UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isabelli Prudencio Tedeschi

Previsão de evasão estudantil na Faculdade de Computação utilizando mineração de dados

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Isabelli Prudencio Tedeschi

Previsão de evasão estudantil na Faculdade de Computação utilizando mineração de dados

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciências da Computação.

Orientador: Paulo Henrique Ribeiro Gabriel

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciências da Computação

Uberlândia, Brasil

2025

Resumo

A evasão escolar no ensino superior é um problema crescente no Brasil. Segundo o Mapa do Ensino Superior, divulgado pelo Semesp em 2024, cerca de 57,2% dos estudantes não concluíram a graduação entre 2018 e 2022. Diante desse cenário, este trabalho utiliza técnicas de mineração de dados, aprendizado de máquina e análise estatística para descobrir padrões e informações relevantes a partir de um conjunto de dados reais dos discentes do Bacharelado em Ciência da Computação (BCC), fornecido pela coordenação do curso de Ciência da Computação da Faculdade de Computação da Universidade Federal de Uberlândia (FACOM-UFU). A base de dados abrange um período de 27 anos e contempla o histórico acadêmico de cerca de 1.500 estudantes, incluindo informações como disciplinas cursadas, notas obtidas, carga horária cumprida, entre outros indicadores. Com base nos dados acadêmicos dos primeiros três primeiros meses de curso, foram aplicadas técnicas de balanceamento, combinadas a diversos algoritmos de aprendizado supervisionado, com o objetivo de criar modelos capazes de identificar quais alunos apresentam maior risco de evasão no ensino superior. O objetivo central é avaliar o desempenho desses modelos, de modo que possam futuramente apoiar a instituição na tomada de decisões voltadas ao bem-estar dos discentes em situação de vulnerabilidade acadêmica. Os resultados mostram que os algoritmos, de forma geral, apresentaram bom desempenho, com estabilidade na identificação das classes do conjunto de dados, ainda que tenham demonstrado pouca sensibilidade às técnicas de balanceamento aplicadas, com destaque aos modelos Floresta Aleatória, Naive Bayes e MLP, embora com diferenças pouco expressivas entre eles.

Palavras-chave: Evasão Escolar, Ensino Superior, Aprendizado de Máquina, Classificação Supervisionado e Modelos Preditivos.

Abstract

Student dropout in higher education is a growing problem in Brazil. According to the Higher Education Map, published by Semesp in 2024, around 57.2% of students did not complete their undergraduate degrees between 2018 and 2022. In this context, this study employs data mining, machine learning, and statistical analysis techniques to uncover patterns and relevant information from a real dataset of students enrolled in the Bachelor's Degree in Computer Science (BCC), provided by the coordination of the Computer Science course at the Faculty of Computing of the Federal University of Uberlândia (FACOM-UFU). The dataset spans a period of 27 years and includes the academic records of approximately 1,500 students, containing information such as completed courses, grades, completed credit hours, among other indicators. Based on academic data from the first three months of the course, balancing techniques were applied in combination with several supervised learning algorithms to develop models capable of identifying students at higher risk of dropping out of higher education. The main goal is to evaluate the performance of these models so they may eventually support the institution in decision-making aimed at promoting the well-being of students in situations of academic vulnerability. The results show that the algorithms generally performed well, with stable identification of classes within the dataset, although they showed limited sensitivity to the applied balancing techniques. Random Forest, Naive Bayes, and MLP models stood out, albeit with only minor differences among them.

Keywords: school dropout, higher education, machine learning, supervised classification, predictive models.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

UFU	Universidade Federal de Uberlândia
Sisu	Sistema de Seleção Unificada
FACOM	Faculdade de Computação
MGA	Média Geral Acadêmica
MEC	Ministério da Educação
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
AM	Aprendizado de Máquina
IA	Inteligência Artificial
MSE	*Mean Squared Error* (Erro Quadrático Médio)
SVM	*Support Vector Machine* (Máquinas de Vetores de Suporte)
RNA	Rede Neural Artificial
MLP	*Multilayer Perceptron* (Perceptron Multicamadas)
ETL	*Extract, Transform and Load* (Extrair, Transformar e Carregar)
NDA	Nenhuma Técnica de Balanceamento Aplicada
SMOTE	*Synthetic Minority Over-sampling Technique*
RO	*Random Oversampling*
WC	*Weighted Class*
TI	Tecnologia da Informação
MSE	Erro Quadrático Médio
AUC	*Area Under the Curve* (Área Sob a Curva)
ROC	*Receiver Operating Characteristic* (Característica de Operação do Receptor)
BCC	Bacharelado em Ciências da Computação

Sumário

1 Introdução

A evasão estudantil é um problema que tem atingido recorrentemente as instituições de ensino superior do Brasil, conforme o Resumo Técnico do Censo da Educação Superior disponibilizado pelo INEP (MEC, 2023). O termo é definido pelo Ministério da Educação (BORDAS, 1996) como "a saída definitiva do estudante do curso de origem, sem concluí-lo", seja ela por motivos internos ou externos. Segundo Mello et al. (2013), a evasão por fatores internos se caracteriza por desistência devido ao método didático utilizado ou ao descontentamento com a infraestrutura da universidade. Os fatores externos são os que envolvem a vida pessoal do aluno, como dificuldade de adaptação ou problemas financeiros e sociais (MELLO et al., 2013). O Semesp (2022) mostra que uma das áreas que apresentou maior evasão em 2022 se relaciona aos cursos de Tecnologia da Informação (TI). Muitos estudantes ingressam nos cursos de TI com expectativas que não condizem com a realidade acadêmica e profissional, levando à frustração e eventual abandono (MARINHO, 2024). Compreender os motivos que levam um discente à evasão, independentemente de quais sejam, é fundamental para que os recursos institucionais sejam direcionados de maneira eficaz no enfrentamento desse problema.

A educação é a chave para o progresso da sociedade, que influencia aspectos como desenvolvimentos ecológicos, socioeconômicos e científicos. É classificada, também, como a base para uma sociedade, pois promove a coexistência, tolerância, participação crítica e criativa dos cidadãos justa, culta e com razão de mudança (UNESCO, 1997). De acordo com o Ministério da Educação (MEC, 2012), as universidades são consideradas patrimônio social público, e tem o dever de responder às necessidades sociais por meio da pesquisa e do ensino de qualidade. Esses dados evidenciam que a evasão escolar se torna não somente um ônus para a sociedade, como, também, um desperdício de verba pública, já que ocorre um uso inadequado da vaga, sinalizando equívocos na orientação profissional dos ingressantes (GOMES et al., 2010). O índice médio de evasão no período de 2000 a 2012, na UFU, estava em torno de 20%, sendo que alguns cursos apresentam taxas consideravelmente altas, como os cursos na área de Ciências Matemáticas e Naturais: Estatística (76,50%); Física Médica (69,55%); e Química (66,79%) (MOURA, 2018). Como discutido, entender os motivos da evasão escolar e conseguir predizer quais discentes têm maior probabilidade de abandonarem o curso, ajuda não só a universidade a direcionar melhor suas políticas de recursos, como ajuda o aluno a se entender melhor e procurar a melhor forma de continuar os estudos de maneira edificadora.

A mineração de dados é um processo que utiliza aprendizado de máquina e análise estatística para identificar padrões e extrair informações relevantes a partir de um conjunto de dados. De acordo com IBM (2025), as técnicas de mineração de dados podem ter duas

principais finalidades: descrever o conjunto de dados analisado ou prever resultados com base em algoritmos de aprendizado de máquina.

O uso da mineração de dados é amplamente difundido em diversas áreas, conforme destacado por IBM (2025). Entre suas aplicações, destacam-se a detecção de anomalias, a avaliação de riscos no mercado financeiro e a melhoria no atendimento ao cliente. Além disso, no campo da educação, instituições de ensino têm utilizado a mineração de dados para coletar e analisar informações sobre suas populações estudantis, possibilitando a tomada de decisões estratégicas para a criação de um ambiente mais propício ao sucesso acadêmico. Neste trabalho, o aprendizado de máquina foi empregado para identificar padrões e correlações na base de dados disponibilizada, permitindo a construção de modelos capazes de realizar previsões a partir de novos dados inseridos no sistema.

Nesta monografia o objetivo é de classificar e, assim, predizer quais são as situações que mais contribuem para a evasão dos discentes dentro da Faculdade de Computação, aplicando técnicas de Mineração de Dados, de modo que seja possível identificar os alunos que correm maior risco de abandonar o curso, entender suas motivações para tal ato e tentar prevenir que tal situação ocorra. Os objetivos específicos do trabalho se baseiam em selecionar as melhores técnicas de classificação de dados dado o problema, tomando como base outros trabalhos relacionados; avaliar empiricamente os algoritmos e técnicas escolhidas, com o propósito de encontrar os modelos mais eficientes; e por último analisar as informações resultantes, para verificar a influência do desempenho acadêmico na evasão estudantil. É importante ressaltar que as descobertas têm como propósito auxiliar os responsáveis da instituição a implementar políticas específicas que tenham a finalidade de reduzir a taxa de evasão e auxiliar, também, os alunos a entenderem quais são as maiores dificuldades dos cursos do BCC para aprenderem a lidar da melhor forma com os empecilhos apresentados.

Para a realização deste trabalho, foi utilizada uma base de dados disponibilizada pela coordenação do curso de Ciência da Computação da Universidade Federal de Uberlândia (UFU). Essa base contém informações coletadas ao longo de 27 anos, abrangendo cerca de 1.500 discentes. Os dados incluem histórico de disciplinas cursadas, notas obtidas, carga horária cumprida, entre outros indicadores relevantes. Com base nessas informações, foi possível calcular o desempenho acadêmico dos alunos ao longo dos três primeiros semestres, por meio da fórmula da Média Geral Acadêmica (MGA). Os valores obtidos a partir desses cálculos foram utilizados como três atributos principais na construção dos modelos preditivos de evasão.

A implementação foi desenvolvida utilizando a linguagem de programação Python, devido à sua ampla variedade de bibliotecas voltadas à manipulação de dados e aprendizado de máquina. Dentre elas, destacam-se a biblioteca pandas, empregada na organização e pré-processamento dos dados, e a *scikit-learn*, utilizada nas etapas de balanceamento,

treinamento e classificação.

Inicialmente foram selecionados os atributos que melhor seriam aproveitados pela classificação. Com os atributos já selecionados foi feita a limpeza dos dados, retirando da base todos os elementos que não se encaixavam ou que poderiam prejudir o modelo. A base, assim, foi submetida a três métodos de balanceamento de classes, ao todo foram utilizadas quatro metodologias, entre elas o *Random Oversampler*, SMOTE, *weight-classes* e a base original.Em seguida, foram empregado os algoritmos Floresta Aleatória, Árvore de Decisão, Regressão Logística, Regressão Linear, *Naive Bayes*, Rede Neural Perceptron Multicamadas e *Support Vector Machine*, que realizaram a classificação dos dados e criação dos modelos. Por fim, para a avaliação dos modelos, foram utilizadas as métricas acurácia, sensibilidade (*recall*), precisão e matriz de confusão.

Os resultados mostraram que os algorimos que tiveram melhor desempenho, considerando os métodos de avaliação, foram Perceptron Multicamadas, Naive Bayes e Floresta Aleatória e em sua maioria utilizando os métodos do *SMOTE* e sem nenhuma modificação da base original. No final, as diferenças de desempenho entre as técnicas de balanceamento e os classificadores foram pouco expressivas, reforçando que a utilização de um conjunto de dados mais robusto e de maior qualidade pode levar a melhores resultados. Ainda assim, os modelos cumpriram seu objetivo ao predizer, com altas taxas de sucesso, os alunos com maior probabilidade de evasão.

O restante desta monografia está organizado da seguinte forma: o Capítulo 2 apresenta a fundamentação teórica, abordando os principais conceitos utilizados ao longo da pesquisa. São discutidos tópicos relacionados à estrutura acadêmica da UFU, como o funcionamento da Matrícula e do Histórico Acadêmico, além de conceitos de Inteligência Artificial, Aprendizado de Máquina Supervisionado, Algoritmos de Classificação aplicados neste trabalho e as principais Métricas de Avaliação de Desempenho. No Capítulo 3, são descritas as etapas de construção dos modelos de classificação, incluindo o pré-processamento dos dados, com eliminação de atributos, limpeza e transformação dos dados, aplicação de técnicas de balanceamento e normalização, além da geração dos modelos preditivos. O Capítulo 4 é dedicado à apresentação e análise dos resultados, com a interpretação dos desempenhos dos modelos propostos. Por fim, o Capítulo 5 reúne as conclusões baseadas nos experimentos realizados, destacando a relevância deste estudo e sugerindo possíveis caminhos para pesquisas futuras.

2 Fundamentação teórica

Este capítulo apresenta os fundamentos teóricos que sustentam a realização deste trabalho. Inicialmente, são discutidos os conceitos relacionados à matrícula e ao histórico acadêmico, contextualizando o problema da evasão escolar no ensino superior. Em seguida, são abordados os princípios da Inteligência Artificial, com foco em aprendizado de máquina, mais especificamente no aprendizado supervisionado, que é a abordagem adotada neste estudo.

São também descritos os principais algoritmos de classificação utilizados para a construção dos modelos preditivos, incluindo Árvore de Decisão, Floresta Aleatória, *Naive Bayes*, Máquina de Vetores de Suporte (SVM), Regressão Logística, Regressão Linear e Perceptron Multicamadas (MLP). Além disso, são apresentadas as técnicas de balanceamento de dados aplicadas para lidar com a desproporção entre as classes no conjunto de dados. Por fim, são descritas as métricas utilizadas para avaliar o desempenho dos modelos, como acurácia, precisão, sensibilidade (*recall*) e *F1-score*.

2.1 Matrícula e Histórico Acadêmico

Para ingressar na graduação da UFU, cada aluno é submetido a um processo de seleção que consiste de três formas de ingresso: Sistema de Seleção Unificado (SiSU), o Vestibular tradicional da UFU ou ingresso por portador de diploma. Após ingressar na universidade, o curso é finalizado quando integralizar a carga horária necessária, representada por créditos. No registro do aluno, ele é representado por um número de matrícula e um histórico associado, onde contém seu desempenho nas disciplinas enquanto usa essa matrícula (UFU, 2024).

O curso de escolha do estudante tem uma grade curricular, composta por disciplinas, tanto obrigatórias quanto optativas. A grade é dividida em períodos. Para obter o título de bacharel ou licenciado, o aluno deve ser aprovado em todas as disciplinas e obter a quantidade de horas necessárias para se formar.

No caso do BCC-UFU, o curso é dividido em oito períodos com, em média, seis disciplinas por semestre. O aluno tem a possibilidade de realizar as disciplinas de acordo com o proposto na grade ou modificar seus horários, de modo que não ultrapassem o limite de anos para formar e não reprovar em uma mesma disciplina mais de um número determinado de vezes (FACOM/UFU, 2025).

Neste trabalho, a avaliação semestral é realizada por meio da fórmula da Média Geral Acadêmica (MGA), que considera a média final e a carga horária das disciplinas

cursadas pelo aluno. De acordo com a padronização da UFU, quanto mais alto um MGA melhor é o desempenho no semestre. No critério de aprovação, um aluno é considerado aprovado se sua média final for maior ou igual a 60; caso contrário, ele é reprovado. A fórmula utilizada para o cálculo do MGA é dada pela equação 2.1 (UFU, 2022).

$$MGA = \frac{\sum(Nota \times C_i)}{\sum C_i} \tag{2.1}$$

Onde:

- $Nota$: Resultado da avaliação do discente nas atividades desenvolvidas no componente curricular;
- C_i: Carga horária cursada (componentes curriculares cursados com aprovação e componentes curriculares cursados com reprovação).

2.2 Evasão Escolar no Ensino Superior

Evasão escolar é definida pelo Ministério da Educação (MEC, 2012) como saída antecipada, antes da conclusão do ano, série ou ciclo, por desistência (independentemente do motivo), representando, portanto, condição terminativa de insucesso em relação ao objetivo de promover o aluno ao ensino superior (INEP, 2017). Segundo Santos, Pereira e Borges (2023), alguns dos fatores que influenciam o acadêmico a evadir está: razões prévias à entrada na faculdade, como escolaridade e contexto familiar; a relação entre a instituição e o aluno, como performance acadêmica, interação com os funcionários da instituição e atividades extracurriculares; e fatores que envolvem a vida pessoal do aluno, como dificuldade de adaptação ou problemas financeiros e sociais.

A evasão é um desafio significativo em todos os níveis de ensino, independentemente de ser público ou privado. O abandono escolar acarreta perdas tanto para a sociedade, em termos de recursos investidos no ambiente acadêmico, quanto para os próprios alunos, que, sem uma educação superior, contribuem para o agravamento de problemas econômicos e sociais no país (LOBO, 2012). Segundo o Semesp (2024), no Mapa do Ensino Superior, cerca de 57,2% dos estudantes não concluíram a graduação entre 2018 e 2022. Esse índice é ainda maior nas instituições privadas, atingindo 60,8%. Dentre esses dados, a Sindicato das Entidades Mantenedoras de Ensino Superior de São Paulo (SEMESP, 2022), aponta que uma das áreas que teve a maior evasão em 2022, porém mantém essa tendência até os tempos atuais, são os cursos relacionados a TI (técnico em informática), de acordo com a o Mapa do Ensino Superior no Brasil foi contastado que a taxa de abandono nos cursos de TI presenciais chega a 38,5%, comparada à média geral de 30,7% em outras áreas. Dados que mostram essa tendência, de acordo com Hoed (2016), apenas um

a cada três ingressantes no curso de Sistemas de Informação irá concluir o curso. Independente das motivações do discente, é importante entender as causas que o levaram a evadir, para, assim, conseguir direcionar, de forma efetiva, os recursos a combater o problema.

