

APLICAÇÃO DE REDES NEURAIS PARA EVASÃO ESCOLAR

Diego Araújo Silva
Escola Politécnica de Pernambuco (POLI)
Universidade de Pernambuco (UPE)
Recife, Brasil
diego.araujos@upe.br

Roberta Andrade de Araújo Fagundes
Escola Politécnica de Pernambuco (POLI)
Universidade de Pernambuco (UPE)
Recife, Brasil
roberta.fagundes@upe.br

Resumo—A evasão no ensino superior representa um desafio significativo para instituições de ensino, impactando no desenvolvimento social e econômico. Pois, essa evasão leva ao desperdício de recursos, como também, salas de aula vazias, ociosidade de professores e funcionários, e subutilização da infraestrutura disponível são consequências diretas. A previsão ajuda a otimizar o planejamento de turmas, alocação de docentes e uso das instalações. Portanto, esse estudo propõe a aplicação de modelos de redes neurais artificiais, especificamente Multilayer Perceptron (MLP) e Extreme Learning Machine (ELM), para prever a evasão de alunos no ensino superior. Foi utilizada uma base de dados pública do Instituto Politécnico de Portalegre, composta por informações demográficas, socioeconômicas e acadêmicas de 4.424 estudantes. A metodologia adotada seguiu o processo CRISP-DM, envolvendo etapas de entendimento dos dados, pré-processamento, modelagem e avaliação. Os resultados indicam que ambos os modelos são eficazes na tarefa de predição, com acurácia de 91,73% para a MLP e 91,87% para a ELM. Destaca-se que o ELM apresenta tempo de treinamento significativamente menor, sem a necessidade de alto poder computacional, tornando-se uma alternativa viável e eficiente para instituições com recursos limitados. Este estudo contribui para a adoção de técnicas de inteligência artificial como ferramenta de apoio na prevenção e gestão da evasão acadêmica.

Palavras-chave—Evasão escolar, Redes neurais, Multilayer Perceptron, Extreme Learning Machine, Ensino superior

I. INTRODUÇÃO

O ensino superior é uma fase crucial na vida dos estudantes, representando não apenas um período de formação acadêmica, mas também um momento de transição para a vida adulta e profissional. No entanto, a evasão e o insucesso acadêmico ainda são desafios significativos enfrentados pelas instituições de ensino superior em todo o mundo. A evasão e o fracasso acadêmico não apenas afetam os alunos individualmente, mas também têm implicações socioeconômicas mais amplas, influenciando o crescimento econômico, os empregos disponíveis, a competitividade e a produtividade do país [1].

De acordo com [1], a taxa de evasão no ensino superior em 2022 chegou na média de 27,7% para cursos presenciais e para 40% para cursos EAD. Considerando ainda um ciclo de 2018 a 2022, há no total uma taxa de evasão de 57,2%, com apenas 25,6% dos estudantes concluindo sua graduação e 21,8% ainda cursando. A desistência afeta todos os tipos de instituições, sendo que instituições de pequeno porte registraram uma média

de evasão total de 50,5%, e as gigantes um valor ainda maior, de 65,2%. Durante a pandemia, de acordo com [2], 84% das IES (Instituições de Ensino Superior) tomaram ações para prevenir a evasão, seja por abandono ou trancamento, de cursos por parte do aluno.

No Brasil, uma das estratégias adotadas pelo governo para enfrentar o problema da evasão no ensino superior é a Política Nacional de Assistência Estudantil (Pnaes), criada em 2007 [3] e atualizada em 2024 [4]. A política tem como foco estudantes de cursos presenciais de graduação, e, em alguns casos, também de mestrado e doutorado, oferecendo assistência por meio de programas voltados à permanência de estudantes de baixa renda. Entre esses programas estão o Programa de Apoio ao Transporte do Estudante (Pate), que oferece transporte gratuito a estudantes residentes em regiões sem acesso ao transporte público, e o Programa Estudantil de Moradia (PEM), voltado a garantir moradia a estudantes em situação de vulnerabilidade. Em 2024, segundo o governo, 400 mil estudantes foram atendidos por esses auxílios.

Além das políticas federais, instituições de ensino superior implementam ações próprias que contribuem para a permanência dos estudantes. A Universidade de Pernambuco (UPE) oferece bolsas de Auxílio Deslocamento [5] e Moradia [6], entre outras. A Universidade Federal do Rio Grande do Sul (UFRGS) oferece auxílio financeiro, moradia estudantil, alimentação e saúde [7] e a Universidade Federal do Amazonas (UFAM) implementa ações voltadas à permanência do discente, como bolsas acadêmicas e auxílios diversos, descritos em seu plano de assistência estudantil [8]. Todos esses programas são voltados a estudantes em situação de vulnerabilidade socioeconômica, visando evitar sua evasão.

Mesmo com a existência de programas voltados para a redução da evasão, suas causas são diversas, o que exige o desenvolvimento de novas abordagens para enfrentar esses desafios educacionais. Assim, a utilização de redes neurais artificiais na previsão da evasão escolar não é apenas uma melhoria tecnológica; é uma ferramenta estratégica que capacita as instituições a serem mais inteligentes, proativas e eficazes na promoção do sucesso de seus alunos e na construção de uma sociedade mais educada e desenvolvida. Além disso, temos mais alunos concluindo o ensino superior, significando mais

profissionais qualificados ingressando no mercado de trabalho, impulsionando a economia e a inovação.

