesma análise com a totalidade da base de dados de alunos, porém desta vez utilizando o algoritmo de *Deep Learning*, intitulado como modelo DL_soc. Neste caso, houve uma melhora significativa em todos os resultados do modelo, tendo como única exceção o aumento no número de falsos positivos. Sua matriz de confusão pode ser visualizada na Fig. 27.

Figura 26. Matriz de confusão para o preditor de nota com dados sociodemográficos usando SVM

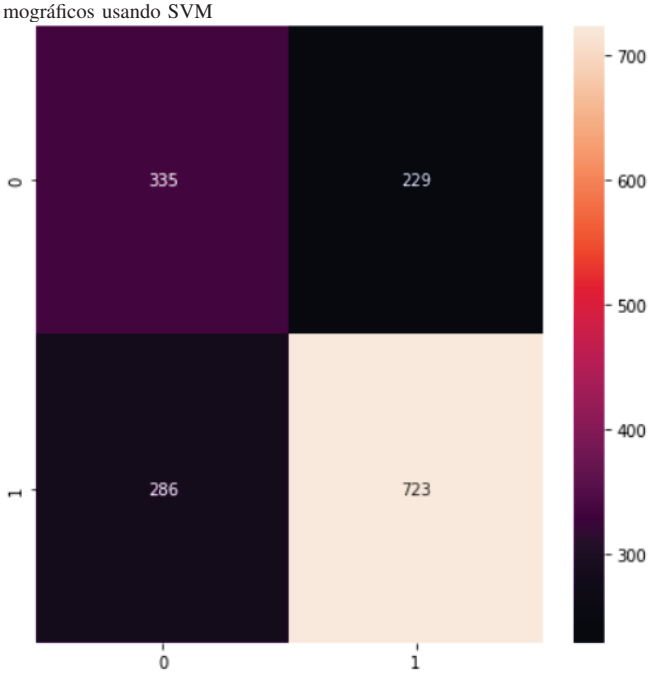
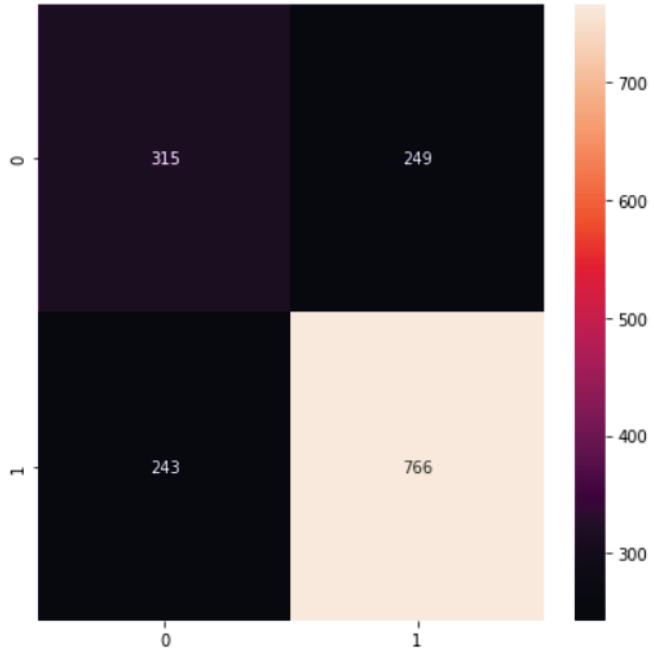


Figura 27. Matriz de confusão para o preditor de nota com dados sociodemográficos usando *Deep Learning*