Ao estudar as medidas que estão sendo realizadas para a prevenção da evasão estudantil no ensino superior, é observado que na graduação, mesmo que não resolvido todo o problema, o Estado buscou injetar recursos financeiros por meio de políticas específicas. De modo que o foco esteja na criação de medidas assistencialistas, com centralidade no atendimento aos estudantes em vulnerabilidade socioeconômica (SANTOS JUNIOR; MAGALHÃES; REAL, 2020). Desse modo, as ações para diminuição da frêquencia de abandono de ensino superior são centralizadas no Ministério da Educação (MEC), no órgão gestor central, e nas universidades, os órgãos gestores locais. Entre seus deveres, o de subsidiar o funcionamento das Instituições de Ensino Superior (IES) a partir da formulação de políticas mais amplas para a graduação e se atentar para questões do cotidiano universitário, implementando estratégias que visem não só a permanência, mas, também, a conclusão do curso (SANTOS JUNIOR; MAGALHÃES; REAL, 2020).

2.3 Inteligência Artificial e Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA) que visa desenvolver técnicas e sistemas capazes de adquirir conhecimento de forma automática. Em outras palavras, trata-se de um sistema que aprende a partir de dados históricos, identificando padrões e tomando decisões com base em experiências passadas (MONARD; BARANAUSKAS, 2003). Aprendizado de máquina permite que o sistema aprenda sem a intervenção humana e possui uma grande quantidade de aplicações (RASTROLLO-GUERRERO; GÓMEZ-PULIDO; DURÁN-DOMÍNGUEZ, 2020).

A história da inteligência artificial (IA) com foco no aprendizado de máquina (AM) remonta à década de 1970, período em que houve uma maior disseminação do uso de técnicas de computação baseadas em IA para a solução de problemas reais. Antes, os problemas eram tratados computacionalmente, em sua maioria, com ajuda de especialistas do problema dado, normalmente usando regras lógicas, conhecidos como Sistemas Especialistas ou Sistemas Baseados em Conhecimento. Nesses sistemas, eram conduzidas entrevistas com especialistas para entender e documentar a lógica e conjunto de regras utilizadas na tomada de decisões. No entanto, essa abordagem apresenta limitações, como a resistência dos especialistas em compartilhar conhecimento, por receio de se tornar dispensável, e a subjetividade das regras, por vezes feitas com base na intuição e experiência dos profissionais. (KONONENKO, 2001).

Com a crescente complexidade dos problemas e do volume dos problemas a serem tratados, tornou-se essencial o desenvolvimento de ferramentas computacionais sofisti-

cadas que reduzissem a dependência de especialistas e da intervenção humana. Assim, surgiram métodos capazes de inferir hipóteses ou funções a partir de experiências passadas, processo denominado Aprendizado de Máquina (FACELI et al., 2011).

O avanço do AM foi impulsionado pela evolução dos computadores eletrônicos na segunda metade do século XX, com o uso dos computadores eletrônicos e desenvolvimento dos algoritmos que permitiram modelar e analisar grandes conjuntos de dados (KONONENKO, 2001). A partir disso, três principais ramos do AM se consolidaram: a aprendizagem simbólica, descrito por Hunt, Marin e Stone (1966); em métodos estatísticos de Nilsson (1965); e as redes neurais, desenvolvidas por Rosenblatt (1962) (KONONENKO, 2001). Desde então, com os avanços de poder computacional e aumento de dados disponíveis para pesquisa, a área de inteligência artificial demonstrou avanços impressionantes, essa área é utilizada para solucionar inúmeros problemas tecnológicos e econômicos, por exemplo, sendo o Aprendizado de Máquina um dos principais responsáveis por esse progresso (LUDERMIR, 2021).

No presente trabalho, investiga-se a capacidade de diversos modelos utilizando Aprendizado de Máquina para fazer predições em ambiente de educação superior, para analisar, dado um conjunto de atributos, quais alunos são mais propensos a evadir da faculdade de Computação. O uso do AM nesta área permite identificar padrões nos perfis estudantis e prever o risco de desistência, auxiliando instituições de ensino na formulação de estratégias para mitigar esse problema (TEODORO; KAPPEL, 2020).

2.3.1 Aprendizado Supervisionado

Algoritmos de aprendizado utilizam diferentes paradigmas para solucionar problemas. De acordo com esse critério, as tarefas de aprendizado podem ser classificadas como Preditivas ou Descritivas (FACELI et al., 2011). Em tarefas preditivas, os algoritmos são treinados utilizando dados de treinamento rotulados para desenvolver um modelo preditivo capaz de analisar novos dados e realizar previsões com base nas informações extraídas do treinamento realizado com as amostras de treinamento (MANDELLI, 2023). Neste trabalho, será abordado com maior detalhamento o método de classificação, que é uma subdivisão do aprendizado supervisionado.

Esses algoritmos seguem o paradigma de aprendizado supervisionado, termo que faz referência à presença de um "supervisor externo", que conhece a saída (rótulo) esperada para cada conjunto de valores dos atributos de entrada. Dessa forma, o supervisor avalia a capacidade da hipótese induzida de predizer corretamente a saída para novos exemplos (FACELI et al., 2011).

Alguns dos algoritmos encontrados neste método são, Árvore de Decisão, classificador representado pela partição recursiva do conjunto de instâncias; Regressão Linear,

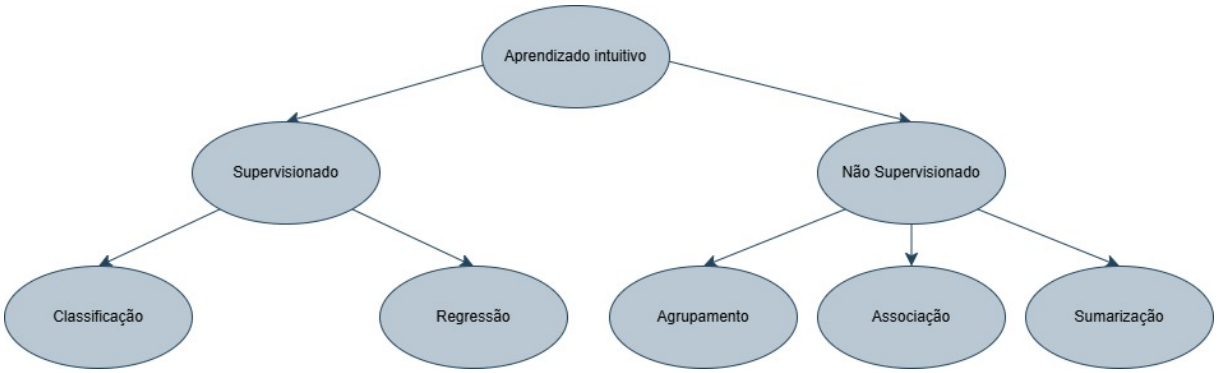
algoritmo de regressão que encontra relação e dependência entre variáveis; *Naive Bayes*, classificador mais usado para fins estatísticos; entre outros exemplos (NASTESKI, 2017).

Esse tópico de aprendizado tem diversas aplicações, pode ser usado para avaliar risco em serviços financeiros, classificação de imagens, identificar se transações realizadas pelo usuário são autênticas, capacidade de identificar objetos (BANOULA, 2023). Todos esses feitos são realizados a partir de detecção de padrões e, diante deles, a criação de modelos.

A Figura 1 a seguir ilustra a hierarquia do aprendizado. O nó raiz representa o processo de aprendizado indutivo. Dos nós internos, há a divisão entre aprendizado supervisionado, responsável por tarefas preditivas, como classificação e regressão, e aprendizado não supervisionado, associado a tarefas descritivas, como agrupamento, associação e sumarização.

Figura 1 – Hierarquia de Aprendizado

Hierarquia do aprendizado de máquina: do aprendizado intuitivo, ramificando-se em aprendizado supervisionado (classificação e regressão) e não supervisionado (agrupamento, associação e sumarização).



Fonte: Adaptado de Faceli et al. (2011)

2.4 Algoritmos de classificação

Nesta monografia o foco está em métodos de aprendizado supervisionado. A classificação aborda a situação de prever a adesão a grupos dada uma instância de dados e representa uma função essencial no domínio do aprendizado de máquina e mineração de dados.

Tradicionalmente, técnicas de classificação operam estabelecendo limites de decisão no espaço de dados com base nos atributos presentes no conjunto de treinamento. Dessa forma, quando uma nova instância é submetida, sua classificação é determinada pela sua posição relativa a esses limites (CARNEIRO, 2017). Embora existam várias técnicas

disponíveis para aprendizado de máquina, a classificação é a técnica mais amplamente usada (SOOFI; AWAN, 2017).

O modelo requer uma base de dados previamente rotulada para realizar o processo de treinamento. Essa base, conhecida como conjunto de treino, é composta por exemplos já classificados, que servem como referência para que o modelo aprenda a identificar padrões e realizar futuras classificações de forma autônoma (SILVA, 2016).

No contexto da evasão no ensino superior, é comum a conversão das classes categóricas em valores numéricos para facilitar a adaptação aos classificadores. Uma abordagem frequentemente utilizada é a definição de duas classes principais, como "Evadiu" (0) e "Formou" (1), permitindo que algoritmos de aprendizado de máquina identifiquem padrões nos dados e realizem previsões para novos exemplos.

Os algoritmos de aprendizado de máquina supervisionado são amplamente empregados na classificação de dados educacionais. Modelos treinados a partir de bases de dados institucionais possibilitam a comparação de diferentes técnicas para avaliar sua eficácia na previsão da evasão. Os principais métodos utilizados nesse tipo de estudo são descritos a seguir.

2.4.1 Árvore de Decisão

O classificador Árvore de Decisão é um classificador simples, que utiliza a estratégia de dividir para conquistar para resolver o problema, os subproblemas são combinados, na forma de uma árvore, para reduzir um problema complexo. A árvore é basicamente um grafo acíclico direcionado, formado por nós de divisão e nós folhas. O nó de divisão contém um teste condicional baseado nos valores do atributo e os nós folhas são rotulados com as classes dos dados. O objetivo é criar um modelo que aprendendo regras de decisão simples inferidas a partir dos dados consiga prever o valor de uma variável teste.

O Índice de Gini e a Entropia são amplamente usados em algoritmos de árvore de decisão para selecionar o melhor recurso para dividir os dados em cada nó/nível da árvore de decisão. O parâmetro Índice de Gini como critério para medir o grau de impureza dos nós da árvore de decisão. A Figura 2 exemplifica a estrutura de uma árvore de decisão, que começa pelo nó raiz, contendo o conjunto de dados completos. À medida que a árvore é percorrida e os nós são divididos, a impureza dos subconjuntos diminui, pois os dados passam a ser melhor separados. A divisão dos nós ocorre com base em métricas de decisão, que calculam o grau de impureza de cada divisão utilizando o Índice de Gini. Esse processo continua iterativamente, refinando a separação dos dados até que os nós finais (folhas) sejam alcançados, formando um modelo adequado para classificar novos dados de teste (LEITE; MORAES; LOPES, 2020). A Equação 2.2 apresenta a fórmula do Índice de Gini.

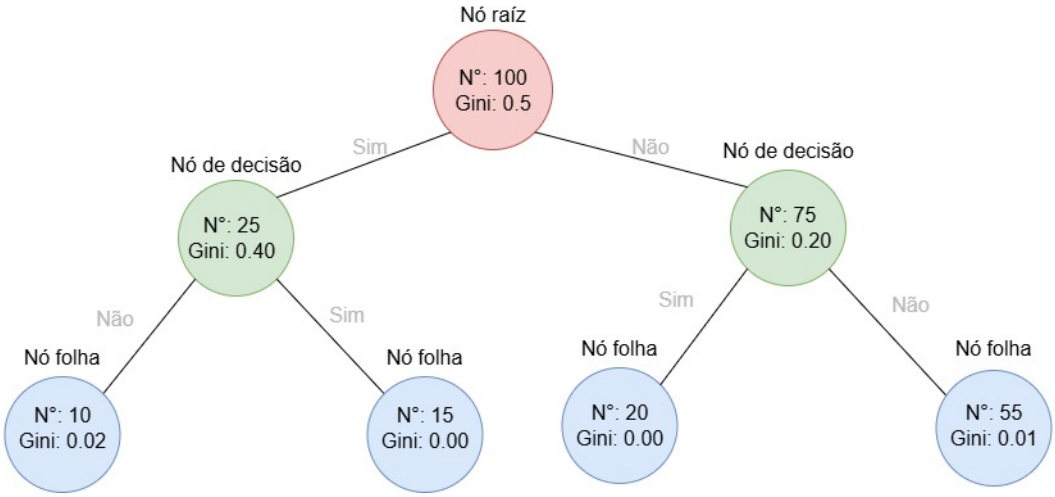
$$G = 1 - \sum_{i=1}^{n} p_i^2 \tag{2.2}$$

Em que:

- G: Índice de Gini;
- p_i: Proporção de elementos da classe i;
- n: Número total de classes.

Figura 2 – Árvore de Decisão

Árvore de decisão utilizando o índice de Gini. A cada divisão (ramo), é calculado o valor do índice, indicando o nível de pureza da separação. O objetivo é alcançar os nós mais puros, o que demonstra uma classificação eficaz, observado nos nós folha.



Fonte: Adaptado de Freitas (2022)

Já o parâmetro Entropia mede o grau de desordem, basicamente, é a mensuração da impureza ou aleatoriedade dos dados observados. Ela varia de 0 a 1, quanto mais próxima de 1 maior é a desordem dos dados, trazendo um menor ganho de informação para a predição de um dado futuro, se aproximar de 0 o nó está puro. (LEITE; MORAES; LOPES, 2020). O valor da entropia é calculado confore a Equação 2.3

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{2.3}$$

Em que:

- $H(S)$: Entropia do conjunto de dados S.
- n: Número total de classes distintas no conjunto de dados.

- p_i: Probabilidade de um exemplo pertencer à classe i, ou seja, a proporção de elementos da classe i no conjunto S.
- $\log_2 p_i$: Logaritmo base 2 da probabilidade p_i.

2.4.2 Floresta Aleatória

O classificador Floresta Aleatória (*Random Forest*) é um algoritmo de aprendizagem por agrupamento (*ensemble learning*), que combina múltiplas árvores de decisão para melhorar a precisão preditiva e reduzir o overfitting. Cada árvore do conjunto é treinada a partir da técnica conhecida como *Bootstrap Aggregation*. A partir dessa técnica, é criado um conjunto de treinos diferentes utilizando a amostragem *bootstrap* (diversas árvores de decisão treinadas em diferentes porções de amostra de um mesmo dataset), com cada conjunto de amostra de dados é feito um treinamento com árvores de decisão, por último, é combinado as predições individuais, por meio de medidas estatísticas (IBM, 2024b; LEITE; MORAES; LOPES, 2020).

A predição final, no caso de classificação, é determinada pelo voto majoritário entre as árvores do modelo. Além disso, a amostra oob serve como método de validação cruzada interna, dispensando a necessidade de um conjunto de testes separados (LEITE; MORAES; LOPES, 2020).

A Figura 3 ilustra as múltiplas divisões realizadas pelas árvores de decisão no algoritmo Random Forest, são iniciadas com um mesmo dataset e, a partir dele são realizadas a amostragem bootstrap, as amostragens passam por modelos de treinamento utilizando Árvores de Decisão. Ao final do processo, a decisão final é tomada com base nas previsões individuais das árvores, permitindo identificar os parâmetros que melhor contribuem para a construção de um modelo mais eficiente com o conjunto de dados.

O desempenho da Floresta Aleatória depende de três principais hiperparâmetros ajustáveis: tamanho de nós, número de árvores e quantidade de variáveis amostradas. Com esses ajustes, o algoritmo pode ser aplicado tanto para tarefas de regressão quanto de classificação (IBM, 2024b).