Este trabalho propõe a aplicação de modelos de redes neurais artificiais, especificamente Multilayer Perceptron (MLP) e Extreme Learning Machine (ELM), para identificar fatores que influenciam o desempenho acadêmico dos alunos e sua probabilidade de evadir através da análise de um conjunto de dados do Instituto Politécnico de Portalegre (IPPortalegre) [9] que combina informações demográficas, socioeconômicas e acadêmicas de alunos de nível superior. Por meio de uma análise criteriosa desses dados, e o estudo de outros modelos desenvolvidos para resolver problemas similares, como os apresentados em [10] [11] [12] [13] [14], o objetivo deste trabalho é identificar padrões, tendências e correlações relevantes para as instituições de ensino superior. O foco central é utilizar os modelos preditivos para prever o desempenho acadêmico dos alunos e o risco de evadir. Esses modelos levam em consideração muitos fatores, incluindo características individuais dos alunos, dados socioeconômicos e indicadores acadêmicos.

II. TRABALHOS RELACIONADOS

Diversas pesquisas recentes utilizaram a base de dados portuguesa [9] para explorar a previsão de evasão escolar. Em sua maioria, esses estudos, para tornar o problema em uma classificação binária, retiraram os alunos classificados como matriculados (enrolled) da base de dados.

Em 2023, Sean Kim, Eliot Yoo e Samuel Kim [13] utilizaram ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) como métrica de avaliação do desempenho dos modelos Random Forest, Árvores de Decisão, SVM (Support Vector Machine) e KNN (k-Nearest Neighbors) na tarefa de prever a evasão. A Random Forest obteve o melhor resultado, com uma pontuação de 0,955. Além disso, os autores analisaram o impacto da remoção de diferentes tipos de dados: a ausência de informações acadêmicas provocou a maior queda de desempenho (ROC-AUC de 0,823), enquanto a exclusão de dados socioeconômicos, demográficos e macroeconômicos reduziu a desempenho para 0,945, 0,953 e 0,954, respectivamente.

Em 2024, quatro estudos também utilizaram a base do Instituto Politécnico de Portalegre, empregando técnicas de aprendizado de máquina para antecipar a evasão acadêmica.

Pedro Luiz Regattieri Da Cunha [10] realizou um pré-processamento que unificou os alunos matriculados com os graduados, considerando ambos como casos de não evasão. Em seguida, aplicou métodos de seleção de características, utilizando Árvores de Decisão para SVM, Random Forest, Regressão Logística, Gradient Boosting e Árvores de Decisão, e Árvores Extras para KNN e AdaBoost. Após o treino dos modelos, SVM alcançou a maior acurácia, com 85,7%. Além disso, foi testada uma rede MLP sem seleção prévia de atributos, que obteve 88,1% de acurácia com, segundo o autor, os seguintes hiperparâmetros: 4 camadas, sendo 1 de entrada com 34 neurônios e função de ativação ReLU, 2 camadas ocultas com 20 neurônios e função de ativação ReLU, e 1 camada de saída com 1 neurônio e função de ativação sigmoid.

Este pré-processamento e os hiperparâmetros foram replicados e testados, e foram obtidos resultados similares, com acurácia de 87,1%.

Outro estudo relevante, conduzido por Achmad Ridwan, Arif Mudi Priyatno e Lidya Ningsih [12], utilizou o algoritmo XGBoost e avaliou diferentes proporções de divisão entre treino e teste: 50:50, 60:40, 70:30, 80:20 e 90:10. A configuração 80:20 apresentou os melhores resultados, com acurácia de 87%, precisão de 88%, recall de 74% e F1-Score de 81%. Embora a acurácia tenha variado cerca de 1% entre as divisões, outras métricas apresentaram diferenças mais significativas. Por exemplo, a divisão 90:10 resultou em uma precisão de 80%, o que representa uma queda de 8% em relação à divisão 80:20.

O trabalho de Hai Thanh Nguyen, Phuong Le, Tuyen Thanh Thi Nguyen e Anh Kim Su [14] empregou os modelos Random Forest, XGBoost, LightGBM e CatBoost para a previsão das 3 classificações dos alunos disponíveis na base de dados: graduado, matriculado e evadido. Apesar da abordagem mais ampla, obtiveram resultados significativamente menores do que os outros trabalhos, com o XGBoost apresentando o melhor desempenho, atingindo acurácia de 78%, precisão de 82%, recall de 76% e F1-Score de 79%.

Por fim, Anas Alhardi e Selahattin Alan [11] analisaram seis algoritmos de aprendizado de máquina: Árvores de Decisão, SVM, Regressão Logística, Random Forest, Naïve-Bayes e XGBoost. Utilizando as variáveis Estado Civil, Gênero, Idade na Matrícula, Bolsista, Escolaridade Anterior, Escolaridade da Mãe, Curso, Disciplinas do 1º Semestre (aprovadas), Disciplinas do 2º Semestre (aprovadas) para a previsão, obtendo acurácias entre 83% (Naïve-Bayes) e 90% (XGBoost), sendo este último o mais eficaz no conjunto avaliado.

Embora não tenham utilizado a base de dados portuguesa, outros trabalhos se destacam por abordarem problemas de previsão e classificação utilizando redes neurais. Três desses estudos foram desenvolvidos em 2021 e apresentam metodologias e resultados relevantes para a área.

O trabalho de Erik Gabriel Cruz Sena e Honovan Paz Rocha [15] utilizaram redes neurais do tipo MLP e Convolutacional (CNN), além dos algoritmos SVM (Support Vector Machine), Random Forest e Gradient Boosting, para analisar um banco de dados contendo imagens de raio-X da região torácica, visando classificar se os pacientes estavam infectados com COVID-19. Para a rede MLP, foram empregadas as funções de ativação ReLU e Softmax nas camadas oculta e de saída, respectivamente, resultando em uma acurácia de 97,78%. A rede CNN obteve uma acurácia de 89,60%, enquanto os modelos SVM, Random Forest e Gradient Boosting alcançaram 79,93%, 86,59% e 75,76%, respectivamente.