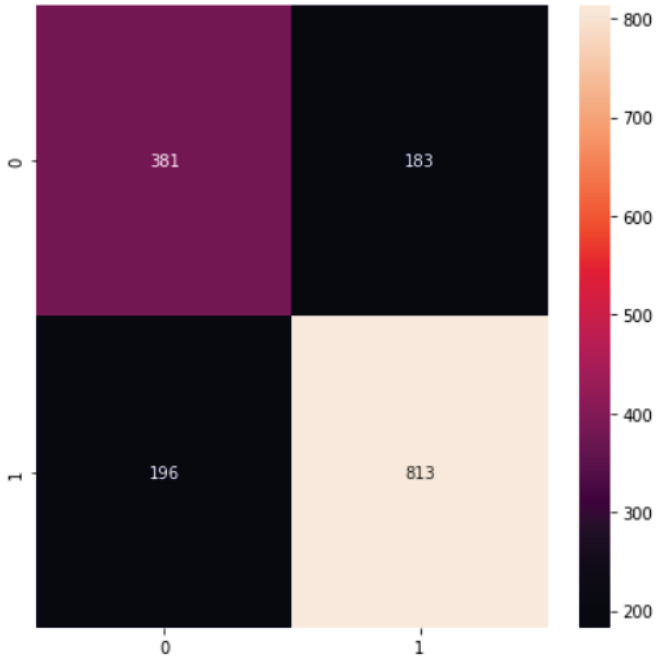
A Tabela IX contém os dados de acurária, precisão, *recall* e *f1-score* deste modelo. Este modelo apresentou valores de acurária e precisão da classe 0 melhores que o modelo SVM_soc, principalmente pela redução de falsos negativos, porém teve uma penalidade no *recall* da classe 0.

Tabela IX
RESULTADOS DO PREDITOR DE NOTA COM DADOS SOCIODEMOGRÁFICOS USANDO *Deep Learning*

	precisão	recall	f1-score	suporte
0	0.56	0.56	0.56	564
1	0.75	0.76	0.76	1009
acurácia			0.69	1573
média macro	0.66	0.66	0.66	1573
média ponderada	0.69	0.69	0.69	1573

Por fim, a mesma análise foi realizada com o algoritmo de *Random Forest*, entitutalado RF_soc. Assim como o SVM_soc, esse algoritmo não apresentou grandes alterações em comparação com o seu paralelo utilizado para dados de histórico, RF_hist. Tal fato corrobora na teoria de que dados socioeconômicos podem não ser os mais relevantes para se determinar o sucesso de um aluno em uma determina disciplina. A principal diferença deste modelo com o SVM_soc é que este apresentou uma pequena melhora na qualidade dos resultados com relação ao seu equivalentes de dados de histórico, RF_hist. Entre os seis modelos o que apresentou o menor número de falsos positivos e de falsos negativos. A matriz de confusão deste modelo pode ser visualizada na Fig. 28.

Figura 28. Matriz de confusão para o preditor de nota com dados sociodemográficos usando *Random Forest*



A Tabela X contém os valores de acurácia, precisão e *recall* do modelo. Entre os seis modelo este é o que apresentou a maior acurácia, a maior precisão e o *recall* para a classe 0 e a maior precisão e *recall* para a classe 1.

Tabela X
RESULTADOS DO PREDITOR DE NOTA COM DADOS SOCIODEMOGRÁFICOS USANDO *Random Forest*

	precisão	recall	f1-score	suporte
0	0.66	0.68	0.67	564
1	0.82	0.81	0.81	1009
acurácia			0.76	1573
média macro	0.74	0.74	0.74	1573
média ponderada	0.76	0.76	0.76	1573

C. Comparação Geral entre Modelos

Para que fosse possível uma comparação mais eficaz entre os seis modelos treinados, foram selecionados para comparação como dados mais importantes para avaliação a acurácia geral do modelo, bem como valores de precisão e recall para a classe 0. A classe 0 foi considerada como de maior interesse para esse estudo, pelo fato de se supor um efeito negativo maior quando se prevê falasamente que um aluno irá ser aprovado em uma matéria do que se prever falsamente que o mesmo será reprovado. Tais dados podem ser encontrados na Tabela XI.