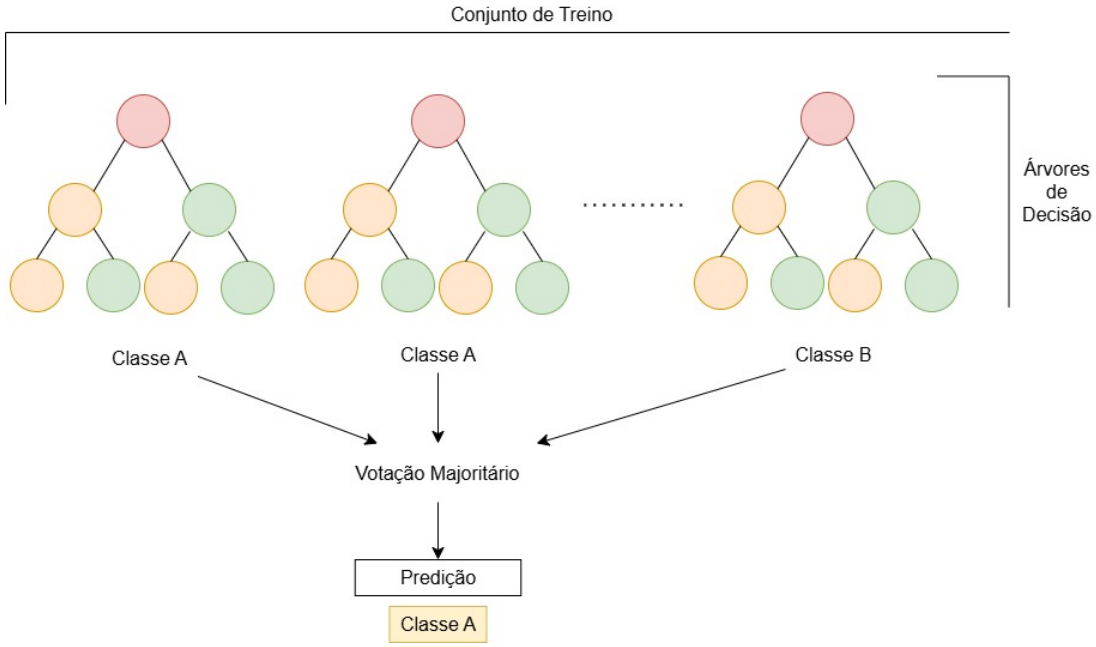
O parâmetro do Índice Gini, igualmente nas Árvores de Decisão, é utilizado para determinar qual das ramificações é mais provável de ocorrer, medindo o grau de heterogeneidade dos dados. Logo, pode ser utilizado para medir a impureza de um nó. Este índice num determinado nó é dado por pela fórmula 2.2:

2.4.3 Naive Bayes

Naive Bayes é um classificador probabilístico que é baseado no teorema de Bayes, ele assume que não há dependência entre variáveis dada classe e nem há ocorrências

Figura 3 – Floresta Aleatória

Esquema da Floresta Aleatória, em que diversas Árvores de Decisão são geradas a partir de diferentes subconjuntos de dados e parâmetros. A classificação final é determinada por votação majoritária entre as árvores, permitindo identificar a classe à qual o conjunto de entrada pertence.



Fonte: Do Autor

de atributos escondidos ou latentes (Scikit-Learn Developers, 2024). O termo *naive*, que significa ingênuo em inglês, se deve ao fato de que o Naive Bayes não parte de um "grau de certeza" inicial; nós começamos a análise dos dados sem qualquer certeza a priori, e construímos esse grau analisando as frequências presentes nos dados que temos disponíveis.

A fórmula básica do Teorema de Bayes, que expressa a probabilidade condicional de um evento A dado que B ocorreu, pode ser exemplificada a partir da Equação 2.4.

$$\Pr(A \mid B) = \frac{\Pr(B \mid A)\Pr(A)}{\Pr(B)} \tag{2.4}$$

Em que:

- $\Pr(A|B)$: Probabilidade de A acontecer, dado que B já aconteceu;
- $\Pr(B|A)$: Probabilidade de B acontecer, dado que A já aconteceu;
- $\Pr(A)$: Probabilidade de A acontecer sozinho (probabilidade a priori de A);
- $\Pr(B)$: Probabilidade de B acontecer sozinho (probabilidade total de B, considerando todos os casos possíveis.

A fórmula do classificador *Naive Bayes* se baseia no Teorema de Bayes, que calcula a probabilidade de uma classe y e dado um conjunto de atributos do vetor x_1 a x_n. No entanto, para simplificação, assume-se que os atributos possuem independência condicional entre si, ou seja, dado y, uma classe já conhecida, cada atributo xi é condicionalmente independente dos demais. Na equação 2.5 é apresentada uma versão simplificada da probabilidade condicional.

$$\Pr(y|x_1, \ldots, x_n) = \frac{\Pr(y) \prod_{i=1}^{n} \Pr(x_i|y)}{\Pr(x_1, \ldots, x_n)} \tag{2.5}$$

Em que:

- $\Pr(y|x_1, \ldots, x_n)$: Probabilidade de uma classe y dado um conjunto de características $x_1, \ldots, x_n$.;
- $\Pr(y)$: Probabilidade a priori da classe y;
- $\Pr(x_i|y)$: Probabilidade condicional de cada característica x_i, assumindo independência entre elas;
- $\Pr(x_1, \ldots, x_n)$: Probabilidade conjunta das características (constante para todas as classes).

Já a equação 2.6 destaca o critério de decisão final, no qual a classe escolhida a um novo atributo é aquela que maximiza essa probabilidade (Scikit-Learn Developers, 2024).

$$\hat{y} = \arg\max_{y} \Pr(y) \prod_{i=1}^{n} \Pr(x_i|y) \tag{2.6}$$

Em que:

- $\hat{y}$: Classe prevista, a mais provável para um novo exemplo;
- $\arg\max_y$: Das classes possíveis, é escolhida aquela com a maior probabilidade, o valor de y que maximiza a equação;
- $\Pr(y)$: Probabilidade a priori da classe, probabilidade de selecionar aleatoriamente um objeto dessa classe dentre todas as classes;
- $\Pr(x_i|y)$: Mede a chance de ver cada característica x_i se soubéssemos que a classe é y;
- $\prod_{i=1}^{n} \Pr(x_i|y)$: Multiplicação de todas as probabilidades individuais, ou seja, a chance de todas as características aparecerem juntas, dado que o exemplo pertence à classe y

O algoritmo *Naive Bayes* possui diferentes variações, cada uma adequada para tipos específicos de dados e problemas, como o Naive Bayes Multinomial, Bernoulli e Gaussiano, sendo esta última a utilizada neste trabalho.

O modelo Gaussiano no *Naive Bayes*, assume que os atributos contínuos sigam uma distribuição normal dentro de cada classe. Baseado na curva Gaussiana é possível calcular a probabilidade de um valor pertencer a determinada classe. A fórmula Gaussiana 2.7 é utilizada pelo modelo (Scikit-Learn Developers, 2024).

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{2.7}$$

Em que:

- x: Variável aleatória.
- μ: Média da distribuição (valor esperado).
- σ^2: Variância (dispersão dos dados).
- σ: Desvio padrão ($\sigma = \sqrt{\sigma^2}$).
- exp: Função exponencial e^x.

2.4.4 Máquina de Vetores de Suporte

Máquina de vetores de suporte (*Support Vector Machines*, SVM) é um conjunto de métodos de aprendizado supervisionado, utilizado por se destacar em espaços de alta dimensão e por ter boa capacidade de generalização (Scikit-Learn Developers, 2024). O algoritmo analisa os dados e reconhece padrões e constrói hiperplanos em um espaço multidimensional, nesse espaço os hiperplanos separam os dados da amostra em diferentes categorias. É possível que exista mais de um limiar de separação entre os dados, e com base na distância de um elemento de uma classe com de outra classe, o algoritmo decide a melhor forma de separar as classes.

O espaço utilizado neste trabalho é bidimensional, já que é utilizado para classificar as classes em duas categorias. Nele é dividido os dados por um hiperplano para que os pontos da mesma categoria se mantenham do mesmo lado, tem como objetivo maximizar a distância das amostradas de dados de ambas as categorias a este hiperplano, maximizando a margem entre as classes, garantindo que as instâncias de categorias diferentes fiquem o mais distante possível do hiperplano de separação.

O subconjunto de dados que é encontrado mais próximos do hiperplano de separação é conhecido como os vetores de suporte. Eles são considerados importantes pois definem a separação entre classes, se os pontos mudarem de posição, o plano também

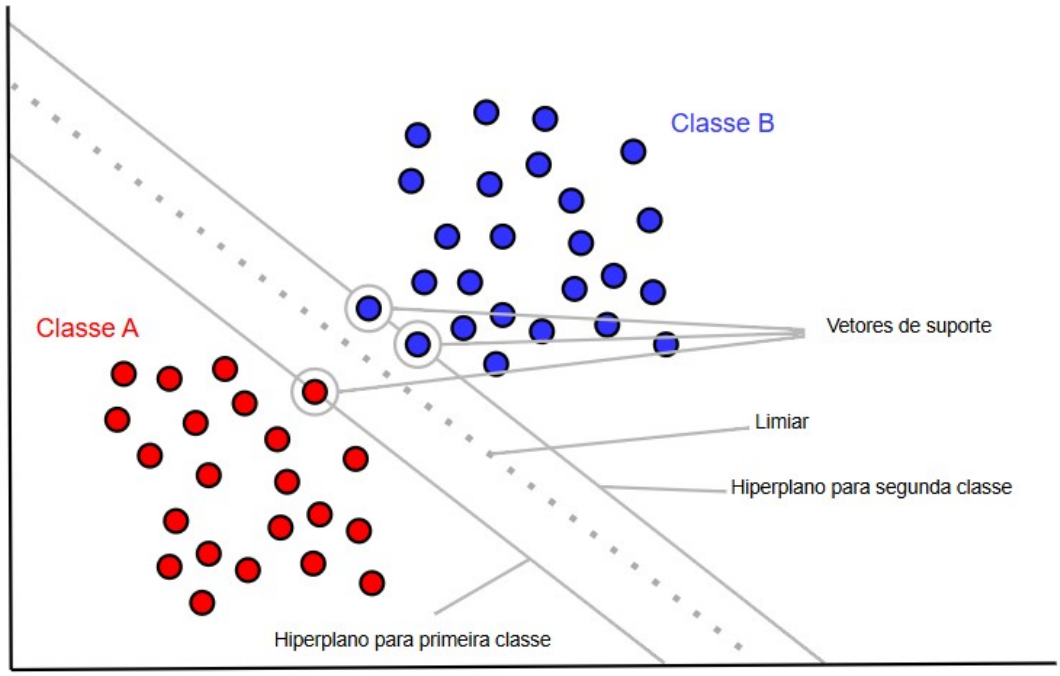
irá. Portanto, o objetivo do SVM é encontrar o hiperplano que maximize a margem, a distância entre o hiperplano e os vetores de suporte, para minimizar o risco de erro ao classificar os novos dados (LEITE; MORAES; LOPES, 2020).

Na Figura 4 é possível observar um exemplo genérico de hiperplano separando duas classes, Classe A (em vermelho) e Classe B (em azul). A posição entre os hiperplanos corresponde ao limiar de separação entre os hiperplanos das classes, a posição é definida durante a etapa de treinamento e pode ser calculada com base na metade da menor distância entre o elemento da Classe A e o da Classe B que mais se aproximam.

Depois do treinamento, os novos dados serão classificados com base no hiperplano de separação encontrado pelo modelo. Os pontos que estiverem de um lado do hiperplano serão atribuídos a uma classe, enquanto os do outro lado pertencerão à classe oposta. Assim, ao testar o modelo, cada novo dado será classificado de acordo com a sua posição em relação ao hiperplano, considerando a proximidade em relação aos vetores de suporte e à classe correspondente.

Figura 4 – Máquina de Vetores de Suporte (SVM) - Exemplo 1

Representação de um modelo de SVM, separando duas classes (A e B) por meio de um hiperplano ótimo. Os vetores de suporte são os pontos mais próximos do limiar de separação, responsáveis por definir a margem máxima entre os hiperplanos das classes.



Fonte: Adaptado de Faceli et al. (2011)

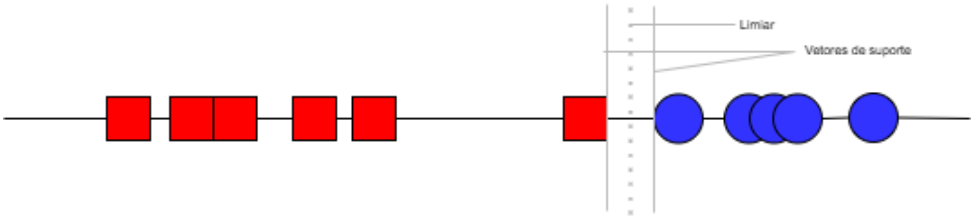
Outro cenário possível é observado na Figura 5. Nesse caso, ao aplicar a técnica do SVM, a menor distância entre os elementos de classes diferentes não reflete adequadamente a tendência da divisão entre elas. *Outliers* ou dados que se distanciam das características do seu grupo podem afetar a posição da fronteira de decisão e, consequentemente, compro-

meter a precisão do modelo. Após o treinamento, novas entradas podem ser classificadas erroneamente, como Classe A, quando na verdade pertencem à Classe B.

Para mitigar esse problema, uma solução seria ajustar o parâmetro de margem, permitindo uma maior flexibilidade na definição da fronteira de decisão, como ilustrado na Figura 6. Nesse caso, como a teoria das SVM foi originalmente desenvolvida para maximizar a margem de separação dos hiperplanos, para mitigar o problema descrito anteriormente, o SVM utiliza um parâmetro de regularização que equilibra a complexidade do modelo e a adaptação aos dados. Esse parâmetro permite que certos pontos discrepantes sejam ignorados, garantindo uma separação mais robusta e generalizável. Assim, o modelo consegue classificar novas amostras de forma mais confiável. (FERREIRA et al., 2010).

Figura 5 – Máquina de Vetores de Suporte (SVM) - Exemplo 2

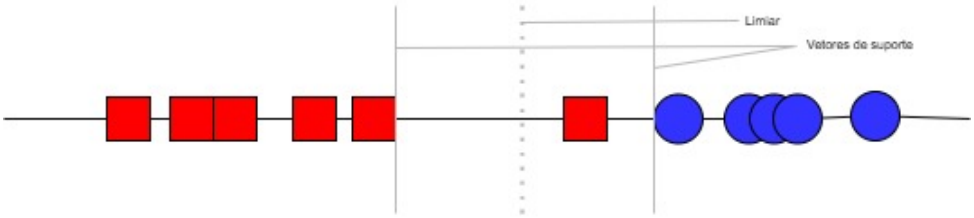
Representação de um modelo de SVM em que a separação entre as classes pode ocasionar erros de classificação futuros, devido à presença de instâncias com características mais distantes do núcleo do seu grupo.



Fonte: Adaptado de Faceli et al. (2011)

Figura 6 – Máquina de Vetores de Suporte (SVM) - Exemplo 3

Representação do mesmo conjunto de dados, com a aplicação de margens mais flexíveis nos hiperplanos, permitindo melhor separação entre as classes e reduzindo o risco de erro de classificação.



Fonte: Adaptado de Faceli et al. (2011)

2.4.5 Regressão Logística

A Regressão Logística é um método estatístico utilizado em cenários em que o objetivo é prever a probabilidade de ocorrência de um evento binário. Esse modelo permite estimar a relação entre diferentes variáveis independentes (x_1, x_2, x_3, ...) e uma variável

dependente (y), que assume apenas dois possíveis estados. O modelo analisa o conjunto de variáveis independentes e combina essas informações para estimar a probabilidade de um determinado resultado.

Para realizar a classificação, inicialmente calcula-se a soma ponderada dos atributos. Cada entrada possui um conjunto de atributos (por exemplo, x_1, x_2, x_3, ...) e cada uma dessas características é multiplicada por um peso correspondente (w_1, w_2, w_3, ...). Esses valores são somados com um termo de viés, representado por b, que é adicionado à equação para ajudar o modelo a se ajustar de maneira mais adequada aos dados. Como é necessário que o valor da soma ponderada (z) esteja no intervalo de 0 a 1, utiliza-se a função sigmóide para converter z em uma probabilidade (FERNANDES et al., 2021).

Se o resultado da função sigmóide for próximo de 1, a entrada é classificada em uma categoria, enquanto, se for mais próximo de 0, é classificada em outra. Para finalizar essa etapa, define-se um limiar, tradicionalmente no valor de 0,5. Nesse caso, se a probabilidade for maior que o limiar, a entrada é classificada em uma classe; caso contrário, é atribuída à outra classe. Após determinar a classe de saída, o erro é calculado usando uma função de perda, com base na probabilidade de cada dado de saída em relação aos rótulos reais. A função de perda (*log loss*) 2.8, é frequentemente utilizada em modelos de regressão logística para quantificar a diferença entre as previsões do modelo e os resultados reais. Essa função penaliza de maneira mais severa as previsões que estão longe do rótulo verdadeiro, aumentando assim a precisão do modelo. Por ser uma função contínua e diferenciável, a função de perda permite que as técnicas de otimização, como o gradiente descendente, funcionem de maneira eficaz (FERNANDES et al., 2021; IBM, 2024a).

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right) \tag{2.8}$$

Em que:

- n: Número total de amostras (ou exemplos) no conjunto de dados.
- y_i: Rótulo da i-ésima amostra no conjunto de dados
- p_i: Probabilidade prevista de que a i-ésima amostra pertença à classe positiva (ou classe 1).

De maneira iterativa, o modelo é ajustado várias vezes durante o processo de treinamento. Isso ocorre porque o modelo utiliza o método de descida do gradiente para minimizar o erro. A cada iteração, os pesos e o viés são atualizados. O viés, que inicialmente começa com valor zero, pode ser ajustado com valores aleatórios ao longo das iterações. A atualização do viés segue a fórmula apresentada na Equação 2.9, em que α

representa a taxa de aprendizado e a segunda parte da fórmula corresponde ao gradiente de erro em relação ao viés. O processo de iteração continua até que o modelo atinja um desempenho satisfatório (IBM, 2024a; SILVEIRA et al., 2021; JAMES et al., 2013; GELMAN; HILL, 2007).

$$b = b - \alpha \frac{\partial Loss}{\partial b} \tag{2.9}$$

Em que:

- b: Viés (bias), um parâmetro ajustável do modelo.
- α: Taxa de aprendizado (learning rate), que controla o tamanho dos ajustes no modelo.
- $\frac{\partial Loss}{\partial b}$: Gradiente da função de perda em relação ao viés, ou seja, a taxa de variação da perda em relação a b.