No estudo de Barchi et al. [16], foi utilizada as métricas Erro Quadrático Médio (Mean Squared Error – MSE) e Erro Percentual Absoluto Médio (Mean Absolute Percentage Error – MAPE) para a tarefa de previsão do preço do açúcar no Brasil. Foram empregados os modelos de redes neurais MLP, ELM e Redes Neurais com Estados de Eco (ESN), além do modelo estatístico autorregressivo (AR). O estudo foi

conduzido utilizando duas bases de dados: uma do Centro de Estudos Avançados em Economia Aplicada (CEPEA) e outra da Bolsa de Nova York (NY), considerando três horizontes de previsão. Os resultados indicaram que a ELM obteve o melhor desempenho na base do CEPEA, superando os demais modelos nos três horizontes de previsão. No primeiro horizonte, os valores de MSE e MAPE foram, respectivamente, 0,1775 e 2,3989. Já na base da Bolsa de Nova York, a MLP apresentou os melhores resultados, com MSE de 1,4023 e MAPE de 6,4845. Apesar da superioridade da ELM na base nacional e da MLP na base internacional, a diferença de desempenho entre ambas foi mínima em todos os cenários avaliados. Os modelos AR e ESN ficaram consistentemente nas terceira e quarta posições em termos de desempenho, tanto nas duas bases quanto para ambas as métricas analisadas.

O artigo de Jonathan Bandeira, Mêuser Valença e Renan Alencar [17] testaram os algoritmos K-Nearest Neighbors (K-NN), Naïve Bayes, Support Vector Machine (SVM), Random Forest e rede neural MLP para o auxílio no diagnóstico precoce da doença de Alzheimer, com base em dados biomoleculares contendo valores de 120 proteínas sanguíneas. O modelo MLP apresentou o melhor desempenho, com acurácia de 94%, sensibilidade (recall) de 100% e especificidade de 87%. O K-NN também obteve bom desempenho, com acurácia de 90%, recall de 100% e especificidade de 79%. Na sequência, os modelos SVM (acurácia de 88%, recall de 100% e especificidade de 74%), Naïve Bayes e Random Forest (ambos com acurácia de 81%, recall de 98% e especificidade de 64%) apresentaram desempenhos inferiores. Além disso, o estudo avaliou o uso de dados sintéticos gerados por uma Rede Generativa Adversária (GAN), demonstrando potencial para otimizar o desempenho do modelo MLP por meio da ampliação do conjunto de dados.

Em 2023, Diego Ramiro Melo Monteiro e Adriana Rosa Garcez Castro [18] propuseram uma abordagem utilizando Redes Neurais Convolucionais Bidimensionais (CNN 2D) combinadas com Gráficos de Recorrência (Recurrence Plot – RP) para a previsão a curto prazo da geração de energia elétrica de uma micro usina fotovoltaica conectada à rede elétrica, localizada no Centro de Excelência em Eficiência Energética da Amazônia (CEAMAZON), da Universidade Federal do Pará (UFPA). O método consiste na conversão das séries temporais em imagens por meio de gráficos de recorrência, que foram posteriormente utilizados como entrada para a CNN 2D. Os resultados foram comparados com modelos tradicionais, incluindo uma rede MLP e uma CNN unidimensional (CNN 1D). A CNN 2D apresentou o melhor desempenho, com um RMSE de 0,1845 e MAE de 0,1542, superando tanto a MLP (RMSE de 0,2389 e MAE de 0,1973) quanto a CNN 1D (RMSE de 0,2649 e MAE de 0,2254).

III. FUNDAMENTAÇÃO TEÓRICA

Esta seção descreve os algoritmos de machine learning utilizados nesse estudo para previsão da evasão escolar.

A. Rede Neural Artificial

As redes neurais artificiais tiveram como origem o trabalho de Warren S. McCulloch e Walter Pitts [19], publicado em 1943. Neste trabalho, os autores propuseram o que ficou conhecido como neurônio de McCulloch-Pitts, um modelo binário no qual o neurônio é ativado apenas quando a soma ponderada das entradas excede um limiar, representando um comportamento "tudo-ou-nada", imitando um estado verdadeiro ou falso. Esse modelo permite a construção de redes capazes de realizar operações lógicas básicas, como AND, OR e NOT, por meio da interconexão entre unidades.

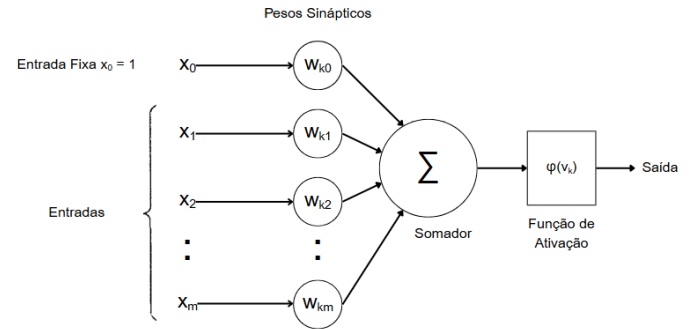


Figura 1. Neurônio de McCulloch-Pitts.

Um neurônio artificial típico é composto por [20]:

- Entradas ($x_0, x_1, x_2, \dots, x_m$): representam os sinais de entrada recebidos pelo neurônio, que podem corresponder a dados de entrada do sistema ou às saídas de outros neurônios da rede. A entrada x_1 é fixada em +1 e serve como o viés (bias) ao modelo. O viés permite o ajuste do limiar de ativação, deslocando a função de ativação horizontalmente no eixo cartesiano, permitindo que o neurônio seja ativado mesmo quando a combinação linear das entradas for igual a zero, o que ampliando a capacidade de representação e aprendizado da rede neural.
- Pesos Sinápticos ($w_{k0}, w_{k1}, w_{k2}, \dots, w_{km}$): representam os coeficientes atribuídos a cada entrada, responsáveis por ajustar a contribuição relativa de cada sinal na saída do neurônio. O peso w_{k0} está associado ao viés do neurônio.
- Unidade de soma: também denominada somador ou summing junction, essa unidade realiza o somatório ponderado dos sinais de entrada com seus respectivos pesos, resultando no potencial de ativação, conforme a equação:

$$v_k = \sum_{i=0}^m w_{ki}x_i \quad (1)$$

- Função de ativação ($\varphi(v_k)$): responsável por aplicar uma transformação não linear ao valor obtido pela unidade de soma. Essa função permite que a rede neural aprenda padrões complexos e não lineares. Algumas das funções mais usadas são as funções sigmóide (sigmoid), a função tangente hiperbólica (tahn) e a função ReLU (Rectified Linear Unit).

