Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da Computação

# Uso de MLOps para automatização do processamento de modelos de Aprendizado de Máquina no acompanhamento de sinais vitais

**Daniel Ribeiro Lavra**

JUIZ DE FORA
AGOSTO, 2024

# Uso de MLOps para automatização do processamento de modelos de Aprendizado de Máquina no acompanhamento de sinais vitais

Daniel Ribeiro Lavra

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Victor Ströele de Andrade Menezes

JUIZ DE FORA

AGOSTO, 2024

# Uso de MLOps para automatização do processamento de modelos de Aprendizado de Máquina no acompanhamento de sinais vitais

Daniel Ribeiro Lavra

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Victor Ströele de Andrade Menezes
Doutor em Engenharia de Sistemas e Computação - COPPE/UFRJ

Regina Maria Maciel Braga
Doutora em Engenharia de Sistemas e Computação - COPPE/UFRJ

Luciano Jerez Chaves
Doutor em Ciência da Computação - IC/UNICAMP

JUIZ DE FORA
21 DE AGOSTO, 2024

# Resumo

Atualmente, no cenário tecnológico, os algoritmos de Inteligência Artificial têm ganhado destaque em diversos setores da sociedade, representando um papel significativo na área da saúde. Ferramentas tecnológicas e modelos de Aprendizado de Máquina já estão sendo utilizados para acompanhar a situação de pacientes e apoiar especialistas no diagnóstico de doenças. Esses modelos podem oferecer informações valiosas para o monitoramento de pacientes. Entretanto, esses algoritmos precisam ser gerenciados de forma segura e escalável para apresentarem resultados corretos, principalmente, por envolverem a vida de pessoas. Nesse contexto, surge o conceito de MLOps, uma metodologia com o intuito de gerenciar todo o ciclo de vida desses tipos de sistemas que envolvem aprendizado de máquina. Este trabalho propõe um estudo sobre a viabilidade do uso de MLOps na automação de sistemas inteligentes que coletam dados vitais de pacientes para tomada de decisão usando Aprendizado de Máquina. Práticas de gerenciamento nesses modelos são essenciais, por se tratarem de modelos automatizados e adaptativos, além de também precisarem de um desempenho eficaz. O contexto do problema envolve a captura de dados de um usuário, o tratamento, e a construção de modelos de Aprendizado de Máquina mediante um gerenciamento por meio de práticas de MLOps, verificando como essas práticas podem auxiliar na automação de sistemas inteligentes.

**Palavras-chave:** MLOps, Aprendizado de Máquina, sinais vitais, **frequência cardíaca, bpm, monitoramento de saúde**, Inteligência Artificial.

# Abstract

In the technological scenario, Artificial Intelligence algorithms have gained prominence in various sectors of society, playing a significant role in health. Technological tools and Machine Learning models are already being used to monitor patients' situations and support specialists in diagnosing diseases. These algorithms can provide valuable information for patient monitoring. However, these algorithms must be managed safely and scalable to present correct results, mainly because they involve people's lives. In this context, the concept of MLOps emerges, intending to manage the entire life cycle of these systems involving machine learning. This work proposes a study on the feasibility of using MLOps to automate intelligent systems that collect vital patient data for decision-making using Machine Learning. Management practices in these models are essential, as they are automated and adaptive models and require effective performance. The context of the problem involves capturing user data, processing it, and building Machine Learning models with management through MLOps practices. The study will analyze how these practices can help automate intelligent systems.

**Keywords:** MLOps, Machine Learning, vital signs, **heart rate, bpm, health monitoring**, Artificial Intelligence.

# Agradecimentos

Primeiramente, à Deus. Aos meus pais e familiares, por me fornecerem todo apoio e suporte. Aos meus professores do Departamento de Ciência da Computação e aos demais professores que passaram por toda a minha jornada acadêmica.

# Conteúdo

# Lista de Figuras

# Lista de Tabelas

# Lista de Abreviações

| | |
|---|---|
| AAL | *Active Assisted Living* |
| bpm | Batimentos Por Minuto |
| CD4ML | *Continuous Delivery for Machine Learning* |
| GBC | *Gradient Boosting Classifier* |
| IA | Inteligência Artificial |
| IoT | Internet das Coisas |
| KNN | *K-Neighbors Regressor* |
| LR | *Logistic Regression* |
| ML | *Machine Learning* |
| SMS | *Short Message System* |
| SVM | *Support Vector Machine* |
| SVR | *Support Vector Regressor* |

# 1 Introdução

Neste capítulo, é abordado o impacto das tecnologias emergentes na saúde, ao explorar a integração de sensores e algoritmos de Inteligência Artificial (IA), assim como a importância da utilização das práticas MLOps (ALLA; ADARI, 2020), para gerenciamento do ciclo de vida de sistemas de Aprendizado de Máquina. Após isso, é discutido a contextualização dos sistemas baseados em Aprendizado de Máquina para este cenário, o seu funcionamento e a necessidade da adoção de medidas para acompanhar metricamente essas arquiteturas. A seguir, são apresentadas a descrição do problema situado, a justificativa e as questões desta pesquisa, e, posteriormente, os objetivos e a organização deste trabalho.

## 1.1 Contextualização

Já existem sistemas que monitoram a rotina de pacientes, capturam dados, analisam e fornecem informações em formato de *insights*. Existem arquiteturas que permitem monitorar e acompanhar as visitações aos cômodos da casa pelos pacientes idosos e/ou com alguma comorbidade crônica e comunicar esse padrão de atividades aos familiares ou especialistas, caso ocorra uma alteração no comportamento (LARCHER et al., 2020). Além desses tipos de sistemas, existem modelos que monitoram as batidas cardíacas de um paciente via dispositivos IoT, como sensores do relógio de pulso (SERGIO et al., 2023).

Por meio de dispositivos de Internet das Coisas (IoT, do inglês *Internet of Things* )(MADAKAM; RAMASWAMY; TRIPATHI, 2015), sensores, algoritmos de Inteligência Artificial (LONGO et al., 2020) e Aprendizado de Máquina (ML, do inglês *Machine Learning*) (CARBONELL; MICHALSKI; MITCHELL, 1983), é possível classificar, filtrar dados e obter informações úteis para a tomada de decisões e acompanhamento de pacientes. De fato, com algoritmos de classificação, é possível observar anomalias que fogem ao padrão estabelecido anteriormente, as quais podem ser verificadas e analisadas posteriormente por um médico (MADAKAM; RAMASWAMY; TRIPATHI, 2015).

Com o surgimento de novas tecnologias, como a Inteligência Artificial (IA) (LONGO et al., 2020), diversas áreas da sociedade foram beneficiadas, sendo uma delas a saúde. Ferramentas tecnológicas podem acompanhar pacientes, monitorar resultados, diagnosticar doenças e até mesmo fornecer informações sobre um determinado paciente. Atualmente, sistemas estão cada vez mais presentes no dia a dia de pessoas da área da saúde e, portanto, soluções tecnológicas devem ser qualificadas, seguras e constantemente atualizadas para serem fornecidas de forma eficiente aos profissionais da saúde e pacientes.

Esses algoritmos de Aprendizado de Máquina podem fornecer ótimos resultados e identificações para a saúde dos usuários, sistemas dessa magnitude precisam de um gerenciamento adequado para o tratamento dos seus modelos e automatização das tarefas. Com a adesão de algoritmos dessa magnitude, faz-se necessário utilizar práticas que permitam uma rápida interação entre as ferramentas de Aprendizado de Máquina para o desempenho ser monitorado. Através da coleta de métricas, e da automação desses modelos para gerenciamento eficaz e bom funcionamento do sistema, surge o contexto das práticas de MLOps para a gestão dos modelos. Neste trabalho, foi desenvolvida uma pesquisa para avaliar, nesse contexto, se essas práticas de gerenciamento surtem efeito nas métricas avaliadas para a seleção de um modelo mais eficiente.

## 1.2 Descrição do Problema

Diversos fatorem levam as pessoas ao não acompanhamento de sua saúde pessoal em uma frequência adequada. Portanto, é interessante o uso de sistemas automatizados que fornecessem informações sobre alguns sinais vitais do corpo humano. Com isso, surge a ideia de capturar dados dos usuários de relógios inteligentes, através de seus sensores, tratá-los, analisá-los e propor modelos que identificassem padrões com o auxílio de algoritmos de Aprendizado de Máquina. No entanto, alguns desses modelos foram construídos sem muita preocupação com as práticas da engenharia de software dos sistemas tradicionais, visto que as arquiteturas dos sistemas são diferentes e necessitam de práticas mais automatizadas.

Modelos de Aprendizado de Máquina necessitam de um fluxo constante de dados para o treinamento de seus sistemas. Todavia, o armazenamento desses dados em um

ambiente automatizado, considerando que os modelos devem estar atualizados constantemente, pode ser considerado uma tarefa complexa que exija um gerenciamento adequado. Diversos dados são produzidos pelos dispositivos a cada instante, e isso deve ser integrado constantemente na arquitetura do sistema para o modelo estar sempre atualizado conforme as informações recentes. Ademais, diversos treinamentos de modelos com diferentes algoritmos deixam o sistema mais complexo e cada vez mais dependente das práticas de MLOps para a construção de uma boa infraestrutura e a geração de boas predições (ALLA; ADARI, 2020).

As técnicas de MLOps permitem uma melhor integração dos modelos de produção nas diversas fases do ciclo de vida de uma infraestrutura de Aprendizado de Máquina. Em comparação com os *softwares* tradicionais, tem-se a Engenharia de *Software* que especifica os requisitos, melhores práticas, entre outros. Nesse contexto, tem-se o MLOps, como um conjunto dessas práticas de engenharia associadas a algoritmos de aprendizado de máquina.

## 1.3 Justificativa

Atualmente, com a utilização de sistemas de Aprendizado de máquina na área da saúde, é comum os usuários desejarem avaliar o desempenho dos algoritmos e propor melhorias caso seja necessário (DEO, 2015). No entanto, modelos automatizados são mais difíceis de serem comparados metricamente devido à sua automação e complexidade dos ciclos de vida. *Softwares* dessa categoria são adaptativos e variam conforme os dados tratados e os parâmetros especificados, seus modelos podem ser alterados conforme a mudança na entrada dos dados, diferentemente dos *softwares* tradicionais. Algoritmos são alimentados por determinados dados e podem ser retreinados por outros completamente diferentes dos anteriores que foram treinados, trazendo uma peculiaridade de autoadaptação a essa arquitetura.

Tratando-se de arquiteturas que envolvem a saúde das pessoas, é necessária a adoção de medidas que comprovem a eficácia e a segurança desses sistemas, além da necessidade de rápida transformação em suas configurações quando necessário. Por isso, é relevante verificar através das práticas de MLOps, como a adoção dessa metodologia

pode impactar na apresentação de melhores modelos, através da extração de métricas e comparações de algoritmos.

## 1.4 Questões de Pesquisa

A arquitetura proposta por Sergio et al. (2023) adota um modelo de Aprendizado de Máquina para extração dos dados vitais de um relógio inteligente de um usuário até o treinamento do modelo. No entanto, não fornece informações sobre o funcionamento do sistema, ou seja, não é possível estabelecer metricamente as qualidades dos sistemas. Como pesquisa deste trabalho, foi importante destacar e extrair informações desses sistemas automatizados com Aprendizado de Máquina, estabelecer suas métricas e gerar uma arquitetura semelhante utilizando um gerenciamento através das práticas de MLOps.

Assim, a questão de pesquisa deste trabalho é: *O uso de MLOps em sistemas de monitoramento de pessoas é uma solução viável para o gerenciamento e atualização dos modelos de Aprendizagem de Máquina?*

## 1.5 Objetivos

Esta pesquisa estudou e investigou práticas de MLOps no cenário de Aprendizado de Máquina para acompanhamento da saúde de usuários, para permitir o acompanhamento do funcionamento dos algoritmos de Aprendizado de Máquina nesse contexto, seus benefícios, desafios e aspectos a serem melhorados. Este estudo permitiu uma análise do uso dessas práticas nos algoritmos dessa categoria para a geração de *insights* sobre o usuário, e na avaliação por meio de métricas dos modelos e das complexidades obtidas ao trabalhar com esse tema recente da literatura.

Este trabalho busca aprimorar as contribuições de Sergio et al. (2023) ao focar na melhoria das métricas para a avaliação de algoritmos de Aprendizado de Máquina e na criação de uma infraestrutura robusta para treinar, avaliar e gerenciar modelos com diferentes parâmetros e algoritmos. Além de selecionar as melhores execuções conforme o contexto, este estudo propõe uma arquitetura para o armazenamento e manutenção contínua dos modelos, alinhada com as práticas de MLOps. Essa infraestrutura visa

não apenas obter métricas quantitativas, mas também extrair informações relevantes dos modelos, mantendo-os atualizados com o fluxo constante de dados e selecionado a melhor combinação de parâmetros para os algoritmos. O foco são dados captados por sensores de relógios de pulso, monitorando momentos de estresse e ansiedade por meio das batidas cardíacas do usuário, expandindo e refinando o trabalho descrito.

