UNIVERSIDADE FEDERAL DO RIO GRANDE - FURG
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

Dissertação de Mestrado

Predição de Lesões em *Cross training*: Um Estudo Comparativo com Algoritmos de Aprendizado de Máquina

João Mateus Daltro de Athayde

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestre em Engenharia de Computação

Orientador: Prof. Dr. Eduardo Nunes Borges

Rio Grande, 2025

FURG
UNIVERSIDADE FEDERAL
DO RIO GRANDE

IC3 CENTRO DE CIÊNCIAS COMPUTACIONAIS

Dissertação de Mestrado

Predição de Lesões em *Cross training*: Um Estudo Comparativo com Algoritmos de Aprendizado de Máquina

João Mateus Daltro de Athayde

Banca examinadora:

gov.br Documento assinado digitalmente
ALESSANDRO DE LIMA BICHO
Data: 09/05/2025 12:11:22-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Alessandro de Lima Bicho

gov.br Documento assinado digitalmente
GIANCARLO LUCCA
Data: 07/05/2025 17:44:34-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Giancarlo Lucca

gov.br Documento assinado digitalmente
VAGNER SANTOS DA ROSA
Data: 07/05/2025 12:26:44-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Vagner Santos da Rosa

ICP Brasil Documento assinado digitalmente
EDUARDO NUNES BORGES
Data: 06/05/2025 11:14:41-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Eduardo Nunes Borges
Orientador

Ficha Catalográfica

A865p Athayde, João Mateus Daltro de.
Predição de lesões em *Cross training*: um estudo comparativo com algoritmos de aprendizado de máquina / João Mateus Daltro de Athayde. – 2025.
112 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2025.
Orientador: Dr. Eduardo Nunes Borges.

1. *Cross training* 2. Lesoes 3. Predição 4. Aprendizado de máquina I. Borges, Eduardo Nunes II. Título.

CDU 004

Catalogação na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

AGRADECIMENTOS

Concluir este trabalho foi, sem dúvida, um dos maiores desafios que já enfrentei. Por mais que, para alguns, possa parecer uma missão simples ou apenas mais uma etapa acadêmica, para mim significou conciliar muitas responsabilidades, lidar com limitações de tempo, cansaço e dúvidas ao longo do caminho. Cada página escrita carrega noites mal dormidas, escolhas difíceis e um esforço constante para não desistir. E é justamente por isso que este momento é tão especial, porque ele representa a superação de muitos obstáculos e a realização de um objetivo que exigiu muito mais do que apenas conhecimento técnico.

Gostaria de começar primeiramente agradecendo a **DEUS**, por muitas vezes iluminar os caminhos e permitir que concluísse mais essa etapa!

Aos meus pais, **Eugênio** e **Marlice** por todo o amor, dedicação e esforço desde os primeiros passos. Esta conquista também é de vocês, que sempre acreditaram em mim e me deram as bases para chegar até aqui.

Ao meu irmão, **Moisés**, pelo apoio constante, companheirismo e por todo o icentivo ao longo dessa jornada.

À minha esposa, **Paola**, meu amor e minha base, por todo o apoio, paciência, compreensão e incentivo durante essa looooonga caminhada. Sua presença foi fundamental em TODOS os momentos, dos mais difíceis aos mais felizes. Obrigado por acreditar em mim mesmo quando eu duvidava, por carregar comigo o peso das escolhas e por ser essa mulher incrível que me inspira todos os dias. TE AMO!

À minha filha **Júlia**, minha maior motivação! Que este trabalho seja também um exemplo para você sobre a importância da dedicação e da persistência.

Ao meu orientador, **Prof. Eduardo Nunes Borges**, pela amizade, confiança e por todo o conhecimento compartilhado ao longo deste trabalho. Obrigado por me guiar com sabedoria e mostrar que, com determinação e vontade, os objetivos podem ser alcançados.

Agradecimento especial ao meu amigo **Everson**, Técnico do Centro de Ciências Computacionais da FURG, pelos direcionamentos precisos, prontidão e por sempre estar disposto a ajudar quando necessário.

Ao diretor do Centro de Gestão de Tecnologia da Informação da FURG, **Diogo**, aos coordenadores da Divisão de Sistemas, **Fábio** e **Lisandro**, pela confiança depositada em mim e pelo incentivo à minha qualificação profissional.

Aos professores do **PPGCOMP**, pela dedicação e pelas valiosas contribuições ao longo do curso, que foram fundamentais para o desenvolvimento deste trabalho e para o meu crescimento acadêmico.

E por fim, deixo aqui registrado o meu mais sincero muito obrigado aos demais amigos, familiares, professores e a todas as pessoas que, de alguma forma, me incentivaram, apoiaram e acreditaram em mim ao longo desta caminhada.

MUITO OBRIGADO!!

“Parte da jornada é o fim.”
— TONY STARK, *Vingadores: Ultimato*

RESUMO

ATHAYDE, João Mateus Daltro de. **Predição de Lesões em *Cross training*: Um Estudo Comparativo com Algoritmos de Aprendizado de Máquina**. 2025. 112 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

A prática de esportes envolve riscos, especialmente em modalidades intensas como o *cross training*. Essa prática combina exercícios de alta intensidade com movimentos funcionais e, embora proporcione condicionamento físico, também pode levar a lesões quando realizada de forma inadequada. A análise de dados esportivos com técnicas de Aprendizado de Máquina (AM) tem se mostrado promissora na minimização desses riscos, permitindo identificar padrões e sugerir ajustes no treinamento. Diante desse cenário, este trabalho tem como objetivo desenvolver e validar modelos de AM para a predição de risco de lesões em praticantes de *cross training*, com foco na comparação de diferentes algoritmos. O estudo partiu de uma base de dados composta por 673 amostras, provenientes de uma pesquisa anterior sobre lesões no treinamento funcional de alta intensidade. Foram testados alguns algoritmos supervisionados, incluindo *Random Forest*, *Support Vector Machine*, C4.5, entre outros. A avaliação dos modelos foi realizada com base em métricas como acurácia, *precision*, *recall*, *F1-Score*, área sob a curva ROC (AUC) e significância estatística. Embora os modelos não tenham apresentado desempenho expressivo em termos de acurácia geral, o algoritmo C4.5 destacou-se por alcançar os melhores resultados relativos e por sua interpretabilidade, o que motivou sua escolha para a etapa de validação. Para isso, foi desenvolvido um aplicativo WEB interativo, no qual os usuários preenchem um formulário com informações pessoais e de treino, recebendo como resposta a probabilidade de ocorrência de lesão. O sistema permitiu a avaliação das predições em confronto com os relatos reais. Apesar das limitações, como a amostra reduzida na validação e a natureza autodeclarada dos dados, os resultados indicam que modelos de AM possuem potencial para serem utilizados como ferramentas complementares em estratégias preventivas de lesões. Mesmo com desempenho modesto, o modelo demonstrou capacidade de classificar casos com razoável sensibilidade e utilidade prática para acompanhamento de risco. O estudo também aborda aspectos éticos importantes no uso de modelos preditivos, reforçando a necessidade de transparência, consentimento e uso responsável dos dados. Para trabalhos futuros, sugere-se ampliar a base de dados e incluir variáveis relacionadas a fatores externos, como sono e nutrição. Essa ampliação pode contribuir para modelos mais robustos e com maior aplicabilidade no contexto esportivo.

Palavras-chave:
cross training, lesões, predição, aprendizado de máquina.

ABSTRACT

ATHAYDE, João Mateus Daltro de. **Using machine learning to sports injury prediction**. 2025. 112 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Sports practice involves risks, especially in high-intensity modalities such as cross-training. This discipline combines functional movements with high-intensity exercises, and while it promotes physical conditioning, it can also lead to injuries when performed improperly. The analysis of sports data using Machine Learning (ML) techniques has shown promise in minimizing such risks by identifying patterns and suggesting training adjustments. In this context, this study aims to develop and validate ML models for injury risk prediction in cross-training practitioners, focusing on the comparison of different algorithms. The research is based on a dataset of 673 samples collected through a previous survey on injuries related to high-intensity functional training. Several supervised algorithms were tested, including Random Forest, Support Vector Machine, and C4.5. The models were evaluated based on metrics such as accuracy, precision, recall, F1-Score, area under the ROC curve (AUC), and statistical significance. Although the models did not achieve outstanding overall accuracy, the C4.5 algorithm stood out by yielding relatively better performance and greater interpretability, which supported its selection for the validation phase. An interactive web application was developed to implement the model in practice. Users completed a form with personal and training-related information and received the predicted probability of injury occurrence. The system allowed a comparison between expected outcomes and actual self-reported cases. Despite limitations such as a small validation sample and the self-reported nature of the data, the results suggest that ML models have the potential to be used as complementary tools in injury prevention strategies. Even with modest performance, the model demonstrated reasonable sensitivity and practical value for monitoring individual risk. This study also addresses important ethical aspects of using predictive models, emphasizing transparency, informed consent, and the responsible use of data. Future research should focus on expanding the dataset and incorporating additional variables related to external factors such as sleep and nutrition. This could lead to more robust models with broader applicability in sports contexts.

Keywords: cross-training, injuries, prediction, machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
AUC	Área sob a Curva (do gráfico ROC)
CD	Ciência de Dados
DCBD	Descoberta de Conhecimento em Banco de Dados
DT	Decision Tree (Árvore de Decisão)
FN	Falso Negativo
FP	Falso Positivo
IA	Inteligência Artificial
ID3	Iterative Dichotomiser 3 (precursor do C4.5)
J48	Implementação do algoritmo C4.5 no Weka
KDD	Knowledge Discovery in Databases
KNN	K-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LPO	Levantamento de Peso Olímpico
MPV	Mínimo Produto Viável
N/A	Not Applicable (Não se Aplica)
NN	Neural Networks (Redes Neurais)
RF	Random Forest
RFImp	Random Forest Importance
RFE	Recursive Feature Elimination
RL	Regressão Logística
RNA	Rede Neural Artificial
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
TP	Verdadeiro Positivo
TN	Verdadeiro Negativo

SUMÁRIO

1 INTRODUÇÃO

O *cross training* é uma modalidade esportiva que surgiu na Califórnia, Estados Unidos, por volta de 1980, tornando-se mais difundido em 2001 por meio da criação da marca CrossFit®. O *cross training* é composto por uma série de práticas de condicionamento físico com o objetivo de aprimorar diversos domínios da competência física, como resistência cardiorrespiratória e muscular; potência, força, velocidade, flexibilidade, agilidade, equilíbrio e coordenação motora [20]. O esporte tem como finalidade aumentar o condicionamento físico dos indivíduos pela aplicação de técnicas combinadas de intensidade, funcionalidade e variação [70].

Diversos são os benefícios oferecidos com a prática deste esporte, como a redução de taxas de doenças crônicas, doenças cardiovasculares, problemas psicológicos, artrites, entre outras. Apesar dos benefícios, a prática de exercícios de alta intensidade pode gerar malefícios, como lesões musculares e articulares [17]. Acredita-se que isso se deve ao desequilíbrio entre alta intensidade e curtos intervalos de descanso, aliado a períodos insuficientes de recuperação em relação à fadiga sofrida pelos exercícios [14]. Lesões esportivas são comuns em diferentes modalidades, tanto para atletas de elite quanto para amadores, podendo comprometer a saúde, o desempenho e, em casos mais graves, causar problemas permanentes [53].

O aumento da popularidade de esportes de alta intensidade, como o *cross training*, trouxe consigo não apenas benefícios à saúde e ao condicionamento físico, mas também novos desafios no que se refere à segurança dos praticantes. Enquanto a intensidade e a variedade dos exercícios são fatores essenciais para o sucesso da modalidade, esses mesmos fatores aumentam a complexidade do monitoramento e da prevenção de lesões. Cada indivíduo apresenta respostas fisiológicas e biomecânicas distintas, o que exige uma análise personalizada e em tempo real para garantir que os limites de segurança não sejam ultrapassados. Assim, a necessidade de soluções que integrem dados em larga escala, provenientes de diferentes fontes, tornou-se um tema central no campo do treinamento esportivo moderno [17].

Nesse contexto, o Aprendizado de Máquina (AM) surge como uma ferramenta poderosa, capaz de identificar padrões ocultos em grandes volumes de dados esportivos. Di-

ferentemente das abordagens estatísticas tradicionais, os algoritmos de AM são capazes de lidar com múltiplas variáveis simultaneamente e adaptar-se a mudanças no comportamento dos dados. Essa capacidade é particularmente relevante em modalidades como o *cross training*, onde a dinâmica dos treinos e a variabilidade dos praticantes tornam difícil o emprego de abordagens convencionais. Estudos indicam que a aplicação de técnicas de AM tem mostrado sucesso na análise de desempenho e prevenção de lesões em outras modalidades esportivas, como futebol e hóquei [87, 64, 68], sugerindo que seu uso pode oferecer benefícios semelhantes no *cross training*.

Por exemplo, uma pessoa iniciante na prática da modalidade que resolve participar de uma aula de alta intensidade com levantamento de peso e movimentos repetitivos com cargas elevadas. Devido à falta de supervisão apropriada, ela acaba comprometendo sua postura em determinados movimentos, o que, aliado ao acúmulo de fadiga muscular, o leva a desenvolver uma lesão no ombro. Esse cenário poderia ser evitado se houvesse um sistema preditivo que, baseado no histórico de dados do praticante (como tempo de prática, histórico de lesões, tipo de treino e nível de fadiga), alertasse sobre o risco elevado de lesão e recomendasse ajustes no treino ou maior supervisão.

Outro cenário envolve competidores, que frequentemente se submetem a treinos intensos visando competições de alto nível. Nesses casos, o excesso de treinos, associado a uma recuperação inadequada, pode predispor esses atletas a lesões que os afastam das competições por longos períodos. A capacidade de prever o surgimento dessas lesões com base em padrões de dados coletados ao longo dos treinos poderia não apenas melhorar a performance, mas também prevenir lesões debilitantes.

A análise de dados voltada ao esporte pode ser uma forma de prevenção ou redução de lesões e danos à saúde. Para isso, é necessário extrair conhecimento útil a partir de grandes volumes de dados, o que pode ser alcançado por meio do processo de Descoberta de Conhecimento em Banco de Dados (DCBD) [23]. Esse processo envolve etapas como seleção, limpeza, transformação, mineração de dados e interpretação dos resultados. Técnicas de AM, parte integrante da etapa de mineração, vêm sendo utilizadas para superar as limitações dos métodos estatísticos tradicionais [36].

O AM é um ramo da Inteligência Artificial composto por uma série de algoritmos que fornecem aos sistemas a capacidade de aprender e progredir automaticamente com a experiência. A aplicação de AM visa promover a automação no processo de engenharia do conhecimento utilizando conjuntos extensos de dados, substituindo o trabalho humano. O AM tem mostrado avanços significativos na predição de dados relacionados ao esporte, devido ao aumento da especificidade dos dados esportivos disponíveis e ao desenvolvimento de técnicas aplicadas para o AM [87].

Além disso, o uso de aprendizado de máquina no contexto esportivo já demonstrou sucesso em outras modalidades, como o futebol [64] [37] e o hóquei [68], onde modelos preditivos são utilizados para avaliar a performance de jogadores e prever o risco de

lesões. No *cross training*, onde a variabilidade dos tipos de treino e a intensidade impõem desafios adicionais, a aplicação de AM pode fornecer uma abordagem inovadora para lidar com a prevenção de lesões.

Portanto, diante desse cenário, este trabalho propõe a aplicação de técnicas de AM para o desenvolvimento e comparação de modelos preditivos capazes de identificar padrões em dados de praticantes de *cross training* e prever o risco de lesões. Cabe destacar que o modelo proposto neste estudo tem como objetivo prever a ocorrência de lesões de forma binária, classificando os praticantes entre as categorias "Sim" (risco identificado) e "Não" (sem risco identificado), com base em suas características e histórico de treino. Junto à predição da classe, o modelo fornece uma estimativa de probabilidade associada à predição, o que pode ser interpretado como uma medida contínua de risco. Essa probabilidade permite não apenas a classificação, mas também a compreensão do grau de confiança da predição, oferecendo uma visão mais informativa para auxiliar em decisões preventivas.

Além de construir um modelo aplicável na prática, o estudo também visa comparar diferentes algoritmos com base em métricas estatísticas e sua viabilidade para uso em um ambiente real. Tais técnicas têm sido utilizadas em áreas como medicina esportiva, onde a análise de grandes conjuntos de dados permite identificar relações complexas entre múltiplas variáveis. Modelos preditivos baseados em AM têm o potencial de oferecer análises mais precisas sobre os riscos de lesão, permitindo que treinadores e praticantes adotem medidas preventivas personalizadas.

1.1 Objetivo geral

Implementar e comparar modelos de aprendizado de máquina para prever o risco de lesões em praticantes de *cross training*, com foco na identificação do algoritmo mais adequado para validação prática.

1.2 Objetivos específicos

- Realizar o tratamento e a transformação de dados para aplicação de algoritmos de AM;
- Comparar a capacidade preditiva de diferentes algoritmos de aprendizado de máquina na tarefa de prever lesões;
- Identificar os fatores de risco mais relevantes na ocorrência de lesões, com base na importância das variáveis;
- Construir um Mínimo Produto Viável (MPV) [51] utilizando interface web interativa para validação prática do modelo preditivo em um ambiente real;

- Investigar limitações e a viabilidade do uso de modelos preditivos como ferramenta de suporte à decisão em treinamento funcional de alta intensidade.

1.3 Organização do texto

O restante do texto está organizado da seguinte forma:

O Capítulo 2 trata da fundamentação teórica, abordando os principais conceitos relacionados à modalidade esportiva, às lesões no contexto esportivo, à Descoberta de Conhecimento em Banco de Dados e às técnicas de Aprendizado de Máquina. São apresentados os algoritmos utilizados na literatura, bem como as métricas adotadas para avaliação dos modelos.

O Capítulo 3 apresenta trabalhos relacionados, destacando alguns estudos voltados à predição de lesões em diferentes esportes, com foco em abordagens baseadas em AM, além de uma análise comparativa dos métodos encontrados.

O Capítulo 4 descreve a metodologia adotada, detalhando a origem dos dados, o processo de pré-processamento e transformação das variáveis, as técnicas de seleção de atributos e os modelos de aprendizado utilizados. Também são apresentadas as estratégias de avaliação e validação dos modelos preditivos.

O Capítulo 5 reúne os resultados obtidos, incluindo a preparação final da base de dados, a análise descritiva dos participantes, a comparação entre algoritmos, a escolha do modelo preditivo e sua aplicação em um ambiente interativo. São ainda discutidas as predições realizadas e consideradas as implicações éticas do uso do modelo desenvolvido.

Por fim, o Capítulo 6 traz as considerações finais, com a síntese dos resultados alcançados, as limitações do estudo, sugestões para pesquisas futuras e reflexões sobre a aplicabilidade prática do modelo preditivo desenvolvido.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos fundamentais relacionados ao tema de pesquisa e uma visão geral sobre os principais assuntos relacionados ao objeto central de estudo: aprendizado de máquina.

2.1 *Cross Training*

A prática de *cross training* é composta de diferentes exercícios que permitem ao indivíduo dominar vários aspectos físicos como, por exemplo, Levantamento de Peso Olímpico (LPO), escalada de corda, esportes tradicionais, movimentação de grandes cargas, entre outros. A estrutura do treino pode variar entre os diversos centros de treinamento os quais são aplicados mas, geralmente, cada sessão de treinamento dura cerca de uma hora. A composição do treino é basicamente dividida em um período específico de aquecimento, força e/ou técnicas de treinamento, processo de ganho de força ou treino de condicionamento por 10-30 minutos, e finalizado com exercícios de volta à calma e/ou mobilidade. No *cross training*, o programa de treinamento difere dependendo da condição individual de cada atleta e dos seus objetivos, podendo variar as condições de intensidade, duração, complexidade e forma de organização [75] [84].

A atividade física como um todo tem seus benefícios à saúde extremamente difundidos na literatura. Quando se trata especificamente sobre a modalidade *cross training* o cenário voltado à saúde não é diferente. Diversos são os aspectos benéficos associados à prática. O *cross training* auxilia na melhoria das capacidades físicas, tais como aumento da capacidade aeróbia e anaeróbia, ganho de força e flexibilidade [86]. Além disto, o treinamento proporciona impactos positivos aos praticantes como, a melhora da composição corporal por meio da perda de gordura. Com o aumento da popularidade do esporte e os grandes benefícios ligados à sua prática, alguns aspectos negativos têm sido avaliados, como o risco de lesões devido à alta intensidade da prática [24].

2.2 Lesões

As lesões são comuns a toda prática esportiva. No entanto, quando se fala em *cross training*, os riscos de lesões podem ser associados a fatores intrínsecos e extrínsecos. Dentre estes, idade, lesões prévias, iniciantes no esporte, despreparo de professores, excesso de treinamento, entre outros motivos [71]. Alguns estudos têm demonstrado que esta prática esportiva desencadeia reações bioquímicas diversas, como estresse oxidativo, maior produção de lactato, cortisol e aumento da função metabólica [80]. Os riscos de lesões musculoesqueléticas ligadas ao *cross training*, em geral acometem, principalmente, a articulação dos ombros e a coluna lombar [75].

O levantamento de dados acerca destes aspectos, aliado à aplicação de algoritmos de predição, poderia detalhar de forma mais sistemática o acometimento das lesões e auxiliar na prevenção. Isso poderia acarretar benefícios à saúde de praticantes e atletas, além de reduzir custo e tempo de recuperação [22].

2.3 Descoberta de Conhecimento em Banco de Dados

Com os avanços ligados à tecnologia de coleta e armazenamento de dados, surgiu a possibilidade de utilização desses materiais para a extração de novas e relevantes informações. Porém, essas descobertas são desafiadoras, uma vez que técnicas tradicionais de análises de dados mostram dificuldades de aplicação em conjuntos massivos de dados. Com o intuito de minimizar os desafios e ampliar as respostas do grande volume de dados, surgiu a Descoberta de Conhecimento em Banco de Dados (DCBD), em inglês *Knowledge Discovery in Databases* (KDD). A DCBD é uma tecnologia que combina métodos tradicionais de análise de dados com algoritmos sofisticados para processar grandes volumes de dados. Ela também abriu oportunidades interessantes para explorar e analisar novos tipos de dados e para analisar tipos antigos de dados de novas maneiras [76].

A DCBD é composta por uma série de etapas que incluem desde a preparação dos dados até a interpretação dos resultados. A sua aplicação possibilita a identificação de padrões, tendências e correlações que passariam despercebidos em análises convencionais [23]. A DCBD é dividida em cinco etapas, conforme ilustra a Figura 1:

1. Seleção dos dados: Nesta fase, é realizada a seleção de um conjunto ou subconjunto de dados que serão utilizados para a análise, considerando apenas as variáveis de interesse. Esse passo é essencial para reduzir a quantidade de dados a serem manipulados e focar no objetivo da análise [76].

2. Pré-processamento dos dados: Nesta etapa é realizada uma avaliação da qualidade dos dados, uma vez que os dados geralmente contêm ruídos, valores faltantes e redundâncias. Portanto, uma limpeza e transformação dos dados é aplicada,

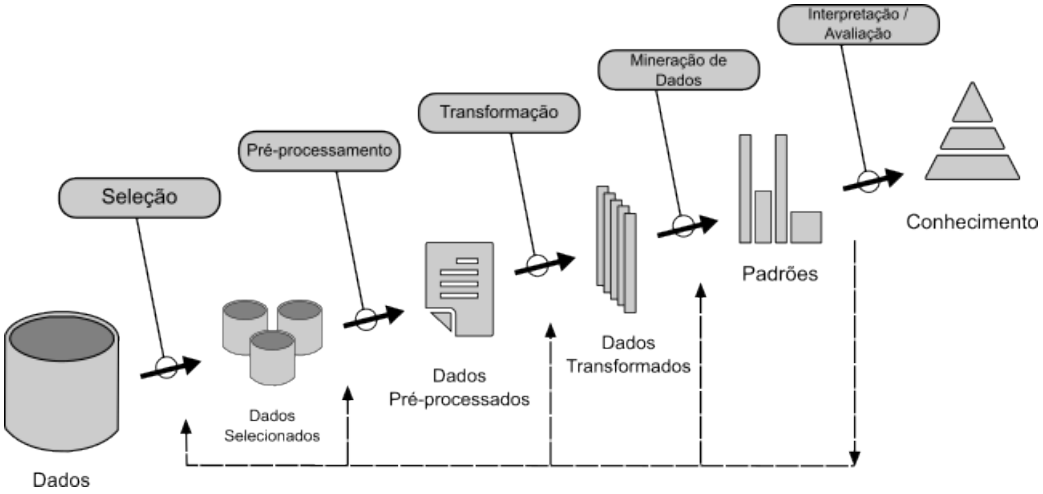


Figura 1: Etapas do processo de DCBD
Fonte: Adaptado de [23]

utilizando técnicas como substituição de valores nulos, remoção de duplicatas e normalização [30].

3. Transformação dos Dados: Esta fase de transformação se baseia na aplicação de técnicas que transformem os dados de modo a facilitar a descoberta de padrões. As transformações podem ser, por exemplo, normalização, agregação, criação ou derivação, conversão de dados nominais para numéricos e discretização. [23].

4. Mineração de Dados: Nesta fase são aplicadas técnicas de aprendizado de máquina e estatística para descobrir padrões e tendências nos dados. Técnicas comuns incluem árvores de decisão, redes neurais, algoritmos de agrupamento (*clustering*) e análise de associações. A escolha da técnica depende do tipo de padrão que se deseja identificar [3]. A mineração dos dados pode ser subdividida em:

    - Mineração de Dados Descritiva: Busca resumir as características gerais dos dados. Por exemplo, o agrupamento (*clustering*) organiza os dados em grupos com características semelhantes, enquanto as regras de associação descobrem relações frequentes entre variáveis [30];
    - Mineração de Dados Preditiva: Foca na construção de modelos que permitam prever o comportamento futuro dos dados. Métodos como a regressão linear, árvores de decisão e redes neurais são utilizados para criar modelos que podem prever valores ou categorias com base em dados históricos [3].

5. Interpretação e Avaliação: Nesta etapa é realizada uma análise dos padrões descobertos para avaliar a validade e relevância do modelo. Nem todos os padrões encontrados são úteis ou aplicáveis, portanto, a validação pode ser aplicada para avaliar os dados. Os resultados obtidos podem ser demonstrados por gráficos, tabelas e relatórios [23].

2.4 Aprendizado de máquina

O Aprendizado de Máquina é um campo de estudo dentro da Inteligência Artificial (IA), focado no desenvolvimento de programas de computador capazes de aprender a realizar uma tarefa específica com base em experiências próprias [46]. Dentro do amplo campo da IA, o AM destaca-se como uma abordagem específica que permite aos sistemas aprenderem padrões e tomar decisões com base em dados.

A Figura 2 mostra que a IA se enquadra no domínio da Ciência de Dados (CD) e engloba o AM que por sua vez contém muitos modelos e métodos, incluindo aprendizagem profunda (AP) e redes neurais artificiais (RNA).

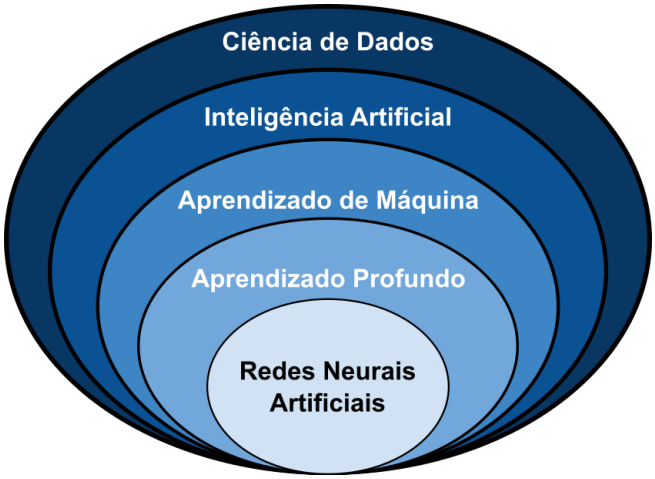


Figura 2: Representação da Ciência de Dados
Fonte: Choi et al. (2020) [19]

No contexto da IA, o AM é uma técnica fundamental que capacita os sistemas a aprenderem com experiências anteriores e aprimorarem seu desempenho ao longo do tempo, sem a necessidade de programação explícita para tarefas específicas. Como mencionado por Alpaydin [7], o AM é uma disciplina que se concentra na criação de algoritmos e modelos capazes de aprender com dados, adaptar-se e realizar previsões ou tomar decisões.

Segundo Choi et al. [19] existem quatro abordagens de Aprendizado de Máquina comumente utilizadas, cada uma útil para resolver diferentes tarefas: supervisionado, não supervisionado, semi-supervisionado e aprendizado por reforço.

- Aprendizado Supervisionado: Neste método, o modelo é treinado com um conjunto de dados rotulado, ou seja, dados que já têm respostas conhecidas. O modelo aprende a mapear as entradas para as saídas desejadas, permitindo a previsão de novos dados [12];

- Aprendizado Não Supervisionado: Aqui, o modelo é treinado em dados não rotulados, e o objetivo é descobrir padrões ou estruturas subjacentes nos dados. O modelo agrupa os dados com base em similaridades ou características comuns [31];

- Aprendizado Semi-Supervisionado: Combina elementos do aprendizado supervisionado e não supervisionado, utilizando conjuntos de dados que contêm tanto dados rotulados quanto não rotulados. Isso é útil quando obter um grande conjunto de dados rotulado é difícil ou dispendioso [19];

- Aprendizado por Reforço: Nesse método, um agente aprende a realizar ações em um ambiente para atingir um objetivo específico. O modelo recebe *feedback* na forma de recompensas ou penalidades, ajustando seu comportamento para maximizar as recompensas [12].

Com o avanço dos recursos computacionais, a aplicação de AM em diversos campos têm crescido, incluindo a medicina esportiva. A avaliação, mitigação e prevenção de lesões são de extrema importância, dadas as consequências físicas, emocionais e financeiras, especialmente em níveis profissionais. Vários modelos de AM foram propostos na literatura para elucidar os fatores complexos que contribuem para lesões atléticas e permitir maior precisão preditiva [9].

Apesar de revisões recentes explorarem aspectos específicos desse campo, limitações existem, como abordagens centradas em mineração de dados sem considerar a atualidade, foco em esportes específicos, escopo limitado ou concentração apenas em esportes coletivos.

2.4.1 *One-Hot Encoding*

Dentro do campo de AM, uma das etapas do pré-processamento de dados é a transformação dos dados, sendo que uma das transformações mais comuns é a conversão de variáveis categóricas para um formato numérico, já que alguns algoritmos de AM exigem que os dados sejam representados dessa forma. Nesse contexto, no caso das variáveis categóricas, que representam classes ou categorias distintas, uma técnica utilizada é o *One-Hot Encoding* [57].

Essa técnica converte cada categoria de uma variável em uma nova coluna binária, onde o valor 1 indica a presença da categoria e 0 a ausência. Isso permite que os dados sejam representados de forma que os algoritmos possam processar variáveis categóricas sem introduzir uma hierarquia implícita entre as categorias [57]. A Tabela 1 ilustra o procedimento da técnica.