- Saída: o valor final produzido pelo neurônio após a aplicação da função de ativação. Esse valor pode ser utilizado como saída da rede neural ou como entrada para outros neurônios em camadas subsequentes.

Segundo os autores [19], redes desse tipo, quando combinadas com mecanismos auxiliares como fita de entrada, leitores e atuadores, poderiam realizar os mesmos cálculos que uma máquina de Turing. Isso indica que, em determinadas condições, as redes neurais artificiais possuem capacidade computacional equivalente à de sistemas formais de computação. Com base nesse estudo, foram desenvolvidos modelos mais complexos e flexíveis de redes neurais, que deram origem às arquiteturas modernas utilizadas em tarefas de aprendizado de máquina.

B. Multilayer Perceptron (MLP)

O Perceptron de Múltiplas Camadas (Multilayer Perceptron – MLP) é uma das arquiteturas mais utilizadas de redes neurais artificiais supervisionadas. Ele surgiu como uma melhoria do Perceptron de camada única proposto por Frank Rosenblatt [21] em 1958, que era capaz de resolver apenas problemas linearmente separáveis. O MLP supera essa limitação ao introduzir uma ou mais camadas ocultas entre as camadas de entrada e saída, permitindo a modelagem de funções não lineares complexas [20].

Uma MLP é composta por três tipos de camadas: uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída. Cada camada é formada por neurônios artificiais, conectados entre si por pesos ajustáveis.

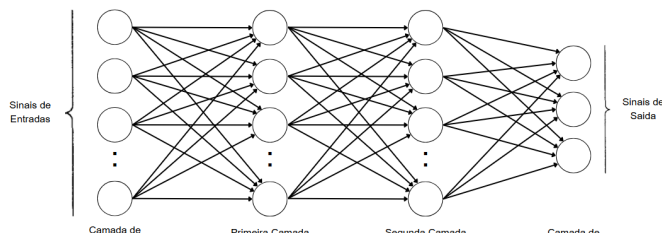


Figura 2. Perceptron de Múltiplas Camadas.

De acordo com [20], um método popular de treino da rede MLP é o algoritmo de retro-propagação (back-propagation), que consiste em duas fases: primeiro, os dados percorrem a rede com os pesos fixos até a saída (fase direta), e depois, o erro entre a saída obtida e a desejada é propagado de volta pela rede (fase reversa), ajustando os pesos sinápticos da rede. O cálculo desse ajuste é simples na camada de saída, mais complexo nas camadas escondidas. A retro-propagação se provou um método computacionalmente eficiente para o treino das redes neurais MLP, se tornando um marco em seu desenvolvimento.

A configuração do treinamento de uma rede MLP envolve a definição de diversos hiperparâmetros, que influenciam diretamente a capacidade de aprendizado e o desempenho do modelo. Entre os principais, destacam-se:

- Número de camadas ocultas e quantidade de neurônios por camada: esses dois fatores determinam o poder de representação da rede, sendo que um número maior de camadas ou neurônios permite à rede MLP modelar relações mais complexas nos dados. Isso, porém, pode aumentar o risco de sobreajuste (overfitting) e exigir uma capacidade computacional maior. A escolha desses valores é geralmente feita por experimentação ou validação cruzada.
- Funções de ativação para as camadas de entradas, escondidas, e de saída: são responsáveis por introduzir não-linearidades no modelo, o que permite que a rede aprenda padrões complexos. Geralmente, a escolha da função para cada camada depende da tarefa, sendo, por exemplo, a função da camada de saída definida através de qual objetivo se busca alcançar, seja para resolver um problema binário ou de múltiplas classes.
- Tamanho da amostra (batch size): Define o número de amostras utilizadas antes da atualização dos pesos durante o treinamento. Valores menores aceleram o treinamento, mas podem gerar instabilidade; batches maiores são mais estáveis, porém mais exigentes computacionalmente.
- Quantidade de épocas (epochs): Representa o número de vezes que todo o conjunto de treinamento é apresentado à rede. Um número pequeno de épocas pode resultar em subajuste (underfitting), enquanto um número excessivo pode levar ao sobreajuste.

C. Extreme Learning Machine (ELM)

O Extreme Learning Machine (ELM), criado por Guang-Bin Huang em 2006 [22], é uma rede neural do tipo SLFN (single-hidden layer feedforward neural network), na qual contém apenas uma camada de entrada, escondida, e de saída, e utiliza apenas propagação para o seu treino, diferente da rede MLP, que utiliza retro-propagação.

Na ELM, os pesos entre a camada de entrada e a camada escondida são definidos aleatoriamente, e se mantêm fixos ao longo do treinamento, o que faz seu treinamento ser extremamente rápido, sem a necessidade de múltiplas amostras e épocas para o ajuste dos pesos. A única etapa de ajuste ocorre nos pesos da camada de saída, determinados por meio de solução de sistemas lineares, utilizando geralmente a pseudo-inversa de Moore-Penrose, que é a generalização mais conhecida da matriz inversa.