Tabela XI
COMPARAÇÃO ENTRE OS MODELOS DE PREDIÇÃO

modelo	acurácia	precisão	recall
SVM_hist	0.68	0.55	0.59
DL_hist	0.67	0.53	0.65
RF_hist	0.74	0.63	0.67
SVM_soc	0.67	0.54	0.59
DL_soc	0.69	0.56	0.56
RF_soc	0.76	0.66	0.68

Conforme comentado anteriormente, a adição dos dados sociodemográficos não gerou um efeito uniforme nos três algoritmos. Enquanto a adição das novas colunas de dados reduziu a qualidade da classificação para o algoritmo utilizando SVM, estas melhoraram a qualidade dos resultados para os algoritmos de *Deep Learning* e *Random Forest*. Ao se comparar os três algoritmos de classicação, SVM, *Deep Learning* e *Random Forest*, foi possível perceber que o *Random Forest* teve o melhor desempenho entre eles, sendo que o modelo RF_soc apresentou resutados de acurácia, precisão e *recall* superiores aos do modelo RF_hist. Entre os seis modelos analisados, o RF_soc se mostrou consistentemente superior nos três parâmetros analisados.

D. Relevância dos Parâmetros no Resultado

Nos modelos que utilizaram o algoritmo de *Random Forest*, RF_hist e RF_soc, foi possível obter quais eram os parâmetros mais relevantes para definição se um determinado aluno estaria sob risco ou não de reprovar em uma matéria.

Na Tabela XII é possível visualizar quais foram os 10 parâmetros mais relevantes no modelo RF_hist. Primeiramente, repara-se que a carga horária cursada é o parâmetro mais relevante de todos. Porém não é possível inferir com estes dados se uma matéria com uma maior carga horária costuma resultar em uma maior taxa de reprovações ou se o contrário seria verdadeiro. Dois outros parâmetros que surpreendemente se mostraram relevantes para a possibilidade de aprovação de um aluno foram o ano e semestre (período) no qual a disciplina foi cursada. Tal fato fomenta a hipótese de que o nível de cobrança de um professor em uma determinada matéria possa não se manter estável com o tempo. Por fim, o resultado de 7 disciplinas se mostraram de grande relevância para o sucesso de um aluno ao longo do curso: Projeto de Algoritmos e Práticas de Programação, Introdução à Arquitetura de Computadores, Engenharia de Requisitos, Administração de Sistemas, Estatística para Computação, Matemática para Computação e Sistemas de Informação. Todas as matérias listadas são referentes ao primeiro semestre do curso e a ordem de importância delas está ordenada por carga horária, sendo que Projeto de Algoritmos e Práticas de Programação tem 90h, Introdução à Arquitetura de Computadores, Engenharia de Requisitos e Administração de Sistemas tem 60h e as demais 30h de carga horária.

Tabela XII
PARÂMETROS MAIS RELEVANTES NO MODELO RF_HIST

Parâmetro	Relevância
CH	0.055761
Período	0.051806
Ano	0.028985
nota_PROJETO DE ALGORITMOS E PRÁTICA DE PROGRAMAÇÃO	0.017200
nota_INTRODUÇÃO À ARQUITETURA DE COMPUTADORES	0.016942
nota_ENGENHARIA DE REQUISITOS	0.016284
nota_ADMINISTRAÇÃO DE SISTEMAS	0.013913
nota_ESTATÍSTICA PARA COMPUTAÇÃO	0.013705
nota_MATEMÁTICA PARA COMPUTAÇÃO	0.013412
nota_SISTEMAS DE INFORMAÇÃO	0.012796

Na Tabela XIII é possível visualizar os 15 parâmetros mais relevantes no modelo RF_soc. Para este modelo, foi recolhido um número maior de parâmetros já que o mesmo demonstrou uma capacidade superir de classificação. Neste é possível visualizar alguns parâmetros repetidos com relação ao modelo anterior como a carga horária que se mostrou como o parâmetro de maior relevância. O período e o ano na qual a disciplina foi cursada também se mostraram como parâmetros de grande relevância. Além disso, as notas obtidas nas disciplinas de Engenharia de Requisitos e Projeto de Algoritmos e Práticas de Programação se mostraram como de grande relevância para o modelo novamente.