O *One-Hot Encoding* aplicado na coluna de respostas do dataset ilustrado na tabela 1 transforma as categorias presentes em valores binários, permitindo representar as respostas multivaloradas de maneira adequada. No exemplo, foram definidas quatro grandes categorias: Treinamento de Força/Funcional, Esportes de Resistência, Esportes Competitivos, e Outros Esportes e Atividades. Para cada linha de resposta, o algoritmo cria uma nova coluna para cada uma dessas categorias, e atribui o valor 1 quando a categoria está presente na resposta, e 0 quando a categoria não está. Por exemplo, a resposta

Tabela 1: Exemplo de aplicação de *one hot encoding*

Exemplo de respostas	**Categorias**			
	Treinamento de Força/Funcional	**Esportes de Resistência**	**Esportes Competitivos**	**Outros Esportes e Atividades**
"Volei, andar de bicicleta, corrida, yoga"	0	1	1	1
"Yoga, corrida e calistenia"	1	1	0	0
"Natação, corrida, ciclismo, trilha"	0	1	0	0
"Musculação, calistenia, yoga e corrida"	1	1	0	1
"Corrida , pedal, futebol"	0	1	1	0
"Voleibol e futsal"	0	0	1	0
"Surf, natação, futebol"	0	1	1	0
"Natação, ciclismo, corrida de rua"	0	1	0	0
"Musculação"	1	0	0	0

"Musculação, calistenia, yoga e corrida" é representada com 1 nas colunas correspondentes a Treinamento de Força/Funcional, Esportes de Resistência, e Outros Esportes e Atividades, e 0 nas demais, permitindo que uma única linha de dados represente múltiplas categorias de forma independente. Isso facilita a análise e a aplicação de modelos que exigem dados numéricos, e permite a avaliação de mais de uma resposta por item, preservando a integridade das informações originais.

A principal vantagem desta técnica está na eliminação do viés introduzido por técnicas como *label encoding* [93], que atribuem valores inteiros às categorias, podendo induzir interpretações ordinais indevidas pelos modelos de AM. Isso ocorre porque alguns algoritmos podem interpretar os valores numéricos como tendo uma relação de ordem ou magnitude, o que não é inerente às categorias originais. O *one-hot encoding* evita essa ambiguidade ao representar cada categoria como uma dimensão independente, permitindo que os modelos aprendam sem pressupor qualquer ordenação entre as categorias [6].

Além disso, é eficaz para lidar com dados faltantes. Diferentemente dos métodos tradicionais de imputação, que tentam estimar valores ausentes com base em outras observações, o *one-hot encoding* pode tratar valores faltantes como uma nova categoria distinta. Isso evita a interferência na estrutura original dos dados, e preserva a independência dos valores ausentes em relação às demais categorias, o que pode ser crucial para a integridade do modelo e para evitar vieses na classificação [92].

No entanto, uma limitação é o aumento significativo da dimensionalidade dos dados, especialmente quando a variável categórica possui muitas categorias distintas. Isso pode levar a vetores esparsos e a um custo computacional elevado, além de possíveis problemas de privacidade, pois a presença explícita de categorias pode revelar informações sensíveis. [90].

2.5 Métodos de Aprendizado Supervisionado

Os algoritmos de aprendizado supervisionado são utilizados para a construção de modelos preditivos a partir de dados rotulados, onde a variável dependente é conhecida.

Esses métodos buscam identificar padrões e relações entre as variáveis independentes e a variável alvo para realizar previsões ou classificações [55]. No presente trabalho, serão abordados alguns algoritmos, como Regressão Logística, KNN, *Random Forest*, SVM, *Naive Bayes*, *AdaBoost*, C4.5 e C5.0. A seguir, são apresentados os fundamentos teóricos de cada um, destacando suas vantagens e limitações em contextos práticos.

2.5.1 Regressão Logística - RL

A Regressão Logística (RL) é uma técnica estatística que pode ser utilizada para modelar variáveis dependentes binárias, isto é, aquelas que assumem apenas dois valores possíveis, como "sim" ou "não", "lesão" ou "sem lesão". Diferente da regressão linear, que prevê valores contínuos, a regressão logística é projetada para prever a probabilidade de ocorrência de um evento, assumindo valores no intervalo $[0, 1]$ [35].

A principal característica desse modelo é o uso da função logística, também chamada de função sigmoide, que transforma a combinação linear dos preditores em uma probabilidade. A função logística é dada por:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{1}$$

No contexto do modelo, o valor de z é calculado a partir de uma combinação linear das variáveis preditoras $X_1, X_2, ..., X_k$:

$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k. \tag{2}$$

Com isso, a equação final da regressão logística pode ser escrita como:

$$P(Y = 1 \mid X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k)}}. \tag{3}$$

Essa equação fornece a probabilidade estimada de ocorrência de um evento (por exemplo, o risco de lesão) com base nas variáveis independentes. A Figura 3 ilustra a função sigmoide utilizada pela regressão logística, que transforma o valor predito por uma combinação linear de variáveis em uma probabilidade contínua entre 0 e 1.

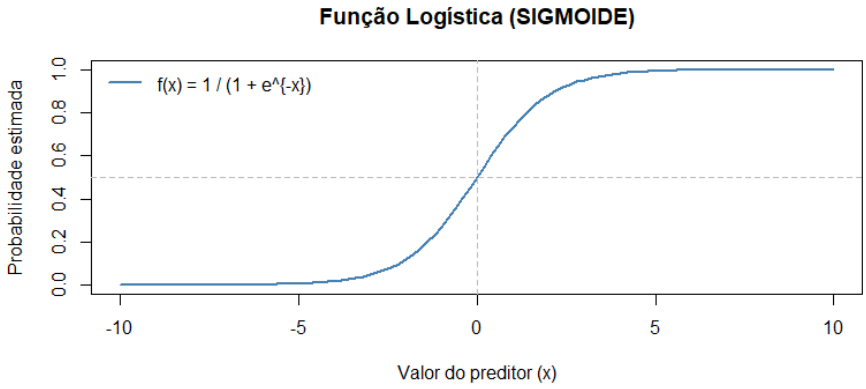


Figura 3: Curva da função logística (sigmoide), utilizada na regressão logística para mapear preditores em probabilidades.

Apesar de ser bem utilizada, a Regressão Logística apresenta limitações importantes. Uma das principais desvantagens é a suposição de linearidade entre os preditores e o logaritmo das chances do evento, o que pode não refletir adequadamente relações complexas nos dados [35]. Além disso, conforme apontado por Vittinghoff and McCulloch [83], a Regressão Logística pode ter desempenho comprometido em conjuntos de dados muito pequenos, especialmente quando há um número elevado de preditores.

2.5.2 *K-Nearest Neighbor*

O algoritmo *K-Nearest Neighbor* (KNN) é um método de aprendizado supervisionado inicialmente proposto em 1951 e, posteriormente, aprimorado [29]. Trata-se de um algoritmo não paramétrico baseado em instâncias, que pode ser utilizado em tarefas de classificação e regressão, com aplicações em áreas como mineração de dados, sistemas de recomendação, Internet das Coisas (IdC) e Indústria 4.0 [29].

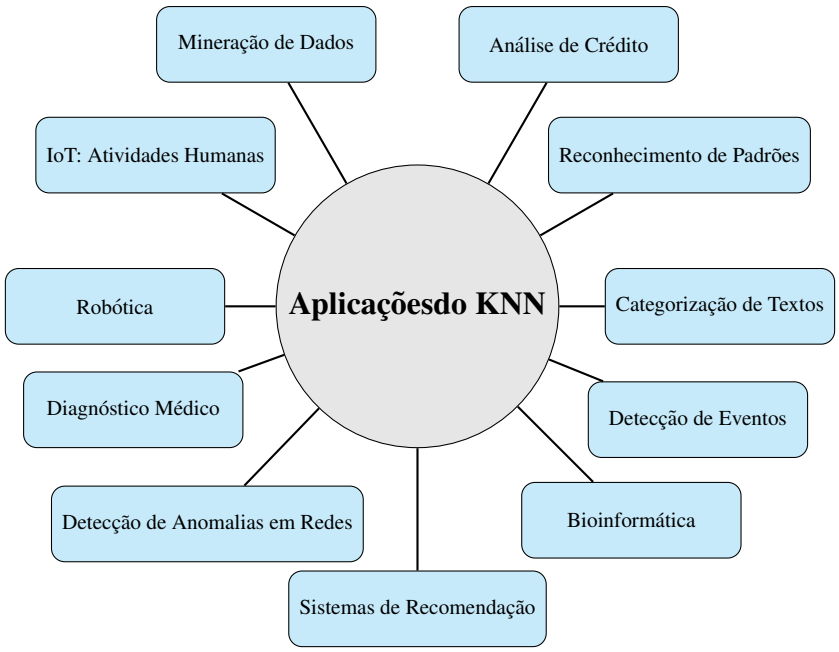
Ao contrário de métodos baseados em modelos, o KNN é considerado um algoritmo de aprendizado preguiçoso, pois não requer uma fase de treinamento explícita. A previsão é realizada com base na similaridade entre um novo ponto e os exemplos no conjunto de dados, utilizando métricas de distância como Euclidiana [21].

O valor de k, número de vizinhos a serem considerados, é um hiperparâmetro relevante que influencia diretamente o desempenho do modelo [21]. Para classificação, o algoritmo atribui ao novo ponto a classe mais frequente entre os k vizinhos; para regressão, calcula a média dos valores associados aos k vizinhos [21].

O algoritmo KNN é aplicado em muitos campos devido à sua simplicidade. No domínio da mineração de dados, é utilizado em tarefas de classificação e regressão, como na análise de crédito, onde auxilia na distinção entre bons e maus pagadores com base em dados financeiros históricos [29]. Sua versatilidade também o torna uma escolha comum em aplicações como reconhecimento de padrões, categorização de textos e detecção de eventos [1]. O desempenho do algoritmo depende fortemente da seleção adequada da

métrica de distância, fator essencial para garantir resultados confiáveis [21].

Figura 4: Principais aplicações do algoritmo KNN.



Fonte: [29, 1, 21].

O KNN tem sido aplicado em diversas áreas: na bioinformática, é utilizado para a análise de expressão gênica; em sistemas de recomendação, sugere itens com base na similaridade entre usuários; e na detecção de anomalias em redes, contribui para o monitoramento de segurança. Também é empregado no diagnóstico médico, na robótica — para controle de movimentos e interação humano-robô — e em aplicações de Internet das Coisas, como o reconhecimento de atividades humanas [29]. Essas e outras aplicações do algoritmo estão ilustradas na Figura 4.

2.5.3 *Random Forest*

O *Random Forest* foi introduzido por Breiman em 2001, baseado em ideias anteriores de Amit e Geman, e Ho [63]. O conceito é utilizado em sistemas de detecção de intrusões e em outras áreas como a previsão de safras. É um modelo de Aprendizado de Máquina que consiste em um conjunto de Árvores de Decisão, podendo ser utilizado tanto para classificação quanto para regressão. No caso de classificação, a previsão é baseada na votação majoritária dos valores previstos pelas árvores de decisão, enquanto na regressão, o resultado é a média dos resultados das árvores [63].

A principal vantagem do *Random Forest* é sua capacidade de melhorar o desempenho preditivo ao combinar múltiplas árvores de decisão, cada uma construída a partir de um subconjunto aleatório de dados e características. Essa aleatoriedade ajuda a criar árvores diferentes, que, quando combinadas, geralmente alcançam um desempenho preditivo su-

perior [63]. Além disso, é conhecido por ser um algoritmo fácil de usar, devido à sua capacidade de operar como um comitê de modelos relativamente independentes, o que resulta em previsões mais precisas do que qualquer modelo individual [27]. Também tem sido aplicado em diferentes domínios devido à sua capacidade de generalização. Em sistemas de detecção de intrusões, é utilizado não apenas como classificador, mas também para seleção de atributos e definição de métricas de proximidade [63].

Na agricultura, tem sido empregado na seleção de características relevantes e na previsão da adequação de culturas a determinadas áreas, estimando a produção esperada com base em dados do solo e clima [27]. Sua capacidade está associada à estrutura de conjunto, que reduz a influência de erros individuais das árvores e contribui para decisões mais estáveis. No campo dos recursos hídricos, tem sido utilizado na previsão de vazão de rios e na análise do desempenho de modelos de simulação de eventos de inundação [39].

O *Random Forest* é um método de *bagging* [74], onde as árvores de decisão são executadas em paralelo, sem interação entre elas, o que contribui para a eficácia do modelo. Essa abordagem de "inteligência coletiva" permite que um grande número de modelos relativamente independentes opere como um comitê, superando qualquer modelo individual [27].

2.5.4 *Support Vector Machine*

Support Vector Machine (SVM) é um método de aprendizado de máquina supervisionado, desenvolvido inicialmente por Vapnik e Chervonenkis na década de 1960. No entanto, sua popularização ocorreu a partir da década de 1990, com o desenvolvimento da técnica conhecida como *kernel trick* [65], que permitiu a aplicação do SVM a problemas de classificação altamente não lineares. Posteriormente, sua formulação foi estendida para tarefas de regressão, originando a chamada máquina de regressão de vetor de suporte (SVR) [82].

Tradicionalmente utilizado em problemas de classificação binária, o SVM tem sido adotado em diversos contextos da ciência aplicada. Sua principal característica é a capacidade de construir um hiperplano ótimo que separa as classes com a maior margem possível, promovendo uma separação eficaz dos dados e favorecendo a capacidade de generalização do modelo [85]. Apesar de sua fundamentação em conceitos avançados de otimização convexa, álgebra linear e teoria do aprendizado estatístico, o princípio do SVM pode ser compreendido de forma intuitiva: o algoritmo utiliza a geometria dos dados para identificar padrões, diferenciando-se de abordagens estatísticas tradicionais [82].

Entre suas vantagens, destacam-se a capacidade de lidar com diferentes tipos de dados, a boa generalização mesmo em conjuntos com alta dimensionalidade e a obtenção de soluções ótimas. Tais propriedades decorrem da fundamentação na *Statistical Learning Theory* (SLT), que oferece ao algoritmo uma base teórica sólida para a tomada de

decisão. Essas características tornam o SVM uma escolha recorrente em aplicações como diagnóstico de falhas em sistemas mecânicos, reconhecimento de padrões biométricos e previsão de estruturas biológicas [40].

Apesar de seus benefícios, a aplicação do SVM pode ser limitada por desafios como a complexidade matemática envolvida em sua formulação e o elevado custo computacional, especialmente em contextos com grandes volumes de dados (*big data*) ou que demandam ajuste fino de hiperparâmetros [73].

A forma como o SVM lida com diferentes problemas depende da separabilidade das classes envolvidas:

1. **Problemas Lineares**: Quando as classes são linearmente separáveis, o SVM utiliza um modelo matemático direto, cuja equação do hiperplano é dada por:

$$y = wx' + \gamma. \tag{4}$$

   Nesse cenário, o objetivo do algoritmo é identificar o hiperplano que maximiza a margem entre as classes, garantindo uma melhor generalização e minimizando o erro em novas amostras [73].

2. **Problemas Não Lineares**: Em situações nas quais os dados não apresentam separabilidade linear, o SVM emprega funções *kernel* para mapear os dados para um espaço de características de dimensão superior. Nesse novo espaço, torna-se possível encontrar um hiperplano linear que, no espaço original, corresponde a uma fronteira de decisão não linear. Essa estratégia é complementada pela introdução de variáveis de folga (*slack variables*), que conferem ao modelo certa tolerância a erros de classificação, tornando-o mais robusto à presença de ruídos e à sobreposição entre classes [82, 73, 40].

Dessa forma, o SVM demonstra uma capacidade adaptativa, ajustando suas estratégias conforme a complexidade dos dados e o grau de separabilidade entre as classes. A conjugação entre técnicas de mapeamento não linear, princípios de otimização e fundamentação teórica permite a ampla aplicação do SVM em diferentes domínios da ciência e engenharia, especialmente em tarefas que requerem elevada acurácia na separação entre categorias [73, 82].

Entre as funções *kernel* mais utilizadas, destacam-se o **kernel radial** e o **kernel polinomial**, que conferem ao SVM flexibilidade para lidar com fronteiras de decisão complexas:

- **SVM com Kernel Radial (RBF - *Radial Basis Function*)**: Este kernel é definido pela equação:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{5}$$

em que γ é um parâmetro que controla a largura da função radial. O kernel RBF é especialmente útil para problemas nos quais a separação entre as classes não é linear no espaço original dos dados.

- **SVM com Kernel Polinomial**: Neste caso, o kernel é definido como:

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d \tag{6}$$

onde d representa o grau do polinômio e c é uma constante de ajuste. Esse kernel é adequado para capturar relações polinomiais entre as variáveis de entrada.

Ambos os kernels permitem que o SVM encontre um hiperplano de separação em um espaço transformado, viabilizando a classificação de dados que não são linearmente separáveis no espaço original [40, 82].

2.5.5 *Naive Bayes*

O Naive Bayes é um classificador baseado em probabilidade que utiliza o teorema de Bayes para prever a classe de um dado com base em dados previamente rotulados. Ele assume que todos os atributos de entrada são independentes entre si, o que é uma simplificação que nem sempre é verdadeira na prática. Essa abordagem é conhecida como ”ingênua”(naive) devido a essa suposição de independência. O classificador Naive Bayes é simples e menos complexo, proporcionando resultados de classificação em um curto espaço de tempo, mas geralmente é menos preciso em comparação com outros algoritmos, como as árvores de decisão [50, 27].

O teorema de Bayes é a base matemática do Naive Bayes. Ele permite calcular a probabilidade de uma hipótese com base em evidências [50]. A fórmula básica do teorema de Bayes é:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{7}$$

onde:

- $P(A|B)$ é a probabilidade da hipótese A ser verdadeira dado que B é verdadeiro,
- $P(B|A)$ é a probabilidade de B ser verdadeiro dado que A é verdadeiro,
- $P(A)$ é a probabilidade de A ser verdadeiro,
- $P(B)$ é a probabilidade de B ser verdadeiro.

O Naive Bayes classifica os dados com base na probabilidade de cada classe, dada a entrada. Ele utiliza a suposição de independência para simplificar o cálculo das probabilidades, resultando em um modelo eficiente e de baixo custo computacional [27].

O Naive Bayes é utilizado em aplicações como filtragem de spam, análise de sentimentos e sistemas de recomendação. No entanto, a precisão do Naive Bayes pode ser comprometida quando as características são altamente correlacionadas, como observado em estudos de desempenho comparativo [50].

As principais vantagens do Naive Bayes incluem sua simplicidade, eficiência computacional e capacidade de lidar com grandes conjuntos de dados. No entanto, a suposição de independência entre as características pode levar a resultados menos precisos em alguns casos [50].

O Naive Bayes continua a ser uma ferramenta valiosa em machine learning, especialmente em cenários onde a simplicidade e a velocidade são mais importantes do que a precisão absoluta. No entanto, é importante estar ciente de suas limitações, especialmente em relação à suposição de independência das características [27, 50].

2.5.6 *AdaBoost*

AdaBoost, ou Adaptive Boosting, é um algoritmo de aprendizado de máquina que melhora o desempenho de classificadores fracos combinando-os em um classificador forte. Desenvolvido por Freund e Schapire, o AdaBoost ajusta iterativamente os pesos dos exemplos de treinamento, aumentando os pesos dos exemplos que foram incorretamente classificados e diminuindo os pesos dos exemplos corretamente classificados. Isso força o próximo classificador a focar nos exemplos mais difíceis [41].

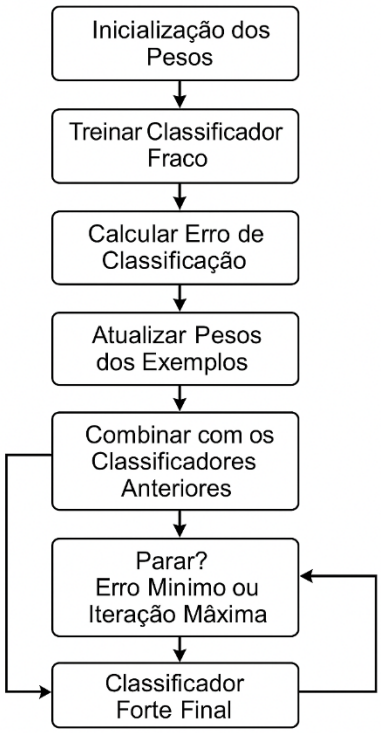
O algoritmo é conhecido por sua simplicidade, eficiência e capacidade de combinar com outros algoritmos, como SVM, Redes Neurais e Random Forests, para alcançar um desempenho ótimo [41]. AdaBoost é utilizado devido à sua precisão, facilidade de implementação e rápido tempo de treinamento. No entanto, é frequentemente considerado uma ”caixa preta”devido à sua estrutura interna complexa, que geralmente envolve centenas a milhares de árvores de decisão rasas [32].

Além de sua forma tradicional, o AdaBoost pode ser estendido para uma versão real, na qual os classificadores fracos não se limitam a produzir saídas binárias. Nessa abordagem, os aprendizes fornecem valores contínuos que indicam a plausibilidade de uma determinada classe, ampliando a flexibilidade e a expressividade do modelo. A dinâmica iterativa do AdaBoost, aliada às diversas interpretações teóricas já propostas, fornece uma base sólida para investigações futuras, tanto no aprimoramento do algoritmo quanto na compreensão de seu comportamento em longo prazo [11].

O funcionamento do AdaBoost segue um fluxo sistemático, baseado na combinação sequencial de classificadores fracos com o objetivo de construir um classificador forte e mais robusto [41]. Este fluxo de trabalho é visualizado em diagramas de fluxo, como o

mostrado na Figura 5.

Figura 5: Diagrama de fluxo representando o funcionamento do algoritmo AdaBoost.



Fonte: Adaptado de [41].

O diagrama ilustra como os classificadores fracos são iterativamente ajustados e combinados para formar um classificador forte, destacando a atualização dos pesos e a combinação dos classificadores [41].

2.5.7 C4.5 (J48)

O C4.5, também conhecido como J48 no software WEKA, é um algoritmo de árvore de decisão desenvolvido por Quinlan Ross em 1993 como sucessor do ID3. Ele é utilizado para problemas de classificação e é conhecido por sua capacidade de lidar com atributos categóricos e contínuos, além de gerenciar valores ausentes [2]. Conhecido por produzir resultados mais precisos e de funcionar bem em casos de valores ausentes [60].

O algoritmo constrói árvores de decisão a partir de um conjunto de dados de treinamento usando o conceito de entropia da informação. Para lidar com atributos contínuos, o C4.5 cria um limiar e divide os dados em duas partes: aqueles com valores de atributo acima do limiar e aqueles com valores iguais ou abaixo [91]. Além disso, o C4.5 realiza a poda das árvores após a criação, substituindo nós internos por nós folha para reduzir a taxa de erro [2, 91].

O C4.5 utiliza o método de razão de ganho para avaliar o atributo de divisão, o que ajuda a remover o viés do ganho de informação quando há muitos valores de resultado de

um atributo [2]. Ele é considerado adequado para problemas do mundo real devido à sua capacidade de lidar com atributos numéricos e valores ausentes [2].

Ele é aplicado na classificação de doenças, como a febre tifóide, onde ajuda a prever a presença da doença com base em sintomas e outros dados clínicos [2]. Em ambientes de e-learning, o C4.5 é utilizado para prever o desempenho acadêmico dos alunos, analisando dados como notas e participação em atividades, o que auxilia na identificação de alunos que possam precisar de suporte adicional [45, 48]. Além disso, o algoritmo é empregado na análise de churn de clientes em setores como seguros, ajudando a identificar padrões que indicam a probabilidade de um cliente cancelar um serviço [91].

2.5.8 C5.0

O algoritmo C5.0 é uma técnica de aprendizado de máquina utilizada para a classificação de tráfego na internet. Derivado do algoritmo C4.5, desenvolvido por Ross Quinlan, o C5.0 é reconhecido por apresentar melhorias significativas em eficiência e precisão na construção de árvores de decisão. Uma de suas principais vantagens é a capacidade de ponderar diferentes atributos e tipos de erros de classificação, além de reduzir automaticamente o ruído nos dados, o que contribui para um desempenho mais preciso e eficiente [91]. O C5.0 também se destaca por ser mais rápido, consumir menos memória e gerar árvores de decisão menores e mais interpretáveis [60]. Em comparação com o C4.5, o C5.0 apresenta melhor desempenho no tratamento de valores ausentes e na geração de regras em formato mais legível, enquanto o C4.5 tende a produzir regras exclusivamente a partir da estrutura da árvore [2]. Apesar de ser um produto comercial e de código fechado, existem implementações gratuitas que permitem interpretar e utilizar as árvores e regras geradas pelo algoritmo [91]. Tais características tornam o C5.0 uma escolha relevante para aplicações que exigem alta precisão e eficiência na classificação de dados.

Desenvolvido como uma evolução do classificador C4.5, o C5.0 oferece vantagens adicionais, como a robustez frente a dados ruidosos e ausentes, além da capacidade de mitigar problemas de sobreajuste por meio de técnicas aprimoradas de poda. Sua eficácia na identificação de atributos relevantes o torna uma ferramenta poderosa para a análise de conjuntos de dados complexos [38].

Como algoritmo baseado em árvores de decisão, o C5.0 classifica o tráfego formando estruturas hierárquicas a partir de estatísticas extraídas das variáveis independentes. No contexto da classificação de tráfego em redes, essas estatísticas são obtidas a partir de dados coletados em ambientes como redes de campus, sendo essenciais para a construção de árvores eficientes. Sua capacidade de lidar com atributos categóricos e contínuos, bem como com ruídos nos dados, garante resultados confiáveis em diferentes aplicações [38].

Na prática, o C5.0 tem sido empregado com sucesso na identificação de aplicações em redes, alcançando taxas de acurácia superiores a 98% em tarefas de classificação [2]. Esse nível de desempenho é alcançado por meio de um processo estatístico de classificação em

duas fases, como descrito por autores que propuseram arquiteturas específicas baseadas no C5.0 para ambientes reais [91, 60].

2.6 Métodos para Seleção de Atributos

A seleção de atributos consiste em escolher um subconjunto de atributos relevantes a partir de um conjunto maior de atributos originais, com base em critérios previamente definidos, como o desempenho de classificação ou a separação entre classes. Essa etapa desempenha um papel significativo em aplicações de aprendizado de máquina. Nesta seção, abordaremos alguns dos métodos, como o RFE (*Recursive Feature Elimination*), o *Boruta*, o *Least Absolute Shrinkage and Selection Operator* - Lasso e o *Random Forest Importance* - RFImp, discutindo suas vantagens, limitações e as circunstâncias em que cada um é mais adequado para ser aplicado.

2.6.1 *Recursive Feature Elimination* - RFE

Recursive Feature Elimination é um algoritmo de seleção de atributos que funciona eliminando gradualmente os atributos menos importantes. O RFE se tornou um método popular para seleção de atributos em diversas aplicações de aprendizado de máquina, como classificação e previsão [61]

O RFE busca melhorar o desempenho de generalização ao remover os atributos menos importantes, cuja exclusão tende a ter o menor impacto no erro de treinamento. Além disso, o RFE está intimamente relacionado às Máquinas de Vetores de Suporte (SVMs), que demonstraram boa capacidade de generalização mesmo em cenários com poucas amostras [16].

No funcionamento do RFE, um modelo de aprendizado de máquina (comumente uma árvore de decisão, *SVM* ou *Random Forest*) é treinado com o conjunto completo de atributos. Em seguida, os atributos são ranqueados de acordo com sua importância no modelo, e os menos relevantes são eliminados. O processo é repetido com o subconjunto remanescente até que se atinja um número pré-definido de atributos ou outro critério de parada. A principal vantagem do RFE é sua capacidade de considerar interações entre variáveis, o que é particularmente útil em conjuntos de dados com alta dimensionalidade [28]

Além disso, o RFE pode ser combinado com validação cruzada, permitindo avaliar o desempenho de diferentes subconjuntos de variáveis ao longo do processo. Essa abordagem, conhecida como *RFECV*, é particularmente útil para determinar automaticamente o número ótimo de atributos a serem mantidos no modelo final [10]

Apesar de sua popularidade e desempenho satisfatório em diversas aplicações, o método RFE apresenta algumas limitações importantes. Uma das principais desvantagens é o seu elevado custo computacional, especialmente quando aplicado a bases de dados com um grande número de atributos e instâncias. Isso ocorre porque o RFE realiza

sucessivas iterações de treinamento do modelo para eliminar gradualmente os atributos menos relevantes, o que pode se tornar inviável em contextos com alta dimensionalidade ou restrições de tempo e recursos computacionais. Além disso, o RFE pode apresentar desempenho subótimo ao eliminar atributos que, embora fracos isoladamente, fornecem informação relevante quando combinados com outros. Conforme apontado por Guyon and Elisseeff [28], atributos redundantes ou aparentemente irrelevantes podem, em conjunto, contribuir para uma melhor separação das classes, e sua exclusão prematura pode prejudicar a capacidade preditiva do modelo final.

2.6.2 *Random Forest Importance* - RFImp

No modelo RF, a importância das variáveis é calculada com base na contribuição de cada variável para a redução do erro de classificação ou regressão. O processo é realizado durante a construção das árvores de decisão, que formam o conjunto do modelo.

A medida de importância das variáveis é tipicamente calculada de duas formas principais: redução da impureza e importância por permutação. A redução da impureza (*Mean Decrease Impurity* - MDI) é uma métrica que quantifica o quanto cada variável contribui para a diminuição da impureza nos nós da árvore. Durante a construção das árvores, as variáveis são usadas para dividir os dados em subconjuntos mais homogêneos. A variável que resulta em uma maior redução da impureza (como o índice de Gini ou a entropia) é considerada mais importante. Essa abordagem calcula a contribuição das variáveis em todas as divisões ao longo das árvores [13] [66].

Além disso, a importância por permutação é outra técnica utilizada para medir a relevância de cada variável. Nesse caso, as variáveis são embaralhadas aleatoriamente e o modelo é reavaliado para verificar o impacto da permutação na acurácia do modelo. Se a acurácia do modelo cair significativamente quando uma variável é permutada, isso indica que a variável é importante para a predição. A importância de uma variável é medida pela redução no desempenho do modelo devido à sua permutação [66].

A partir dessas medidas, é possível classificar as variáveis de acordo com a sua importância relativa. As variáveis mais importantes são aquelas que têm maior impacto na redução do erro ou na melhoria da precisão do modelo. Esse processo de atribuição de importância pode ser utilizado para seleção de atributos, onde as variáveis mais relevantes são mantidas e as menos importantes são descartadas, o que ajuda a reduzir a dimensionalidade do conjunto de dados e a evitar o *overfitting* [43].