Assim, a rede ELM possui apenas três hiperparâmetros necessitando ajuste durante seu treinamento:

- Número de neurônio na camada escondida: definindo a capacidade da rede neural de captar os dados. Igual à rede MLP, uma quantidade maior resulta numa rede capaz de captar dados mais complexos, com um custo computacional maior.
- Função de ativação da camada oculta: neste caso, apenas a camada oculta recebe uma função de ativação, sendo que a escolha deste hiperparâmetro impacta diretamente na forma como os dados são transformados antes do ajuste da saída.

- Parâmetro de regularização (C): Controla o equilíbrio entre o erro de treinamento e a complexidade do modelo. Um valor alto de C favorece o ajuste perfeito aos dados, enquanto valores menores promovem melhor generalização, evitando sobreajuste.

D. Python

Criado por Guido van Rossum em 1991 [23], Python é uma linguagem orientada a objetos baseada na linguagem imperativa ABC. Foi projetada para ser simples, legível e eficiente, com foco na produtividade do desenvolvedor e na clareza do código.

É uma linguagem de alto nível, com tipagem dinâmica e suporte a múltiplos paradigmas, possuindo ampla aplicação em áreas como ciência de dados, automação, inteligência artificial e desenvolvimento web, e é reconhecida por sua vasta biblioteca e comunidade ativa.

E. TensorFlow e API Keras

O TensorFlow [24] é uma biblioteca de código aberto desenvolvida pelo Google, lançada em 2015, voltada para aprendizado de máquina, principalmente para o treino de redes neurais. Baseado em grafos computacionais, ele permite construir e treinar modelos com eficiência, aproveitando recursos de CPUs diversas e GPUs da NVIDIA. Sua arquitetura flexível oferece suporte a redes neurais profundas e aplicações de larga escala, sendo amplamente adotado tanto na indústria quanto em meios acadêmicos. Integrada oficialmente ao TensorFlow, a API Keras fornece uma interface de alto nível para construção de modelos de redes neurais, facilitando o desenvolvimento, a prototipagem e o ajuste fino dos modelos. Com foco em simplicidade e modularidade, Keras permite definir camadas, funções de perda, otimizadores e métricas com poucas linhas de código, mantendo o acesso à infraestrutura robusta do TensorFlow para execução e escalabilidade.

F. HPELM

O HPELM (High Performance Extreme Learning Machine - Máquina de Aprendizado Extremo de Alta Desempenho) [25] é uma biblioteca de código aberto desenvolvida em Python por Anton Akusok, Kaj-Mikael Björk, Yoan Miche e Amaury Lendasse, lançada em 2015, para treinamento de modelos ELM. Segundo os autores, sua abordagem não iterativa oferece um ganho de desempenho de até 5 ordens de magnitude em relação ao MLP e 6 ordens em relação ao Support Vector Machines (SVM).

A biblioteca suporta diferentes funções de ativação, como linear, sigmóide, tangente hiperbólica e funções baseadas em Radial Basis Function (RBF). Também permite a utilização de regularização por meio do parâmetro C, que controla a complexidade do modelo, além de oferecer suporte a múltiplos núcleos de processamento para acelerar o treinamento. O HPELM pode ser aplicado tanto em tarefas de classificação quanto de regressão, sendo projetado para lidar com grandes volumes de dados de forma eficiente, mantendo baixo custo computacional e alta precisão.

IV. METODOLOGIA

Neste trabalho, utilizamos a metodologia CRISP-DM como base para o fluxo de pesquisa e desenvolvimento do projeto, desde a aquisição da base de dados até o resultado de previsão das redes neurais. A figura 3 demonstra as etapas da construção e avaliação do projeto.

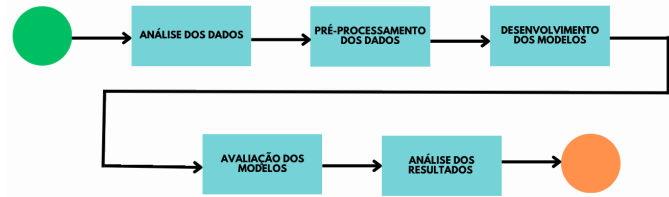


Figura 3. Metodologia Proposta

A. Análise dos Dados

Neste trabalho, utilizamos a base de dados disponibilizada pelo Instituto Politécnico de Portalegre (IPPortalegre), juntamente com seu artigo que fornece explicação detalhada sobre sua montagem e conteúdo [9]. Essa base contém dados de 4424 alunos de 17 cursos de graduação da instituição, com 35 atributos cada, divididos em dados acadêmicos, socioeconômicos, demográficos e macroeconômicos, além de seu dado alvo, que classifica os alunos como matriculados, evadidos ou graduados. Os dados referem-se aos registros de estudantes matriculados entre os anos letivos de 2008/2009 e 2018/2019, sendo esses seus primeiro e segundo semestres na instituição.

Segundo os autores, essa base foi montada através da coleta de dados do Sistema de Gestão Acadêmica (SGA) da IPPortalegre, o Sistema de Apoio à Atividade Docente (PAE), os dados anuais da Direção-Geral do Ensino Superior (DGES) relativos às admissões através do Concurso Nacional de Acesso ao Ensino Superior (CNAES), e a base de dados PORDATA, que disponibiliza indicadores macroeconômicos de Portugal. A figura a seguir mostra o fluxo de aquisição e montagem da base de dados, com as linguagens usadas para a montagem de cada etapa, sendo elas Visual Basic for Application para os dados do CNAES e Python para as outras.

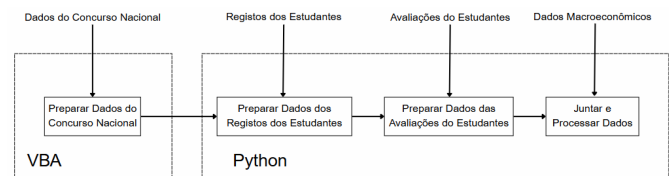


Figura 4. Montagem da base de dados

B. Pré-processamento dos dados

Inicialmente, verificou-se que a base de dados não apresenta valores ausentes nem registros duplicados. Além disso, ao analisar a quantidade de alunos classificados como matriculados e evadidos, constatou-se que os dados estão suficientemente balanceados, não sendo necessária a aplicação de técnicas de balanceamento, conforme ilustrado na figura a seguir.

