**NOÉ OSÓRIO MACÁRIO**

# AVALIAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DO DIABETES MELLITUS

**LAVRAS-MG**

**2025**

**NOÉ OSÓRIO MACÁRIO**

# AVALIAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DO DIABETES MELLITUS

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, para obtenção do título de Mestre.

Prof. Dr. Paulo Henrique Sales Guimaraes
Orientador

**LAVRAS-MG**
**2025**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Macário, Noé Osório.
Avaliação de modelos de aprendizado de máquina para predição do diabetes mellitus / Noé Osório Macário. - 2025.
92 p. : il.

Orientador(a): Paulo Henrique Sales Guimarães.

Dissertação (mestrado) - Universidade Federal de Lavras, 2025.
Bibliografia.

1. Vigitel. 2. Balanceamento. 3. Sensibilidade. I. Guimarães, Paulo Henrique Sales. II. Título.

**NOÉ OSÓRIO MACÁRIO**

# AVALIAÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DO DIABETES MELLITUS

# EVALUATION OF MACHINE LEARNING MODELS FOR PREDICTING DIABETES MELLITUS

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do Programa de Pós-Graduação em Estatística e Experimentação Agropecuária, para obtenção do título de Mestre.

APROVADA em 20 de fevereiro de 2025

Geraldo Magela Da Cruz Pereira

Anderson Castro Soares De Oliveira

Crysttian Arantes Paixão

Prof. Dr. Paulo Henrique Sales Guimaraes
Orientador

**LAVRAS-MG**
**2025**

## DEDICATÓRIA

*Aos meus pais e irmãos, por serem o alicerce firme que sustentou meus sonhos e por seu amor incondicional que sempre me impulsionou a seguir em frente.*

*À minha esposa e aos meus filhos, minha fonte inesgotável de inspiração, razão do meu esforço diário e meu maior tesouro, mesmo nos momentos em que a distância nos separa.*

*A todos que acreditaram em mim, minha mais profunda gratidão pelo apoio e confiança que tornaram esta conquista possível.*

## AGRADECIMENTOS

A Deus, pela oportunidade de realizar este sonho e por me fortalecer em cada etapa dessa jornada.

À Universidade Federal de Lavras, especialmente ao Departamento de Estatística e Experimentação Agropecuária, por proporcionarem o ambiente e os recursos necessários para a realização deste trabalho.

À CAPES, pelo suporte financeiro que viabilizou minha formação por meio da bolsa de estudos.

Ao professor e orientador Paulo Henrique, pela confiança depositada em mim, pela orientação cuidadosa e por contribuir tão significativamente para o meu amadurecimento acadêmico e pessoal.

Aos amigos que estiveram comigo nos momentos iniciais dessa caminhada, pela dedicação e pelo companheirismo que tornaram essa jornada mais leve e significativa.

Agradeço ao Governo do Distrito de Mocuba por compreender a importância dessa oportunidade e por me dispensar de minhas funções como professor na Escola Secundária Samora Machel, permitindo que eu me dedicasse integralmente a este projeto.

Aos meus pais e irmãos, por todo amor, ensinamentos e apoio incondicional que me sustentaram ao longo dessa caminhada.

À minha esposa e aos meus filhos, que, mesmo à distância, foram meu porto seguro e minha maior motivação, me encorajando a superar os desafios e a honrar o sacrifício de estar longe de casa.

A todos, meu mais profundo reconhecimento e gratidão.

It's not an experiment if you know it's going to work.

Jeff Bezos

## RESUMO

O presente trabalho avalia o desempenho de diferentes modelos de aprendizado de máquina (AM) na predição de Diabetes, uma condição crônica de grande relevância para a saúde pública. Utilizando dados do VIGITEL 2023, que incluem mais de 21 mil observações, foi realizado um processo de pré-processamento completo, que envolveu seleção de variáveis, balanceamento de classes, tratamento de valores ausentes e padronização dos dados. Os algoritmos analisados foram Árvores de Decisão, Florestas Aleatórias, *Naive Bayes*, Redes Neurais Artificiais e *XGBoost*. A avaliação do desempenho dos modelos foi conduzida com base em métricas como sensibilidade e área sob a curva ROC, fundamentais para a identificação de casos positivos e para uma discriminação eficiente entre as classes. O modelo *XGBoost* se destacou como o mais eficaz, apresentando as melhores métricas de sensibilidade, especificidade e área sob a curva ROC em quase todas as abordagens (considerando todas as variáveis, MIC - *Maximal Information Coefficient* e PCA - *Principal Component Analysis*), tanto para dados balanceados quanto desbalanceados, o que evidencia sua superior capacidade preditiva. Em contraste, o modelo de Árvore de Decisão obteve o pior desempenho, destacando suas limitações quando aplicado a dados desbalanceados. Os resultados reforçam o potencial do aprendizado de máquina na detecção precoce de doenças crônicas, como o Diabetes, sublinhando sua relevância para aprimorar diagnósticos médicos, otimizar custos e fornecer suporte crucial para intervenções clínicas mais eficazes.



**Palavras-chave:** vigitel; balanceamento; curva roc; sensibilidade; modelos supervisionados.

## ABSTRACT

The present work evaluates the performance of different models of machine learning (ML) in the prediction of Diabetes, a chronic condition of great relevance for the public health. Using the VIGITEL (2023) data, which include more than 21 thousand observations, a full pre-processing process was carried out, which evolved selection of variables, balancing of groups, treatment of missing values and data standardization. The analyzed programs were Decision Trees, Random Forests, Naive Bayes, Artificial Neural Nets and XGBoost. The evaluation of the performance of the models was held on the basis of metrics such as sensibility and area under the ROC curve, fundamental to identify positive cases and make an efficient discrimination of the groups. The XGBoost model stood out as the most efficient, presenting the better metrics of sensibility, specificity and area under a ROC curve in almost all approaches (considered all the variables, MIC- Maximal Information Coefficient and PCA - Principal Component Analysis), either for balanced data either unbalanced, which shows its predictive superior capacity. Contrarily, the model of Decision Tree had the worst performance, highlighting its limitations when applied to unbalanced data. The results strengthen the potential of learning machine in the earlier detection of chronic diseases, such as Diabetes, underlining its relevance to master medical diagnostics, optimize costs and give crucial support for clinical interventions more efficient.



**Keywords:** vigitel; balancing; roc curve; sensitivity; supervised models.

## INDICADORES DE IMPACTO

Este estudo analisou o desempenho de diferentes modelos de aprendizado de máquina na predição de Diabetes, com base nos dados do VIGITEL 2023, abrangendo mais de 21 mil observações. Os resultados indicaram que o modelo *XGBoost* obteve o melhor desempenho, destacando-se em sensibilidade, especificidade e área sob a curva ROC, superando modelos como Árvore de Decisão, tanto em cenários de dados balanceados quanto desbalanceados (considerando variáveis originais, MIC e PCA). Em contrapartida, a Árvore de Decisão apresentou as menores métricas de desempenho, evidenciando desafios no processamento de dados desbalanceados. Esses achados reforçam o papel do aprendizado de máquina na identificação precoce de doenças crônicas, como o Diabetes, e sua importância para a precisão diagnóstica, redução de custos e mitigação de complicações. Além disso, os resultados podem subsidiar a incorporação de inteligência artificial na prática clínica, promovendo decisões mais assertivas no manejo da doença.

## IMPACT INDICATORS

The study evaluated the performance of different machine learning models in predicting Diabetes, using data from VIGITEL 2023, with more than 21 thousand observations. The results showed that the XGBoost model was the most effective, presenting the best metrics of sensitivity, specificity and area under the ROC curve, outperforming other models, such as Decision Tree, both in approaches with balanced and unbalanced data (considering original variables, MIC and PCA). In contrast, the Decision Tree model presented the lowest performance, evidencing limitations when applied to unbalanced data. These results highlight the potential of machine learning for the early detection of chronic diseases such as Diabetes, underlining its relevance for improving medical diagnoses, optimizing costs and preventing complications, with significant implications for public health management. In addition, the study findings can contribute to the adoption of artificial intelligence technologies in clinical practice, assisting in making more informed and effective decisions in Diabetes treatments.

## LISTA DE TABELAS

## LISTA DE FIGURAS

## SUMÁRIO

# 1 INTRODUÇÃO

O Diabetes é um distúrbio caracterizado pela elevação anormal da glicose no sangue, decorrente de deficiência na liberação ou utilização de insulina (Brutti *et al*., 2019). Esse problema de saúde pública tem se consolidado como uma das principais causas de mortalidade entre adultos, com estimativas indicando que cerca de 6,7 milhões de pessoas entre 20 e 79 anos morreram em decorrência do Diabetes ou de suas complicações em 2021 (International Diabetes Federation, 2021).

Nesse contexto alarmante, o Diabetes é amplamente reconhecido como uma epidemia global, representando desafios relevantes para os sistemas de saúde em todo o mundo.

Diversos fatores contribuem para o aumento da incidência e prevalência do Diabetes em escala mundial. Entre eles, destacam-se o envelhecimento populacional, o aumento da urbanização e a adoção de estilos de vida pouco saudáveis, como o sedentarismo, a dieta inadequada e a obesidade (Casarin, Dalmagro, *et al*., 2022).

A Sociedade Brasileira de Diabetes (2019) reforça que o Diabetes está em ascensão e afeta países de diferentes níveis de desenvolvimento, o que evidencia sua importância como um problema global.

Em 2021 a *International Diabetes Federation* estimou que 10,5% da população mundial entre 20 e 79 anos, equivalente a 537 milhões de pessoas, vivia com Diabetes. Apesar dos esforços já realizados por governos, organizações de saúde e indivíduos, projeções indicam que, sem intervenções ainda mais eficazes, esse número poderá subir para 11,3% em 2030 e 12,2% em 2045, alcançando 783 milhões de pessoas.

No Brasil, o cenário é igualmente preocupante. O país concentra o maior número de pessoas com Diabetes na América Latina e ocupa o quinto lugar no mundo. Entre 1996 e 2019, a mortalidade pela doença quase dobrou, passando de 16,3 para 29 óbitos por 100 mil habitantes, representando 30,1% de todas as mortes prematuras no país (Muzy, Campos, *et al*., 2021).

A prevalência do Diabetes no Brasil varia regionalmente: 6,8% no Norte, 8,7% no Nordeste, 10,5% no Sudeste, 8,5% no Sul e 10,3% no Centro-Oeste, com destaque para as altas taxas de subnotificação, especialmente no Norte, no qual chegam a 72,8% (Reis, Duncan, *et al*., 2022).

Mudanças no estilo de vida têm desempenhado um papel crucial no aumento da prevalência do Diabetes. Um estudo realizado em uma comunidade de origem japonesa mostrou

que a prevalência da doença cresceu de 18,3% em 1993 para 34,0% em 2000, refletindo os impactos de alterações nos hábitos alimentares e na atividade física (Sociedade Brasileira de Diabetes, 2019). Embora realizado em uma população específica, o estudo ilustra uma tendência semelhante observada em outras populações urbanas brasileiras.

Nesse contexto, a inteligência artificial (IA), com destaque para o aprendizado de máquina (AM), tem se mostrado uma ferramenta promissora no estudo e predição do Diabetes.

Com dados clínicos, como históricos médicos, padrões comportamentais e medições laboratoriais, os modelos de aprendizado de máquina identificam padrões que auxiliam na previsão do risco de desenvolvimento da doença. Isso possibilita a implementação de intervenções preventivas.

Nos últimos anos, a IA tem se tornado uma ferramenta essencial, devido ao crescimento exponencial da disponibilidade de dados.

O aprendizado de máquina, um campo da IA, permite que sistemas aprendam a partir de dados sem necessidade de programação explícita (Batista e Filho, 2019). Com a crescente complexidade dos problemas e o volume de dados, o uso de ferramentas computacionais que gerem hipóteses baseadas em experiências anteriores torna-se cada vez mais essencial.

A justificativa deste estudo baseia-se na necessidade de aprimorar as estratégias de diagnóstico do Diabetes, considerando que o diagnóstico tardio está frequentemente associado a complicações graves, comprometimento da qualidade de vida e aumento dos custos de tratamento.

Nesse contexto, o uso de aprendizado de máquina surge como uma abordagem inovadora, oferecendo maior precisão e eficiência na identificação de padrões complexos em grandes volumes de dados, algo que os métodos tradicionais frequentemente apresentam limitações.

Além disso, o aprendizado de máquina também apresenta potencial para ser aplicado em outras doenças crônicas, como hipertensão e doenças cardiovasculares, que igualmente exigem intervenções rápidas e eficazes para mitigar suas consequências.

Diante desse cenário, o objetivo deste estudo é comparar a eficácia de diferentes modelos de aprendizado de máquina na predição de Diabetes, avaliando algoritmos como Árvores de Decisão (AD), Florestas Aleatórias (FA), *Naive Bayes* (NB), Redes Neurais Artificiais (RNA) e *Extreme Gradient Boosting* (XGBoost).

A comparação entre os algoritmos será realizada utilizando métricas de desempenho, como acurácia, precisão, sensibilidade, especificidade e área sob a curva ROC (*Receiver Operating Characteristic*).

A partir deste estudo, espera-se identificar, entre os modelos de AM estudados, aquele que seja o mais eficaz para a predição precoce do Diabetes. Isso proporcionará uma ferramenta valiosa para os profissionais de saúde, contribuindo para a melhoria dos diagnósticos e para a redução dos custos associados ao diagnóstico tardio.

# 2 REFERENCIAL TEÓRICO

Nesta seção, são apresentados os termos e conceitos fundamentais para a compreensão deste trabalho. São abordadas as definições relacionadas ao Diabetes e seus principais tipos, além de tópicos sobre aprendizado de máquina, suas terminologias mais relevantes e os modelos utilizados ao longo deste estudo.

## 2.1 Diabetes

O Diabetes é um grupo de doenças metabólicas caracterizadas por hiperglicemia, associadas a complicações, disfunções e insuficiência de diversos órgãos, especialmente olhos, rins, nervos, cérebro, coração e vasos sanguíneos. A condição pode ser causada por defeitos na secreção e/ou ação da insulina, incluindo processos patogênicos específicos, como a destruição das células beta do pâncreas (responsáveis pela produção de insulina), resistência à ação da insulina e distúrbios na secreção desse hormônio, entre outros. O Diabetes é uma das principais causas de mortalidade, insuficiência renal, amputações de membros inferiores, cegueira e doenças cardiovasculares, resultando em uma significativa perda de qualidade de vida (Neves, Tomasi, *et al*., 2023).

De acordo com a Sociedade Brasileira de Diabetes (2019), a prevalência do diabetes reflete a magnitude do impacto que essa doença impõe aos sistemas de saúde e à sociedade. Além de ser um preditor da carga futura que as complicações crônicas do diabetes representarão, a entidade destaca que, na maioria dos países, entre 5% e 20% do orçamento total da saúde é destinado ao tratamento do diabetes. Além dos custos financeiros, a doença traz outros impactos, como dor, ansiedade, limitações diárias e redução na qualidade de vida, afetando tanto os pacientes quanto suas famílias.

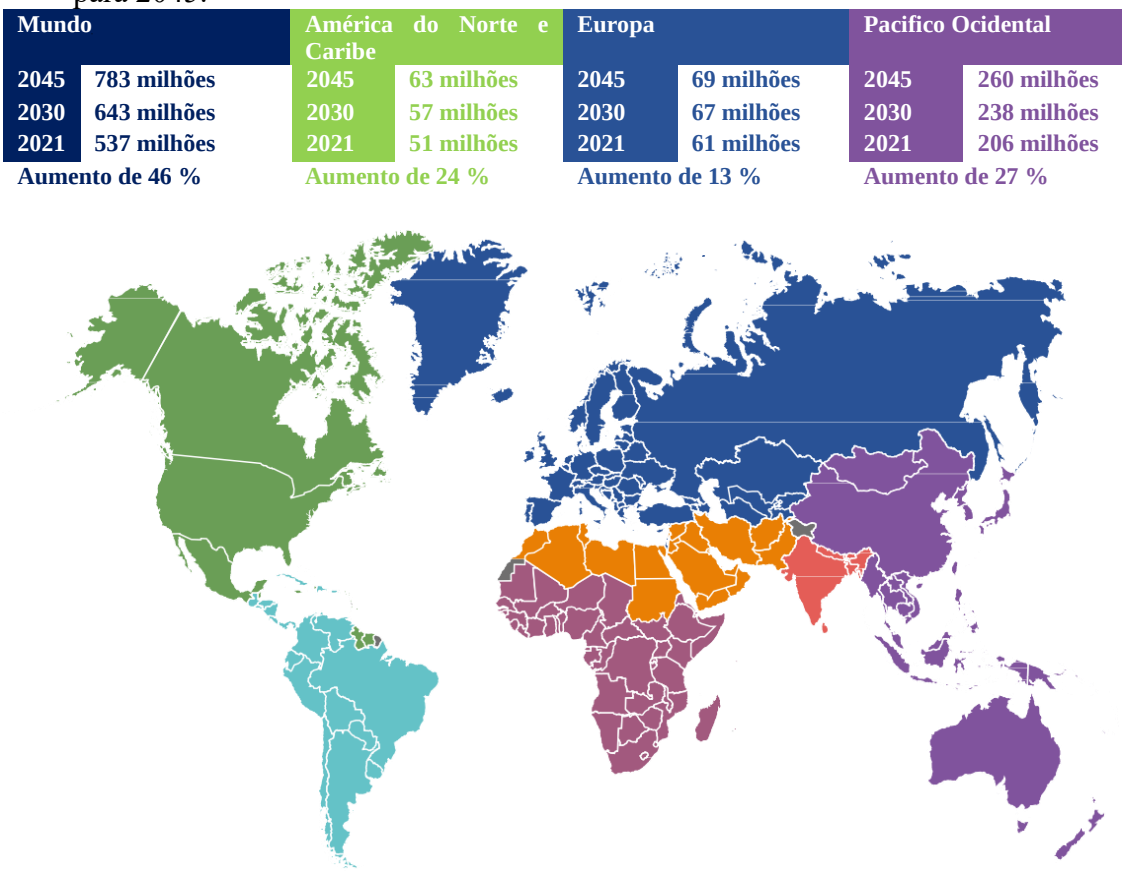
A prevalência do Diabetes varia significativamente entre diferentes grupos étnicos. Por exemplo, os indígenas norte-americanos têm 2,7 vezes mais chances de desenvolver Diabetes do que a população não indígena. Nos índios Pima, do Arizona, praticamente metade dos adultos vive com a doença. No Brasil, foi documentada uma prevalência elevada do Diabetes (28,2%) entre os indígenas Xavante, no estado de Mato Grosso, o que destaca a vulnerabilidade da população nativa das Américas (Sociedade Brasileira de Diabetes, 2019).

O Diabetes contribui de forma expressiva para a mortalidade global, sendo responsável por 10,7% de todas as mortes, superando o número de óbitos causados por doenças infecciosas

como Vírus da HIV/AIDS (Vírus da Imunodeficiência Humana /Síndrome da Imunodeficiência Adquirida) (1,1 milhão), tuberculose (1,8 milhão) e malária (0,4 milhão) somadas. Muitas vezes, o diabetes é subnotificado nas certidões de óbito, pois suas complicações, principalmente as cardiovasculares, são frequentemente listadas como a causa primária de morte (Sociedade Brasileira de Diabetes, 2019).

Conforme ressaltado pela *International Diabetes Federation* (2021), o diabetes alcançou proporções pandêmicas, estando amplamente fora de controle. Globalmente, mais de um em cada 10 adultos convive com a doença conforme ilustrado na Figura 1. Em alguns países as taxas são ainda mais preocupantes, com até um em cada cinco adultos afetados pelo diabetes.

Figura 1 - Número de pessoas com Diabetes no mundo e por região em 2021, com previsão para 2045.




Fonte: Adaptado de *International Diabetes Federation* (2021).

## Tipos de Diabetes

A classificação do Diabetes mellitus (DM) é essencial para garantir um tratamento adequado e para definir estratégias eficazes de rastreamento de doenças associadas e problemas de saúde a longo prazo. Recomenda a classificação com base nas causas e no desenvolvimento da doença, que compreende Diabetes tipo 1 (DM1), Diabetes tipo 2 (DM2), Diabetes gestacional (DMG) e outros tipos de Diabetes (Maraschin, Murussi, *et al.*, 2010). Os casos mais comuns são DM1 e DM2.

### Diabetes Tipo 1

É caracterizado pela destruição das células beta do pâncreas, levando a uma deficiência de insulina. Ele é subdividido em dois tipos: 1A e 1B. O DM tipo 1A é a forma mais frequente de DM1, confirmada pela presença de um ou mais autoanticorpos. O DM tipo 1B é atribuído aos casos de DM1 nos quais os autoanticorpos não são detectáveis na circulação (Sociedade Brasileira De Diabetes, 2019).

### Diabetes Tipo 2

É a forma mais comum de DM, verificada em 90 a 95% dos casos. Caracteriza-se por defeitos na ação e secreção da insulina, além da regulação da produção hepática de glicose. A resistência à insulina e o defeito na função das células beta estão presentes precocemente na fase pré-clínica da doença. O DM2 é causado por uma interação de fatores genéticos e ambientais (Sociedade Brasileira De Diabetes, 2016).

### Diabetes Gestacional

É a hiperglicemia diagnosticada na gravidez, de intensidade variada, que geralmente se resolve no período pós-parto, mas pode retornar anos depois em grande parte dos casos. O diagnóstico do diabetes gestacional é controverso. A Organização Mundial da Saúde recomenda detectá-lo com os mesmos procedimentos diagnósticos empregados fora da gravidez, considerando como diabetes gestacional os valores referidos fora da gravidez como

indicativos de diabetes ou de tolerância à glicose diminuída (Federação Brasileira das Associações de Ginecologia e Obstetrícia, 2019).

**Outras Formas de Diabetes**

Esta categoria inclui todas as formas menos comuns de Diabetes mellitus, que têm apresentações clínicas bastante variadas. Essas variações dependem das alterações subjacentes que provocam o distúrbio do metabolismo do açúcar no sangue. Exemplos incluem: Diabetes neonatal (que pode ser transitório ou permanente), Diabetes mitocondrial, Diabetes lipoatrófico e Diabetes secundário ao uso de medicamentos (Rodacki, Teles e Gabbay, 2023). Cada uma dessas formas requer atenção específica para o diagnóstico e tratamento.

**Principais Sintomas de Diabetes**

O Diabetes pode ser assintomático em muitos casos, o que torna o diagnóstico mais difícil (Ministério da Saúde, 2006). Frequentemente, a suspeita surge devido a fatores de risco, como histórico familiar, obesidade, hipertensão e sedentarismo. Em alguns casos, o diagnóstico é feito tardiamente, quando complicações crônicas já estão presentes, como neuropatia (danos nos nervos), retinopatia (problemas nos olhos) e doenças cardiovasculares (acúmulo de placas nas artérias) (Ministério da Saúde, 2006). Essas complicações têm grande impacto na qualidade de vida e exigem acompanhamento médico constante.

Os sintomas mais comuns do diabetes incluem poliúria (aumento da produção de urina), polidipsia (sede excessiva), polifagia (aumento do apetite) e perda de peso involuntária. Esses sinais indicam problemas no metabolismo da glicose, e seu reconhecimento precoce é fundamental para um diagnóstico rápido e para evitar complicações futuras. Além disso, sintomas como fadiga, fraqueza, falta de energia, coceira na pele e na região genital feminina, inflamação da glande e infecções frequentes também são comuns em pessoas com diabetes descontrolado (Sociedade Brasileira de Diabetes, 2019).

É importante lembrar que esses sintomas podem ser confundidos com os de outras doenças, como problemas hormonais (hipotireoidismo, síndrome de Cushing), distúrbios renais, infecções crônicas ou até questões psicológicas, como depressão e distúrbios alimentares. Por isso, um diagnóstico preciso exige uma avaliação clínica detalhada e exames laboratoriais, como a dosagem de glicose e os testes de hemoglobina glicada, para confirmar a

presença do diabetes e excluir outras condições com sintomas semelhantes (Sociedade Brasileira de Diabetes, 2019).

## 2.2 Vigitel

O Vigitel (Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico) é um sistema de monitoramento contínuo da saúde da população adulta brasileira, conduzido pelo Ministério da Saúde desde 2006. Seu principal objetivo é coletar dados sobre fatores de risco e proteção relacionados a doenças crônicas não transmissíveis (DCNT), como hipertensão, diabetes, obesidade e tabagismo (Brasil, 2023). A importância do Vigitel reside na sua capacidade de fornecer informações estratégicas para o desenvolvimento de políticas públicas voltadas à prevenção e ao controle dessas enfermidades, contribuindo para a redução da morbimortalidade no país.

Os dados do Vigitel são obtidos por meio de entrevistas telefônicas realizadas anualmente com adultos residentes nas capitais dos 26 estados brasileiros e no Distrito Federal. A amostragem é probabilística e baseada nos cadastros de linhas telefônicas fixas. O questionário aplicado inclui perguntas sobre hábitos alimentares, prática de atividade física, consumo de álcool e tabaco, diagnóstico de doenças crônicas e acesso a serviços de saúde (Monteiro et al., 2008). Além dessas informações autorreferidas, o Vigitel também disponibiliza variáveis sociodemográficas, como idade, sexo, escolaridade e estado civil, permitindo análises mais abrangentes dos fatores de risco para doenças crônicas não transmissíveis.

A relevância do Vigitel para estudos de aprendizado de máquina decorre da riqueza e diversidade dos dados coletados, que possibilitam a construção de modelos preditivos voltados à identificação de padrões de risco na população. A aplicação de técnicas de aprendizado de máquina pode melhorar a capacidade de previsão de doenças crônicas, auxiliando na tomada de decisão em saúde pública (Silva, 2023). Além disso, a utilização desses modelos permite identificar interações complexas entre variáveis, possibilitando análises mais aprofundadas sobre os determinantes da saúde e a eficácia de intervenções preventivas. Dessa forma, o Vigitel se consolida como uma importante base de dados para estudos na área de inteligência artificial aplicada à epidemiologia e à predição de riscos à saúde (Dias, 2024).

## 2.3 O processo KDD e a mineração de dados

A Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases* - KDD) é um processo sistemático voltado para a extração de informações relevantes a partir de grandes volumes de dados. Esse processo envolve múltiplas etapas e é amplamente utilizado para apoiar a tomada de decisões em diversas áreas, como saúde, economia e engenharia de software (Fayyad, Piatetsky-Shapiro e Smyth, 1996).