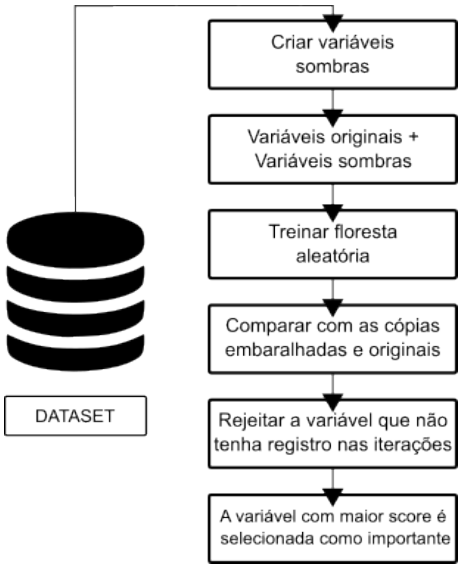
Vantagens do cálculo da importância das variáveis incluem a capacidade de interpretação do modelo, permitindo que os cientistas de dados compreendam quais variáveis estão influenciando as previsões. Além disso, esse processo auxilia na redução da dimensionalidade, pois permite a exclusão de variáveis que não contribuem significativamente para o modelo.

2.6.3 *Boruta*

O algoritmo *Boruta* é uma técnica robusta de seleção de atributos que utiliza florestas aleatórias para identificar variáveis relevantes em conjuntos de dados [44]. O algoritmo funciona criando cópias aleatórias das variáveis originais, denominadas variáveis sombra, e comparando a importância dessas variáveis com as variáveis originais. Se a importância de uma variável original for significativamente maior que a das variáveis sombra, ela é considerada relevante e mantida no modelo [44].

Diferentemente de métodos tradicionais que buscam identificar um subconjunto mínimo de atributos para otimizar um modelo específico, o *Boruta* visa selecionar todos os atributos que são relevantes para a variável dependente, independentemente de sua relação com o modelo utilizado [49].

Figura 6: Diagrama de fluxo representando o funcionamento do algoritmo *Boruta*.



Fonte: Adaptado de [72].

1. O algoritmo cria variáveis sombra por meio de cópias duplicadas do conjunto de dados, com os valores embaralhados em todas as colunas.

2. Os valores originais do conjunto de dados são combinados com as variáveis sombra.

3. Após a combinação, um classificador de florestas aleatórias é utilizado no conjunto de dados combinado, para que a importância das variáveis seja medida. Por padrão, é utilizada a métrica de redução média da acurácia.

4. O *Boruta* verifica a maior importância das variáveis originais ao calcular o *Z-score* e encontrar o maior entre as variáveis sombra.

5. O *Z-score* das variáveis originais e das variáveis embaralhadas são comparados

a cada iteração para verificar qual delas é mais relevante em relação à variável existente.

6. Para aumentar a *robustez*, o algoritmo valida a importância da variável comparando-a com as cópias aleatoriamente embaralhadas.

7. Quanto maior o *Z-score*, maior será a importância da variável.

Apesar de suas vantagens, o *Boruta* também apresenta algumas limitações. O principal desafio é o alto custo computacional, uma vez que o algoritmo requer a construção de múltiplas árvores de decisão e, portanto, pode ser lento em conjuntos de dados grandes. Além disso, o *Boruta* depende de *Random Forest*, que pode ser ineficiente quando o número de árvores cresce muito ou quando o conjunto de dados tem muitas variáveis altamente correlacionadas, o que pode afetar o desempenho do algoritmo [49].

2.6.4 *Least Absolute Shrinkage and Selection Operator* - LASSO

O LASSO (*Least Absolute Shrinkage and Selection Operator*) é uma técnica estatística utilizada para regressão e seleção de variáveis, especialmente em contextos onde o número de preditores é elevado. Introduzido por Tibshirani [78], o método se baseia na penalização L1 aplicada aos coeficientes do modelo de regressão linear, promovendo o encolhimento de alguns coeficientes para zero, o que equivale à exclusão desses atributos do modelo final. Essa característica torna o LASSO não apenas uma ferramenta de regularização, mas também de seleção de variáveis.

Ao minimizar a soma dos erros quadráticos sujeita a uma penalidade proporcional à soma dos valores absolutos dos coeficientes, o LASSO permite controlar diretamente a complexidade do modelo, sendo especialmente eficaz em cenários com dados colineares ou quando o número de variáveis é maior que o número de observações [95]. A abordagem é útil em situações onde se deseja identificar subconjuntos de atributos realmente relevantes para a variável resposta, o que é essencial em aplicações de aprendizado de máquina com dados de alta dimensionalidade.

A formulação matemática do Lasso é dada pela seguinte equação:

$$\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right] \tag{8}$$

Onde:

- y_i são as observações da variável dependente (também conhecida como variável resposta),
- X_i são os atributos (variáveis explicativas) do modelo,
- β_j são os coeficientes dos atributos,

- λ é o parâmetro de regularização, que controla a força da penalização aplicada aos coeficientes.

O parâmetro λ é crucial na escolha do modelo. Quando λ é igual a zero, o Lasso se comporta como uma regressão linear tradicional, sem penalização. À medida que λ aumenta, mais coeficientes são forçados a zero, e o modelo se torna mais esparso.

Em estudos recentes, a aplicação do LASSO tem sido validada em domínios diversos, incluindo o diagnóstico e a predição de doenças, onde a acurácia do modelo e a redução da dimensionalidade são igualmente críticas. Chen et al. [15], por exemplo, demonstraram a eficácia do LASSO na classificação de doenças, mostrando que o método é capaz de selecionar variáveis informativas e eliminar ruídos de forma robusta, mesmo em bases de dados pequenas e com alta correlação entre atributos.

Contudo, é importante destacar que o LASSO pode apresentar limitações quando há grupos de variáveis altamente correlacionadas, favorecendo apenas uma variável do grupo e descartando as demais, mesmo que também sejam relevantes. Para mitigar esse problema, alternativas como o *Elastic Net* foram propostas, combinando penalizações L1 e L2 para equilibrar entre seleção e estabilidade dos coeficientes [95].

2.7 Métricas de Avaliação em Classificação de Dados

A avaliação de algoritmos de classificação é essencial para medir seu desempenho e garantir que os resultados obtidos sejam confiáveis e representativos. Dentre as principais métricas utilizadas destacam-se a tabela de confusão, acurácia, *precision*, *recall*, *F1-Score*, *p-valor* e a AUC/curva ROC. Essas métricas oferecem diferentes perspectivas sobre a qualidade do modelo e devem ser selecionadas conforme o contexto do problema abordado.

2.7.1 Tabela de Confusão

A tabela de confusão é uma ferramenta essencial para avaliar o desempenho de modelos de classificação, especialmente em cenários onde há desbalanceamento entre as classes. É uma matriz que organiza os resultados da classificação em quatro categorias principais: verdadeiros positivos (VP), falsos positivos (FP), verdadeiros negativos (VN) e falsos negativos (FN). A partir desses elementos, é possível calcular diversas métricas de avaliação como precisão, *recall* e acurácia [58].

- ***Verdadeiro Positivo* (TP)**: Casos em que o modelo previu corretamente a classe positiva.

- ***Falso Positivo* (FP)**: Casos em que o modelo previu a classe positiva, mas o valor real era negativo.

- ***Verdadeiro Negativo* (VN)**: Casos em que o modelo previu corretamente a classe negativa.

- ***Falso Negativo* (FN)**: Casos em que o modelo previu a classe negativa, mas o valor real era positivo.

Essa matriz é utilizada na avaliação de algoritmos de aprendizado de máquina, especialmente na análise do desempenho em conjuntos de dados desbalanceados. É uma ferramenta para compreender os erros cometidos por um modelo e identificar possíveis melhorias no processo de treinamento e validação [18].

Utilizada para calcular métricas como precisão, *recall* e acurácia. Por exemplo, no diagnóstico de Alzheimer em pacientes com COVID-19, a análise da tabela de confusão foi importante para identificar se o modelo apresentava maior tendência a gerar falsos negativos, comprometendo a segurança clínica [5]. Na área da segurança e da detecção de fraudes, a tabela de confusão permite identificar com clareza se um sistema de monitoramento está gerando um número excessivo de falsos positivos ou falsos negativos. Um elevado número de falsos negativos indica que transações fraudulentas estão sendo ignoradas, enquanto um excesso de falsos positivos pode resultar em clientes legítimos sendo indevidamente bloqueados. A avaliação da tabela de confusão permite o ajuste de parâmetros do modelo para equilibrar essas métricas e garantir uma solução mais adequada [58, 94].

Vamos considerar um exemplo prático no contexto da predição de lesões esportivas em praticantes de *cross training*. Os valores apresentados na Tabela **??**, exemplificam a estrutura de uma matriz de confusão para o problema de classificação binária. Supondo que, após testar o modelo com 100 praticantes, os resultados reais sejam os seguintes:

Tabela 2: Exemplo de Tabela de Confusão para Predição de Lesões Esportivas

Predição	**Lesionado (real)**	**Não Lesionado (real)**
Lesionado (previsto)	45 (VP)	10 (FN)
Não Lesionado (previsto)	5 (FP)	40 (VN)

- 55 praticantes realmente sofreram lesão.

- 45 praticantes não sofreram lesão.

As previsões realizadas pelo modelo foram:

- 45 praticantes foram corretamente classificados como lesionados (VP).

- 5 praticantes foram classificados incorretamente como lesionados (FP).

- 40 praticantes foram corretamente classificados como não lesionados (VN).
- 10 praticantes foram classificados incorretamente como não lesionados (FN).

Além disso, a tabela de confusão é empregada na recuperação de informações e na classificação de imagens. Em mecanismos de busca, por exemplo, essa matriz permite identificar se documentos irrelevantes estão sendo recuperados com frequência (falsos positivos) ou se documentos relevantes estão sendo omitidos (falsos negativos), ajudando a ajustar o modelo conforme o objetivo desejado. Da mesma forma, na classificação de imagens para detecção de lesões dermatológicas, a tabela de confusão é importante para avaliar se o modelo está errando principalmente na detecção de casos positivos ou negativos, fornecendo *insights* para aprimorar seu desempenho [8, 33].

2.7.2 Acurácia

A acurácia é uma das métricas mais comuns na avaliação de modelos de classificação e representa a proporção de previsões corretas realizadas pelo modelo, considerando tanto os acertos nas classes positivas quanto nas negativas [52]. Em outras palavras, mede o quão frequentemente o classificador está correto em relação ao total de observações avaliadas.

Matematicamente, a acurácia é definida como:

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{9}$$

onde:

- VP é o número de verdadeiros positivos;
- VN é o número de verdadeiros negativos;
- FP é o número de falsos positivos;
- FN é o número de falsos negativos.

Apesar de sua simplicidade e popularidade, a acurácia pode ser enganosa em conjuntos de dados desbalanceados, nos quais uma classe é significativamente mais frequente que a outra. Nesses casos, um modelo que simplesmente prediz sempre a classe majoritária pode obter alta acurácia, mas sem oferecer real capacidade de generalização ou valor preditivo sobre a classe minoritária [59].

2.7.3 *Precision* e *Recall*

A métrica *Precision* (ou Precisão) refere-se à proporção de instâncias classificadas como positivas que realmente pertencem à classe positiva. Em contrapartida, a métrica *Recall* (ou Revocação) indica a proporção de instâncias positivas que foram corretamente

identificadas como tal. Ambas são especialmente relevantes em cenários onde se busca um equilíbrio entre evitar falsos positivos e identificar corretamente todas as instâncias relevantes [94, 33].

A *Precision* é definida como:

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

Enquanto o *Recall* é expresso por:

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

As métricas *Precision* e *Recall* são empregadas em diversos cenários, especialmente em aplicações que exigem uma avaliação precisa da detecção de eventos. São utilizadas em sistemas de informação, onde se busca avaliar a qualidade da recuperação de documentos relevantes [58]. No contexto de detecção de fraudes e diagnóstico médico, o *Recall* é especialmente relevante, uma vez que a identificação correta de todas as instâncias positivas é importante [94]. Por exemplo, em detecção de doenças, um alto *Recall* é desejável para identificar a maior quantidade possível de casos positivos, enquanto um alto *Precision* é essencial para minimizar falsos alarmes [58].

Essas métricas assim ajudam na identificação de doenças graves. Em possíveis diagnósticos como Lesões esportivas, um alto *Recall* é crucial para identificar a maior quantidade possível de casos positivos, minimizando o risco de deixar pacientes críticos sem tratamento. Por outro lado, a *Precision* se destaca quando o objetivo é reduzir falsos positivos, como em testes de triagem para doenças raras, onde falsos alarmes podem sobrecarregar o sistema de saúde com exames desnecessários [5]. Da mesma forma, na detecção de doenças como a varíola dos macacos por meio de redes neurais convolucionais, essas métricas foram fundamentais para avaliar a precisão na distinção de lesões cutâneas específicas de outras condições dermatológicas [8].

Em sistemas de recomendação e segurança, essas métricas também desempenham papéis importantes. Em plataformas de recomendação, como serviços de *streaming* ou *e-commerce*, a *Precision* garante que as recomendações apresentadas ao usuário sejam altamente relevantes, melhorando a experiência do usuário. No entanto, quando o objetivo é cobrir o maior número possível de opções relevantes, o *Recall* se torna mais relevante. Já na detecção de fraudes, como em bancos e sistemas financeiros, um alto *Recall* é essencial para minimizar o risco de transações fraudulentas passarem despercebidas. Contudo, um alto *Recall* pode gerar falsos positivos, sendo necessária uma boa *Precision* para reduzir investigações desnecessárias e otimizar custos operacionais [94, 58].

2.7.4 *F1-Score*

O *F1-Score* é utilizado em cenários onde há a necessidade de equilibrar a precisão e a abrangência em sistemas de classificação. Essa métrica é especialmente relevante em áreas como diagnóstico médico, onde identificar corretamente todos os casos positivos é essencial, mas a redução de falsos positivos também é importante. O *F1-Score* é a média harmônica entre *Precision* e *Recall*, sendo uma métrica que combina ambas as medidas em uma única pontuação. Essa métrica é especialmente útil quando há um desequilíbrio significativo entre as classes e quando tanto a precisão quanto o *recall* são igualmente importantes. Ele é útil quando se deseja equilibrar a taxa de verdadeiros positivos e a precisão do modelo. [18]. O *F1-Score* é definido como:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{12}$$

Essa métrica é empregada em tarefas de classificação com conjuntos de dados desbalanceados, onde a avaliação isolada da *Precision* ou do *Recall* pode ser insuficiente para representar corretamente o desempenho do modelo [18].

Por exemplo, na detecção de Alzheimer em pacientes com COVID-19, o *F1-Score* foi utilizado para avaliar o desempenho de um modelo híbrido de aprendizado profundo, apresentando resultados robustos mesmo em conjuntos de dados desbalanceados [5]. Da mesma forma, em sistemas de detecção de varíola dos macacos com aprendizado profundo, o *F1-Score* foi empregado para avaliar a precisão e abrangência da classificação de lesões cutâneas, destacando-se como uma métrica adequada para avaliar modelos que enfrentam desafios de equilíbrio entre classes [8].

Em ambientes industriais e de segurança, o *F1-Score* é essencial para avaliar a eficácia de sistemas de monitoramento e detecção de falhas. Essa métrica é particularmente eficaz quando há um forte desequilíbrio entre classes, como na detecção de eventos raros. Por exemplo, em sistemas de prevenção de fraudes bancárias, o *F1-Score* permite identificar um bom equilíbrio entre a identificação de atividades suspeitas e a redução de falsos alertas, minimizando impactos operacionais desnecessários [18]. Além disso, na área de aprendizado de máquina aplicado à genômica, o *F1-Score* é utilizado para avaliar modelos que classificam pacientes com base em suas características genéticas, destacando-se como uma alternativa eficaz à métrica de acurácia, especialmente quando o conjunto de dados apresenta classes com distribuições desiguais [18].

Outro campo em que o *F1-Score* se destaca é na recuperação de informações e em sistemas de recomendação. Nessas áreas, essa métrica é útil para avaliar a qualidade das recomendações, especialmente em cenários onde a precisão isolada pode resultar na perda de documentos relevantes ou onde o foco exclusivo no *Recall* pode incluir uma grande quantidade de resultados irrelevantes. O uso do *F1-Score* garante que tanto a precisão quanto a abrangência sejam equilibradas, proporcionando uma avaliação mais completa

do desempenho desses sistemas [58].

2.7.5 *P-valor*

O p-valor é uma métrica estatística fundamental utilizada em testes de hipóteses para determinar a evidência contra uma hipótese nula [77]. Ou seja, ele mede a força da evidência contra a hipótese nula. Quanto menor o p-valor, mais forte é a evidência contra a hipótese nula.

O p-valor obtido a partir de uma tabela de confusão mede a probabilidade de que a relação observada entre as classes preditas e as classes reais seja devido ao acaso. É utilizado para testar a hipótese nula de que não existe relação entre as classes preditas e reais, ou seja, o modelo não tem poder preditivo. Quando o p-valor é pequeno (geralmente menor que 0.05), isso indica que a hipótese nula pode ser rejeitada, sugerindo que o modelo de classificação tem um desempenho significativamente melhor do que o esperado por acaso [31].

No contexto de uma tabela de confusão, o p-valor é frequentemente calculado por meio de um teste estatístico, como o teste qui-quadrado, que compara as frequências observadas com as esperadas sob a hipótese nula. Contudo, assim como em qualquer teste estatístico, a interpretação do p-valor deve ser feita com cautela, pois valores próximos de 0.05 podem indicar resultados marginalmente significativos [25].

Além disso, enquanto o p-valor oferece uma visão sobre a significância do desempenho do modelo, ele não fornece informações sobre a magnitude do efeito ou sobre a qualidade do modelo. Para uma avaliação mais completa, é importante considerar outras métricas, como AUC, precisão, recall e F1-score, que fornecem uma visão mais detalhada sobre o desempenho do modelo de classificação [30].

2.7.6 AUC e Curva ROC

A métrica *Area Under the Curve* (AUC) e a curva *Receiver Operating Characteristic* (ROC) são ferramentas usadas na avaliação de modelos de classificação, especialmente em cenários onde é necessário analisar o comportamento do modelo sob diferentes pontos de decisão. A métrica AUC refere-se à área sob a curva ROC, que representa graficamente a relação entre a taxa de verdadeiros positivos (TPR) e a taxa de falsos positivos (FPR) em diferentes limiares de decisão. O valor da AUC varia entre 0 e 1, sendo que valores mais próximos de 1 indicam melhor desempenho do modelo [58].

A curva ROC é um gráfico que representa a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos, variando o limiar de decisão. A AUC quantifica a capacidade do modelo de distinguir entre classes positivas e negativas, sendo que valores próximos de 1 indicam excelente desempenho. Essa abordagem foi aplicada, por exemplo, na detecção de lesões causadas pela varíola dos macacos, onde o modelo apresentou uma AUC de 0,99, destacando sua elevada capacidade discriminativa [8].

O gráfico apresentado na Figura 7 exibe as curvas ROC de três modelos com diferentes desempenhos, representados pelas cores vermelha (AUC = 0.90), verde (AUC = 0.66) e azul (AUC = 0.53), além da linha de referência de AUC = 0.5. A curva ROC ilustra a relação entre a Taxa de Falsos Positivos (FPR) e a Taxa de Verdadeiros Positivos (TPR), e a AUC quantifica a capacidade do modelo em distinguir entre as classes. O modelo vermelho (bom) apresenta uma AUC alta, indicando excelente desempenho, enquanto o modelo verde (médio) tem um desempenho moderado, e o modelo azul (ruim) possui uma AUC baixa, sugerindo um desempenho quase aleatório. A linha pontilhada preta representa o desempenho aleatório (AUC = 0.5). Vale ressaltar que os dados utilizados para gerar essas curvas são aleatórios e servem apenas para fins ilustrativos.

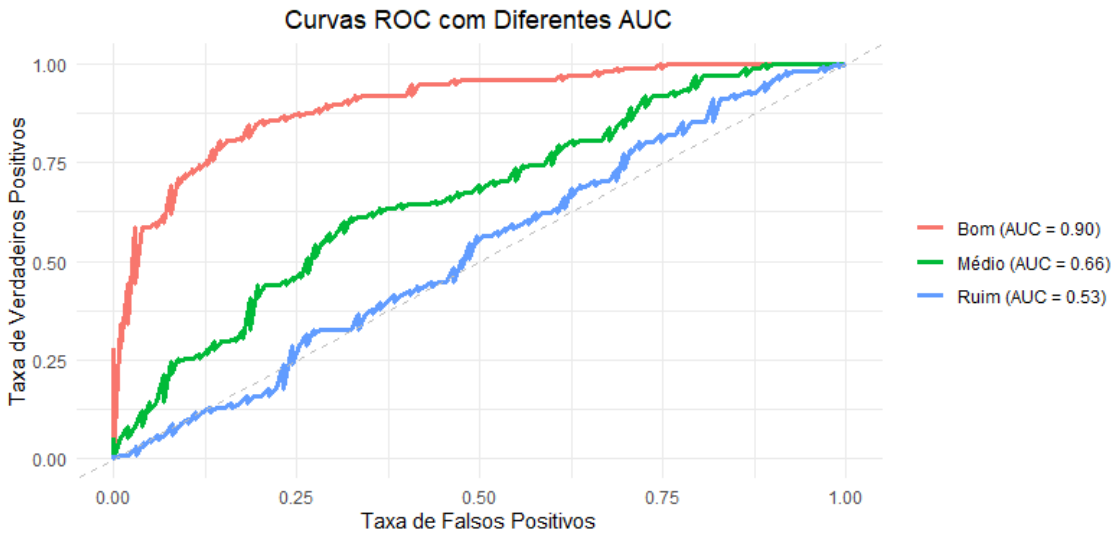


Figura 7: Curvas ROC com diferentes AUCs.

A curva ROC é usada para avaliar modelos em cenários onde os custos associados a erros de classificação variam significativamente [8]. Essa abordagem é utilizada na área da saúde para determinar pontos de corte que equilibrem a sensibilidade e a especificidade do modelo. Essa análise é particularmente aplicada em cenários críticos, como na detecção precoce de doenças neurodegenerativas relacionadas à COVID-19 [5]. Ao permitir essa avaliação gráfica, a curva ROC proporciona uma visão clara sobre o impacto de diferentes pontos, auxiliando na escolha da configuração mais adequada ao contexto clínico. Em modelos de diagnóstico, a curva ROC permite identificar o ponto ideal que maximiza a detecção de casos positivos (*Recall*) ao mesmo tempo que reduz os falsos alarmes (*Precision*) [26].

3 TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns estudos relacionados ao tema da pesquisa: Técnicas de aprendizado de máquina para predição. São abordados diversos algoritmos utilizados no processo de AM e suas aplicações. Por último, uma análise comparativa é realizada entre os estudos.

3.1 Estudos sobre predição de lesões

A Tabela 3 mostra alguns estudos relacionados com AM na predição de lesões em diversos esportes, bem como os objetivos, algoritmos e demais informações.

No estudo desenvolvido por Moustakidis et al. [53], os autores destacam o crescente reconhecimento global do CrossFit entre populações fisicamente ativas, mas levanta preocupações sobre os riscos de lesões devido à natureza intensa do treinamento. Com objetivo de identificar fatores de risco e desenvolver modelos de aprendizado de máquina para prever lesões, foi realizado um estudo na Grécia onde os dados sobre lesões musculoesqueléticas foram coletados entre praticantes da modalidade. O fluxo do aprendizado de máquina envolveu pré-processamento, seleção de características e a aplicação de cinco algoritmos para AM. O melhor modelo alcançou uma AUC (*Area Under the Curve*) de 77,93% usando o algoritmo Adaboost com seis fatores de risco selecionados. A abordagem proposta foi avaliada em métricas como precisão, sensibilidade e especificidade, mostrando grande relevância.

Wu et al. [89] desenvolveram um estudo sobre um sistema de diagnóstico de lesões esportivas baseados em nuvem, utilizando técnicas avançadas de aprendizado profundo. O sistema busca melhorar a precisão no diagnóstico de lesões esportivas por meio da implementação de algoritmos de aprendizado profundo em um ambiente de computação em nuvem. Essa abordagem visa proporcionar uma solução mais eficiente e acessível para o diagnóstico preciso de lesões esportivas, aproveitando a potência computacional disponível na nuvem e os avanços em técnicas de aprendizado profundo. Os autores utilizaram 45 jogadores de futebol para a condução do estudo. A aplicação de DL foi comparada com modelos convencionais de AM que utilizam algoritmos como, *Neural*

Tabela 3: Estudos relacionados com AM na predição de lesões em diversos esportes

Estudo	Objetivo	Área de aplicação	Algoritmos	Nº de amostras	Nº de atributos
Moustakidis et al. [53]	Aplicação de AM para predição de lesões	Crossfit®	SVM, LR, DT, KNN, Adaboost, RF	1224	33
Wu et al. [89]	Uso de *deep learning-assisted* na predição de lesões	Futebol	DLS	45	-
Lövdal et al. [47]	Predição de lesões a partir de AM	Corredores	XGBoost	74	-
Huang and Jiang [36]	Desenvolvimento de modelo de predição de lesões	Jogadores de futebol	ANN	21	-
Jauhiainen et al. [37]	Predição de lesões em jovens atletas utilizando AM	Jogadores de futebol e *floorball*	L1 - *regularized logistic regression*, RF)	314	54
Naglah et al. [54]	Aplicação de AM para predição de lesões	Jogadores de futebol	KNN, SVM	21	65
Oliver et al. [56]	Utilização de AM para o entendimento de lesões em esportes de elite	Jogadores de futebol	DT, J48 *consolidated* (J48con), *Alternating Decision Tree* (ADT) e *Reduces Error Pruning Tree* (REPTree)	355	20
Henriquez et al. [34]	Predição de lesoes muscoesqueléticas utilizando AM	Atletas diversos	RF	122	-
Shringarpure et al. [68]	Uso de AM para predição de lesões	Hóquei	RF	-	-
Rommers et al. [64]	Uso de AM para avaliar o risco de lesões	Jogadores de futebol	DT	734	29

Networks (NN), *Fuzzy Logic* (FL), *Decision Trees* (DT), e *Random Forests* (RF). Os autores concluem que o método de aprendizado profundo apresenta melhor desempenho na predição de lesões que os modelos convencionais mais utilizados.

O estudo discorrido por Shringarpure et al. [68], sugere um sistema de previsão de lesões esportivas utilizando o algoritmo *Random Forest*. O sistema busca antecipar a ocorrência de lesões em atletas, empregando um modelo preditivo baseado no RF, um algoritmo de aprendizado de máquina que utiliza uma abordagem de árvores de decisão. O estudo avalia uma base de dados disponível contendo praticantes de hóquei durante uma temporada de jogos. Os autores concluem que o modelo é robusto podendo ser utilizado para diversos esportes, assim predizendo problemas e gerando economia na indústria esportiva.

Henriquez et al. [34] propõem o uso de aprendizado de máquina para prever o risco de lesões musculoesqueléticas nos membros inferiores em estudantes atletas. Utilizando o RF, o estudo desenvolve um modelo preditivo que identifica fatores de risco e antecipam a probabilidade de lesões. O objetivo é aprimorar a prevenção e gestão de lesões em atletas estudantis por meio da aplicação de abordagens avançadas de análise de dados e aprendizado de máquina. Para o desenvolvimento do trabalho uma população de 122 atletas foi utilizada, dentre eles, homens e mulheres, além de praticantes de diferentes esportes. No entanto, este estudo apresenta algumas limitações de trabalho como, o pequeno tamanho da amostra (122 alunos/atletas) e a ausência de características voltados para alunos como os hábitos nutricionais dos atletas, fatores de estresse e estatísticas de jogo.

Jauhiainen et al. [37] realizaram um estudo acerca da predição de lesões em jovens atletas utilizando aprendizado de máquina. Os autores aplicaram *Logistic Regression* e *Random Forest* para o desenvolvimento da predição. Foram utilizados 314 atletas no estudo e 54 características foram avaliadas. Os autores concluem que a AUC obtida foi de 65%, sendo considerada relativamente baixa e não foram verificadas diferenças entre o modelo linear e não-linear. Embora o modelo ainda seja capaz de prever as lesões pesquisas futuras foram sugeridas para maiores avaliações dos métodos de AM.

3.2 Análise comparativa

A análise comparativa dos estudos revela tendências comuns e distinções importantes sobre o uso de aprendizado de máquina e aprendizado profundo para a predição de lesões em atletas, mostrando tanto avanços significativos quanto limitações. Ao considerar os trabalhos de Moustakidis et al. [53], Wu et al. [89], Shringarpure et al. [68], Henriquez et al. [34], e [37], é possível observar a diversidade de abordagens e contextos, bem como a aplicação de diferentes algoritmos que, em conjunto, oferecem um panorama relevante da literatura sobre predição de lesões com técnicas de AM.

Primeiramente, os estudos de Moustakidis et al. [53] e Wu et al. [89] compartilham

o foco em modalidades esportivas de alta intensidade (CrossFit® e futebol, respectivamente). No entanto, suas abordagens divergem significativamente. Enquanto o primeiro utilizou algoritmos mais tradicionais, como Adaboost, e obtiveram uma AUC elevada de 77,93%, o segundo focou em técnicas de aprendizado profundo implementadas em um ambiente de computação em nuvem. Essa última abordagem permitiu que Wu et al. [89] alcançasse maior precisão no diagnóstico de lesões, superando modelos convencionais como Redes Neurais e *Random Forest*. Esse contraste destaca a crescente relevância do aprendizado profundo em cenários onde há uma vasta quantidade de dados e grande necessidade de processamento computacional.

Em paralelo, o estudo de Shringarpure et al. [68] também utiliza o *Random Forest*, mas aplica esse algoritmo a uma população de atletas de hóquei. Uma das contribuições mais relevantes desse trabalho é a sua adaptabilidade a diferentes esportes, sugerindo que o *Random Forest* pode ser um modelo generalista eficaz para a predição de lesões. No entanto, uma limitação comum a este e ao estudo de Henriquez et al. [34] é o tamanho reduzido da amostra, que pode comprometer a robustez dos modelos preditivos em termos de generalização. O estudo de Henriquez et al. [34] esbarra em uma limitação crítica: a falta de fatores complementares, como hábitos nutricionais e indicadores de estresse, que poderiam fornecer uma visão mais holística das causas de lesões.

Já o estudo de Jauhiainen et al. [37] traz uma perspectiva mais cautelosa sobre o uso de aprendizado de máquina na predição de lesões. Apesar de aplicarem *Random Forest* e Regressão Logística, ambos os modelos resultaram em uma AUC de 65%, indicando um desempenho relativamente modesto. Esse resultado sugere que, para certos contextos, os modelos tradicionais de AM podem ter limitações na predição de lesões, principalmente quando há uma alta complexidade de fatores envolvidos e as características dos dados não são suficientemente discriminativas.

Na literatura sobre predição de lesões com Aprendizado de Máquina, há um consenso de que, embora os modelos convencionais de AM (como *Random Forest* e Regressão Logística) sejam úteis e amplamente aplicados, o uso de Aprendizado Profundo oferece resultados promissores, especialmente em situações onde há grandes volumes de dados complexos, como no estudo de Wu et al. [89]. No entanto, o Aprendizado Profundo também demanda recursos computacionais substancialmente maiores, o que pode limitar sua aplicabilidade em contextos onde esses recursos não estão disponíveis. Estudos recentes apontam que técnicas de seleção de características, como a implementada por Moustakidis et al. [53], também desempenham um papel crucial ao melhorar a precisão dos modelos, uma vez que a inclusão de variáveis irrelevantes pode prejudicar o desempenho preditivo.