A combinação linear dos atributos no modelo é dada pela Equação 2.10, onde w_1, w_2, ..., w_n são os pesos associados aos atributos x_1, x_2, ..., x_n e b é o viés.

$$z = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b \tag{2.10}$$

Em que:

- $x_1, x_2, \ldots, x_n$: Atributos (variáveis de entrada) do modelo.
- $w_1, w_2, \ldots, w_n$: Pesos que o modelo aprende para cada atributo.
- b: Viés (bias), um valor adicional que ajuda a ajustar a saída do modelo.
- z: Resultado da combinação linear, que será usado na função de ativação.

O valor de z, resultante dessa combinação, será usado na função de ativação. O modelo então usa a Equação 2.11 para transformar a combinação linear z em uma probabilidade, $\Pr(Y = 1)$, que é a probabilidade da classe ser 1.

$$\Pr(Y = 1) = \frac{1}{1 + e^{-z}} \tag{2.11}$$

Em que:

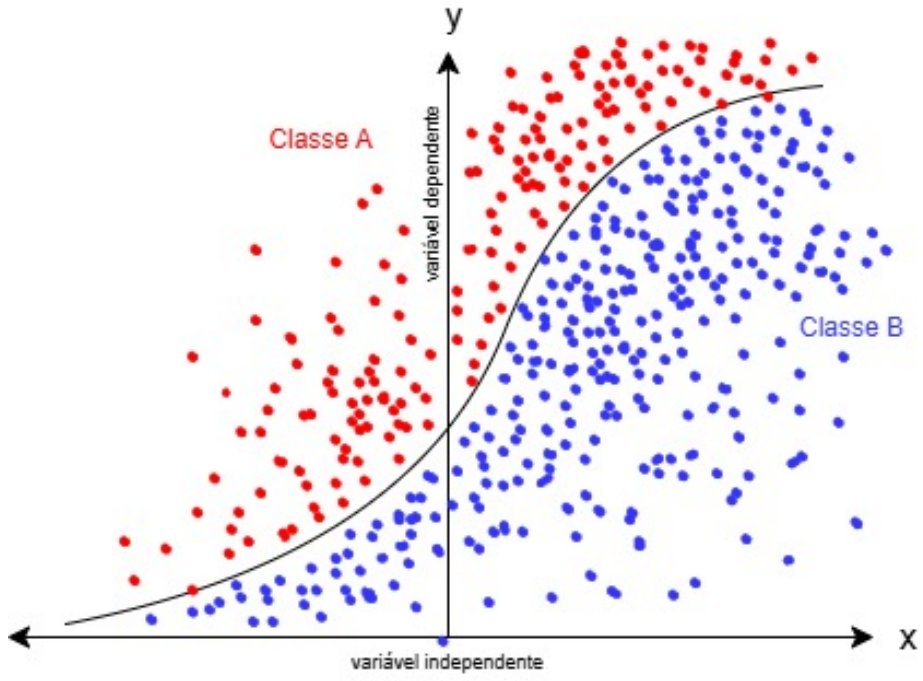
- $\Pr(Y = 1)$: Probabilidade de a classe ser 1 (classe positiva).
- z: Combinação linear dos atributos e pesos de z

- $1 + e^{-z}$: Mantém a saída sempre no intervalo $(0, 1)$, transformando z em uma probabilidade.

A Figura 7 ilustra um modelo de regressão logística. Os pontos representam exemplos do conjunto de dados, divididos em duas classes distintas: a classe 0 (representada em vermelho) e a classe 1 (representada em azul). A curva sigmoide (linha azul) modela a probabilidade de uma observação pertencer à classe 1, mapeando a saída do modelo para um intervalo entre 0 e 1. Essa curva reflete a relação entre a variável independente (x) e a variável dependente (y), permitindo a classificação dos dados com base em um limiar de decisão.

Figura 7 – Regressão Logística

Representação da Regressão Logística após o treinamento do modelo. A curva sigmoide é ajustada de forma a separar as instâncias das classes A e B com base na probabilidade de pertencimento a cada uma, utilizando um limiar (geralmente 0,5) como critério de decisão.



Fonte: Adaptado de Hair et al. (2019)

2.4.6 Regressão Linear

A regressão linear é um dos modelos mais simples e fundamentais no aprendizado de máquina. Ela é utilizada para prever um valor numérico com base em uma ou mais variáveis independentes, com o objetivo de encontrar a melhor reta que se ajusta aos dados.

Inicialmente, para a equação geral da regressão linear utilizando múltiplas variáveis independentes, os coeficientes (b_0, b_1, b_2, ...) são atribuídos com valores aleatórios, sendo b_i os coeficinetes que ajustam a influência de cada variável independente sobre y. O modelo aprende um hiperplano em um espaço multidimensional que busca minimizar a diferença entre os valores previstos e os valores reais da variável dependente, utilizando a Equação 2.12,

$$y = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_n x_n \tag{2.12}$$

Onde:

- y: Variável dependente (o valor a ser previsto);
- $x_1, x_2, \ldots, x_n$: Variáveis independentes (fatores que influenciam y);
- b_0: Valor de y quando todas as variáveis x_i são zero;
- $b_1, b_2, \ldots, b_n$: Coeficientes de regressão (pesos atribuídos a cada variável independente).

Uma vez calculada a previsão de y com a equação acima, é necessário ajustar os coeficientes, pois o modelo mede o erro entre a previsão y' e o valor real esperado y. Esse erro pode ser definido com a equação 2.13.

$$Erro = y - y' \tag{2.13}$$

O objetivo do modelo é minimizar esse erro, para que as previsões fiquem o mais próximas possíveis dos valores reais. Para isso, aplica-se uma função de perda, que quantifica o erro total do modelo. Uma das funções de perda mais utilizadas na regressão linear é o Erro Quadrático Médio (MSE), representado pela equação 2.14.

$$MSE = \frac{1}{m} \sum_{i=1}^{m} (y_{real}^{(i)} - \hat{y}^{(i)})^2 \tag{2.14}$$

Em que:

- m: Número total de exemplos no conjunto de dados,
- $y_{real}^{(i)}$: Valor real da i-ésima observação,
- $\hat{y}^{(i)}$: Valor previsto pelo modelo para a i-ésima observação.

Se o MSE for baixo, isso indica que as previsões do modelo estão próximas dos valores esperados. Já um MSE elevado sugere que o modelo está cometendo grandes erros de previsão.

Para minimizar o MSE, os coeficientes da regressão são ajustados por meio do gradiente descendente, um algoritmo de otimização que reduz a função de perda atualizando os coeficientes iterativamente. A atualização de cada coeficiente é feita com a Equação 2.15.

$$b_j = b_j - \alpha \frac{\partial MSE}{\partial b_j} \tag{2.15}$$

- b_j: Coeficiente a ser atualizado,
- α: Taxa de aprendizado, que controla o tamanho do passo na atualização,
- $\frac{\partial MSE}{\partial b_j}$: Derivada parcial do Erro Quadrático Médio (MSE) em relação a b_j.

Essas etapas são realizadas de forma iterativa até que a função de perda atinja um valor mínimo ou convirja para um ponto estável. Após encontrar o melhor conjunto de coeficientes, o modelo final é avaliado utilizando um conjunto de dados de teste para validar sua capacidade de generalização (SILVEIRA et al., 2021; BISHOP; NASRABADI, 2006; ALPAYDIN, 2020).

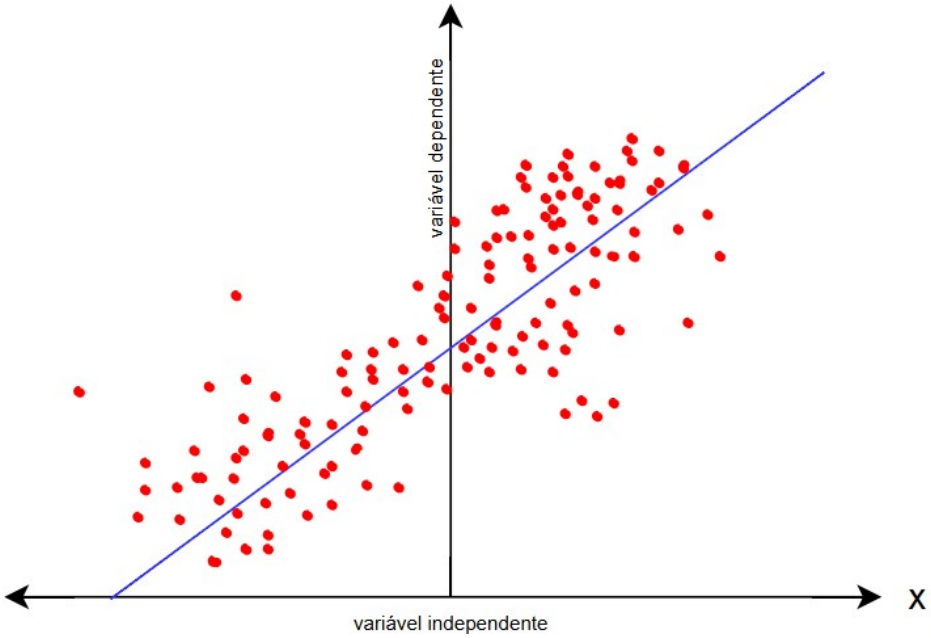
A Figura 8 ilustra um modelo de regressão linear simples. Os pontos vermelhos representam os exemplos do conjunto de dados, cada um contendo um par de valores (x, y). A linha azul corresponde à reta ajustada pelo modelo, que representa a relação linear entre a variável independente (x) e a variável dependente (y). O objetivo da regressão linear é encontrar a equação dessa reta, minimizando a diferença entre os valores reais dos dados e os valores previstos pelo modelo.

2.4.7 Perceptron Multicamadas

Uma rede neural artificial (RNA) do tipo passagem direta (*feed forward*) processa informações em uma única direção, da camada de entrada para a camada de saída, sem retroalimentação ou ciclos, passando por uma ou mais camadas ocultas. Isso significa que os dados seguem apenas para frente, sendo modificados em cada camada até atingir a saída final. O modelo é inspirado na rede de neurônios do cérebro humano, ele possui nós interconectados chamados neurônios artificiais, organizados em camadas. As informações iniciam na camada de entrada e fluem pela rede, cada neurônio processa as informações passadas e esse sinal produz um sinal de saída que influencia outros neurônios da rede. A rede é formada por nós (ou neurônios) que ficam conectados a todos os outros nós da camada subsequente (KOTSIOPOULOS et al., 2021). O *Multilayer Perceptron* (MLP),

Figura 8 – Regressão Linear

Representação gráfica da regressão linear simples, mostrando a relação entre uma variável independente (eixo x) e uma variável dependente (eixo y), com a linha de tendência estimada.



Fonte: Adaptado de Hair et al. (2019)

um modelo central em redes neurais artificiais, é composto por múltiplas camadas de neurônios organizadas de forma hierárquica. Cada neurônio nesta arquitetura recebe sinais de entrada, realiza cálculos utilizando uma função de ativação e produz um sinal de saída, que pode ser transmitido para os neurônios nas camadas subsequentes (GOODFELLOW et al., 2016).

A técnica de treinamento mais comumente utilizada para redes MLP é a retropropagação (*backpropagation*), que envolve várias etapas.

De início é feita o *feedforward*, em que as camadas de entrada são alimentadas com os dados de treinamento e cada neurônio da camada de entrada ativa o conjunto de neurônios da camada oculta. Durante essa fase as entradas são multiplicadas pelos pesos, valores pequenos e aleatórios por não ter conhecimento prévio dos parâmetros ideais.

Em seguida uma função de ativação é aplicada, que resulta em saídas que são passadas para a próxima camada. Esse processo é repetido pelas camadas ocultas até que a informação chegue à camada de saída, onde é obtida as previsões do modelo. Após a passagem direta, por uma função de perda, é calculado o erro entre as previsões da rede e os valores esperados. É por meio desse cálculo que o desempenho do modelo é validado (GOODFELLOW et al., 2016; SOUZA et al., 2004; ALVES; LOTUFO; LOPES, 2013).

Na fase do uso do algoritmo de retropropagação, o erro que foi calculado é pro-

pagado de volta pela rede, iniciando pela camada de saída e movendo-se em direção a camada de entrada. Esse processo envolve o cálculo dos gradientes do erro em relação aos pesos de cada neurônio, que depende da função de ativação, para que, utilizando sua derivação, possa atualizar os pesos. As funções de ativação desempenham um papel crítico no modelo, já que, ao introduzir a não linearidade, ela possibilita que a rede aprenda e modele padrões complexos que não poderiam ser identificados através de modelos lineares simples. Por exemplo, se fosse usar só operações lineares ela se comportaria de modo semelhante ao modelo de regressão linear simples, de modo que impediria de aprender padrões complexos. Funções como ReLU (*Rectified Linear Unit*) ou sigmóide, são exemplos de operações não lineares que a rede cria para se adaptar aos dados, criando limites de decisão curvos corretamente.

O algoritmo determina como o erro deve ser ajustado para cada peso da rede, com o objetivo de minimizar a função de perda e minimizar o erro. O erro do neurônio de saída afeta os neurônios da camada anterior, e assim por diante, até que todos os pesos da rede tenham suas contribuições para o erro atualizados (GOODFELLOW et al., 2016; SOUZA et al., 2004; ALVES; LOTUFO; LOPES, 2013).

O ajuste dos pesos é realizado a partir dos gradientes calculados, de acordo com uma taxa de aprendizado definida pelo usuário. Essa taxa permite que a rede convirja para uma solução ideal de minimização de erros. O processo de retropropagação e ajustes dos pesos é repetido por múltiplas iterações em que a rede é exposta repetidamente aos dados de treinamento, para permitir que ela aprenda as complexas relações dos dados. A iteração é feita até que um critério de parada seja alcançado. Esse critério pode ser um número pré-definido de iterações ou até que a função de perda esteja abaixo de um limiar, que indica que a rede aprendeu a mapear as entradas de modo desejado.

Por fim, como esse método iterativo, a MLP deve ser capaz de aprender da melhor forma possível e generalizar para novos dados. Após o treinamento, a rede pode ser utilizada para fazer previsões sobre os dados de teste, aplicando o mesmo processo de feedforward mas sem o reajuste de pesos, pois já teria encontrado a configuração ideal da rede (GAMBALLI et al., 2022; SOUZA et al., 2004; ALVES; LOTUFO; LOPES, 2013; SOARES; SILVA, 2011).

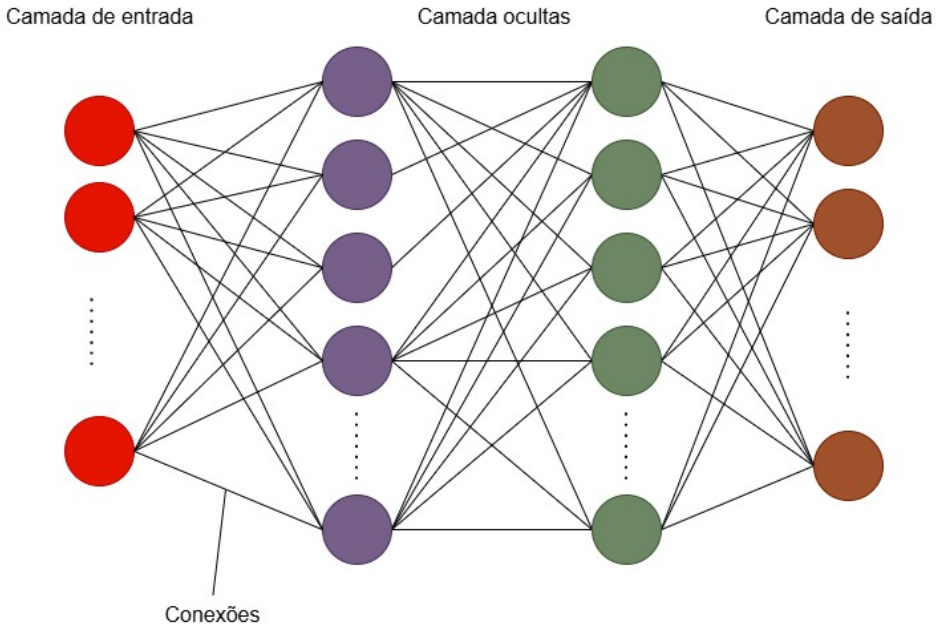
A Figura 9 exemplifica uma MLP com duas camadas ocultas, além de uma camada de entrada e uma de saída. Na camada de entrada, cada neurônio recebe um atributo dos dados, que são posteriormente propagados para as camadas ocultas e a camada de saída.

2.5 Medidas de Desempenho

A avaliação de modelos de classificação em aprendizado de máquina é uma etapa crucial para garantir que os algoritmos não apenas aprendam a partir dos dados, mas

Figura 9 – Multilayer Perceptron

O modelo MLP realiza sucessivas atualizações nos seus parâmetros durante o processo de treinamento. A cada iteração, o erro é calculado com base na saída gerada, e o algoritmo de backpropagation é aplicado para ajustar os pesos das conexões, propagando o erro de volta pelas camadas. Esse processo continua até que o modelo atinja um desempenho satisfatório, estando então pronto para fazer previsões com novos dados.