O processo KDD é composto por etapas fundamentais que garantem a identificação de padrões e conhecimentos útis nos dados. Segundo Han, Kamber e Pei (2011), esse processo pode ser dividido em:

a) seleção de dados: escolha dos dados relevantes para análise;
b) preparação dos dados: tratamento de inconsistências, remoção de ruídos e preenchimento de valores ausentes;
c) transformação dos dados: conversão dos dados para um formato adequado, incluindo normalização e redução de dimensionalidade;
d) mineração de dados: aplicação de algoritmos para identificação de padrões e tendências;
e) interpretação e avaliação: validação dos padrões descobertos e extração de conhecimentos acionáveis.

Cada uma dessas etapas desempenha um papel essencial na eficácia do processo KDD, permitindo a extração de informações valiosas de forma confiável.

A mineração de dados é a etapa central do KDD e envolve a utilização de algoritmos avançados para descobrir relações não triviais nos dados. O aprendizado de máquina desempenha um papel essencial nesse processo, pois permite a criação de modelos preditivos e descritivos capazes de identificar padrões ocultos sem a necessidade de regras pré-definidas (Witten, Frank, Hall e Pal, 2016).

As principais abordagens de aprendizado de máquina utilizadas na mineração de dados incluem:

a) aprendizado supervisionado: modelos treinados com dados rotulados para realizar previsões futuras, como redes neurais artificiais e métodos baseados em árvores de decisão (Kotsiantis, 2007);
b) aprendizado não supervisionado: algoritmos que identificam padrões em dados não rotulados, como métodos de *clustering* e redução de dimensionalidade (Han, Kamber e Pei, 2011);

c) aprendizado por reforço: técnica baseada em tentativa e erro para otimização de decisões em ambientes dinâmicos (Sutton e Barto, 2018).

O aprendizado de máquina aplicado à mineração de dados tem impulsionado inovações significativas em diversas áreas, como detecção de fraudes, diagnóstico médico e análise de sentimentos em redes sociais. A capacidade de processar grandes volumes de informação e extrair padrões complexos tem tornado esse campo essencial para o avanço da inteligência artificial e análise de dados.

## 2.4 Aprendizado de Máquina: Conceitos e Tipos

O aprendizado de máquina (AM), ou *machine learning*, é uma área da inteligência artificial (IA) que se concentra em modelos e algoritmos matemáticos e estatísticos que aprendem a partir da experiência, melhorando seu desempenho ao longo do tempo. Amplamente empregado para detectar padrões em dados, automatizar tarefas complexas e realizar predições, o aprendizado de máquina tornou-se um diferencial significativo em várias áreas (Inazawa, *et al.*, 2019)

Um sistema de aprendizado é um programa de computador que toma decisões com base nas experiências acumuladas pela solução bem-sucedida de problemas anteriores. O objetivo principal é permitir que esses sistemas aprendam por meio da experiência, sem a necessidade de programação explícita ou intervenção humana, melhorando seu desempenho com exemplos (Monard e Baranauskas, 2003). Para isso, é necessário um grande número de exemplos para gerar conhecimento, resultando em hipóteses derivadas dos dados (Ludermir, 2021).

O aprendizado de máquina, frequentemente associado à inteligência artificial, também se apoia em diversas outras áreas de pesquisa que desempenham um papel essencial no seu avanço. Disciplinas como Probabilidade e Estatística, Teoria da Computação, Neurociência e Teoria da Informação oferecem contribuições diretas e significativas, fornecendo os fundamentos teóricos e práticos necessários para o desenvolvimento e aprimoramento do AM (Faceli *et al*., 2021).

Existem três tipos fundamentais de aprendizado de máquina: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço, cada um com abordagens distintas para processar dados e tomar decisões (Sadeq, 2024).

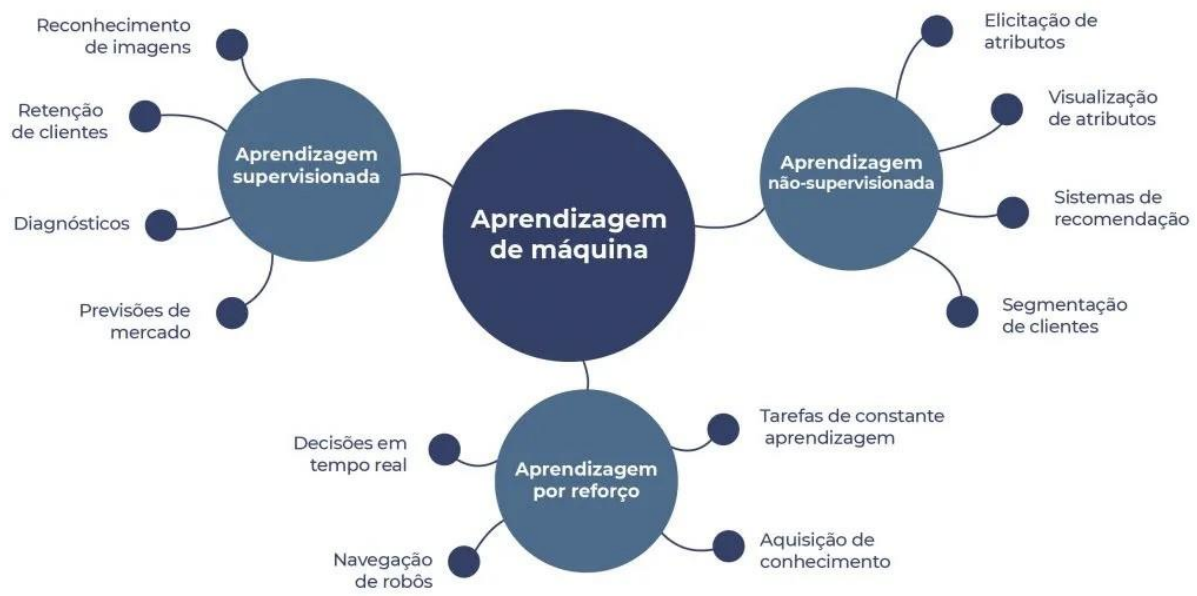
Conforme descrito por Ludermir (2021), esses três tipos principais estão ilustrados na Figura 2, representando as bases centrais do campo. No entanto, o aprendizado de máquina não

se limita a essas metodologias clássicas. Como destacado por Géron (2019), existem inúmeras outras formas de aprendizado que se adaptam a diferentes necessidades e ampliam as possibilidades de aplicação.

Entre essas variações, destaca-se o aprendizado semi-supervisionado, que combina dados rotulados e não rotulados para melhorar o desempenho do modelo, especialmente em cenários com escassez de rótulos (Zhu, 2005). Outra abordagem importante é o aprendizado autossupervisionado, que permite ao modelo aprender de maneira eficiente sem depender de rótulos explícitos, utilizando informações contidas nos próprios dados (Devlin *et al*., 2019).

Além disso, dentro do aprendizado supervisionado, o aprendizado profundo se sobressai como uma subcategoria, mostrando grande eficácia em tarefas complexas, como o reconhecimento de imagens e a tradução de idiomas, por meio de redes neurais profundas (LeCun *et al*., 2015).

Figura 2 - Classificação dos sistemas aprendizado de máquina.




Fonte: Maia (2020).

A figura ilustra a divisão dos principais tipos de aprendizado de máquina, conforme os conceitos descritos por Ludermir (2021). Ela apresenta as três abordagens fundamentais do campo: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por

reforço, que representam as bases do aprendizado de máquina, processando dados e tomando decisões de formas distintas.

### 2.4.1 Aprendizado de Máquina Supervisionado

No aprendizado de máquina supervisionado, a ideia central é utilizar preditores (dados de entrada) para prever uma ou mais respostas (dados de saída). As respostas podem ser quantitativas ou qualitativas, correspondendo a problemas de regressão ou classificação, respectivamente (Morettin e Singer, 2021). Cada exemplo é representado por um vetor de valores (atributos) e uma resposta associada, que pode ser uma classe ou um valor contínuo, dependendo da natureza do problema (Ludermir, 2021).

O objetivo desse processo é construir um modelo de AM capaz de generalizar a partir desses exemplos, permitindo a previsão correta das respostas para novos dados ainda não rotulados. No aprendizado supervisionado, os dados de treinamento são rotulados, ou seja, cada entrada está associada a uma saída específica. A partir desses dados, o modelo aprende a identificar padrões e a fazer previsões, que podem envolver a categorização em classes (classificação) ou a estimativa de valores contínuos (regressão).

Para garantir a eficácia do modelo e evitar o problema do superajuste, os dados são divididos em conjuntos de treinamento e teste. Essa divisão é essencial para estimar o erro de predição e assegurar que o modelo seja capaz de generalizar bem para novos dados.

#### Regressão

Conforme Sicsú, Samartini e Barth (2023), os algoritmos supervisionados de previsão têm como objetivo prever o valor de uma variável alvo quantitativa *Y*, utilizando como base as variáveis preditoras $X_1, X_2, \ldots, X_p$, em que $p$ representa o número total de variáveis preditoras no conjunto de dados.

Para um conjunto de $n$ padrões, em que cada padrão é composto por variáveis explicativas (independentes) e uma variável resposta (dependente), que pode ser contínua ou discreta, o objetivo é construir um modelo de regressão. Este modelo deve estimar o valor mais provável da variável resposta para um novo padrão, identificado pelo índice *i*, em que *i* representa um padrão específico.

O valor da resposta desejada para o padrão $i$ é denotado por $y_i$, enquanto $\hat{y}_i$ representa a resposta predita pelo algoritmo com base na entrada do padrão $i$. A diferença $y_i - \hat{y}_i$, reflete o erro observado para o objeto $i$. O processo de treinamento do modelo visa reduzir esse erro ajustando os parâmetros do modelo, de modo que as saídas preditas se aproximem dos valores desejados (Silva, 2023).

Os métodos de estimação desempenham um papel crucial na construção de modelos de regressão, pois determinam os parâmetros que melhor descrevem a relação entre as variáveis.

O método de Mínimos Quadrados Ordinários (MQO) se destaca entre os mais utilizados, pois minimiza a soma dos quadrados dos resíduos entre os valores observados e os estimados pelo modelo. Este método é amplamente reconhecido por sua simplicidade e eficácia, desde que seus pressupostos fundamentais sejam atendidos (Hastie, Tibshirani e Friedman, 2009).

Para garantir estimativas confiáveis, é fundamental que certos pressupostos sejam cumpridos. De acordo com Zonato et al. (2018), os principais pressupostos incluem:

a) linearidade: a relação entre as variáveis independentes e a dependente deve ser linear. caso essa condição não seja atendida, podem ser necessárias transformações nas variáveis ou a adoção de modelos não lineares;
b) independência dos erros: os resíduos devem ser independentes entre si. dependências temporais ou espaciais podem demandar abordagens específicas, como modelos autorregressivos ou de efeitos mistos (Shumway e Stoffer, 2017);
c) homoscedasticidade: a variância dos erros deve permanecer constante em todas as observações. a heteroscedasticidade pode ser corrigida com modelos ponderados ou transformações nas variáveis (Montgomery, Peck e Vining, 2021);
d) normalidade dos erros: os erros devem seguir uma distribuição normal. embora essa suposição tenha maior relevância para inferência do que para previsão, técnicas robustas podem ser aplicadas quando houver desvios significativos da normalidade.

A violação desses pressupostos pode comprometer a validade das inferências estatísticas derivadas do modelo. A heteroscedasticidade, por exemplo, pode gerar estimativas ineficientes e testes de hipóteses imprecisos. Para identificar e corrigir essas violações, utilizam-se testes e técnicas de diagnóstico, como o teste de Breusch-Pagan para heteroscedasticidade e a análise de resíduos para verificar normalidade e independência.

Além do MQO, existem outros métodos de estimação que podem ser adotados conforme as características dos dados e os objetivos da análise. Em situações com erros não normais ou a presença de *outliers*, métodos robustos são mais adequados. Técnicas como regressão

quantílica e redes neurais artificiais também se destacam, pois têm se mostrado eficazes em aplicações complexas devido à sua capacidade de modelar relações não lineares e lidar com distribuições atípicas (Goodfellow, Bengio e Courville, 2016).

## Classificação

A classificação é uma das categorias mais significativas e amplamente utilizadas no AM. O objetivo principal dos modelos empregados para classificação é aprender regras gerais que mapeiem corretamente as entradas para as saídas correspondentes. Os dados de entrada são geralmente divididos em dois grupos: *X*, representando os atributos que serão usados para determinar a classe de saída, e *Y*, que representa a própria classe de saída (Silva, 2023).

Conforme observado por Sicsú, Samartini e Barth (2023), os algoritmos supervisionados de classificação são essenciais para classificar novas observações em uma das categorias de uma variável qualitativa *Y*, ou seja, para prever em qual categoria uma nova observação deve ser classificada. A variável alvo *Y* pode ser binomial ou multinominal.

De acordo com Aggarwal (2015), a aplicação de problemas de classificação é comum em diversas áreas, incluindo:

a) marketing direcionado ao cliente: variáveis descritivas do cliente podem ser usadas para prever seus interesses de compra com base em exemplos de treinamento anteriores. Aqui, a variável alvo pode indicar o interesse de compra do cliente;
b) diagnóstico de doenças médicas: características extraídas de registros médicos podem ajudar a prever se um paciente desenvolverá uma doença no futuro. Este contexto é crucial para fazer previsões sobre problemas de saúde com base nessas informações;
c) detecção de eventos supervisionada: em cenários temporais, rótulos de classe podem ser associados a carimbos de tempo correspondentes a eventos incomuns, como atividades de intrusão. Métodos de classificação de séries temporais são particularmente úteis nesses casos;
d) análise de dados multimídia: a classificação é valiosa para analisar grandes volumes de dados multimídia, como fotos, vídeos e áudio, que frequentemente apresentam desafios devido à complexidade do espaço de características e à lacuna semântica entre as características e as inferências correspondentes;

e) análise de dados biológicos: em dados biológicos, métodos de classificação são usados para prever propriedades de sequências específicas ou representações de redes biológicas, permitindo uma análise abrangente e detalhada desses dados;
f) categorização e filtragem de documentos: aplicações como serviços de notícias dependem da categorização em tempo real de grandes quantidades de documentos. Essa área, conhecida como categorização de documentos, é fundamental para fornecer informações relevantes aos usuários;
g) análise de redes sociais: a classificação é fundamental na análise de redes sociais, permitindo a associação de rótulos a nós para prever os rótulos de outros nós. Essas aplicações são valiosas para entender e prever o comportamento dos atores em uma rede social.

Estes exemplos ilustram a versatilidade e importância dos algoritmos de classificação em diversos contextos aplicados. Esses métodos supervisionados são fundamentais quando há dados rotulados. No entanto, em muitos cenários, como ao explorar dados não rotulados ou identificar padrões ocultos, o aprendizado não supervisionado pode ser mais adequado.

### 2.4.2 Aprendizado de Máquina não Supervisionado

Ao contrário dos métodos supervisionados, nos quais a qualidade dos resultados pode ser avaliada comparando a previsão com o valor conhecido *Y*, a interpretação e validação dos resultados nos algoritmos não supervisionados são mais desafiadoras (Sicsú, Samartini e Barth, 2023).

Nestes algoritmos não supervisionados, não há uma variável alvo para orientar os resultados, sendo sua função identificar padrões de comportamento entre as observações da amostra. O algoritmo é programado para aprender a reconhecer esses padrões, o que torna as saídas mais difíceis de interpretar devido à ausência de um alvo *Y*.

No aprendizado não supervisionado, os exemplos são fornecidos sem categorias específicas. O algoritmo analisa os atributos desses exemplos e os agrupa com base em suas similaridades, buscando identificar padrões nos dados para formar *clusters* de exemplos semelhantes. Após realizar os agrupamentos, é necessário analisar o significado de cada grupo dentro do contexto do problema em questão para compreender melhor as informações contidas nos dados (Ludermir, 2021).

Embora o aprendizado não supervisionado permita a descoberta de padrões ocultos, o aprendizado por reforço vai além, aplicando *feedback* dinâmico para otimizar ações em ambientes complexos e em constante mudança.

### 2.4.3 Aprendizado de Máquina por Reforço

O aprendizado por reforço é uma abordagem em que uma máquina aprende a partir da interação com um ambiente dinâmico por meio de tentativa e erro. Em vez de precisar de novos exemplos ou de um modelo predefinido da tarefa a ser executada, a única fonte de aprendizado é a própria experiência do agente. O objetivo é desenvolver uma política de ações que maximize o desempenho geral do agente em diversas situações (Anselmo, 2020).

Neste processo, o algoritmo de aprendizado por reforço não recebe explicitamente a resposta correta para suas ações. Em vez disso, ele é guiado por sinais de reforço, que podem ser recompensas ou punições (Géron, 2019). Esses sinais indicam a qualidade das ações realizadas em relação aos objetivos estabelecidos. O algoritmo faz suposições baseadas em suas experiências e realiza ações para testar essas suposições, recebendo *feedback* na forma de recompensas ou punições que ajudam a ajustar seu comportamento (Ludermir, 2021).

O aprendizado por reforço é amplamente utilizado em várias áreas, incluindo jogos e robótica. Um exemplo notável dessa técnica é o *AlphaGo*, um programa de inteligência artificial desenvolvido pela *DeepMind* para jogar o jogo de tabuleiro, que ganhou notoriedade por sua capacidade de competir e vencer jogadores humanos de alto nível, demonstrando o potencial do aprendizado por reforço para resolver problemas complexos e dinâmicos (Ludermir, 2021).

## 2.5 Pré-processamento de dados

O pré-processamento de dados é uma etapa fundamental que ocorre imediatamente após a coleta de dados, desempenhando um papel crucial na qualidade das análises subsequentes. Seu principal objetivo é resolver problemas comuns, como a identificação e correção de dados inconsistentes, a remoção de atributos irrelevantes, a redução de dimensionalidade, o tratamento de valores ausentes e o balanceamento de classes em cenários de desbalanceamento. Essas ações asseguram que os dados estejam limpos, consistentes e prontos para as etapas seguintes da análise (Faceli, et al., 2021).

Além de abordar essas questões, o pré-processamento permite a extração de informações valiosas a partir dos dados. Técnicas de visualização, por exemplo, podem ser empregadas para revelar padrões e insights relevantes. Outro aspecto essencial dessa fase é a modificação da estrutura dos dados, ajustando o nível de granularidade conforme necessário, o que contribui para aprimorar o desempenho dos modelos subsequentes.

Assim, o objetivo principal do pré-processamento é preparar os dados de maneira adequada, facilitando a fase seguinte, que envolve a extração de conhecimento, tornando-a mais eficiente e precisa.

### 2.5.1 Seleção de variáveis e redução de dimensionalidade

A seleção de variáveis é uma etapa fundamental na análise de dados, visando escolher um subconjunto de variáveis de entrada que minimizem o impacto de ruídos e eliminem aquelas que não contribuem significativamente para a descrição ou diferenciação dos exemplos. Esse processo é crucial para aumentar a precisão das predições, como demonstram (Oliveira, Dutra e Rennó, 2005; Zebari et al., 2020).

De acordo com Witten, Frank e Mark (2011), a seleção de variáveis pode ser realizada manualmente, utilizando o conhecimento prévio sobre o problema e a interpretação dos atributos, o que resulta em modelos mais robustos e interpretáveis. A identificação de variáveis elimináveis pode ser complexa, pois muitos atributos não são facilmente visualizáveis, como enfatizam Faceli et al. (2021). Para lidar com essa dificuldade, a literatura propõe técnicas automáticas de seleção, que podem ser classificadas em três categorias: métodos embutidos, baseados em filtro e em *wrapper*.

Os métodos *wrapper* combinam a seleção de características com algoritmos de aprendizado, avaliando a qualidade dos subconjuntos de variáveis por meio de uma estratégia de busca. Por outro lado, os métodos de filtragem são independentes dos algoritmos de aprendizado, tratando a seleção como uma fase de pré-processamento que envolve a remoção de variáveis irrelevantes e a eliminação de redundâncias. Métodos embutidos, por sua vez, integram a seleção diretamente no processo de aprendizado, permitindo a identificação de subconjuntos ideais de variáveis ao final do treinamento, buscando equilibrar a simplicidade dos algoritmos de filtragem com a complexidade dos métodos *wrapper* (Zhiqin et al., 2019).

Para avaliar a associação entre variáveis contínuas e categóricas, foram utilizados o coeficiente de correlação de Pearson e o V de Cramér respectivamente (Barbetta, 2012). O

coeficiente de correlação de Pearson (r) mede a força e a direção da relação linear entre variáveis contínuas. Varia de -1 a 1, no qual valores próximos a 1 ou -1 indicam uma forte associação positiva ou negativa, respectivamente, sendo expresso da forma:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

em que *n* é o número de pares de observações, *x* e *y* são as variáveis numéricas.

O V de Cramér que é uma extensão do teste qui-quadrado e fornece uma medida de efeito que varia de 0 a 1. Valores mais próximos de 1 indicam uma associação forte. A fórmula do V de Cramér é:

$$V = \sqrt{\frac{\chi^2}{n \,.\min\,(k-1, r-1)}}$$

em que $\chi^2$ é a estatística do teste qui-quadrado, *n* é o total de observações, *k* é o número de categorias da variável categórica e *r* é o número de categorias da outra variável categórica.

O V de Cramér facilita a interpretação dos resultados do teste qui-quadrado, permitindo quantificar a força das associações de forma clara e comparável (Barbetta, 2012).

### 2.5.2 Transformação de variáveis

A transformação das variáveis é um processo essencial na análise de dados, especialmente ao lidar com variáveis de diferentes naturezas. Variáveis quantitativas são medidas em uma escala numérica, enquanto as qualitativas indicam a presença ou ausência de um atributo, sem permitir uma medição direta. Para tornar as variáveis qualitativas compatíveis com as análises quantitativas, adotou-se a estratégia de criação de variáveis artificiais, conhecidas como variáveis *dummy* (Missio e Jacobi, 2007), que assumem o valor 1 para indicar a presença de um atributo e 0 para sua ausência. Essa estratégia permite a conversão de dados categóricos em uma forma binária numérica, facilitando a análise estatística e a aplicação de modelos de aprendizado de máquina que exigem entradas numéricas.

Além disso, muitos algoritmos de AM apresentam restrições quanto aos tipos de valores que podem processar, preferindo entradas numéricas padronizadas. Alguns algoritmos podem

ser sensíveis às diferenças de escala entre variáveis, o que pode levar a uma influência desproporcional das variáveis com maior amplitude nos resultados (Faceli et al., 2021). Para mitigar esse efeito, utiliza-se a padronização, ajustando as variáveis numéricas a uma faixa comum. A padronização pode ser realizada utilizando a transformação *Z-score* (Morettin e Singer, 2021), que ajusta os dados a uma distribuição com média 0 e desvio padrão 1, condição essencial para o bom desempenho de diversos algoritmos.

### 2.5.3 Tratamento de valores faltantes

Os dados faltantes representam um desafio comum em diversas áreas de pesquisa, especialmente em estudos que dependem de informações coletadas por meio de questionários ou levantamentos. A ausência de respostas não apenas dificulta a análise estatística e o aprendizado de máquina, mas também pode comprometer a integridade dos resultados, introduzindo vieses e reduzindo a precisão das inferências. Quando não tratados adequadamente, os valores ausentes podem distorcer os achados e limitar a validade das conclusões. Portanto, é essencial aplicar métodos rigorosos para a identificação e o tratamento desses dados (Miot, 2019; Wang, 2019).

Os dados faltantes podem surgir de diversas fontes, que vão desde erros de digitação até a real indisponibilidade de informações durante o processo de coleta. Para melhor compreensão, esses valores são normalmente classificados em três categorias principais:

a) *missing completely at random* (MCAR): a ausência ocorre de forma completamente aleatória, sem qualquer relação com as variáveis estudadas (Miot, 2019);
b) *missing at random* (MAR): o valor ausente está relacionado a outra covariável observável, mas não à própria variável faltante (Miot, 2019);
c) *missing not at random* (MNAR): os dados ausentes estão diretamente ligados à variável estudada, o que pode introduzir vieses significativos na análise (Wang, 2019).

Cada um desses padrões demanda métodos específicos para mitigar impactos negativos na análise. Quando a proporção de dados faltantes é pequena e segue um padrão aleatório (MAR ou MCAR), existem várias opções de imputação disponíveis. No entanto, para dados ausentes com padrão não aleatório (MNAR), é fundamental contar com a supervisão de um profissional estatístico experiente, capaz de identificar e tratar adequadamente esse tipo de ausência.

Uma abordagem comum para lidar com dados faltantes é a exclusão da variável ou observação que apresenta essa condição, uma vez que muitos algoritmos dependem de dados

completos. Contudo, essa não é a única ou a mais viável solução, dependendo do contexto da análise.

Outra estratégia é a imputação de dados, que é uma técnica utilizada para lidar com valores ausentes e pode ser dividida em imputação simples e múltipla. A imputação simples preenche valores ausentes com uma única estimativa, como a média; no entanto, pode introduzir vieses e subestimar a variabilidade. Por outro lado, a imputação múltipla gera várias estimativas para cada valor ausente, capturando a incerteza e resultando em análises mais robustas e confiáveis.

Embora a imputação múltipla seja mais complexa e demore mais, seus benefícios em precisão e validade a tornam uma escolha preferencial em muitos casos no qual a qualidade dos dados é essencial (Xuan, An e Shan, 2014).

De acordo com Xuan, An e Shan (2014), um dos métodos de imputação eficazes é o *k-Nearest Neighbor* (*K-NN Imputation*), que utiliza valores de vizinhos próximos completos para estimar os dados ausentes. Devido à sua simplicidade e elevada eficácia na imputação, este método é amplamente estudado e aplicado, sendo uma excelente técnica. Sua formulação matemática é:

$$d(E_i, E_j) = \sqrt{\sum_{r=1}^{M} (x_{ir} - x_{jr})^2}$$

em que $E_i$ e $E_j$ são duas observações (vizinhos) e $M$ é o número de variáveis presentes no conjunto de dados, $x_{ir}$ é o valor assumido pela *r*-ésima variável $A_r$ do *i*-ésimo exemplo $E_i$ e $x_{jr}$ é o valor assumido pela *j*-ésima variável $A_r$ do *i*-ésimo exemplo $E_j$.