Além disso, os trabalhos indicam que o sucesso na predição de lesões depende fortemente da qualidade dos dados e das características escolhidas para o modelo. Fatores como histórico de lesões, tipo de esporte, intensidade de treinamento e perfil físico do

atleta são frequentemente identificados como essenciais para a construção de modelos eficazes. No entanto, como destacado por Henriquez et al. [34], a ausência de dados contextuais adicionais, como alimentação e fatores psicológicos, pode limitar a capacidade preditiva dos modelos, sugerindo a necessidade de uma abordagem mais multidimensional.

Portanto, ao relacionar os estudos, fica claro que o campo da predição de lesões está evoluindo em direção ao uso de métodos mais sofisticados, como o aprendizado profundo, enquanto modelos convencionais de AM ainda oferecem uma base sólida, especialmente em contextos onde os recursos computacionais são limitados. Ao mesmo tempo, a literatura aponta que o sucesso desses modelos depende não apenas da técnica utilizada, mas também da qualidade e da relevância dos dados coletados. Assim, futuras pesquisas devem continuar explorando novas técnicas e integrar mais variáveis contextuais, a fim de desenvolver modelos mais precisos e generalizáveis para a predição de lesões esportivas.

4 METODOLOGIA

Neste capítulo, será detalhada a metodologia empregada para a condução deste trabalho, aplicando os conceitos teóricos do processo de descoberta de conhecimento em bancos de dados previamente apresentado na Seção 2.3. Todo o desenvolvimento deste trabalho foi realizado em um computador pessoal, no ambiente *RStudio*, utilizando a linguagem R e diversas bibliotecas para manipulação, modelagem e análise dos dados. A seguir, são descritas as etapas realizadas, desde a obtenção dos dados até as técnicas de modelagem preditiva.

4.1 Fonte dos Dados

A base de dados utilizada foi proveniente de uma pesquisa sobre lesões associadas ao treinamento funcional de alta intensidade, do inglês, *High Intensity Functional Training* (HIFT), conduzida por Serafim et al. [67]. A coleta de dados foi realizada entre janeiro e maio de 2021, por meio de um questionário *online* divulgado nas redes sociais, com foco em praticantes de HIFT maiores de 18 anos. O questionário continha perguntas sobre as características demográficas, o nível e o tipo de treinamento, a prática de outras atividades físicas e o histórico de lesões dos participantes.

Os dados coletados incluíram variáveis como a frequência semanal de treinamento, os objetivos do treinamento, o uso de aquecimento e desaquecimento, o envolvimento em competições, além de informações detalhadas sobre a ocorrência de lesões, incluindo localização, severidade e fatores associados. A base de dados foi utilizada para análises descritivas e preditivas que embasaram os experimentos realizados no presente estudo.

4.1.1 Características do Conjunto de Dados

O conjunto de dados obtido possui 673 (seiscentos e setenta e três) observações e é composto por 32 (trinta e dois) atributos, sendo predominantemente categóricos, o que corresponde a aproximadamente 78,1% das variáveis. Entre eles, a maioria é do tipo nominal, representando 62,5% do total, enquanto os atributos ordinais constituem 15,6%. Já os atributos numéricos somam 21,9% do *dataset*, sendo 15,6% discretos e

6,3% contínuos.

4.1.2 Descrição dos atributos

Nesta seção, serão apresentados de forma detalhada os principais atributos que compõem o *dataset* utilizado para a análise. Cada atributo desempenha um papel fundamental na compreensão do perfil dos participantes e das possíveis relações com a ocorrência de lesões na prática do *cross training*. A seguir, a Tabela 4 resume todos os atributos presentes no *dataset*, descrevendo suas características e fornecendo uma visão completa dos dados disponíveis para o estudo.

Tabela 4: Detalhamento dos atributos presentes no *dataset*.

Atributo	**Tipo**	**Min**	**Máx**	**Med/Moda**	**Descrição**
DATA_HORA	Data	NA	NA	2021-04-14, 2021-04-16	Data e hora de coleta da resposta
EMAIL	Categórico	NA	NA	isa***.ama*** @***.com	Endereço de e-mail para contato
CONSENTIMENTO	Categórico	NA	NA	Concordo	Confirmação de consentimento para participação da pesquisa
NOME	Categórico	NA	NA	Isa*** So***	Nome completo do participante
DATA _NASCIMENTO	Data	NA	NA	1990-06-19	Data de nascimento do participante
CIDADE_ATUAL	Categórico	NA	NA	Florianópolis	Cidade onde reside o participante
ALTURA	Contínuo	149	193	169,56	Altura do participante
PESO	Contínuo	48,8	118	73,59	Peso do participante
SEXO	Categórico	NA	NA	Feminino	Sexo do participante
EXPERIENCIA	Discreto	0	120	25,21	Tempo que o participante pratica Crossfit®
OBJETIVO	Categórico	NA	NA	Saúde	Objetivo do participante com a prática do Crossfit®
MEDIA_DE_DIAS _TREINADOS	Categórico	NA	NA	5-6 dias/semana	Média de dias treinados no Crossfit®, durante 1 semana, considerando as últimas 4 semanas
MEDIA_DE_HORAS _TREINADAS	Categórico	NA	NA	5-6h	Média de horas treinadas no Crossfit®, durante 1 semana, considerando as últimas 4 semanas
AQUECIMENTO	Categórico	NA	NA	Sim	Presença de aquecimento nos treinos de Crossfit®
DESAQUECIMENTO	Categórico	NA	NA	Não	Presença de desaquecimento nos treinos de Crossfit®
QTD_ALUNOS _AULA	Discreto	1	202	10,9	Média de alunos presentes durante uma aula de Crossfit®
QTD_PROF _AULA	Discreto	0	10	1,71	Média de professores presentes em uma aula de Crossfit®
TIPO_TREINO	Categórico	NA	NA	Quadro/Aula normal	Método de treinamento realizado pelo participante
COMPETICAO	Categórico	NA	NA	Não	Participação em competições de Crossfit®
NIVEL _COMPETICAO	Categórico	NA	NA	Não se aplica	Nível de abrangência das competições realizadas
CATEGORIA _COMPETICAO	Categórico	NA	NA	Não se aplica	Categoria em que o participante costuma competir

Atributo	Tipo	Min	Máx	Med/Moda	Descrição
ATIVIDADE _EXTRA	Categórico	NA	NA	Sim	Realização de exercício físico paralelamente ao Crossfit®
TIPO_ATIVIDADE _EXTRA	Categórico	NA	NA	-	Tipo de exercício físico realizado paralelamente ao Crossfit®
LESAO	Categórico	NA	NA	Sim	Ocorrência de lesão durante a prática de Crossfit®
AVISO	Categórico	NA	NA	Ok	Aviso de orientação em respeito às próximas perguntas
LOCAL_LESAO	Categórico	NA	NA	Não se aplica	Parte do corpo em que ocorreu a lesão
LESAO_ANTERIOR	Categórico	NA	NA	Não se aplica	Existência de lesão anterior ao treinamento de Crossfit®
LESAO _REPERCUSSAO	Categórico	NA	NA	Não me machuquei (Não se aplica)	Tipo de repercussão nos treinos, causada pela lesão
LESAO_TEMPO _SEM_TREINO	Categórico	NA	NA	Não se aplica	Duração (em dias) do período sem treinos, em decorrência da lesão
LESAO _MOVIMENTO _ASSOCIADO	Categórico	NA	NA	-	Movimento do Crossfit® que o participante associa à ocorrência da lesão
LESAO _TRATAMENTO	Categórico	NA	NA	Não se aplica	Tipo de tratamento realizado para tratar a lesão
LOCAL_TREINO	Categórico	NA	NA	São *** Crossfit® Fl***	Cidade atual e nome do local onde treina Crossfit®

A predominância de dados categóricos, especialmente nominais, evidencia a necessidade de técnicas adequadas para o tratamento e análise de variáveis qualitativas, como *one-hot encoding*, além de métodos apropriados de pré-processamento e modelagem. A presença de variáveis numéricas, apesar de menor, também exige atenção, especialmente no que tange à normalização e padronização dos valores contínuos para garantir uma melhor integração com as variáveis categóricas no modelo preditivo.

4.2 Pré-processamento dos Dados

Dados reais geralmente apresentam problemas como inconsistências, valores ausentes, ruído, redundância ou formatos inadequados, que podem prejudicar a qualidade dos resultados das análises. Visando a preparação dos dados para que se tornem adequados às etapas posteriores, algumas técnicas de pré-processamento foram aplicadas para garantir que o conjunto de dados fosse limpo, consistente e relevante para a pesquisa.

4.2.1 Divisão Inicial dos Dados

Antes da aplicação das etapas de pré-processamento, o conjunto de dados original foi particionado em dois subconjuntos: treino e teste. A divisão foi realizada de forma estratificada, preservando a proporção da variável-alvo “LESAO” em ambos os conjuntos.

Essa separação inicial é fundamental para evitar o vazamento de informações do con-

junto de teste durante o processo de preparação dos dados, garantindo que os parâmetros utilizados para imputação, transformação e normalização sejam obtidos exclusivamente a partir do conjunto de treino.

```
library(caret)

# Divisao estratificada dos dados em treino (70%) e teste (30%)
set.seed(123)
particao <- createDataPartition(dataset$Lesao, p = 0.7, list = FALSE)
trainData <- dataset[particao, ]
testData <- dataset[-particao, ]
```

Figura 8: Divisão estratificada dos dados

O código da Figura 8 apresenta a divisão do conjunto de dados por meio da função `createDataPartition()`, disponível no pacote `caret`. Esse procedimento assegura uma separação estratificada, preservando a distribuição da variável-alvo "LESAO" em ambos os subconjuntos. A divisão foi realizada com 70% das observações destinadas ao conjunto de treino e os 30% restantes ao conjunto de teste.

4.2.2 Limpeza dos dados

A etapa de limpeza dos dados foi fundamental para garantir a qualidade dos dados e evitar que valores inconsistentes ou faltantes afetassem os resultados da análise. A limpeza foi realizada com base em critérios específicos para valores ausentes, ruídos nos dados e entradas duplicadas.

4.2.2.1 Remoção de Duplicatas

Durante o processo de pré-processamento, foi realizada uma verificação para identificar registros duplicados no conjunto de dados. A detecção foi realizada por meio da função *duplicated()* da linguagem R, que permite identificar observações completamente idênticas com base em todas as variáveis disponíveis.

Este procedimento contribui para a melhoria da qualidade do conjunto de dados, assegurando maior confiabilidade nas etapas posteriores de análise exploratória e modelagem preditiva.

4.2.2.2 Tratamento de Valores Ausentes

Para preencher esses valores, utilizou-se a imputação pela média dos dados de treino, levando em consideração a distribuição dos dados por sexo, para garantir maior precisão e relevância. Ou seja, para os valores ausentes de Idade e Altura, a média dos dados de treinamento foi calculada separadamente para homens e mulheres e usada para substituir os dados faltantes no *dataset* de treino e teste.

4.2.2.3 Eliminação de atributos

Nesta etapa, foi realizada uma redução de dimensionalidade por meio da remoção manual de colunas que não seriam relevantes para os objetivos da análise.

Foram removidos atributos de caráter pessoal e outras informações identificáveis, visando garantir a privacidade dos participantes. Além disso, como o foco do estudo é predizer a ocorrência de lesões, informações relacionadas ao tratamento das lesões e ao acompanhamento posterior à ocorrência foram excluídas.

4.2.3 Transformação dos Dados

Visando adaptar os dados para o formato e a estrutura que melhor se ajustem aos algoritmos de análise e aprendizado de máquina, durante este estudo, algumas transformações (normalização e codificação) foram aplicadas para garantir a consistência e a qualidade da base de dados.

4.2.3.1 Atributos Contínuos

Os atributos contínuos da base de dados passaram por um processo de padronização com o objetivo de uniformizar seus formatos e corrigir possíveis inconsistências. A variável “EXPERIENCIA”, preenchida de forma textual (ex: “1 ano e meio”, “dois anos e 6 meses”), foi convertida em número de meses. Para isso, foi criada uma função que identificava e transformava os valores escritos por extenso, além de detectar expressões relacionadas a anos, meses e até mesmo o termo “meio”, somando 6 meses ao total. Essa transformação resultou em uma variável numérica contínua, facilitando a análise posterior.

A variável “IDADE” foi calculada a partir da data de nascimento fornecida, utilizando uma data de referência fixa.

A variável “ALTURA”, inicialmente preenchida com variações como “1,75”, “175” ou “1.75m”, foi padronizada para o formato em centímetros, com eliminação de vírgulas, pontos e sufixos. De forma semelhante, o “PESO” foi tratado para remover unidades como “kg”, padronizar o uso de vírgulas como separador decimal e calcular médias em casos de múltiplos valores indicados (ex: “92/93”). Por fim, as variáveis “QTD_PROFS_AULA” e “QTD_ALUNOS_AULA”, preenchidas de maneira textual ou intervalar, foram transformadas em valores numéricos utilizando uma função que identifica números por extenso e calcula médias em casos de expressões como “3 a 5” ou “4/5”.

Tabela 5: Transformações aplicadas aos atributos contínuos do conjunto de dados

Atributo	**Transformação**	**Unidade Padronizada**	**Observações**
ALTURA	Normalização	Centímetros (cm)	Remoção de ruídos textuais
PESO	Normalização	Quilogramas (kg)	Remoção de valores inconsistentes
EXPERIÊNCIA	Normalização	Meses	Conversão e limpeza textual
QTD ALUNOS AULA	Conversão	Numérica	Extração de números válidos
QTD PROF AULA	Conversão	Numérica	Extração de números válidos
IDADE	Cálculo derivado	Anos	Obtida a partir de "DATA NASCIMENTO"

4.2.3.2 Atributos Categóricos

A padronização dos atributos categóricos teve como objetivo principal eliminar variações semânticas e textuais, além de converter os dados para estruturas adequadas à modelagem. A variável "SEXO", por exemplo, foi recodificada a partir de valores como "Feminino", "Faminino" e "Masculino", gerando os rótulos padronizados "F" e "M" e posteriormente convertida em fator.

Os atributos "MEDIA_DE_DIAS_TREINADOS", "MÉDIA_DE_HORAS_TREINADAS", "NIVEL_COMPETICAO e "CATEGORIA_COMPETICAO" foram convertidos para fatores ordenados, com níveis definidos conforme a hierarquia semântica de cada escala.

Variáveis binárias, como "AQUECIMENTO", "DESAQUECIMENTO", "ATIVIDADE_EXTRA" e "COMPETIÇÃO", foram tratadas a partir de respostas textuais ("Sim"/"Não") e convertidas para valores numéricos binários (1 e 0).

A variável "OBJETIVO", com ampla diversidade de respostas textuais, foi agrupada em seis categorias principais definidas previamente: saúde, desempenho, condicionamento para outro esporte, emagrecimento, hipertrofia e outros.

A coluna "TIPO_ATIVIDADE_EXTRA" exigiu tratamento especial devido à presença de uma grande quantidade de termos variados. Foi aplicado um processo de limpeza textual, remoção de palavras irrelevantes, normalização semântica e, por fim, a técnica de *one-hot encoding* que resultou no agrupamento das atividades em quatro categorias principais: **"Esportes de Resistência"**, **"Treinamento de Força/Funcional**, **"Esportes Competitivos"** e **"Outros Esportes/Atividades Recreativas"**.

Tabela 6: Transformações aplicadas aos atributos categóricos

Atributo	**Transformação**	**Observações**
MEDIA_DE_DIAS_TREINADOS	Conversão para categorias	4 Categorias: ”1-2 dias/semana”, ”3-4 dias/semana”, ”5-6 dias/semana”, ”7 dias/semana”
MÉDIA_DE_HORAS_TREINADAS	Conversão para Categorias	5 Categorias: ”1-2h”, ”3-4h”, ”5-6h”, ”7h-10h”, ”11h ou mais”
TIPO_TREINO	Conversão para Categorias	2 Categorias: ”Quadro/Aula normal”, ”Planilha/Individual”
NIVEL_COMPETICAO	Conversão para Categorias	5 Categorias: ”Não se aplica”, ”Local”, ”Regional/Estadual”, ”Nacional”, ”Internacional”
CATEGORIA_COMPETICAO	Conversão para Categorias	4 Categorias: ”Não se aplica”, ”Iniciante ou *Scaled*”, ”Intermediário”, ”RX ou Elite”
LESAO	Conversão para Categorias	2 Categorias: ”Sim”, ”Não”
SEXO	Padronização para “F” ou “M” e conversão para Categorias	2 Categorias: ”M”, ”F”
OBJETIVO	Padronização e conversão para Categorias	6 Categorias: ”saúde”, ”desempenho”, ”condicionamento para outro esporte”, ”emagrecimento”, ”hipertrofia”, ”outros”
TIPO_ATIVIDADE_EXTRA	*One hot encoding*	Númerico (0 = Não, 1 = Sim)
ATIVIDADE_EXTRA	Codificação binária	Númerico (0 = Não, 1 = Sim)
AQUECIMENTO	Codificação binária	Númerico (0 = Não, 1 = Sim)
DESAQUECIMENTO	Codificação binária	Númerico (0 = Não, 1 = Sim)
ATIVIDADE EXTRA	Codificação binária	Númerico (0 = Não, 1 = Sim)
COMPETIÇÃO	Codificação binária	Númerico (0 = Não, 1 = Sim)

4.3 Seleção de Atributos

A etapa de seleção de atributos tem como objetivo identificar as variáveis mais relevantes para a tarefa de predição de lesões, de forma a reduzir a dimensionalidade do conjunto de dados, eliminar redundâncias e melhorar a performance dos modelos de aprendizado de máquina. A escolha adequada dos preditores contribui não apenas para a eficiência computacional, mas também para a interpretabilidade dos resultados, permitindo *insights* mais claros sobre os fatores associados ao risco de lesão [13].

Inicialmente, foi realizada uma seleção manual de atributos baseada em conhecimento prévio e na literatura existente. Embora essa abordagem seja valiosa, ela pode estar sujeita a vícios e limitações inerentes ao julgamento humano. Para aprimorar e validar essa

seleção inicial, optou-se por empregar métodos automatizados de seleção de atributos, que oferecem uma análise mais objetiva e quantitativa da relevância de cada variável [78].

Neste estudo, foram selecionados quatro algoritmos para realizar a seleção de atributos. Cada um desses algoritmos foi aplicado ao conjunto de dados de treinamento, gerando rankings individuais de importância dos atributos. Posteriormente, esses rankings foram combinados para formar um ranking consolidado, considerando a frequência e a posição de cada atributo nos diferentes métodos.

4.3.1 *Recursive Feature Elimination* - RFE

O método RFE foi implementado utilizando a função `rfe()` do pacote `caret`, em conjunto com `rfFuncs`, que utiliza florestas aleatórias como modelo base. A validação cruzada com dez partições foi aplicada para garantir estabilidade nas iterações, com controle de reprodutibilidade definido por sementes fixas. Foram avaliados subconjuntos de 2 a 30 preditores. Ao final, o modelo retornou a combinação ótima de atributos baseada na performance média entre os subconjuntos.

```
#### 1. Recursive Feature Elimination (RFE)

control <- rfeControl(functions = rfFuncs, method = "cv", number = 10,
   returnResamp = "final", seeds = seeds)

results <- rfe(trainData[, -which(names(trainData) == "Lesao")],
   trainData$Lesao, sizes = seq(2, 30, 1), rfeControl = control)

rfe_vars <- results$optVariables
```

Figura 9: Aplicação do RFE com florestas aleatórias

4.3.2 *Random Forest Importance* (RFIMP)

O modelo de Floresta Aleatória foi treinado com a variável-alvo “LESAO”, e a importância das variáveis foi extraída com base na métrica *Mean Decrease in Accuracy*, que mede a perda de desempenho do modelo ao permutar aleatoriamente os valores de uma variável.

```
#### 2. Random Forest Importance

rf_model <- randomForest(Lesao ~ ., data = trainData, importance = TRUE
   )
rf_importance <- importance(rf_model)
rf_ranking <- data.frame(Variable = rownames(rf_importance),
                         Score = rf_importance[, 1])
```

Figura 10: Extração da importância das variáveis via *Random Forest*

4.3.3 LASSO - *Least Absolute Shrinkage and Selection Operator*

O algoritmo LASSO foi utilizado como método incorporado de seleção de atributos, implementado por meio da biblioteca `glmnet`, especializada em modelos lineares penalizados. O LASSO aplica uma penalização L1 sobre os coeficientes da regressão logística, o que induz a redução de alguns coeficientes a zero, realizando assim uma seleção automática de variáveis.

O modelo LASSO foi ajustado com validação cruzada para seleção automática do parâmetro de penalização (`lambda.min`). Os coeficientes resultantes foram convertidos em valores absolutos, e utilizados como medida de importância.

```
#### 3. LASSO (L1 Regularization)

x <- model.matrix(Lesao ~ ., trainData)[,-1]
y <- trainData$Lesao

lasso_model <- cv.glmnet(x, y, alpha = 1)
lasso_coefs <- coef(lasso_model, s = "lambda.min")
lasso_ranking <- data.frame(Variable = rownames(lasso_coefs)[-1],
                            Score = abs(lasso_coefs[-1, ]))
```

Figura 11: Treinamento do modelo LASSO e extração dos coeficientes

4.3.4 Boruta

O algoritmo Boruta foi aplicado por meio da função `Boruta()` do pacote homônimo em R, com configuração padrão e utilizando florestas aleatórias como base para a avaliação da importância das variáveis.

Cada atributo é classificado como “Confirmado”, “Rejeitado” ou “Tentativa”. Para fins de integração ao ranking final, foram considerados como relevantes todos os atributos “Confirmados” e “Tentativas” para evitar exclusões prematuras de variáveis potencialmente úteis.

```
#### 4. Boruta

boruta_model <- Boruta(Lesao ~ ., data = trainData, doTrace = 0)
boruta_ranking <- data.frame(Variable = getSelectedAttributes(
                             boruta_model, withTentative = TRUE),
                             Score = 1)
```

Figura 12: Execução do algoritmo Boruta

4.3.5 Combinação dos Métodos de Seleção

Após a aplicação individual dos algoritmos de seleção de atributos — RFE, RFIMP, LASSO e Boruta — foi elaborado um ranking consolidado com o objetivo de reunir as diferentes perspectivas sobre a relevância das variáveis preditoras no modelo de classificação de lesões.

Como cada método utiliza escalas próprias para quantificar a importância dos atributos, foi realizada a normalização dos scores individuais para o intervalo de 0 a 1, de modo a permitir sua agregação justa. A pontuação final foi calculada como a soma dos valores normalizados atribuídos por cada método a uma determinada variável.

```
#### 5. RANKEAMENTO

normalize <- function(x) (x - min(x)) / (max(x) - min(x))

ranking_final_completo <- bind_rows(
mutate(rf_ranking, Method = "Random Forest"),
mutate(lasso_ranking, Method = "LASSO"),
mutate(data.frame(Variable = rfe_vars, Score = 1, Method = "RFE")),
mutate(boruta_ranking, Method = "Boruta")
)

ranking_final_completo$Score <- normalize(ranking_final_completo$Score)

tabela_ranking <- ranking_final_completo %>%
spread(key = Method, value = Score, fill = 0) %>%
arrange(desc(Random Forest))

tabela_ranking$Score_Final <- rowSums(tabela_ranking[, -1], na.rm =
    TRUE)
tabela_ranking <- tabela_ranking %>% arrange(desc(Score_Final))
```

Figura 13: Procedimento para normalização e consolidação dos rankings

O critério de corte adotado para definição dos preditores finais foi a pontuação total superior a 1, garantindo que as variáveis selecionadas tenham sido reconhecidas como

importantes por, ao menos, dois algoritmos. A Figura 14 apresenta o fluxo do processo de seleção de atributos adotado neste trabalho.

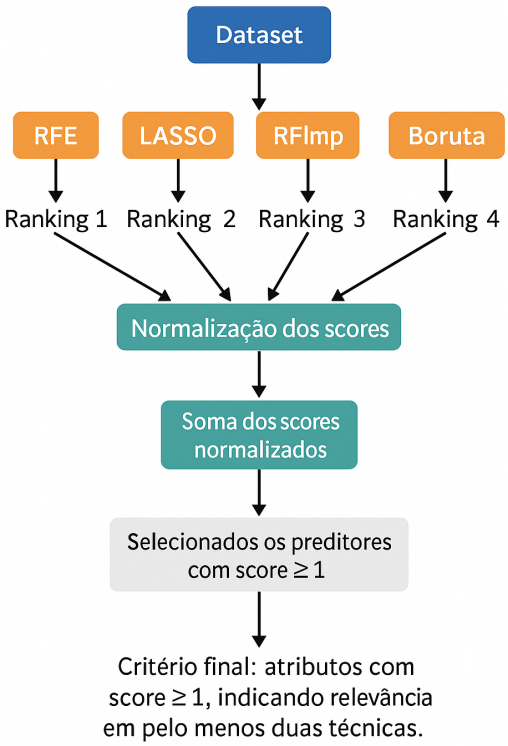


Figura 14: Fluxo de integração das técnicas de seleção de atributos.

A utilização de variáveis reconhecidas como relevantes por diferentes métodos, contribui para a robustez dos modelos a serem construídos. A partir dessa seleção, serão avaliados diversos algoritmos de aprendizado de máquina com o objetivo de identificar aqueles com melhor desempenho na tarefa de predição de lesões entre praticantes de *Cross training*.

4.4 Modelos de Aprendizado de Máquina

Com a definição do conjunto de atributos preditores, foram aplicados diferentes algoritmos de aprendizado de máquina, com o objetivo de construir modelos capazes de predizer a ocorrência de lesões entre praticantes de *Cross training*. A escolha dos algoritmos levou em consideração sua representatividade em diferentes paradigmas de classificação, incluindo métodos probabilísticos, baseados em árvore, de vetores de suporte, instância-baseados e métodos lineares.

Tabela 7: Modelos utilizados no treinamento e suas respectivas configurações

Modelo	**Método (caret)**	**Tipo**	**Biblioteca**
Naive Bayes	`naive_bayes`	Probabilístico	`naivebayes`
Random Forest	`ranger`	Ensemble (Árvores)	`ranger`
C4.5 (J48)	`J48`	Árvore de Decisão	`RWeka`
C5.0	`C5.0`	Árvore / Boosting	`C50`
SVM (Linear)	`svmLinear`	SVM	`kernlab`
SVM (Radial)	`svmRadial`	SVM (RBF)	`kernlab`
SVM (Polinomial)	`svmPoly`	SVM (Polinomial)	`kernlab`
AdaBoost	`AdaBoost.M1`	Ensemble / Boosting	`adabag`
KNN	`kknn`	Instância Baseada	`kknn`
Regressão Logística	`glmnet`	Linear / Regularizada	`glmnet`

Com o objetivo de avaliar diferentes abordagens de aprendizado de máquina na predição de lesões, foram testados diversos algoritmos representando distintas famílias de modelos. A Tabela 8 apresenta os métodos empregados no estudo, acompanhados de seus principais hiperparâmetros testados durante o processo de ajuste fino (*hyperparameter tuning*). O ajuste adequado dos hiperparâmetros é uma etapa fundamental para maximizar o desempenho dos modelos, pois influencia diretamente sua capacidade de generalização e sua eficácia na identificação de padrões complexos nos dados. Os hiperparâmetros utilizados nos modelos que apresentaram os melhores desempenhos são detalhados na seção 5.4.2, juntamente com a análise dos respectivos desempenhos.

Tabela 8: Algoritmos utilizados e principais hiperparâmetros

Algoritmo	Hiperparâmetros
Naive Bayes	`laplace = [0--3], usekernel = TRUE/FALSE, adjust = [0.1--3]`
Random Forest	`mtry = [3--8], splitrule = gini / extratrees, min.node.size = [1, 5, 10, 15], ntree = [1000, 1250, 1500], pesos = [1.1--1.3]`
C4.5 (J48)	`confidenceFactor = [0.1--0.4], minNumObj = [2, 5, 10], unpruned = TRUE/FALSE, binarySplits = TRUE/FALSE`
C5.0	`trials = [1--100], winnow = TRUE/FALSE, rules = TRUE/FALSE, CF = [0.1, 0.25], minCases = [5, 10], earlyStopping = TRUE`
SVM (Radial)	`C = [0.1--100], sigma = [0.01--1], preProcess = center/scale`
SVM (Linear)	`C = [0.1--100], preProcess = center/scale`
SVM (Polinomial)	`C = [0.1--100], degree = [2--4], scale = [0.5, 1, 2], preProcess = center/scale`
AdaBoost	`mfinal = [100, 200, 300], maxdepth = [1--5], coeflearn = Breiman / Freund / Zhu`
KNN	`kmax = [1--9], distance = 1 (Manhattan) / 2 (Euclidiana), kernel = optimal, rank, inv, rectangular, triangular, biweight, triweight, gaussian, cos, epanechnikov`
Regressão Logística (LR)	`alpha = [0--1], lambda =` 10^{-4} `--` 10^{1}`, método = glmnet`

4.4.1 Abordagem Geral

Para garantir a comparabilidade dos resultados, todos os modelos foram submetidos ao mesmo processo de validação cruzada. Com esse objetivo, foi utilizado um controle de validação com 10 subconjuntos, configurado por meio da função `trainControl()` do pacote `caret`. Para garantir reprodutibilidade dos resultados, foi criada manualmente uma lista de *seeds*, uma para cada subdivisão do processo de treino e uma final para o modelo completo, atribuída ao parâmetro `seeds`.

```
# Num. de folds
k <- 10

# Gerar lista de seeds para reprodutibilidade
set.seed(123)
seeds <- vector(mode = "list", length = k + 1)
for(i in 1:k) seeds[[i]] <- sample.int(1000, 50)
seeds[[k + 1]] <- sample.int(1000, 1)

# Controle de treino
train_control <- trainControl(
  method = "cv",                  # Valida  o cruzada
  number = k,                     # Num. de folds
  classProbs = TRUE,              # Habilita probabilidades
  summaryFunction = twoClassSummary, # Usa m tricas ROC, Sens, Spec
  search = "grid",                # Busca por grade
  seeds = seeds,                  # Lista de seeds
)
```

Figura 15: Configuração do controle de validação cruzada com 10 folds

Adicionalmente, a opção `classProbs = TRUE` foi ativada para permitir a geração de probabilidades preditivas. A função de sumarização adotada foi `twoClassSummary`, que permite o cálculo de métricas como ROC, AUC e é utilizada na avaliação de modelos binários. Por fim, o parâmetro `search = ''GRID''` define que a busca por hiperparâmetros será feita de forma exaustiva com base nos valores fornecidos na grade de *tuning*.