Fonte: Adaptado de Faceli et al. (2011)

também ofereçam resultados precisos e interpretáveis. Entre os métodos mais comuns utilizados para essa avaliação, destacam-se a acurácia, a curva ROC (Característica de Operação do Receptor), a área sob a curva (AUC), precisão, sensibilidade e a métrica F1.

A acurácia é uma métrica simples que representa a proporção de previsões corretas feitas pelo modelo em relação ao total de previsões. Contudo, a acurácia pode ser enganosa, especialmente em conjuntos de dados desbalanceados, onde a classe de interesse pode ser significativamente menor que as outras.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.16}$$

- TP (True Positives): Quantidade de verdadeiros positivos (acertos na classe positiva);
- TN (True Negatives): Quantidade de verdadeiros negativos (acertos na classe negativa);
- FP (False Positives): Quantidade de falsos positivos (erros ao prever a classe positiva);

- FN (False Negatives): Quantidade de falsos negativos (erros ao prever a classe negativa);

A curva ROC é um método mais robusto que permite avaliar o desempenho do modelo em diferentes limiares de classificação, traçando a taxa de verdadeiros positivos em função da taxa de falsos positivos. A AUC (Área sob a Curva ROC) mede a capacidade do modelo em distinguir entre classes positivas e negativas. O valor da AUC varia de 0 a 1 e representa a área sob a curva ROC.

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \tag{2.17}$$

$$AUC = \int_0^1 TPR(t)\, dFPR(t) \tag{2.18}$$

- Taxa de Verdadeiros Positivos (TPR - True Positive Rate): Também chamada de Sensibilidade, mede a proporção de exemplos positivos corretamente classificados;
- Taxa de Falsos Positivos (FPR - False Positive Rate): Mede a proporção de exemplos negativos incorretamente classificados como positivos;
- FP (False Positives): Quantidade de falsos positivos (erros ao prever a classe positiva);
- A Curva ROC faz um gráfico em que TPR representa o eixo y e FRP o eixo x, variando o limiar de decisão;

As métricas de precisão e sensibilidade também desempenham um papel fundamental na avaliação do desempenho de classificadores. A precisão, indicada pela equação 2.19, indica a proporção de verdadeiros positivos em relação ao total de positivos previstos, enquanto a sensibilidade, equação 2.20, mede a proporção de verdadeiros positivos em relação ao total de reais positivos

$$Precisão = \frac{TP}{TP + FP} \tag{2.19}$$

$$Sensibilidade = \frac{TP}{TP + FN} \tag{2.20}$$

A métrica F1 combina ambas para fornecer uma medida única que considera tanto a precisão quanto a sensibilidade, sendo particularmente útil em cenários onde existe uma necessidade de equilíbrio entre as duas (FACELI et al., 2011).

$$F1 = 2 \times \frac{Precisão \times Sensibilidade}{Precisão + Sensibilidade} \tag{2.21}$$

Assim, a escolha e a combinação dessas métricas de avaliação são fundamentais para a construção de modelos de classificação robustos.

2.6 Revisão Bibliográfica

Esta seção tem como objetivo apresentar os principais estudos relacionados ao tema deste trabalho, ou seja, a previsão de evasão escolar de ensino superior.

O trabalho de Melo (2016) aborda o problema da detecção de evasão associado ao aprendizado de máquina supervisionado. Ele tem como objetivo selecionar um conjunto de procedimentos de *Machine Learning* relevantes para o problema, avaliar os algoritmos empiricamente sobre a base de dados, com o objetivo de encontrar o modelo de classificação mais eficiente e, por fim, analisar as informações obtidas, e extrair fatores que são relevantes para identificação da motivação dos estudantes ao evadirem. O autor utiliza o classificador Floresta Aleatória e conclui que os atributos mais importantes para prever a evasão são aqueles que já estão inseridos nos registros acadêmicos, mas dependem do curso e período em que o aluno está. Além, como resultado, foi constatado que o *F-measure* das árvores atingem até 88% de acerto, para alunos dos primeiros períodos, porém a pesquisa se torna aleatória e os classificadores obtém resultados piores do que esperado, à medida que os períodos vão se passando. A pesquisa do Melo (2016) se relaciona a esta monografia pois considera que a predição de quais estudantes são mais propícios a abandonar o curso possa ser feita por meio de aprendizado supervisionado, mais precisamente, por meio de classificação, apesar de este ser relacionado a todos os cursos de uma universidade e o trabalho atual ser envolto no ambiente da Faculdade de Computação, especificamente.

Semelhante ao trabalho anterior, o trabalho de Teodoro e Kappel (2020) tem como objetivo investigar e identificar padrões nas características de estudantes de graduação, como atributos sobre o estudante e sobre a universidade em que ele se encontra, a partir de técnicas de Aprendizado de Máquina, de modo a prevenir o risco de abandono de curso, identificando os discentes com maior probabilidade de evadir precocemente. Dessa forma, os autores aplicaram cinco técnicas de aprendizado de máquina nos dados retirados da base de dados públicos do INEP, e dentre elas foi escolhida a técnica que apresentou os melhores resultados. A técnica escolhida pelo autor foi Floresta Aleatória, que alcançou uma taxa de acertos de aproximadamente 80% das previsões de evasão. O estudo revela que os parâmetros que se destacaram na hora de prever sua tendência de abandonar o ensino público superior, são a idade, a participação em atividades extracurriculares e a carga horária total do curso. De modo similar, neste trabalho irá ser feito uma coleta de dados, que vão ser filtrados, selecionados, balanceados e pré-processados, de forma que,

posteriormente seja possível utilizar classificadores que foram treinados e testados, para encontrar as variáveis mais importantes para a previsão.

Já o trabalho de Viana, Santana e Rabêlo (2022) se apresenta mais semelhante a esta monografia, isso se dá por além de utilizar estratégias de Mineração de Dados, a proposta é realizada com dados dos cursos de Computação e Sistemas da Informação da Universidade Federal do Piaui. Assim, a igualdade está no uso do aprendizado supervisionado, utilizando avaliação de classificadores para predição de evasão no ensino superior. Para sua realização, foi utilizado o processo de descoberta de conhecimento em banco de dados, conhecido como *Knowledge Discovery in Databases* (KDD), nele é feito a coleta, pré-processamento, transformação, mineração e avaliação dos resultados com as devidas interpretações. O autor utilizou para realizar a decisão o algoritmo Floresta Aleatória, que obteve resultados promissores com acurácia entre 85% e 96%.

Em sua monografia, Bianchi (2017) realizou um estudo com dados de diversos cursos da Universidade Federal de Santa Maria (UFSM), com o objetivo de criar um modelo capaz de prever a evasão escolar a partir de dados relacionados à aprovação dos alunos. Para isso, os dados foram transformados por meio de um script, de forma que cada aluno fosse representado por um único registro, contendo um atributo para cada disciplina passível de ser cursada. Entre os atributos utilizados, estavam o "ano de conclusão" e o "período de conclusão" da disciplina, além do "ano ideal" e do "período ideal" para a sua realização. Com essas informações, o autor desenvolveu um script em Python para calcular um coeficiente que representa o adiantamento ou o atraso do discente em relação ao percurso ideal do curso. O número total de atributos no registro de cada aluno corresponde ao número de disciplinas possíveis de serem cursadas, preenchidas com os respectivos coeficientes. O atributo de classe foi representado de forma binária: "Evadiu" ou "Não Evadiu". Para a etapa de classificação, foi utilizada a ferramenta WEKA, na qual foi aplicado o modelo Logistic Model Tree (LMT), que combina características de árvores de decisão e regressão logística. Os resultados obtidos demonstraram alto desempenho, com acurácia superior a 90% para os anos de 2015 e 2016, indicando uma elevada eficiência na previsão da evasão escolar. A pesquisa de Bianchi (2017) se relaciona com o presente trabalho por também utilizar dados acadêmicos históricos e técnicas de aprendizado de máquina para prever a evasão no ensino superior, embora com abordagens distintas de modelagem e representação dos dados dos discentes.

Por fim, o trablho de Manhães e Cruz (2020) apresenta um estudo sobre predição do desempenho acadêmico de alunos de graduação por meio de mineração de dados. O objetivo é fornecer aos gestores educacionais uma abordagem para acompanhar o desempenho acadêmico dos graduandos e prever aqueles com risco de evasão. Os autores analisaram dados de alunos de diversos cursos da UFRJ ao longo de 16 anos, utilizando apenas informações acadêmicas. Foi realizada uma comparação entre 12 algoritmos clas-

sificadores para identificar aqueles com maior precisão na previsão de evasão. As variáveis analisadas incluíram notas, período das disciplinas, Coeficiente de Rendimento (CR) e Coeficiente de Rendimento Acumulado (CRA), entre outras. Antes da aplicação dos modelos de aprendizado supervisionado, os dados passaram por um processo de ETL. Algoritmos como *Naive Bayes*, Árvore de Decisão e SVM foram testados e avaliados com base em métricas como acurácia, validação cruzada, matriz de confusão e coeficiente Kappa. O estudo propôs uma solução modular baseada em mineração de dados para auxiliar gestores acadêmicos na identificação de alunos em risco de evasão.

Ambos os trabalhos se relacionam diretamente com esta pesquisa, pois também utilizam dados acadêmicos históricos e aprendizado supervisionado com o objetivo de prever a evasão escolar. Embora o presente estudo se diferencie ao focar no desempenho dos três primeiros semestres por meio do cálculo da Média Geral Acadêmica (MGA) e na análise do impacto de técnicas de balanceamento nos modelos preditivos.

3 Desenvolvimento

Este capítulo descreve as etapas desenvolvidas para a construção de um modelo preditivo de evasão escolar no ensino superior, com base em técnicas de aprendizado de máquina. O objetivo é identificar, com boa precisão, os alunos com maior probabilidade de evasão, considerando os dados dos três primeiros semestres do curso.

Utilizando as informações acadêmicas dos discentes do curso de BCC, fornecidas pela coordenação, foram realizadas análises exploratórias, ajustes de parâmetros, além de procedimentos de balanceamento e normalização dos dados. Por fim, são apresentadas as técnicas de classificação aplicadas à tarefa de predição da evasão.

3.1 Análise da Base de dados

Para a coleta de dados utilizada na pesquisa, foi necessário que a coordenação da Faculdade de Ciência da Computação (FACOM-UFU) disponibilizasse as informações acadêmicas dos discentes do BCC. A base de dados utilizada foi previamente anonimizada, garantindo a privacidade dos estudantes, e contém informações essenciais que foram processadas e incorporadas ao modelo, tais como o período em que o aluno cursou cada disciplina, a respectiva carga horária, a nota final obtida e a classificação do estudante quanto à sua trajetória acadêmica (se evadiu ou se formou). Além desses dados, outras informações passaram por um processo de pré-processamento antes da análise.

Os dados disponibilizados estavam organizados em formato tabular, em que cada linha representava uma disciplina cursada por um aluno. Dessa forma, um mesmo discente possuía múltiplas linhas na base, correspondentes às disciplinas cursadas ao longo de sua trajetória acadêmica, até sua saída da instituição – seja por evasão, conclusão do curso ou até a data de disponibilização dos dados.

A planilha continha as seguintes colunas:

- ID: Identificador fictício para o discente
- Ano Ingresso: Ano de ingresso no curso
- Período Ingresso: Período de ingresso no curso
- Ano Disciplina: Ano em que foi cursada a disciplina x_i pelo discente
- Período Disciplina: Período em que foi cursada a disciplina x_i pelo discente
- Média Final: Nota que o discente ficou no final da disciplina x_i

- Carga Horária: Carga horária total da disciplina
- Situação: De que forma o discinte terminou a disciplina x_i, se foi por ter sido aprovado ou reprovado
- Forma Evasão: Situação que o discinte se encontra no fim da faculdade, por exemplo, se evadiu, formou ou outro
- Código Disciplina: Código numérico da disciplina
- Código Curso: Código numérico do curso
- Ano-Mês Disciplina: Período no formato ano-mês que o discente cursou a disciplina x_i
- Nome Disciplina: Nome da disciplina
- Descrição Disciplina: Descrição do tipo de disciplina, por exemplo, se é obrigatória, optativa ou outro
- Curso: Nome do curso dentro da Facom, por exemplo, Ciência da Computação, Sistemas da Informação ou outro

O processamento e a análise dos dados foram realizados utilizando a linguagem de programação Python, com ênfase no uso das bibliotecas pandas para a manipulação dos datasets e *scikit-learn* para as etapas de pré-processamento, treinamento e classificação dos dados. Com essas ferramentas e com as informações extraídas da base, foi possível iniciar o tratamento e a modelagem dos dados para os experimentos propostos.

3.2 Pré-processamento

Apesar de algoritmos de AM serem frequentemente adotados para extrair conhecimento de conjuntos de dados, seu desempenho é geralmente afetado pelo estado desses dados. Os valores dos atributos podem estar limpos ou conter ruídos e imperfeições, incluindo valores incorretos, inconsistentes, duplicados ou ausentes. Por isso, técnicas de pré-processamento de dados são amplamente utilizadas para melhorar a qualidade das informações, eliminando ou minimizando esses problemas. Além disso, o pré-processamento pode tornar os dados mais adequados para sua utilização por determinados algoritmos, facilitando o uso de técnicas de AM, levando à construção de modelos mais fiéis à distribuição real dos dados e reduzindo sua complexidade computacional.

A base de dados inicial continha informações de 1.421 alunos distintos, cada um com um status acadêmico específico. A seguir, serão apresentadas algumas das operações de pré-processamento que foram aplicadas antes da utilização dos dados por algoritmos de AM.

3.2.1 Eliminação manual de atributos

Nem todos os atributos do conjunto de dados original são necessários para a previsão de evasão escolar para os propósitos deste trabalho. Atributos que não contribuem significativamente para a estimativa do valor do atributo-alvo são considerados irrelevantes e, portanto, podem ser removidos. Exemplos de atributos que não foram considerados atributos de interesse incluem Nome da Disciplina, Código da Disciplina e Código do Curso, entre outros.

A Tabela 1 apresenta um resumo dos dados utilizados nesta monografia, após a remoção dos atributos irrelevantes.

Tabela 1 – Exemplo de dados brutos utilizados como entrada após primeira etapa de pré-processamento.

ID	Ano Ingresso	Período Ingresso	Ano Disciplina	Período Disciplina	Média Final	Carga Horária	Situação	Forma Evasão
1	2010	1	2010	1	60	75	Aprovado	Jubilamento
1	2010	1	2010	1	45	60	Repr.Freq.	Jubilamento
2	2012	2	2012	2	90	90	Aprovado	Desistente Oficial
3	2020	2	2020	1	50	30	Reprovado	Formado
3	2020	2	2020	2	75	60	Aprovado	Formado
3	2020	2	2022	1	80	90	Aprovado	Formado

Fonte: Do Autor

3.2.2 Limpeza dos Dados

Conjuntos de dados frequentemente requerem um processo de limpeza, mesmo técnicas de classificação robustas podem ser afetadas pela qualidade dos dados, o que pode comprometer a precisão das análises. Para mitigar esse problema, algumas etapas de limpeza foram aplicadas antes do processamento.

Inicialmente, todas as linhas em que a nota final do aluno na disciplina, representada pelo campo MEDIA FINAL na Tabela 1, estava ausente foram removidas. A nota é essencial para o cálculo da Média Geral Acadêmica (MGA), e, como sua ausência indica que o aluno não realizou a disciplina, essas entradas foram desconsideradas.

Outro critério adotado foi a remoção de registros com a situação de disciplina dada como “S/Aproveitamento”. Essa situação indica que a disciplina foi cursada durante um Período Atípico, no qual, em caso de reprovação, a nota final não impactava o desempenho acadêmico do aluno. Assim, optou-se por remover esses registros, uma vez que são equivalentes a não ter cursado a disciplina.

Além disso, inconsistências no preenchimento dos dados foram corrigidas. Um exemplo foi a coluna PERÍODO, que indica se o aluno ingressou no curso no primeiro ou no segundo semestre do ano. Os valores esperados para esse campo são 1 (ingresso no início do ano) e 2 (ingresso no meio do ano). No entanto, algumas entradas continham o valor incorreto “ANO”, o que levou à exclusão dessas linhas.

Também foram removidos os registros de alunos que ainda estavam matriculados na universidade, pois não pertenciam a nenhuma das duas classes consideradas no estudo (evasão ou conclusão). Por fim, todas as linhas com valores ausentes em atributos essenciais para a análise foram eliminadas da base de dados.

3.2.3 Transformação de Dados

Alguns dados da base original não estavam formatados da maneira mais adequada para serem utilizados nos modelos. Por isso, algumas transformações foram necessárias. A primeira etapa envolveu a conversão de valores do tipo string para valores numéricos. No caso do Período Atípico, os dados na coluna PERÍODO eram representados como "1° Per. Esp." e "2° Per. Esp.", indicando o primeiro e o segundo período especial, respectivamente. Como não há distinção matemática entre esses períodos e os períodos regulares, esses valores foram convertidos para 1 e 2, respectivamente.