## 1.6 Organização do Trabalho

Este trabalho está organizado da seguinte forma: o Capítulo 2 descreve os principais conceitos necessários para uma melhor compreensão do tema, além de apresentar definições da literatura relevantes para o estudo desse projeto. No Capítulo 3, são evidenciados e discutidos os trabalhos que servirão como base, as pesquisas exploradas são descritas, seus resultados analisados, e os aspectos positivos e negativos destacados, assim como exemplos de uso na prática. No final desse capítulo, há uma tabela que compara esses artigos em relação às suas contribuições. No Capítulo 4, é definida a metodologia utilizada e a arquitetura proposta neste trabalho para automatizar o processo de construção de modelos preditivos. O Capítulo 5 apresenta as conclusões obtidas e as possibilidades de pesquisas e trabalhos futuros.

# 2 Fundamentação Teórica

Neste capítulo, são apresentadas todas as definições e os conceitos necessários para uma melhor compreensão do tema e as abordagens da literatura exploradas para a escrita deste trabalho. Na Seção 2.1, são apresentados e discutidos todos os conceitos relacionados ao tema principal desta monografia, MLOps e as suas relações. A Seção 2.2 apresenta alguns temas da literatura relacionados à área da saúde, com ênfase na utilização de algoritmos de Aprendizado de Máquina, práticas de MLOps e suas motivações. A definição dos algoritmos que são utilizados nesta pesquisa são encontrados na Seção 2.3. Na Seção 2.4, são feitas as considerações finais do capítulo, reafirmando e resumindo os conceitos introdutórios que são necessários para a compreensão e contextualização da temática abordada neste trabalho.

## 2.1 Conceitos Relacionados a MLOps

Primeiramente, faz-se necessário destacar o conceito de Inteligência Artificial, sendo uma ramificação da Ciência da Computação capaz de desenvolver algoritmos que possam resolver problemas e realizar tarefas feitas por humanos (LONGO et al., 2020). A IA pode simular o comportamento humano, como o aprendizado, o raciocínio, a tomada de decisões, entre outros. Dentro desse contexto, existem técnicas e subáreas diferentes usadas nos modelos desses sistemas que podem ser destacadas. O Aprendizado Profundo (ALZUBAIDI et al., 2021), o Processamento de Linguagem Natural (CHOWDHARY, 2020), a Visão Computacional (COMPUTER. . . , 2021) e o Aprendizado de Máquina são algumas dessas áreas.

Aprendizado de Máquina (ML, do inglês *Machine Learning*) é uma ramificação da IA, que define o treinamento de seus modelos através do conjunto de dados de entrada. Com base nos dados fornecidos, os algoritmos são definidos para formar modelos e tomar decisões para o treinamento de máquinas. Dentro de ML existem três principais abordagens, o Aprendizado Supervisionado, o Aprendizado Não Supervisionado e o Aprendizado

por Reforço (CARBONELL; MICHALSKI; MITCHELL, 1983).

O aprendizado supervisionado envolve o treinamento de modelos de dados ao utilizar um conjunto de dados rotulados, que incluem pares de entrada e saída correspondentes. Durante esse processo, o modelo busca identificar padrões e relações nos dados de entrada que permitam mapear corretamente as entradas para suas respectivas saídas, de forma que, ao ser exposto a novos dados, o modelo possa prever com precisão as saídas esperadas. Por exemplo, considere um conjunto de dados de entrada contendo imagens das letras do alfabeto e suas respectivas identificações. Após o treinamento, quando o modelo receber uma nova imagem da letra "A", ele poderá identificar corretamente a letra, graças ao conhecimento adquirido durante o treinamento.

No aprendizado não supervisionado o modelo é treinado a partir de um conjunto de dados cujas saídas não são pré-definidas. Tem como objetivo descobrir padrões, classificar os dados, encontrar tendências e detectar discrepâncias. Os algoritmos de agrupamento de dados são os mais populares nesse tipo de aprendizado.

O aprendizado por reforço, conforme descrito por (CARBONELL; MICHALSKI; MITCHELL, 1983), envolve o treinamento de algoritmos para aprenderem a tomar decisões em um ambiente adaptativo mais independente por meio do recebimento de *feedbacks*. Através das recompensas recebidas pelas suas decisões, maximiza-se a função de tomada de decisões. Existem modelos desse tipo em alguns jogos, simulando as ações de um jogador. Os modelos de aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço são os tipos mais comuns em Aprendizado de Máquina.

Vale destacar também as práticas de Engenharia de *Software* (BOEHM, 1976), referências na construção de sistemas por muitas empresas e desenvolvedores no gerenciamento do ciclo de vida dos seus *softwares*. Como citado, o termo Engenharia de *Software* pode ser definido como um conjunto de práticas sistemáticas e planejadas para a construção e desenvolvimento de *softwares*. São ferramentas e princípios seguidos para construir sistemas de qualidade, escaláveis e seguros, principalmente com a utilização de metodologias ágeis. Assim, observa-se que o desenvolvimento de *software* envolve diversas etapas e abordagens para enfrentar os desafios dessa área. Algumas das etapas incluem modelagem, análise de requisitos, testes, implementação, manutenção, dentre ou-

tras. Compreender os objetivos das partes envolvidas, a documentação, a estrutura e a arquitetura do *software*, a modularização, o relacionamento entre os processos e suas dependências, a escalabilidade, bem como realizar testes para validação e confirmação do atendimento dos requisitos, são elementos essenciais. Essas são apenas algumas especificações da Engenharia de *Software*, que aplica técnicas da engenharia para garantir uma documentação adequada e um gerenciamento eficiente durante a criação de um sistema de qualidade que atenda às funcionalidades especificadas.

Por tudo isso, deve ser destacado também o conceito de DevOps (EBERT et al., 2016), a junção das palavras "Dev" e "Ops" refletem no desenvolvimento por meio de operações. Essa definição está relacionada a uma cultura organizacional estabelecida nas organizações e nos meios acadêmicos, mediante filosofias, práticas e ferramentas que colaboram para o desenvolvimento do *software* e a adoção de processos automatizados em seus diferentes ciclos de vida. Isso faz com que a empresa se consolide no mercado atendendo melhor a demanda dos seus clientes, promovendo entregas mais rápidas. As equipes envolvidas na construção do *software* devem trabalhar em união, aumentando a integração entre as mais diversas áreas necessárias para a construção de um sistema. Alguns benefícios do DevOps, como descrito na literatura, incluem a entrega rápida, as melhorias contínuas no produto e alcançar uma maior facilidade na correção de erros, operações em alta velocidade ao trazer soluções mais dinâmicas, a confiabilidade, trazendo inovações e atualizações constantemente, e também a segurança. Esse modelo DevOps pode ser implantado sem alterações nas configurações de segurança do sistema para a utilização das suas práticas.

Ao elucidar como o DevOps é importante na construção de *softwares* juntamente com as práticas da Engenharia de *Software*, tem-se que os algoritmos de IA também devem seguir um conjunto de práticas para o treinamento de seus modelos e na construção de soluções (LONGO et al., 2020). No entanto, essas práticas utilizadas nos sistemas tradicionais não podem ser simplesmente copiadas para sistemas automatizados, mas sim adaptadas conforme a arquitetura dos sistemas de Aprendizado de Máquina. Nesse contexto surge o conceito MLOps. No significado da palavra, "ML" equivale a *machine Learning* (Aprendizado de Máquina) e "Ops" equivale à palavra "operações". Esse tema

surgiu recentemente com a adesão de novos algoritmos de ML e a preocupação com as melhores técnicas e abordagens para a implantação de um modelo seguro, escalável e confiável dessa magnitude.

Portanto, pode-se definir MLOps como um conjunto de práticas que devem ser seguidas para um melhor desenvolvimento de sistemas que trabalham com modelos de Aprendizado de Máquina. Isso envolve a melhoria da eficiência, da confiabilidade, no aumento da produção e no monitoramento. No entanto, como modelos de Aprendizado de Máquina envolvem muitas fases de produção, esse ciclo dificulta uma melhoria contínua dos sistemas. Alguns benefícios do uso dessas práticas, além dos já citados acima, estão relacionados com a eficiência ao desenvolver modelos mais rápidos de ML com mais qualidade e uma implantação mais eficiente. A escalabilidade é alcançada por meio de implementações contínuas, geração de modelos controlados e supervisionados, além da redução de riscos, com maior transparência e respostas mais rápidas. A complexidade desses métodos reside em seus processos. De fato, um sistema que envolve exploração, análise e tratamento de dados, treinamento, ajustes, governança e inferência de modelos e serviços, implantação, monitoramento de modelos e retreinamento automatizado de modelos deve ser cuidadosamente ajustado e gerenciado.

Algumas pesquisas enfatizam as tendências e os desafios de MLOps, como em (ALLA; ADARI, 2020). De fato, esses tipos de *software* recebem uma entrada e produzem uma saída, assim como os *softwares* tradicionais. Todavia, as operações desse gerenciamento envolvem orquestração em nuvem, transporte e transformação dos dados, re-treinamento e reimplantação contínua, ou seja, um conjunto complexo de componentes para a geração de uma solução.

É importante ressaltar que, à medida que a complexidade das operações de IA aumenta, essas operações tornam-se menos sustentáveis (LONGO et al., 2020). Começam a surgir diversos componentes a serem gerenciados, uma arquitetura de software mais robusta, o manuseio de componentes autônomos pode se tornar cada vez maior, esses são alguns exemplos dessa complexidade. Um *software* deve ser sustentável e conseguir se manter operacional, no qual suas soluções devem melhorar continuamente. Alguns fatos que refletem o atual cenário da IA: a maioria das indústrias não sabe o que fazer com

seus dados, 75% dos cientistas de dados não são cientistas da computação e aproximadamente 90% das empresas estão com falta de engenheiro de dados (TAMBURRI, 2020). Em relação aos profissionais capacitados, isso demonstra que as empresas precisam de cientistas e engenheiros de dados mais qualificados, principalmente se tratando de um novo contexto: o Aprendizado de Máquina com práticas de MLOps .

## 2.2 Conceitos Relacionados a MLOps na saúde

A tecnologia vem promovendo diversos avanços na medicina e na saúde, é difícil imaginar cenários nos quais sistemas tecnológicos não estão presentes. Desde gerenciamento simples de sistemas até o controle e monitoramento de uma doença, através da análise de dados e extração de informações. O conceito *Active Assisted Living* (AAL)[1] pode ser compreendido como um programa europeu que financia a inovação tecnológica para manter os cidadãos conectados, saudáveis e ativos na velhice. Esse programa apoia o desenvolvimento de produtos e serviços que fazem a diferença na realidade das pessoas idosas ou portadoras de doenças crônicas. "A Inteligência Artificial, especialmente Aprendizado de Máquina, está contribuindo para otimizar muitos serviços, permitindo maior customização e maior automação. No AAL, IA é utilizada não só para adaptar as interfaces às necessidades dos utilizadores mas também para monitorizar o comportamento dos idosos e antecipar (através da análise de padrões) potenciais situações nocivas".

Tornou-se cada vez mais comum na área da medicina o acompanhamento dos pacientes por meio de um grande volume de dados sobre a saúde dele. Isso pode ser observado em artigos que exploram sistemas de Aprendizado de Máquina para analisar dados de pacientes. Pesquisadores estão monitorando e coletando informações sobre as atividades diárias de idosos, pacientes com doenças crônicas e pessoas com deficiência por meio de sensores instalados em diferentes cômodos da casa. O objetivo é reduzir a carga de trabalho de cuidadores e profissionais da saúde, propondo um *framework* para definir fluxo de eventos e processá-los para a identificação de padrões e mudanças nos ambientes assistidos pela tecnologia (AAL) em tempo real. Quando uma alteração significativa nas atividades de um indivíduo é detectada, essa informação deverá ser comunicada aos

[1] http://www.aal-europe.eu/

especialistas e familiares, alertando-os para uma possível revisão das condições do paciente (LARCHER et al., 2020).

Nesse contexto, surgem os desafios relacionados ao planejamento de uma infraestrutura para esses ambientes, utilizando computação em nuvem para armazenamento de dados e processamento, fornecendo desempenho, escalabilidade e processamento de dados (ALLI; ALAM, 2020). É fundamental evitar gargalos na rede, assegurar a segurança dos dados e do sistema e se adaptar às restrições de tempo e outros fatores.

## 2.3 Definição dos algoritmos de Aprendizado de Máquina utilizados

Nesta seção, são apresentadas descrições detalhadas dos algoritmos escolhidos para esta pesquisa, explicando suas definições, funcionamento, e como cada um se ajusta aos dados. O objetivo principal desta seção é fornecer uma base teórica sobre as particularidades desses modelos, demonstrando a eficácia para lidar com os desafios propostos no monitoramento da saúde e detecção de anomalias.