### 2.5.4 Balanceamento de dados

O problema de dados desbalanceados é uma questão comum na classificação, afetando o desempenho de diversos algoritmos de aprendizado de máquina. Em muitos conjuntos de dados reais, o número de observações varia significativamente entre as classes, com algumas classes apresentando um volume desproporcional de registros. Esse desequilíbrio ocorre em aplicações como diagnóstico de doenças, detecção de fraudes e previsão de inadimplência, em que certos subconjuntos de dados são mais frequentes que outros. Quando algoritmos de classificação enfrentam essa situação, tendem a favorecer a classe majoritária, resultando em

previsões tendenciosas e afetando a precisão do modelo (Rufino, Veiga e Nakamoto, 2016; Faceli, et al., 2021).

Para lidar com o desbalanceamento, existem duas abordagens principais: (1) reestruturar o conjunto de treinamento para modificar a distribuição dos exemplos e (2) adaptar o algoritmo de aprendizado para compensar o impacto das classes desbalanceadas. A primeira categoria inclui métodos de reamostragem, enquanto a segunda abrange técnicas como *ensemble* de classificadores, aprendizado sensível a custos e seleção de características, que ajudam o algoritmo a identificar e dar atenção a classes minoritárias (Zhi-Fei, Yi-Min e Bao-Liang, 2009).

Os métodos de reamostragem, amplamente utilizados, visam balancear o conjunto de treinamento aumentando o número de exemplos da classe minoritária com amostragem para cima (*up-sampling*) e reduzindo o número de exemplos da classe majoritária com amostragem para baixo (*down-sampling*). O *up-sampling* busca aumentar a representatividade das classes minoritárias, enquanto o *down-sampling* reduz a presença da classe majoritária para alcançar uma distribuição mais uniforme. Essas técnicas auxiliam na melhoria da taxa de reconhecimento das classes menos representadas, aumentando a exatidão das previsões.

## 2.6 Avaliação e validação dos modelos de classificação

A avaliação e a validação de modelos de classificação são etapas fundamentais no desenvolvimento de sistemas preditivos, pois garantem que os algoritmos escolhidos não apenas se ajustem bem aos dados de treinamento, mas também sejam capazes de generalizar corretamente para novos dados. Esse processo envolve a comparação entre as previsões geradas pelo modelo e os valores reais, permitindo a identificação do método mais adequado para o problema em questão (Géron, 2019; Faceli et al., 2021).

A seleção do melhor modelo pode levar em conta diferentes critérios, como interpretabilidade, eficiência computacional e desempenho preditivo. Entre esses, o desempenho preditivo frequentemente assume papel central, sendo avaliado por métricas específicas que quantificam a qualidade das previsões.

### 2.6.1 Métricas de avaliação

A avaliação do desempenho de um modelo de aprendizado de máquina é realizada por meio da comparação entre as predições do modelo, representadas pelos valores previstos, e os

valores reais. Essa análise permite estimar a eficácia do modelo e sua adequação ao problema em questão.

Em problemas de regressão, as métricas de avaliação quantificam a diferença entre os valores previstos e os valores observados. Entre as principais métricas, destacam-se o erro médio absoluto (MAE), o erro quadrático médio (MSE) e a raiz do erro quadrático médio (RMSE).

Por outro lado, em tarefas de classificação, nas quais as previsões são categóricas, a avaliação do desempenho é baseada em métricas específicas que analisam a taxa de acertos e erros. Entre as métricas mais comuns para esse tipo de problema, estão: acurácia, precisão, especificidade, *recall* (sensibilidade), F1-score e a área sob a curva ROC (AUC-ROC).

**Matriz de confusão**

Uma ferramenta essencial para a análise do desempenho do modelo é a matriz de confusão, que compara as classes previstas com as classes reais. Ela fornece uma visão detalhada dos acertos e erros cometidos pelo modelo, permitindo uma análise mais profunda dos resultados, como ilustrado no Quadro 1.

Quadro 1- Representação da matriz de confusão.

| | **Classe Positiva** | **Classe Negativa** |
|---|---|---|
| **Previsto como positivo** | **VP** | **FP** |
| **Previsto como negativa** | **FN** | **VN** |

Fonte: Autor (2025).

A matriz de confusão é composta por quatro elementos principais:

a) verdadeiros positivos (VP): são os casos em que o modelo previu corretamente as instâncias da classe positiva;
b) falsos positivos (FP): representam os casos em que o modelo previu incorretamente uma instância como positiva, quando na verdade era negativa;
c) verdadeiros negativos (VN): são os casos em que o modelo previu corretamente as instâncias da classe negativa;

d) falsos negativos (FN): representam os casos em que o modelo previu incorretamente uma instância como negativa, quando na verdade era positiva.

A disposição na matriz permite uma análise clara do desempenho do modelo em relação a cada classe e auxilia no cálculo de métricas como acurácia, *recall*, especificidade e *F1-Score*, entre outras, que são importantes para a avaliação do modelo de classificação.

Acurácia é uma métrica geral que dá uma visão rápida do desempenho global do modelo. No entanto, em conjuntos de dados desbalanceados (no qual uma classe é muito mais frequente que outra), a acurácia pode ser enganosa, pois um modelo que sempre prediz a classe majoritária terá alta acurácia, mas será inútil. A acurácia é dada por meio da fórmula:

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + VN + FN}$$

Precisão é uma métrica de avaliação que indica a proporção de verdadeiros positivos entre todos os exemplos classificados como positivos.

$$\text{Precisão} = \frac{VP}{VP + FP}$$

*Recall* ou sensibilidade é a proporção de verdadeiros positivos entre todos os exemplos verdadeiramente positivos e é crucial em contextos em que a identificação dos casos positivos é extremamente importante, como em diagnósticos médicos de doenças graves, no qual falha em identificar um caso positivo (falso negativo) pode ter consequências severas.

$$Recall \text{ ou Sensibilidade } = \frac{VP}{VP+FN}$$

Especificidade é a proporção de verdadeiros negativos entre todos os exemplos verdadeiramente negativos, importante em contextos no qual é crucial evitar falsos positivos, como em testes de triagem em que um falso positivo pode levar a tratamentos desnecessários.

$$\text{Especificidade} = \frac{VN}{VN + FP}$$

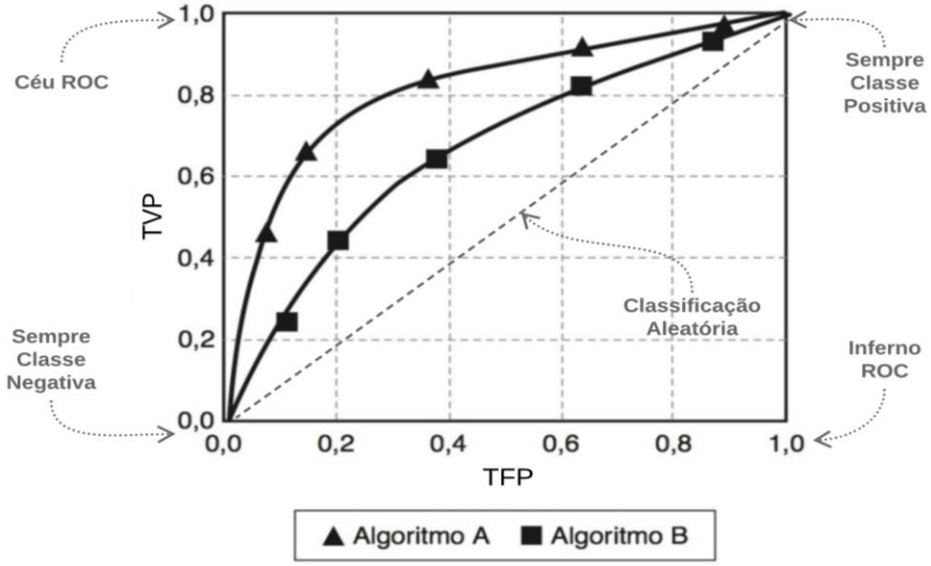
*$F_1$_Score* é a média harmônica entre precisão e recall, e é especialmente útil quando há um equilíbrio entre a precisão e a sensibilidade e quando há uma necessidade de um único valor que reflita ambas as métricas, especialmente em situações de classes desbalanceadas.

$$\text{F1\_Score} = \frac{2 * \text{Precisão} * \text{Sensibilidade}}{\text{Precisão+ Sensibilidade}}$$

**Curva ROC e área sob a curva (AUC)**

A curva ROC (*Receiver Operating Characteristic Curve*) é essencial para avaliar modelos de classificação binária, como na distinção entre pacientes com e sem certa doença. Ela plota a sensibilidade (verdadeiros positivos) no eixo vertical e a taxa de falsos positivos (1 - especificidade) no eixo horizontal. A Figura 10 ilustra essas características, incluindo pontos como Céu ROC e Inferno ROC, que representam casos extremos de classificação.

Figura 3 - Exemplo de curvas ROC.




Fonte: Vergara (2020).

Um modelo ideal se aproxima do canto superior esquerdo, indicando alta sensibilidade e baixa taxa de falsos positivos. Uma linha diagonal representa um classificador aleatório, onde sensibilidade e especificidade são iguais (Metz, 1978).

A curva ROC ajuda a comparar o desempenho de diferentes modelos e a escolher o limite ótimo de classificação com base nos custos dos erros. A área sob a curva ROC (AUC) resume o desempenho geral do modelo, variando entre 0 e 1. Uma AUC de 0,5 indica

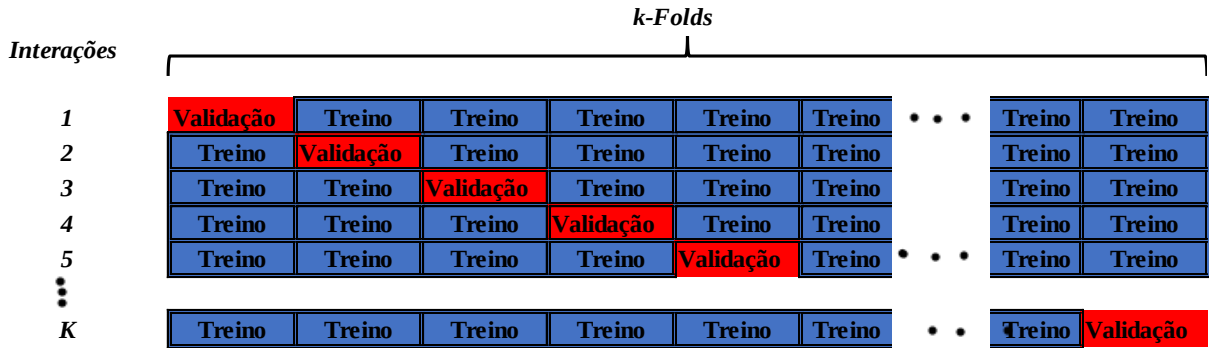
desempenho aleatório, enquanto valores próximos a 1 sinalizam excelente capacidade de distinguir entre classes (Vergara, 2020).

### 2.6.2 Validação cruzada

A validação cruzada é um método estatístico essencial para avaliar e comparar algoritmos de aprendizado, dividindo os dados em dois segmentos: um para treinamento e outro para validação. Quando não há conjuntos de validação pré-definidos, a técnica de validação cruzada *k-fold* é amplamente utilizada (Morettin e Singer, 2021).

Na validação cruzada *k-fold*, os dados são divididos em $k$ segmentos (ou *folds*) de tamanhos iguais (ou quase iguais). O modelo é treinado e validado $k$ vezes, com um *fold* sendo utilizado para validação e os $k - 1$ *folds* restantes para treinamento em cada iteração, como podemos ver na Figura 3. Essa abordagem garante que cada ponto de dado tenha a chance de ser validado, proporcionando uma avaliação robusta do desempenho do modelo. As métricas de desempenho, como a acurácia, são calculadas para cada iteração, e a média dessas métricas é usada para avaliar o modelo.

Figura 4 - Processo de validação cruzada *k-Fold*.




Fonte: O autor (2025).

A figura ilustra o processo de validação cruzada *k-fold*, conforme descrito por Morettin e Singer (2021). As principais variações dessa técnica incluem:

a) *k-fold cross-validation*: Divide os dados em $k$ partes iguais e repete o processo $k$ vezes;
b) *leave-one-out cross-validation* (LOOCV): Cada observação é usada uma vez como conjunto de teste. Ideal para conjuntos de dados muito pequenos, embora seja mais intensiva em termos de computação;

c) *stratified k-fold cross-validation*: Preserva a proporção de classes do conjunto original, sendo útil para problemas de classificação com classes desbalanceadas.

Além disso, a validação cruzada pode ser combinada com a busca de hiperparâmetros para otimizar o desempenho do modelo e minimizar o risco de *overfitting* (ou sobreajuste), que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, prejudicando sua capacidade de generalizar para novos dados. Embora essa abordagem demande maior custo computacional, os benefícios em termos de uma avaliação mais precisa frequentemente justificam o investimento (Yadav e Shukla, 2016).

### 2.6.3 Treinamento e teste

No aprendizado de máquina, especialmente no supervisionado, o objetivo principal é desenvolver modelos com alta capacidade preditiva, que sejam capazes de fazer boas previsões em novas observações. Para alcançar esse objetivo, é fundamental dividir o banco de dados original em duas partes: o conjunto de treinamento e o conjunto de teste (Monard e Baranauskas, 2003).

O conjunto de treinamento é a parte do banco de dados usada para treinar o modelo. Durante essa fase, o modelo aprende os padrões presentes nos dados e ajusta seus parâmetros para minimizar os erros nas previsões. Essa etapa é crucial, pois determina como o modelo irá se comportar ao lidar com novas observações. Em geral, o conjunto de treinamento corresponde à maior parte do banco de dados original, cerca de 75% das observações, embora essa proporção possa variar conforme o caso. O objetivo é fornecer ao modelo dados suficientes para que ele capture as relações subjacentes entre as variáveis de entrada e a variável de saída (Silva, 2020).

Para avaliar o desempenho de um classificador em dados inéditos, é essencial medir sua taxa de erro em um conjunto de teste, que normalmente corresponde a cerca de 25% das observações do banco de dados original e é mantido separado durante o treinamento do modelo. Essa abordagem assegura que o modelo seja avaliado com dados que ele nunca viu antes, proporcionando uma estimativa mais realista de seu desempenho em situações práticas.

O conjunto de teste permite verificar se as previsões do modelo são consistentes e próximas dos valores reais, garantindo que ele não apenas memorize os dados de treinamento, mas também consiga generalizar bem para novos casos. Essa prática é fundamental para evitar o superajustamento, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento e perde a capacidade de se desempenhar bem em dados desconhecidos.

Ao adotar essa estratégia de separação, é possível desenvolver modelos mais robustos e confiáveis, capazes de realizar previsões precisas em diferentes contextos e aplicações, atendendo às demandas de situações reais (Monard e Baranauskas, 2003).

### 2.6.4 Viés e variância

Um dos desafios fundamentais na modelagem é equilibrar o viés e a variância. Modelos simples tendem a apresentar alto viés e baixa variância, enquanto modelos mais complexos geralmente exibem baixo viés e alta variância. O desafio consiste em encontrar um ponto de equilíbrio entre esses dois extremos, um conceito conhecido como *trade-off* entre viés e variância.

De acordo com Sicsú, Samartini e Barth (2023), para criar um modelo preditivo eficaz, é crucial minimizar os erros entre os valores ajustados e os observados da variável alvo (Y). Esses erros, ou discrepâncias, precisam ser reduzidos para que o modelo seja capaz de generalizar adequadamente para outras amostras da população.

Esse *trade-off* é ilustrado por dois fenômenos: *underfitting* e *overfitting*. O *underfitting* ocorre quando um modelo excessivamente simples não consegue capturar a complexidade das relações entre a variável alvo (Y) e as variáveis preditoras ($X_1, X_2, ..., X_p$), sendo $X_p$ o número de variáveis preditoras. Isso resulta em alto viés e grandes discrepâncias entre os valores previstos e observados.

Por outro lado, o *overfitting* acontece quando um modelo excessivamente complexo se ajusta de forma exagerada aos dados de treinamento, incluindo o ruído presente neles. Isso gera alta variância, tornando o modelo excessivamente sensível a pequenas variações nos dados de treinamento e prejudicando seu desempenho ao ser aplicado a novos dados.

## 2.7 Modelos de classificação em aprendizado supervisionado

A escolha do modelo de AM é essencial para o sucesso das previsões em um sistema de diagnóstico assistido, como na predição do Diabetes. Modelos diferentes possuem arquiteturas e algoritmos próprios, capazes de capturar padrões específicos nos dados, o que influencia diretamente o desempenho preditivo. Entre as abordagens de AM, cinco algoritmos têm relevância e aplicabilidade destacadas, são eles:

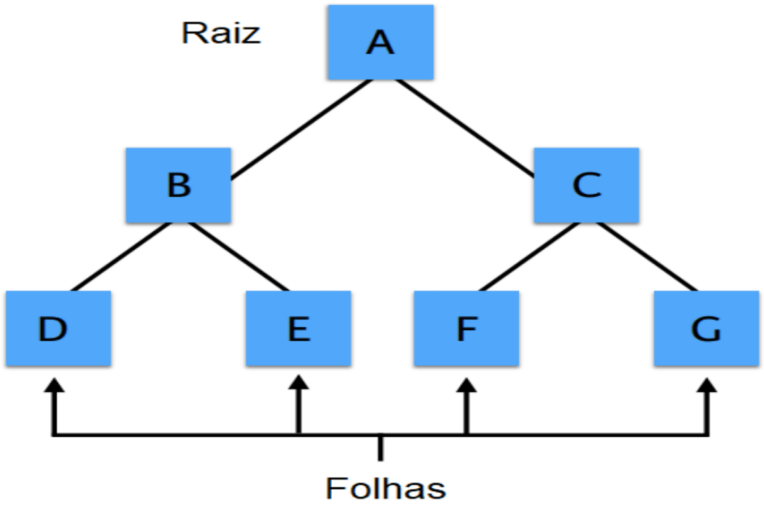
### 2.7.1 Árvores de Decisão (AD)

Uma árvore de decisão é um modelo de AM supervisionado usado para prever a classe ou o valor de uma variável alvo, aprendendo regras de decisão simples inferidas dos dados. O algoritmo começa com o conjunto de dados completo e procura a melhor maneira de dividi-los em subconjuntos mais puros. Isso é feito escolhendo uma variável (ou atributo) e um ponto de divisão (ou valor de corte) que resulte na separação mais eficaz dos dados, com base em algum critério de impureza, como entropia ou índice Gini (Rokach e Maimon, 2005).

A construção da árvore é feita de cima para baixo, começando com o nó raiz que representa todo o conjunto de dados. Cada nó subsequente representa um subconjunto dos dados originais, dividido com base nas condições estabelecidas nos passos anteriores. O processo de divisão continua até que sejam cumpridos critérios de parada, como profundidade máxima da árvore, número mínimo de amostras por nó, ou quando novas divisões não melhoram a pureza (Rokach e Maimon, 2005; Aggarwal, 2015).

Para fazer uma previsão, uma nova observação é passada pela árvore de acordo com as regras estabelecidas nos nós. A observação segue o caminho da raiz até um nó folha, onde a classe ou valor é previsto com base na maioria das observações de treinamento que chegaram àquela folha. AD são modelos interpretáveis, permitindo que suas decisões sejam facilmente entendidas e visualizadas. Cada nó representa uma condição de divisão em um atributo, e os caminhos das raízes às folhas representam conjuntos de regras de decisão, conforme ilustrado na Figura 4.

Figura 5 - Representação de um modelo de árvore de decisão.




Fonte: Silva (2020).

As Árvores de Decisão podem ser usadas tanto para classificação quanto para regressão, embora sejam mais comuns na primeira. Ele processa uma instância, percorrendo os nós da árvore e realizando comparações baseadas em atributos importantes por meio de declarações condicionais. A direção tomada, seja para o ramo esquerdo ou direito, depende do resultado dessas comparações. Em geral, os atributos mais relevantes para a tomada de decisão estão localizados próximos à raiz da árvore.

Existem dois critérios comumente usados para calcular a impureza ou nível de informação de um nó e orientar suas divisões:

a) o índice de Gini pode ser obtido por expressão:

$$Gini = 1 - \sum_{j=1}^{c} (p_j)^2$$

em que $p_j$ é a proporção de observações da classe $j$ no nó e $c$ são o número total de classes. Este mede a probabilidade de uma amostra aleatória ser classificada incorretamente se escolhêssemos aleatoriamente um rótulo de acordo com a distribuição das classes no nó. Valores menores de Gini indicam nós mais puros, onde a maioria das observações pertence a uma única classe;

b) a entropia é obtida por meio da expressão

$$H = -\sum_{j=1}^{c} p_j \log (p_j)$$

em que $p_j$ é a proporção de observações da classe $j$ no nó, $c$ é o número total de classes e o logaritmo (geralmente na base 2) da proporção de observações da classe $j$. A entropia é uma medida de desordem ou incerteza. Em um nó, a entropia é calculada somando-se os produtos das proporções de cada classe $p_j$ pelo logaritmo dessas proporções. Quanto menor a entropia, mais puro é o nó. A entropia também é usada para calcular o ganho de informação, que mede a redução na incerteza sobre a classe da variável alvo após a divisão do nó;

c) O ganho de informação é dado por:

$$GI = H(antes) - \sum_{i=1}^{m} \frac{n_i}{n} H(i)$$

em que $H(antes)$ é a entropia do nó antes da divisão, $H(i)$ é a entropia dos nós filhos, $n_i$ é o número de observações no nó filho *i*, *n* é o número total de observações no nó original e $m$ é o número de nós filhos. O ganho de informação busca maximizar a redução da incerteza, favorecendo divisões que resultem em nós mais puros.

Os algoritmos mais comuns para construir árvores de decisão incluem o ID3 (*Iterative Dichotomiser* 3), que utiliza a entropia e o ganho de informação para selecionar os atributos mais relevantes; o C4.5, uma evolução do ID3 que suporta dados contínuos e implementa poda para evitar complexidade excessiva; e o CART (*Classification and Regression Trees*), que se baseia no índice Gini e permite tanto classificação quanto regressão (Breiman et al., 1984).

Um aspecto crítico no uso de árvores de decisão é a possibilidade de ocorrer *overfitting*, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento e perde capacidade de generalização. Para mitigar esse problema, podem ser empregadas técnicas como a limitação da profundidade máxima da árvore, a definição de um número mínimo de amostras para divisão de nós e a poda. A poda pode ser pré-pronta, no qual restrições são aplicadas durante a construção da árvore, ou pós-pronta, em que ramos pouco significativos são eliminados após a árvore ter sido completamente construída (Quinlan, 2014).

Além disso, a validação cruzada pode ser utilizada para avaliar o desempenho do modelo em diferentes subconjuntos dos dados, ajudando na escolha de parâmetros que minimizem o risco de *overfitting*. Essas práticas, que incluem a validação cruzada e a otimização dos parâmetros, permitem construir árvores de decisão robustas, que equilibram simplicidade estrutural e alta capacidade preditiva.

Em problemas de regressão, o Erro Quadrático Médio (MSE) é utilizado para avaliar a qualidade das divisões e pode ser obtido por meio da formula

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

em que $y_i$ são o valor real da variável alvo para a observação *i*, $\hat{y}_i$ são o valor previsto pela árvore para a observação *i* e *N* é o número total de observações no nó.

O MSE mede a média dos quadrados dos erros, ou seja, a média das diferenças ao quadrado entre os valores previstos e os valores reais. Valores menores de MSE indicam previsões mais precisas.

O índice Gini, a entropia e o MSE ajudam a determinar a qualidade das divisões ao longo da árvore, orientando o algoritmo a criar nós que maximizem a pureza das subdivisões ou minimizem o erro nas previsões (Anselmo, 2020).

### 2.7.2 Florestas Aleatórias (FA)

A Floresta Aleatória é um modelo supervisionado que aprimora o conceito das Árvores de Decisão. Em uma Árvore de Decisão, um novo valor é avaliado por meio de uma série de perguntas para prever a melhor resposta. A Floresta Aleatória expande essa ideia criando várias Arvores de Decisão a partir de diferentes subconjuntos de dados aleatórios, cada uma treinada independentemente. A decisão final é obtida pela votação majoritária entre todas as árvores da floresta (Breiman, 2001; Anselmo, 2020).