A Figura 16 ilustra o procedimento de validação cruzada adotado para a avaliação dos modelos preditivos. O conjunto de treino é utilizado para o desenvolvimento e ajuste dos modelos, enquanto o conjunto de teste é reservado para a avaliação final. No conjunto de treino, é realizada uma validação cruzada estratificada com dez *folds*, onde o conjunto é particionado em dez subconjuntos aproximadamente iguais. Em cada iteração, nove subconjuntos são utilizados para o treinamento e o subconjunto restante é utilizado para a validação. Este processo é repetido até que cada subconjunto tenha sido utilizado uma vez como validação. Durante essa etapa, os hiperparâmetros dos modelos são ajustados com base no desempenho observado em cada divisão. Após a definição dos melhores parâmetros, o modelo final é avaliado utilizando o conjunto de teste, assegurando uma estimativa imparcial do desempenho em dados não vistos.

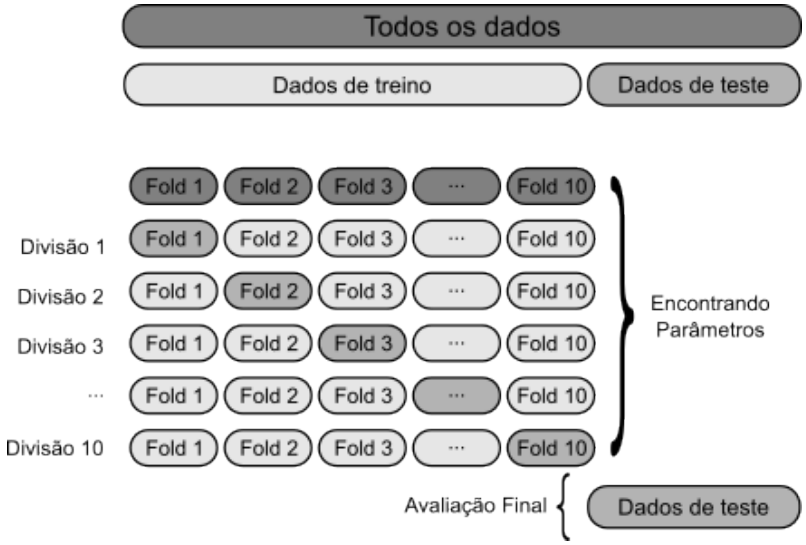


Figura 16: Esquema do processo de validação cruzada estratificada (10-folds).

O treinamento também foi realizado com auxílio do pacote `caret` do R, que oferece uma interface padronizada para diversos algoritmos distintos disponíveis pela função `train()`. A Tabela 7 apresenta os modelos de aprendizado de máquina empregados neste estudo, juntamente com o nome do método utilizado na função. Essa abordagem permitiu maior consistência na aplicação dos diferentes modelos, ao mesmo tempo em que simplificou a integração de técnicas de pré-processamento, seleção de atributos e avaliação de desempenho.

4.4.2 Escolha do modelo

O objetivo desta etapa é identificar os modelos com melhor capacidade preditiva e que apresentem resultados estatisticamente significativos, de modo a fundamentar a escolha dos algoritmos mais promissores para etapas posteriores de avaliação e validação.

Para avaliação do desempenho dos modelos, utilizou-se a matriz de confusão gerada com base nas previsões realizadas sobre o conjunto de teste. As métricas de avaliação foram extraídas utilizando a função `confusionMatrix()` do pacote `caret`, considerando como classe positiva a categoria “Sim”. Foram calculadas acurácia, *Precision*, *Recall*, *F1-Score* e o p-valor.

A extração das métricas foi realizada diretamente a partir dos objetos retornados pela função, conforme o exemplo abaixo:

```
conf_mat <- confusionMatrix(predicoes, testData$Lesao,
                            positive = "Sim")

conf_mat$overall["Accuracy"]          # Acur cia
conf_mat$overall["AccuracyPValue"]    # p-valor
conf_mat$byClass["Sensitivity"]       # Recall
conf_mat$byClass["Pos Pred Value"]    # Precision
conf_mat$byClass["F1"]                # F1-Score
```

Figura 17: Exemplo de extração das métricas de avaliação dos modelos

A Figura 18 resume e ilustra o fluxo de avaliação dos modelos preditivos considerados neste estudo. Embora as métricas tradicionais de classificação forneçam uma visão geral do desempenho dos algoritmos, optou-se também por realizar uma análise por meio da curva ROC e da AUC. A partir da comparação gráfica das curvas e dos valores obtidos para os modelos com desempenho estatisticamente significativo, foi possível identificar aquele mais adequado para representar este estudo.

Figura 18: Fluxo de avaliação e comparação dos modelos preditivos.

4.5 Aplicação de Modelo e Validação

Com o objetivo de avaliar a aplicabilidade prática do modelo escolhido, foi realizada uma validação em ambiente interativo por meio da criação de um Mínimo Produto Viável. Essa abordagem buscou demonstrar a viabilidade do uso do modelo de predição de lesões em um contexto real, acessível e dinâmico.

Para tornar o modelo preditivo acessível de forma prática e validar sua aplicabilidade em um cenário real, o aplicativo foi desenvolvido utilizando *Shiny*, uma biblioteca da linguagem R que permite a construção de aplicações web interativas com base em *scripts*

estatísticos, e foi hospedado na plataforma *Shinyapps.io*, um serviço de *deploy* em nuvem mantido pela *Posit* (antiga *RStudio*), voltado especificamente para aplicações web criadas com R. A escolha dessa plataforma se deu pela sua integração nativa com o ambiente *RStudio* e pela facilidade de disponibilizar aplicações sem a necessidade de infraestrutura própria.

Essa plataforma oferece um plano gratuito com funcionalidades limitadas, incluindo quantidade restrita de horas de uso mensal, número reduzido de aplicações ativas e não há permissão para armazenamento de dados no servidor. Dessa forma, como alternativa à limitação de armazenamento local, foi utilizada uma integração com o *Google Drive* para registrar as respostas dos participantes. A integração foi realizada por meio dos pacotes `googledrive` e `googlesheets4`, ambos desenvolvidos especificamente para facilitar o acesso e a manipulação de arquivos e planilhas na nuvem, a partir da linguagem R. Essa estratégia permitiu que os dados inseridos pelos usuários, fossem automaticamente registrados em uma planilha hospedada no *Google Sheets*, contornando assim as restrições de escrita impostas pelo plano gratuito do `shinyapps.io`.

As configurações de *deploy* e os detalhes técnicos de publicação estão descritos na Seção 5.5.1 dos resultados, como parte da validação prática do modelo em ambiente interativo.

5 RESULTADOS OBTIDOS

Neste capítulo, são apresentados os resultados obtidos a partir das análises realizadas durante o desenvolvimento da pesquisa. Inicialmente, descreve-se a preparação final do conjunto de dados, com base nas etapas de pré-processamento, transformação e seleção de atributos. Em seguida, são relatados os desempenhos dos modelos de aprendizado de máquina aplicados, considerando-se métricas utilizadas em tarefas de classificação binária, tais como acurácia, precisão, sensibilidade, *F1-score* e *p-value*.

Os modelos com desempenho mais expressivo são discutidos de forma aprofundada, com ênfase na interpretação das variáveis mais relevantes e no comportamento observado nas matrizes de confusão. O capítulo é finalizado com o processo de validação em ambiente interativo, por meio da implementação de um aplicativo web funcional que permitiu a simulação do uso da solução proposta em um contexto real.

5.1 Preparação da Base Final

A primeira etapa da apresentação dos resultados consiste na caracterização da base de dados resultante após o processo de preparação descrito na metodologia. Esta seção tem como objetivo evidenciar o impacto das etapas de pré-processamento, limpeza e transformação na estrutura final do conjunto de dados, que serviu como base para o treinamento e avaliação dos modelos preditivos.

5.1.1 Remoção de Duplicatas

Em respeito a registros duplicados, foram identificadas 39 observações duplicadas em um total de 673 registros originais, representando aproximadamente 5,79% do conjunto de dados. A presença desses registros poderia atrapalhar a análise estatística e comprometer a performance dos modelos de aprendizado de máquina, especialmente em tarefas de classificação supervisionada. Por esse motivo, tais registros foram removidos, mantendo-se apenas a primeira ocorrência de cada instância. A Tabela 9 apresenta um resumo quantitativo do processo de remoção das duplicatas:

Tabela 9: Resumo da detecção e remoção de duplicatas no conjunto de dados

Descrição	Quantidade	Porcentagem (%)
Total de registros originais	673	100,00
Registros duplicados	39	5,79
Registros únicos após remoção	634	94,21

5.1.2 Divisão dos Dados

O conjunto de dados final foi dividido em dois subconjuntos: treino e teste. O conjunto de treino, utilizado para o ajuste dos modelos, contém 445 observações. Já o conjunto de teste, destinado à avaliação do desempenho dos modelos em dados não vistos, é composto por 189 observações.

Tabela 10: Distribuição da variável-alvo "LESAO" nos conjuntos de treino e teste

Classe	Qtd. Treino	Qtd. Teste
Sim	248 (55,7%)	105 (55,6%)
Não	197 (44,3%)	84 (44,4%)
TOTAL	445 (100%)	189 (100%)

A variável-alvo "LESAO" apresentou uma distribuição relativamente balanceada em ambos os conjuntos. No conjunto de treino, 55,7% dos registros foram classificados como "Sim", enquanto no conjunto de teste esse percentual foi de 55,6%, conforme apresentado na Tabela 10.

5.1.3 Tratamento de Valores Ausentes

Sobre o tratamento de valores ausentes, durante a análise inicial do conjunto de dados, foi observado que algumas variáveis apresentavam valores ausentes. Os únicos atributos que apresentaram valores faltantes foram Idade e Altura, com aproximadamente 8 (1.26%) e 4 (0.63%) registros.

Tabela 11: Distribuição de valores ausentes por variável nos conjuntos de treino e teste

Variável	Qtd Treino	Qtd Teste	Total
Idade	7 (1,57%)	1 (0,53%)	8 (1,26%)
Altura	3 (0,67%)	1 (0,53%)	4 (0,63%)

A média de Idade para os homens foi de 30 anos, enquanto para as mulheres foi de 31 anos. A média de Altura para os homens foi de 176 cm, e para as mulheres foi de 165

cm. Após a imputação, os valores ausentes foram completamente resolvidos, com todos os atributos numéricos completos.

5.1.4 Eliminação de atributos

Foram removidos atributos de caráter pessoal, como “NOME”, “CIDADE_ATUAL”, “E-MAIL” e outras informações identificáveis, visando garantir a privacidade dos participantes. Além disso, como o foco do estudo é predizer a ocorrência de lesões, informações relacionadas ao tratamento das lesões e ao acompanhamento posterior à ocorrência foram excluídas.

Tabela 12: Atributos removidos do dataset.

Atributo	**Descrição**
DATA_HORA	Data e hora de coleta da resposta
EMAIL	Endereço de e-mail para contato
CONSENTIMENTO	Confirmação de consentimento para pesquisa
NOME	Nome completo do participante
CIDADE_ATUAL	Cidade onde reside o participante
AVISO	Aviso de orientação para as próximas perguntas
LOCAL_LESAO	Parte do corpo em que ocorreu a lesão
LESAO_ANTERIOR	Lesão anterior ao treinamento de Crossfit®
LESAO_REPERCUSSAO	Tipo de repercussão nos treinos, causada pela lesão
LESAO_TEMPO _SEM_TREINO	Dias sem treinos, em decorrência da lesão
LESAO_MOVIMENTO_ASSOCIADO	Movimento associado à ocorrência da lesão
LESAO_TRATAMENTO	Tipo de tratamento realizado para tratar a lesão
LOCAL_TREINO	Cidade atual e nome do local onde treina Crossfit®

O atributo “LESÃO_ANTERIOR” poderia ser um preditor interessante, pois indica se o participante já sofreu alguma lesão, o que pode estar diretamente relacionado a uma maior predisposição a novas lesões. No entanto, o atributo “LESÃO_ANTERIOR” foi excluído da análise preditiva porque ele foi questionado exclusivamente para os participantes que responderam “Sim” ao atributo alvo “LESÃO” no questionário. Embora existam alguns poucos registros onde há indicação de lesão anterior e ausência de lesão atual, esses casos devem ser desconsiderados, pois, de acordo com o questionário, a pergunta deveria ter sido respondida apenas por aqueles que apresentam lesão atual. Assim, a presença desses registros incorretos não é representativa para a análise.

Além disso, a forte dependência entre “LESÃO_ANTERIOR” e “LESÃO” compromete seu valor como preditor independente, uma vez que, na maioria dos casos, os participantes que indicaram ter tido uma lesão anterior também possuem lesão atual. Essa

relação direta implica que “LESÃO_ANTERIOR” não adiciona informações novas ou independentes ao modelo preditivo, já que sua resposta está condicionada pela resposta ao atributo alvo. Portanto, o atributo foi removido para evitar redundância e inconsistências na análise.

Após a etapa de remoção manual dos atributos, o número de variáveis do conjunto de dados foi reduzido de 32 para 19.

5.1.5 Transformação dos Dados

No processo de pré-processamento dos dados, transformações foram aplicadas aos atributos do *dataset* com o objetivo de torná-los adequados para a análise e modelagem preditiva. Essas transformações envolveram a conversão de valores categóricos em formatos numéricos, o ajuste de unidades de medida para uniformidade, e a extração de informações relevantes de atributos textuais. A Tabela 13 exemplifica algumas das transformações realizadas nos dados originais, destacando as mudanças mais significativas e os valores transformados.

Tabela 13: Exemplos de transformações aplicadas aos dados

Atributo	**Valor Original**	**Valor Transformado**
EXPERIENCIA	“2 anos e meio”	30 (meses)
IDADE	“15/05/1988”	37 (anos)
ALTURA	“1,75 m”	175(cm)
PESO	“92/93 kg”	92,5 (kg)
SEXO	“Feminino”	F
OBJETIVO	“foco em saúde”	saúde
QTD_PROF_AULA	“3 ou quatro”	3
QTD_ALUNO_AULA	“dois”	2
ATIVIDADE_EXTRA	“natação e corrida”	*Esportes de Resistência*
AQUECIMENTO	“Sim”	1
DESAQUECIMENTO	“Não”	0

Entre as transformações mais notáveis, destaca-se a conversão do atributo “ATIVIDADE_EXTRA”. Originalmente, este atributo apresentava informações textuais que descreviam atividades extras, como ”corrida e caminhada”. Esta transformação categórica resultou na conversão de cada atividade em uma variável binária. Após o mapeamento das respostas contidas neste atributo, de forma a generalizar e contemplar inclusive aqueles que possuíam mais de uma atividade registrada, as respostas foram agrupadas sob 4 grandes categorias, como ilustrado na Tabela 14.

Tabela 14: Distribuição das categorias após a aplicação do One-Hot Encoding

Categoria	Distribuição	Descrição
"Treinamento de Força/Funcional"	100	Abrange exercícios focados no fortalecimento muscular e resistência, como musculação, levantamento de peso e treinamento funcional.
"Esportes de Resistência"	244	Representa atividades aeróbicas e de *endurance*, como corrida, ciclismo natação entre outros.
"Esportes Competitivos"	79	Refere-se a esportes em equipe, como futebol, basquete, vôlei e atividades físicas relacionadas a artes marciais e lutas, como judô, karatê, entre outras.
"Outros Esportes e Atividades"	51	Agrupa atividades físicas que não se enquadram nas demais categorias, incluindo modalidades alternativas ou recreativas.

5.2 Análise Exploratória de Dados

A análise exploratória de dados (AED) constitui uma etapa fundamental em projetos de ciência de dados e aprendizado de máquina, pois permite obter uma compreensão preliminar do conjunto de dados, suas distribuições, padrões e possíveis inconsistências.

A seguir, serão apresentados os principais achados da AED aplicada ao conjunto de dados tratado na etapa de pré-processamento. Inicialmente, são exploradas as variáveis demográficas e comportamentais dos participantes, seguidas pelas variáveis relacionadas à prática do Crosstraining.

5.2.1 Perfil Demográfico dos Participantes

A Figura 19 mostra a distribuição das observações por faixa etária e sexo. Tal distribuição é consideravelmente equilibrada entre os sexos nas faixas intermediárias, mas com uma leve predominância feminina. Através dela podemos perceber que a faixa etária mais representada entre homens e mulheres é a de 23-28 anos, com 78 e 100 amostras respectivamente.

Podemos perceber, por meio da Figura 20b, que os homens participantes da pesquisa possuem uma estatura maior que as mulheres, o que é uma diferença esperada em termos de características físicas gerais. A altura mediana dos homens é 176cm com valores que variam de 160cm a 193cm. Para as mulheres, a mediana da altura é de 164cm, com as alturas variando entre 149cm e 182cm. Já a Figura 20a, revela uma diferença entre os sexos, com os homens tendendo a ser mais pesados que as mulheres. A mediana do peso masculino é de 82kg com a maioria dos valores variando entre 51kg e 118kg. Para as mulheres, a mediana é de 65kg, com os valores variando entre 48kg e 117kg.

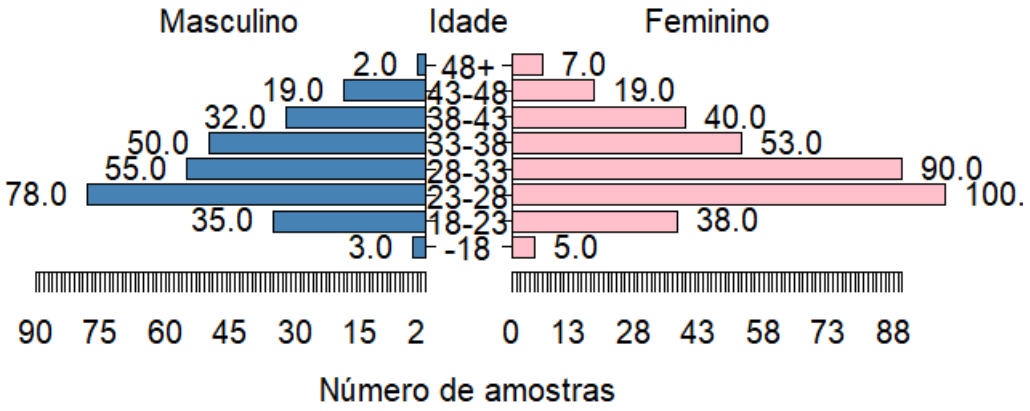
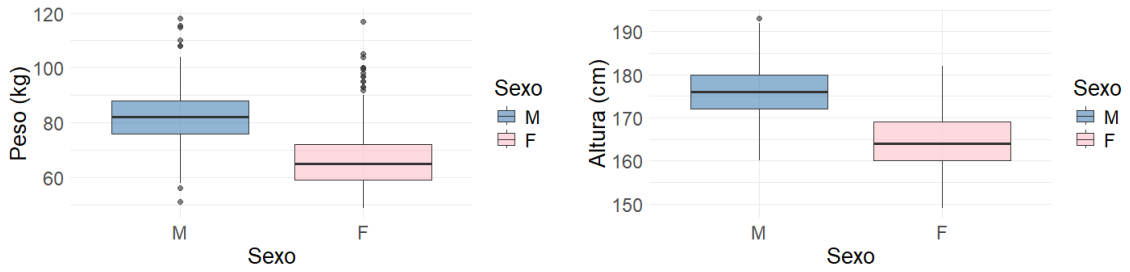


Figura 19: Dados demográficos dos participantes



(a) Peso X Sexo

(b) Altura X Sexo

Figura 20: Boxplot das distribuições dos atributos Peso (kg) e Altura (cm), ambos relacionados com o atributo Sexo dos participantes

5.2.2 Objetivos Declarados

Os objetivos (Figura 21) mais frequentes, tanto para homens quanto para mulheres, são desempenho e saúde, com a maior quantidade de homens (102) se concentrando no objetivo de desempenho e a maioria das mulheres (160) focando em saúde. Outros objetivos como emagrecimento e hipertrofia apresentam frequências intermediárias, com uma prevalência maior entre as mulheres para emagrecimento (74), e entre os homens para hipertrofia (26). Os objetivos menos mencionados são condicionamento e outros, ambos com baixas frequências em ambos os sexos.

5.2.3 Comportamento de Treinamento

Os dados da Figura 22 demonstram que a maioria dos participantes mantém uma rotina intensa de treinos, com uma frequência de 5 a 6 dias por semana e uma média de 5 a 6 horas semanais de treinamento. No primeiro gráfico 22a, a segunda faixa mais comum é de 3 a 4 dias por semana, com uma prevalência maior entre as mulheres (134), enquanto os homens (84) se distribuem de forma menos acentuada nesta faixa. Poucos indivíduos,

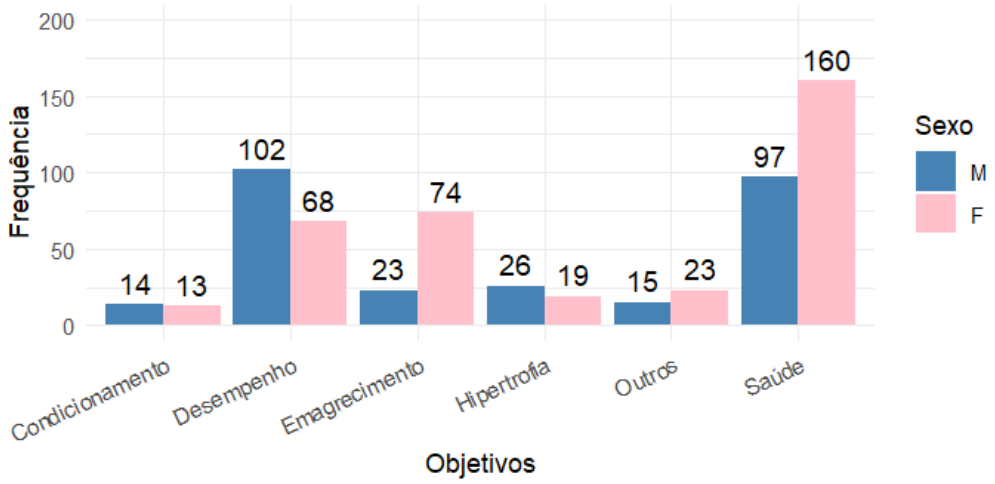


Figura 21: Objetivos dos participantes com a prática do Crossfit®.

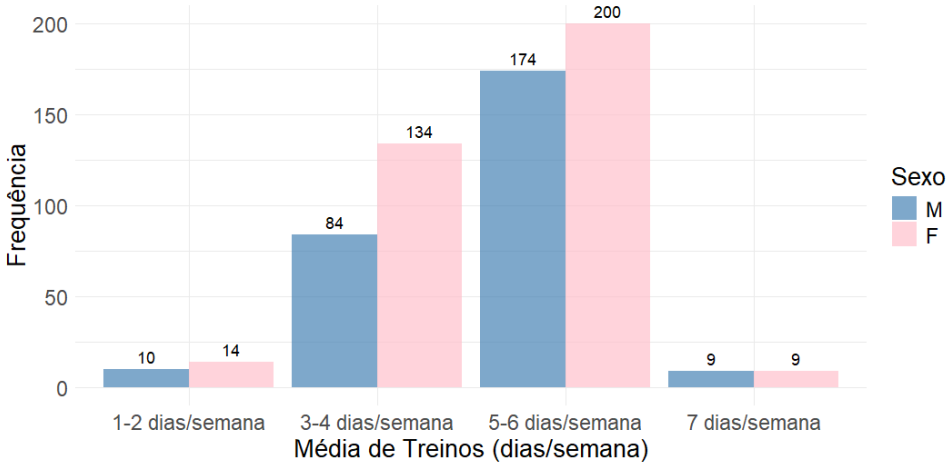
de ambos os sexos, treinam 7 dias por semana ou menos de 3 dias.

No segundo gráfico 22b, sobre a média de horas treinadas, observamos que a maioria dos participantes (tanto homens quanto mulheres) se encontra na faixa de 5 a 6 horas de treino por semana, sendo mais frequente entre as mulheres (116) do que entre os homens (92). Também é notável que uma parcela significativa das mulheres treina de 1 a 2 horas por semana (105), enquanto para os homens essa frequência é ligeiramente menor (86). Poucos participantes relataram treinar mais de 7 horas por semana, com uma leve predominância feminina nas faixas mais altas.

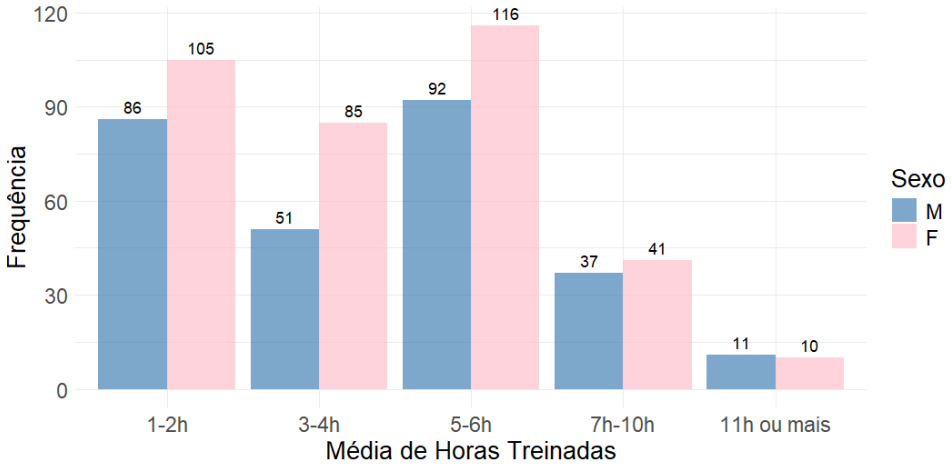
A Figura 23 mostra que, enquanto o aquecimento é amplamente adotado pela maioria dos participantes, o desaquecimento é frequentemente negligenciado. Na figura 23a observa-se que a grande maioria dos participantes, tanto homens (274) quanto mulheres (349), relataram que realizam aquecimento antes dos treinos. Apenas uma pequena porcentagem indicou que não realiza. já na figura 23b, os resultados indicam uma menor adesão a a prática do desaquecimento. A maioria dos homens (191) e mulheres (257) indicaram que não realizam a "volta à calma" após os treinos, enquanto apenas uma parte menor afirmou realizar (86 homens e 100 mulheres).

5.2.4 Prática de Atividades Físicas Extras

A combinação das informações apresentadas na Figura 24 sugere que a maioria dos participantes seguem o treinamento em grupo ofertados em seus centros de treinamento e muitos realizam atividades extras fora do contexto do Crossfit®. A Figura 24a mostra que a maioria dos participantes segue o formato de aulas normais ou treinos em grupo (249 homens e 341 mulheres), enquanto apenas uma minoria segue treinos individuais ou planilhas (28 homens e 16 mulheres). Já a Figura 24b, referente à prática de atividades extras, vemos que uma quantidade significativa de participantes (184 homens e 212 mulheres) indicou que realiza atividades físicas complementares. Um número menor de

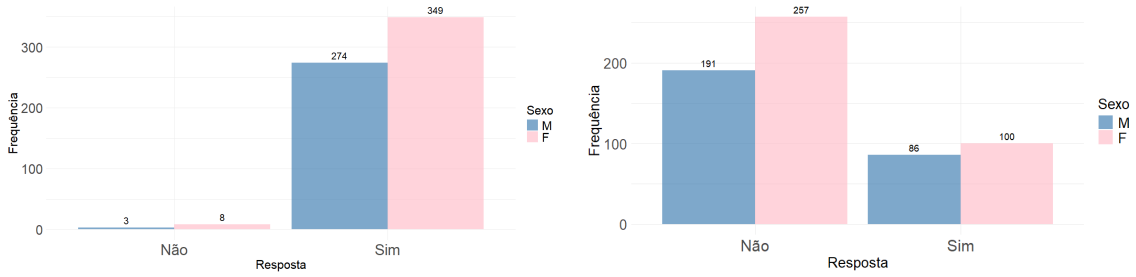


(a) Média de treinos (dias/semana)



(b) Média de horas treinadas

Figura 22: Distribuição das médias dos dias e horas treinadas no Crossfit®, durante 1 semana, considerando as últimas 4 semanas.

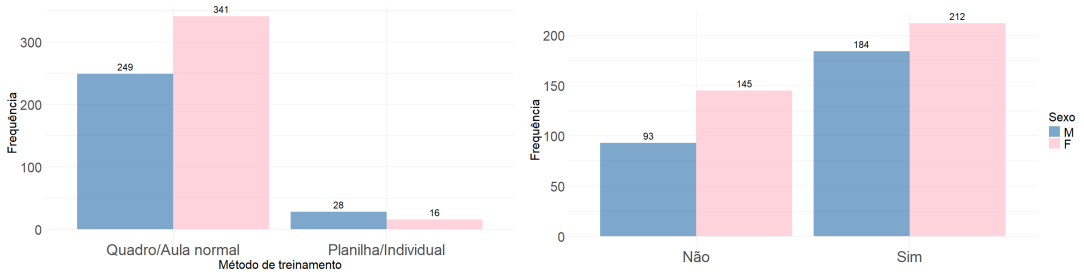


(a) Aquecimento (b) Desaquecimento

Figura 23: Realização de aquecimento e desaquecimento durante as sessões de treino de Crossfit®

participantes (93 homens e 145 mulheres) afirmou que não realiza essas atividades.

Na Figura 25 observa-se que "Esportes de Resistência" (como corrida, ciclismo e natação) foram os mais frequentes entre os respondentes, com destaque para a maior participação do sexo feminino (134) em relação ao masculino (110). A segunda categoria



tipo

(a) Método de treinamento (b) Realização de exercício físico extra

Figura 24: Método de treinamento realizado pelos participantes e realização de exercício físico paralelo ao treinamento de Crossfit®

com maior frequência foi "Treinamento de Força & Funcional", com uma leve predominância do público feminino (55) em comparação ao masculino (45). Por outro lado, "Esportes Competitivos" foram mais relatados por homens (48) do que por mulheres (31), sugerindo uma maior adesão masculina a modalidades como futebol ou lutas. A categoria "Outros Esportes & Atividades Recreativas" apresentou as menores frequências, mas com predominância do sexo feminino (28 contra 23). Esses dados sugerem diferenças nos interesses e práticas, o que pode refletir preferências pessoais, disponibilidade de atividades ou objetivos distintos em relação ao treinamento principal.