O AdaBoostRegressor é um algoritmo de grupo que cria um modelo mais robusto combinando vários modelos de regressão mais básicos. Ele aumenta o peso dos erros de todos os modelos, enfatizando os erros cometidos anteriormente. Isso é feito diversas vezes para aumentar a precisão geral, é útil quando se deseja aumentar a precisão de modelos simples, como árvores de decisões pequenas (FREUND; SCHAPIRE, 1997).

O Suport Vector Regressor (SVR) é utilizado para problemas de regressão, ele tenta encontrar um hiperplano que melhor se ajusta aos dados de uma forma que os erros menores sejam ignorados, isso evidencia um algoritmo que utiliza os pontos mais próximos sem dar importância para os *outliers*. A ideia é focar nos pontos próximos a margem para definir realmente o modelo, é super eficaz para cenários complexos, principalmente, quando os dados possuem uma estrutura não linear (CORTES; VAPNIK, 1995).

Já o algoritmo Decision Tree Regressor utiliza uma árvore de decisão para dividir os dados em perguntas binárias. Cada divisão ou nó faz uma escolha para reduzir a variância dentre os segmentos resultantes, diminuindo o erro de previsão. As divisões

vão ocorrendo até se encontrar a profundidade máxima da árvore ou até que os nós não possam ser mais divididos. Esse algoritmo é mais interessante para casos em que um modelo é mais fácil de interpretar, ideal para lógica de decisões sequenciais (KALYANE et al., 2024).

Gradient Boosting é um algoritmo de Aprendizado de Máquina que através de modelos fracos criados sequencialmente, se torna possível a criação de um modelo forte. Os novos modelos são ajustados para corrigir os erros cometidos pelos modelos anteriores, através do gradiente da função de perda. Esse algoritmo possui a capacidade de se ajustar aos dados através de interações complexas e isso o torna um algoritmo bastante utilizado na literatura (FRIEDMAN, 2001).

O algoritmo K-Nearest Neighbors (KNN) baseia suas previsões nas proximidades dos pontos de entrada com seus 'k-vizinhos' mais próximos. Ele calcula a distância, que é definida como um parâmetro desse algoritmo entre o ponto de entrada e todos os outros pontos dos conjuntos mais próximos para determinar a previsão. A sua maior utilidade é quando a relação entre os pontos locais é mais importante do que um ajuste global, além disso, é um algoritmo mais simples de se entender (COVER; HART, 1967).

O Random Forest, como o nome já diz, é um algoritmo de conjunto que cria várias árvores de decisão aleatórias e combina suas previsões para um melhor modelo, ou seja, uma única previsão mais assertiva. Cada árvore possui um subconjunto aleatório de dados para aumentar a diversidade e fugir da correlação entre as árvores. Ele se mostra relevante porque sua tendência é fugir do *overfitting*, que são modelos que se ajustam bem aos dados de treinamento, mas tendem gerar previsões ruins para novas amostras. Geralmente, a previsão desse modelo é obtida através da média das previsões de todas as árvores geradas (BREIMAN, 2001).

## 2.4 Considerações

Neste capítulo, foram discutidas algumas definições relacionadas ao tema de pesquisa deste trabalho de conclusão de curso. Pode-se ressaltar o crescente uso das tecnologias de IA para a resolução de problemas que antes eram resolvidos pelos humanos. O Aprendizado de Máquina é uma ramificação desse tipo de tecnologia que define seus modelos através dos

dados de entrada. Ou seja, conforme novos tipos de dados são recebidos pelos sistemas, os modelos são retreinados, e eles se ajustam de forma autoadaptativa. As organizações que produzem *software* prezam pela qualidade e eficácia no serviço prestado; portanto, essas entidades utilizam práticas da Engenharia de *software* para a construção de sistemas, prezando pela qualidade que atende à demanda do cliente.

Entretanto, sistemas complexos autônomos, como os que envolvem algoritmos de Aprendizado de Máquina, requerem práticas mais específicas para seu gerenciamento. Devido às características próprias dos modelos, principalmente por serem sistemas automatizados, surge assim o conceito de MLOps. Com o atual cenário tecnológico, diversas áreas da saúde estão trabalhando com *softwares*, e recentemente, surgiram alguns trabalhos relacionados ao Aprendizado de Máquina, através de sensores na área de saúde. Este trabalho planejou estudar esse cenário e avaliar sistemas através desses métodos de gerenciamento no contexto da área da saúde, utilizando modelos de Aprendizado de Máquina.

# 3 Análise da Literatura e Estudos Relacionados

Neste capítulo são abordados alguns trabalhos relacionados, evidenciando aspectos como a descrição do projeto, os resultados encontrados, os pontos positivos e negativos, e as possíveis contribuições relevantes para este estudo. Esses trabalhos servem também como embasamento teórico para essa pesquisa. Primeiramente, as pesquisas são abordadas, contextualizadas e descritas. Em seguida, são discutidos os resultados, as definições, as contribuições e suas limitações, com o intuito de explorar campos que foram abordados neste estudo.

Este capítulo está dividido em várias seções para detalhar e analisar o atual cenário da literatura de forma independente. Na Seção 3.1, são levantados os desafios e as tendências do MLOps. Na Seção 3.2, é discutida a contribuição do MLOps para o trabalho dos cientistas de dados. A Seção 3.3, explora uma estrutura para detectar eventos incomuns através de padrões, enquanto a Seção 3.4 apresenta uma arquitetura de *E-Healthcare* que envia notificações quando o paciente apresenta padrões incomum de saúde.

Na Seção 3.5 são discutidos os principais desafios e é feita uma revisão da literatura para o conceito de MLOps. A Seção 3.6, apresenta as definições e algumas arquiteturas propostas para o MLOps. A Seção 3.7, utiliza técnicas de GridSearchCV para aprimoramento dos modelos de Aprendizado de Máquina. Na Seção 3.8, métricas de avaliação são apresentadas para analisar esses modelos. Por fim, a Seção 3.9 oferece as considerações finais do capítulo.

## 3.1 Tendências e desafios de MLOps

Na pesquisa realizada, é mostrado que ao fazer uma busca sobre a palavra MLOps no *Google Trends*, fica fácil ver que este tema tem despertado o interesse de pesquisadores,

estudantes e curiosos. Isso ocorre porque existem diversas plataformas de Aprendizado de Máquina com modelos de alta qualidade e desempenho. Todavia, à medida que surgem recursos de inteligência artificial, surge a questão sobre até que ponto esse ecossistema possa ser sustentável. Este trabalho define os principais conceitos e desafios envolvidos com a sustentabilidade de software de IA e os recursos de MLOps (TAMBURRI, 2020).

Dos resultados encontrados, é importante ressaltar as principais definições e desafios abordados pelo autor. Sustentabilidade, nesse contexto, é tratada como a capacidade do software de IA existir e operar continuamente, mantendo-se operacional, na qual suas decisões devem ser sempre melhoradas, assim como a sua solução. Segundo este trabalho, as pesquisas de MLOps ainda são iniciais e estão mais focadas na definição de conceitos e processos. Os principais desafios no cenário de IA são: a maioria das empresas não sabe o que fazer com os seus dados; muitos dos cientistas de dados não são cientistas da computação; ocorre uma falta de engenheiros de dados, e os que existem, já estão sobrecarregados com as suas tarefas.

Vale destacar os pontos positivos: esse trabalho explica sobre o funcionamento de softwares envolvendo Aprendizado de Máquina, e analisa detalhadamente as operações de MLOps, como orquestração em nuvem, transporte e transformação de dados, retreinamento contínuo do modelo, reimplantação contínua, produção e apresentação, ou seja, um conjunto complexo de componentes para gerar soluções baseadas no gerenciamento de sistemas de Aprendizado de Máquina. Além disso, as operações de MLOps são interessantes para a obtenção de métricas, já que essas tarefas podem começar a ser quantificadas. Este trabalho, entretanto, poderia explorar melhor o funcionamento dessas operações, na prática, para ilustrar esses conceitos no funcionamento de um software.

Em resumo, este artigo apresenta uma visão geral sobre as tendências e os maiores desafios associados ao contexto do software relacionado ao Aprendizado de Máquina, principalmente em relação à sustentabilidade dessa categoria de sistemas. Fazer previsões e tomar decisões são alguns exemplos da capacidade dos softwares desse tipo. Este artigo se preocupou em explorar o atual cenário de pesquisas nessa área e forneceu uma visão panorâmica dos últimos avanços em MLOps. O autor evidenciou também os desafios enfrentados pelas indústrias, já que muitas vezes não sabem como lidar com seus dados, além

de existirem poucos profissionais especializados e acostumados com essa nova categoria de software.

## 3.2 Como as práticas MLOps podem ajudar os cientistas de dados?

Em Makinen et al. (2021) é proposto um estudo sobre a importância de MLOps no cenário das atividades diárias dos cientistas de dados. Foram utilizadas diversas pesquisas em diferentes países para organizar dados e retirar informações sobre esse contexto. Devido às diversas práticas de engenharia de software na produção dos sistemas tradicionais, há um interesse crescente na implantação de práticas semelhantes para modelos de Aprendizado de Máquina. Após a coleta dessas informações, foi possível perceber que o maior problema dos cientistas de dados gira em torno dos próprios dados, principalmente relacionado à implantação de modelos para a produção. Como hipótese dessa pesquisa, foram levantados três tipos de organizações: i) empresas que estão descobrindo formas de uso de seus dados; ii) as organizações focadas em construir modelos para produção; iii) as que já estão com modelos prontos e focam na melhoria e reimplantação contínua. A maioria das organizações está classificada nos tipos i) e ii).

Esta pesquisa se preocupa em detalhar melhor o funcionamento de um sistema de Aprendizado de Máquina, definindo que os módulos devem possuir dados disponíveis para o treinamento, modelos devem ser treinados com esses dados e, se o modelo for validado, ele está pronto para implantação. Após implantado, deve ser monitorado. Destacou também o CD4ML, sendo práticas para automatizar o ciclo de vida de aplicativos de Aprendizado de Máquina de ponta a ponta. Foi feita uma coleta para captar dados por meio de profissionais que tivessem cargos relacionados a essa área. O autor define que as organizações que já estão na fase centrada em modelos podem transformar os procedimentos em automação mediante *pipelines*, com o uso de MLOps.

Este trabalho focou em definir melhor os modelos de Aprendizado de Máquina, demonstrar sua estrutura de forma teórica, pesquisar sobre as atividades diárias dos cientistas de dados e identificar os principais usos de Inteligência Artificial. Separar e estudar

níveis de maturidade das organizações conforme a utilização de seus dados e apresentar uma ferramenta de gerenciamento, o CD4ML, esses foram os principais pontos positivos. Todavia, o uso dessa ferramenta na prática ilustrada em um caso de uso poderia ser muito eficaz na proposta do artigo, o que pôde ser identificado como um aspecto negativo.

Este artigo apresenta algumas pesquisas para obter informações sobre o atual cenário dos dados nas organizações, como a maioria das empresas utiliza essas informações e a importância de MLOps. Da mesma forma que vem se destacando o uso de *DevOps* nos *softwares* tradicionais, surge a importância da sustentabilidade dos *softwares* de Aprendizado de Máquina e uma produção contínua, já que empresas usam sistemas dessa categoria em produção. No entanto, não basta aplicar as ferramentas de *DevOps* para arquiteturas desse tipo, por existirem diversas características que os diferem dos *softwares* convencionais. Além disso, a complexidade aumenta à medida que o sistema cresce, o número de modelos cresce também, surgem novos conjuntos de dados, entre outros fatores.

## 3.3 Estrutura para detectar padrões incomuns em ambientes AAL

Esta pesquisa explora, na prática, o uso de sensores através da coleta de dados para acompanhar pacientes idosos ou pessoas com doenças crônicas em seus domicílios, com o intuito de reduzir a carga dos cuidadores de idosos e profissionais da saúde (LARCHER et al., 2020). Existem pesquisas e organizações relacionadas ao envelhecimento saudável da população, como o AAL Programme[2], que tem o objetivo de financiar estudos para a melhoria da vida da população idosa através da tecnologia. Esta pesquisa foca no monitoramento do ambiente através de sensores não vestíveis que coletam uma grande quantidade de dados, utilizando aprendizagem de máquina para detectar padrões incomuns. Isso foi feito por meio de um estudo de caso em um ambiente real e os resultados mostram a viabilidade da proposta.

Nos resultados apresentados, vale destacar que os modelos e as abordagens propostos funcionaram para identificar os padrões e as atividades incomuns das pessoas. No

[2]http://www.aal-europe.eu/

entanto, é necessário possuir mais informações para avaliar com mais cautela os *insights* gerados. Ao utilizar três casas de pacientes para o processamento dos dados, a arquitetura proposta mostrou-se eficiente. Ou seja, este estudo mostrou-se eficiente para apresentar informações relevantes aos familiares e profissionais da saúde sobre seus pacientes, mas necessita de mais dados para tornar as informações mais claras para os usuários finais.