O processo de construção de uma Floresta Aleatória inclui:

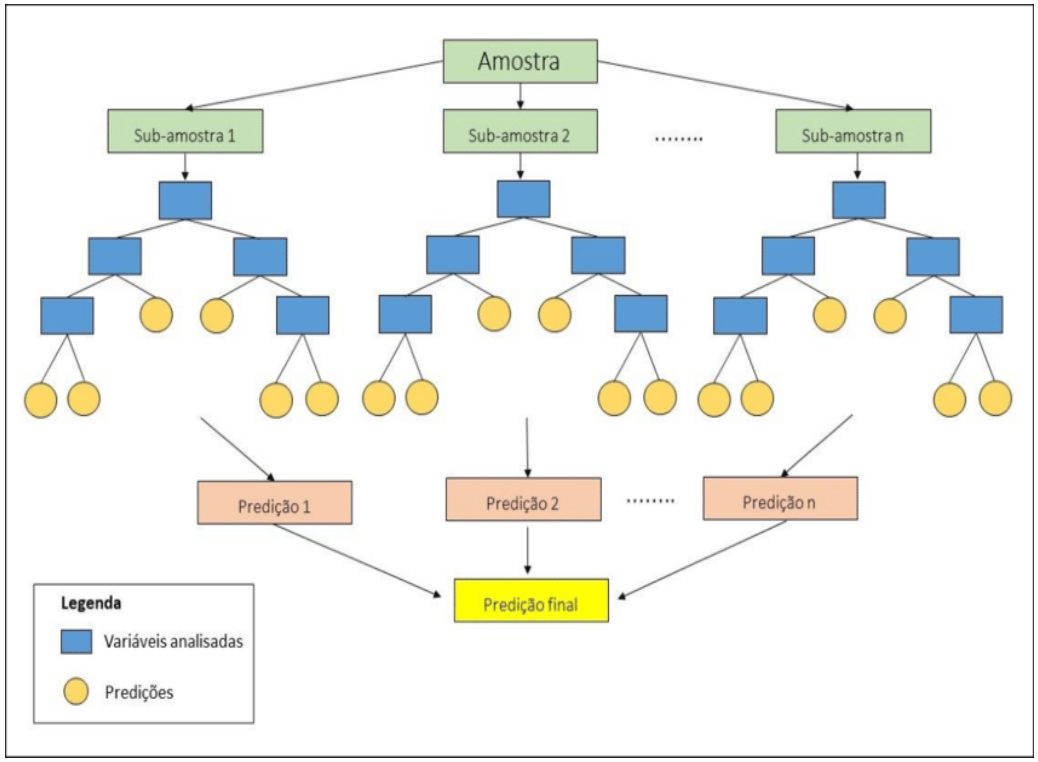
a) criação de múltiplas árvores: A floresta é composta por várias Árvores de Decisão, treinadas com subconjuntos diferentes dos dados originais, gerados por amostragem com reposição, conhecida como *bootstrap*. Esta, é uma técnica de reamostragem que seleciona aleatoriamente observações com reposição, ou seja, algumas podem ser repetidas enquanto outras ficam de fora. Esse processo assegura que cada árvore receba uma versão ligeiramente distinta dos dados, aumentando a diversidade entre elas e melhorando a precisão do modelo;
b) seleção aleatória de atributos: Para cada divisão em uma árvore, um subconjunto aleatório dos atributos é considerado. Essa abordagem aumenta ainda mais a diversidade entre as árvores, permitindo que cada uma se especialize em diferentes aspectos dos dados, evitando que o modelo dependa excessivamente de variáveis específicas. Isso contribui para a robustez do modelo e reduz o risco de *overfitting*;
c) importância das variáveis: Durante o treinamento das árvores, a importância de cada variável é medida com base em sua contribuição para a redução da impureza nas divisões. Variáveis que ajudam a melhorar a previsão, ou seja, aquelas que mais reduzem a impureza, são consideradas mais importantes. Essa análise de importância permite

interpretar o modelo e entender quais atributos têm maior influência nas previsões, facilitando a seleção de variáveis mais relevantes e a melhoria do modelo;

d) combinação dos resultados: Cada árvore realiza uma previsão para novas entradas, e a previsão final é determinada pela classe mais votada entre todas as árvores da floresta. Este processo de votação fortalece a robustez do modelo, pois combina a diversidade de opiniões de várias árvores, resultando em um modelo menos sensível a variações nos dados de treinamento.

Embora o número elevado de árvores possa aumentar o tempo de processamento, a Floresta Aleatória mantém a capacidade de interpretar os resultados, permitindo a análise da importância das variáveis e proporcionando uma visão clara de como os atributos contribuem para a decisão final, como ilustrado na Figura 5.

Figura 6 - Representação esquemática de uma floresta aleatória.




Fonte: Ariza, *et al.* (2022).

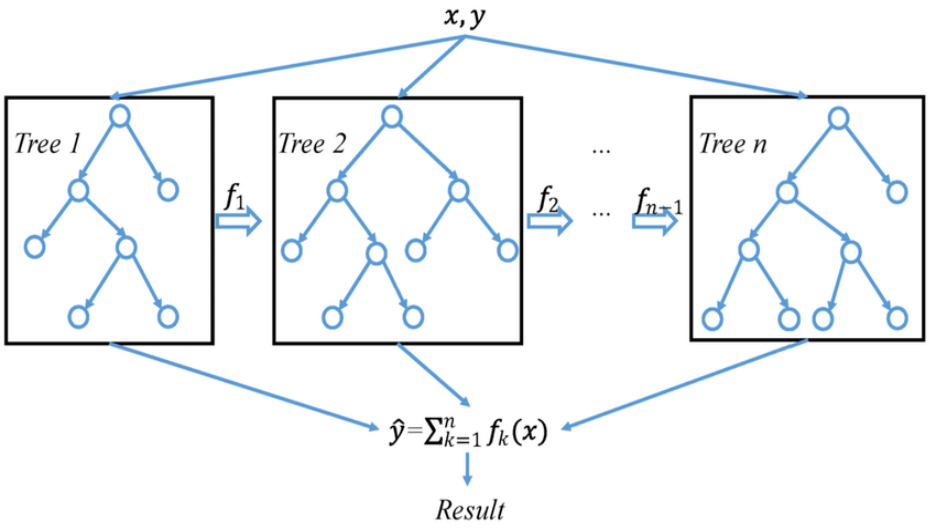
A Floresta Aleatória oferece melhorias significativas em precisão e robustez em comparação com AD individuais, mantendo a capacidade de interpretação e ajustabilidade (Breiman, 2001).

### 2.7.3 *Extreme Gradient Boosting* (XGBoost)

O *XGBoost* é uma técnica avançada de aprendizado em conjunto (*ensemble learning*) baseada no conceito de *boosting*, destinada a melhorar a precisão dos modelos de AM. Conhecido por sua eficiência computacional e desempenho superior (Ali, *et al*., 2023). *XGBoost* é uma implementação de *gradient boosting* que utiliza múltiplas árvores de decisão iterativas.

Cada árvore em *XGBoost* aprende a partir dos resíduos de todas as árvores anteriores. Em vez de adotar a saída de votação majoritária, como na Floresta Aleatória, a saída prevista do *XGBoost* é a soma de todos os resultados, conforme ilustrado na Figura 6 (Wang, *et al.*, 2019).

Figura 7 - Arquitetura geral do *XGBoost*.




Fonte: Wang, *et al.* (2019).

Desenvolvido para lidar com grandes conjuntos de dados, *XGBoost* é amplamente utilizado em competições de ciência de dados, como *Kaggle* (plataforma *online* que oferece um ambiente para a prática de ciência de dados, AM e análise de dados), e em diversas aplicações do mundo real. Suas principais características incluem várias técnicas de regularização, como L1 (*Lasso*) e L2 (*Ridge*) que são técnicas para evitar *overfitting* e melhorar a generalização dos modelos.

O Lasso adiciona à função de custo a soma dos valores absolutos dos coeficientes, forçando alguns a zero, o que elimina variáveis irrelevantes e gera um modelo mais simples (Tibshirani, 1996). Já Ridge adiciona a soma dos quadrados dos coeficientes, reduzindo a

magnitude deles, mas sem eliminar variáveis, o que ajuda a aumentar a estabilidade do modelo, e a flexibilidade de permitir que os usuários definam suas próprias funções de perda para adaptar o modelo a diferentes tipos de problemas (Hoerl e Kennard, 1970).

O *XGBoost* é altamente eficiente em termos de processamento paralelo, o que o torna adequado para grandes volumes de dados. Comparado a outros algoritmos de aprendizado de máquina, o *XGBoost* geralmente produz resultados com maior precisão, é robusto em relação a *outliers* e ruído nos dados, e foi otimizado para executar mais rapidamente, reduzindo o tempo necessário para treinamento e predição. É amplamente utilizado em uma variedade de tarefas de AM, como classificação e regressão, sendo especialmente eficaz em problemas complexos onde a precisão é crucial (Ali, *et al*., 2023).

O funcionamento do modelo *XGBoost*, um poderoso algoritmo de aprendizado de máquina baseado em árvores de decisão, pode ser descrito detalhadamente a partir dos seguintes passos:

a) entrada (*x, y*): os dados de entrada para o modelo são apresentados como *x* e *y*. Aqui, *x* refere-se às *features* (características) que descrevem os dados, enquanto *y* representa o rótulo, ou seja, o valor que buscamos prever. Essa estrutura é fundamental, pois permite que o modelo aprenda a relação entre as características e os resultados;
b) múltiplas árvores de decisão: o *XGBoost* se distingue por não depender de uma única árvore de decisão, mas sim de um conjunto delas. Cada árvore (denotada como *Tree 1, Tree 2, ..., Tree n*, conforme mostrado na Figura 6) é construída de maneira sequencial. Cada nova árvore é criada com base nos erros das árvores anteriores, permitindo que o modelo aprenda e se ajuste continuamente. Esse processo de construção sequencial melhora a precisão do modelo ao corrigir as falhas das árvores anteriores;
c) função f(x): cada árvore no modelo *XGBoost* representa uma função *f(x)*. Essa função é responsável por mapear as features de entrada (x) para uma previsão específica. A capacidade de cada árvore em fazer previsões contribui para a robustez do modelo como um todo;
d) soma das funções: a previsão final do *XGBoost* é calculada pela soma das previsões de todas as árvores. Isso implica que cada árvore vota e, assim, a previsão final é uma combinação ponderada desses votos. A equação na parte inferior da Figura, $\hat{y} = \sum_{k=1}^{n} f_k(x)$, exemplifica essa soma matemática, onde $\hat{y}$ é a previsão final e $f_k(x)$ representa a contribuição de cada árvore para a previsão;

e) resultado: o resultado final é a predição do modelo para um novo conjunto de dados, resultando em um valor que pode ser interpretado de acordo com o contexto do problema em questão.

Essa estrutura esclarece o funcionamento do *XGBoost* e destaca a importância de cada componente na geração de previsões precisas. O uso de múltiplas árvores e a combinação de suas previsões demonstram a capacidade do modelo de aprender e se adaptar, tornando-o uma ferramenta valiosa em tarefas de aprendizado de máquina.

### 2.7.4 *Naive Bayes* (NB)

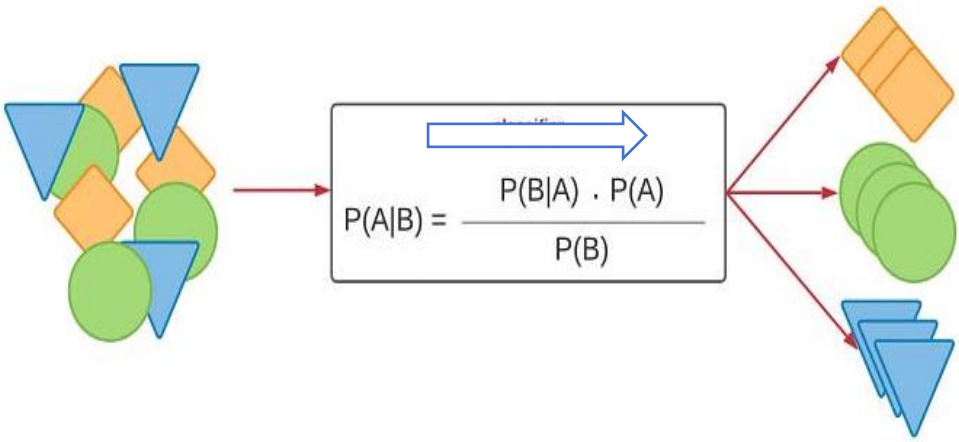
O algoritmo *Naive Bayes* é um classificador probabilístico baseado no Teorema de Bayes, formulado por Thomas Bayes (1701-1761) (Anselmo, 2020). Ele calcula probabilidades condicionais para classificar os dados, utilizando a fórmula do Teorema de Bayes:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

em que $P(A \mid B)$ é a probabilidade de A ocorrer dado que B ocorreu, $P(B \mid A)$ é a probabilidade condicional de B dado A, $P(A)$ *e* $P(B)$ são as probabilidades marginais da classe A e dos atributos B, respectivamente.

A Figura 7 ilustra o conceito, mostrando como os atributos (B) são usados para determinar a classe (A) com base no teorema de Bayes.

Figura 8 - Exemplo de aplicação do *Naive Bayes*.




Fonte: Adaptado de Ajaymehta (2023).

O funcionamento do *Naive Bayes* e seus principais elementos podem ser descritos da seguinte maneira:

a) entrada de dados: os dados são representados por formas geométricas que simbolizam as diferentes características que serão analisadas pelo classificador. Cada forma corresponde a um conjunto de atributos que define os dados a serem classificados;
b) cálculo de probabilidades: o classificador utiliza a fórmula de Bayes para calcular a probabilidade de um dado pertencer a uma determinada classe, considerando suas características;
c) classificação: com base nas probabilidades calculadas, o classificador atribui cada dado à classe com a maior probabilidade. Essa etapa é crucial, pois garante que cada entrada de dados seja classificada de maneira eficiente e precisa, permitindo ao modelo fazer previsões sobre novas entradas com base no que aprendeu.

O *Naive Bayes* é flexível e pode ser implementado com diferentes distribuições probabilísticas, incluindo:

a) distribuição gaussiana (Normal): assume que os atributos seguem uma distribuição normal. É particularmente útil quando os dados são contínuos e podem ser modelados com uma curva de sino;
b) distribuição multinomial: adequada para dados de contagem e é frequentemente utilizada em problemas de classificação de texto, como análise de sentimentos e filtragem de spam;
c) distribuição de bernoulli: utilizada para dados binários, em que cada atributo é tratado como um evento binário (presença ou ausência).

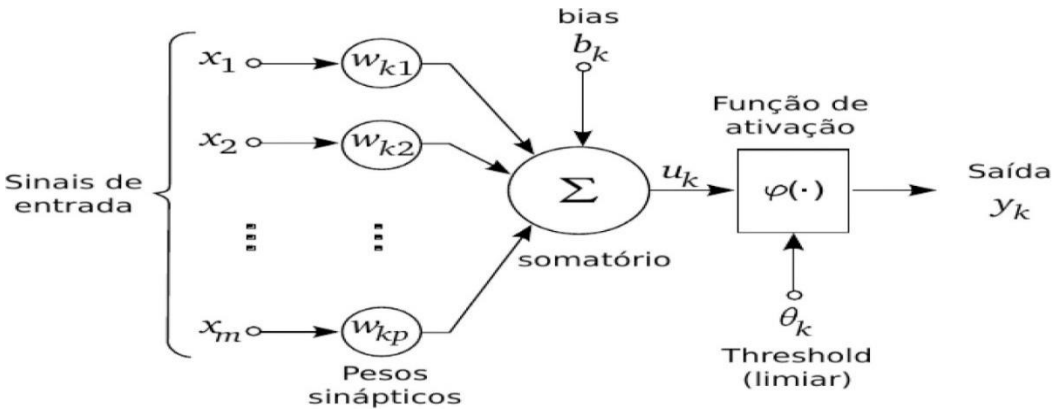
Essa flexibilidade permite que o NB seja aplicado a uma ampla variedade de problemas, como previsões em tempo real, classificação de múltiplas classes, sistemas de recomendação, análise de texto e análise de sentimentos (Ali *et al*., 2023).

### 2.7.5 Redes Neurais Artificiais (RNA)

A arquitetura mais simples de uma rede neural é o *Perceptron*, um algoritmo de aprendizado supervisionado para classificação binária. Ele consiste em nós de entrada e um nó de saída, em que cada entrada está associada a pesos $w$, que calculam uma função $\varphi(\cdot)$ das entradas. O número de unidades de entrada depende da dimensionalidade dos dados e variáveis

categóricas podem precisar ser convertidas em representações binárias. Os pesos w são análogos às sinapses do cérebro biológico (Aggarwal, 2015), conforme ilustrado na Figura 8.

Figura 9 - Modelo de neurônio artificial.




Fonte: Silva e Schimidt (2016).

A ilustração apresentada representa um neurônio artificial simplificado, o componente fundamental de uma rede neural. Os elementos principais incluem:

a) sinais de entrada $(x_1, x_2, \ldots, x_n)$: dados recebidos pelo neurônio, que podem ser números, valores booleanos ou outras formas numéricas;

b) pesos sinápticos $(w_{k1}, w_{k2}, \ldots, w_{kp})$: associados a cada entrada, determinam a importância relativa de cada sinal na saída. Pesos maiores indicam maior influência da entrada correspondente;

c) soma (Σ): realiza a soma ponderada das entradas

$$uk = (x_1 * w_{k1}) + (x_2 * w_{k2}) + \ldots + (x_n * w_{kn}) + b_k$$

em que, $b_k$ é o *bias* (viés), um valor constante adicionado à soma para ajustar a saída do neurônio.

d) função de ativação φ(.): aplica uma transformação não linear ao resultado da soma $(u_k)$, introduzindo não linearidade na rede neural e permitindo a aprendizagem de representações complexas. Sem essas funções, a saída seria linear, limitando a capacidade de aprendizado. Funções de ativação são essenciais para lidar com dados complexos (Sharma, Sharma e Athaiya, 2020). Entre as funções de ativação comuns, destacam-se:

a) sigmoide: produz uma saída entre 0 e 1, utilizada em camadas ocultas e na saída em problemas de classificação binária;

b) *rectified linear unit* (ReLU): retorna zero para entradas negativas e a própria entrada para entradas positivas. É eficiente computacionalmente e ajuda a mitigar o problema do gradiente desaparecido;

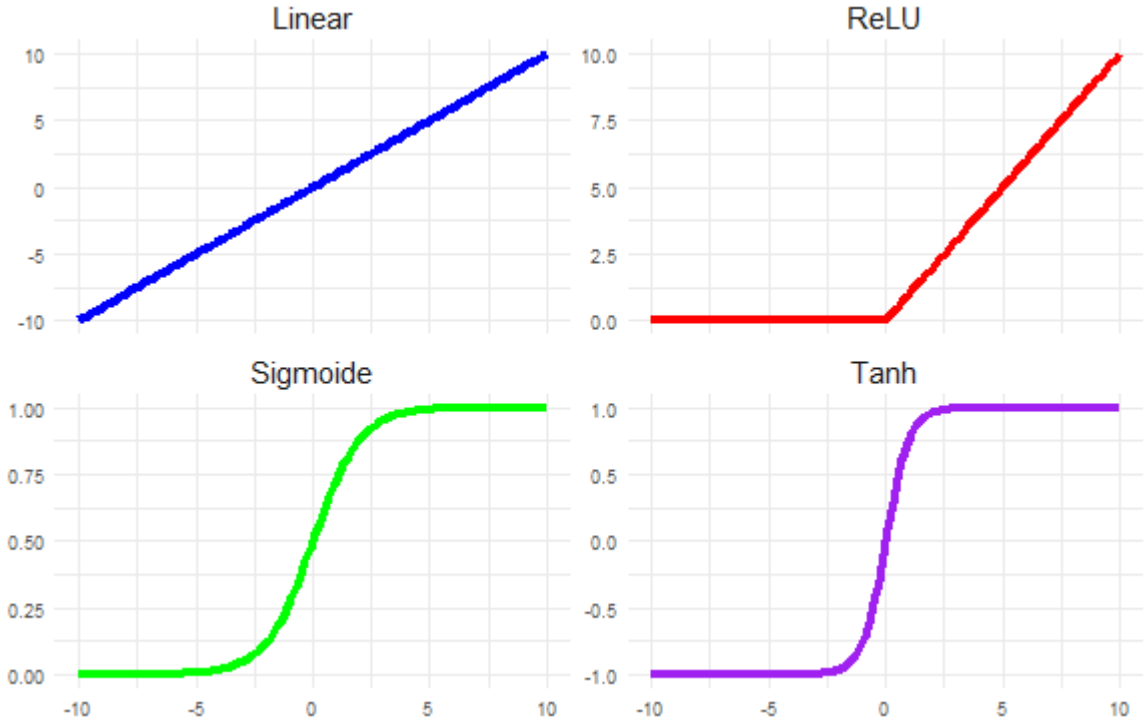
c) tangente hiperbólica (Tanh): produz uma saída entre -1 e 1, utilizada em camadas ocultas devido à sua simetria em torno de zero;

d) linear: retorna a entrada sem modificação, usada quando a saída desejada é proporcional à entrada (Sharma, Sharma e Athaiya, 2020);

e) saída ($y_k$)**:** resultado final do neurônio após a aplicação da função de ativação. Essa saída pode ser utilizada como entrada para outros neurônios em camadas subsequentes da rede.

Como ilustrado na Figura 9, os gráficos dessas funções de ativação mostram claramente o comportamento e as características de cada uma delas.

Figura 10 - Ilustração das diferentes funções de ativação.




Fonte: O autor (2024).

Além do *Perceptron*, diversas arquiteturas de redes neurais desempenham papéis importantes em diferentes áreas. As Redes *Perceptron* Multicamadas são amplamente utilizadas em tarefas de classificação e regressão (Goodfellow, Bengio e Courville, 2016), enquanto as Redes Convolucionais são aplicadas no processamento de imagens e vídeos (LeCun et al., 1998). As Redes de Transformadores, como BERT e GPT, são essenciais para o aprendizado contextual de texto (Vaswani et al., 2017). Essas e outras arquiteturas ampliam significativamente as aplicações das redes neurais em diversas áreas.

# 3 MATERIAL E MÉTODOS

Este capítulo descreve a origem dos dados, o pré-processamento e as técnicas aplicadas na avaliação dos modelos de aprendizado de máquina para predição do Diabetes Mellitus, garantindo a reprodutibilidade dos experimentos.

## 3.1 Material

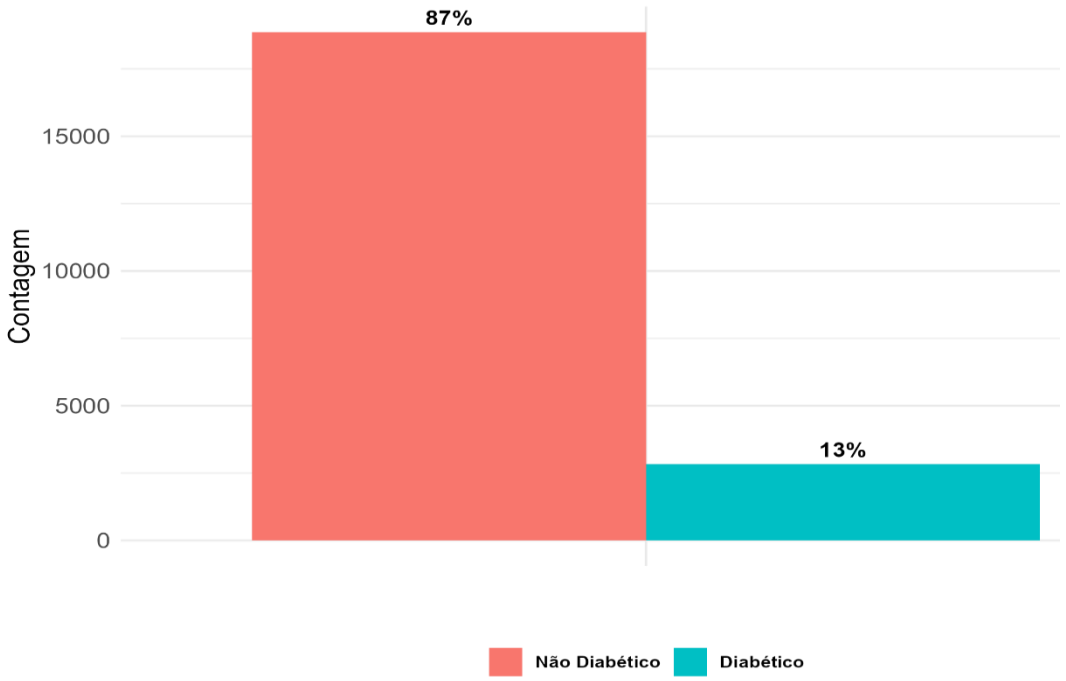
Apresenta-se o conjunto de dados utilizado e os procedimentos de preparação para a aplicação dos modelos preditivos, incluindo tratamento e seleção de variáveis para otimizar a análise.

### 3.1.1 Conjunto de Dados

Este estudo utiliza dados do VIGITEL (Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico), um banco de dados disponibilizado pelo Ministério da Saúde do Brasil. Os dados foram coletados por meio de entrevistas telefônicas, nas quais os participantes forneceram informações autorrelatadas. O conjunto de dados selecionado é referente ao VIGITEL de 2023, contendo 21690 observações e 233 variáveis. Desse total, 8132 registros correspondem a homens e 13558 a mulheres, com idades variando entre 18 e 109 anos.

Os participantes foram categorizados em duas classes de diagnóstico: 2826 são diabéticos e 18864 não diabéticos. Tanto os dados quanto o dicionário de variáveis estão disponíveis para consulta no site do VIGITEL (https://svs.aids.gov.br/download/Vigitel/). A Figura 11 ilustra a distribuição dessas classes, no qual a cor azul representa os pacientes diabéticos e a cor rosa os não diabéticos.

Figura 11 - Dimensão do conjunto de dados.




Fonte: O autor (2025).

### 3.1.2 Recursos Computacionais

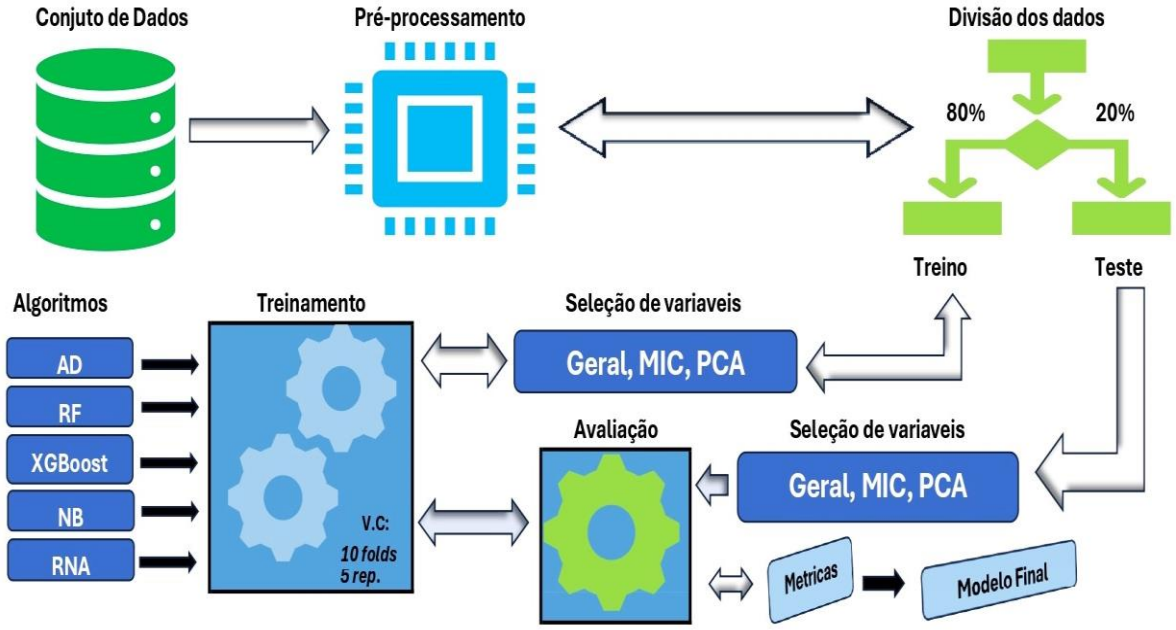
O estudo foi conduzido utilizando a linguagem R (R Core Team, 2023), reconhecida por sua capacidade de lidar com grandes volumes de dados e por oferecer uma ampla gama de pacotes voltados à manipulação e modelagem estatística avançada. Como ambiente de desenvolvimento, utilizou-se o *RStudio Desktop*, que otimizou o fluxo de trabalho ao integrar funcionalidades como gestão de projetos, suporte a sistemas de controle de versão e ferramentas para a geração de gráficos.