Outra transformação essencial foi a binarização das formas de evasão. Inicialmente, a base de dados continha diversas classificações para alunos que não concluíram o curso, como "Desligamento Convênio", "Jubilamento", "Desistente Oficial", "Abandono", "Desistente" e "Desligamento". Embora essas categorias tenham motivos distintos, todas indicam que o aluno deixou a universidade antes de concluir o curso. Assim, para este trabalho, todos esses casos foram agrupados na classe única "Evadiu". Por outro lado, alunos que concluíram o curso foram identificados apenas pela categoria "Formado".

3.2.4 Criação de Dataset

A Tabela 2 apresenta uma amostra representativa dos dados após o processo de pré-processamento. As colunas MGA1, MGA2 e MGA3 correspondem às médias gerais acadêmicas dos alunos nos três primeiros semestres do curso, sendo consideradas atributos preditivos essenciais para o modelo. A coluna Forma Evasão representa a variável alvo do problema de classificação, indica a situação final do aluno e assume valores binários, "Evadiu" ou "Formado". Esse conjunto de dados será utilizado nas etapas de balanceamento e construção dos modelos preditivos.

Tabela 2 – Exemplo de dados prontos para modelagem após o pré-processamento.

ID	MGA1	MG2	MG3	Forma Evasão
1	45.50	60.00	20.00	Evadiu
2	55.50	45.00	30.50	Evadiu
3	40.00	50.00	50.00	Evadiu
4	60.00	70.00	50.00	Formado
5	80.00	90.00	85.00	Formado

Fonte: Do Autor

3.2.5 Balanceameno das classes

O desbalanceamento de classes é um problema comum na área de classificação de dados. Em muitos conjuntos de dados reais, algumas classes aparecem com maior frequência do que outras, o que pode comprometer o desempenho dos algoritmos de Aprendizado de Máquina (AM). Quando treinados com dados desbalanceados, esses algoritmos tendem a favorecer a classe majoritária, reduzindo a precisão na predição da classe minoritária (FACELI et al., 2011).

A Figura 10 apresenta a distribuição das classes após o pré-processamento. A classe "Formado" é a majoritária, com 708 alunos, enquanto a classe "Evadiu" é a minoritária, com 443 alunos.

É importante destacar que com o *dataset* devidamente preparado, os dados foram divididos em conjuntos de treinamento e teste utilizando o método *train_test_split* da biblioteca *scikit-learn*. A separação foi feita de 80% dos dados para treinamento do modelo e 20% para o conjunto de teste. Essa divisão permite que o modelo seja treinado com a maior parte dos dados disponíveis e depois testado em um subconjunto não visto anteriormente, de forma que permite uma avaliação mais fiel de seu desempenho.

O processo de balanceamento deve ser aplicado apenas ao conjunto de treinamento. Isso se dá pelo fato de que o conjunto de teste deve representar uma situação próxima do cenário real, com a distribuição original das classes. Alterar a proporção no conjunto de teste pode levar a resultados artificialmente inflados ou irreais, o que compromete a análise o desempenho do modelo. Além de criar a possibilidade de um viés na avaliação, que pode levar o modelo ao *overfitting*, em que os dados se ajustam de tal forma aos dados de treinamento que perde a capacidade de generalizar para novos dados. Por isso, o balanceamento é realizado somente nos dados de treino, pois assim garante uma avaliação

mais confiável da capacidade do modelo de lidar com a desproporção entre classes de dados reais.

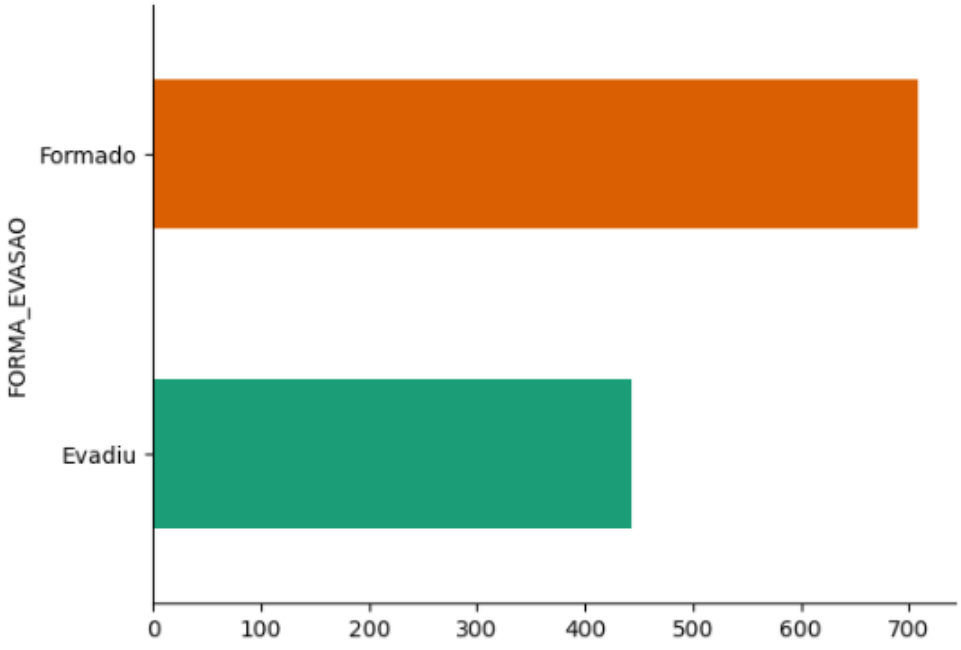
Para lidar com esse desbalanceamento, existem técnicas como *Oversampling* e *Undersampling*. O *Oversampling* aumenta a quantidade de dados da classe minoritária, gerando exemplos artificiais até que haja um equilíbrio entre as classes. Já o *Undersampling* reduz a classe majoritária, eliminando amostras ou selecionando um subconjunto representativo dos dados.

Neste trabalho, optou-se pelo *Oversampling*, pois a base de dados já possui um número relativamente pequeno de amostras. Assim, ampliar a base artificialmente foi considerado mais vantajoso do que reduzir ainda mais a quantidade de dados disponíveis.

É importante destacar que essas técnicas foram aplicadas apenas à base de treinamento, garantindo que a base de teste permanecesse com sua distribuição original. Isso permite uma avaliação realista do modelo, evitando vieses que poderiam gerar resultados excessivamente otimistas.

Figura 10 – Distribuição das classes

A distribuição de classes inclui 708 alunos na classe “Formado” e 443 alunos na classe “Evadiu”.



Fonte: Do Autor

3.2.5.1 Técnica Random Oversample (RO)

O *Random Oversampler* (RO) é um método de reamostragem utilizado para lidar com o desbalanceamento de classes em conjuntos de dados. A técnica consiste em aumentar a quantidade de instâncias da classe minoritária por meio da duplicação aleatória

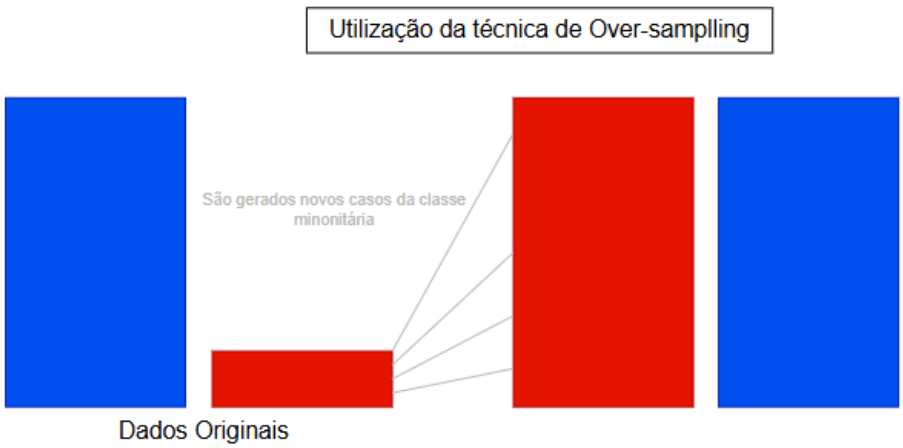
de amostras existentes, garantindo que ambas as classes tenham um número equivalente de observações. Dessa forma, o modelo de Aprendizado de Máquina pode aprender de maneira mais equilibrada, reduzindo a tendência de favorecer a classe majoritária.

O RO é uma abordagem simples e eficiente, sendo especialmente útil quando há um número limitado de amostras da classe minoritária. No entanto, como ele apenas replica instâncias já existentes, pode aumentar o risco de overfitting, uma vez que os dados adicionados não trazem novas informações ao modelo.

A Figura 11 ilustra visualmente o efeito do *Random Oversampling*, demonstrando como a distribuição das classes é balanceada após a aplicação da técnica.

Figura 11 – Random Oversampler

A técnica de Random Oversampling (RO) duplica aleatoriamente instâncias da classe minoritária com o objetivo de equilibrar a quantidade de instâncias entre as classes.



Fonte: Do Autor

3.2.5.2 Técnica Synthetic Minority Oversampling Technique (SMOTE)

A técnica SMOTE (*Synthetic Minority Oversampling Technique*) é amplamente utilizada para lidar com o desbalanceamento de classes em conjuntos de dados. Diferente do *Random Oversampling*, que apenas replica instâncias da classe minoritária, o SMOTE gera novos exemplos sintéticos por meio da interpolação de dados existentes, evitando a duplicação exata e reduzindo o risco de overfitting.

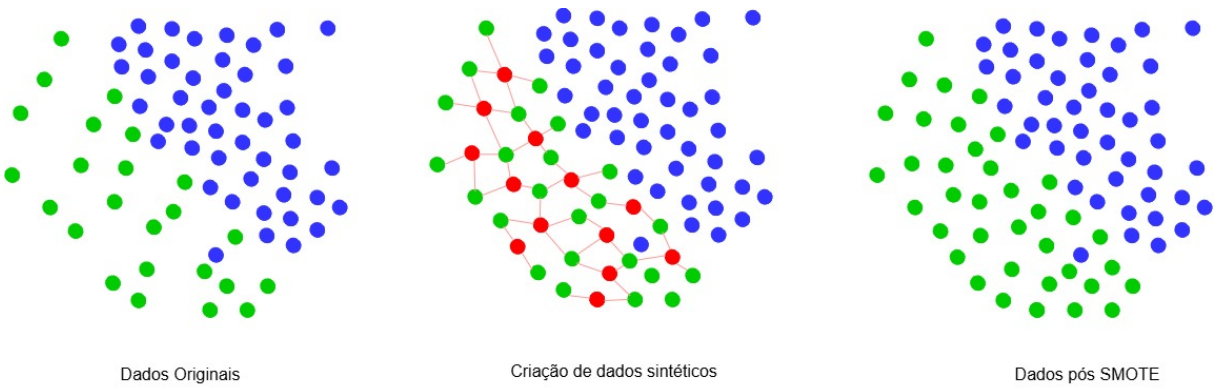
O algoritmo funciona se distingue por manter a linearidade entre as variáveis, utilizando o algoritmo KNN (*k-nearest neighbors*), identificando os k vizinhos mais próximos de cada instância da classe minoritária com base na distância euclidiana. Em seguida, novas amostras são criadas ao longo do segmento que conecta a instância original a um de seus vizinhos escolhidos aleatoriamente. Esse processo mantém a estrutura dos dados, criando exemplos mais representativos e preservando a relação entre as variáveis.

Embora eficaz, o SMOTE também apresenta desafios. Como os novos dados são gerados sem considerar diretamente a classe majoritária, pode ocorrer a criação de amostras artificiais em regiões de sobreposição, aumentando a chance de introduzir ruído e outliers (LEE; KIM; KIM, 2017).

A Figura 12 ilustra o fluxo de funcionamento do algoritmo. Inicialmente, as classes são analisadas para identificar a minoritária. Em seguida, são calculadas as distâncias entre os elementos dessa classe (representadas pelas linhas vermelhas claras), e os novos pontos sintéticos (vermelho escuro) são gerados. No final, a classe minoritária atinge um número de instâncias equivalente à classe majoritária.

Figura 12 – Funcionamento do SMOTE

A técnica SMOTE gera novas instâncias sintéticas da classe minoritária com base nas suas vizinhas mais próximas, até que ambas as classes possuam quantidades equilibradas de exemplos.



Fonte: Do Autor

3.2.5.3 Parâmetro Class Weigth

Neste método, pesos diferentes são atribuídos às classes do conjunto de dados com base no número de amostras em cada uma. A classe minoritária recebe um peso maior, aumentando sua influência no treinamento, enquanto a classe majoritária é penalizada com um peso menor. Essa abordagem busca mitigar o impacto do desbalanceamento, reduzindo a tendência do modelo de favorecer a classe com mais instâncias.

A atribuição dos pesos afeta diretamente o desempenho da classificação durante o treinamento. O objetivo é minimizar erros na previsão da classe minoritária ao aumentar sua representatividade e, ao mesmo tempo, reduzir o viés do modelo em relação à classe majoritária.

A Equação 3.1 foi utilizada para calcular os pesos das classes, garantindo que as classes menos representadas tenham maior influência no ajuste do modelo (BAKIRARAR; ELHAN, 2023).

$$w_i = \frac{n_{samples}}{n_{classes} \times bincount(y_i)} \tag{3.1}$$

Em que:

- w_i - Peso atribuído à classe i.
- $n_{samples}$ - Número total de amostras no conjunto de dados.
- $n_{classes}$ - Número total de classes.
- $bincount(y_i)$ - Número de ocorrências da classe i no vetor de rótulos y.

O resultado final é um vetor no formato $[x, y]$, onde x corresponde ao peso atribuído à classe A e y ao peso da classe B.

3.2.5.4 Nada foi feito

Para fins comparativos, nenhuma técnica de balanceamento foi aplicada em um dos testes realizados. O conjunto de dados foi mantido em sua proporção original, com 61,2% das amostras pertencendo à classe "Formados" e 38,8% à classe "Evasores".

3.2.6 Normalização

Após a aplicação das técnicas de balanceamento, ou mesmo para o conjunto de dados sem modificações, foi realizada a normalização dos dados.

Neste trabalho, foi utilizado um pacote do Python que aplica a normalização baseada na distribuição estatística dos dados. Esse tipo de normalização é particularmente útil quando os dados seguem uma distribuição normal (gaussiana), pois padroniza os valores para terem média igual a 0 e desvio padrão igual a 1 (FACELI et al., 2011).

O uso da normalização foi utilizada pois alguns algoritmos de machine learning são sensíveis à escala dos dados, como Regressão Logistica, Redes Neurais, SVM, entre outros. Como usam distâncias ou cálculos baseados em gradientes em seus modelos, quando os dados não estão normalizados, atributos com valores maiores podem ter mais influência do que deveriam. Além que modelos que utilizam gradiente descendente, como redes neurais e regressão logística, convergem mais rápido quando os dados estão padronizados, o que é uma reação desejada.

3.3 Modelos de Classificação dos Dados

Após o balanceamento das classes no conjunto de treinamento, foi realizado o treinamento dos modelos de classificação. Foram testados Floresta Aleatória, Árvore de

Decisão, SVM, Regressão Logística, MLP, *Naive Bayes* e Regressão Linear, considerando quatro abordagens:

- Um conjunto ajustado por *Random Oversampling*
- Um conjunto gerado com SMOTE, criando amostras sintéticas da classe minoritária
- O conjunto original, aplicando ponderação de classes (WC) diretamente no modelo (Esta abordagem não foi aplicável para MLP e *Naive Bayes*)
- O conjunto original sem qualquer técnica de balanceamento, utilizado como referência

O algoritmo Floresta Aleatória é baseado na construção de múltiplas árvores de decisão. Para cada árvore criada, foram utilizados os mesmos conjuntos de hiperparâmetros: critério Gini para divisão dos nós, profundidade máxima de 3 níveis. O classificador Árvore de Decisão foi configurado com os mesmos hiperparâmetros que o algoritmo do Floresta Aleatória.

O classificador SVM foi configurado com um kernel linear, buscando separar as classes por meio de um hiperplano otimizado.

A rede neural MLP foi configurada com uma única camada oculta de 8 neurônios e limite de 1000 iterações, visando garantir a convergência. Como redes neurais são altamente sensíveis à escala dos dados, a normalização foi essencial para melhorar o desempenho do modelo.

O Naive Bayes foi utilizado na versão Gaussiana, que assume uma distribuição normal dos atributos. Diferentemente de outros modelos, esse classificador não permite a aplicação de ponderação de classes.

Da mesma forma, o MLP também apresenta limitações quanto à ponderação de classes. Por esse motivo, ambos foram avaliados apenas com RO, SMOTE e sem balanceamento.

Os modelos de Regressão Logística e Regressão Linear foram testados com os hiperparâmetros padrão da biblioteca.

Em todos os casos, foi utilizado o valor fixo *random_state* igual a 42 para garantir a reprodutibilidade dos resultados. Os parâmetros utilizados podem ser observados na Tabela 3

Para todos os classificadores, após o treinamento, foram feitas previsões nos dados de teste, permitindo uma análise comparativa do impacto de cada abordagem na classificação.

Com os modelos treinados, as previsões foram geradas sobre os dados de teste, e os resultados analisados com métricas como acurácia, precisão, sensibilidade, F1-score e ROC-AUC. A partir desses indicadores, foi possível avaliar o impacto das técnicas de balanceamento na performance dos classificadores e identificar as estratégias mais adequadas para o problema estudado.