O destaque deste trabalho é fornecer uma abordagem prática no monitoramento de idosos em seus lares e identificar padrões incomuns em suas atividades. Apresenta diversas arquiteturas para o Aprendizado de Máquina, o uso de sensores e como são organizados esses dados extraídos, além de abordar a arquitetura *Fog-cloud*. Todavia, essas arquiteturas utilizadas não seguem práticas de engenharia de software, principalmente por não se tratar de um sistema tradicional, mas de um sistema adaptativo, o que faz com que não seja possível um acompanhamento fiel das métricas dos seus modelos. Essa abordagem também se limita a um estudo de caso na residência dos pacientes. Caso eles estejam em outros ambientes, esses dados vitais não seriam acompanhados.

Considerando todos os aspectos, essa pesquisa superou os desafios de usar sensores para o monitoramento de idosos, fornecendo uma alternativa ao colocar sensores não vestíveis nos cômodos da casa. Além disso, foram utilizadas arquiteturas interessantes, como a arquitetura Lambda para permitir processamento em tempo real e em lote, o paradigma *Fog-Cloud*, reduzindo o número de acessos à nuvem, e experimentos usando vários modelos para identificar qual algoritmo se adaptaria melhor aos dados. Como trabalhos futuros, propõem consolidar os padrões detectados e apresentá-los de forma mais objetiva aos usuários finais, juntamente com os profissionais de saúde. Pretendem, também, enviar mensagens automáticas (e-mail, SMS) caso seja detectado um padrão incomum do paciente.

## 3.4 Uma Arquitetura de *E-Healthcare* para gerar notificações sobre a saúde de um paciente

É proposta uma arquitetura capaz de enviar dados aos usuários sobre os seus sinais vitais. A arquitetura é baseada em *Fog-Cloud* e algoritmos de Aprendizado de Máquina

para coletar dados de frequência cardíaca do paciente, processar e gerar notificações caso seja detectado um padrão incomum. A abordagem foi testada na prática, coletando dados de pacientes por três meses e mostrando aos usuários em quais momentos do dia seus batimentos cardíacos apresentavam maior instabilidade, indicando, assim, momentos viáveis para alertar os pacientes por notificações (SERGIO et al., 2023).

Na geração dos resultados, foram considerados diversos fatores devido à complexidade e ao escopo no tratamento dos dados, considerando diferentes cenários e, por se tratar de sinais vitais, deve haver uma rigorosa interpretação dos dados. Foram utilizados relógios inteligentes, através de seus sensores, para a captura dos dados. O modelo monitora um paciente por três meses, no qual a maioria das amostras gira em torno dos 70 bpm. Ocorre uma variação na faixa entre 60 e 80 bpm e uma baixa variabilidade entre os valores 100 a 120 bpm. Essas medições podem indicar se o usuário tem problemas cardíacos ou até mesmo identificar possíveis primórdios de estresse físico ou mental. Após esses momentos serem registrados, as notificações são agendadas para serem enviadas ao usuário nesses horários do dia.

Este estudo serve como um bom parâmetro na captura de dados de sensores dos relógios inteligentes, no armazenamento, na limpeza e no tratamento desses dados, além da construção de modelos automatizados de Aprendizado de Máquina. Entretanto, o único sinal vital estudado dos usuários foi o monitoramento das frequências cardíacas. Além desse, poderiam ser analisados outros dados vitais também, como temperatura do corpo e pressão arterial, oferecendo informações mais completas sobre o estado de saúde do paciente, o que eles propõem como trabalho futuro. Outrossim, o autor utiliza somente um algoritmo específico de Aprendizado de Máquina, seria interessante uma automatização na escolha do modelo conforme o melhor cenário.

Por tudo isso, este projeto visa propor uma arquitetura de *software* para monitoramento dos momentos estressantes dos usuários. Com os resultados obtidos, conclui-se que a arquitetura proposta foi viável para identificar momentos incomuns na frequência cardíaca. Existe a intenção de trabalhar futuramente com dados mais precisos e uma arquitetura mais robusta na coleta dos dados e no processamento, além de verificar como a estrutura lida com inúmeros usuários. Há interesse em trabalhar também com experi-

mentos na área de identificação de estresse e análise de outros dados vitais.

## 3.5 Revisão da Literatura sobre os desafios do *pipeline* de MLOps

Em Siddappa (2022), é realizado um estudo do estado da arte ao revisar a literatura, fornecendo uma visão geral dos desafios, soluções e ferramentas de MLOps utilizadas. A adoção de práticas da Engenharia de Software para sistemas é conhecida como *DevOps*, enquanto técnicas e práticas para a produção de algoritmos de Aprendizado de Máquina são chamadas de MLOps. Este artigo define esse conceito como componentes de *pipelines*: *pipeline* de manipulação de dados, *pipeline* de criação de modelo e *pipeline* de implantação. Foi realizada uma revisão da literatura para identificar os desafios desses *pipelines* e fornecer uma síntese com orientações práticas aos profissionais, inclusive com ferramentas e suportes. Além disso, demonstram como alguns desafios podem ser superados na prática utilizando as ferramentas propostas e seguindo as diretrizes mencionadas neste estudo.

Alguns resultados podem ser destacados nesse projeto, já que existem alguns desafios para os quais não há soluções, como operações e *loops* de *feedback*, explicabilidade, acoplamento entre os requisitos do sistema e do modelo, e múltiplas ferramentas, que são alguns exemplos de desafios sem solução. Por fim, os desafios MLOps são categorizados em quatro grupos, ou seja, desafios de gerenciamento de dados, desafios na criação dos modelos, desafios de implantação de modelos e desafios transversais.

Este trabalho foi muito eficiente em demonstrar os desafios do uso dos *pipelines* de MLOps, apresentando soluções e ferramentas para solucionar diversos problemas no contexto de Aprendizado de Máquina. Vale destacar a preocupação do autor em utilizar as soluções propostas no trabalho para solucionar desafios, mostrando as ferramentas na prática. Como aspectos negativos, vale destacar que o autor não trabalhou com um problema envolvendo Aprendizado de Máquina sem o uso de técnicas MLOps, nem comparou a resolução desse mesmo problema envolvendo essas técnicas a fim de verificar a viabilidade do uso desses *pipelines*.

Esta pesquisa expõe os desafios na construção dos *pipelines* para o gerenciamento.

Primeiramente, foram identificados os principais desafios relacionados a cada *pipeline*. Algumas diretrizes foram demonstradas metricamente para uma avaliação quantitativa usando exemplos na prática. Como trabalhos futuros, destacaram o *pipeline* de dados para ser trabalhado com diferentes formatos de dados, como dados em lote e dados em *streaming*, já que diferentes formatos de dados geram diversos desafios. Como os sistemas de Aprendizado de Máquina não são explicitamente programados, mas se desenvolvem aprendendo com os dados, isso dificulta a implantação, operação e manutenção dos modelos, principalmente ao trazer dados de diferentes tipos.

## 3.6 MLOps: Revisão, definição e arquitetura

Este artigo fornece uma visão mais teórica, porém geral. O intuito é apresentar definições e arquiteturas das operações de Aprendizado de Máquina (MLOps) mais utilizadas atualmente. Desenvolver um projeto de Aprendizado de Máquina e colocá-lo em produção é altamente desafiador, e muitos sistemas falham em atender às expectativas. MLOps surge com o intuito de resolver este problema, todavia, ainda é um termo recente e vago (KREUZBERGER; KüHL; HIRSCHL, 2023). Essa lacuna foi preenchida com pesquisas, revisão de ferramentas disponíveis e entrevistas com especialistas. Essas investigações contribuíram para a obtenção de conhecimento sobre arquiteturas, fluxos de trabalho, componentes e funções associados. Portanto, este trabalho fornece uma base sólida para pesquisadores e profissionais de Aprendizado de Máquina que desejam automatizar suas operações com a tecnologia.

Foi feita uma revisão detalhada dos princípios mais relevantes, além da definição de diversos termos relacionados a MLOps. Princípios em Aprendizado de Máquina, podem ser relacionados com melhores práticas, e os mais comuns definidos pelo autor são a automação, o fluxo de trabalho, a reprodutibilidade, as versões, a rastreabilidade, a colaboração, o treinamento e a avaliação contínua, o rastreamento/registro de metadados, o monitoramento contínuo e *loops* de *feedback*. Além desses principais princípios definidos pelo trabalho, alguns componentes também devem ser implementados no projeto de Aprendizado de Máquina para seguir essas operações.

Os principais aspectos positivos podem ser destacados como a revisão literária

feita neste estudo. Este artigo define uma tabela com uma lista de tecnologias avaliadas que podem ser utilizadas e como elas podem ajudar. Fornece uma base muito sólida para quem está começando a estudar ou até mesmo para profissionais da área. Um aspecto negativo foi a falta de uso dessas tecnologias na prática, no formato de tutorial para os profissionais ou acadêmicos.

Pelo aumento da quantidade de dados e pelas necessidades dos sistemas se autoadaptarem ao meio, mais produtos de Aprendizado de Máquina estão sendo desenvolvidos. O ambiente acadêmico e as organizações têm-se preocupado e dado mais importância ao desenvolvimento e à construção de modelos, esquecendo das operações complexas que esses algoritmos devem seguir para atender às demandas reais. Muitos fluxos de trabalho ainda são gerenciados manualmente, e é nesse contexto que surge o paradigma MLOps para abordar esses desafios. Através do estudo das ferramentas disponíveis na literatura e das entrevistas realizadas com especialistas na área, este trabalho define conceitos para ajudar pesquisadores e estudantes desse campo.

## 3.7 Uso de técnicas de Aprendizado de Máquina com GridSearchCV para diagnóstico médico

A pesquisa realizada em Ahmad et al. (2022), aborda a previsão de doenças cardíacas, um dos maiores desafios do campo da medicina. Este estudo utiliza alguns algoritmos de Aprendizado de Máquina como Regressão Logística (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM) e Gradient Boosting Classifier (GBC), além da aplicação de técnicas de validação cruzada GridSearchCV para prever essas doenças. Os resultados foram comparados com estudos anteriores, verificando-se que o Extreme Gradient Boosting Classifier com GridSearchCV produz a melhor combinação de parâmetros para a acurácia.

Como pontos positivos desse trabalho destaca-se o uso de diversos algoritmos de Aprendizado de Máquina e a aplicação de técnicas de validação cruzada para garantir a robustez dos modelos. A utilização de *datasets* diversificados também fortalece a generalização dos resultados. No entanto, um aspecto negativo é a limitação dos sinais vitais

analisados, focando apenas na frequência cardíaca, enquanto outros sinais vitais poderiam proporcionar uma visão mais abrangente do estado de saúde dos pacientes. Além disso, a aplicação prática dos resultados poderia ser explorada em maior profundidade para verificar a eficácia dos modelos em um ambiente real.

Este artigo representa uma ótima contribuição para a previsão de doenças cardíacas, demonstrando a eficácia dos algoritmos de Aprendizado de Máquina com o GridSearchCV. Com a intenção de desenvolver técnicas para solucionar problemas da realidade, no futuro, os autores pretendem aprimorar o modelo para ser utilizado com vários algoritmos de seleção de características e explorar outras possibilidades, como a utilização do GridSearchCV com diferentes classificadores de *Boosting*.

## 3.8 Métricas de avaliação para o Aprendizado de Máquina

O estudo realizado por Handelman et al. (2019) aborda como Aprendizado de Máquina e a Inteligência Artificial (IA) estão sendo cada vez mais utilizados na medicina, o que gera preocupações sobre a necessidade de que os profissionais da saúde compreendam melhor essas tecnologias e saibam avaliar os estudos e as ferramentas relacionados. Este trabalho resume as métricas utilizadas em Aprendizado de Máquina, explicando seus significados e alguns dos termos técnicos utilizados na área.

Os resultados desse estudo destacam a importância do conhecimento técnico para a avaliação das métricas para modelos dessa categoria. Um dos pontos positivos de destaque é a abordagem educativa, fornecendo uma introdução clara e compreensível às métricas de avaliação. Isso inclui explicações sobre precisão, erro em problemas de regressão e o coeficiente de determinação ($R^2$), o que é essencial para discernir a eficácia dos modelos preditivos apresentados. Entretanto, podem ser observados alguns aspectos negativos como a ausência de casos reais e práticos de aplicações, o que ajudaria a ilustrar melhor os conceitos explorados.

No geral, este artigo destaca a importância da transparência dos dados envolvendo algoritmos de Aprendizado de Máquina na medicina para que os profissionais da saúde

avaliem a eficácia dos modelos preditivos. Além disso, fornece uma base sólida para pesquisadores e médicos entenderem sobre essas aplicações e a sua adoção no cenário da saúde.