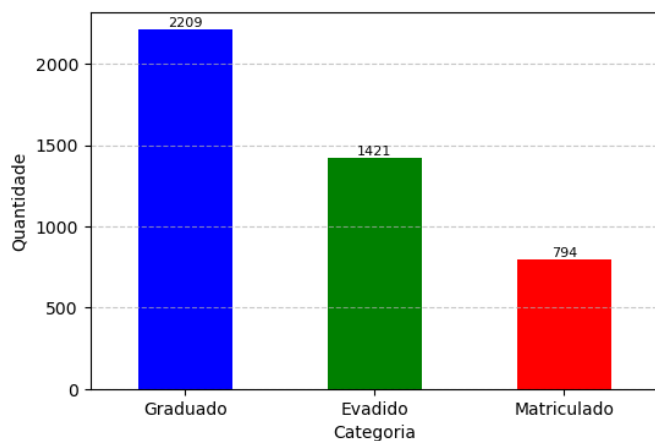


Figura 5. Distribuição das Classes no Alvo

Considerando os objetivos deste trabalho e a replicação do estudo realizado por Pedro Luiz Regattieri Da Cunha [10], identificou-se a necessidade de remover da base os alunos classificados como matriculados. A análise dos resultados indicou que a junção das classes graduados e matriculados, utilizando os mesmos hiperparâmetros empregados por Pedro em seu modelo MLP, resultou em um desempenho inferior em comparação ao modelo treinado sem essa junção. Tal comportamento sugere que a inclusão de alunos ainda matriculados, que não concluíram o curso mas também não evadiram, introduz ambiguidade na base de dados. Esses alunos podem apresentar perfis distintos dos alunos graduados, além de estarem sujeitos à possibilidade de evasão futura, o que compromete a qualidade da modelagem. Adicionalmente, a incorporação de mais registros na classe dos graduados pode provocar um desbalanceamento entre as classes, prejudicando o desempenho do modelo.

Observou-se, ainda, que com a adoção de diferentes combinações de hiperparâmetros, os resultados não tiveram uma mudança significativa, apenas a remoção dos alunos matriculados alterou significativamente os resultados, conforme apresentado nas tabelas a seguir.

Tabela I
DESEMPENHO DOS MODELOS MLP COM JUNCÃO DAS CATEGORIAS GRADUADO E MATRICULADO

Modelos MLP	Acurácia	Precisão	Recall	F1-Score
Pedro	88,10%	88,30%	95,10%	91,60%
Replicado	87,07%	88,70%	92,21%	90,42%
Melhor com Junção	87,43%	88,16%	93,58%	90,79%
Melhor sem Junção	91,73%	92,19%	94,65%	93,40%

C. Desenvolvimento dos Modelos

Após os testes iniciais, verificou-se que, para realizar a padronização dos dados, dois métodos distintos apresentaram melhor desempenho para cada modelo. Para a rede MLP, o uso do StandardScaler resultou em melhores desempenhos, enquanto para o modelo ELM o MinMaxScaler apresentou resultados superiores.

No modelo MLP, adotou-se a função de ativação Sigmoid na camada de saída, considerando que o problema tratado possui uma saída binária. Dessa forma, não se identificou a necessidade de testar outras funções para esta camada.

Os dados para treino e teste para ambos os modelos foram separados numa proporção 80:20 aleatoriamente, 2904 alunos para treino e 726 para teste, uma proporção amplamente usada para testes com modelos de aprendizagem de máquina e, considerando os resultados do trabalho de Achmad Ridwan [12], ideal para o uso nesta base de dados.

Para o treinamento da rede MLP, foi utilizado o método GridSearchCV, que realiza uma busca exaustiva sobre os hiperparâmetros definidos. Entretanto, devido ao alto custo computacional desse processo, mesmo com o uso de GPU via TensorFlow, os testes foram realizados uma única vez para cada configuração de quantidade e número de camadas ocultas. A Tabela II apresenta os hiperparâmetros utilizados na MLP.

Tabela II
HIPERPARÂMETROS UTILIZADOS NO MODELO MLP

Parâmetro	Valores
Camadas Ocultas	(10), (10,10), (10,10,10) (20), (20,20), (20,20,20) (50), (50,50), (50,50,50) (100), (100,100), (100,100,100) (200), (200,200), (200,200,200)
Ativação na Entrada	None, ReLU, Tanh
Ativação nas Ocultas	ReLU, Tanh, Leaky ReLU
Ativação na Saída	Sigmoid
Batch Size	32, 64, 128
Épocas	50, 100

Para o modelo ELM, foi realizada uma busca exaustiva via um laço de repetição que testou todas as combinações possíveis dos hiperparâmetros definidos. Devido ao baixo custo computacional desse modelo, foi possível executar todos os testes de maneira rápida e eficiente. A Tabela III apresenta os hiperparâmetros utilizados no ELM.

Tabela III
HIPERPARÂMETROS UTILIZADOS NO MODELO ELM

Parâmetro	Valores
Número de Neurônios	10, 20, 50, 100, 200, 500, 1000
Função de Ativação	Sigmoid (sigm), Tanh, Linear (lin), RBF L2 (rbf_l2), RBF L1 (rbf_l1), RBF Linf (rbf_linf)
Parâmetro de Regularização (C)	1, 10, 50, 100

D. Avaliação dos Modelos

As métricas escolhidas para a avaliação dos modelos MLP e ELM se baseiam em uma métrica chamada de Matriz de Confusão, que indica, de maneira clara e simples, o desempenho do modelo de acordo com resultados obtidos, os comparando com dados reais.