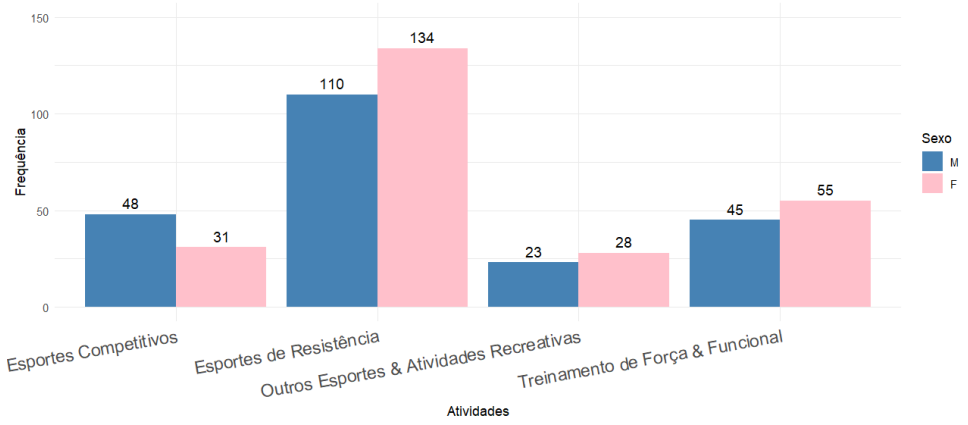
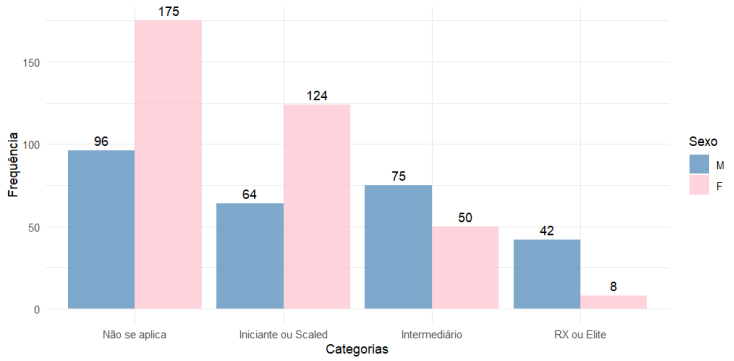


Figura 25: Distribuição das atividades físicas extras praticadas, segmentadas por sexo.

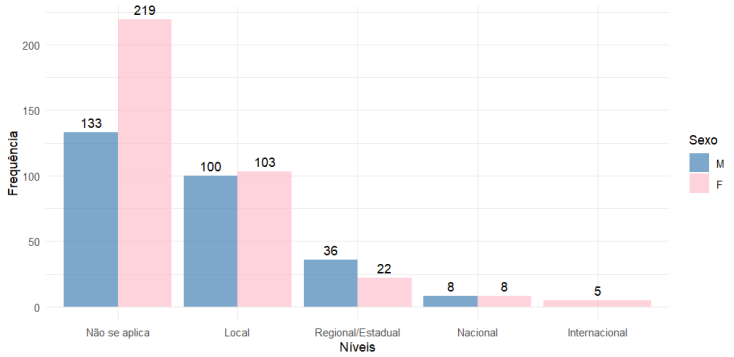
5.2.5 Participação em Competições

Conforme mostra a Figura 26c, enquanto a maioria dos participantes não compete, entre os que o fazem, a maioria compete em níveis mais acessíveis, como iniciantes e intermediários, e em competições locais. Na figura 26a, a maioria (96 homens e 175 mulheres) indicou que essa informação "não se aplica", sugerindo que não competem, conforme comentado anteriormente. Entre os competidores, as categorias iniciantes ou *scaled* e intermediário são as mais populares, com uma predominância feminina em iniciantes ou *scaled* e uma leve predominância masculina na categoria intermediário. Apenas uma pequena fração dos participantes compete em níveis mais altos como RX ou Elite,

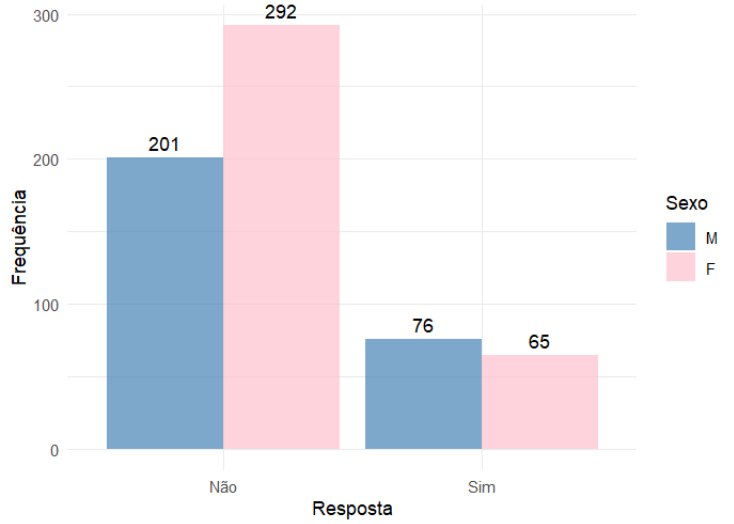
com maior frequência entre os homens (42). A Figura 26b mostra que a maioria dos competidores participam de competições locais (100 homens e 103 mulheres), enquanto uma menor parte compete em níveis regional/estadual (36 homens e 22 mulheres), nacional (8 homens e 8 mulheres), e internacional (5 mulheres).



(a) Categorias dos participantes



(b) Nível de abrangência das competições



(c) Participação em campeonatos de Crossfit®

Figura 26: Dados referentes a participação em campeonatos, bem como as categorias escolhidas pelos participantes e a abrangência das competições

5.2.6 Ocorrência de Lesões

Por fim, os dados apresentados na Figura 27, mostram os números referentes a ocorrência de lesões durante o treinamento de Crossfit®. A diferença entre os gêneros é relativamente pequena, com as mulheres apresentando uma leve predominância tanto entre as que se lesionaram quanto entre as que não se lesionaram. Observa-se que a maioria dos participantes (169 homens e 184 mulheres) relataram já ter sofrido algum tipo de lesão durante a prática do esporte. Por outro lado, um número significativo de pessoas também indicou não ter se lesionado (108 homens e 173 mulheres), embora essa quantidade seja menor do que a dos que já sofreram lesões.

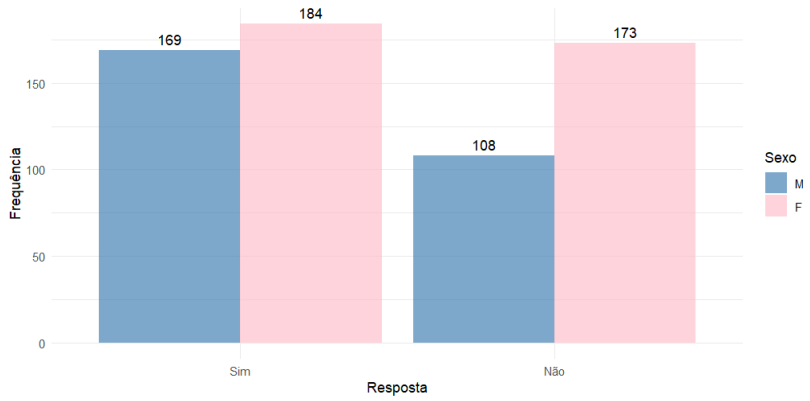


Figura 27: Ocorrência de lesão durante treinamento de Crossfit®

5.3 Seleção de Atributos

Para a identificação dos atributos mais relevantes à predição de lesões, foram aplicadas quatro abordagens distintas de seleção de atributos, cada uma baseada em um princípio metodológico diferente. Estudos recentes têm demonstrado a eficácia de abordagens híbridas que combinam múltiplos métodos de seleção de atributos, como Boruta, RFE e LASSO, visando melhorar o desempenho de modelos preditivos em contextos de saúde [81, 4].

5.3.1 *Recursive Feature Elimination* - RFE

Dentre as variáveis selecionadas, destaca-se “EXPERIENCIA”, que foi também uma das mais relevantes em outros algoritmos. A inclusão de “OBJETIVO” sugere que a motivação do praticante pode ter influência na ocorrência de lesões, o que será analisado com maior profundidade nos modelos subsequentes.

A Figura 28 apresenta a evolução da acurácia do modelo em função do número de atributos selecionados por meio do método RFE. Verifica-se que a acurácia máxima foi obtida com a seleção de apenas duas variáveis (“EXPERIENCIA” e “OBJETIVO”), alcançando valor superior a 0,60. A partir desse ponto, observa-se tendência geral de redução da

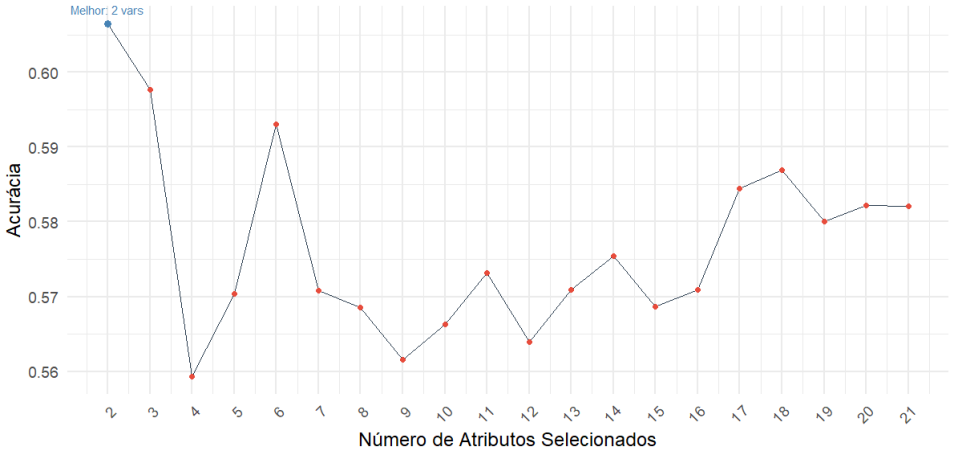


Figura 28: Relação entre o número de atributos selecionados e a acurácia utilizando RFE

acurácia com o aumento do número de atributos, ainda que com algumas oscilações.

5.3.2 *Random Forest Importance* (RFIMP)

A Figura 29 apresenta as variáveis que apresentaram relevância positiva, segundo a métrica *Mean Decrease Accuracy* do modelo *Random Forest*. Observa-se que "IDADE" e "EXPERIENCIA" se destacam como os principais preditores, indicando que fatores relacionados ao tempo de prática e maturidade do praticante exercem papel central na predição de lesões. Outros atributos com alta importância incluem "QTD_DE_PROFS_EM_AULA", "PESO" e "OBJETIVO", o que reforça a influência de aspectos comportamentais e de contexto de treino. Por outro lado, algumas variáveis tradicionalmente menos discutidas, como "AQUECIMENTO" e "TIPO_DE_TREINO", também aparecem com impacto relevante, sugerindo que a organização da rotina de treino pode ter relação com o risco de lesões.

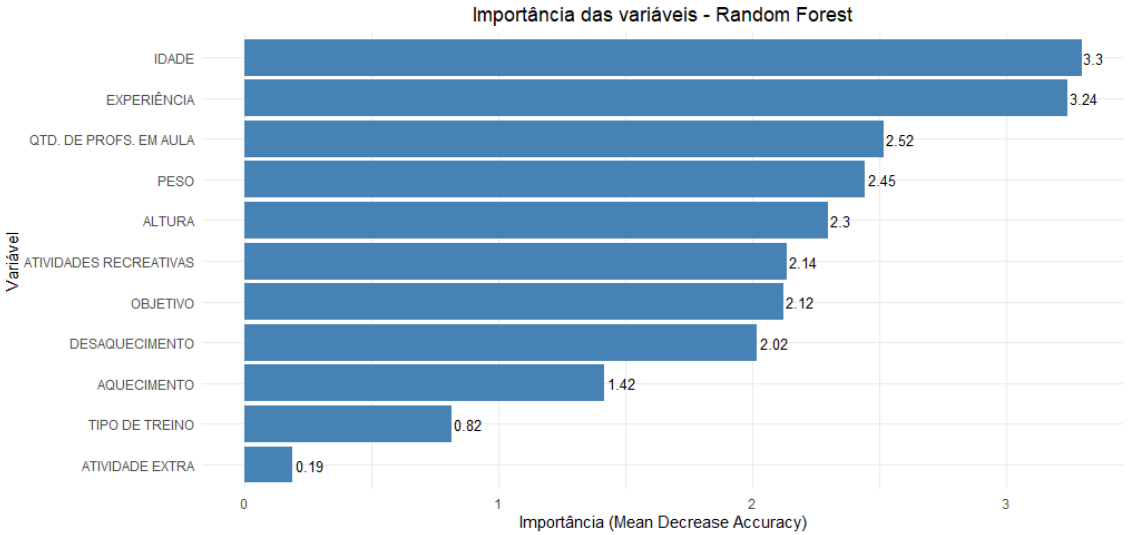


Figura 29: Importância das variáveis segundo o modelo *Random Forest*

5.3.3 LASSO - *Least Absolute Shrinkage and Selection Operator*

A análise mostra que variáveis relacionadas à estrutura da prática e à carga de treino foram aquelas com maior influência no modelo. Destaque para "DESAQUECIMENTO" e "TIPO_DE_TREINOPLANILHA/INDIVIDUAL", que apresentaram os maiores coeficientes absolutos. Além disso, aspectos relacionados ao volume semanal de treino, como "MEDIA_HORAS_TREINADAS.C" e "MEDIA_HORAS_TREINADASˆ4", também foram selecionados com peso relevante. A presença de "CATEGORIA_DE_COMPETICAO.Q" sugere que o nível de competitividade pode ser um fator adicional a ser considerado na predição de lesões.

A Figura 30 apresenta a evolução do erro quadrático médio (*Mean Squared Error*) em função do parâmetro de regularização (λ) no processo de seleção de atributos utilizando o método *LASSO*. Verifica-se que a redução progressiva de λ inicialmente proporciona diminuição do erro, até alcançar um valor mínimo próximo a $\log(\lambda) \approx -4$. Nesse ponto, o modelo seleciona 13 variáveis (Tabela 15), correspondendo ao conjunto de atributos que melhor equilibra a redução do erro e a simplicidade do modelo. A partir desse ponto, a redução adicional de λ leva ao aumento do erro, indicando tendência de sobreajuste.

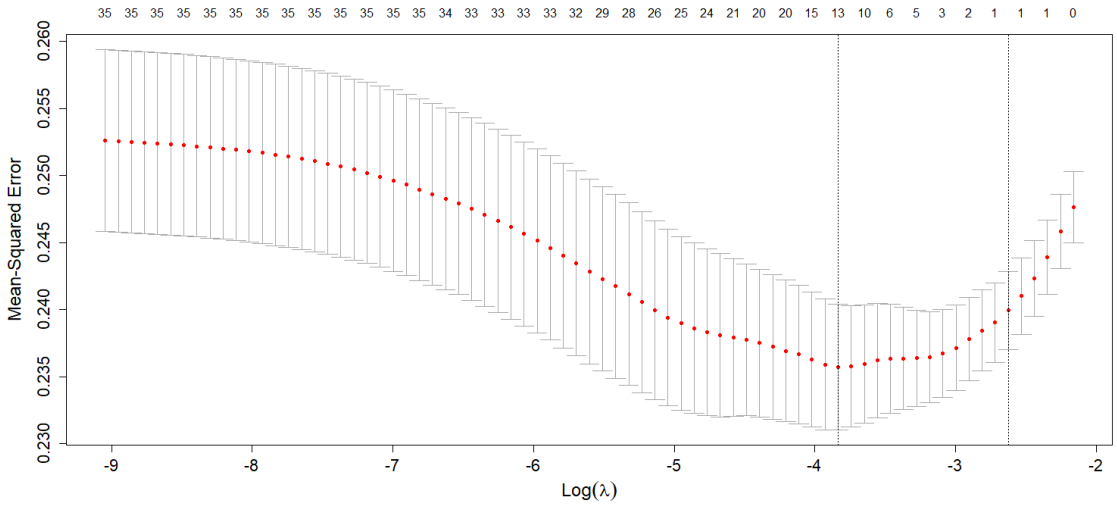


Figura 30: Seleção do parâmetro λ no LASSO para identificação de atributos relevantes

Tabela 15: Variáveis selecionadas pelo *LASSO* e seus coeficientes absolutos.

Variável	**Coeficiente Absoluto**
Desaquecimento	0,1052
Tipo_de_treino/Individual	0,1014
Média_horas_treinadas.C	0,0621
Catg_de_comp.Q	0,0421
Média_horas_treinadas^4	0,0307
Catg_de_comp.L	0,0274
Qtd_de_profs_em_aula	0,0077
SexoF	0,0073
Experiência	0,0046
atv_Trein_de_Forca_Func	0,0025
Qtd_de_alunos_em_aula	0,0008
Idade	0,0005
Altura	0,0002

5.3.4 Boruta

O Boruta identificou cinco variáveis relevantes. As variáveis "EXPERIENCIA" e "TIPO_DE_TREINO" foram classificadas como confirmadas, reforçando sua importância na predição de lesões. As variáveis "DESAQUECIMENTO", "OBJETIVO" e "QTD_DE_PROFS_EM_AULA" foram classificadas como tentativas, mas mantidas como relevantes no contexto deste estudo por apresentarem potencial preditivo identificado em outros métodos. A presença recorrente dessas variáveis reforça sua relevância no cenário de risco de lesões em praticantes de *Cross training*.

A Figura 31 apresenta os resultados do processo de seleção de atributos utilizando o algoritmo *Boruta*. Neste gráfico, são exibidas as distribuições da importância das variáveis ao longo das iterações. As variáveis confirmadas como importantes são destacadas em verde, enquanto as rejeitadas aparecem em vermelho, e as variáveis indeterminadas são indicadas em amarelo. As variáveis artificiais (*shadows*), utilizadas como referência para a comparação da importância real dos atributos, são representadas em azul. Observa-se que variáveis como "EXPERIENCIA", "TIPO_DE_TREINO" apresentaram alta importância e foram consistentemente selecionadas. Por outro lado, atributos com importância inferior às variáveis sombra foram rejeitados. A Tabela 16 apresenta um resumo das variáveis selecionadas.

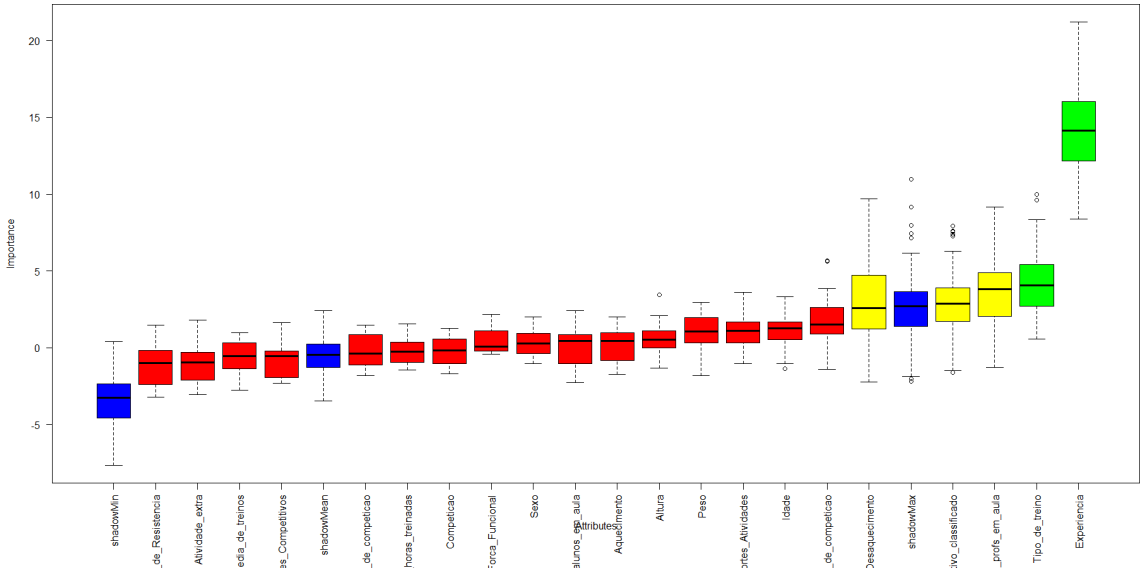


Figura 31: Importância das variáveis no processo de seleção de atributos utilizando o algoritmo *Boruta*.

Tabela 16: Variáveis selecionadas como relevantes pelo algoritmo Boruta

Ordem	**Variável**	**Classificação**
1	“EXPERIENCIA”	Confirmada
2	“DESAQUECIMENTO”	Tentativa
3	“QTD_DE_PROFS_EM_AULA”	Tentativa
4	“TIPO_DE_TREINO”	Confirmada
5	“OBJETIVO”	Tentativa

5.3.5 Combinação dos Métodos de Seleção

Como cada método utiliza escalas próprias para quantificar a importância dos atributos, foi realizada a normalização dos scores individuais para o intervalo de 0 a 1, de modo a permitir sua agregação justa. A pontuação final foi calculada como a soma dos valores normalizados atribuídos por cada método a uma determinada variável.

Tabela 17: Ranking consolidado das variáveis preditoras

Variável	Boruta	LASSO	RFIMP	RFE	Score Final
“EXPERIENCIA”	0.656	0.507	0.992	0.656	**2.810**
“OBJETIVO”	0.656	0.000	0.824	0.656	**2.136**
“QTD_DE_PROFS_EM_AULA”	0.656	0.507	0.883	0.000	**2.046**
“DESAQUECIMENTO”	0.656	0.522	0.809	0.000	**1.986**
“IDADE”	0.000	0.506	1.000	0.000	**1.506**
“PESO”	0.000	0.506	0.872	0.000	**1.378**
“ALTURA”	0.000	0.506	0.851	0.000	**1.357**
“ATV_OUTROS_ESPORTES_ATIVIDADES”	0.000	0.506	0.826	0.000	**1.332**
“TIPO_DE_TREINO”	0.656	0.000	0.628	0.000	**1.284**
“AQUECIMENTO”	0.000	0.506	0.719	0.000	**1.225**
“ATIVIDADE_EXTRA”	0.000	0.506	0.534	0.000	**1.040**

A Tabela 17 apresenta o ranking consolidado das variáveis preditoras após a aplicação e agregação dos quatro algoritmos de seleção. Variáveis como “EXPERIENCIA”, “QTD_DE_PROFS_EM_AULA” e “DESAQUECIMENTO” destacaram-se com as maiores pontuações totais, sendo consistentemente reconhecidas como relevantes por múltiplos métodos. A variável “OBJETIVO”, apesar de apresentar coeficiente nulo no LASSO, obteve alto score pela Boruta e RFIMP, o que reforça sua importância de forma complementar.

5.4 Desempenho dos Modelos de Aprendizado

Esta seção apresenta os resultados obtidos a partir da aplicação dos modelos de aprendizado de máquina treinados com o conjunto de dados preparado. A análise é conduzida de forma a identificar os modelos mais adequados à tarefa de predição de lesões em praticantes de Crosstraining, considerando tanto a performance estatística quanto a significância dos resultados obtidos.

5.4.1 Comparativo Geral das Métricas

A Tabela 18 apresenta o comparativo entre os modelos avaliados. A métrica de acurácia foi utilizada como critério principal de ordenação, enquanto o p-valor serviu como indicador de significância estatística da diferença entre o modelo e um classificador aleatório.

Tabela 18: Comparativo de desempenho dos modelos segundo métricas de classificação

Modelo	**Acurácia**	***Precision***	***Recall***	***F1-Score***	**p-valor**
C4.5 (J48)	0.6614	0.6614	0.8000	0.7241	0.0019
Random Forest	0.6243	0.6250	0.8095	0.7053	0.0330
KNN	0.6138	0.6429	0.6857	0.6635	0.0616
C5.0	0.6085	0.6202	0.7619	0.6837	0.0817
Regressão Logística	0.6032	0.6056	0.8190	0.6963	0.1064
Naive Bayes	0.5873	0.6080	0.7238	0.6608	0.2107
AdaBoost	0.5820	0.6048	0.7143	0.6550	0.2556
SVM (Linear)	0.5820	0.6016	0.7333	0.6609	0.2556
SVM (Poly)	0.5608	0.6000	0.6286	0.6139	0.4719
SVM (Radial)	0.5503	0.5909	0.6190	0.6046	0.5879

Observa-se que o modelo C4.5 (J48) apresentou o melhor desempenho em termos de acurácia (66,14%), além de um p-valor altamente significativo (p = 0,0019), o que valida estatisticamente sua capacidade preditiva. O modelo Random Forest também obteve desempenho expressivo, com F1-score de 0,71 e p-valor inferior a 0,05. Já o modelo KNN apresentou bons resultados nas métricas de precisão e F1-score, mas seu p-valor (0,0616) não atinge o nível convencional de significância estatística.

Os demais modelos avaliados apresentaram desempenhos inferiores ou estatisticamente não significativos. O C5.0, por exemplo, obteve métricas consistentes de precisão (62,02%) e recall (76,19%), porém com p-valor de 0,0817, não atingindo o limiar de significância adotado. A regressão logística apresentou o segundo maior recall (81,90%), o que indica boa capacidade de identificação de casos positivos, mas sua acurácia (60,32%) e p-valor (0,1064) foram inferiores aos modelos selecionados. Já os modelos SVM — nas variantes linear, polinomial e radial — tiveram desempenhos mais modestos em todas as métricas avaliadas, com p-valores acima de 0,25, o que indica que seu desempenho não difere significativamente do acaso.

Por fim, os modelos Naive Bayes e AdaBoost, apesar de apresentarem métricas intermediárias, também não demonstraram significância estatística suficiente para justificar sua seleção. Dessa forma, apenas os modelos que apresentaram p-valor significativo (p $<$0,05) — neste caso, C4.5 e Random Forest — foram considerados os melhores candidatos à análise mais aprofundada nas subseções seguintes.

5.4.2 Modelos com Melhor Desempenho

5.4.2.1 C4.5 (J48)

O primeiro modelo selecionado para análise detalhada foi o C4.5 (J48), utilizado para construção de árvores de decisão. Esse modelo obteve a maior acurácia entre todos os al-

goritmos avaliados (66,14%), além de um p-valor altamente significativo (p = 0,0019), indicando que seu desempenho é estatisticamente superior ao de um classificador aleatório.

A matriz de confusão apresentada na Tabela 19 mostra que o modelo conseguiu classificar corretamente 84 casos positivos (“Sim”) e 41 negativos (“Não”), totalizando 125 acertos. O modelo demonstrou boa capacidade de identificar casos de lesão (alta sensibilidade), o que é desejável em cenários em que os falsos negativos (lesionados não identificados) têm maior impacto.

Tabela 19: Matriz de confusão do modelo C4.5 (J48)

Predição	**Sim (real)**	**Não (real)**
Sim (previsto)	84	43
Não (previsto)	21	41

De acordo com Witten and Frank [88] O C4.5(J48) permite o ajuste de dois hiperparâmetros principais, que afetam diretamente a complexidade e o desempenho do modelo gerado:

- ***Confidence Factor (C):*** Este parâmetro controla o nível de confiança aplicado durante o processo de poda da árvore. A poda é responsável por reduzir o risco de sobreajuste ao eliminar subdivisões que não apresentam ganho estatístico significativo. Quanto menor o valor de C, mais agressiva será a poda, resultando em árvores menores e mais generalistas [88].

- ***Minimum Number of Instances per Leaf (M):*** Define o número mínimo de instâncias que um nó precisa conter para ser convertido em uma folha. Valores maiores de M levam a árvores mais simplificadas, prevenindo a geração de folhas com pouca representatividade, o que pode contribuir para maior generalização do modelo [88].

A Figura 32 apresenta os resultados do processo de ajuste fino (*tuning*) do modelo C4.5 (J48), com o objetivo de identificar a combinação ideal dos hiperparâmetros C e M para maximizar o desempenho preditivo. Observa-se que, de forma geral, valores mais altos de C (acima de 0,5) não contribuíram significativamente para a melhoria da acurácia, frequentemente resultando em desempenho inferior ou semelhante às configurações com menor grau de confiança.

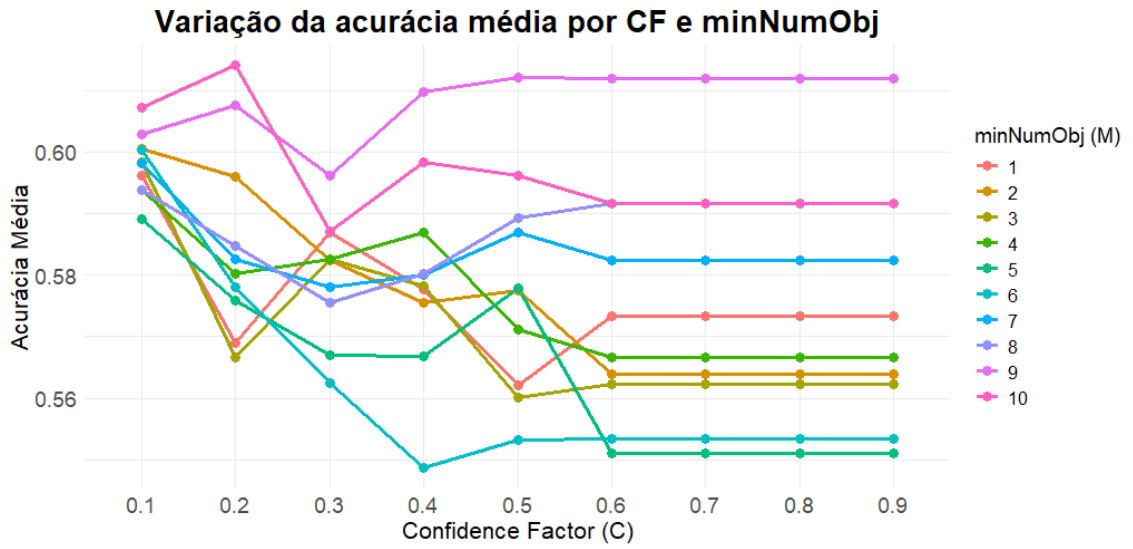


Figura 32: Acurácia média do modelo C4.5 (J48) por combinações dos hiperparâmetros *Confidence Factor* (C) e *minNumObj* (M).

A melhor configuração identificada foi C = 0,2 e M = 10, que alcançou a maior acurácia média durante a validação cruzada de 10 *folds*, com valor de 0,6141. Vale destacar que, apesar de configurações como C = 0,4, M = 9 e C = 0,5, M = 9 também apresentarem bons resultados (acima de 0,60 de acurácia), o modelo com C = 0,2 e M = 10 mostrou-se mais consistente e com melhor equilíbrio entre as métricas avaliadas. Com base nisso, esta foi a configuração escolhida para o treinamento final do modelo C4.5 neste estudo.

A Figura 33 representa a estrutura final da árvore gerada pelo modelo C4.5 após o processo de ajuste dos hiperparâmetros. A árvore resultante possui 10 folhas e um total de 19 nós, indicando um modelo de complexidade moderada.

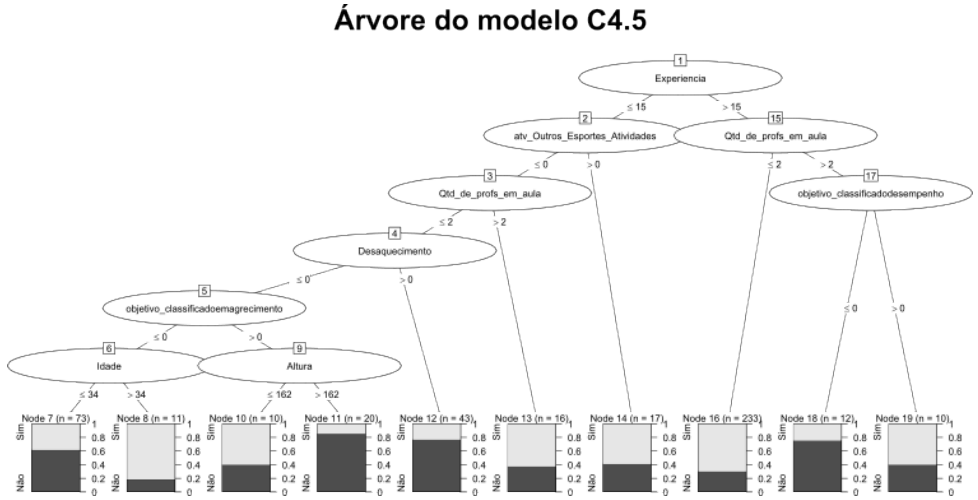


Figura 33: Árvore de decisão gerada pelo modelo C4.5 (J48) após tuning dos hiperparâmetros.