Os experimentos foram realizados em um sistema computacional equipado com o sistema operacional *Windows 10 Pro*, equipado com 8 GB de memória RAM e um processador Intel Core i5-8250U de 8ª geração, garantindo a capacidade de processamento necessária para a execução dos modelos de aprendizado de máquina aplicados à predição. Para análises com maior demanda computacional, recorreu-se à infraestrutura do Laboratório de Análise de Dados (LAD) da Universidade Federal de Lavras pertencente ao Departamento de Estatística, que disponibilizou recursos computacionais de alta performance, permitindo maior eficiência e agilidade no processamento das tarefas mais intensivas.

## 3.2 Metodologia

Este estudo segue uma abordagem sistemática para garantir a eficiência e a precisão dos modelos de aprendizado de máquina aplicados à predição de diabetes. A metodologia adotada compreende diversas etapas fundamentais, incluindo a seleção de variáveis, transformação e padronização dos dados, tratamento de valores ausentes e balanceamento do conjunto de dados. A Figura 12 apresenta um esquema geral do fluxo metodológico adotado.

Figura 12 - Esquema de um projeto de aprendizado de máquina.




Fonte: O autor (2025).

### Seleção de variáveis

Inicialmente, foram removidas 14 variáveis não informativas, ou seja, aquelas que não agregavam valor à predição e poderiam introduzir ruídos. Em seguida, eliminamos variáveis redundantes, caracterizadas por forte correlação, o que poderia comprometer a estabilidade do modelo. O coeficiente de correlação de Pearson foi utilizado para avaliar a associação entre variáveis contínuas, sendo aplicado um limite de correlação de 0,8. Em casos de correlação superior a esse valor, uma das variáveis do par foi excluída, resultando na retenção de 13 variáveis. Para variáveis categóricas, o índice V de Cramér foi utilizado, levando à exclusão de variáveis com associações excessivamente fortes, restando 70 variáveis ao final dessa etapa.

Além disso, foram comparados três conjuntos de variáveis: o conjunto completo, um conjunto selecionado pelo *Maximal Information Coefficient* (MIC) e outro reduzido por meio da *Principal Component Analysis* (PCA).

### *Maximal Information Coefficient*

O MIC foi empregado devido à sua capacidade de capturar relações não lineares entre variáveis, sendo definido como:

$$MIC(D) = \max_{X,Y<B(n)} \left(\frac{I(D,X,Y)}{\lg(\min(|X|,|Y|)}\right).$$

em que $D = \{(f_{1i}, f_{2i}, i = 1,2, \dots, n\}$ representam o conjunto de pares ordenados, em que cada par consiste em duas variáveis $f_1$ e $f_2$, $X$ e $Y$ são divisões dos intervalos $f_1$ e $f_2$, que podem estar representando discretizações das variáveis, $B(n)$ limita o número de grades ou intervalos que podem ser utilizados para a análise, $\lg(\min(|X|, |Y|)$ é o termo de normalização que limita o MIC entre 0 e 1, com |X| e |Y| representando o número de partições nos eixos $X$ e $Y$, e $I(D, X, Y)$ é a informação mútua máxima alcançada por qualquer grade $X$ e $Y$.

Calculada utilizando um esquema de *binning* (divisão em intervalos) dependente dos dados (Kinney e Atwal, 2014). Esse processo assegura que o valor do MIC varie entre 0 e 1.

### *Principal Component Analysis*

A PCA foi aplicada para transformar variáveis correlacionadas em componentes principais não correlacionados, preservando a maior parte da variância dos dados originais, conforme a equação:

$$\boldsymbol{Z = XW}$$

em que **Z** é a matriz dos componentes principais, **X** é a matriz de dados originais (com observações em linhas e variáveis em colunas) e **W** é a matriz de pesos ou autovetores, derivados da decomposição da matriz de covariância dos dados.

**Transformação e padronização dos dados**

A preparação dos dados incluiu a codificação de variáveis categóricas em *dummies* para permitir sua incorporação nos modelos de aprendizado de máquina. Por exemplo, a variável "Sexo" foi transformada em "Sexo_Masculino" e "Sexo_Feminino", enquanto "Raça" foi desmembrada em categorias como "Raça_Branca", "Raça_Negra" e "Raça_Parda".

Para padronizar as variáveis numéricas, utilizamos a transformação *Z-score*, conforme a fórmula:

$$Z(x_{ij}) = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$$

em que $Z(x_{ij})$ é o valor padronizado (ou *Z-score*) do elemento $x_{ij}$, que representa o quão distante (em termos de desvios padrão) está o valor $x_{ij}$ da média $\bar{x}_j$, $x_{ij}$ é o valor original do dado que está sendo padronizado, pertencente à observação i-ésima da variável j, $\bar{x}_j$ é a média dos valores do atributo j (ou seja, a média de todos os valores para a variável j no conjunto de dados) e $\sigma_j$ é o desvio padrão dos valores do atributo j, uma medida de dispersão que indica o quanto os valores do atributo j variam em relação à média $\bar{x}_j$.

Essa transformação permitiu a normalização dos dados, garantindo comparações mais consistentes e melhorando o desempenho dos modelos preditivos.

**Tratamento de valores ausentes**

Os dados ausentes foram classificados em três categorias:

a) *missing not at random* (MNAR): Ausência relacionada diretamente ao valor faltante, como ocorre em perguntas sensíveis;
b) *missing completely at random* (MCAR): Ausência totalmente aleatória, sem relação com outras variáveis;
c) *missing at random* (MAR): Ausência dependente de outras variáveis, mas não da variável faltante em si.

Para os dados classificados como MNAR e MCAR, utilizamos o método *K-NN Imputation* do pacote "VIM" no software R, considerando os cinco vizinhos mais próximos. Já

as variáveis MAR foram removidas caso apresentassem mais de 50% de dados ausentes, resultando na exclusão de 79 variáveis.

**Balanceamento do conjunto de dados**

Para lidar com o desbalanceamento dos dados, utilizamos a técnica *Synthetic Minority Over-sampling Technique* (*SMOTE*). Implementado pelo pacote *DMwR* no R, o *SMOTE* gera exemplos sintéticos para a classe minoritária a partir da interpolação entre os atributos dos cinco vizinhos mais próximos. Diferentemente do *down-sampling*, que pode reduzir a precisão dos modelos, o *SMOTE* permite uma representação mais equitativa das classes, favorecendo a robustez do modelo e garantindo previsões mais equilibradas.

As etapas descritas asseguram uma preparação criteriosa dos dados, permitindo a construção de modelos preditivos mais eficientes e confiáveis. A combinação de técnicas de seleção de variáveis, normalização, imputação de dados faltantes e balanceamento contribuiu para a melhoria da acurácia dos algoritmos testados.

**Construção, ajuste e avaliação dos modelos**

Nesta etapa, discutimos o treinamento e a avaliação de cinco modelos de aprendizado de máquina: Árvore de Decisão (AD), Floresta Aleatória (FA), *Naive Bayes* (NB), Redes Neurais Artificiais (RNA) e *Extreme Gradient Boosting* (XGBoost). Estes modelos foram selecionados devido à sua eficácia em tarefas de classificação e suas características distintas.

Cada modelo foi treinado utilizando as arquiteturas padrão do pacote *caret* (Kuhn, *et al.*, 2020). A Árvore de Decisão, por exemplo, adota o critério de Gini como função de impureza, permitindo a construção de árvores sem um limite pré-definido para a profundidade, o que ajusta automaticamente sua estrutura conforme os dados (Merkle e Shaffer, 2011).

Para a Floresta Aleatória, três parâmetros principais foram definidos: o número de árvores (*ntree*), o número mínimo de dados por nó terminal (*nodesize*) e o número de variáveis consideradas nas divisões de cada nó (*mtry*) (Liaw e Wiener, 2002). No sistema, o padrão de *ntree* é 500 árvores, *nodesize* foi configurado para 1, adequado para tarefas de classificação, e *mtry* corresponde à raiz quadrada do número total de variáveis preditoras (Liaw e Wiener, 2002; Carvalho et al., 2016).

O modelo *Naive Bayes* foi ajustado com suavização de Laplace, o que evita a atribuição de probabilidades zero a classes ausentes e ajusta probabilidades de variáveis raras ou ausentes (Sammut e Webb, 2011).

No modelo de Redes Neurais Artificiais, a camada de entrada possui um neurônio para cada variável preditora (82 na base de dados original e 20 após a aplicação de MIC e PCA). A camada oculta, com o número de neurônios ajustado automaticamente, é geralmente menor que a camada de entrada. Ambas utilizam a função de ativação sigmoide para mapear as entradas para valores entre 0 e 1. A camada de saída, composta por um único neurônio, também emprega a função sigmoide para prever a probabilidade de diabetes (classe 0 ou 1), sendo considerada positiva quando a probabilidade é superior a 0,5.

O treinamento do modelo utiliza o parâmetro *maxit* do pacote *caret*, que define o número máximo de iterações (ou épocas), com um padrão de 100 iterações. No entanto, o processo pode ser interrompido antes se o modelo atingir a convergência, ou seja, o ponto em que a função de perda estabiliza e não apresenta melhorias significativas (Fleck et al., 2016).

O modelo *XGBoost*, quando aplicado ao diagnóstico de diabetes, utiliza o método "xgbTree", que implementa o algoritmo *XGBoost* baseado em árvores de decisão (Chen e Guestrin, 2016). Durante o treinamento, o *caret* ajusta automaticamente os parâmetros do modelo, realizando uma busca em 10 diferentes combinações, incluindo parâmetros como a taxa de aprendizado ($\eta$) e a profundidade máxima das árvores (*max_depth*). Esse ajuste automático visa otimizar o desempenho do modelo, selecionando os melhores parâmetros a partir dos testes realizados, o que contribui para uma performance mais robusta na tarefa de diagnóstico (Chen e Guestrin, 2016).

A avaliação dos modelos foi conduzida utilizando métricas como acurácia, precisão, recall, F1-score e AUC-ROC, fornecendo uma visão completa do desempenho de cada modelo e facilitando comparações justas entre eles. Essas métricas permitiram uma análise detalhada, considerando tanto a capacidade de classificação correta quanto a habilidade de minimizar falsos positivos e falsos negativos.

Para garantir a reprodutibilidade dos resultados, foi definida uma semente (123) para controlar a aleatoriedade dos processos. O conjunto de dados foi dividido em 80% para treinamento e 20% para teste, utilizando o pacote *caret* (Kuhn, 2008). O treinamento dos modelos incluiu validação cruzada com 10 *folds* repetido cinco vezes, método padrão recomendado por Witten, Frank e Mark (2011) para uma avaliação confiável da taxa de erro dos modelos.

O procedimento utilizado neste trabalho pode ser visualizado no *pipeline* do Quadro 2, permitiu a otimização da seleção de hiperparâmetros, testando múltiplas combinações. A acurácia foi definida como a métrica de avaliação, com a exploração de 10 combinações para identificar a configuração ideal. Após o treinamento, a melhor configuração foi validada no conjunto de teste, com a avaliação conduzida por meio da matriz de confusão e outras métricas fornecidas pelo *caret*.

Quadro 2- Esquema do Pipeline de Treinamento e Avaliação de Modelos.

***Entrada:*** O conjunto de dados de treinamento $\boldsymbol{X}\_treinamento$ e a variável alvo $\boldsymbol{Y}_$treinamento

***Saida:*** Modelo treinado $\boldsymbol{M}\_Algorimo$ e previsões $\boldsymbol{Y}\_predição$

***Inicio:***

1. C – trainControl (method='repeatedcv', number = 10, repeats = 5)
2. $\boldsymbol{M}\_Algoritmo$ – train (Y_treinamento ~ X_treinamento, method =" Algoritmo", preProcess = c ("center", "scale"), trControl = C)
3. $\boldsymbol{Y}\_predição$ – predict (M_*Algoritmo*, $\boldsymbol{X}\_teste$)
4. CM – confusionMatrix($\boldsymbol{Y}\_predição$, $\boldsymbol{Y}\_teste$)
5. Metrics – {Acurácia, Precisão, Sensibilidade, Medida F1, Especificidade*}*

***Fim***

Fonte: O autor (2025).

Foram implementados algoritmos de aprendizado de máquina amplamente reconhecidos por sua eficácia na previsão do Diabetes (Zia e Khan, 2017; Sarwar et al., 2018). A escolha do modelo final foi baseada nas métricas de validação, conforme descrito por (Morettin e Singer, 2021). A aplicação da validação cruzada *k-fold* garantiu a confiabilidade dos resultados.

Por fim, a matriz de confusão foi utilizada para calcular métricas essenciais como acurácia, sensibilidade, especificidade e a área sob a curva ROC, fundamentais para avaliar a eficácia dos modelos de aprendizado de máquina (Junior et al., 2022).

**Análise de desempenho dos modelos**

Neste estudo, os modelos foram avaliados tanto em dados balanceados quanto desbalanceados para investigar como a estrutura dos dados impacta a eficácia preditiva. Foram utilizadas todas as variáveis disponíveis (abordagem Geral) e técnicas de redução de dimensionalidade, como o *Maximal Information Criterion* (MIC) e a Análise de Componentes Principais (PCA), para avaliar o impacto das variáveis relevantes na identificação de pacientes com diabetes.

O Quadro 3 apresenta as principais métricas de desempenho dos modelos, incluindo acurácia, sensibilidade, especificidade, F1-score e AUC. A sensibilidade foi destacada pela sua importância na identificação de pacientes diabéticos, pois minimizar falsos negativos é crucial para evitar atrasos no tratamento e a progressão da doença.

Quadro 3 - Principais métricas e seu impacto na predição do Diabetes.

| Métricas | Descrição | Impacto na predição do Diabetes |
|---|---|---|
| Acurácia | Proporção de previsões corretas em relação ao total de previsões realizadas. | Embora importante, em problemas desbalanceados, pode ser enganosa, já que modelos podem ter alta acurácia simplesmente por identificar corretamente a classe majoritária. |
| Sensibilidade | Proporção de casos positivos (diabéticos) corretamente identificados. | Crucial no diagnóstico do Diabetes, pois minimiza o risco de falsos negativos, identificando corretamente os pacientes com a doença. |
| Especificidade | Proporção de casos negativos (não diabéticos) corretamente identificados. | Embora importante para evitar alarmes falsos, em contextos médicos, a sensibilidade tende a ser mais crítica para detectar doenças. |
| F1-score | Média harmônica entre precisão e sensibilidade. | Oferece uma visão mais equilibrada do modelo, sendo importante quando se busca um equilíbrio entre a detecção de diabéticos e a minimização de falsos positivos. |
| AUC (Área sob a Curva ROC) | Medida da capacidade de discriminação do modelo entre as classes. | Importante para avaliar a capacidade do modelo em distinguir entre diabéticos e não diabéticos, com foco no equilíbrio entre sensibilidade e especificidade. |

Fonte: O autor (2025).

A curva ROC foi utilizada para analisar o equilíbrio entre sensibilidade e especificidade, permitindo identificar os modelos mais eficazes na discriminação entre diabéticos e não

diabéticos. A ênfase na redução de falsos negativos reforça a necessidade de modelos que garantam maior segurança na detecção da doença.

Os modelos selecionados para a predição do Diabetes incluem abordagens baseadas em árvores, métodos probabilísticos, redes neurais e algoritmos de *boosting*. As Árvores de Decisão foram escolhidas pela sua simplicidade e interpretabilidade, enquanto as Florestas Aleatórias, ao combinar múltiplas árvores, aumentam a robustez e mitigam o *overfitting*. O *Naive Bayes* se destaca pela rapidez e eficiência em grandes conjuntos de dados, enquanto as Redes Neurais capturam relações complexas e não lineares. O *XGBoost* com sua alta performance e mecanismos de regularização, lida bem com dados desbalanceados. Essa diversidade de técnicas permite uma comparação abrangente entre diferentes abordagens preditivas.

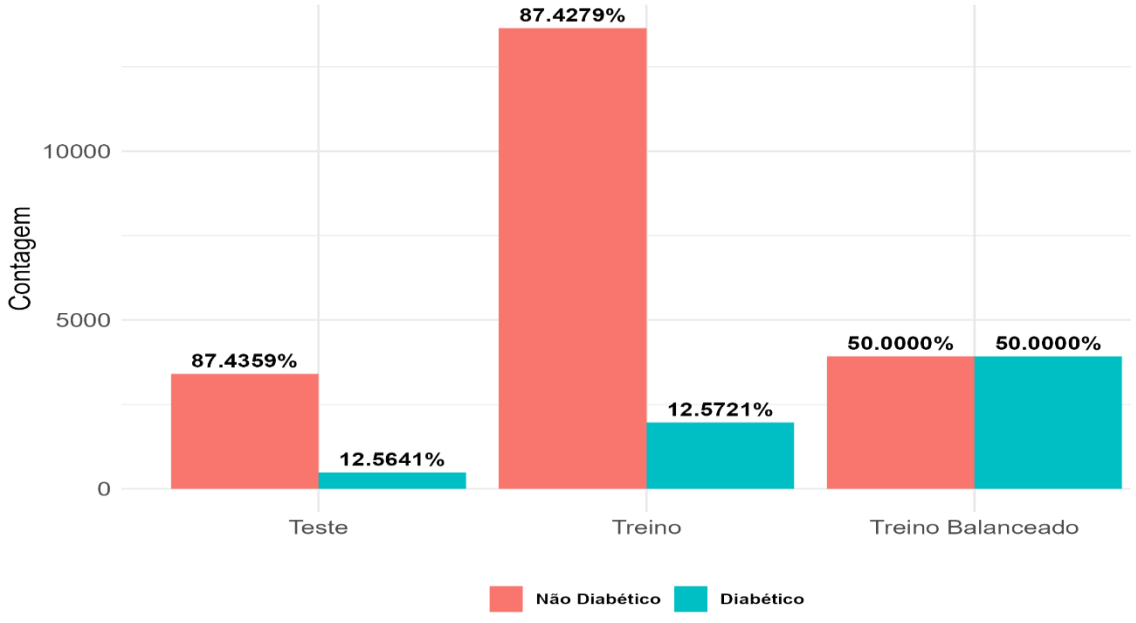
# 4 RESULTADOS E DISCUSSÃO

Nesta seção são apresentados os resultados das análises realizadas, com ênfase no desempenho dos modelos de aprendizado de máquina na predição do Diabetes, tanto em conjuntos de dados desbalanceados quanto balanceados. Além disso, discutem-se os principais resultados e suas implicações.

## 4.1 Descrição do banco de dados

Após o pré-processamento dos dados, o número de variáveis foi reduzido de 233(229 explicativas e 1 resposta) para 154, sendo 116 variáveis categóricas e 38 numéricas, após a remoção das variáveis com mais de 50% de valores ausentes. Com a aplicação do V de Cramér para as variáveis categóricas e do Coeficiente de Pearson para as variáveis numéricas, o conjunto final de dados foi reduzido para 70 variáveis categóricas e 13 variáveis numéricas. O conjunto de dados final manteve 82 variáveis explicativas e 19506 observações. O conjunto de treinamento foi balanceado de 15606 para 7848 observações, distribuídas igualmente entre diabéticos e não diabéticos, enquanto o conjunto de teste permaneceu com 3900 observações, de forma desbalanceada, conforme ilustrado na Figura 13.

Figura 13 - Distribuição dos casos de pacientes diabéticos e não diabéticos.




Fonte: O autor (2025).

## 4.2 Dados desbalanceados

Com os dados desbalanceados, o conjunto de treinamento manteve a distribuição natural das classes, caracterizada pela predominância da classe majoritária (não diabéticos). Nesse cenário, foram avaliadas as abordagens Geral, MIC e PCA para analisar como essa desigualdade afeta a identificação de pacientes com Diabetes.

## 4.2 Geral

A Tabela 1 apresenta as métricas de desempenho dos modelos quando todas as variáveis foram utilizadas, considerando o conjunto de treinamento desbalanceado. A análise destaca como cada modelo lida com o desbalanceamento das classes, com ênfase na identificação da classe minoritária (diabéticos).

Tabela 1 - Desempenho dos modelos com todas as variáveis no conjunto desbalanceado.

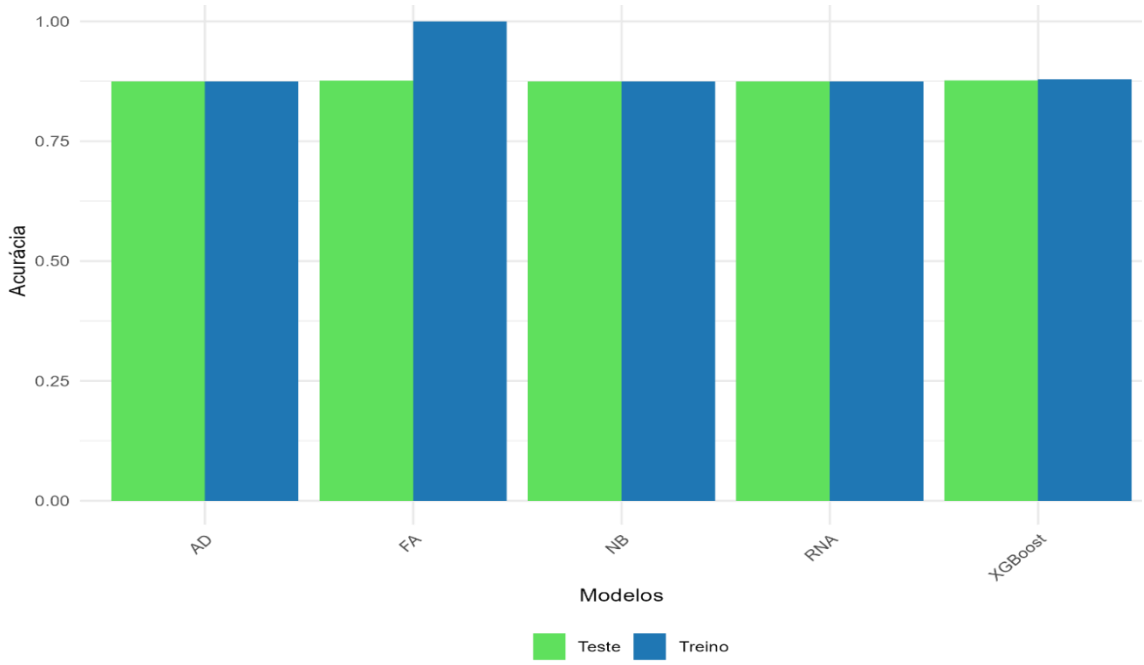
| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,8743 | 0,0000 | 0,0000 | - | 1,0000 |
| FA | 0,8758 | 0,6364 | 0,0285 | 0,0547 | 0,9976 |
| NB | 0,8743 | 0,0000 | 0,000 | - | 1,0000 |
| RNA | 0,8743 | 0,0000 | 0,0000 | - | 1,0000 |
| XGBoost | 0,8769 | 0,6041 | 0,0592 | 0,1078 | 0,9944 |

Fonte: O autor (2025).

Esses indicadores evidenciam a dificuldade dos modelos em lidar com o desbalanceamento das classes, especialmente na identificação da classe minoritária (diabéticos) (Ariza, *et al.*, 2022). Apesar da maior precisão ser apresentada pela Floresta Aleatória (0,6364), sua baixa sensibilidade (0,0285) resultou em um F1-score de apenas 0,0547, indicando desempenho limitado. O *XGBoost* destacou-se com o melhor desempenho geral, apresentando um F1-score de 0,1078, superior aos demais. Contudo, todos os modelos demonstraram alta especificidade, com valores acima de 0,99, refletindo bom desempenho na identificação de casos negativos que é a classe majoritária.

A comparação das acurácias entre os conjuntos de treino e teste permite avaliar o potencial de generalização dos modelos. A Figura 14 apresenta essa comparação para os modelos treinados com todas as variáveis, evidenciando possíveis padrões de *overfitting* ou *underfitting*.

Figura 14 - Comparação de acurácia de treino e teste usando todas variáveis no conjunto desbalanceado.

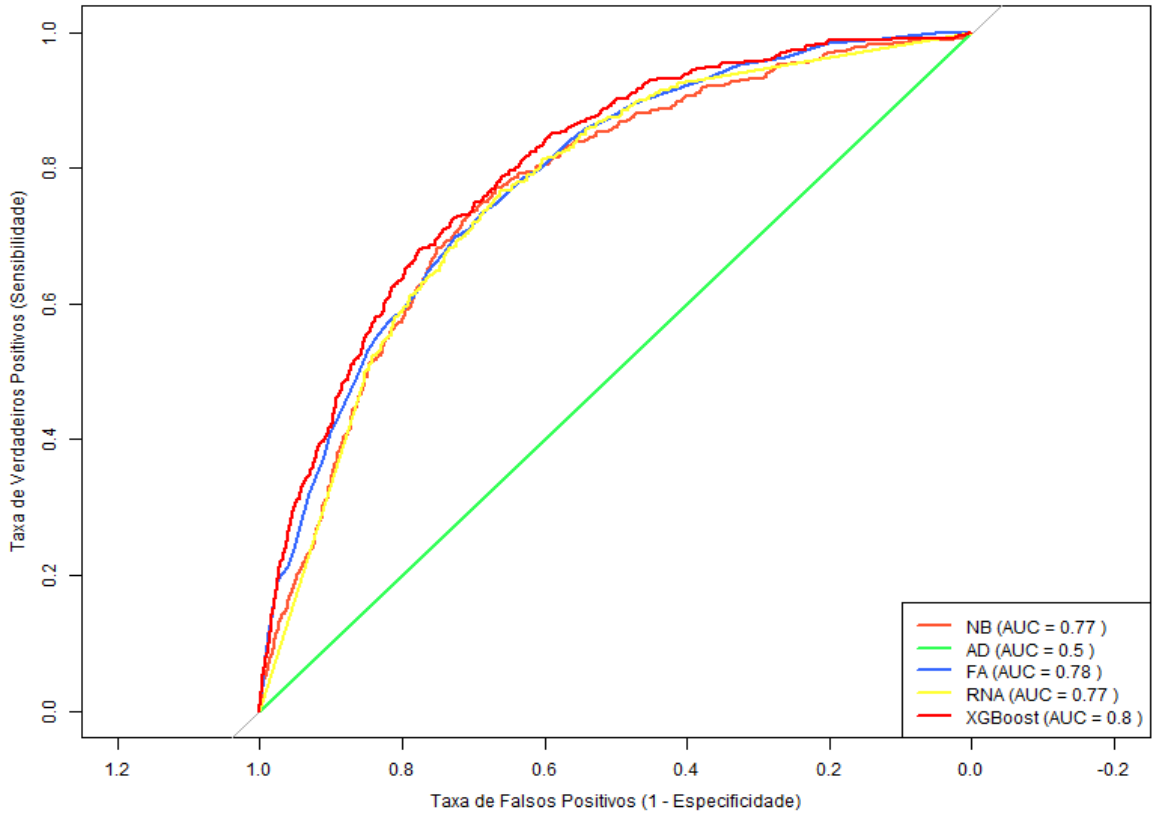



Fonte: O autor (2025).