Entre os novos parâmetros relevantes, tem o IRA total do aluno. Este nada mais é que uma média ponderada de todas as disciplinas já cursadas pelo aluno até o momento, podendo funcionar como um resumo de todos os parâmetros dos dados de históricos. Em seguida, tem-se o número total de reprovações por frequência e em seguida de reprovações por nota. Isso provavelmente se dá pelo fato de reprovações por frequência poderem ser interpretadas como falta de comprometimento com a matéria ou um sinal de que há alguma situação pessoal na vida do aluno que o impeça de se dedicar para a disciplina.

Tabela XIII
PARÂMETROS MAIS RELEVANTES NO MODELO RF_SOC

Parâmetro	Relevância
CH	0.057089
IRA	0.031583
Reprovações por frequência	0.024093
CH Matriculada	0.022677
Período	0.022181
Reprovações por nota	0.019711
Ano	0.019591
CH Integralizada - Obrigatória	0.014514
CH Integralizada - Total (%)	0.012937
NSC	0.011969
Natureza_Obrigatória	0.010216
nota_ENGENHARIA DE REQUISITOS	0.008972
nota_PROJETO DE ALGORITMOS E PRÁTICA DE PROGRAMAÇÃO	0.008885
Ano Ing Currículo	0.008652
Ano Ingresso Curso	0.008639

Outro parâmetro apontado como relevante foi a carga horária matriculada pelo aluno. Isso leva a duas possíveis hipóteses. A primeira é de que alunos que estavam matriculados em um número excessivo de matérias não teriam condições de se dedicar para todas elas, levando eles a inevitavelmente falhar em algumas. A outra hipótese é a de que um aluno matriculado em poucas disciplinas já estaria próximo a trancar o curso ou evadir, portanto este teria uma maior probabilidade de reprovação.

Também tem-se como novos parâmetros o quanto o aluno já concluiu do curso em carga horária. Isso leva a algumas hipóteses. A primeira é a de que alunos que estão em um estágio mais avançado do curso teriam um maior comprometimento e maturidade com relação às matérias. A segunda é de que os alunos com maior dificuldade no curso já tenham evadido anteriormente, sendo que os alunos que sobraram são os que teriam a maior probabilidade de aprovação. Já a terceira é de que as matérias no início do curso possuem um nível de exigência muito superior ao qual o aluno está acostumado (principalmente após saída do ensino médio), e que a partir do momento que esse conhecimento está sedimentado, as matérias no final do curso se tornam relativamente mais fáceis. O mesmo conceito pode ser utilizado para a alta relevância apontado para o parâmetro NSC.

Outro parâmetro que não se pode ter certeza no momento se tem efeito positivo ou negativo na classificação é o fato da matéria ser ou não obrigatória. Normalmente pode-se supor que o nível de exigência em uma disciplina obrigatória seja superior, porém, ao mesmo tempo, espera-se que haja uma dedicação maior do aluno neste caso. Por fim, o ano de ingresso no curso e o ano de ingresso no currículo levantam a mesma questão do período e ano que a disciplina foi cursada.

E. Resultado da Análise dos Falsos Positivos e Falsos Negativos

No modelo RF_soc, o qual apresentou os melhores resultados, foram analisadas quais seriam as características em comum das predições que foram feitas erroneamente, a fim de identificar melhor a influência de diferentes parâmetros na análise. Levando em consideração a relevância do IRA, do NSC e da Carga Horária Cursada, a primeira teoria a ser testada é a capacidade do algoritmo de identificar as possíveis reprovações ao longo do tempo no curso.

A primeira análise foi com relação ao número de semestres cursados pelo aluno e está presente na Tabela XIV. Ela mostra que a maior acurária na predição está para os alunos de primeiro semestre, os quais teriam NSC igual a zero. Porém, estes mesmos alunos são os que apresentam o menor *recall*, e o maior percentual de predições positivas, indicando uma tendência a falsos positivos para os alunos cursando o primeiro semestre.