## 3.9 Considerações finais do capítulo

Neste capítulo, foram abordados diversos trabalhos relacionados ao tema, com o objetivo de fornecer uma base teórica mais consolidada para a pesquisa. Alguns trabalhos foram explorados conforme a sua descrição, resultados obtidos, aspectos positivos e negativos, e as contribuições relevantes de cada tema. Após o estudo dessas contribuições científicas, pode-se observar o atual cenário do uso de MLOps e os principais desafios associados, especialmente por ser um tema recente na literatura. A seguir, a Tabela 3.1 compara os trabalhos estudados, explorando os principais aspectos.

Tabela 3.1: Comparação dos trabalhos relacionados

| Trabalhos | Tipo | Aspectos Positivos | Aspectos Negativos |
|---|---|---|---|
| (TAMBURRI, 2020) | Teórico | Definições e desafios de MLOps | Falta de prática |
| (MAKINEN et al., 2021) | Teórico | ML nas organizações | Sem exemplo |
| (LARCHER et al., 2020) | Teórico e Prático | Exemplo real | Sem uso de métricas |
| (SERGIO et al., 2023) | Prático | ML na saúde | Poucas análises |
| (SIDDAPPA, 2022) | Teórico e Prático | Desafios e ferramentas | Não verifica viabilidade |
| (KREUZBERGER; KüHL; HIRSCHL, 2023) | Teórico | Revisão geral MLOps | Sem caso de uso |
| (AHMAD et al., 2022) | Teórico e Prático | ML e validação cruzada | Não aplica em casos reais |
| (HANDELMAN et al., 2019) | Teórico | Métricas de ML | Falta de prática |

# 4 Materiais e Métodos

Neste capítulo, é apresentada a arquitetura proposta para o processamento automático dos dados durante a fase de treinamento dos modelos preditivos. A infraestrutura recebe os dados vitais de um indivíduo, realiza o treinamento utilizando diferentes modelos de Aprendizado de Máquina, seleciona as melhores combinações de parâmetros, processa suas métricas e armazena os seus resultados para, posteriormente, gerar o melhor modelo preditivo com base nos dados do usuário. Esta pesquisa explora o treinamento dos modelos propostos em (SERGIO et al., 2023), avançando a pesquisa através do uso de práticas recomendadas em MLOps para automatizar o processo de atualização dos modelos, considerando o fluxo constante de novos dados.

Este capítulo está organizado da seguinte forma: na Seção 4.1, é apresentada a arquitetura proposta nesta pesquisa, detalhando a automação aplicada na construção dos modelos preditivos. A Seção 4.2 aborda a avaliação da solução, com uma análise métrica dos resultados para verificar a viabilidade do trabalho, além de descrever o ambiente de experimentação e os parâmetros utilizados para a combinação dos algoritmos. Por fim, na Seção 4.3, são discutidas as limitações encontradas durante o desenvolvimento e execução dos experimentos.

## 4.1 Arquitetura proposta

Este trabalho visa aprimorar a implementação de um sistema de monitoramento da saúde dos indivíduos, a fim de entender o seu comportamento e detectar possíveis anomalias. Nesta pesquisa, foram desenvolvidos mecanismos de processamento e distribuição dos dados vitais para a construção de modelos preditivos. A arquitetura proposta neste trabalho busca aprimorar a camada *Fog* da pesquisa apresentada no trabalho citado. Nesta camada ocorre o processamento dos dados e, com base nessa arquitetura, foi proposta uma extensão que integra as métricas de MLOps, validações cruzadas, combinação de diferentes parâmetros e uma gama de algoritmos de Aprendizado de Máquina para fornecer diversos

modelos preditivos. Essa abordagem permite um processo mais robusto e complexo para a seleção dos melhores modelos.

Além disso, essa proposta traz maior automação, auxilia na reprodutibilidade e proporciona maior escalabilidade, pois, com diferentes algoritmos e diversas combinações de parâmetros, os modelos tendem a se adaptar melhor a diferentes cenários. Este trabalho visa a criação de sistemas de Aprendizado de Máquina mais robustos e adaptáveis, capazes de lidar com a variação dos dados.

Na Figura 4.1, pode-se observar a sequência de passos necessários até a seleção do melhor modelo preditivo. Em síntese, a arquitetura recebe os dados para treinamento e validação de modelos. Diversos modelos são gerados devido à alta capacidade de combinação de parâmetros. Com base na extração das métricas desses modelos e na seleção dos melhores algoritmos, essa arquitetura consegue selecionar o melhor modelo entre todos para o cenário.

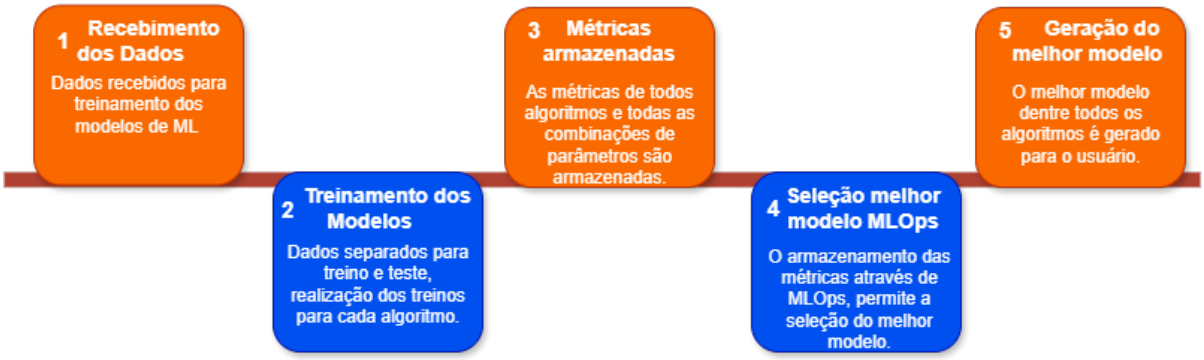


Figura 4.1: Sequência da arquitetura.

### 4.1.1 Automação do processo de construção dos modelos preditivos

Esta seção apresenta a solução proposta neste trabalho para automatizar a camada de processamento da arquitetura proposta por (SERGIO et al., 2023). Essa camada processa os dados recebidos e gera um modelo de Aprendizado de Máquina específico para cada usuário, conforme o comportamento dos seus sinais vitais.Este trabalho se preocupou em aprimorar a camada de processamento desenvolvida por Sergio et al. (2023), que utilizou o algoritmo K-Nearest-Neighbors (KNN) para gerar modelos preditivos. Esta

pesquisa propõe uma arquitetura aprimorada para otimizar o processamento dos modelos através dos conceitos e práticas de MLOps. Essa arquitetura é capaz de receber dados dos usuários, gerar diversos modelos de Aprendizado de Máquina e aplicar validação cruzada para automatizar a seleção das melhores combinações de parâmetros para cada algoritmo. Após isso, o melhor modelo é selecionado com base em seu desempenho através das métricas rastreadas. Essa abordagem concentra-se principalmente na melhoria da camada de processamento, garantindo maior eficiência na seleção dos modelos, a visão da arquitetura geral pode ser encontrada na 4.2.

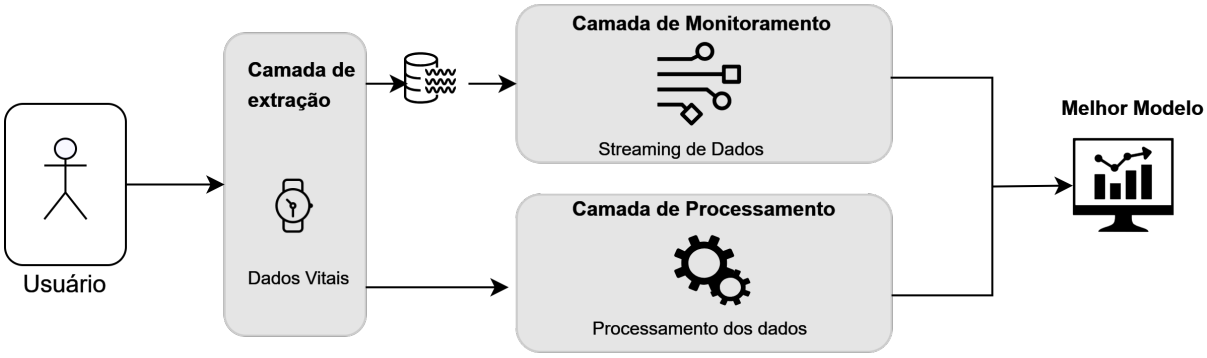


Figura 4.2: Visão geral da arquitetura

A primeira etapa da arquitetura consiste no recebimento dos dados, onde ocorre a coleta dos sinais vitais através de dispositivos que se comunicam entre si na camada de extração. A camada de Monitoramento é responsável por receber esses dados e fornecer respostas rápidas. Essa camada é capaz de capturar os sinais vitais recebidos dos usuários e utilizar os modelos de comportamentos previamente treinados pela camada de Processamento para gerar uma resposta instantânea. Já na camada de processamento, um conjunto com os dados vitais de um determinado usuário é recebido, e o pré-processamento desses dados é feito manualmente, como destacado em (SERGIO et al., 2023), utilizando estratégias de limpeza e detecção de *outliers*, isso pode ser modificado ao acrescentar novas técnicas de limpeza de dados na solução proposta. Posteriormente, ainda na camada de processamento, onde este trabalho realizou a maior automação, diversos algoritmos de Aprendizado de Máquina são adotados para a construção de modelos preditivos. Os algoritmos utilizados neste estudo foram: KNN (K-Nearest Neighbors), SVR (Support Vector Regressor), AdaBoostRegressor, DecisionTreeRegressor, GradientBoostingRegres-

sor e RandomForestRegressor. Esses modelos foram selecionados devido à sua adequação para problemas de regressão e ao seu baixo custo computacional na fase de treinamento. Dadas as particularidades do problema abordado neste trabalho, foi necessária a seleção de modelos que fossem treinados rapidamente, pois o objetivo é a identificação de momentos de estresse e ansiedade de um indivíduo, possibilitando uma rápida tomada de decisão. A arquitetura desenvolvida também se preocupa em permitir a inserção de novos algoritmos de Aprendizado de Máquina posteriormente de forma simples.

Durante o treinamento desses modelos, foi utilizado o GridSearchCV (AHMAD et al., 2022) para ajustar os hiper parâmetros, a fim de maximizar sua capacidade de predição. Esta é uma técnica de busca exaustiva que testa todas as combinações possíveis de um conjunto especificado de parâmetros. Um conjunto de parâmetros foi definido para cada algoritmo, e esta técnica selecionou a melhor combinação para o *dataset* do paciente. Este método realiza uma validação cruzada para cada combinação, garantindo que o modelo selecionado seja o mais eficiente com base nos dados fornecidos.

Após o treinamento dos modelos, ocorre a fase de avaliação das métricas específicas. Neste trabalho, foi selecionado o coeficiente de determinação para avaliar os modelos e a qualidade das predições, essa métrica é interessante para identificar se um modelo é adequado aos dados. Com a utilização das práticas de MLOps, é possível monitorar e analisar as métricas de desempenho real durante a validação cruzada. O modelo escolhido como melhor é o que possui o maior valor dessa métrica $R^2$. A métrica $R^2$ é uma forma de avaliação de modelos de regressão que indica o quão bem os dados são representados pelo modelo, baseado na proporção da variação dos resultados. Assim, a métrica quantifica a proporção da variabilidade dos dados, fornecendo uma visão sobre a qualidade das previsões feitas. A métrica $R^2$ está representada na Equação 4.1.

$$R^2 = 1 - \frac{SS_r}{SS_t} \tag{4.1}$$

Onde:

- $SS_r$ é a soma dos quadrados dos resíduos, calculada como:

$$SS_r = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{4.2}$$

- $y_i$ são os valores observados (verdadeiros). - $\hat{y}_i$ são os valores preditos pelo modelo. - $SS_t$ é a soma dos quadrados totais, calculada como:

$$SS_t = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{4.3}$$

- $\bar{y}$ é a média dos valores observados.

Na Figura 4.3, pode-se observar o detalhamento da camada de processamento da arquitetura, que inclui o recebimento dos dados já estruturados para o treinamento dos modelos. Em seguida, ocorre a recepção desses dados por cada algoritmo de Aprendizado de Máquina. Após isso, o treinamento é realizado utilizando diversas combinações de parâmetros, e o melhor modelo com a melhor combinação é selecionado para cada algoritmo. Por último, ocorre a validação comparativa através da métrica $R^2$ para a definição do modelo que melhor se ajustou aos dados do indivíduo em questão.