Tabela 3 – Parâmetros por classificador

Classificador	**Parâmetros**
Floresta Aleatória	Índice de Gini Profundidade máxima de 3 níveis Random state = 42
Árvore de Decisão	Índice de Gini Profundidade máxima de 3 níveis Random state = 42
SVM	Kernel linear Random state = 42
MLP	Uma camada oculta de 8 neurônios Limite de 1000 iterações Random state = 42
Naive Bayes	Versão Gaussiana Random state = 42
Regressão Logística	Hiperparâmetros padrões da biblioteca Random state = 42
Regressão Linear	Hiperparâmetros padrões da biblioteca Random state = 42

Fonte: Do Autor

4 Resultados

Após a aplicação das técnicas de balanceamento e o treinamento dos modelos de classificação, foram calculadas as métricas de avaliação para cada um dos classificadores utilizados. As métricas analisadas incluem acurácia, precisão, sensibilidade, F1-score, suporte e ROC-AUC, permitindo uma avaliação abrangente do desempenho de cada modelo.

Com esses resultados em mãos, foi realizada uma análise comparativa entre os classificadores, considerando o impacto das diferentes abordagens de balanceamento utilizadas. O objetivo dessa análise é identificar qual combinação de técnica de balanceamento e modelo de aprendizado de máquina obteve o melhor desempenho na predição dos dados, levando em conta não apenas a acurácia geral, mas também a capacidade do modelo de lidar com a classe minoritária. Essa comparação será apresentada por meio de tabelas e gráficos, destacando as diferenças no desempenho dos modelos e fornecendo uma visão clara de quais abordagens foram mais eficazes para o problema estudado.

A Figura 13 visa ilustrar e comparar o desempenho de múltiplos classificadores em termos de acurácia, avaliados sob quatro técnicas de balanceamento distintas de dados: *Oversampling*, SMOTE, *Weight Class*, e NDA (Nenhum Dado Adicional). Vale destacar que os valores de Precisão e Sensibilidade são calculados com base na *Weighted Average*, que corresponde à média ponderada considerando a proporção de amostras em cada classe.

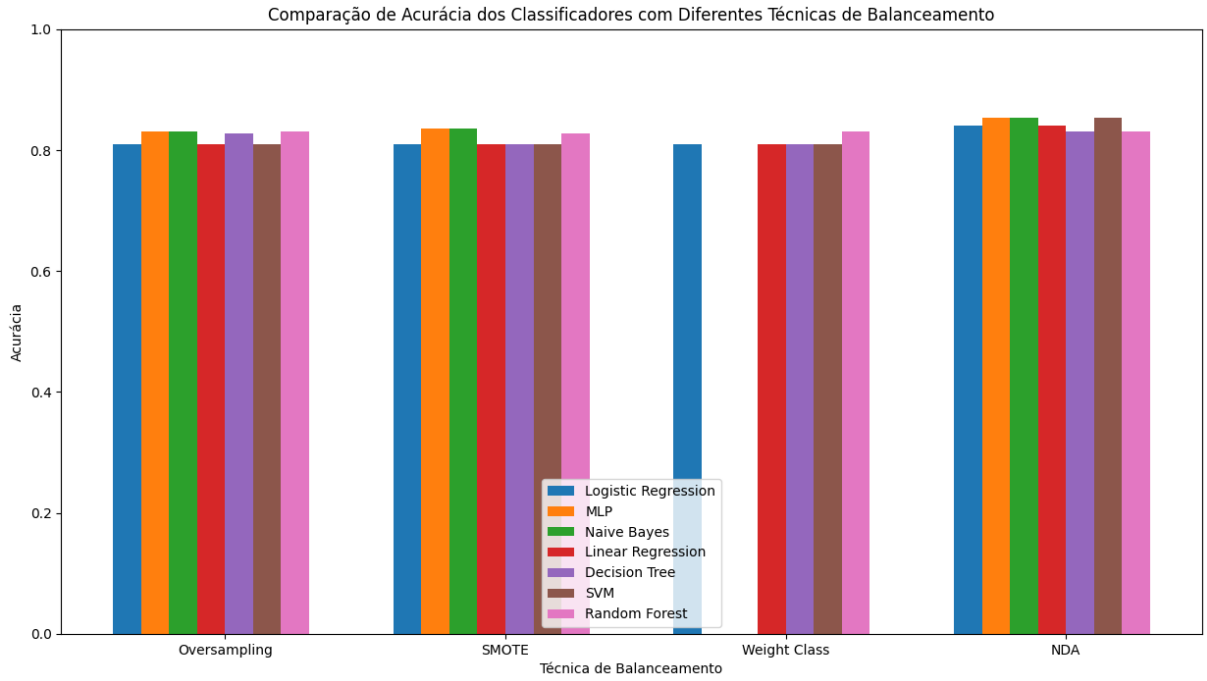
A comparação é feita utilizando todos os classificadores analisados neste trabalho (Floresta Aleatória, Árvore de Decisão, Regressão Logística, Regressão Linear, MLP, SVM e Naive Bayes), o gráfico gerado permite uma visualização clara e intuitiva das diferenças no desempenho de cada classificador em relação às diferentes abordagens de balanceamento. No caso dos classificadores MLP e Naive Bayes, a técnica de balanceamento *Weight Class* não é aplicável, esse valor é ignorado ao gerar o gráfico, evitando a exibição de uma barra para essa técnica.

A Tabela 4 apresenta um comparativo do desempenho geral dos classificadores utilizados no estudo. As métricas avaliadas foram acurácia, precisão, sensibilidade, F1-score e AUC-ROC, com o objetivo de comparar o desempenho dos algoritmos em diferentes cenários de balanceamento de dados. De modo geral, o classificador MLP combinado com a técnica NDA destacou-se por apresentar os melhores resultados de forma consistente em praticamente todas as métricas analisadas. Além disso, observou-se que a técnica de balanceamento NDA contribuiu para o bom desempenho de outros modelos, como SVM e Regressão Linear, enquanto o uso do SMOTE proporcionou os melhores resultados de AUC-ROC para diversos classificadores.

Mais especificamente, foi observado que a Floresta Aleatória apresentou resulta-

Figura 13 – Comparativo entre classes

Representação de todos os classificadores utilizados, organizados de acordo com as técnicas de balanceamento aplicadas.



Fonte: Do Autor

dos bastante consistentes, com valores de acurácia, precisão e sensibilidade variando entre 0,827 e 0,831, e AUC-ROC praticamente constante entre 0,898 e 0,902 em todas as abordagens. A técnica SMOTE obteve uma ligeira vantagem nas métricas principais (acurácia de 0,827 e F1-score de 0,827), embora a diferença em relação às demais configurações tenha sido mínima. Isso demonstra que esse classificador é estável e pouco sensível às técnicas de balanceamento aplicadas.

Já o SVM, conforme apresentado na Tabela 4, demonstrou um comportamento mais sensível às diferentes abordagens. Nas três técnicas de balanceamento (RO, SMOTE e WC), os resultados de acurácia, precisão e sensibilidade permaneceram quase constantes com valores próximos a 0,810, 0,817 e 0,810, respectivamente. Contudo, ao utilizar o conjunto de dados original (NDA), essas métricas aumentaram para 0,853 (acurácia), 0,853 (precisão) e 0,853 (sensibilidade), indicando que o SVM teve um desempenho superior sem técnicas de balanceamento, ao menos neste conjunto de dados. A AUC-ROC permaneceu com valores satisfatórios, maiores de 0,900 para todos os métodos.

O MLP também apresentou bons resultados, com desempenho constante utilizando RO (acurácia, precisão e sensibilidade de 0,831) e SMOTE (acurácia e sensibilidade de 0,836 e precisão de 0,835), e desempenho superior ao utilizar os dados originais (NDA), com acurácia, precisão e sensibilidade de 0,835 e F1-score de 0,851. A técnica de Weight

Class (WC) não foi aplicada ou não teve seus dados disponibilizados para este classificador, impossibilitando comparações nesse ponto. Assim como nos demais classificadores, a AUC-ROC manteve-se entre 0,900 e 0,902, refletindo uma boa capacidade discriminativa em todos os cenários.

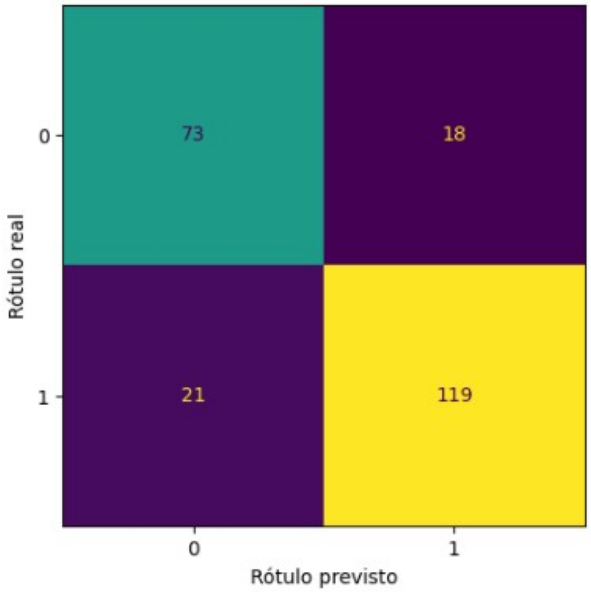
Tabela 4 – Tabela Comparativa de Classificadores

		RO	**SMOTE**	**WC**	**NDA**
Floresta Aleatória	Acurácia	0,836	0,827	0,831	0,831
	Precisão	0,837	0,828	0,832	0,831
	Sensibilidade	0,836	0,827	0,831	0,831
	F1-Score	0,836	0,827	0,832	0,829
	AUC-ROC	0,898	0,902	0,900	0,901
Árvore de Decisão	Acurácia	0,827	0,810	0,810	0,831
	Precisão	0,826	0,814	0,815	0,830
	Sensibilidade	0,827	0,810	0,810	0,831
	F1-Score	0,826	0,811	0,811	0,830
	AUC-ROC	0,901	0,902	0,900	0,901
Regressão Logística	Acurácia	0,810	0,810	0,810	0,810
	Precisão	0,819	0,819	0,819	0,819
	Sensibilidade	0,810	0,810	0,810	0,810
	F1-Score	0,811	0,811	0,811	0,811
	AUC-ROC	0,901	0,902	0,900	0,901
Regressão Linear	Acurácia	0,810	0,810	0,810	0,840
	Precisão	0,819	0,819	0,819	0,839
	Sensibilidade	0,810	0,810	0,810	0,840
	F1-Score	0,811	0,811	0,811	0,838
	AUC-ROC	0,901	0,902	0,900	0,900
MLP	Acurácia	0,831	0,836	-	0,853
	Precisão	0,831	0,835	-	0,853
	Sensibilidade	0,831	0,836	-	0,853
	F1-Score	0,831	0,835	-	0,851
	AUC-ROC	0,901	0,902	-	0,900
SVM	Acurácia	0,810	0,810	0,810	0,853
	Precisão	0,817	0,817	0,819	0,853
	Sensibilidade	0,810	0,810	0,810	0,853
	F1-Score	0,811	0,811	0,811	0,851
	AUC-ROC	0,901	0,902	0,900	0,900
Naive Bayes	Acurácia	0,836	0,836	-	0,831
	Precisão	0,836	0,836	-	0,830
	F1-Score	0,836	0,836	-	0,831
	Sensibilidade	0,836	0,836	-	0,831
	AUC-ROC	0,901	0,902	-	0,900

Fonte: Do Autor

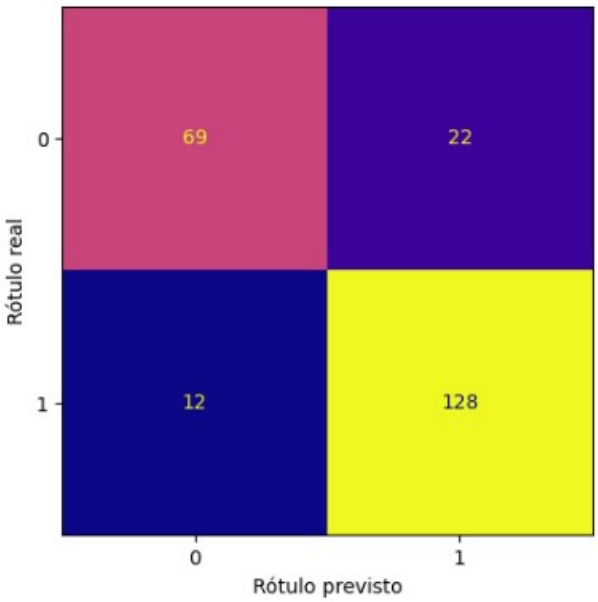
As Figuras 14, 15 e 16 apresentam a matriz de confusão dos classificadores que obtiveram os melhores desempenhos gerais, considerando acurácia, precisão e sensibilidade.

Figura 14 – Matriz de confusão – Floresta Aleatória



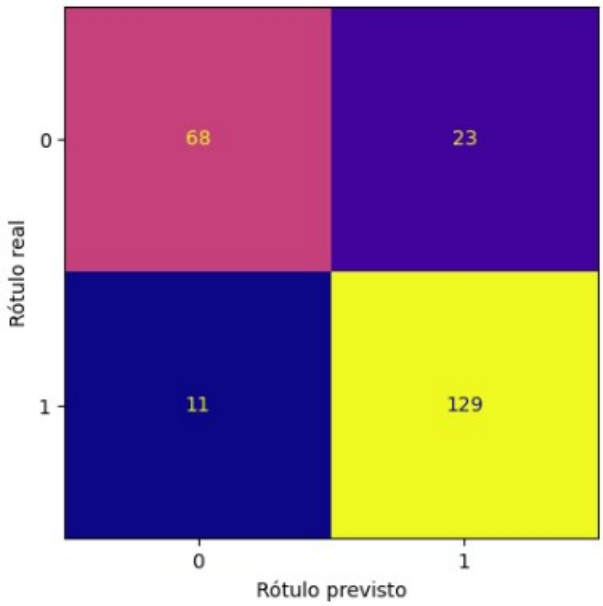
Fonte: Do Autor

Figura 15 – Matriz de confusão – SVM



Fonte: Do Autor

Figura 16 – Matriz de confusão – MLP



Fonte: Do Autor

Técnicas como RO e NDA (Nenhum Dado Adicional) mostraram eficazes para a maioria dos classificadores, contribuindo para uma melhora dos níveis de acurácia. É observado, também, que o classificador Floresta Aleatória foi o que apresentou menor variação nos resultados, independentemente da técnica de balanceamento adotada, o que pode indicar uma menor sensibilidade desse modelo ao desbalanceamento das classes. De maneira geral, todos os classificadores apresentaram bons valores de acurácia, com destaque para o MLP que se teve seu desempenho em destaque na maiori das técnicas, masntendo a consiencia em todos as formas de balanceamento.

De modo geral, os três classificadores apresentaram bons desempenhos em todos os cenários. A Floresta Aleatória e Naive Bayes foram os modelos mais estáveis, com métricas praticamente constantes, independentemente da técnica de balanceamento utilizada. O MLP demonstrou melhor desempenho com os dados originais (NDA), o que pode indicar que o desbalanceamento das classes não comprometeu significativamente os resultados. A AUC-ROC se manteve constante com valores acima de 0,89 para todos os classificadores e abordagens, indicando que os modelos são igualmente eficazes na separação das classes. Essa análise evidencia que, para o conjunto de dados avaliado, o uso de técnicas de balanceamento não trouxe ganhos significativos de desempenho e, em alguns casos, o modelo obteve melhores resultados sem aplicá-las.

Dessa forma, conclui-se que a escolha da técnica de balanceamento pode impactar o desempenho dos classificadores de forma distinta, sendo que modelos como o MLP demonstraram robustez ao lidar com diferentes estratégias de balanceamento e, inclusive, com a ausência delas. O objetivo da monografia foi alcançado, uma vez que foi possí-

vel desenvolver modelos suficientemente robustos para classificar, com alto desempenho, utilizando apenas os dados dos primeiros semestres do curso. Com uma precisão média de aproximadamente 85%, os modelos permitem que a coordenação atue de forma rápida e eficaz já no primeiro ano e meio de graduação. Essa capacidade de previsão precoce reforça a relevância dos semestres iniciais, como indicadores de possíveis dificuldades enfrentadas pelos discentes. Quando os desafios são predominantemente acadêmicos, os dados oferecem auxílio para intervenções da coordenações dos cursos de modo mais assertivas, visando a melhoria no desempenho e permanência dos estudantes. As análises e experimentos realizados nesta monografia foram implementados em Python, utilizando bibliotecas como Scikit-learn e Pandas. O código completo pode ser acessado no seguinte link: <git@github.com:IsaTedeschi/previsao-estudantil.git>.

5 Conclusão

Este trabalho teve como objetivo auxiliar na gestão universitária por meio da análise do risco de evasão dos alunos, com base em seu desempenho acadêmico. Uma das formas mais utilizadas de identificação de estudantes com maior probabilidade de evadir depende da experiência e dedicação dos gestores. Considerando o grande número de alunos matriculados, essa tarefa torna-se desafiadora. Além disso, as desistências geram vagas ociosas, o que dificulta a organização das disciplinas e o planejamento para os novos ingressantes. Assim, o modelo proposto neste trabalho busca não apenas reduzir desperdícios financeiros, mas também permitir que o aluno em risco receba o suporte necessário antes de desistir do curso, contribuindo para sua formação como profissional.