A variável “EXPERIENCIA” foi identificada como o principal critério de divisão na

raiz da árvore, separando os participantes com até 15 meses de prática daqueles com mais tempo de experiência. Esse ponto de corte inicial já sugere uma diferença marcante no risco de lesão entre iniciantes e praticantes mais experientes.

Para os indivíduos com menos de 15 meses de experiência, observou-se uma sequência de divisões adicionais envolvendo as variáveis "ATV_OUTROS_ESPORTES_ATIVIDADES", "QTD_DE_PROFS_EM_AULA", "DESAQUECIMENTO", "OBJETIVO" e "IDADE". Essas divisões refletem a influência de múltiplos fatores comportamentais e fisiológicos na predição de lesões nesse grupo. Notadamente, a presença de outras atividades físicas, a ausência de desaquecimento e objetivos voltados ao emagrecimento parecem estar associados a uma maior probabilidade de lesões.

Já para os praticantes com mais de 15 meses de experiência, a árvore apresenta uma estrutura mais simples, com destaque para a variável "QTD_DE_PROFS_EM_AULA" como principal discriminador. Interessantemente, participantes com maior número de professores por aula (mais de 2) foram classificados com base em seu objetivo de treino, especialmente o foco em desempenho, indicando que a combinação de supervisão qualificada e objetivos específicos pode influenciar o risco de lesão também entre os mais experientes.

Esses resultados reforçam a capacidade interpretativa do modelo C4.5, fornecendo uma estrutura de decisão compreensível e coerente com aspectos conhecidos do treinamento físico e da prevenção de lesões.

5.4.2.2 Random Forest

Este modelo obteve a segunda maior acurácia (62,43%) entre os algoritmos avaliados e um p-valor de 0,0330, indicando que a acurácia obtida é estatisticamente superior à taxa de acerto por acaso, estatisticamente confirmando a sua capacidade preditiva.

A Tabela 20 apresenta a matriz de confusão referente ao modelo *Random Forest*. Observa-se que o modelo obteve 85 classificações corretas da classe "Sim" (verdadeiros positivos) e 33 acertos para a classe "Não" (verdadeiros negativos). Por outro lado, foram registrados 51 falsos positivos, em que a classe "Não" foi erroneamente classificada como "Sim", e 20 falsos negativos, em que a classe "Sim" foi classificada incorretamente como "Não".

Tabela 20: Matriz de confusão do modelo Random Forest

Predição	**Sim (real)**	**Não (real)**
Sim (previsto)	85	51
Não (previsto)	20	33

A sensibilidade (*recall*) do modelo, isto é, sua capacidade de identificar corretamente a classe positiva ("Sim"), foi de 80,95%, o que demonstra uma boa capacidade de detecção

dos casos positivos. No entanto, a especificidade, que reflete a capacidade de identificar corretamente a classe negativa (“Não”), foi de apenas 39,29%, indicando dificuldade em classificar corretamente os casos negativos.

De forma geral, o Random Forest apresentou bom desempenho na identificação da classe majoritária (“Sim”), mas com dificuldades consideráveis na correta classificação da classe minoritária (“Não”), o que deve ser considerado na escolha do modelo final, especialmente em contextos onde os falsos positivos representam um risco maior.

No *ranger*, que é uma implementação eficiente do *Random Forest*, alguns parâmetros importantes controlam o comportamento do modelo:

- ***mtry***: Número de variáveis aleatórias consideradas em cada divisão. Valores menores favorecem maior variabilidade entre as árvores; valores maiores podem levar ao sobreajuste [62].
- ***splitrule***: Critério de divisão utilizado para determinar o melhor ponto de separação em cada nó. Para classificação, os critérios disponíveis incluem `"gini"` e `"extratrees"` [62].
- ***min.node.size***: Define o número mínimo de observações em um nó para que ele seja considerado terminal. Valores maiores geram árvores mais rasas, promovendo maior generalização [62].

A Tabela 21 apresenta as melhores combinações de hiperparâmetros obtidas para o modelo Random Forest. A estrutura da tabela foi organizada de modo a agrupar os resultados conforme o número de árvores (*num.trees*) utilizadas na floresta, permitindo uma visualização mais clara do desempenho obtido em cada cenário.

Tabela 21: Melhores combinações de hiperparâmetros para o Random Forest, agrupadas por número de árvores

num.trees	weight	mtry	splitrule	min.node.size	Acurácia	Precisão	Recall	F1-Score	p-valor
	1.3	5	extratrees	1	0.6085	0.6202	0.7619	0.6838	0.0817
1000	1.2	4	extratrees	1	0.6032	0.6136	0.7714	0.6835	0.1064
	1.1	4	extratrees	1	0.5979	0.6160	0.7333	0.6696	0.1360
	1.2	5	extratrees	1	0.6085	0.6202	0.7619	0.6838	0.0817
1250	1.3	6	extratrees	1	0.6085	0.6220	0.7524	0.6810	0.0817
	1.1	4	extratrees	1	0.6032	0.6190	0.7429	0.6753	0.1064
	1.1	**3**	**extratrees**	**10**	**0.6243**	**0.6250**	**0.8095**	**0.7053**	**0.0330**
1500	1.2	4	extratrees	1	0.5979	0.6107	0.7619	0.6780	0.1360
	1.3	5	extratrees	5	0.5979	0.6074	0.7810	0.6833	0.1360

Além dos hiperparâmetros tradicionais do pacote `ranger` — *mtry* (número de variáveis consideradas em cada divisão), *splitrule* (critério de divisão) e *min.node.size*

(tamanho mínimo dos nós) —, o processo de tuning considerou variações nos parâmetros *num.trees* e *weight*, que foram combinados de forma cruzada. Para cada par (*num.trees*, *weight*), foi realizada uma validação cruzada com 10 folds, e em seguida foram ajustadas as demais combinações de hiperparâmetros internos.

Cada linha da tabela representa a melhor configuração interna (*mtry*, *splitrule*, *min.node.size*) encontrada para um par específico de *num.trees* e *weight*. Entre os resultados obtidos, destaca-se a configuração com 1500 árvores, peso 1.1, *mtry* igual a 3 e *min.node.size* igual a 10, que alcançou a maior acurácia (0,6243) e o maior F1-Score (0,7053), com significância estatística (p-valor = 0,0330). Esse desempenho indica um bom equilíbrio entre precisão e sensibilidade na classificação da classe positiva, sendo, portanto, considerada a melhor configuração para esse modelo dentro do conjunto de validação analisado.

A Figura 34 apresenta o desempenho do modelo Random Forest em diferentes combinações de hiperparâmetros, considerando o cenário que obteve os melhores resultados na Tabela 21, com *num.trees* igual a 1500 e *weight* de 1.1. No gráfico, observa-se a variação da acurácia em função do número de variáveis utilizadas em cada divisão da árvore (*mtry*), separando os resultados conforme o critério de divisão adotado — *gini* e *extratrees* — e o tamanho mínimo dos nós (*min.node.size*).

De forma geral, o critério *extratrees* demonstrou desempenho superior ao *gini* em praticamente todas as combinações, apresentando acurácias mais altas e estáveis. Enquanto o *gini* mostra uma tendência de queda no desempenho à medida que o valor de *mtry* aumenta, o *extratrees* mantém resultados mais consistentes, especialmente para *min.node.size* igual a 1 e 10. A melhor configuração visualizada no gráfico ocorre para *mtry = 3* e *min.node.size = 10*, exatamente a mesma combinação que obteve a maior acurácia (0,6243) e F1-Score (0,7053) no cenário de tuning com *num.trees* = 1500.

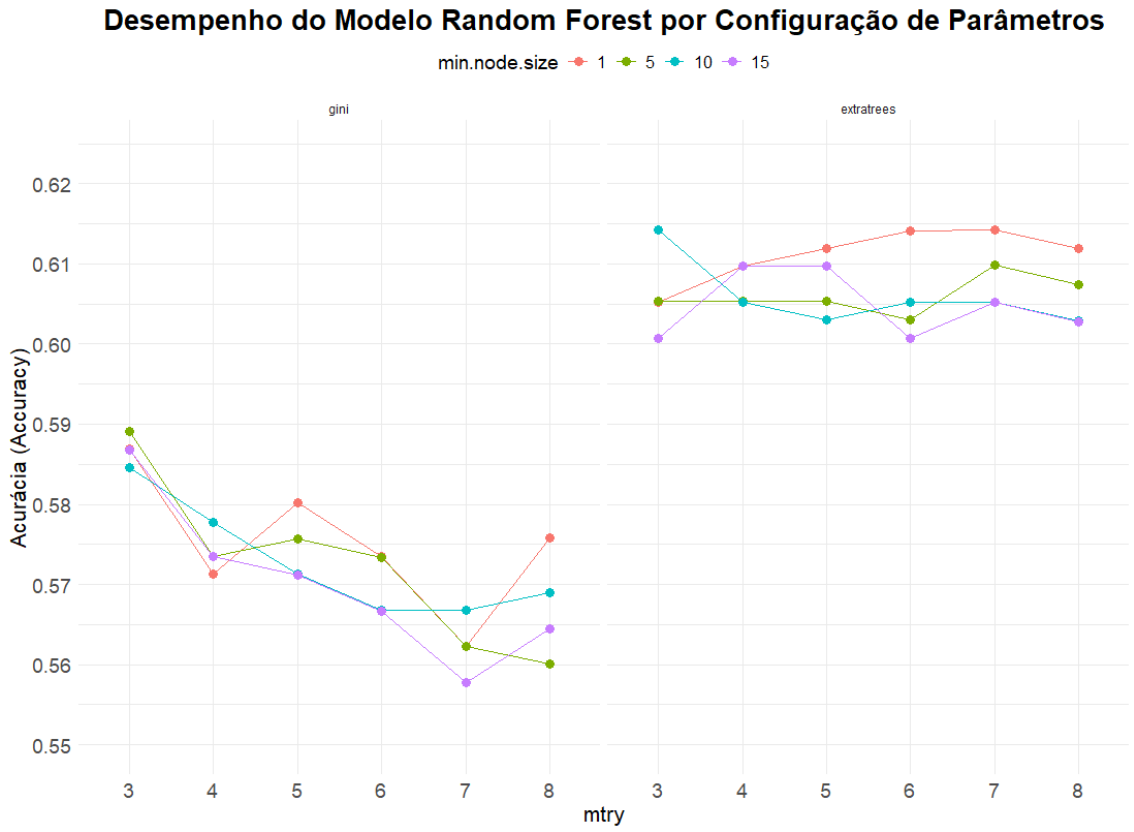


Figura 34: Desempenho do modelo Random Forest por combinação de hiperparâmetros. A figura representa os resultados obtidos com *num.trees* = 1500 e *weight* = 1.1, configuração que obteve o melhor desempenho na Tabela 21.

5.4.3 Escolha do Modelo Preditivo

A escolha do modelo preditivo final não se baseou exclusivamente nas métricas tradicionais de avaliação, como acurácia, precisão ou F1-score, mas também considerou indicadores mais robustos da capacidade discriminativa dos algoritmos. Nesse contexto, a análise da curva ROC e da métrica AUC foi essencial para compreender o desempenho dos modelos frente ao desequilíbrio da base e à variação nos limiares de decisão.

A Figura 35 ilustra a comparação entre os dois modelos finalistas, evidenciando uma leve superioridade do algoritmo escolhido com base na área sob a curva.

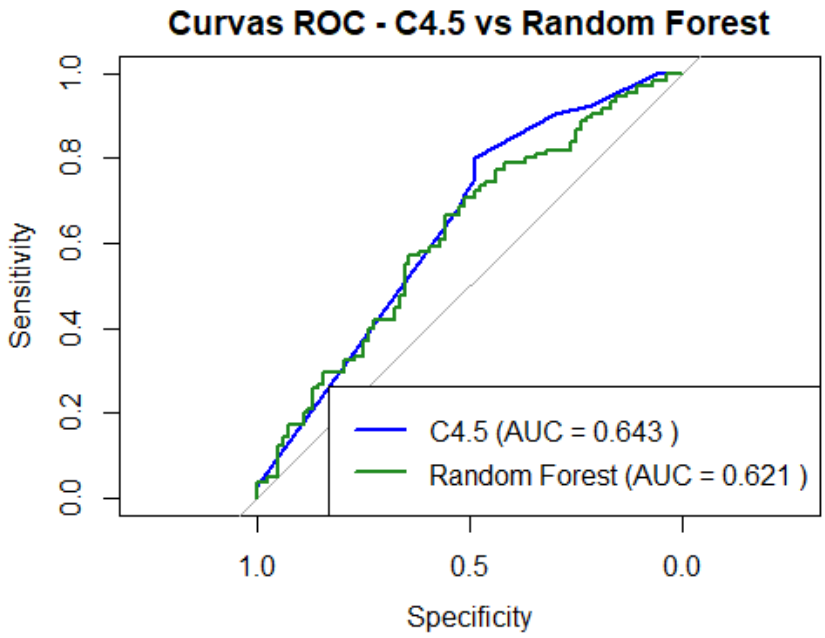


Figura 35: Curvas ROC comparando os algoritmos C4.5 e Random Forest no conjunto de teste

A análise da curva ROC foi conduzida para comparar a capacidade discriminativa dos modelos em predizer corretamente a ocorrência de lesões. A Figura 35 apresenta as curvas ROC dos algoritmos avaliados, com base nas probabilidades atribuídas à classe positiva “Sim” no conjunto de teste. Dentre os modelos comparados, o algoritmo C4.5 (J48) apresentou a melhor performance, com uma AUC de 0,643, demonstrando ligeira superioridade na capacidade de separação entre as classes.

O modelo *Random Forest*, embora também tenha se destacado entre os melhores, apresentou um valor de AUC = 0,621, confirmando um desempenho próximo, porém levemente inferior ao C4.5.

Com base nessa análise, somada aos resultados anteriores, reforça-se a escolha do C4.5 como o modelo com melhor equilíbrio entre desempenho estatístico e interpretabilidade, justificando sua seleção como modelo preditivo para a aplicação prática no contexto deste estudo.

5.5 Validação em Ambiente Interativo

Com base no modelo preditivo selecionado, foi desenvolvida uma aplicação interativa utilizando o *Shiny* para validar sua utilização em um cenário real. A aplicação resultante funcionou como um protótipo funcional, disponibilizado em ambiente web por meio da plataforma *Shinyapps.io*, possibilitando a coleta de respostas e dando subsídios para verificar a efetividade do modelo na predição de lesões, simulando seu uso prático em tempo real.

A interface do aplicativo foi desenvolvida com foco na simplicidade e usabilidade. Os participantes acessavam a URL pública disponibilizada (https://mdaltro.shinyapps.io/lesoesAppC45/), preenchiam o formulário com as informações relacionadas aos preditores do modelo e recebiam como resposta a probabilidade associada à ocorrência ou não de lesões. A Figura 36 ilustra a tela inicial da aplicação e o formulário de preenchimento.

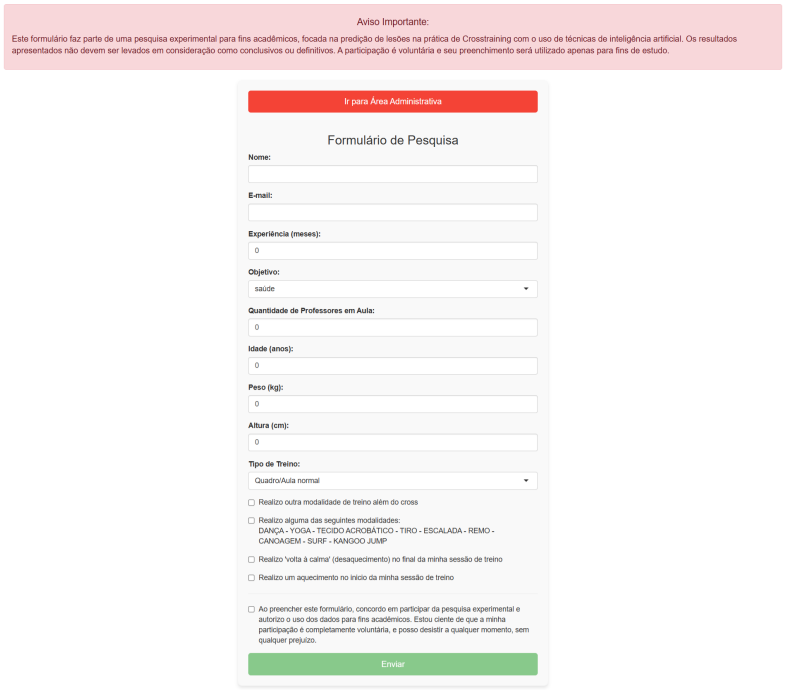


Figura 36: Interface inicial da aplicação e formulário de preenchimento para predição de lesões.

Os usuários foram convidados a participar de forma voluntária, sendo informados de que o formulário fazia parte de uma pesquisa experimental com fins acadêmicos, focada na predição de lesões na prática de Crosstraining por meio de técnicas de inteligência artificial. Foi destacado que os resultados gerados não deveriam ser considerados diagnósticos definitivos, e que os dados seriam utilizados exclusivamente para fins de estudo.

Além da funcionalidade principal de predição, a aplicação contou com um módulo restrito de visualização das respostas submetidas. Essa área, acessível apenas mediante autenticação (conforme ilustrado na Figura 37), foi implementada com o objetivo de oferecer uma camada de segurança para conferência e extração dos dados armazenados durante os testes do aplicativo.

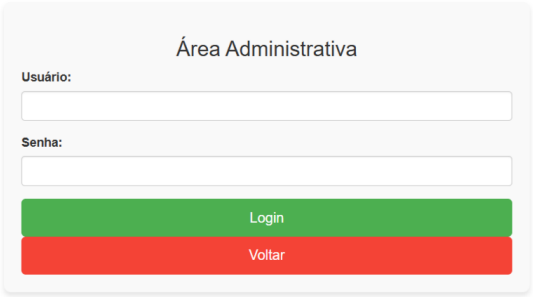


Figura 37: Tela de autenticação para acesso à área administrativa do aplicativo.

As informações preenchidas pelos usuários eram automaticamente enviadas e armazenadas em uma planilha hospedada no *Google Drive*, por meio de integração com os serviços do *Google Sheets* via API. Isso permitiu que os dados fossem registrados em tempo real, sem a necessidade de intervenção manual ou armazenamento local, facilitando tanto o monitoramento quanto a posterior análise dos dados coletados.

A interface de visualização incluía filtros básicos e permitia o download completo da base de respostas em formato .csv, o que viabilizou a exportação dos dados para ambientes estatísticos como o R. Essa funcionalidade se mostrou útil para verificar a integridade dos registros, identificar possíveis inconsistências no preenchimento e ampliar a base de observações disponíveis para reavaliação do desempenho do modelo em diferentes contextos. A Figura 38 apresenta a interface da área administrativa com a listagem dos dados submetidos.

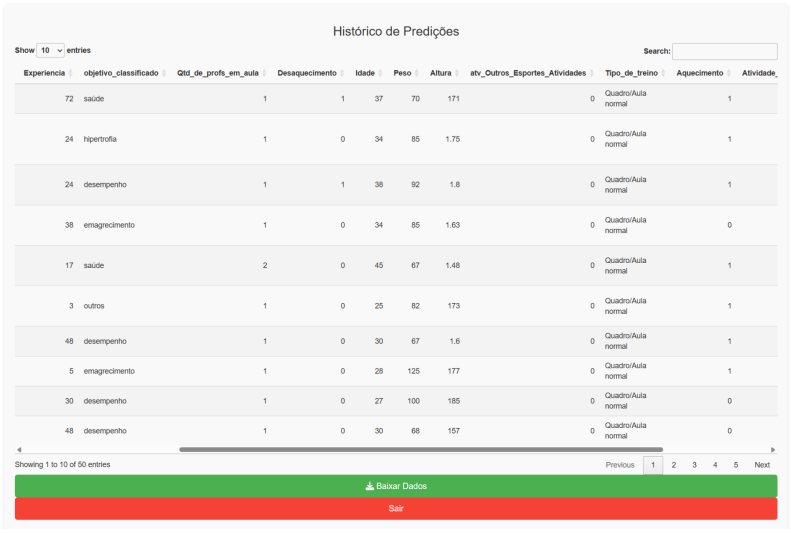


Figura 38: Área administrativa com visualização das respostas armazenadas no *Google Drive*.

5.5.1 Deploy e Configurações no Ambiente *Shinyapps.io*

O processo de deploy envolveu inicialmente a criação de uma conta na plataforma e a configuração de autenticação via pacote `rsconnect`, que permitiu a publicação direta

do aplicativo a partir do *RStudio*. O projeto foi estruturado em um único arquivo principal `app.R`, contendo todo o código da aplicação, além de dependências externas como os pacotes `shiny` e `googlesheets4` que viabilizaram a interface e a integração com o *Google Drive*.

Durante o processo de publicação, foi utilizada uma chave secreta de autenticação (token), que permite o vínculo seguro entre o ambiente local de desenvolvimento (RStudio) e a plataforma *Shinyapps.io*. Esse token é fundamental para autenticar o usuário e autorizar o envio de aplicações ao servidor remoto, evitando o uso de credenciais diretamente no código.

A Figura 39 ilustra a tela de gerenciamento dos tokens na plataforma, onde é possível criar e revogar chaves de autenticação associadas a cada dispositivo.

Figura 39: Tela de gerenciamento de tokens de autenticação na plataforma *Shinyapps.io*.

Durante a publicação, foram definidos os parâmetros de uso da aplicação, incluindo o número máximo de conexões simultâneas (limitado no plano gratuito), o tempo de inatividade permitido por sessão e o nível de acesso ao app, configurado como público para o formulário. Além disso, o ambiente oferece configurações de desempenho relacionadas ao uso da aplicação, como a quantidade de instâncias (*workers*) simultâneas e a alocação de memória. A quantidade de instâncias define quantos usuários podem acessar o app de forma paralela e pode ser ajustada manualmente para equilibrar o desempenho e o consumo de recursos, especialmente em planos com limites de uso.

Uma vantagem importante da plataforma é a disponibilidade de um painel de monitoramento, que fornece métricas como número de acessos, tempo médio de uso, além de registros de log de erros. Essas informações foram úteis para acompanhar o comportamento dos usuários durante o período de testes e realizar eventuais ajustes na aplicação.

A Figura 40 apresenta a interface de configuração das instâncias no painel do aplicativo, enquanto a Figura 41 exibe o painel de monitoramento, que oferece informações em tempo real sobre acessos, erros e consumo de recursos.

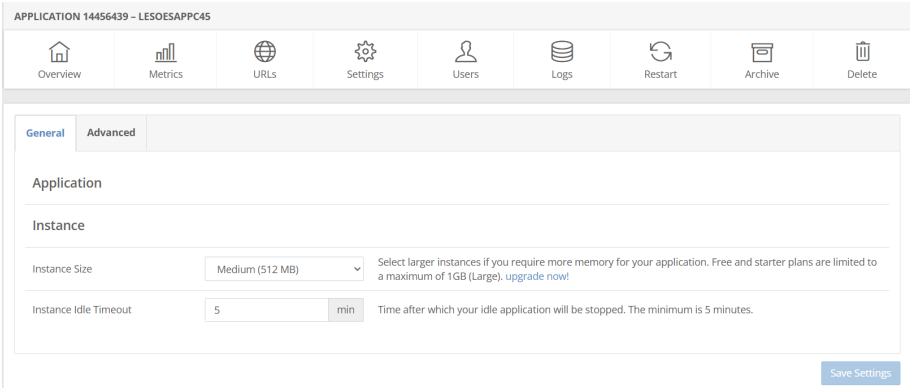


Figura 40: Configuração de número de instâncias e parâmetros de execução da aplicação.

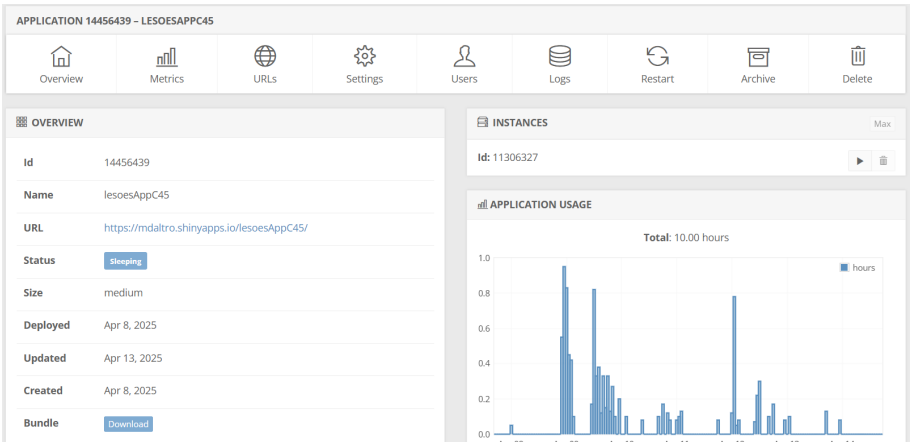


Figura 41: Painel de monitoramento e logs da aplicação, com visualização de erros e estatísticas de uso.

As configurações detalhadas do deploy da aplicação na plataforma Shinyapps.io são apresentadas na Tabela 22, refletindo os limites operacionais do plano gratuito e os ajustes realizados para garantir estabilidade durante o uso.

Tabela 22: Configurações utilizadas para o deploy da aplicação na plataforma Shinyapps.io

Parâmetro	Valor	Descrição
Tamanho da instância	Medium (512 MB)	Instância com memória intermediária. Instâncias maiores são indicadas para aplicações com maior demanda de recursos. O plano gratuito permite até 1 GB.
Tempo limite de inatividade da instância	5 min	Tempo após o qual uma instância inativa é interrompida automaticamente.
Número inicial de instâncias	1	Número de instâncias iniciadas ao carregar o app. Pode ser ajustado para lidar com picos de acesso.
Máximo de processos de trabalho	1	Define quantos processos de trabalho (workers) podem ser executados por instância.
Conexões simultâneas por processo	50	Número máximo de conexões ativas permitidas por processo de trabalho.
Fator de carga do trabalhador	5%	Percentual de carga que aciona a criação de um novo worker, até o limite definido.
Tempo limite de conexão	900 seg	Tempo máximo de inatividade da conexão entre navegador e worker antes de ser encerrada.
Tempo limite de leitura	3600 seg	Tempo máximo sem atividade de leitura entre navegador e worker. Usado para sessões interativas.
Tempo de inicialização do trabalhador	60 seg	Tempo de espera para que um worker inicialize. Aumentar esse valor ajuda com apps mais pesados.
Tempo de inatividade do processo	5 seg	Tempo antes de encerrar um processo de trabalho ocioso e sem conexões ativas.
Fator de carga da instância	50%	Percentual de carga que aciona a criação de uma nova instância, até o limite do plano.

Em termos de limitações, o plano gratuito do serviço impõe restrições quanto ao número de aplicações publicadas, ao volume de uso mensal e à quantidade de usuários simultâneos, o que pode comprometer a escalabilidade em projetos de maior porte. No entanto, para fins acadêmicos e de validação experimental, a plataforma se mostrou plenamente adequada, oferecendo uma solução prática e funcional para tornar o modelo acessível em um ambiente real.

5.5.2 Perfil da amostra utilizada na validação

Para a etapa de validação do modelo preditivo desenvolvido, foi selecionado um centro de treinamento com foco em *cross training* que conta atualmente com pouco mais de 70 alunos ativos. Durante uma semana, o sistema ficou disponível para os praticantes, que foram convidados a utilizar a aplicação e fornecer suas respostas. As informações inseridas foram baseadas na percepção individual dos usuários em relação aos seus treinos nos três meses anteriores. Isso permitiu que os dados fossem preenchidos de forma retrospectiva, considerando um intervalo recente e relevante para a análise de risco de lesões.

A aplicação proposta nesse trabalho foi validada por um total de 60 alunos (31 partici-

pantes do sexo feminino e 29 do sexo masculino), representando aproximadamente 85% da população-alvo da unidade. Todas as respostas foram consideradas para análise, sem a necessidade de processos adicionais de pré-processamento, uma vez que a aplicação foi previamente configurada para garantir a compatibilidade com o conjunto de dados utilizado no desenvolvimento do modelo.

O perfil demográfico dos participantes foi analisado com base na idade, experiência prévia com o *Cross training* e objetivos relacionados à prática. A Figura 42 apresenta a pirâmide etária da amostra, distribuída por sexo. Observa-se que a maior concentração de participantes encontra-se entre 28 e 38 anos, especialmente na faixa de 33 a 38 anos, com destaque para o sexo feminino (n = 12) e masculino (n = 13). Há baixa representatividade nas faixas etárias extremas, como menores de 23 anos e maiores de 48 anos.

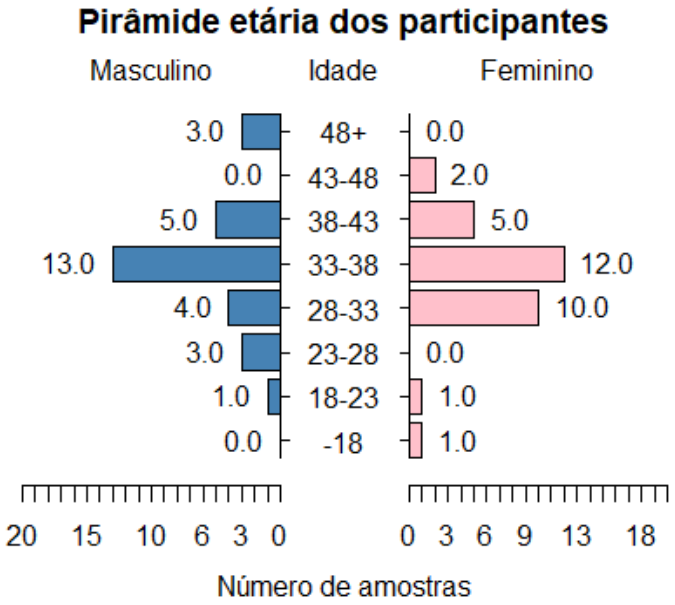


Figura 42: Pirâmide etária dos participantes por sexo.

A experiência com a modalidade foi categorizada em intervalos de 4 meses, conforme ilustra a Figura 43. Verifica-se que a maioria dos participantes possui entre 15 e 24 meses de experiência, com os maiores contingentes nas faixas de 15-19 e 20-24 meses (ambas com n = 10). Faixas intermediárias, como 35-39 e 45-49 meses, também apresentam valores expressivos. Após esse ponto, há uma queda indicando menor frequência de indivíduos com tempo de prática mais elevado. Apesar disso, nota-se ainda a presença de praticantes com mais de 60 meses de experiência, embora em menor proporção. Isso sugere um grupo predominantemente composto por indivíduos com experiência intermediária ou iniciante na prática de *Cross training*.