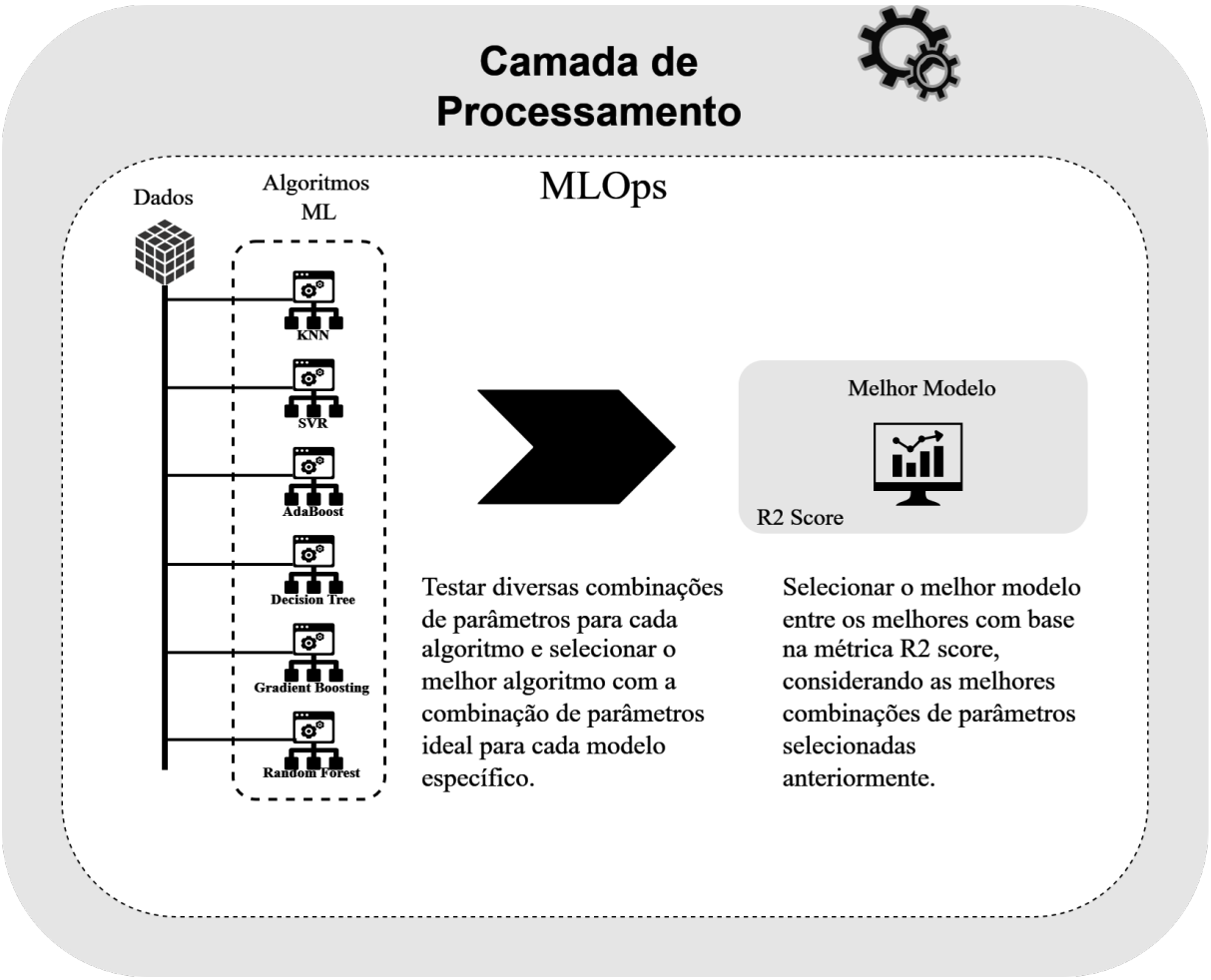


Figura 4.3: Camada de processamento

## 4.2 Avaliação da Solução

A avaliação da solução busca verificar se a aplicação de MLOps para monitoramento da saúde e detecção de anomalias influencia positivamente na tomada de decisão, permitindo a seleção de um modelo mais adequado aos dados de cada usuário. Esta seção apresenta os resultados obtidos a partir da arquitetura proposta anteriormente comparando as métricas para avaliar a viabilidade da proposta sugerida. Ao final, as limitações são discutidas.

Esta seção está organizada em subseções: na Subseção 4.2.1 é detalhado o ambiente de experimentação, incluindo configuração da máquina utilizada, as bibliotecas e a linguagem de programação. Já na Subseção 4.2.2, são definidos os algoritmos e os parâmetros selecionados para a realização das validações cruzadas, proporcionando uma análise criteriosa dos resultados.

### 4.2.1 Ambiente de experimentação

Para a realização desta pesquisa, foi necessária a utilização de algumas tecnologias e a criação de um ambiente de configuração que permitisse a geração de diversos modelos de Aprendizado de Máquina. A linguagem utilizada foi Python, para a configuração de um ambiente devido à sua versatilidade e à abundância de bibliotecas disponíveis para o Aprendizado de Máquina e a manipulação de dados. Foi utilizado *scikit-learn* [3] para a implementação de algoritmos de Aprendizagem de Máquina; a biblioteca *pandas* [4] foi adotada para manipulação e tratamento de dados; e o *matplotlib* [5] foi usado para a geração de gráficos que auxiliaram na interpretação das informações e análise dos resultados.

A configuração da máquina utilizada para os experimentos incluiu um processador Intel Core i5-1035G1, 8 GB de RAM, e a criação de um ambiente virtual através da ferramenta *virtualenv* para garantir a compatibilidade de pacotes e resolver problemas de dependência no sistema operacional Ubuntu 22.04.2 LTS. Esse ambiente permitiu uma configuração robusta para executar os experimentos eficientemente. Além disso, ambientes de desenvolvimento foram utilizados para auxiliar na programação; o Jupyter Notebook 7 foi essencial na geração dos resultados e visualização das métricas obtidas.

[3] https://scikit-learn.org/stable/
[4] https://pandas.pydata.org/docs/
[5] https://matplotlib.org/

### 4.2.2 Definição dos parâmetros de cada algoritmo

Como especificado anteriormente, esta pesquisa propôs uma diversificação e otimização de diversos algoritmos de Aprendizado de Máquina. No estudo selecionado para complementação, apenas o algoritmo K-Nearest Neighbors (KNN) foi empregado (SERGIO et al., 2023). Nesta pesquisa, foram testados e comparados os seguintes algoritmos: AdaBoostRegressor, Support Vector Regressor (SVR), Decision Tree, Gradient Boosting, K-Nearest Neighbors (KNN) e Random Forest.

Nas tabelas abaixo, são apresentados os parâmetros utilizados por cada algoritmo e os valores passados para a realização da validação cruzada. Os valores adotados são baseados nos padrões definidos para esses algoritmos como sendo a faixa de valores ideais.

Para o algoritmo KNN, foram utilizados os parâmetros abaixo, com uma síntese da faixa de valores adotados para cada um deles apresentada na Tabela 4.1:

- **n_neighbors**: os números de vizinhos que são considerados para fazer a previsão, quanto menor o valor, o modelo consegue capturar mais ruídos nos dados, caso contrário, valores maiores podem suavizar a previsão. Foram utilizados os seguintes valores: [2,6].

- **weights**: esse parâmetro define uma função peso para ser utilizada, determina como os vizinhos contribuem na previsão do valor. Foram utilizados as seguintes funções: [“uniform”, “distance”].

- **algorithm**: aqui é definido o algoritmo, que de fato será utilizado para calcular os vizinhos mais próximos. Foram utilizados os seguintes valores: [“ball_tree”, “kd_tree”, “brute”]. O “ball_tree” é mais eficaz para dados de alta dimensionalidade e que não são uniformemente distribuídos ao usar esferas para dividir os espaços de dados em grupos. O “kd_tree” divide os espaços para cada nível da árvore, sendo eficiente para baixas dimensões e ineficiente para altas dimensões. Já o “brute” é lento para grande conjunto de dados, pois calcula a distância entre todos os pontos.

- **leaf_size**: esse é definido como o tamanho da folha da árvore passada para os algoritmos “ball_tree” e “kd_tree”, isso afeta a velocidade da construção e eficiência na busca através da árvore. Foram utilizados os seguintes valores: [20,30,40].

- **p**: parâmetro "p" é definido como a métrica de distância utilizada para encontrar vizinhos mais próximos. Quando temos, "p" = 1, a referência utilizada é a distância de Manhattan, que calcula a soma das distâncias absolutas entre os pontos, para "p" = 2, a distância Euclidiana é utilizada, responsável por calcular a raiz quadrada da soma dos quadrados da diferença entre os pontos. Foram utilizados as seguintes distâncias "p" : [1,2].

Tabela 4.1: KNeighborsRegressor - Parâmetros

| **Parâmetro** | **Valor** |
|---|---|
| Algorithm | [ball_tree, kd_tree, brute] |
| leaf_size | [20, 30, 40] |
| n_neighbors | [2, 6] |
| p | [1, 2] |
| weights | [distância, uniforme] |

Para o algoritmo SVR foram utilizados os parâmetros abaixo, a combinação de valores utilizadas para esse algoritmo pode ser encontrada na Tabela 4.2:

- **C**: esse é o parâmetro de regularização do algoritmo SVR, permite controlar a penalização de erros na função objetivo, ou seja, tem o intuito de ajustar os pontos de treinamento, um valor mais baixo permite mais erros e a criação de um modelo mais generalizável. Foram utilizados as seguintes valores: [0.1, 1, 10].

- **epsilon**: a definição da zona de margem na qual os erros não são penalizados permeia nesse parâmetro, tem o objetivo de criar uma faixa de tolerância a erros na previsão, quanto maior o valor, maior a taxa de tolerância aos erros. Foram utilizados os seguintes valores: [0.1, 1, 10].

- **kernel**: aqui é definida a função kernel do SVR, essa escolha pode ter um grande impacto na transformação dos dados. Foram utilizadas as seguintes opções de kernel: ["poly", "rbf", "sigmoid"].

Tabela 4.2: Support Vector Regressor (SVR) - Parâmetros

| **Parâmetro** | **Valor** |
|---|---|
| C | [0.1, 1, 10] |
| epsilon | [0.1, 1, 10] |
| kernel | [“poly”, “rbf”, “sigmoid”] |

Para o algoritmo AdaBoostRegressor foram utilizados os seguintes parâmetros e a faixa de valores utilizados no experimento pode ser encontrada na Tabela 4.3:

- **n_estimators**: esse é o parâmetro de número de estimadores o qual define a quantidade de estimadores a serem utilizados, um número maior desse parâmetro define um modelo mais preciso, entretanto, pode ocasionar um tempo maior de treinamento. Foram utilizados as seguintes valores: [50, 100, 150].

- **learning_rate**: a definição desse parâmetro está relacionada a rapidez que o modelo de Aprendizado de Máquina aprende com os dados durante o treinamento, quanto maior a taxa, mais rápido é o treinamento, porém o modelo pode se tornar mais instável. Por outro lado, um valor menor desse parâmetro pode tornar o modelo mais estável, porém um treinamento mais lento. Foram utilizados os seguintes valores: [0.01, 0.1, 1.0].

Tabela 4.3: AdaBoostRegressor - Parâmetros

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | [50, 100, 150] |
| learning_rate | [0.01, 0.1, 1.0] |

Para o algoritmo Decision Tree foram utilizados os parâmetros abaixo e a combinação deles pode ser encontrada na Tabela 4.4:

- **max_depth**: esse parâmetro é responsável por definir a profundidade máxima que a árvore pode alcançar, uma profundidade menor representa uma árvore menos complexa, todavia, esse formato pode não capturar todas as combinações, uma maior profundidade representa uma árvore mais complexa ao capturar mais padrões nos dados. Foram utilizados os seguintes valores: [None, 10, 20, 30, 40, 50].

- **min_samples_split**: esse parâmetro define um valor mínimo para dividir um nó em amostras. Valores menores podem aprofundar as árvores, valores maiores limitam. Foram utilizados as seguintes funções: [2, 5, 10].

- **min_samples_leaf**: nesse contexto, é definido o número mínimo de amostras com relação ao que deve conter em uma folha. Valores menores geram menos amostras em cada folha, valores maiores fazem as folhas possuírem mais amostras. Foram utilizados os seguintes valores: [1, 2, 4].

- **max_features**: a quantidade máxima de características a serem utilizadas para encontrar a melhor divisão em cada nó é descrito nesse parâmetro. 'None' e 'auto' consideram todas as características e possuem comportamento semelhante. 'sqrt' considera a raiz quadrada do número total de características, para um conjunto que possui 9 dados, 3 características seriam consideradas. 'log2' considera o logaritmo base 2 para calcular as características. Foram utilizados os seguintes valores para este parâmetro: ["None", "auto", "sqrt", "log2"].

Tabela 4.4: Decision Tree - Parâmetros

| **Parâmetro** | **Valor** |
|---|---|
| max_depth | [None, 10, 20, 30, 40, 50] |
| min_samples_split | [2, 5, 10] |
| min_samples_leaf | [1, 2, 4] |
| max_features | [None, "auto", "sqrt", "log2"] |

Para o algoritmo Gradient Boosting Regressor foram utilizados os parâmetros abaixo e a combinação deles pode ser encontrada na Tabela 4.5:

- **n_estimators**: nesse contexto, é definido como a quantidade de árvores de decisão a serem criadas nesse processo. O valor 100 é um valor razoável considerando uma boa complexidade com um tempo de treinamento razoável. Quanto maior o valor desse parâmetro, maior é a capacidade do modelo de aprender diversos padrões, por outro lado, o tempo de treinamento pode ser aumentado drasticamente. Foram utilizados os seguintes valores: [100, 200, 300].