Os modelos Árvores de Decisão, *Naive Bayes* e Redes Neurais Artificiais apresentaram acurácias idênticas no treino (0,8742) e no teste (0,8743), indicando baixa capacidade de generalização e dificuldade em capturar padrões complexos. O *XGBoost* obteve a melhor acurácia no treino (0,8797) e manteve um desempenho competitivo no teste (0,8769), demonstrando boa generalização. Por outro lado, a Floresta Aleatória, com acurácia perfeita no treino (1), apresentou queda no teste (0,8758), sugerindo sobreajuste (*overfitting*).

Por fim a Figura 15 apresenta as curvas ROC dos modelos treinados com todas as variáveis, possibilitando uma análise mais detalhada do desempenho de cada abordagem na separação das classes. A área sob a curva (AUC) é um indicador essencial para avaliar a capacidade discriminativa dos modelos, sendo especialmente relevante em cenários de desbalanceamento.

Figura 15 - Curva ROC para os modelos usando todas variáveis no conjunto desbalanceado.




Fonte: O autor (2025).

Entre os modelos avaliados, o *XGBoost* apresentou o melhor desempenho, com uma área sob a curva (AUC) de 80%. Isso significa que o modelo conseguiu distinguir corretamente entre indivíduos com e sem Diabetes em 80% das situações, mostrando o melhor equilíbrio entre sensibilidade (capacidade de identificar casos positivos) e especificidade (capacidade de evitar alarmes falsos).

A Floresta Aleatória obteve uma AUC de 78% indicando um desempenho ligeiramente inferior ao *XGBoost*, mas ainda satisfatório para discriminar entre classes. Os modelos *Naive Bayes* e Redes Neurais Artificiais alcançaram desempenhos similares, com AUC de 77%, refletindo uma capacidade razoável de separação entre casos positivos e negativos, embora com limitações práticas relacionadas à sensibilidade e ao equilíbrio geral.

Em contraste, a Árvore de Decisão apresentou o pior desempenho, com uma AUC de 50%. Esse resultado equivale a uma classificação aleatória, sugerindo que o modelo não foi capaz de capturar padrões significativos nos dados, especialmente em um conjunto desbalanceado.

As curvas ROC mostram que modelos mais complexos, como o *XGBoost* e a Floresta Aleatória, possuem maior capacidade de identificar padrões nos dados, alinhando-se aos achados de Kumar et al. (2020). Esses algoritmos demonstram superioridade na superação dos desafios associados ao desbalanceamento de classes. Os resultados destacam a relevância do uso de técnicas avançadas para aprimorar a precisão diagnóstica, especialmente em aplicações críticas, como o diagnóstico do Diabetes.

### 4.2.2 MIC

A Tabela 2 apresenta as métricas de desempenho dos modelos usando as 20 variáveis selecionadas pelo MIC, com foco na redução da dimensionalidade e preservação das variáveis mais informativas para a predição do Diabetes.

Tabela 2 – Desempenho dos modelos com variáveis selecionadas pelo MIC no conjunto desbalanceado.

| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,8743 | 0,0000 | 0,0000 | - | 1,0000 |
| RF | 0,8726 | 0,3793 | 0,0224 | 0,0424 | 0,9947 |
| NB | 0,8743 | 0,0000 | 0,0000 | - | 1,0000 |
| RNA | 0,8741 | 0,4706 | 0,0163 | 0,0316 | 0,9973 |
| XGBoost | 0,8753 | 0,5555 | 0,0408 | 0,076 | 0,9953 |

Fonte: O autor (2025).

Os resultados evidenciam que o desbalanceamento das classes influenciou diretamente o desempenho dos modelos, com impacto significativo nas métricas de sensibilidade e F1-score. Apesar de sua alta acurácia (0,8743), a Árvore de Decisão apresentou sensibilidade nula, indicando total incapacidade de prever casos positivos. Sua especificidade perfeita (1) reforça sua eficácia em identificar corretamente a classe negativa, mas isso ocorre às custas da classe minoritária. De forma semelhante, o *Naive Bayes* também apresentou sensibilidade nula e alta especificidade, o que o torna igualmente ineficaz para prever a classe positiva.

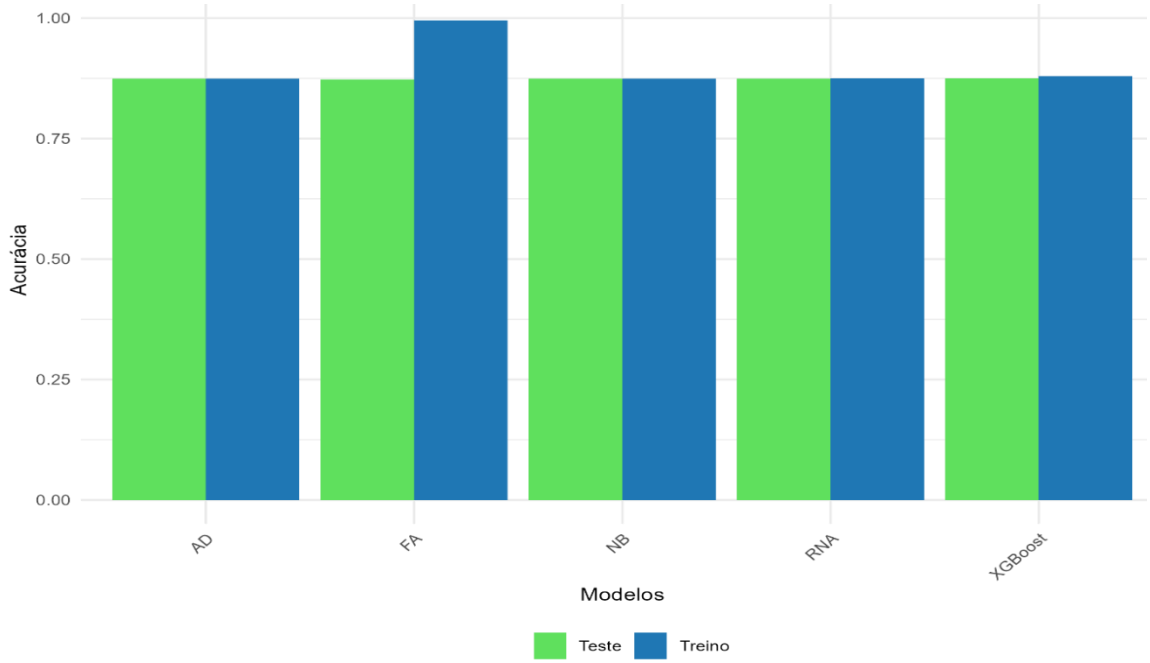
A Floresta Aleatória mostrou uma precisão de 0,3793 e especificidade elevada (0,9947), porém com uma sensibilidade extremamente baixa (0,0224), confirmando sua dificuldade em identificar casos da classe minoritária. Por sua vez, a Rede Neural Artificial trouxe uma leve

melhora, alcançando precisão de 0,4706 e sensibilidade de 0,0163. Ainda assim, esses valores permanecem insuficientes para aplicações práticas.

O modelo *XGBoost* destacou-se como o mais robusto entre os avaliados, obtendo sensibilidade de 0,0408, precisão de 0,5555 e F1-score de 0,076. Embora os resultados do *XGBoost* ainda sejam limitados, ele apresentou o melhor equilíbrio entre as métricas analisadas, evidenciando maior capacidade de identificar a classe minoritária em um cenário desbalanceado.

A Figura 16 compara as acurácias de treino e teste dos modelos treinados com as 20 variáveis selecionadas pelo MIC, permitindo avaliar a capacidade de generalização de cada modelo com o conjunto reduzido de variáveis.

Figura 16 - Comparação de acurácia de treino e teste usando MIC no conjunto desbalanceado.




Fonte: O autor (2025).

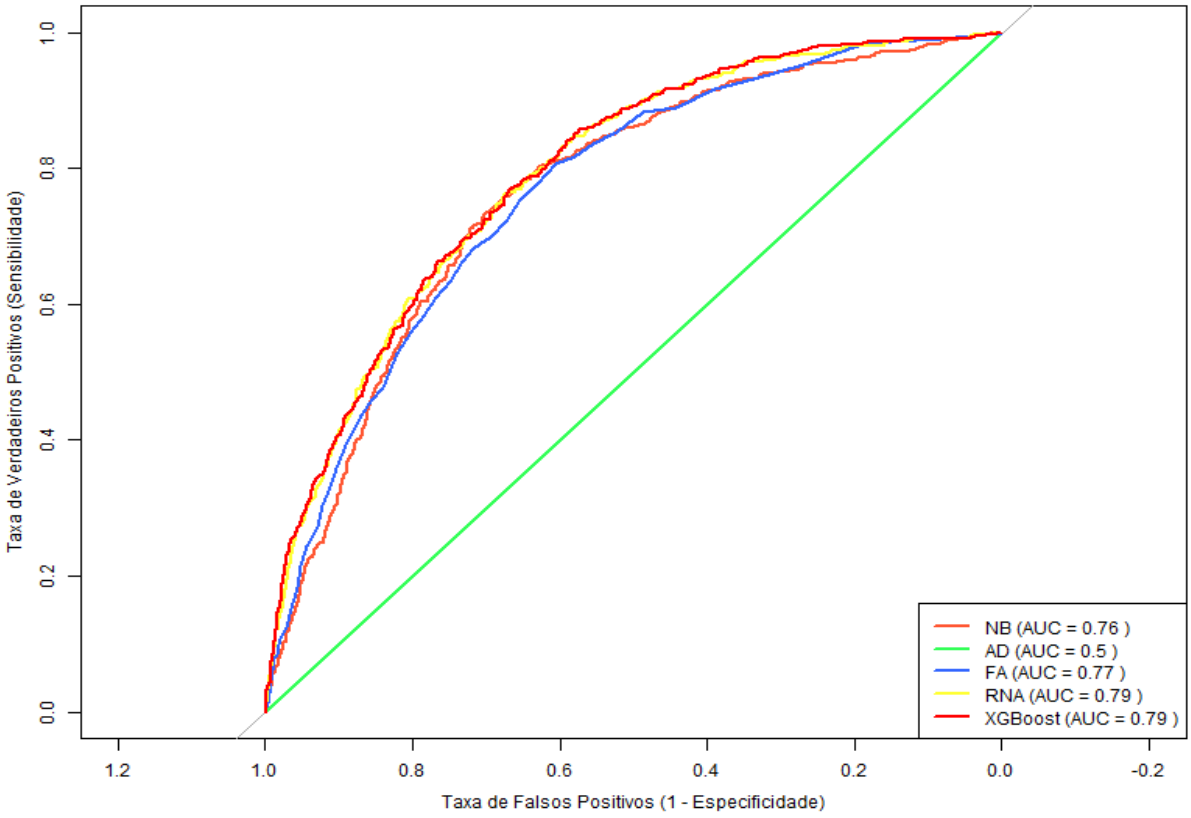
De maneira geral, os modelos apresentaram desempenhos consistentes entre os conjuntos de treino e teste, indicando ausência significativa de sobreajuste na maioria dos casos. O *XGBoost* destacou-se com as maiores acurácias, atingindo 0,8795 no treino e 0,8753 no teste, demonstrando um bom equilíbrio entre aprendizado e generalização.

Por outro lado, a Floresta Aleatória exibiu uma alta acurácia no treino (0,9954), mas sofreu uma queda considerável no teste (0,8726), sugerindo uma tendência ao sobreajuste. Em

contraste, os modelos Árvore de Decisão, *Naive Bayes* e Redes Neurais Artificiais apresentaram acurácias consistentes, próximas a 0,874 em ambos os conjuntos, o que reforça sua estabilidade, mas sem ganhos expressivos na capacidade de generalização para novos dados.

A Figura 17 apresenta as curvas ROC dos modelos avaliados com as 20 variáveis selecionadas pelo MIC, oferecendo uma análise detalhada da capacidade de cada modelo em distinguir as classes e proporcionando uma visão mais clara do equilíbrio entre sensibilidade e especificidade.

Figura 17 - Curva Roc para os modelos usando MIC no conjunto desbalanceado.




Fonte: O autor (2025).

Entre os modelos avaliados, o *Naive Bayes* apresentou uma AUC de 76%, indicando que, em 76% das vezes, o modelo consegue distinguir corretamente entre indivíduos com e sem Diabetes. No entanto, sua sensibilidade nula significa que o modelo falhou completamente em identificar casos positivos, tornando-o ineficaz para aplicações práticas.

A Árvore de Decisão demonstrou o pior desempenho, com uma AUC de 50%, o que equivale a uma classificação aleatória — como jogar uma moeda para decidir se um indivíduo tem ou não Diabetes.

Já a Floresta Aleatória alcançou uma AUC de 77%, mostrando que o modelo tem uma precisão razoável para separar casos positivos e negativos. Apesar disso, sua baixa sensibilidade indica que ele frequentemente deixa de identificar pacientes com diabetes, o que limita seu uso em triagens.

A Rede Neural Artificial destacou-se, com uma AUC de 79%, sugerindo que ela tem uma boa capacidade de distinguir entre pessoas com e sem Diabetes em quase 80% das situações. Contudo, seu baixo F1-score indica que a precisão e o equilíbrio entre falsos positivos e falsos negativos ainda não são satisfatórios para aplicações práticas.

Por fim, o modelo *XGBoost* apresentou uma AUC de 79% e a melhor sensibilidade, sendo essa última considerada uma das métricas mais importantes em estudos médicos, segundo Hicks et al. (2022). A sensibilidade é fundamental, pois o objetivo é minimizar ao máximo os erros na identificação de instâncias positivas, o que resulta em um alto *recall*. Esse aspecto é crucial em diagnósticos médicos, uma vez que um falso negativo – quando a doença está presente, mas não é detectada – pode ocasionar atrasos no tratamento e complicações graves, destacando a eficácia do modelo na distinção entre classes.

O *XGBoost* se mostrou particularmente eficaz em lidar com os desafios de discriminação, mesmo em um conjunto de dados desbalanceado, embora ainda existem desafios a serem superados em relação a outras métricas, como precisão e sensibilidade.

### 4.2.3 PCA

A avaliação de desempenho dos modelos após a redução de dimensionalidade pelo PCA está apresentada na Tabela 3, permitindo analisar o impacto dessa técnica no desempenho dos modelos com dados desbalanceados.

Tabela 3 – Desempenho dos modelos usando PCA no conjunto desbalanceado.

| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,8776 | 0,5670 | 0,1122 | 0,1873 | 0,9877 |
| RF | 0,8871 | 0,7192 | 0,1673 | 0,2715 | 0,9906 |
| NB | 0,8841 | 0,5913 | 0,2510 | 0,3524 | 0,9750 |
| RNA | 0,8989 | 0,7000 | 0,3428 | 0,4603 | 0,9789 |
| XGBoost | 0,8921 | 0,6949 | 0,2510 | 0,3688 | 0,9842 |

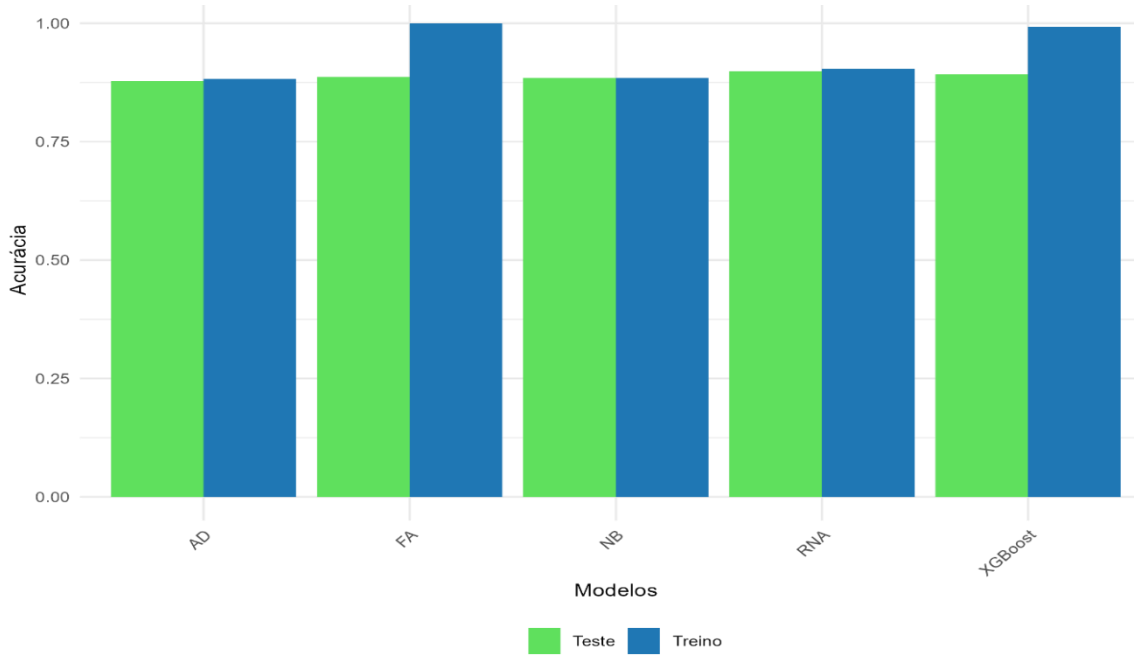
Fonte: O autor (2025).

Os resultados evidenciam o impacto da redução de dimensionalidade nos modelos, com destaque para as métricas de sensibilidade e especificidade. O modelo de Floresta Aleatória se destacou pela alta precisão de 0,7192 e especificidade de 0,9906, mas apresentou uma sensibilidade limitada de 0,1673, indicando dificuldades em prever a classe positiva de forma eficaz. O *XGBoost* embora com um desempenho ligeiramente superior, apresentou uma precisão de 0,6949 e sensibilidade de 0,251, mantendo o maior F1-score (0,3688) entre os modelos avaliados, mas ainda enfrentando desafios na identificação precisa da classe positiva.

Quanto aos demais modelos, a Árvore de Decisão obteve a maior acurácia (0,8776), porém com uma sensibilidade muito baixa (0,1122), refletindo sua ineficiência em identificar a classe positiva. O modelo *Naive Bayes* apresentou um F1-score de 0,3524 e sensibilidade de 0,251, sugerindo também dificuldades em prever adequadamente os casos positivos. Por outro lado, as Redes Neurais Artificiais, com sensibilidade de 0,3428 e F1-score de 0,4603, demonstraram um desempenho relativamente melhor, embora ainda com limitações na distinção clara da classe positiva.

A comparação das acurácias de treino e teste para os modelos está ilustrada na Figura 18, destacando as variações no desempenho dos modelos após a redução dimensional.

Figura 18 - Comparação de acurácia de treino e teste usando PCA no conjunto desbalanceado.




Fonte: O autor (2025).

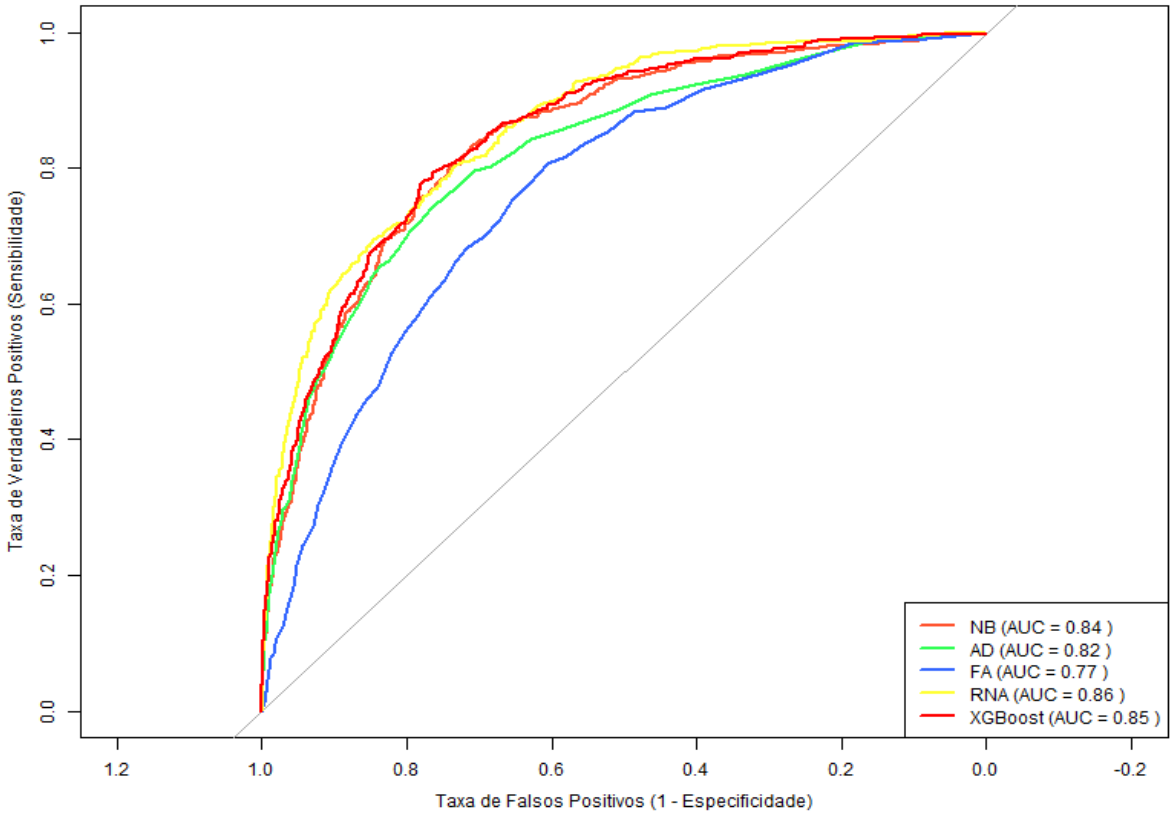
De forma geral, os modelos demonstraram boa capacidade de generalização, com diferenças mínimas entre as acurácias de treino e teste na maioria dos casos. A Rede Neural Artificial destacou-se como o modelo mais consistente, alcançando acurácia de 0,9036 no treino e 0,8989 no teste, o que sugere ausência de sobreajuste.

O *XGBoost* obteve alta acurácia no treino (0,9929) e manteve um desempenho competitivo no teste (0,8921), indicando um equilíbrio razoável entre aprendizado e generalização. Em contrapartida, a Floresta Aleatória apresentou um declínio mais acentuado, passando de 1,0 no treino para 0,8741 no teste, o que aponta para uma tendência ao sobreajuste.

Por fim, os modelos *Naive Bayes* e Árvore de Decisão exibiram acurácias consistentes, próximas de 0,88 em ambos os conjuntos de dados, reforçando sua estabilidade e ausência de sinais evidentes de sobreajuste.

A Figura 19 apresenta as curvas ROC, oferecendo uma análise detalhada da capacidade dos modelos em discriminar entre as classes, complementando a avaliação de desempenho.

Figura 19 - Curva Roc para os modelos usando PCA no conjunto desbalanceado.




Fonte: O autor (2025).

As análises das curvas ROC revelaram que as Redes Neurais Artificiais e o *XGBoost* apresentaram os melhores desempenhos, com áreas sob a curva (AUCs) de 86% e 85%, respectivamente. Esses valores indicam que ambos os modelos foram capazes de distinguir corretamente entre indivíduos com e sem diabetes em mais de 85% das situações, demonstrando excelente capacidade de discriminação entre as classes. O *Naive Bayes* alcançou um AUC de 84%, enquanto a Árvore de Decisão obteve 82%, mostrando desempenho consistente, mas inferior aos modelos mais avançados. Em contraste, a Floresta Aleatória apresentou o menor AUC, de 77%, refletindo limitações na separação das classes em comparação aos outros métodos avaliados.

### 4.2.4 Comparação entre Geral, MIC e PCA

Os resultados indicam que o desbalanceamento das classes comprometeu a identificação de pacientes com Diabetes, reduzindo a sensibilidade dos modelos. Mesmo com AUCs elevadas, modelos com alta especificidade tiveram dificuldades na detecção de casos positivos,

resultando em F1-scores reduzidos (He e Garcia, 2009; Japkowicz e Stephen, 2002). Técnicas de reamostragem, como *oversampling* e *undersampling*, são estratégias eficazes para mitigar esse efeito e melhorar o desempenho preditivo.

Entre as abordagens avaliadas, o *XGBoost* demonstrou o melhor desempenho na abordagem geral e também foi superior na abordagem MIC. No entanto, com a redução de dimensionalidade por PCA, a RNA apresentou resultados ligeiramente superiores ao *XGBoost*, destacando-se na sensibilidade e no F1-score. A aplicação do PCA aprimorou a capacidade de identificação da classe minoritária, tornando a RNA mais equilibrada e eficaz nesse cenário (Jolliffe, 2002).

## 4.3 Em dados balanceados

A aplicação de técnicas de balanceamento foi crucial para corrigir o desbalanceamento das classes, melhorando a identificação da classe minoritária (Ariza, *et al.*, 2022). Nesta seção, são analisados os desempenhos dos modelos nas abordagens Geral, MIC e PCA, após o balanceamento dos dados.