Tabela XIV
VARIAÇÃO NA ACURÁRIA, PRECISÃO E RECALL DO MODELO RF_SOC EM FUNÇÃO DO NSC

NSC	Acurácia	Precisão (Classe 0)	Recall (Classe 0)	%Predições Positivas
0	85%	63%	53%	83%
1	81%	66%	74%	66%
2	78%	63%	78%	60%
3	74%	71%	68%	58%
4	66%	62%	58%	59%
5	69%	63%	68%	54%
6	62%	78%	62%	45%
7	69%	70%	73%	44%
8	59%	44%	58%	53%
9	70%	74%	88%	17%
10	67%	82%	75%	27%
11	67%	71%	83%	22%

O efeito contrário acontece com os alunos cursando o nono, décimo e décimo primeiro semestre na faculdade. O algoritmo estimou que menos de 30% dos alunos destas categorias seriam aprovados. A penalidade na acurácia indica uma tendência de falsos negativos nestas categorias.

A segunda análise feita foi com relação a carga horária matriculada e está presente na Tabela XV. O primeiro ponto a ser notada é a baixa incidência de predições positivas para alunos que estão matriculados em poucas disciplinas. Pelo modelo, foi predito que estes seriam aprovados em apenas 33% das máterias cursadas, havendo um alto valor de *recall* e precisão para embasar esse ponto. A penalidade no valor da acurácicia indicaria ainda que houve uma alta incidência de falsos positivos.

O algoritmo parece indicar que a carga horária ideal é entre 700h e 800h, já que é a que traria a maior probabilidade de aprovação. Tal predição é contraintuiva, e além disso, tal indicação deve ser levada com cautela tendo em vista o baixo valor do *recall* para esta faixa. Uma alternativa viável é indicar uma carga horária ideal de 300h a 400h visto que tem os parâmetros de acurácia, precisão e *recall* indicando que esta seria uma boa predição.

A terceira análise feita foi com relação a natureza da disciplina e está presente da Tabela XVI. A partir desta é possível notar que o algoritmo prediz uma maior taxa de aprovação para disciplinas não obrigatórias. Tal fato é corroborado pelos valores altos de acurácia, precisão e *recall* nas duas classes. Isso provavelmente se deve ao fato de disciplinas

Tabela XV
VARIAÇÃO NA ACURÁRIA, PRECISÃO E RECALL DO MODELO RF_SOC EM FUNÇÃO DA CARGA HORÁRIA MATRIULADA

CH Matriculada	Acurácia	Precisão (Classe 0)	Recall (Classe 0)	%Predições Positivas
0-100	67%	79%	73%	33%
100-200	72%	77%	77%	40%
200-300	77%	76%	78%	49%
300-400	80%	63%	63%	72%
400-500	66%	63%	59%	58%
500-600	65%	54%	74%	46%
600-700	64%	55%	69%	48%
700-800	77%	63%	56%	73%
800+	79%	67%	75%	63%

não obrigatórias serem em geral mais fáceis do que disciplinas obrigatórias.

Tabela XVI
VARIAÇÃO NA ACURÁRIA, PRECISÃO E RECALL DO MODELO RF_SOC EM FUNÇÃO DA NATUREZA OBRIGATÓRIA

Natureza	Acurácia	Precisão (Classe 0)	Recall (Classe 0)	%Predições Positivas
Obrigatória	72%	70%	69%	54%
Não Obrigatória	77%	65%	67%	65%

Por fim, foi realizada uma análise com relação ao ano de ingresso no curso e esta se encontra presente na tabela XVII. O ponto mais evidente desta análise é a baixa incidência de predições positivas para alunos que ingressaram no ano de 2012. Tal predição deve ser levada com cautela já que dos 583 alunos que compõem a base de dados analisadas, apenas um ingressou no ano de 2012. Também pode-se notar que o percentual de predições positivas aumenta quanto mais recente é o ano de ingresso do aluno no curso.