		PREVISÃO DO MODELO	
		Evasão (0)	Graduado (1)
REAL	Evasão (0)	Verdadeiro Negativo (VN)	Falso Positivo (FP)
	Graduado (1)	Falso Negativo (FN)	Verdadeiro Positivo (VP)

Onde Verdadeiro Negativo e Verdadeiro Positivo indicam que o modelo previu corretamente a classificação dos alunos em Evadido ou Graduado, e Falso Positivo e Falso Negativo onde ele previu incorretamente. Dessas, um Falso Positivo é o mais grave, pois indica que alunos identificados como Evadido foram classificados como Graduados. Com base nos resultados da Matriz de Confusão, é possível calcular as métricas mais amplamente utilizadas para a avaliação de modelos de aprendizado de máquina:

- Acurácia: determina a porcentagem total de acertos do modelo.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2)$$

- Precisão: verificação dos dados classificados como positivos, indicando a porcentagem de quantos são realmente positivos.

$$\text{Precisão} = \frac{VP}{VP + FP} \quad (3)$$

- Recall: indica qual a porcentagem de dados classificados como positivos comparados com a quantidade real de positivos que existem em nossa amostra.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (4)$$

- F1-Score: une precisão e recall de modo a trazer um número único que determine a qualidade geral do modelo.

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \quad (5)$$

V. RESULTADOS

Depois de executados os modelos, os resultados apresentados nas tabelas IV, V e VI foram obtidos.

Tabela IV
MATRIZ DE CONFUSÃO - MLP

		Previsão do Modelo	
		Evasão (0)	Graduado (1)
Real	Evasão (0)	241 (VN)	36 (FP)
	Graduado (1)	24 (FN)	425 (VP)

Tabela V
MATRIZ DE CONFUSÃO - ELM

		Previsão do Modelo	
		Evasão (0)	Graduado (1)
Real	Evasão (0)	226 (VN)	51 (FP)
	Graduado (1)	8 (FN)	441 (VP)

Tabela VI
MELHORES HIPERPARÂMETROS E MÉTRICAS DOS MODELOS MLP E ELM

Parâmetro / Métrica	MLP	ELM
Ativação Entrada	-	-
Ativação Oculta	Leaky ReLU	rbf_l2
Ativação Saída	Sigmoid	-
Tamanho da amostra (Batch)	128	-
Épocas	50	-
Neurônios por camada	(20, 20)	100
Parâmetro C	-	1
Acurácia	91.74%	91.87%
Precisão	92.19%	89.63%
Recall	94.65%	98.22%
F1-Score	93.41%	93.73%
Tempo de Treinamento (Todas as Métricas)	4h 57m 28s	6.39s

Na análise dos resultados, ambos os modelos apresentaram acurácia pouco acima de 91%, o que indica um desempenho satisfatório. No entanto, focando no problema de evasão para as instituições de ensino, observa-se que os modelos possuem resultados divergentes. O modelo MLP apresentou um total de 36 Falsos Positivos, ou seja, alunos que foram classificados como Graduados, mas na realidade evadiram, e obteve 24 Falsos Negativos. Em comparação, o modelo ELM apresentou um número maior de Falsos Positivos, 51, indicando menor capacidade em identificar corretamente os alunos em risco de evasão, comparado ao MLP. Entretanto, o ELM compensou isso com apenas 8 Falsos Negativos, mostrando uma forte capacidade evitar classificar Graduados como Evadidos.

Apesar do ELM apresentar um número maior de falsos positivos, ele se destaca de forma significativa no tempo de treinamento. Enquanto o MLP exigiu cerca de 4 horas e 57 minutos para treinar todos os conjuntos de hiperparâmetros testados, o ELM realizou esse processo em apenas 6,39 segundos, o que demonstra uma eficiência computacional muito superior.

VI. CONCLUSÃO

Em termos gerais, o MLP oferece um melhor equilíbrio entre erros de classificação, especialmente na redução dos falsos positivos, o que é desejável em cenários onde evitar a evasão é prioritário. Já o ELM, embora mais suscetível a esse tipo de erro, demonstra uma excelente capacidade de não deixar alunos em risco passarem despercebidos, além de ser extremamente rápido no processo de treinamento.

Dessa forma, a escolha entre os modelos pode variar conforme a necessidade institucional: priorizar menos falsos positivos e aceitar um tempo de processamento mais elevado (MLP), ou optar por uma solução muito mais rápida, com excelente detecção de evasão, mas com maior risco de classificar erroneamente alguns alunos que tem chance de evadir (ELM).

REFERENCIA

[1] Instituto Semesp, "14ª edição do Mapa do Ensino Superior no Brasil," São Paulo, 2024. [Online]. Available: <https://www.semesp.org.br/mapa/edicao-14/>. Accessed: Apr. 20, 2025.

[2] Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira, "Relatório de Pesquisa: Resposta

Educacional a Pandemia de Covid-19 no Brasil – Educacao Superior – 2020,"Brasilia, 2022. [Online]. Available: <https://www.gov.br/inep/pt-br/centrais-de-conteudo/acervo-linha-editorial/publicacoes-institucionais/estatisticas-e-indicadores-educacionais/relatorio-de-pesquisa-resposta-educacional-a-pandemia-de-covid-19-no-brasil-educacao-superior-2020>. Accessed: Apr. 20, 2025.

[3] Brasil, Ministerio da Educacao, "Portaria MEC no 39, de 12 de dezembro de 2007. Institui o Programa Nacional de Assistencia Estudantil – PNAES,"Diario Oficial da Uniao, Brasilia, DF, Dec. 13, 2007. [Online]. Available: <https://www.fn.de.gov.br/aceso-a-informacao/institucional/legislacao/item/3230-portaria-mec-n-39-de-12-de-dezembro-de-2007>. Accessed: Apr. 23, 2025.