### 4.3.1 Geral

Com o balanceamento das classes, os modelos apresentaram uma melhoria substancial em seu desempenho, especialmente na sensibilidade, que antes era um desafio. Esse resultado está em consonância com os achados de (Da Silva Filho e Coutinho, 2022). A Tabela 4 oferece uma análise detalhada das métricas de desempenho dos cinco modelos de aprendizado de máquina, considerando todas as variáveis.

Tabela 4 – Desempenho dos modelos com todas as variáveis após balanceamento das classes.

| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,7210 | 0,2565 | 0,6428 | 0,3667 | 0,7323 |
| RF | 0,7528 | 0,2842 | 0,6367 | 0,3929 | 0,7695 |
| NB | 0,2894 | 0,1466 | 0,9653 | 0,2545 | 0,1923 |
| RNA | 0,6008 | 0,2111 | 0,7959 | 0,3337 | 0,5727 |
| XGBoost | 0,7889 | 0,3127 | 0,5673 | 0,4032 | 0,8208 |

Fonte: O autor (2025).

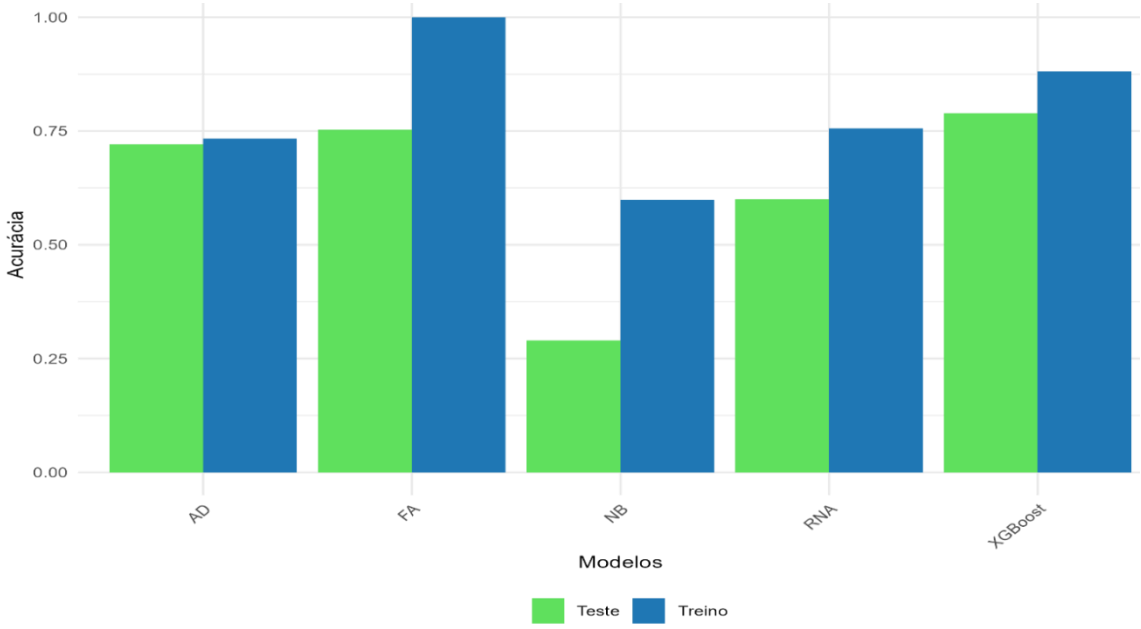
A Floresta Aleatória e o *XGBoos*t se destacaram como os modelos mais robustos nesta abordagem. Ambos apresentaram F1-scores elevados (0,3929 para a Floresta Aleatória e 0,4032 para o *XGBoost*), indicando um bom equilíbrio entre precisão e sensibilidade. A sensibilidade, que anteriormente era um ponto crítico devido ao desbalanceamento dos dados, melhorou significativamente, alcançando 0,6367 para a Floresta Aleatória e 0,5673 para o *XGBoost*, o que demonstra uma melhoria na identificação da classe minoritária. A especificidade também foi mantida em níveis elevados, com a Floresta Aleatória atingindo 0,7695 e o *XGBoost* 0,8208, refletindo sua capacidade de identificar corretamente os casos negativos.

Outros modelos, como as Redes Neurais Artificiais e o *Naive Bayes*, também apresentaram melhorias em relação aos dados desbalanceados. As Redes Neurais Artificiais tiveram um F1-score de 0,3337, enquanto o *Naive Bayes* obteve 0,2545, demonstrando que o balanceamento ajudou a melhorar a sensibilidade, embora com resultados ainda baixos.

A Árvore de Decisão, com sensibilidade de 0,6428 e F1-score de 0,3667, apresentou resultados mais equilibrados que o *Naive Bayes*, mas sua especificidade (0,7323) foi inferior à da Floresta Aleatória e do *XGBoost*.

A Figura 20 mostra a comparação das acurácias entre os conjuntos de treino e teste após o balanceamento dos dados, permitindo uma análise do desempenho dos modelos em ambos os conjuntos, agora com classes balanceadas.

Figura 20 - Comparação de acurácia de treino e teste usando todas variáveis após balanceamento das classes.



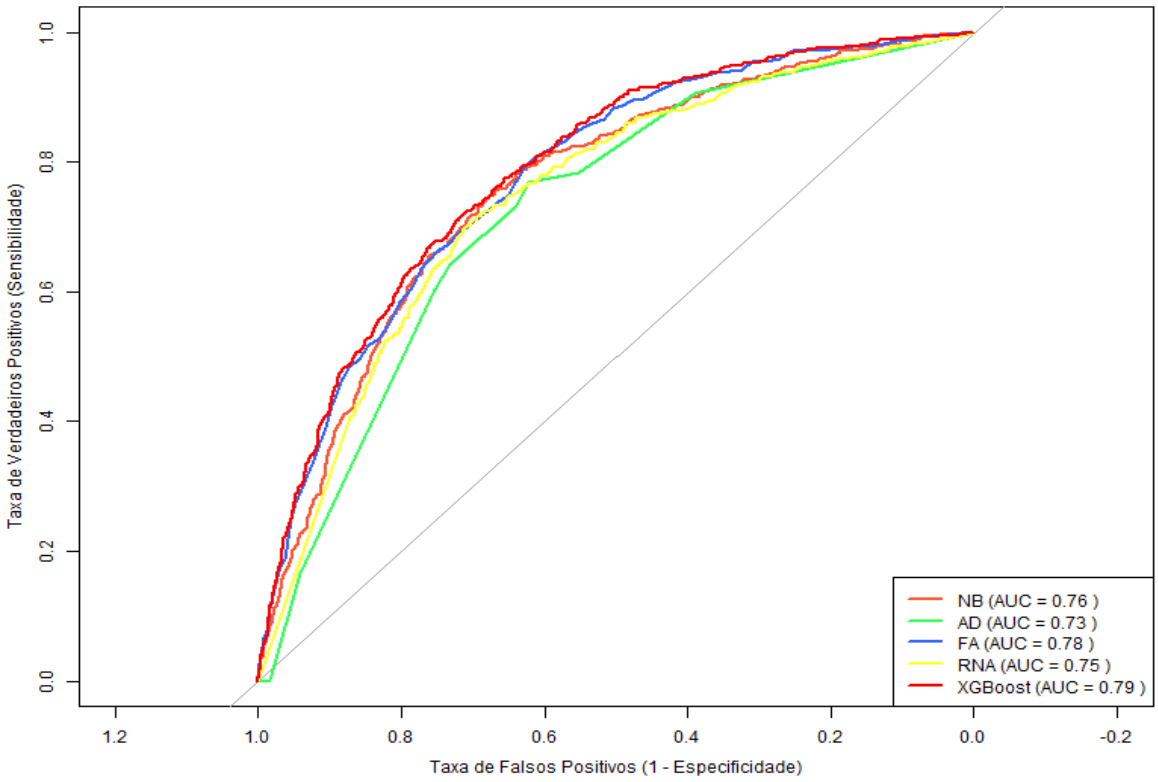

Fonte: O autor (2025).

Os resultados de acurácia indicam que o balanceamento das classes melhorou a generalização dos modelos, exceto para o *Naive Bayes*, que apresentou acurácia muito baixa em comparação com o conjunto desbalanceado. O *XGBoost* e a Árvore de Decisão tiveram pequenas diferenças entre as acurácias de treino e teste (0,8814 e 0,7889 para o *XGBoost*; 0,7332 e 0,721 para a Árvore de Decisão), sugerindo um bom equilíbrio entre aprendizado e generalização, especialmente para o *XGBoost*.

Por outro lado, a Floresta Aleatória apresentou a maior discrepância entre treino e teste (1,0 e 0,7528), indicando possível sobreajuste. Modelos como o *Naive Bayes* e as Redes Neurais Artificiais também mostraram variações significativas, sugerindo dificuldades de generalização.

A Figura 21 apresenta as curvas ROC e os valores de AUC dos modelos, permitindo uma análise mais detalhada de sua capacidade de discriminação entre as classes.

Figura 21 - Curva Roc para os modelos usando todas variáveis após balanceamento das classes.




Fonte: O autor (2025).

Os valores de AUC fornecem uma medida clara da capacidade dos modelos em distinguir entre indivíduos com e sem Diabetes. O *XGBoost* apresentou o maior AUC (0,79), indicando que o modelo é capaz de discriminar corretamente as classes em 79% das situações,

consolidando-se como o mais eficaz. A Floresta Aleatória obteve um AUC de 78%, um desempenho ligeiramente inferior, mas ainda confiável em termos de separação das classes.

A Rede Neural Artificial alcançou um AUC de 75%, o que significa que o modelo acertou na discriminação em 75% dos casos analisados, embora tenha apresentado menor eficácia em comparação aos algoritmos mais avançados.

O *Naive Bayes* com um AUC de 76%, demonstrou alta sensibilidade, sendo eficiente na identificação de casos positivos. Contudo, o desempenho geral foi comprometido pela falta de equilíbrio entre as classes, como evidenciado pelas métricas de precisão e F1-score, que apontam para uma taxa elevada de falsos positivos.

Por fim, a Árvore de Decisão, com um AUC de 73%, apresentou o menor desempenho. Esse resultado sugere que o modelo tem dificuldade em lidar com a complexidade dos dados, muitas vezes agindo de forma semelhante a uma classificação próxima ao acaso.

### 4.3.2 MIC

A seleção de variáveis com base no *Maximal Information Coefficient* (MIC) resultou em um impacto positivo no desempenho dos modelos. A Tabela 5 apresenta as métricas de desempenho para os modelos avaliados utilizando as 20 variáveis mais relevantes selecionadas por essa técnica.

Tabela 5 – Desempenho dos modelos com variáveis selecionadas pelo MIC após balanceamento.

| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,7210 | 0,2565 | 0,6428 | 0,3667 | 0,7323 |
| RF | 0,7559 | 0,2804 | 0,602 | 0,3826 | 0,7780 |
| NB | 0,1610 | 0,1295 | 0,9918 | 0,2290 | 0,0416 |
| RNA | 0,7231 | 0,2585 | 0,6448 | 0,3692 | 0,7343 |
| XGBoost | 0,7895 | 0,3087 | 0,5449 | 0,3941 | 0,8246 |

Fonte: O autor (2025).

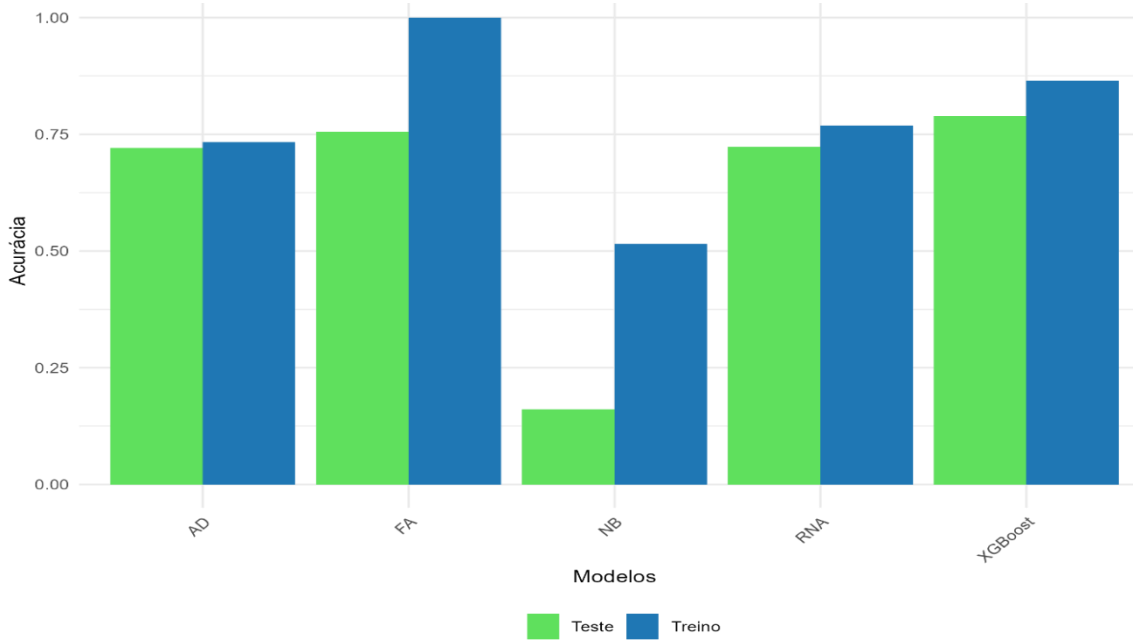
Os resultados indicam que o *XGBoost* foi o modelo de melhor desempenho, alcançando um F1-score de 0,3941 e uma especificidade de 0,8246, destacando-se na identificação da classe majoritária. A Floresta Aleatória obteve um F1-score de 0,3826 com uma sensibilidade de 0,602, o que também reflete um bom desempenho, embora com uma especificidade menor (0,778) em comparação ao *XGBoost*.

As Redes Neurais Artificiais apresentaram um F1-score de 0,3692 mostrando uma leve melhoria em relação a modelos mais simples, mas ainda inferior aos resultados do *XGBoost* e da Floresta Aleatória. O *Naive Bayes* com um F1-score de 0,229, teve o pior desempenho, especialmente na identificação da classe minoritária, sugerindo limitações substanciais neste modelo.

A Árvore de Decisão, com um F1-score de 0,3667, teve o pior desempenho entre os modelos avaliados, indicando dificuldades em identificar a classe minoritária, mesmo após o balanceamento.

A Figura 22 ilustra a comparação entre as acurácias obtidas nos conjuntos de treino e teste para os modelos, após a aplicação da seleção de variáveis baseada no MIC, evidenciando o impacto dessa abordagem na capacidade de generalização dos modelos.

Figura 22 - Comparação de acurácia de treino e teste usando MIC após balanceamento das classes.



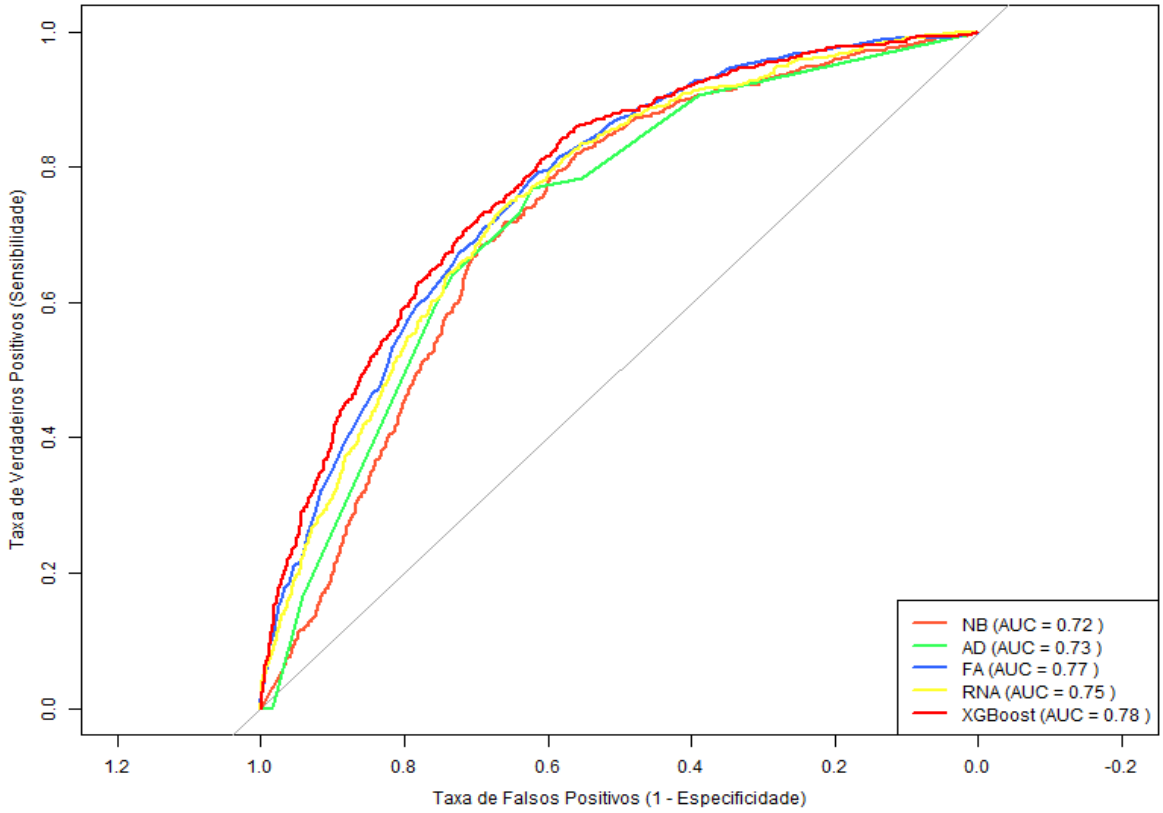

Fonte: O autor (2025).

A Figura destaca que, de modo geral, os modelos melhoraram a generalização após a seleção de variáveis por MIC. A Árvore de Decisão, as Redes Neurais Artificiais e o *XGBoost* apresentaram acurácias de treino e teste próximas, indicando que capturaram padrões relevantes sem sobreajuste. Em contraste, a Floresta Aleatória obteve acurácia perfeita no treino (1,0), mas teve uma leve queda no teste (0,7559), sugerindo um possível sobreajuste.

O *Naive Bayes* por sua vez, manteve uma discrepância considerável entre as acurácias de treino (0,5151) e teste (0,1610), evidenciando suas limitações em cenários mais complexos, mesmo após a seleção de variáveis.

A Figura 23 apresenta as curvas ROC e os valores de AUC dos modelos, permitindo uma análise detalhada de sua capacidade de discriminação entre as classes.

Figura 23 - Curva Roc para os modelos usando MIC após balanceamento das classes.




Fonte: O autor (2025).

Os valores de AUC destacam uma hierarquia clara no desempenho de discriminação dos modelos. O *XGBoost* alcançou o melhor desempenho, com uma AUC de 78%, indicando que, em 78% das situações, o modelo conseguiu distinguir corretamente entre indivíduos com e sem Diabetes. A Floresta Aleatória seguiu de perto, com uma AUC de 77%, também apresentando resultados confiáveis na separação das classes.

Esses resultados são consistentes com outras métricas analisadas, reforçando que o *XGBoost* foi o modelo mais eficaz na identificação tanto de casos positivos quanto negativos.

A Rede Neural Artificial e a Árvore de Decisão obtiveram AUCs intermediárias, de 75% e 73%, respectivamente. Esses valores sugerem que, embora sejam modelos funcionais, possuem limitações na identificação de padrões mais complexos presentes nos dados.

Por outro lado, o *Naive Bayes* apresentou o menor AUC, de 72%. Apesar de sua alta sensibilidade, o modelo demonstrou dificuldades em captar padrões robustos, comprometendo sua capacidade de separar eficazmente as classes.

### 4.3.3 PCA

A aplicação do PCA como método de redução de dimensionalidade, aliada ao balanceamento dos dados, foi avaliada em termos de impacto no desempenho dos modelos. Os resultados estão detalhados na Tabela 6.

Tabela 6 – Desempenho dos modelos com PCA após balanceamento das classes.

| Modelos | Acurácia | Precisão | Sensibilidade | F1-score | Especificidade |
|---|---|---|---|---|---|
| AD | 0,7259 | 0,2625 | 0,6531 | 0,3745 | 0,7364 |
| RF | 0,7526 | 0,2975 | 0,7122 | 0,4197 | 0,7584 |
| NB | 0,7758 | 0,307 | 0,6265 | 0,4121 | 0,7968 |
| RNA | 0,7771 | 0,3217 | 0,6979 | 0,4404 | 0,7886 |
| XGBoost | 0,7715 | 0,3155 | 0,7000 | 0,4350 | 0,7818 |

Fonte: O autor (2025).

As Redes Neurais Artificiais foram o modelo com o melhor desempenho, alcançando um F1-score de 0,4404, refletindo uma boa capacidade de equilibrar sensibilidade (0,6979) e especificidade (0,7886). Esse desempenho destaca a eficácia do modelo em identificar tanto a classe minoritária quanto a majoritária, mesmo após a aplicação de técnicas de redução dimensional e balanceamento dos dados.

O *XGBoost* obteve um F1-score de 0,435, ligeiramente abaixo das Redes Neurais Artificiais, mas ainda apresentou resultados robustos. Com sensibilidade de 0,7000 e especificidade de 0,7818, o modelo demonstrou boa capacidade de identificar corretamente tanto os casos positivos quanto negativos.

A Floresta Aleatória obteve um F1-score de 0,4197, com sensibilidade de 0,7122 e especificidade de 0,7584. Embora tenha ficado em terceiro lugar, o desempenho ainda foi muito bom, com um bom equilíbrio entre sensibilidade e especificidade.

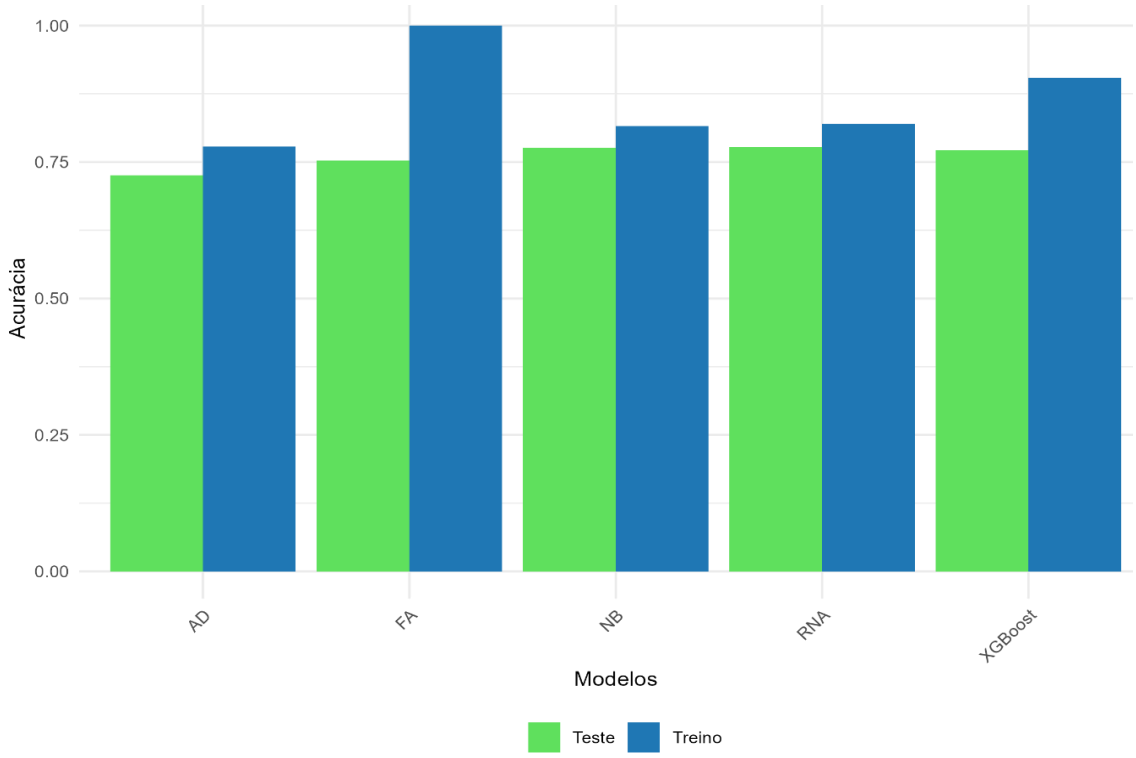
O *Naive Bayes* obteve um F1-score de 0,4121, com sensibilidade de 0,6265 e especificidade de 0,7968. Ainda que tenha ficado abaixo dos modelos mais avançados, o *Naive*

*Bayes* foi eficaz na identificação da classe minoritária, mantendo boa especificidade para a classe majoritária.

A Árvore de Decisão apresentou o desempenho mais modesto, com F1-score de 0,3745, sensibilidade de 0,6531 e especificidade de 0,7364. Apesar de algumas melhorias em relação aos dados desbalanceados, o modelo teve dificuldades em se ajustar ao balanceamento e à redução de dimensionalidade, sugerindo que abordagens mais sofisticadas são necessárias para alcançar melhores resultados.

A Figura 24 apresenta o gráfico comparativo das acurácias de treino e teste para os modelos, após a aplicação da PCA.

Figura 24 - Comparação de acurácia de treino e teste usando PCA após balanceamento das classes.