Por fim, a Figura 44 apresenta a distribuição dos objetivos pessoais dos participantes, agrupados por sexo. O principal motivo relatado foi a busca por saúde (n = 36),

com predominância feminina (n = 20). Objetivos secundários incluíram emagrecimento (n = 12), com maior representação entre as mulheres, e desempenho (n = 11), com maior representação entre os homens. Por outro lado, metas como hipertrofia e condicionamento físico para outros esportes foram menos citadas. Esses dados reforçam que o grupo valoriza majoritariamente os benefícios relacionados à saúde e qualidade de vida, mais do que aspectos estéticos ou competitivos.

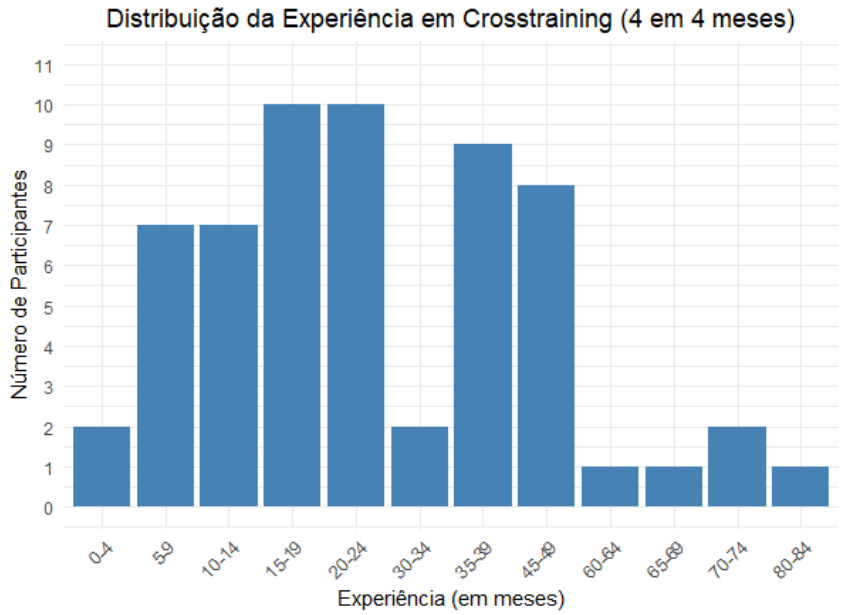


Figura 43: Distribuição da experiência em *Cross training* (em meses).

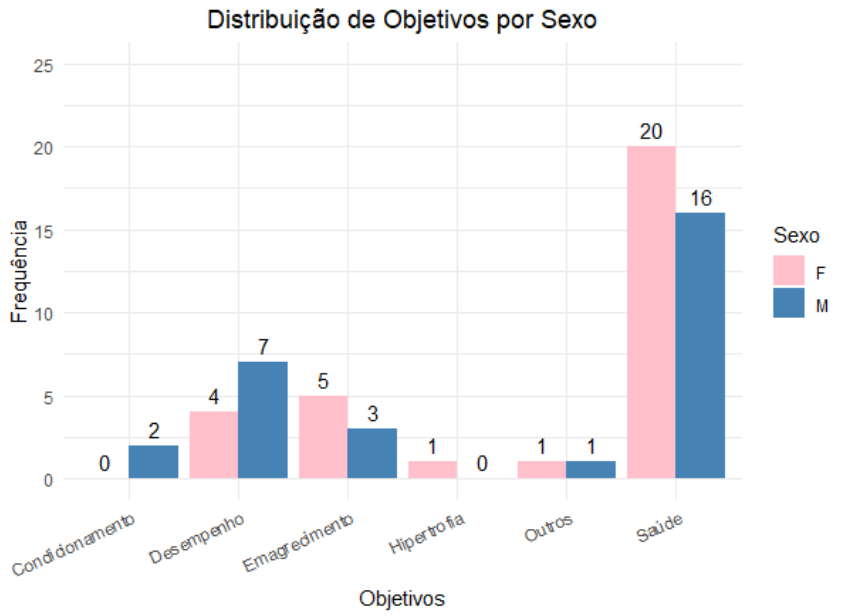


Figura 44: Distribuição dos objetivos dos participantes por sexo.

A Tabela 23 apresenta um resumo descritivo dos principais dados coletados por meio do formulário disponível na aplicação interativa.

Tabela 23: Resumo descritivo da amostra utilizada na validação

Característica	**Resumo**
Tamanho da amostra	60 participantes
Sexo	31 feminino (51,7%), 29 masculino (48,3%)
Idade (anos)	Média: 34,7; Mín: 15; Máx: 53
Peso (kg)	Média: 77,4; Mín: 46; Máx: 125
Altura (cm)	Média: 168,7; Mín: 142; Máx: 198
Experiência	Média: 27,8; Mediana: 24; Mín: 3; Máx: 84
Objetivo mais frequente	Saúde (36/60 – 60%)
Tipo de treino relatado	Predomínio de Quadro/Aula normal
Atividade extra	41% praticam outras atividades físicas
Aquecimento habitual	85% relataram realizar
Desaquecimento habitual	Apenas 6,7% relataram realizar
Qtd. de professores por aula	Moda: 1 professor

5.5.3 Avaliação das Predições

Cada submissão foi processada considerando os preditores previamente selecionados e resultou na atribuição de uma probabilidade associada à ocorrência de lesões. A partir disso, os participantes foram classificados nas classes SIM ou NÃO, conforme o risco estimado. A seguir, apresenta-se a distribuição geral das predições, bem como a análise das probabilidades atribuídas, com o objetivo de compreender a dinâmica dos resultados obtidos pela aplicação prática do modelo.

5.5.3.1 Distribuição das Respostas Submetidas

Após o período de validação em ambiente interativo, foram analisadas as 60 respostas submetidas pelos praticantes. A comparação entre as respostas reais (autorrelatadas) e as predições do modelo selecionado (C4.5) permitiu avaliar sua performance em um cenário real de aplicação. A Tabela 24 apresenta a matriz de confusão gerada com base nos dados obtidos.

Tabela 24: Matriz de confusão das predições realizadas na aplicação

Predição	**Sim (real**	**Não (real)**
Sim (previsto)	14	33
Não (previsto)	1	12

É possível observar que o modelo apresentou uma Acurácia de 43,3%, um valor inferior ao obtido durante a fase de treinamento e teste no dataset original, indicando uma

queda de desempenho em ambiente real. A *precision* de 29,8% evidencia a alta taxa de falsos positivos, já que grande parte das instâncias classificadas como "Sim" (lesão) não corresponde à realidade. No entanto, o modelo manteve um bom *recall* (93,3%), indicando que conseguiu identificar quase todos os casos positivos reais. O *F1-score*, que representa o equilíbrio entre *precision* e *recall*, foi de 44,9%, reforçando a tendência do modelo em priorizar a identificação de casos positivos. O *p-valor* obtido na hipótese de que a acurácia do modelo supera a taxa de acerto por aleatoriedade (*No Information Rate*) foi igual a 1, o que indica ausência de significância estatística. A Figura 45 ilustra de forma complementar e demonstra graficamente a diferença entre as distribuições reais e preditas.

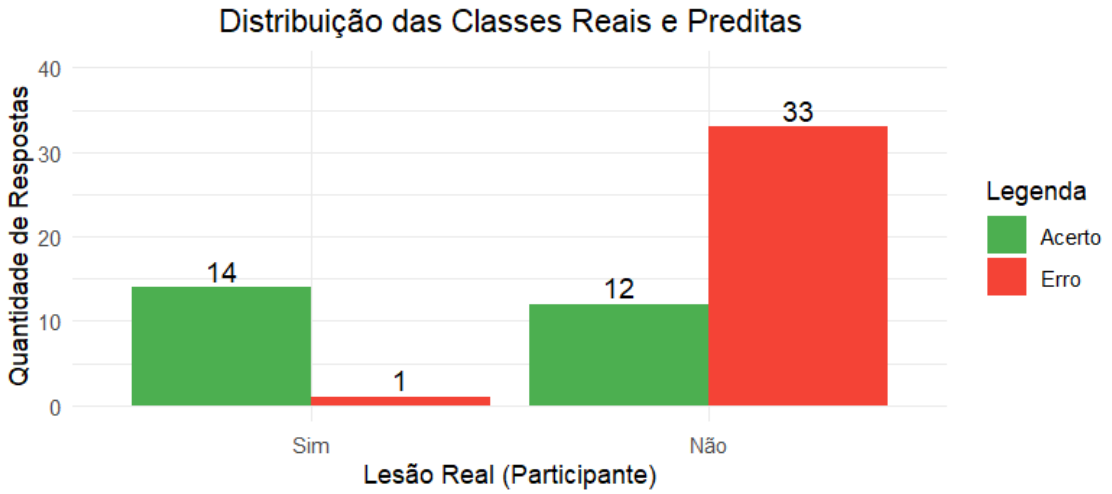


Figura 45: Distribuição das classes reais e preditas

Esses resultados sugerem que, embora o modelo seja eficaz para identificar a maioria dos casos positivos, sua aplicação prática ainda apresenta limitações importantes, especialmente no que diz respeito à especificidade. A discrepância entre o desempenho obtido nos testes iniciais e o observado em ambiente real evidencia os desafios de generalização de modelos preditivos, reforçando a importância da validação com dados externos e da consideração do contexto de aplicação.

5.6 Considerações Éticas na Utilização do Modelo Preditivo

A aplicação de modelos de aprendizado de máquina na área da saúde, como na predição de lesões em praticantes de *cross training*, embora promissora, levanta importantes questões éticas que devem ser consideradas para garantir uma aplicação responsável e segura.

Primeiramente, é crucial reconhecer que modelos preditivos baseados em IA não substituem o diagnóstico clínico individualizado. Sua eficácia depende da qualidade dos dados utilizados no treinamento e pode não capturar detalhes específicos de cada paciente.

Assim, devem ser utilizados como ferramentas complementares à avaliação clínica, auxiliando profissionais de saúde na tomada de decisões, mas nunca substituindo o julgamento clínico fundamentado na experiência e no contexto individual do paciente [79].

Segundo Kaur and Singh [42], a confiança excessiva em sistemas automatizados pode levar a diagnósticos equivocados e intervenções inadequadas, especialmente quando os modelos são aplicados fora do contexto para o qual foram originalmente desenvolvidos. A interpretação dos resultados fornecidos por esses modelos requer a expertise de profissionais qualificados, como médicos e educadores físicos, que possam contextualizar as predições dentro do histórico e das particularidades de cada indivíduo. A ausência dessa interpretação pode levar a decisões inadequadas, potencialmente prejudiciais ao paciente.

Por fim, a implementação de modelos preditivos em contextos esportivos deve ser acompanhada de diretrizes claras e regulamentações específicas que orientem seu uso responsável. Os profissionais devem ser capazes de compreender os critérios e as variáveis que influenciam as predições dos modelos, permitindo uma avaliação crítica de seus resultados. A colaboração entre desenvolvedores, profissionais de saúde, atletas e entidades reguladoras é crucial para estabelecer padrões éticos que assegurem a eficácia, segurança e justiça dessas ferramentas [69].

6 CONSIDERAÇÕES FINAIS

As considerações finais deste trabalho ressaltam a relevância da aplicação de técnicas de aprendizado de máquina (AM) na predição de lesões esportivas, com foco na modalidade *cross training*. Embora o uso de AM para esse fim já seja documentado em esportes como futebol, corrida e modalidades olímpicas, sua aplicação em contextos de treinamentos funcionais de alta intensidade como o *cross training* ainda é pouco explorada. Este estudo propôs uma abordagem prática e replicável para identificar fatores de risco associados à ocorrência de lesões, explorando o potencial de modelos preditivos treinados com dados reais de praticantes da modalidade.

Este trabalho teve como objetivo investigar a aplicabilidade de algoritmos de aprendizado de máquina (AM) na predição de lesões em praticantes de *Crosstraining*, por meio de um estudo comparativo. Com base em um conjunto de dados obtido através de um questionário aplicado a praticantes da modalidade, foram conduzidas etapas de tratamento, transformação e análise dos dados, seguidas do treinamento de diferentes modelos supervisionados.

A proposta metodológica foi centrada na comparação entre algoritmos de classificação, avaliando métricas como acurácia, precisão, *recall*, F1-Score, além da AUC e da curva ROC. Embora nenhum dos modelos tenha apresentado desempenho excepcional, o algoritmo C4.5 demonstrou resultados ligeiramente superiores em algumas das métricas utilizadas, além de ter demonstrado relevante significância estatística e possuir simplicidade interpretativa, o que o tornou o candidato ideal para validação prática.

No que se refere à viabilidade do uso de modelos preditivos como ferramenta de suporte à decisão, embora as métricas obtidas não sejam ideais para decisões clínicas ou intervenções diretas e apontem para desafios na generalização do modelo, os achados sustentam a hipótese de que algoritmos de AM podem ser utilizados como ferramentas complementares em estratégias de gestão de risco, orientação preventiva, especialmente se incorporados em plataformas digitais de fácil acesso, como demonstrado com o uso do aplicativo. Como sugestões para desdobramentos práticos, destaca-se a viabilidade de utilizar modelos semelhantes em ambientes de academia, como ferramenta complementar ao acompanhamento feito por profissionais. A integração com sistemas de gestão de

treino e uso contínuo para monitoramento pode ampliar o impacto de iniciativas como esta na promoção da saúde esportiva.

Em relação ao objetivo de identificar os fatores de risco mais relevantes, a análise da importância dos atributos destacou variáveis relacionadas ao tempo de prática, peso, altura, idade, objetivos, prática de atividades extras entre outros, como significativos na ocorrência de lesões. Entretanto, algumas limitações devem ser consideradas na interpretação desses resultados. O tamanho da amostra, é relativamente modesto para aplicações mais amplas de técnicas de Aprendizado de Máquina, podendo limitar a capacidade de generalização dos modelos construídos. Além disso, a validação dos casos de lesão baseou-se exclusivamente em relatos autodeclarados pelos participantes, sem confirmação clínica, o que pode comprometer a confiabilidade dos fatores identificados. Dessa forma, embora os achados deste estudo forneçam indícios importantes sobre fatores associados ao risco de lesões em praticantes de cross training, não é possível afirmar de maneira definitiva a causalidade ou a aplicabilidade universal dos resultados.

Como sugestões para estudos futuros, destaca-se a ampliação do volume de dados e a introdução de novas variáveis relacionadas a hábitos nutricionais, qualidade do sono, estratégias de recuperação, histórico de lesões, padrão de movimento e carga de treino. A integração dessas variáveis pode contribuir para o desenvolvimento de modelos mais robustos, com maior poder preditivo e aplicabilidade prática.

Este estudo, portanto, oferece uma contribuição à literatura e à prática, indicando caminhos promissores para a aplicação de inteligência artificial na prevenção de lesões em contextos de treino funcional e de alta intensidade. A consolidação de abordagens preditivas, alinhadas à realidade dos praticantes e acessíveis por meio de soluções interativas, representa um avanço no uso da ciência de dados no esporte.

REFERÊNCIAS

[1] Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., and Prasath, V. S. (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: a review. *Big data*, 7(4):221–248.

[2] Adeyemo, O., Adeyeye, T., and Ogunbiyi, D. (2015). Comparative study of id3/c4. 5 decision tree and multilayer perceptron algorithms for the prediction of typhoid fever. *African Journal of Computing & ICT*, 8(1):103–112.

[3] Aggarwal, C. C. (2015). *Data Mining - The Textbook*. Springer.

[4] Ahmed, S., Chowdhury, N., Rahman, N., and Arefin, M. N. I. (2023). Chronic kidney disease classification through hybrid feature selection. *International Journal of Statistics and Medical Research*, 12(1):29–38.

[5] Akter, S., Das, D., Haque, R. U., Tonmoy, M. I. Q., Hasan, M. R., Mahjabeen, S., and Ahmed, M. (2022). Ad-covnet: An exploratory analysis using a hybrid deep learning model to handle data imbalance, predict fatality, and risk factors in alzheimer's patients with covid-19. *Computers in Biology and Medicine*, 146:105657.

[6] Al-Shehari, T. and Alsowail, R. A. (2021). An insider data leakage detection using one-hot encoding, synthetic minority oversampling and machine learning techniques. *Entropy*, 23(10):1258.

[7] Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

[8] Altun, M., Gürüler, H., Özkaraca, O., Khan, F., Khan, J., and Lee, Y. (2023). Monkeypox detection using cnn with transfer learning. *Sensors*, 23(4):1783.

[9] Amendolara, A., Pfister, D., Settelmayer, M., Shah, M., Wu, V., Donnelly, S., Johnston, B., Peterson, R., Sant, D., Kriak, J., and Bills, K. (2023). An overview of machine learning applications in sports injury prediction. *Cureus*, 15.

[10] Awad, M. and Fraihat, S. (2023). Recursive feature elimination with cross-validation with decision tree: Feature selection method for machine learning-based intrusion detection systems. *Journal of Sensor and Actuator Networks*, 12(5).

[11] Beja-Battais, P. (2023). Overview of adaboost: Reconciling its views to better understand its dynamics. *arXiv preprint arXiv:2310.18323*.

[12] Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, volume 4. Springer.

[13] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[14] Buyse, L., Decroix, L., Timmermans, N., Barbé, K., Verrelst, R., and Meeusen, R. (2019). Improving the diagnosis of nonfunctional overreaching and overtraining syndrome. *Medicine and Science in Sports and Exercise*, 51(12):2524–2530.

[15] Chen, J., Li, H., Wu, F., and Zhang, Y. (2016). A lasso-based approach for disease classification and feature selection. In *2016 IEEE International Conference on Computer and Automation Engineering (ICACA)*, pages 314–318. IEEE.

[16] Chen, X.-w. and Jeong, J. C. (2007). Enhanced recursive feature elimination. In *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*, pages 429–435.

[17] Cheng, A. J., Jude, B., and Lanner, J. T. (2020). Intramuscular mechanisms of overtraining. *Redox biology*, 35:101480.

[18] Chicco, D. and Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13.

[19] Choi, R., Coyner, A., Kalpathy-Cramer, J., Chiang, M., and Campbell, J. (2020). Introduction to machine learning, neural networks, and deep learning. *Translational vision science & technology*, 9:14.

[20] CrossFit Inc. (2020). *CrossFit Level 1 Training Guide*. CrossFit Inc., 3rd edition. Available online at: `https://library.crossfit.com/free/pdf/CFJ_English_Level1_TrainingGuide.pdf`.

[21] Cunningham, P. and Delany, S. J. (2021). K-nearest neighbour classifiers-a tutorial. *ACM computing surveys (CSUR)*, 54(6):1–25.

[22] Eetvelde, H., De Michelis Mendonça, L., Ley, C., Seil, R., and Tischer, T. (2021). Machine learning methods in sport injury prediction and prevention: a systematic review. *Journal of Experimental Orthopaedics*, 8:27.

[23] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Mag.*, 17(3):37–54.

[24] Feito, Y., Heinrich, K., Butcher, S., and Poston, W. (2018). High-intensity functional training (hift): Definition and research implications for improved fitness. 6:76.

[25] Field, A. (2012). *Discovering Statistics Using R*. SAGE Publications, London, 2 edition.

[26] Flach, P. A. (2016). Roc analysis. In *Encyclopedia of machine learning and data mining*, pages 1–8. Springer.

[27] Geetha, V., Punitha, A., Abarna, M., Akshaya, M., Illakiya, S., and Janani, A. (2020). An effective crop prediction using random forest algorithm. In *2020 international conference on system, computation, automation and networking (ICSCAN)*, pages 1–5. IEEE.

[28] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(null):1157–1182.

[29] Halder, R. K., Uddin, M. N., Uddin, M. A., Aryal, S., and Khraisat, A. (2024). Enhancing k-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications. *Journal of Big Data*, 11(1):113.

[30] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 3rd edition.

[31] Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

[32] Hatwell, J., Gaber, M. M., and Atif Azad, R. M. (2020). Ada-whips: explaining adaboost classification with applications in the health sciences. *BMC Medical Informatics and Decision Making*, 20:1–25.

[33] Heine, M. (1999). Reassessing and extending the precision and recall concepts. In *MIRA'99*. BCS Learning & Development.

[34] Henriquez, M., Sumner, J., Faherty, M., Sell, T., and Bent, B. (2020). Machine learning to predict lower extremity musculoskeletal injury risk in student athletes. *Frontiers in Sports and Active Living*, 2:576655.

[35] Hosmer, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied Logistic Regression*. Wiley, 3rd edition.

[36] Huang, C. and Jiang, L. (2021). Data monitoring and sports injury prediction model based on embedded system and machine learning algorithm. *Microprocessors and Microsystems*, 81:103654.

[37] Jauhiainen, S., Kauppi, J.-P., Leppänen, M., Pasanen, K., Parkkari, J., Vasankari, T., Kannus, P., and Äyrämö, S. (2020). New machine learning approach for detection of injury risk factors in young team sport athletes. *International journal of sports medicine*, 42.

[38] Jenefa, A. and Moses, M. B. (2018). An upgraded c5. 0 algorithm for network application identification. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 789–794. IEEE.

[39] Jibril, M., Bello, A., Aminu, I. I., Ibrahim, A. S., Bashir, A., Malami, S. I., Habibu, M., and Magaji, M. M. (2022). An overview of streamflow prediction using random forest algorithm. *GSC Advanced Research and Reviews*, 13(1):050–057.

[40] Jun, Z. (2021). The development and application of support vector machine. In *Journal of Physics: Conference Series*, volume 1748, page 052006. IOP Publishing.

[41] Karthikeyan, V. and Suja Priyadharsini, S. (2021). A strong hybrid adaboost classification algorithm for speaker recognition. *Sādhanā*, 46(3):138.

[42] Kaur, J. and Singh, R. (2024). Ethical issues of artificial intelligence in medicine and healthcare. *Cureus*, 16(2).

[43] Kursa, M. B. and Rudnicki, W. (2011). The all relevant feature selection using random forest. *arXiv preprint arXiv:1106.0255*.

[44] Kursa, M. B. and Rudnicki, W. R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13.

[45] Las Johansen, B. C. and Trecene, J. K. D. (2018). Predicting academic performance of information technology students using c4. 5 classification algorithm: a model development. *International Journal of Information Sciences and Application*, 10(1):7–21.

[46] Lorena, A., Faceli, K., Almeida, T., de Carvalho, A., and Gama, J. (2021). *Inteligência Artificial: uma abordagem de Aprendizado de Máquina (2a edição)*.

[47] Lövdal, S., Den Hartigh, R., and Azzopardi, G. (2021). Injury prediction in competitive runners with machine learning. *International Journal of Sports Physiology and Performance*.

[48] Mahboob, T., Irfan, S., and Karamat, A. (2016). A machine learning approach for student assessment in e-learning using quinlan's c4. 5, naive bayes and random forest algorithms. In *2016 19th international multi-topic conference (INMIC)*, pages 1–8. IEEE.

[49] Manikandan, G., Pragadeesh, B., Manojkumar, V., Karthikeyan, A., Manikandan, R., and Gandomi, A. H. (2024). Classification models combined with boruta feature selection for heart disease prediction. *Informatics in Medicine Unlocked*, 44:101442.

[50] Maswadi, K., Ghani, N. A., Hamid, S., and Rasheed, M. B. (2021). Human activity classification using decision tree and naïve bayes classifiers. *Multimedia Tools and Applications*, 80:21709–21726.

[51] Maurya, A. (2012). *Running Lean: Iterate from Plan A to a Plan That Works*. O'Reilly Media, Inc., 2nd edition.

[52] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition.

[53] Moustakidis, S., Siouras, A., Vassis, K., Ioannis, M., Papageorgiou, E., and Tsaopoulos, D. (2022). Prediction of injuries in crossfit training: A machine learning perspective. *Algorithms*, 15:77.

[54] Naglah, A., Khalifa, F., Mahmoud, A., Ghazal, M., Jones, P., Murray, T., Elmaghraby, A., and El-Baz, A. (2018). Athlete-customized injury prediction using training load statistical records and machine learning. pages 459–464.

[55] Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, 4:51–62.

[56] Oliver, J., Ayala, F., De Ste Croix, M., Lloyd, R., Myer, G., and Read, P. (2020). Using machine learning to improve our understanding of injury risk and prediction in elite male youth football players. *Journal of Science and Medicine in Sport*, 23.

[57] Potdar, K., Pardawala, T., and Pai, C. (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9.

[58] Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

[59] Powers, D. M. W. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

[60] Priyanka and Kumar, D. (2020). Decision tree classifier: a detailed survey. *International Journal of Information and Decision Sciences*, 12(3):246–269.

[61] Priyatno, A. M. and Widiyaningtyas, T. (2024). A systematic literature review: Recursive feature elimination algorithms. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 9(2):196–207.

[62] Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):e1301.

[63] Resende, P. A. A. and Drummond, A. C. (2018). A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, 51(3):1–36.

[64] Rommers, N., Rössler, R., Verhagen, E., Vandecasteele, F., Verstockt, S., Vaeyens, R., Lenoir, M., and D'Hondt, E. (2020). A machine learning approach to assess injury risk in elite youth football players. *Medicine Science in Sports & Exercise*, 52:1.

[65] Schölkopf, B. (2000). The kernel trick for distances. In Leen, T., Dietterich, T., and Tresp, V., editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

[66] Scornet, E. (2020). Trees, forests, and impurity-based variable importance. *arXiv preprint arXiv:2001.04295*.

[67] Serafim, T. T., Maffulli, N., Migliorini, F., and Okubo, R. (2022). Epidemiology of high intensity functional training (hift) injuries in brazil. *Journal of Orthopaedic Surgery and Research*, 17.

[68] Shringarpure, A., Shetty, R., Surve, A., and Vidhate, A. (2022). Sports injury prediction system using random forest classifier. In *ITM Web of Conferences*, volume 44, page 03068. EDP Sciences.

[69] Silva, M. A., Pereira, L. M., and Santos, T. R. (2024). Ética e aplicação de modelos de inteligência artificial no esporte: Um ensaio crítico. *Revista Brasileira de Bioética*, 20(1):45–58.

[70] Souza, D. C., Arruda, A., and Gentil, P. (2017). Crossfit®: Riscos para possíveis benefícios? *Revista brasileira de prescrição e fisiologia do exercício*, 11(64):138–139.

[71] Sprey, J., Ferreira, T., Vaz de Lima, M., Duarte, A., Jorge, P., and Santili, C. (2016). An epidemiological profile of crossfit athletes in brazil. *Orthopaedic Journal of Sports Medicine*, 4.

[72] Subbiah, S., Anbananthen, K. S. M., Thangaraj, S., Kannan, S., and Chelliah, D. (2022). Intrusion detection technique in wireless sensor network using grid search random forest with boruta feature selection algorithm. *Journal of Communications and Networks*, 24(2):264–273.

[73] Suthaharan, S. and Suthaharan, S. (2016). Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235.

[74] Sutton, C. D. (2005). 11 - classification and regression trees, bagging, and boosting. In Rao, C., Wegman, E., and Solka, J., editors, *Data Mining and Data Visualization*, volume 24 of *Handbook of Statistics*, pages 303–329. Elsevier.

[75] Szajkowski, S., Dwornik, M., Pasek, J., and Cieslar, G. (2023). Risk factors for injury in crossfit®—a retrospective analysis. *International Journal of Environmental Research and Public Health*, 20:2211.

[76] Tan, P.-N., Karpatne, A., Steinbach, M., and Kumar, V. (2019). *Introduction to Data Mining Global Edition*. Pearson Deutschland.

[77] Thiese, M., Ronna, B., and Ott, U. (2016). P value interpretations and considerations. *Journal of Thoracic Disease*, 8:E928–E931.

[78] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[79] Tilala, M., Chenchala, P. K., Choppadandi, A., Kaur, J., Naguri, S., Saoji, R., and Devaguptapu, B. (2024). Ethical considerations in the use of artificial intelligence and machine learning in health care: A comprehensive review. *Cureus*, 16.

[80] Timón, R., Olcina, G., Camacho-Cardeñosa, M., Camacho-Cardenosa, A., Martinez-Guardado, I., and Marcos-Serrano, M. (2019). 48-hour recovery of biochemical parameters and physical performance after two modalities of crossfit workouts. *Biology of Sport*, 36(3):283–289.

[81] Tripathi, P., Srivastava, A., and Chaurasia, V. (2023). Ensemble classifiers with hybrid feature selection approach for diagnosis of cad. *The Scientific Temper*, 14(1-2):131–140.

[82] Valkenborg, D., Rousseau, A.-J., Geubbelmans, M., and Burzykowski, T. (2023). Support vector machines. *American Journal of Orthodontics and Dentofacial Orthopedics*, 164(5):754–757.

[83] Vittinghoff, E. and McCulloch, C. E. (2007). Relaxing the rule of ten events per variable in logistic and cox regression. *American journal of epidemiology*, 165(6):710–718.

[84] Wagener, S., Hoppe, M., Hotfiel, T., Engelhardt, M., Javanmardi, S., Baumgart, C., and Freiwald, J. (2020). Crossfit® – development, benefits and risks. *Sports Orthopaedics and Traumatology*, 36.

[85] Wang, Y., Liao, W., Shen, H., Jiang, Z., and Zhou, J. (2024). Some notes on the basic concepts of support vector machines. *Journal of Computational Science*, 82:102390.

[86] Weisenthal, B., Beck, C., Maloney, M., DeHaven, K., and Giordano, B. (2014). Injury rate and patterns among crossfit athletes. *Orthopaedic Journal of Sports Medicine*, 2.

[87] Wickramasinghe, I. (2022). Applications of machine learning in cricket: A systematic review. *Machine Learning with Applications*, 10:100435.

[88] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition.

[89] Wu, X., Zhou, J., Zheng, M., Chen, S., Wang, D., Anajemba, J., Zhang, G., Abdelhaq, M., Alsaqour, R., and Uddin, M. (2022). Cloud-based deep learning-assisted system for diagnosis of sports injuries. *Journal of Cloud Computing*, 11.

[90] Xiang, X., Duan, S., Pan, H., Han, P., Cao, J., and Liu, C. (2020). From one-hot encoding to privacy-preserving synthetic electronic health records embedding. In *Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies*, pages 407–413.

[91] Xiaoliang, Z., Hongcan, Y., Jian, W., and Shangzhuo, W. (2009). Research and application of the improved algorithm c4. 5 on decision tree. In *2009 International Conference on Test and Measurement*, volume 2, pages 184–187. IEEE.

[92] Yu, L., Zhou, R., Chen, R., and Lai, K. K. (2022). Missing data preprocessing in credit classification: One-hot encoding or imputation? *Emerging Markets Finance and Trade*, 58(2):472–482.

[93] Zhang, Y., Yao, Y., Chen, S., Jin, P., Zhang, Y., Jin, J., and Lu, J. (2024). Rethinking guidance information to utilize unlabeled samples:a label encoding perspective.

[94] Zhu, M. (2004). Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2(30):6.

[95] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320.