Tabela XVII
VARIAÇÃO NA ACURÁRIA, PRECISÃO E RECALL DO MODELO RF_SOC EM FUNÇÃO DA CARGA HORÁRIA MATRIULADA

Ano de Ingresso	Acurácia	Precisão (Classe 0)	Recall (Classe 0)	%Predições Positivas
2012	50%	64%	70%	21%
2013	58%	67%	53%	54%
2014	75%	68%	70%	59%
2015	73%	64%	58%	66%
2016	71%	59%	59%	65%
2017	80%	64%	75%	66%
2018	78%	67%	63%	70%
2019	77%	73%	71%	60%
2020	77%	70%	87%	44%

V. CONSIDERAÇÕES FINAIS

Este artigo teve como objetivo predizer uma reprovação antes que esta acontecesse utilizando ferramentas de *Data Mining* e KDD. Ao longo deste trabalho foram testados seis algoritmos de classificação diferentes nominados: SVM_hist, DL_hist, RF_hist, SVM_soc, DL_soc e RF_soc. Estes foram compostos por combinação entre três tipos de algoritmos de classificação diferentes (SVM, *Deep Learning* e *Random Forest*) e duas versões da base dados, sendo uma contendo apenas dados de histórico e outra contendo dados de histórico e dados sociodemográficos.

Entre todos os modelos testados, a utilização de *Random Forest* com dados de histórico e sociodemográficos (intitulado como modelo RF_soc) foi o que se mostrou com a melhor capacidade de classificação a partir dos valores de acurácia, precisão e *recall*, sendo estes respectivamente 0,76, 0,66 e 0,68. Ao se analisar os resutlados obtidos, pode-se perceber que os dados mais relevantes foram relativos à carga horária da disciplina cursada, o índice de rendimento acadêmico do aluno, o número de reprovações por frequência obtidas até o momento que este cursa a disciplina, a carga horária matriculada no semestre e o período do aluno.

Dados demográficos como Raça/Cor e Cota não estavam entre os parâmetros mais relevantes para a classificação, o que poderia indicar que estes não surtam um efeito determinante na performance universitária. Outros dados demográficos como gênero e renda familiar poderiam ser inclídos em um estudo posterior para estender o resultado desta análise e entender se estes possuem relevância com relação às reprovações.

O fato de a base de dados disponibilizada contar apenas com dados de 583 alunos, devido à migração do sistema de notas da UFPR em 2019, pode ter exercido influência a cerca do resultado obtidos. Recomenda-se uma nova análise com uma base de dados maior nos próximos 1-2 anos para averiguação deste efeito do resultado final das classificações.

Como trabalhos futuros propõe-se a utilização de outras técnicas de classificação, além da inclusão de outros parâmetros possivelmente relevantes para o desempenho acadêmico. Também sugere-se realizar a subamostragem das aprovações e comparar o resultado com as técnicas presentemente utilizadas.

REFERÊNCIAS

[1] MUSSLINER, B. O. MUSSLINER, M. S. S. MEZA, E. B. M. RODRIGUEZ, G. L. **O Problema da Evasão Universitária no Sistema Público de Ensino Superior: uma Proposta de Ação com base na Atuação de uma Equipe Multidisciplinar** Brazilian Journal of Development Vol 7, No 4 (2021)

[2] OLIVEIRA, E. **Com salas cheias e poucos professores jovens, Brasil tem desafios na reabertura das escolas, apontam dados da OCDE** G1, 2020.

[3] OLIVEIRA, E. **Nº de alunos que abandonam faculdade deve subir após a pandemia, e setores poderão enfrentar falta de mão de obra.** G1, 2020. Disponível em: https://g1.globo.com/educacao/noticia/2020/09/13/no-de-alunos-que-abandonam-faculdade-deve-subir-apos-a-pandemia-e-setores-poderao-enfrentar-falta-de-mao-de-obra.ghtml. Acesso em: 10 jul. 2021.

[4] PAURA, L. ARHIPOVA, I. **Cause Analysis of students' dropout rate in higher education study program.** Procedia Soc. Behav. Sci. 2014.

[5] **The college dropout crisis**. New York Times. Disponível em: https://www.nytimes.com/interactive/2019/05/23/opinion/sunday/college-graduation-rates-ranking.html. Acesso em: 10 jul. 2021.