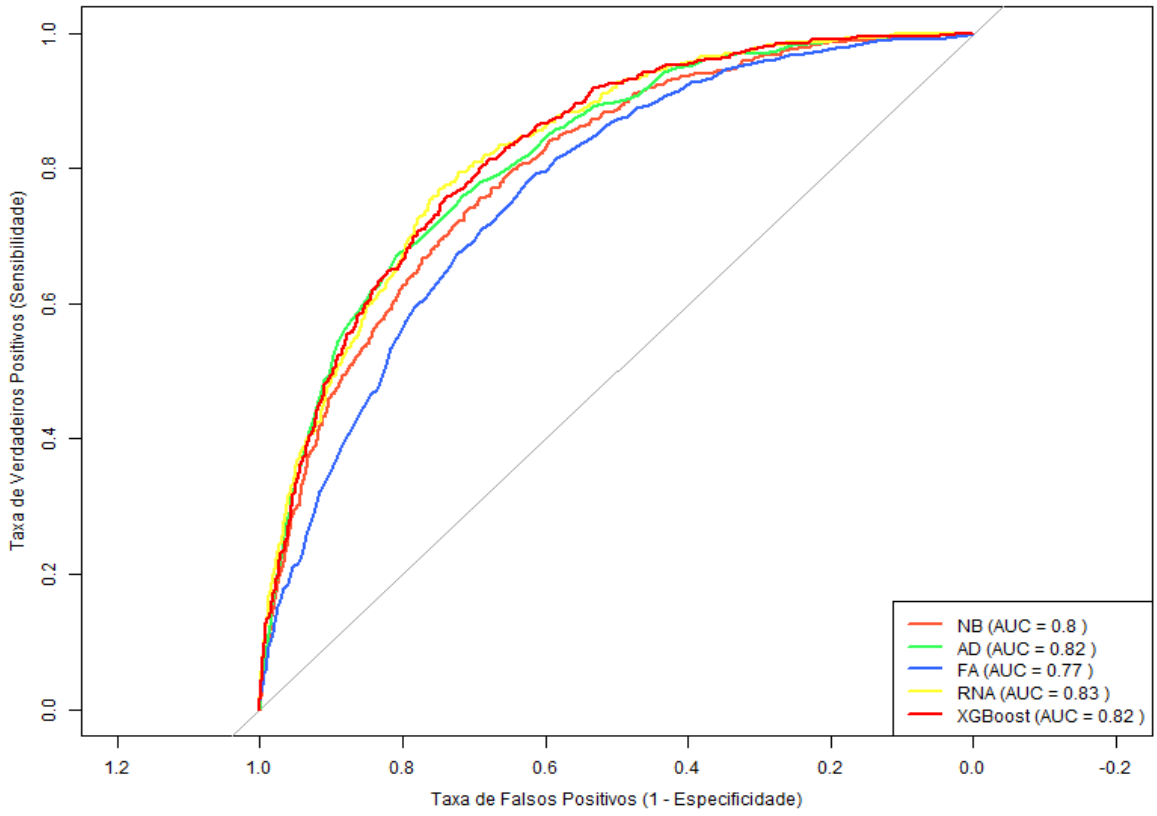

Fonte: O autor (2025).

Observa-se que com exceção da Floresta Aleatória, que apresentou acurácia perfeita no treino (1,0), os demais modelos mostraram um desempenho equilibrado entre treino e teste. O *XGBoost* teve um desempenho estável, com acurácia de 0,9043 no treino e 0,7715 no teste, indicando boa generalização, embora com indícios de sobreajuste. O *Naive Bayes* obteve 0,8156 no treino e 0,7758 no teste, demonstrando um bom desempenho geral.

As Redes Neurais Artificiais apresentaram acurácias de 0,8197 no treino e 0,7771 no teste, refletindo consistência entre as duas fases. Já a Árvore de Decisão, com 0,7783 no treino e 0,7259 no teste, teve um leve decréscimo na acurácia de teste, mas manteve um desempenho relativamente estável.

A Figura 25 exibe as curvas ROC, permitindo uma análise detalhada da capacidade de discriminação dos modelos.

Figura 25 - Curva Roc para os modelos usando PCA após balanceamento das classes.




Fonte: O autor (2025).

Os modelos com redução de dimensionalidade via PCA apresentaram AUCs elevadas, evidenciando excelente capacidade de discriminação entre as classes. As Redes Neurais Artificiais lideraram, com uma AUC de 83%, demonstrando o melhor equilíbrio entre as classes positiva e negativa. Árvore de Decisão e *XGBoost* empataram em segundo lugar, com AUC de 82%, destacando-se pelo desempenho competitivo e confiável.

O *Naive Bayes* obteve uma AUC de 80%, indicando boa discriminação entre as classes, mas com limitações em cenários mais complexos. A Floresta Aleatória alcançou uma AUC de

77%, representando o desempenho mais baixo entre os modelos avaliados, embora ainda satisfatório em termos de separação das classes.

### 4.3.4 Comparação entre Geral, MIC e PCA

Os resultados demonstram que técnicas de redução de dimensionalidade, como o *Maximum Information Coefficient* (MIC) e *Principal Component Analysis* (PCA), tiveram um papel relevante na simplificação da complexidade dos dados e na generalização dos modelos. Conforme discutido por Peng et al. (2005) e Jolliffe (2002), essas técnicas contribuíram para equilibrar a sensibilidade e a especificidade dos modelos, permitindo uma melhor diferenciação entre as classes.

A aplicação do PCA mostrou-se particularmente eficaz na otimização do desempenho dos modelos em diferentes configurações de dados. Tanto em conjuntos desbalanceados quanto balanceados, essa técnica favoreceu uma melhor identificação da classe minoritária, reforçando os achados descritos por Chawla et al. (2002) e Abnoosian, Farnoosh e Behzadi (2023). No entanto, essas abordagens não foram suficientes para mitigar completamente os impactos negativos do desbalanceamento extremo das classes, especialmente quando combinados com conjuntos de treinamento reduzidos. Esse efeito já havia sido destacado por Japkowicz e Stephen (2002), evidenciando que técnicas de redução de dimensionalidade, por si só, podem não solucionar integralmente os desafios impostos pelo desbalanceamento severo dos dados.

## 4.4 Comparação entre Balanceados e Desbalanceados

A estratégia de balanceamento das classes revelou um impacto notável na melhoria do desempenho geral dos modelos, alinhando-se com as previsões de Japkowicz e Stephen (2002). Este efeito foi particularmente significativo na detecção da classe minoritária, representada pelos pacientes com Diabetes, que, em cenários desbalanceados, eram quase indetectáveis pela maioria dos modelos. O balanceamento não só aumentou a acurácia global dos modelos, mas também favoreceu a identificação da classe de interesse, essencial para contextos clínicos.

A análise do desempenho dos modelos por meio das curvas ROC e dos valores de AUC confirmou a eficácia do balanceamento. O *XGBoost* em particular, destacou-se por sua robustez, alcançando elevados valores de AUC, mesmo em cenários desafiadores, como o desbalanceamento entre classes e a ausência de técnicas de redução de dimensionalidade. Esses

resultados reforçam a eficácia do *XGBoost,* corroborando o trabalho de Fawcett (2006), que destaca a importância das métricas de AUC para a avaliação de modelos. Além disso, a Floresta Aleatória mostrou-se superior aos modelos mais simples, como as Árvores de Decisão, especialmente em cenários desbalanceados, destacando a necessidade de técnicas mais sofisticadas para obter melhores resultados em condições adversas.

A importância de modelos complexos, como o *XGBoost* e a Floresta Aleatória, ficou evidente, especialmente quando o objetivo é garantir uma maior precisão na predição, o que é essencial em problemas de classificação com classes desbalanceadas. O balanceamento das classes, combinado com técnicas de pré-processamento, como a seleção e redução de variáveis, demonstrou ser fundamental para otimizar o desempenho dos modelos e lidar com os desafios impostos por dados desbalanceados. Em contextos clínicos, como a predição do Diabetes, é crucial priorizar a identificação correta da classe minoritária, mesmo que isso resulte em um aumento nos falsos positivos. Tal abordagem, conforme sugerido por Moreira, Soares et al. (2011), é estratégica para maximizar o diagnóstico precoce e reduzir os riscos de complicações irreversíveis, o que aprimora a eficácia das intervenções médicas.

# 5 CONCLUSÃO

Este estudo analisou modelos de aprendizado de máquina para a predição do Diabetes, abordando o desbalanceamento de classes e a grande quantidade de variáveis. Os resultados mostraram que o *XGBoost* obteve as melhores áreas sob a curva ROC (AUC), mas teve dificuldades em identificar a classe minoritária em cenários desbalanceados. Em contrapartida, a Rede Neural Artificial (RNA) com PCA apresentou um melhor equilíbrio entre sensibilidade e especificidade, minimizando falsos negativos, essencial em problemas clínicos como a predição do Diabetes.

O balanceamento de dados foi crucial para melhorar a sensibilidade dos modelos, destacando a importância de tratar a classe minoritária. A redução de dimensionalidade também contribuiu para equilibrar as métricas de desempenho. O *XGBoost* é recomendado quando a discriminação entre as classes é prioritária, enquanto a RNA com PCA é mais indicada para um equilíbrio entre sensibilidade e especificidade.

Este estudo sugere a coleta de novas amostras da classe minoritária para lidar com o desbalanceamento de dados, além de explorar modelos híbridos que combinem a robustez do *XGBoost* e o equilíbrio da RNA. As contribuições deste trabalho avançam o uso de aprendizado de máquina na predição do Diabetes, com potencial para melhorar a precisão diagnóstica e as intervenções médicas, impactando positivamente os resultados de saúde globalmente.

# REFERÊNCIAS


AGGARWAL, Charu C. **Data ClassifiCation Algorithms and Applications**. Boca Raton: CRC Press, 2015.

AJAYMEHTA. Harnessing Naïve Bayes Algorithm for Effective Text Analysis and Classification. **Medium**, 2023. Disponível em: <https://medium.com/@dancerworld60/demystifying-na%C3%AFve-bayes-simple-yet-powerful-for-text-classification-ad92b14a5c7>. Acesso em: 14 Dezembro 2023.

ALI, Zeravan A. et al. Exploring the Power of eXtreme Gradient Boosting Algorithm in Machine Learning: a Review. **Academic Journal of Nawroz University**, 12, 2023. 320-334.

AL-HASHEM, M. A.; ALQUDAH, A. M.; QANANWAH, Q. Performance evaluation of different machine learning classification algorithms for disease diagnosis. **International Journal of E-Health and Medical Communications (IJEHMC)**, 12, 2021. 1-28.

ANSELMO, Fernando. Machine Learning na Prática: Modelos em Python, v. 1.0, 2020.

ARIZA, Vinicius Matheus Pimentel et al. Uso do algoritmo"Floresta Aleatória" na identificação do comportamento da população na busca por serviços de saúde após o início da pandemia do novo coronavírus. **Novas práticas em informação e conhecimento**, 2022. 1-15.

BATISTA, André F. D. M.; FILHO, Alexandre D. P. C. Machine Learning aplicado à Saúde. **Sociedade Brasileira de Computação** , 2019.

BREIMAN, L., Friedman, J. H., Olshen, R. A., Stone, C. J. **Classification and Regression Trees**. Chapman and Hall/CRC, 1987.

BREIMAN, LEO. Random Forests. **Machine Learning 45**, 2001. 5–32.

BRASIL. Ministério da Saúde. **Vigitel Brasil 2023: Vigilância de fatores de risco e proteção para doenças crônicas por inquérito telefônico**. Brasília: Ministério da Saúde, 2023.

BRUTTI, Bruna et al. Diabete Mellitus: definição, diagnóstico, tratamento e mortalidade no Brasil, Rio Grande do Sul e Santa Maria, no período de 2010 a 2014. **Brazilian Journal of health Review**, 2, 2019. 3174-3182.

CASARIN, D. E. . D. G. et al. Diabetes mellitus: causas, tratamento e prevenção. **Brazilian Journal of Development**, 8, 2022. 10062-10075.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, 16, 2002, 321-357.

CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. San Francisco, CA: ACM, 2016, 785-794.

CRONE, Sven F.; FINLAY, Steven. Instance sampling in credit scoring: An empirical study of sample size and balancing. **International Journal of Forecasting**, 28, 2012. 224-238.

DA SILVA FILHO, F. R.; COUTINHO, E. F. Aprendizado de Máquina para Predição de Diagnósticos de Doenças Cardiovasculares. **In Anais do XXII Simpósio Brasileiro de Computação Aplicada à Saúde**, 2022. 358-369.

DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine Learning Research**, 7, 2006. 1-30.

DIAS, J. L. Aprendizado de Máquina Aplicado à Predição de Doenças Crônicas: Um Estudo de Caso de Hipertensão Arterial. **Dissertação - Universidade de São Paulo** (USP), 2024. Disponível em: <teses.usp.br>. Acesso em: 17 outubro 2025.

FACELI, K. et al. **Inteligência artificial:** uma abordagem de aprendizado de máquina. 2. ed. Rio de Janeiro: LTC, 2021.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From Data Mining to Knowledge Discovery in Databases. **AI Magazine**, 17, 1996. 37-54. Disponível em: https://doi.org/10.1609/aimag.v17i3.1230. Acesso em: 10 janeiro. 2025.

FAWCETT, Tom. An introduction to ROC analysis. **Pattern recognition letters**, 27, 861-874, 2006.

FEDERAÇÃO BRASILEIRA DAS ASSOCIAÇÕES DE GINECOLOGIA E OBSTETRÍCIA. Diabetes gestacional. **Femina**, 45, 2019. 11.

GÉRON, Aurélien. **Mãos à Obra Aprendizado de Máquina com Scikit-Learn & TensorFlow:** Conceitos, Ferramentas e Técnicas Para a Construção de Sistemas Inteligentes. 2. ed. São Paulo: Editora Alta Books, 2019.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. Burlington: Elsevier, 2011.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The elements of statistical learning:** data mining, inference, and prediction. 2. ed. New York: Springer, 2009.

HE, Haibo; GARCIA, Edwardo A. Learning from imbalanced data. **IEEE Transactions on knowledge and data engineering**, 21, 2009. 1263-1284.

HICKS, Steven A. et al. On evaluation metrics for medical applications of artificial intelligence. **Scientific Reports**, 12, 2022. 5979.

HOERL, Arthur E.; KENNARD, Robert W. Ridge regression: Biased estimation for nonorthogonal problems. **Technometrics**, 12, 1970. 55-67.

HYVÄRINEN, A.; OJA, E. Independent component analysis: algorithms and applications. **Neural networks**, 13, 2023. 411-430.

INAZAWA, Pedro et al. MACHINE LEARNING Desafios para um Brasil competitivo "Projeto Victor". **Revista da Sociedade Brasileira de Computação**, 2019. 19-24.

INTERNATIONAL DIABETES FEDERATION. **IDF Diabetes Atlas**. 10. ed. Brussels: International Diabetes Federation, 2021. Disponível em: <https://diabetesatlas.org/idfawp/resource-files/2021/07/IDF_Atlas_10th_Edition_2021.pdf/>. Acesso em: 11 Dezembro 2023.

IZBICKI, Rafael; MENDONÇA DOS SANTOS, Tiago. **Aprendizado de máquina: uma abordagem estatística**. 1. ed. São Carlos, SP: Rafael Izbicki, 2020. PDF. ISBN 978-65-00-00241-0.

JAPKOWICZ, Nathalie; STEPHEN, Shaju. The class imbalance problem: A systematic study. **Intelligent data analysis**, 6, 2002. 429-449.

JOLLIFFE, Ian T. **Principal Component Analysis**. 2. ed. New York: Springer, 2002.

JOLLIFFE, Ian T.; CADIMA, Jorge. Principal component analysis: a review and recent developments. **Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences**, 374, 2016. 20150202.

JUNIOR, G. B. V. et al. Métricas utilizadas para avaliar a eficiência de classificadores em algoritmos inteligentes. **Revista CPAQV–Centro de Pesquisas Avançadas em Qualidade de Vida| Vol**, 14, 2022. 2.

JÚNIOR, José R. F.; CORREIA, Natália S. C.; DE AZEVEDO MARQUES, Paulo M. Aprendizado de Máquina na Atenção à Saúde Humana. **Computação Brasil**, Porto Alegre, 39, n. 1, 2019. 37-40.

KINNEY, Justin B.; ATWAL, Gurinder S. Equitability, mutual information, and the maximal information coefficient. **Proceedings of the National Academy of Sciences**, 11, 2014. 3354-3359.

KOTSIANTIS, S. B. Supervised Machine Learning: A Review of Classification Techniques. **Informatica**, 31, 2007. 249-268.

KOWARIK, Alexander; TEMPL, Matthias. Imputation with the R Package VIM. **Journal of Statistical Software**, 74, 2016. 1-16.

KUHN, M. Building Predictive Models in R Using the caret Package. **Journal of Statistical Software**, 28, 2008. 1-26.

KUHN, M. et al. Package ‘caret’. **The R Journal**, v. 223, p. 48, 2020.

KUMAR, N. K. et al. Analysis and prediction of cardio vascular disease using machine learning classifiers. **International Conference on Advanced Computing and Communication Systems (ICACCS)**, 2020. 15-21.

KRUSKAL, W. H.; WALLIS, W. A. Use of ranks in one-criterion variance analysis. **Journal of the American Statistical Association**, 47, 1952. 583-621.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, 86, 1998. 2278-2324.

LIAW, Andy; WIENER, M. Classification and regression by randomforest R News, 2, 2002. 18-22. Disponível em: <http://CRAN.R-project.org/doc/Rnews/.>. Acesso em: 28 Maio 2024

LÓPEZ, Victoria et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. **Information Sciences**, 250, 2013, 113-141.

LUDERMIR, TERESA B. Inteligência Artificial e Aprendizado de Máquina :estado atual e tendências. **Estudos Avançados**, 35, 2021. 85-94.

MAIA, Beatriz. Tipos de Aprendizado de Máquina, 2020. Disponível em: <https://beatrizmaiads.medium.com/tipos-de-aprendizado-de-m%C3%A1quina-3-9a9052173bc4>.

MARASCHIN, J. D. F. et al. Classificação do diabete melito. **Arquivos Brasileiros de Cardiologia**, 95, 2010. 40 - 46.

METZ, CE. Basic principles of ROC analysis. **Seminars in nuclear medicine**, 8, 1978. 283-298.

MINISTÉRIO DA SAÚDE. **CADERNOS DE ATENÇÃO BÁSICA: DIABETES MELLITUS**. Secretaria de Atenção à Saúde; Departamento de Atenção Básica. Brasilia. 2006.

MIOT, Hélio A. Valores anômalos e dados faltantes em estudos clínicos e experimentais. **Jornal Vascular Brasileiro**, 18, 2019. e20190004.

MISSIO, F. M.; JACOBI, L. F. Variáveis dummy: especificações de modelos com parâmetros variáveis. **Ciência e Natura**, 2007. 111-135.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, 1, 2003. 32.

MONTEIRO, C. A.; MOURA, E. C.; CLARO, R. M.; CARDOSO, M. A. Validade de indicadores de atividade física e alimentação saudável obtidos por inquérito telefônico. **Revista de Saúde Pública**, 55, 2008.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. **Introduction to linear regression analysis**. 4. ed. Hoboken: Wiley, 2021.

MORETTIN, Pedro ; SINGER, Julio M. **Estatística e Ciência de Dados**. 2. ed. São Paulo: Editora LTC, 2021.

MUZY, J. et al. Prevalence of diabetes mellitus and its complications and characterization of healthcare gaps based on triangulation of studies. **Caderno de Saude**, 37, 2021. e00076120.

NEVES, R. G. et al. Complicações por diabetes mellitus no Brasil: estudo de base nacional, 2019. **Ciência & Saúde Coletiva**, 28, 2023. 3183-3190.

OLIVEIRA, Joanito de Andrade; DUTRA, Luciano V.; RENNÓ, Camilo D. Aplicação de Métodos de Extração e Seleção de Atributos para Classificação de Regiões. **XII SBSR**, 2005. 4201-4208.

PENG, Hanchuan; LONG, Fuhui; DING, Chris. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 27, n. 8, 2005, 1226-1238.

QUINLAN, J. Ross. **C4. 5: programs for machine learning**. Elsevier, 2014.

R CORE TEAM. R: A language and environment for statistical computing. Vienna: R Foundation for Statistical Computing, 2024. Disponível em: <https://www.R-project.org/>.

REIS, R. C. P. D. et al. REIS, Rodrigo Citton Padilha dos et al. Evolution of diabetes in Brazil: prevalence data from the 2013 and 2019 Brazilian National Health Survey. **Cadernos de Saúde Pública**, 38, 2022. e00149321.

RESHEF, David N. et al. Detecting novel associations in large data sets. **science**, 334, 2011. 1518-1524.

RODACKI, Melanie; TELES, Milena ; GABBAY, Monica. Classificação do diabetes. **Diretriz Oficial da Sociedade Brasileira de Diabetes**, 2023.

ROKACH, Lior; MAIMON, Oded. Decision trees. **Data mining and knowledge discovery handbook**, 2005. 165-192.

RUFINO, Hugo L. P.; VEIGA, Antônio C. P.; NAKAMOTO, Paula T. SMOTE_EASY: UM ALGORITMO PARA TRATAR O PROBLEMA DE CLASSIFICAÇÃO EM BASES DE DADOS REAIS. **JISTEM**, 13, 2016. 61-80.

SADEQ, Abdellatif M. **Machine Learning Mastery for Engineers**. eBook Kindle , 2024.

SARWAR, Muhammad A. et al. Prediction of Diabetes Using Machine Learning Algorithms in Healthcare. **24th international conference on automation and computing (ICAC)**, 2018. 1-6.

SHARMA, Siddharth; SHARMA, Simone ; ATHAIYA, Anidhya. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. **International Journal of Engineering Applied Sciences and Technology**, 4, 2020. 310-316.

SHUMWAY, R. H.; STOFFER, D. S. **Time series analysis and its applications:** with R examples. 4. ed. New York: Springer, 2017.

SICSÚ, ABRAHAM L.; SAMARTINI, ANDRÉ; BARTH, NELSON L. **TÉCNICAS DE MACHINE LEARNING**. 2. ed. São Paulo: Editora Blucher, 2023.

SILVA, Adilane R. D. Uma visão geral sobre machine learning – Classificação. **StatPlace**, 2020.

SILVA, Saulo R. E.; SCHIMIDT, Fernando. Silva, Saulo Rodrigues E., and Fernando Schimidt. "Redução de variáveis de entrada de redes neurais artificiais a partir de dados de análise de componentes principais na modelagem de oxigênio dissolvido. **Química Nova**, 39, 2016. 273-278.

SILVA, L. F. M. Análise preditiva baseada em inteligência artificial: um caminho para a transformação do modelo de vigilância das doenças crônicas não transmissíveis. **Dissertação-Universidade Federal do Pampa** (UNIPAMPA), 2023. Disponível em: <repositorio.unipampa.edu.br>. Acesso em: 10 setembro 2024.

SOCIEDADE BRASILEIRA DE DIABETES. Diretrizes da Sociedade Brasileira de Diabetes 2019-2020, São Paulo, 2019. Disponível em: <https://www.saude.ba.gov.br/wp-content/uploads/2020/02/Diretrizes-Sociedade-Brasileira-de-Diabetes-2019-2020.pdf>. Acesso em: 30 Junho 2024.

SOCIEDADE BRASILEIRA DE DIABETES, SBD. Diretrizes da Sociedade Brasileira de Diabetes 2015-2016, Rio de Janeiro, 2016. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/2494325/mod_resource/content/2/DIRETRIZES-SBD-2015-2016.pdf>. Acesso em: 17 Marco 2024.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. Cambridge: MIT Press, 2018.

TIBSHIRANI, R. Regression Shrinkage and Selection via the Lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, 58, 1996. 267-288.

TORGO, L. Data Mining with R, learning with case studies. **Chapman and Hall/CRC**, 2010. Disponível em: <http://www.dcc.fc.up.pt/~ltorgo/DataMiningWithR>. Acesso em: 11 Agosto 2024.

TURLAPATI, Venkata P. K.; PRUSTY, Manas R. Outlier-SMOTE: A refined oversampling technique for improved detectionof COVID-19. **Intelligence-Based Medicine**, 3, 2020. 100023.

VASWANI, A. et al. Attention is all you need. **Advances in Neural Information Processing Systems**, arXiv preprint arXiv:1706.03762, 10, 2017. Disponível em: https://arxiv.org/abs/1706.03762. Acesso em: 20 dezembro. 2024.

VERGARA, PITER O. Desenvolvimento e Avaliação de Algoritmos de Classificação e Decisão na Regulação de Pacientes Encaminhados para a Atenção Especializada Ambulatorial. **Dissertação (mestrado) – Universidade Federal do Rio Grande**, Porto Alegre, 2020.

WANG, An. Comparison of Methods for Processing Missing Values in Psychological Research. **Advances in Psychology**, 11, 2019. 1843-1849.

WANG, Y. et al. A hybrid ensemble method for pulsar candidate classification. **Astrophysics and Space Science**, 364, 2019. 1-13.

WITTEN, I. H.; FRANK, E.; MARK, A. H. **Data Mining: Practical Machine Learning**. 3. ed. Burlington: Morgan Kaufmann, 2011.

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining: Practical Machine Learning Tools and Techniques**. 4. ed. Burlington: Morgan Kaufmann, 2016.

XUAN, Su G.; AN, Chen Z.; SHAN, Xie J. The Impact of Parameters on Missing Values in KNN Imputation Method. **Journal of Kao Yuan University**, 20, 2014. 101-107.

YADAV, Sanjay; SHUKLA, Sanyam. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. **2016 IEEE 6th International conference on advanced computing (IACC)**, 2016. 78-83.

YOU, Wen-jie et al. Feature reduction on high-dimensional small-sample data. **Computer Engineering and Applications**, 36, 2009. 165-169.

ZEBARI, R. et al. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. **Journal of Applied Science and Technology Trends**, 1, 2020. 56-70.

Zou, H., Hastie, T. Elastic Net: Regularization and Variable Selection via a Multivariate Generalization of the Lasso. **Journal of the Royal Statistical Society: Series B (Statistical Methodology***)*, 67, 2005. 301-320.

ZONATO, Willian; DROUBI, Luiz Fernando Palin; HOCHHEIM, Norberto. Pressupostos clássicos dos modelos de regressão linear e suas implicações sobre as avaliações em massa. In: **COBRAC 2018**, 2018.

ZHI-FEI, YE; YI-MIN, WEN; BAO-LIANG, LU. A survey of imbalanced pattern cla ssification problems. **Transactions on Intelligent Systems**, 4, 2009. 147-156.

ZHIQIN, LI et al. Summary of feature selection methods. **Computer Engineering and Applications**, 24, 2019. 10-19.

ZHU, X. Semi-supervised learning literature survey. **Computer Sciences Technical Report 1530, University of Wisconsin-Madison**, 2005.

ZIA, Uswa A.; KHAN, Dr. Naeem. Predicting Diabetes in Medical Datasets Using Machine Learning Techniques. **International Journal of Scientific & Engineering**, 8, 2017. 1523-1551.