- **learning_rate**: esse parâmetro é responsável por definir a contribuição de cada árvore para o modelo resultante, valores menores permitem um aprendizado mais lento, e geralmente fornecem melhores desempenhos. O valor 0.2, por exemplo, aumenta a taxa de aprendizado, na qual o modelo aprende mais rápido, precisando de menos árvores, no entanto, podem fornecer um pior desempenho. Foram utilizados os seguintes valores: [0.01, 0.1, 0.2].

- **max_depth**: responsável por controlar a complexidade das árvores individuais através da profundidade máxima delas. Valores menores representam árvores menos profundas e menos complexas, podendo não capturar toda variabilidade dos dados. Diferentemente dos valores maiores, que representam árvores mais profundas e a captura de mais padrões complexos. Foram utilizados os seguintes valores: [3, 5, 7].

- **min_samples_split**: esse parâmetro define um valor mínimo para dividir um nó em amostras. Valores menores podem aprofundar as árvores, valores maiores limitam. Foram utilizados as seguintes funções: [2, 5, 10].

- **min_samples_leaf**: nesse contexto, é definido o número mínimo de amostras com relação ao que deve conter em uma folha. Valores menores geram menos amostras em cada folha, valores maiores fazem as folhas possuírem mais amostras. Foram utilizados os seguintes valores: [1, 2, 4].

Tabela 4.5: Gradient Boosting - Parâmetros

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | [100, 200, 300] |
| learning_rate | [0.01, 0.1, 0.2] |
| max_depth | [3, 5, 7] |
| min_samples_split | [2, 5, 10] |
| min_samples_leaf | [1, 2, 4] |

Para o algoritmo Random Forest foram utilizados os seguintes parâmetros e a combinação deles pode ser encontrada na Tabela 4.6:

- **n_estimators**: nesse contexto, é definido como a quantidade de árvores individuais que serão treinadas e combinadas para formar o modelo final. O impacto desse parâmetro pode ser avaliado no número de árvores, quanto maior, o desempenho do modelo costuma ser melhor, adicionalmente o tempo de treinamento aumenta na mesma medida. Para valores menores, menor é o tempo de treinamento e o algoritmo se torna menos complexo. Foram utilizados os seguintes valores: [100, 200, 300].

- **max_depth**: responsável por controlar a complexidade das árvores individuais através da profundidade máxima delas. O valor None significa que as árvores crescerão até que todas as folhas sejam puras. Foram utilizados os seguintes valores: [None, 10, 20, 30].

- **min_samples_split**: esse parâmetro define um valor mínimo para dividir um nó em amostras. Valores menores podem aprofundar as árvores, valores maiores limitam. Foram utilizados as seguintes funções: [2, 5, 10].

- **min_samples_leaf**: nesse contexto, é definido o número mínimo de amostras com relação ao que deve conter em uma folha. Valores menores geram menos amostras em cada folha, valores maiores fazem as folhas possuírem mais amostras. Foram utilizados os seguintes valores: [1, 2, 4].

- **max_features**: esse parâmetro é responsável por definir o número de recursos a serem considerados para dividir o nó. “auto” utiliza todos os recursos. “sqrt” usa a raiz quadrada do número total de recursos. O “log2” usa o logaritmo de base 2 do valor total de recursos. Foram utilizados os seguintes valores: [“auto”, “sqrt”, “log2”].

Tabela 4.6: Random Forest

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | [100, 200, 300] |
| max_depth | [None, 10, 20, 30] |
| min_samples_split | [2, 5, 10] |
| min_samples_leaf | [1, 2, 4] |
| max_features | [“auto”, “sqrt”, “log2”] |

### 4.2.3 Análise dos Resultados

Nesta seção, são apresentadas as melhores combinações de parâmetros obtidas por cada algoritmo e o melhor resultado selecionado dentre todos os algoritmos através da métrica $R^2$.

Este trabalho automatizou a escolha da melhor combinação de parâmetros para diferentes algoritmos, como *KNN*, *SVR*, *AdaBoost*, *Decision Tree*, *Gradient Boosting* e *Random Forest*. Cada algoritmo é avaliado por meio de diversas combinações de parâmetros, fornecidas através de uma lista de valores específicos para cada parâmetro. Utilizando a técnica exaustiva de GridSearch, conforme descrito em (AHMAD et al., 2022), é possível identificar a melhor combinação de parâmetros para cada algoritmo.

As Tabelas 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 se referem às melhores combinações de parâmetros obtidas dos algoritmos K-Nearest Neighbors, Support Vector Regressor, AdaBoostRegressor, Decision Tree, Gradient Boosting Regressor, e Random Forest, respectivamente.

Tabela 4.7: Melhor combinação de parâmetros - KNeighborsRegressor

| **Parâmetro** | **Valor** |
|---|---|
| Algorithm | ball_tree |
| leaf_size | 20 |
| n_neighbors | 2 |
| p | 1 |
| weights | distance |

Tabela 4.8: Melhor combinação de parâmetros - Support Vector Regressor (SVR)

| **Parâmetro** | **Valor** |
|---|---|
| C | 10 |
| epsilon | 0.1 |
| kernel | sigmoid |

Tabela 4.9: Melhor combinação de parâmetros - AdaBoostRegressor

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | 100 |
| learning_rate | 1.0 |

Tabela 4.10: Melhor combinação de parâmetros - Decision Tree

| **Parâmetro** | **Valor** |
|---|---|
| max_depth | 50 |
| min_samples_split | 2 |
| min_samples_leaf | 1 |
| max_features | None |

Tabela 4.11: Melhor combinação de parâmetros - Gradient Boosting

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | 300 |
| learning_rate | 0.1 |
| max_depth | 3 |
| min_samples_split | 10 |
| min_samples_leaf | 2 |

Tabela 4.12: Melhor combinação de parâmetros - Random Forest

| **Parâmetro** | **Valor** |
|---|---|
| n_estimators | 100 |
| max_depth | None |
| min_samples_split | 2 |
| min_samples_leaf | 1 |
| max_features | auto |

Após a execução do GridSearch, a melhor combinação de cada algoritmo é comparada entre si utilizando a métrica $R^2$, selecionando assim o melhor algoritmo dentre todos para a predição de momentos de stress e ansiedade do usuário. Para esses dados, o algoritmo que obteve o melhor resultado, segundo a métrica $R^2$, foi o algoritmo SVR, que com a combinação de parâmetros (C: 10, epsilon = 0.1, kernel = 'sigmoid') obteve o maior valor de $R^2$. Além disso, podemos observar os melhores valores de cada algoritmo na Figura 4.13.

Tabela 4.13: Comparação dos algoritmos de Aprendizado de Máquina

| **Algoritmo** | $R^2$ |
|---|---|
| K-Nearest Neighbors | 0.986 |
| SVR | **0.999** |
| AdaBoostRegressor | 0.978 |
| Decision Tree | 0.983 |
| Gradient Boosting | 0.996 |
| Random Forest | 0.988 |

Com esses resultados, é possível fazer uma comparação com os resultados obtidos por (SERGIO et al., 2023), onde o algoritmo *K-Nearest Neighboor* alcançou um valor aproximado de 0.982 no $R^2$. Inicialmente, foi verificado o melhor resultado alcançado por cada um dos modelos preditivos, ou seja, foram selecionados os modelos com maiores $R^2$. Para os algoritmos *K-Nearest Neighboor*, *Support Vector Machine*, *Decision Tree*, *Gradient Boosting* e *Random Forest* foram obtidos valores superiores aos obtidos na pesquisa realizada por (SERGIO et al., 2023). O único algoritmo que não conseguiu obter um valor

melhor de $R^2$ foi o *AdaBoostRegressor*. Estes resultados demonstram que foi possível obter valores significativos através das métricas dos modelos de Aprendizado de Máquina ao propor um incremento do trabalho citado anteriormente por meio das inclusões de novos modelos e a adesão às práticas de MLOps.

Em um segundo momento, foi calculada a média do $R^2$ em cada execução do treinamento dos modelos. O objetivo é verificar a estabilidade dos modelos, já que isso também é um fator importante na escolha de um algoritmo. Através da média das pontuações $R^2$ de um modelo após várias rodadas automatizadas de teste durante a validação cruzada, é possível obter uma visão geral de quão estáveis são os modelos e comparar seus desempenhos. Na Figura 4.4, observamos que, embora o algoritmo *SVR* tenha obtido o melhor valor de $R^2$ (considerando o máximo $R^2$), a média dos valores resultou em desempenhos menos satisfatórios. É possível observar que, na média, o valor alcançado por este modelo é significativamente mais baixo do que o valor máximo. Além disso, esse algoritmo foi o que apresentou a maior variação de valores.

O algoritmo *KNN* obteve valores consistentemente satisfatórios, com pouca variação durante todas as execuções. Esse fato mostra que o *KNN* é mais estável e pode ser por esse motivo que Sergio et al. (2023) identificaram esse modelo como a melhor opção. A modelagem do *AdaBoostRegressor* também apresentou resultados com baixa variação, mas seus valores de $R^2$ não foram tão altos quanto os de outros algoritmos. O *Decision Tree* exibiu uma maior variação nos resultados, com muitos *outliers* e desempenho geral inferior para os dados desse indivíduo. O *Gradient Boosting* mostrou uma grande quantidade de *outliers*, embora a maioria dos valores de $R^2$ ficou concentrada em níveis altos. Por fim, o algoritmo *Random Forest* alcançou bons valores de $R^2$ com baixa variância, mas não tão bons quanto o anterior.

A análise das médias dos valores de $R^2$ na Figura 4.4 indica que os modelos *KNN*, *Random Forest* e *AdaBoostRegressor* são os mais estáveis na representação dos dados do usuário. Assim, embora o *SVR* tenha alcançado o maior $R^2$, é necessário continuar atualizando automaticamente todos os modelos, pois pode ocorrer uma mudança no ranking quando novos dados forem incluídos na fase de treinamento.

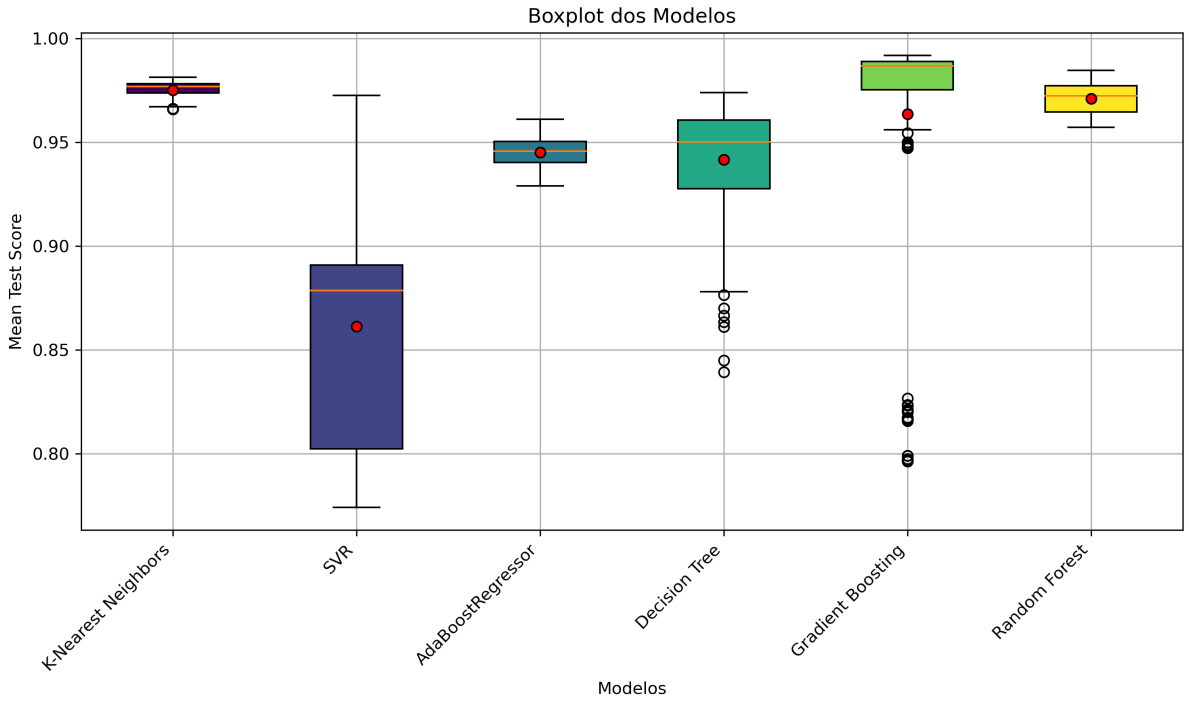


Figura 4.4: Gráfico de boxplot da pontuação de média para cada algoritmo

## 4.3 Limitações

Embora a implementação tenha aprimorado o sistema de monitoramento de saúde descrito por Sergio et al. (2023) e apresentado resultados promissores, é importante discutir as limitações encontradas durante a pesquisa para fornecer uma visão completa sobre este cenário de estudo.