[6] OLIVEIRA, J. J. G., NORONHA, R. V., KAESTNER, C. A. A. **Método de Seleção de Atributos Aplicados na Previsão da Evasão de Cursos de Graduação**. Revista de Informática Aplicada, 2017.

[7] SILVA, G. **Análise de evasão no ensino superior: uma proposta de diagnóstico de seus determinantes**. Avaliação (Campinas) 18 (2), 2013.

[8] CABELLO, A. CHAGAS, T. **Reprovações e evasão no ensino superior - uma análise com base na metodologia do INEP**.Revista Temas em Educação, 2021

[9] GOLDSCHIMDT R, PASSOS E,**Data Mining - um guia prático**. Elsevier, Rio de Janeiro, 2005.

[10] GARCIA, S, LUENGO, J,HERRERA, F. **Data Preprocessing in Data Mining.** Intelligent Systems Reference Library v. 72, 2015.

[11] HODGE, V. AUSTIN, J. **A Survey of Outlier Detection Methodologies.** Artificial intelligence review, v. 22, n. 2, p. 85-126, 2004.

[12] ZHU, J, GE, Z, SONG, Z, GAO, F. **Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data**. Annual Reviews in Control, 2018.

[13] HOUARI, R, BOUNCEUR, A, TARI, A, KECHADI, M, **Handling Missing Data Problems with Sampling**. International Conference on Advanced Networking Distributed Systems and Applications, 2014.

[14] PENG, L, LEI, L, **A Review of Missing Data Treatment Methods**. 2015.

[15] ZHANG, Z, **Multiple imputation with multivariate imputation by chained equation (MICE) package** Ann Transl Med. 2016 Jan; 4(2): 30.

[16] TAO, X, LI, Q, REN, C, GUO, W, LI, C, HE, Q, LIU, R, ZOU, J, **Real-value negative selection over-sampling for imbalanced data set learning** Expert Systems With Applications, 2019.

[17] MOHAMMED, R, RAWASHDEH, J, ABDULLAH, M, **Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results**. 11th International Conference on Information and Communication Systems (ICICS), 2020.

[18] ERTEKIN, S, **Adaptive Oversampling for Imbalanced Data Classification** Information Sciences and Systems 2013

[19] HARDY, M. **Regression with Dummy Variables**. Sage Publications. Newbury Park, CA. 1993

[20] MITCHELL, T. **Machine Learning**. McGraw-Hill Science/Engineering/Math, 1997.

[21] BONACCORSO, G. **Machine Learning Algorithms**. Packt Publishing, Birmingham, 2017.

[22] DAYAN, P. SAHANI, M. DEBACK, G. **Unsupervised Learning**. The MIT encyclopedia of the cognitive sciences, p. 857-859, 1999.

[23] ZHANGA, C, LIUA, C, ZHANG, X, ALMPANIDIS, G. **An up-to-date comparison of state-of-the-art classification algorithms.** Expert Systems with Applications, V. 82, 2017.

[24] DREISEITL, S, OHNO-MACHADO, L, **Logistic regression and artificial neural network classificationmodels: a methodology review**, Journal of Biomedical Informatics 35 (2002) 352–359.

[25] RASTROLLO-GUERRERO, J, L, GÓMEZ-PULIDO, J, A, DURÁN-DOMINGUÉZ.**Analyzing and Predicting Students' Performance by Means of Machine Learning: A Review** Applied sciences, 2020

[26] GALLANT, S. **Neural Network Learning and Expert System**. Cambridge: The MIT Press, 1995.

[27] AGGARWAL, C. **Neural Networks and Deep Learning**. Springer International Publishing, 2018.

[28] KETKAR, N. MOOLAYIL, J. **Deep Learning with Python** Apress, 2021.

[29] SHEA, K. NASH, R. **An Introduction to Convolutional Neural Networks** arXiv preprint arXiv:1511.08458, 2015.

[30] SHARMA, S. ATHAIYA, A. **Activation Functions in Neural Networks**. International Journal of Engineering Applied Sciences and Technology, 2020.

[31] DIETTERICH, T. **Overfitting and Undercomputing in Machine Learning**. ACM computing surveys (CSUR), 1995.