Foi apresentado o processo de análise e refinamento dos dados do discentes do BCC fornecidos pela coordenação do curso da FACOM-UFU dos discentes do BCC, bem como os principais experimentos realizados para alcançar os melhores modelos preditivos. Foram comparados sete classificadores supervisionados, avaliados com base em métricas como acurácia, precisão, sensibilidade, F1-score e AUC-ROC. Os melhores modelos demonstraram boa capacidade de identificar estudantes com risco de evasão nos cursos de graduação.

Observou-se que houve pouca variação entre os resultados dos classificadores, mesmo com a aplicação de diferentes técnicas de balanceamento. Isso indica que os modelos avaliados são estáveis e pouco sensíveis às técnicas utilizadas, uma vez que, para dois dos três melhores classificadores, os resultados foram melhores com os dados originais. De maneira geral, todos os modelos apresentaram bom desempenho. A Floresta Aleatória e Naive Bayes foram alguns dos classificadores mais consistentes, enquanto o Perceptron Multicamadas (MLP) destacou-se ao obterem seus melhores resultados sem balanceamento. Outros classificadores também tiveram desempenho relevante, mas inferior ou equivalente aos destacados, por isso não foram analisados individualmente. A métrica AUC-ROC permaneceu constante em 0,90 na maioria dos casos, evidenciando a boa capacidade discriminativa dos modelos.

Embora as técnicas de balanceamento não tenham melhorado significativamente os resultados, o objetivo principal foi alcançado: utilizar técnicas de balanceamento e de classificação e desenvolver um modelo eficaz e de bom desempenho para auxiliar na detecção de estudantes com maior risco de evasão, utilizando os primeiros três primeiros semestres cursados. Tal modelo traz benefícios para todos os envolvidos: os alunos recebem intervenções mais rápidas e personalizadas; a instituição pode direcionar recursos para os grupos mais vulneráveis (como bolsas, monitorias e projetos de apoio); e a sociedade se

beneficia com maior eficiência no uso de recursos públicos, menor desperdício de vagas e aumento na qualificação profissional da população.

Uma das limitações enfrentadas neste trabalho foi o tamanho da base de dados. Quanto maior o volume de dados disponíveis, melhor o desempenho dos modelos, permitindo análises mais profundas e resultados potencialmente mais significativos. Dessa forma, uma base maior poderia ter proporcionado melhor distinção entre classificadores e técnicas de balanceamento, além de maior entendimento do comportamento dos dados.

As melhorias promovidas tanto na vida do discente quanto na gestão da universidade e na sociedade como um todo reforçam a importância de estudos nesta área. Como sugestões para trabalhos futuros, propõe-se considerar também aspectos sociais, econômicos e de gênero, além dos dados acadêmicos, buscando uma compreensão mais abrangente dos fatores que contribuem para a evasão e o perfil dos grupos mais afetados.

Referências

ALPAYDIN, E. **Introduction to machine learning**. 4. ed. Cambridge, USA: MIT press, 2020. Citado na página 29.

ALVES, M.; LOTUFO, A.; LOPES, M. Seleção de variáveis stepwise aplicadas em redes neurais artificiais para previsão de demanda de cargas elétricas. In: BERNARDES, M.; FREIRE, I. L.; ZANARDI, M. C. (Ed.). **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**. São Carlos, SP: SBMAC, 2013. Citado 2 vezes nas páginas 30 e 31.

BAKIRARAR, B.; ELHAN, A. H. Class weighting technique to deal with imbalanced class problem in machine learning: methodological research. **Turkiye Klinikleri Journal of Biostatistics**, v. 15, n. 1, p. 19–29, 2023. Citado na página 44.

BANOULA, M. **Supervised and Unsupervised Learning in Machine Learning**. 2023. Disponível em: <https://www.simplilearn.com/tutorials/machine-learning-tutorial/supervised-and-unsupervised-learning#what_is_supervised_learning>. Acesso em: 29 out 2023. Citado na página 16.

BIANCHI, J. **Previsão de evasão em cursos de ensino superior através de aprendizado de máquina associado à análise de disciplinas aprovadas**. 50 p. Monografia (Trabalho de Conclusão de Curso) — Universidade Federal de Santa Maria, Santa Maria, RS, 2017. Citado na página 35.

BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. New York, NY: Springer, 2006. Citado na página 29.

BORDAS, M. C. Diplomação, retenção e evasão nos cursos de graduação em instituições de ensino superior públicas: resumo do relatório apresentado à ADIFES, ABRUEM e SESU/MEC pela comissão especial. **Avaliação: Revista da Rede de Avaliação Institucional da Educação Superior**, v. 1, n. 2, p. 55–65, 1996. Disponível em: <https://www.lume.ufrgs.br/handle/10183/225423>. Acesso em: 6 abr 2025. Citado na página 9.

CARNEIRO, M. G. **Redes complexas para classificação de dados via conformidade de padrão, caracterização de importância e otimização estrutural**. Tese (Doutorado) — Universidade de São Paulo, nov. 2017. Citado na página 16.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. L. F. d. **Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina**. Rio de Janeiro, RJ: LTC, 2011. Citado 8 vezes nas páginas 15, 16, 23, 24, 32, 33, 41 e 45.

FACULDADE DE COMPUTAÇÃO. **Grade Curricular do Bacharelado em Ciência da Computação**. 2025. Disponível em: <https://facom.ufu.br/graduacao/bcc/grade-curricular>. Acesso em: 6 abr 2025. Citado na página 12.

FERNANDES, A. A. T.; FIGUEIREDO FILHO, D. B.; ROCHA, E. C. d.; NASCIMENTO, W. d. S. Leia este artigo se você quiser aprender regressão logística.

Revista de Sociologia e Política, SciELO Brasil, v. 28, p. 006, 2021. Citado na página 25.

FERREIRA, V. H.; LAZZARETTI, A.; VIEIRA NETO, H.; RIELLA, R.; OMORI, J. Classificação de eventos em redes de distribuição de energia utilizando transformada wavelet e modelos neurais autônomos. **Learning & Nonlinear Models**, v. 8, n. 2, p. 93–99, 2010. Citado na página 24.

FREITAS, T. Entendendo as árvores de decisão em machine learning. **Sigmoidal**, 2022. Disponível em: <https://sigmoidal.ai/entendendo-as-arvores-de-decisao-em-machine-learning/>. Acesso em: 6 abr 2025. Citado na página 18.

GAMBALLI, L.; TIGLEA, D. G.; CANDIDO, R.; SILVA, M. T. M. Redes mlp distribuídas para classificação de arritmias cardíacas. **Anais Do XL Simpósio Brasileiro De Telecomunicações E Processamento De Sinais**, 2022. Citado na página 31.

GELMAN, A.; HILL, J. **Data analysis using regression and multilevel/hierarchical models**. [S.l.]: Cambridge university press, 2007. Citado na página 26.

GOMES, M. J.; MONTEIRO, M.; DAMASCENO, A. M.; ALMEIDA, T. J. S. Evasão acadêmica no ensino superior: estudo na área da saúde. **Revista Brasileira de Pesquisa em Saúde**, v. 12, n. 1, 2010. Disponível em: <https://periodicos.ufes.br/rbps/article/view/278>. Citado na página 9.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. [S.l.]: MIT press Cambridge, 2016. v. 1. Citado 2 vezes nas páginas 30 e 31.

HAIR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E. **Multivariate Data Analysis**. 8. ed. [S.l.]: Cengage Learning, 2019. Citado 2 vezes nas páginas 27 e 30.

HOED, R. M. **Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação**. 188 p. Dissertação (Mestrado) — Universidade de Brasília, Brasília, DF, 2016. Citado na página 13.

HUNT, E. B.; MARIN, J.; STONE, P. J. Experiments in induction. Academic press, 1966. Citado na página 15.

IBM. **Logistic Regression**. 2024. Disponível em: <https://www.ibm.com/br-pt/topics/logistic-regression>. Acesso em: 24 mar 2025. Citado 2 vezes nas páginas 25 e 26.

_____. **Random Forest**. 2024. Disponível em: <https://www.ibm.com/think/topics/random-forest>. Acesso em: 21 mar 2025. Citado na página 19.

_____. **Data Mining: What it is and why it matters**. 2025. Disponível em: <https://www.ibm.com/br-pt/topics/data-mining>. Acesso em: 02 abr 2025. Citado 2 vezes nas páginas 9 e 10.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS. **Metodologia de Cálculo dos indicadores de fluxo da educação superior**. Brasília, DF, 2017. Citado na página 13.

INSTITUTO SEMESP. **Mapa do Ensino Superior no Brasil – 12 edição**. 2022. Disponível em: <https://www.semesp.org.br/mapa/edicao-12/>. Acesso em: 6 abr 2025. Citado 2 vezes nas páginas 9 e 13.

_____. **14 Edição do Mapa do Ensino Superior no Brasil**. 2024. Disponível em: <https://www.semesp.org.br/mapa/edicao-14/>. Acesso em: 6 abr 2025. Citado na página 13.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. et al. **An introduction to statistical learning**. [S.l.]: Springer, 2013. v. 112. Citado na página 26.

KONONENKO, I. Machine learning for medical diagnosis: history, state of the art and perspective. **Artificial Intelligence in medicine**, Elsevier, v. 23, n. 1, p. 89–109, 2001. Citado 2 vezes nas páginas 14 e 15.

KOTSIOPOULOS, T.; SARIGIANNIDIS, P.; IOANNIDIS, D.; TZOVARAS, D. Machine learning and deep learning in smart manufacturing: The smart grid paradigm. **Computer Science Review**, v. 40, p. 100341, 2021. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S157401372030441X>. Citado na página 29.

LEE, H.; KIM, J.; KIM, S. Gaussian-based smote algorithm for solving skewed class distributions. **International Journal of Fuzzy Logic and Intelligent Systems**, Korean Institute of Intelligent Systems, v. 17, n. 4, p. 229–234, 2017. Citado na página 44.

LEITE, D. R. A.; MORAES, R. M. de; LOPES, L. W. Método de aprendizagem de máquina para classificação da intensidade do desvio vocal utilizando random forest. **Journal of Health Informatics**, v. 12, 2020. Citado 4 vezes nas páginas 17, 18, 19 e 23.

LOBO, M. Panorama da evasão no ensino superior brasileiro: aspectos gerais das causas e soluções. **Associação Brasileira de Mantenedoras de Ensino Superior. Cadernos**, v. 25, p. 14, 2012. Citado na página 13.

LUDERMIR, T. B. Inteligência artificial e aprendizado de máquina: estado atual e tendências. **Estudos Avançados**, v. 35, n. 101, p. 85–94, 2021. Citado na página 15.

MANDELLI, P. **Aprendizado supervisionado: Entenda como funciona**. 2023. Disponível em: <https://domineia.com/aprendizado-supervisionado>. Acesso em: 29 out 2023. Citado na página 15.

MANHÃES, L. M. B.; CRUZ, S. Predição do desempenho acadêmico de alunos da graduação utilizando mineração de dados. **XIX Simpósio de Pesquisa Operacional e Logística da Marinha. Rio de Janeiro, RJ, Brasil**, v. 6, 2020. Citado na página 35.

MARINHO, N. **Quatro em Cada dez estudantes de ti desistem Da Faculdade**. Jornal Correio, 2024. Disponível em: <https://www.correio24horas.com.br/colunistas/empregos-e-solucoes/quatro-em-cada-dez-estudantes-de-ti-desistem-da-faculdade-0524>. Citado na página 9.

MELLO, S. P. T. de; SANTOS, E. G. dos; BRISOLARA, L. S.; SILVA, R. E. S. da; KOGLIN, J. C. de O. O fenômeno evasão nos cursos superiores de tecnologia: um estudo de caso em uma universidade pública no sul do brasil. In: **XIII Colóquio Internacional sobre Gestão Universitária nas Américas**. [S.l.: s.n.], 2013. p. 1–15. Citado na página 9.

MELO, A. S. da C. **Previsão automática de evasão estudantil: um estudo de caso na UFCG**. 54 p. Dissertação (Mestrado) — Universidade Federal de Campina Grande, Campina Grande, PB, 2016. Citado na página 34.

MINISTÉRIO DA EDUCAÇÃO. **Análise sobre a Expansão das Universidades Federais 2003 a 2012**. 2012. Disponível em: <http://bibliotecadigital.economia.gov.br/handle/123456789/194>. Acesso em: 6 abr 2025. Citado 2 vezes nas páginas 9 e 13.

_____. **Resumo Técnico: Censo da Educação Superior 2007**. 2023. Disponível em: <https://download.inep.gov.br/download/superior/censo/2007/Resumo_tecnico_2007.pdf>. Acesso em: 05 out 2023. Citado na página 9.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. O. (Ed.). **Sistemas inteligentes-Fundamentos e aplicações**. Barueri: Manole, 2003. Citado na página 14.

MOURA, L. G. de. **A implantação do Reuni e o seu impacto na evasão discente**. 157 p. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, MG, 2018. Citado na página 9.

NASTESKI, V. An overview of the supervised machine learning methods. **Horizons. b**, v. 4, p. 51–62, 2017. Citado na página 16.

NILSSON, N. **Learning Machines: Foundations of Trainable Pattern-Classifying Systems**. New York, NY: McGraw-Hill, 1965. Citado na página 15.

RASTROLLO-GUERRERO, J. L.; GÓMEZ-PULIDO, J. A.; DURÁN-DOMÍNGUEZ, A. Analyzing and predicting students' performance by means of machine learning: A review. **Applied sciences**, MDPI, v. 10, n. 3, p. 1042, 2020. Citado na página 14.

ROSENBLATT, F. **Principles of Neurodynamics**. Washington, DC: Spartan Books, 1962. Citado na página 15.

SANTOS, C. O. dos; PEREIRA, R. B.; BORGES, R. Evasão no ensino superior brasileiro: Fatores, causas e consequências. In: **Anais do XIII Congresso Brasileiro de Engenharia de Produção**. [S.l.: s.n.], 2023. Citado na página 13.

SANTOS JUNIOR, J. da S.; MAGALHÃES, A. M. da S.; REAL, G. C. M. A gestão da evasão nas políticas educacionais brasileiras: Da graduação à pós-graduação stricto sensu. **ETD Educação Temática Digital**, UNICAMP, v. 22, n. 2, p. 460–478, 2020. Citado na página 14.

Scikit-Learn Developers. **Naive Bayes: Scikit-Learn Documentation**. 2024. Disponível em: <https://scikit-learn.org/stable/modules/naive_bayes.html>. Acesso em: 21 mar 2024. Citado 3 vezes nas páginas 20, 21 e 22.

SILVA, N. F. F. da. **Análise de sentimentos em textos curtos provenientes de redes sociais**. 112 p. Tese (Doutorado) — Universidade de São Paulo, 2016. Citado na página 17.

SILVEIRA, M. B. G. da; BARBOSA, N. F. M.; PEIXOTO, A. P. B.; XAVIER, É. F. M.; XAVIER JÚNIOR, S. F. A. Aplicação da regressão logística na análise dos dados dos fatores de risco associados à hipertensão arterial. **Research, Society and Development**, v. 10, n. 16, p. e20101622964–e20101622964, 2021. Citado 2 vezes nas páginas 26 e 29.

SOARES, P. L. B.; SILVA, J. P. da. Aplicação de redes neurais artificiais em conjunto com o método vetorial da propagação de feixes na análise de um acoplador direcional baseado em fibra ótica. **Revista Brasileira de Computação Aplicada**, v. 3, n. 2, p. 58–72, 2011. Citado na página 31.

SOOFI, A. A.; AWAN, A. Classification techniques in machine learning: applications and issues. **Journal of Basic & Applied Sciences**, Set Publishers, v. 13, n. 1, p. 459–465, 2017. Citado na página 17.

SOUZA, M.; TOMIKAWA, V.; OLIVEIRA, B.; POLATI, M. Uso da rede neural artificial no planejamento cirúrgico da correção do estrabismo. **Arquivos Brasileiros De Oftalmologia**, v. 67, p. 459–462, 2004. Citado 2 vezes nas páginas 30 e 31.

TEODORO, L. de A.; KAPPEL, M. A. A. Aplicação de técnicas de aprendizado de máquina para predição de risco de evasão escolar em instituições públicas de ensino superior no brasil. **Revista Brasileira de Informática na Educação**, v. 28, p. 838–863, 2020. Citado 2 vezes nas páginas 15 e 34.

UNESCO INSTITUTE FOR EDUCATION. **Educação de adultos: Declaração de Hamburgo, agenda para o futuro**. 1997. Disponível em: <https://unesdoc.unesco.org/ark:/48223/pf0000116114_por>. Acesso em: 05 out 2023. Citado na página 9.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA. **Normas Gerais da Graduação**. 2022. Disponível em: <https://prograd.ufu.br/legislacoes/normas-gerais-da-graduacao-resolucao-462022-congrad>. Acesso em: 10 abr 2025. Citado na página 13.

_____. **Matrícula - Graduação**. 2024. Disponível em: <https://prograd.ufu.br/matricula>. Acesso em: 6 abr 2025. Citado na página 12.

VIANA, F. S.; SANTANA, A. M.; RABÊLO, R. de A. L. Avaliaçao de classificadores para prediçao de evasao no ensino superior utilizando janela semestral. In: SBC. **Anais do XXXIII Simpósio Brasileiro de Informática na Educação**. [S.l.], 2022. p. 908–919. Citado na página 35.