Uma das limitações mais significativas deste estudo é o fato de que os dados utilizados para treinamento, validação e seleção dos modelos de Aprendizado de Máquina foram coletados de apenas um indivíduo, o que pode restringir a generalização dos modelos. As características dos pacientes podem variar bastante conforme idade, sexo, estilo de vida, entre outros fatores. Por isso, uma abordagem interessante seria o treinamento com dados de diversas pessoas. Isso permitiria que os modelos de Aprendizado de Máquina capturassem uma gama mais ampla de variações nos dados e, possivelmente, fornecessem previsões ainda mais complexas e precisas para diferentes cenários.

Dada a complexidade e particularidade do pré-processamento dos dados, não foi possível fazer a automação dessa etapa. Assim, toda a etapa de pré-processamento foi feita manualmente, sendo uma limitação do processo de automação dos modelos no

paradigma MLOps, uma abordagem interessante seria pensar em uma arquitetura mais inteligente para limpar os dados conforme a sua estrutura. Os algoritmos de Aprendizado de Máquina podem apresentar algumas limitações, como o ajuste excessivo aos dados de treinamento, resultando em perda de capacidade de adaptação e generalização a novos dados. A qualidade dos dados também pode influenciar significativamente no desempenho dos modelos; a incompletude e a presença de *outliers* podem afetar negativamente os resultados, sendo necessário um tratamento adequado desses aspectos, é valido destacar também, a importância de ter uma grande quantidade de dados de treinamento, para que os algoritmos possam ter contato com diversos contextos de informações.

Adicionalmente, sistemas tecnológicos que envolvem a saúde das pessoas devem ser seguros, transparentes e de fácil interpretação para os profissionais da área. Portanto, é imprescindível que essas arquiteturas se atentem a essas questões. Em resumo, apesar dos avanços significativos que esses algoritmos de Aprendizado de Máquina podem trazer para o cenário da saúde, essas arquiteturas devem garantir a eficácia e a segurança das aplicações em ambientes reais, mantendo a privacidade dos dados dos indivíduos e realizando um tratamento adequado com dados de diversos contextos para gerar bons modelos que representem a realidade.

# 5 Considerações Finais

Neste trabalho, foi proposta uma extensão da arquitetura de Sergio et al. (2023), que se baseia na distribuição em camadas *Edge*, *Fog* e *Cloud* para garantir escalabilidade, eficiência e modularidade no processamento de dados do usuário. Com base na arquitetura, foi desenvolvida uma infraestrutura computacional para automatizar as etapas de treinamento e disponibilização dos modelos preditivos. Esta infraestrutura computacional oferece uma abordagem para o desenvolvimento e implementação de um sistema de monitoramento da saúde das pessoas, focando na detecção de anomalias e na predição do comportamento do usuário com base em seus dados de sinais vitais.

Nesta extensão, foi explorada a aplicação das práticas de MLOps para melhorar o processo de desenvolvimento, treinamento e implantação de modelos de Aprendizado de Máquina. A integração de *pipelines* automatizados, ferramentas de monitoramento de desempenho e a automatização de funções específicas foram fundamentais para garantir a qualidade dos modelos gerados e a adequação dos mesmos ao usuário.

A arquitetura desenvolvida neste trabalho se mostrou eficaz no processamento e distribuição de dados vitais, além de proporcionar uma base sólida para geração de modelos preditivos utilizando Aprendizado de Máquina. Foi utilizada uma variedade de algoritmos para o treinamento desses modelos, como o *K-Nearest Neighbors*, *SVR*, *AdaBoostRegressor*, *Decision Tree*, *Gradient Boosting* e *Random Forest*. A avaliação desses algoritmos através de métricas estatísticas permitiu a seleção automática do modelo mais eficiente para cada contexto.

Por fim, os resultados obtidos indicam que a implementação de práticas de MLOps pode influenciar positivamente na escolha de modelos mais eficientes e adaptáveis, contribuindo significativamente para o desenvolvimento de sistemas de monitoramento de saúde e detecção de anomalias.

Em síntese, esta pesquisa abre caminho para estudos no campo do monitoramento de saúde, criação de modelos de comportamento e saúde de pacientes, destacando a integração da infraestrutura distribuída e a importância do uso de MLOps com os al-

goritmos de Aprendizado de Máquina para o desenvolvimento de sistemas mais eficientes e precisos.

Como trabalhos futuros, o pré-processamento de dados para o Aprendizado de Máquina seria uma área promissora, essa é uma etapa fundamental para a geração dos modelos. Automatizar essas tarefas pode aumentar a eficiência e a consistência dos experimentos, além de criar sistemas inteligentes que se ajustam dinamicamente às características dos dados e as particularidades de cada modelo de Aprendizado de Máquina. Ademais, a seleção de modelos de Aprendizado de Máquina pode ser ampliada para incluir diversos outros algoritmos e fornecer uma abordagem robusta para a explicabilidade com o intuito de gerar modelos mais transparentes e confiáveis. Uma próxima pesquisa poderia incluir algoritmos mais avançados, como redes neurais profundas, aprendizado não supervisionado, entre outros. A integração mais abrangente de modelos pode ser ótima para melhorar a capacidade de encontrar soluções para diferentes conjuntos de dados sobre diversos indivíduos. Essas vertentes combinadas, como a automação do pré-processamento, uma maior abrangência na escolha de modelos e a adoção de práticas de explicabilidade podem criar sistemas ainda mais robustos, adaptáveis e transparentes.

# Bibliografia

AHMAD, G. N.; FATIMA, H.; ULLAH, S.; SAIDI, A. S.; IMDADULLAH. Efficient medical diagnosis of human heart diseases using machine learning techniques with and without gridsearchcv. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 10, p. 80151–80173, 2022. ISSN 2169-3536. Disponível em: ⟨http://dx.doi.org/10.1109/access.2022.3165792⟩.

ALLA, S.; ADARI, S. K. What is mlops? In: _____. *Beginning MLOps with MLFlow*. Apress, 2020. p. 79–124. ISBN 9781484265499. Disponível em: ⟨http://dx.doi.org/10.1007/978-1-4842-6549-9_3⟩.

ALLI, A. A.; ALAM, M. M. The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications. *Internet of Things*, Elsevier BV, v. 9, p. 100177, mar. 2020. Disponível em: ⟨https://doi.org/10.1016/j.iot.2020.100177⟩.

ALZUBAIDI, L.; ZHANG, J.; HUMAIDI, A. J.; AL-DUJAILI, A.; DUAN, Y.; AL-SHAMMA, O.; SANTAMARÍA, J.; FADHEL, M. A.; AL-AMIDIE, M.; FARHAN, L. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, Springer Science and Business Media LLC, v. 8, n. 1, mar. 2021. Disponível em: ⟨https://doi.org/10.1186/s40537-021-00444-8⟩.

BOEHM. Software engineering. *IEEE Transactions on Computers*, Institute of Electrical and Electronics Engineers (IEEE), C-25, n. 12, p. 1226–1241, dez. 1976. Disponível em: ⟨https://doi.org/10.1109/tc.1976.1674590⟩.

BREIMAN, L. Random forests. *Machine Learning*, Springer Science and Business Media LLC, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: ⟨http://dx.doi.org/10.1023/A:1010933404324⟩.

CARBONELL, J. G.; MICHALSKI, R. S.; MITCHELL, T. M. An overview o machine learning. In: *Machine Learning*. Elsevier, 1983. p. 3–23. Disponível em: ⟨https://doi.org/10.1016/b978-0-08-051054-5.50005-4⟩.

CHOWDHARY, K. R. Natural language processing. In: *Fundamentals of Artificial Intelligence*. Springer India, 2020. p. 603–649. Disponível em: ⟨https://doi.org/10.1007/978-81-322-3972-7_19⟩.

COMPUTER Vision. In: COMPUTER Vision. Springer International Publishing, 2021. p. 209–209. Disponível em: ⟨https://doi.org/10.1007/978-3-030-63416-2_300082⟩.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, Springer Science and Business Media LLC, v. 20, n. 3, p. 273–297, set. 1995. ISSN 1573-0565. Disponível em: ⟨http://dx.doi.org/10.1007/BF00994018⟩.

COVER, T.; HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, Institute of Electrical and Electronics Engineers (IEEE), v. 13, n. 1, p. 21–27, jan. 1967. ISSN 1557-9654. Disponível em: ⟨http://dx.doi.org/10.1109/TIT.1967.1053964⟩.

DEO, R. C. Machine learning in medicine. *Circulation*, Ovid Technologies (Wolters Kluwer Health), v. 132, n. 20, p. 1920–1930, nov. 2015. Disponível em: ⟨https://doi.org/10.1161/circulationaha.115.001593⟩.

EBERT, C.; GALLARDO, G.; HERNANTES, J.; SERRANO, N. DevOps. *IEEE Software*, Institute of Electrical and Electronics Engineers (IEEE), v. 33, n. 3, p. 94–100, maio 2016. Disponível em: ⟨https://doi.org/10.1109/ms.2016.68⟩.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, Elsevier BV, v. 55, n. 1, p. 119–139, ago. 1997. ISSN 0022-0000. Disponível em: ⟨http://dx.doi.org/10.1006/jcss.1997.1504⟩.

FRIEDMAN, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 29, n. 5, out. 2001. ISSN 0090-5364. Disponível em: ⟨http://dx.doi.org/10.1214/aos/1013203451⟩.

HANDELMAN, G. S.; KOK, H. K.; CHANDRA, R. V.; RAZAVI, A. H.; HUANG, S.; BROOKS, M.; LEE, M. J.; ASADI, H. Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods. *American Journal of Roentgenology*, American Roentgen Ray Society, v. 212, n. 1, p. 38–43, jan. 2019. ISSN 1546-3141. Disponível em: ⟨http://dx.doi.org/10.2214/AJR.18.20224⟩.

KALYANE, P.; DAMANIA, J.; PATIL, H.; WARDULE, M.; SHAHANE, P. Student's performance prediction using decision tree regressor. In: _____. *Advanced Network Technologies and Intelligent Computing*. Springer Nature Switzerland, 2024. p. 286–302. ISBN 9783031640704. Disponível em: ⟨http://dx.doi.org/10.1007/978-3-031-64070-4_18⟩.

KREUZBERGER, D.; KüHL, N.; HIRSCHL, S. Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, Institute of Electrical and Electronics Engineers (IEEE), v. 11, p. 31866–31879, 2023. Disponível em: ⟨https://doi.org/10.1109/access.2023.3262138⟩.

LARCHER, L.; STROELE, V.; DANTAS, M.; BAUER, M. Event-driven framework for detecting unusual patterns in AAL environments. In: *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2020. Disponível em: ⟨https://doi.org/10.1109/cbms49503.2020.00065⟩.

LONGO, L.; GOEBEL, R.; LECUE, F.; KIESEBERG, P.; HOLZINGER, A. Explainable artificial intelligence: Concepts, applications, research challenges and visions. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020. p. 1–16. Disponível em: ⟨https://doi.org/10.1007/978-3-030-57321-8_1⟩.

MADAKAM, S.; RAMASWAMY, R.; TRIPATHI, S. Internet of things (IoT): A literature review. *Journal of Computer and Communications*, Scientific Research Publishing, Inc., v. 03, n. 05, p. 164–173, 2015. Disponível em: ⟨https://doi.org/10.4236/jcc.2015.35021⟩.

MAKINEN, S.; SKOGSTROM, H.; LAAKSONEN, E.; MIKKONEN, T. Who needs MLOps: What data scientists seek to accomplish and how can MLOps help? In: *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. IEEE, 2021. Disponível em: ⟨https://doi.org/10.1109/wain52551.2021.00024⟩.

SERGIO, W. L.; SILVA, G. di I.; STRöELE, V.; DANTAS, M. A. R. An architecture proposal to support e-healthcare notifications. In: *Advanced Information Networking and Applications*. Springer International Publishing, 2023. p. 157–170. Disponível em: ⟨https://doi.org/10.1007/978-3-031-29056-5_16⟩.

SIDDAPPA, P. *Towards sddressing MLOps pipeline challenges : practical guidelines based on a multivocal literature review*. Universität Stuttgart, 2022. Disponível em: ⟨http://elib.uni-stuttgart.de/handle/11682/12567⟩.

TAMBURRI, D. A. Sustainable MLOps: Trends and challenges. In: *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*. IEEE, 2020. Disponível em: ⟨https://doi.org/10.1109/synasc51798.2020.00015⟩.