[32] BEBIS, G. GEORGIOPOULOS, M. **Feed-forward neural networks.** IEEE Potentials 1994.

[33] DING, R. LIU, Z. SHI, R. MARCULESCU, D. BLANTON R. **LightNN: Filling the Gap between Conventional Deep Neural Networks and Binarized Networks**.Great Lakes Symposium on VLSI, 2017.

[34] SOKOLIC, J. GIRYES, R. SAPIRO, G. RODRIGUES, M. **Robust Large Margin Deep Neural Networks**. IEEE Transactions on Signal Processing. Vol. 65, No. 16, 2017.

[35] NGUYEN, A. YOSINSKI, J. CLUNE, J. **Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images**. IEEE conference on computer vision and pattern recognition, 2015.

[36] SUBASI, A. **Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders** Computers in Biology and Medicine, Volume 43, Issue 5 2013.

[37] CHEN, Y. ZHOU, X. HUANG, T. **One-class SVM for learning in image retrieval** International Conference on Image Processing, 2001.

[38] CAO, Y. XU, J. LIU, T. LI, H. HUANG, Y. HON, H. **Adapting ranking SVM to document retrieval Share on** 29th annual international ACM SIGIR conference on Research and development in information retrieval, 2006

[39] KIM, D. NGUYEN, H. PARK, J. **Genetic algorithm to improve SVM based network intrusion detection system** 19th International Conference on Advanced Information Networking and Applications, 2005

[40] HEARST, M. DUMAIS, S. OSUNA, E. PLATT, J. textbfSupport vector machines. IEEE Intelligent Systems and their applications, v. 13, n. 4, p. 18-28, 1998.

[41] MEYER, D, **Support Vector Machines** The Interface to libsvm in package e1071, 2015

[42] LORENA, A. CARVALHO, A. **Uma Introdução às Support Vector Machines** RITA, Volume XIV,Número 2, 2007

[43] BREIMAN, L. **Random Forests** Machine learning, v. 45, 2001.

[44] CUTLER, A. CUTLER, D. STEVENS, J. **Random Forests**. Ensemble machine learning. Springer, 2012.

[45] LOH, W. **Classification and regression trees.** Wiley interdisciplinary reviews: data mining and knowledge discovery, 2011.

[46] AMARATUNGA, D. CABRERA, J., and LEE, Y. **Enriched random forests.** Bioinformatics 24.18 (2008): 2010-2014.

[47] THARWAT, A. **Classification Assessment Methods** Applied Computing and Informatics Vol. 17 Emerald Publishing Limited, 2021

[48] SOKOLOVA, M. JAPKOWICZ, N. SZPAKOWICZ, S. **Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation** AI 2006: Advances in Artificial Intelligence. AI 2006. Lecture Notes in Computer Science, vol 4304 2006.

[49] ROMERO, C. VENTURA, S. **Educational data mining and learning analytics: An updated survey** Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 10, n. 3, p. e1355, 2020.

[50] **educationaldatamining.org** Disponível em https://educationaldatamining.org/. Acesso em 16 out. 2021.

[51] BUCOS, M, DRĂGULESCU, B, **Predicting Student Success Using Data Generated in Traditional Educational Environments** TEM Journal. Volume 7, Issue 3, Pages 617-625.

[52] JOHNSON, L. SMITH, R. WILLIS, H. LEVINE, A. HAYWOOD, K. **The Horizon Report** The New Media Consortium, 2011.

[53] COELHO, O. B. SILVEIRA, I. F. **Deep Learning applied to Learning Analytics and Educational Data Mining: A Systematic Literature Review** VI Congresso Brasileiro de Informática na Educação, 2017.

[54] BAKER, R. S. J. D. INVENTADO J. S. **Educational Data Mining and Learning Analytics.** Learning Analytics: from Research to Practice, 2014.

[55] ALYAHIAN, E. DUSTEGOR, D. **Predicting academic success in higher education: literature review and best practices** International Journal of Educational Technology in Higher Education volume 17, Article number: 3, 2020.