[4] Brasil, Presidencia da Republica, "Lei no 14.914, de 17 de abril de 2024. Institui a Politica Nacional de Assistencia Estudantil (PNAES),"Diario Oficial da Uniao, Brasilia, DF, Apr. 18, 2024. [Online]. Available: https://www.planalto.gov.br/ccivil_03/_Ato2023-2026/2024/Lei/L14914.htm. Accessed: Apr. 23, 2025.

[5] Universidade de Pernambuco – UPE, "Edital PROGRAD UPE no 03/2023 – Bolsa de Auxilio Deslocamento,"[Online]. Available: https://upe.br/anexos/prograd/auxilio_deslocamento/2023/EDITAL_PROGRAD_N-03-2023_-_BOLSA_AUXILIO_DESLOCAMENTO.pdf. Accessed: Apr. 23, 2025.

[6] Universidade de Pernambuco – UPE, "Edital PROGRAD UPE no 04/2023 – Bolsa de Moradia,"[Online]. Available: https://upe.br/anexos/programas/aux_moradia/EDITAL_MORADIA_2023.pdf. Accessed: Apr. 23, 2025.

[7] Universidade Federal do Amazonas, "Plano de assistencia estudantil: acoes e estrategias de permanencia discente,"Manaus: UFAM, 2022. [Online]. Available: <https://edoc.ufam.edu.br/bitstream/123456789/7902/4/Academico.pdf>. Accessed: Apr. 29, 2025.

[8] Universidade Federal do Rio Grande do Sul, "Edital no 01/2025 – Beneficios e Moradia,"Porto Alegre: UFRGS, 2025. [Online]. Available: <https://www.ufrgs.br/prae/wp-content/uploads/2025/02/EDITAL-n.-01-2025-BENEFICIOSeMORADIAassinado.pdf>. Accessed: Apr. 29, 2025.

[9] V. Realinho, M. Vieira Martins, J. Machado and L. Baptista, "Predict Students' Dropout and Academic Success,"UCI Machine Learning Repository, 2021. [Online]. Available: <https://doi.org/10.24432/C5MC89>. Accessed: Apr. 20, 2025.

[10] P. L. R. Cunha, L. A. Scardua and C. Z. Resende, "Tecnicas de inteligencia artificial na predicao de evasao escolar: um estudo exploratorio,"in *Engenharia: Solucoes e Inovacoes para o Desenvolvimento*, P. F. Pires, Ed. Piracanjuba-GO: Editora Conhecimento Livre, 2024, ch. 2. doi: 10.37423/240508998.

[11] A. Alhardi and S. Alan, "Predicting student dropout in higher education using machine learning techniques: a predictive model using XGBoost algorithm,"2024.

[12] A. Ridwan, A. M. Priyatno and L. Ningsih, "Predict Students' Dropout and Academic Success with XGBo-

ost,"*Journal of Education and Computer Applications*, vol. 1, no. 2, pp. 1–8, 2024. doi: 10.69693/jeca.v1i2.13. Accessed: Apr. 15, 2025.

[13] S. Kim, E. Yoo and S. Kim, "Why Do Students Drop Out? University Dropout Prediction and Associated Factor Analysis Using Machine Learning Techniques,"2023. doi: 10.48550/arXiv.2310.10987. Accessed: Apr. 14, 2025.

[14] T. H. Nguyen, P. Le, T. T. T. Nguyen and A. K. Su, "A multivariate analysis of the early dropout using classical machine learning and local interpretable model-agnostic explanations,"*CTU Journal of Innovation and Sustainable Development*, vol. 16, Special issue ISDS, pp. 98–106, 2024. doi: 10.22144/ctujoisd.2024.327. Accessed: Apr. 15, 2025.

[15] E. G. Cruz Sena and H. P. Rocha, "Classificacao de pacientes com COVID-19 utilizando tecnicas de aprendizado de maquina,"in *Congresso Brasileiro de Inteligencia Computacional (CBIC)*, 2021. doi: 10.21528/CBIC2021-151.

[16] T. M. Barchi et al., "Aplicacao de redes neurais e modelos preditivos na previsao do preco do acucar no Brasil,"in *Congresso Brasileiro de Inteligencia Computacional (CBIC)*, 2021. doi: 10.21528/CBIC2021-81.

[17] L. Moraes, A. Gomes and J. V. Andrade, "Diagnostico precoce de Alzheimer utilizando redes neurais e dados sinteticos,"in *Congresso Brasileiro de Inteligencia Computacional (CBIC)*, 2021. doi: 10.21528/CBIC2021-23.

[18] B. S. Moura et al., "Aplicacao de redes neurais convolucionais 2D e graficos de recorrência para previsao da geracao de energia fotovoltaica,"in *Congresso Brasileiro de Inteligencia Computacional (CBIC)*, 2023. doi: 10.21528/CBIC2023-014.

[19] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity,"*Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 1943. doi: 10.1007/BF02478259.

[20] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2009.

[21] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain,"*Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. doi: 10.1037/h0042519.

[22] G. B. Huang, Q. Y. Zhu and C. K. Siew, "Extreme learning machine: theory and applications,"*Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006. doi: 10.1016/j.neucom.2005.12.126.

[23] Python Software Foundation, "Python Frequently Asked Questions,"[Online]. Available: <https://docs.python.org/3/faq/general.html>. Accessed: May 20, 2025.

[24] M. Abadi et al., "TensorFlow: A system for large-scale machine learning,"*arXiv preprint*, arXiv:1603.04467, 2016. doi: 10.48550/arXiv.1603.04467.

[25] A. Akusok, K. M. Bjork, Y. Miche and A. Lendasse, "High-performance extreme learning machines: a complete toolbox for big data applications,"*IEEE Access*, vol. 3, pp. 1011–1025, 2015. doi: 10.1109/ACCESS.2015.2450498.