



**INSTITUTO  
FEDERAL**  
Pernambuco

# Aplicação de Técnicas Automatizadas de Aprendizagem de Máquina para Classificação de Óbitos em Pacientes com Covid-19

## Application of Automated Machine Learning Techniques for Classification Mortality in Covid-19 Patients

Vinicius Ferreira de Lima<sup>1</sup>, Flávio Rosendo da Silva Oliveira<sup>1</sup>

<sup>1</sup> Análise e Desenvolvimento de Sistemas- Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco– Campus Paulista (IFPE)- Paulista- PE- Brasil

vlfl1@discente.ifpe.edu.br, flavio.oliveira@paulista.ifpe.edu.br

---

**Resumo.** Neste estudo, foi investigada a aplicação de técnicas de Automated Machine Learning na classificação de casos com potencial risco de óbito entre pacientes com COVID-19, utilizando um banco de dados histórico da Fiocruz, com dados de pacientes com COVID-19 em três períodos de tempo diferentes, correspondentes a diferentes fases de circulação da doença, classificando os pacientes que chegaram a óbito. O objetivo foi empregar ferramentas de AutoML ajustadas a esse cenário de classificação, avaliando o uso e potencial para pesquisas na área de análises clínicas. Com a implementação de ferramentas consolidadas na literatura de AutoML, como H2O AutoML, AutoGluon e PyCaret, a pesquisa indicou como essas tecnologias podem simplificar processos analíticos complexos, mostrando a relevância que essas ferramentas têm em tornar técnicas avançadas de aprendizagem de máquina acessíveis a profissionais não especializados e sua eficiência no processamento de dados clínicos. Embora tenham sido identificados desafios relacionados ao tempo de processamento de algumas ferramentas, concluiu-se que o AutoML possui um potencial significativo para facilitar e aprimorar a aplicação de modelos de aprendizagem de máquina na área de análises clínicas.

**Palavras-chave:** Aprendizagem de Máquina, Automated Machine Learning, COVID-19, Classificação de Óbitos

**Abstract.** This study investigated the application of automated machine learning techniques in classifying cases with a potential risk of death among COVID-19 patients, using a historical database from Fiocruz. This database contained data from COVID-19 patients in three different periods, corresponding to different phases of the disease's circulation, classifying patients who succumbed to the disease. The aim was to apply AutoML tools adapted to this classification scenario and to evaluate their use and potential for research in the field of clinical analysis. By implementing well-established AutoML tools in the literature, such as H2O AutoML, AutoGluon and PyCaret, this study demonstrated how these technologies can simplify complex analytical processes, highlighting the relevance of these tools in making advanced machine learning techniques accessible to non-specialists and their efficiency in processing clinical data. Although challenges related to the processing time of some tools were identified, it was concluded that AutoML has significant potential to facilitate and enhance the application of machine learning models in clinical analytics.

**Keywords:** Machine Learning, Automated Machine Learning, COVID-19, Death Classification

## 1. Introdução

A área de Aprendizagem de Máquina tem demonstrado uma relevância crescente em diversos campos de pesquisa devido à grande quantidade de dados disponíveis atualmente, especialmente na análise de dados clínicos (Truong et al., 2019). Há adoção de modelos é cada vez mais comuns para auxiliar profissionais de diversas áreas na tomada de decisões (Yang; Shi; Ni, 2020).

Durante a pandemia de COVID-19, o uso de modelos criados usando técnicas de Aprendizagem de Máquina na saúde ganhou ainda mais importância (Wynants et al., 2020). Contudo, a falta de familiaridade e a escassez de especialistas na área podem limitar o potencial de soluções baseadas na construção e otimização desses modelos (Hutter; Kotthoff; Vanschoren, 2019).

Nesse contexto, os algoritmos de *Automated Machine Learning* (AutoML) surgem como facilitadores, simplificando utilização, principalmente a parametrização desses modelos, auxiliando na otimização de hiperparâmetros e tornando acessíveis técnicas avançadas para profissionais sem formação especializada na área de Aprendizagem de Máquina (He; Zhao; Chu, 2021). Ao permitir a aplicação de algoritmos complexos de forma prática e eficaz, o AutoML desempenha um papel importante na ampliação do uso de modelos de Inteligência Artificial, contribuindo para o avanço de pesquisas clínicas e conseqüentemente para melhorias futuras nos cuidados com aos pacientes (Feurer et al., 2015; Zöller; Huber, 2021).

Diante disso, a aplicação de métodos automatizados de Aprendizagem de Máquina torna-se uma abordagem oportuna para facilitar o uso eficiente de modelos em pesquisas clínicas. Embora os métodos tradicionais, que envolvem a criação manual de modelos sem a utilização do AutoML, permitirem a parametrização detalhada e oferecendo flexibilidade na criação e otimização de modelos, eles têm demandado uma complexidade técnica tem maior, exigindo conhecimentos especializados para sua implementação adequada (Paladino et al., 2023).

Este estudo tem como objetivo analisar a aplicação de diversas ferramentas de AutoML na classificação de óbitos em uma base de dados da COVID-19, avaliando seu uso nesse contexto de dados. A proposta inclui a identificação das vantagens e limitações de cada uma das ferramentas selecionadas. O trabalho também busca oferecer *insights* para profissionais e pesquisadores interessados em adotar soluções de AutoML como essas, especialmente aqueles com nível técnico baixo em Aprendizagem máquina, facilitando a compreensão e aplicação dessas tecnologias.

As seções subsequentes deste documento são organizadas em quatro partes principais: Revisão da Literatura, Metodologia, Resultados, e Conclusão. Na [Seção 2](#), Revisão da Literatura, foi explorado os principais conceitos e estudos anteriores que se relacionam a Aprendizagem de Máquina, AutoML e suas aplicações na análise de dados de saúde, fornecendo a base teórica necessária para esse estudo. A [Seção 3](#), Metodologia, detalha os métodos e técnicas utilizados, incluindo a descrição dos dados, a abordagem experimental adotada, e os algoritmos de Aprendizagem de Máquina empregados, juntamente com os hiperparâmetros ajustados. Na [Seção 4](#), Resultados, foi discutido os procedimentos experimentais e apresentado os resultados obtidos, complementados por tabelas que mostram a eficácia dos modelos analisados. Por fim, a [Seção 5](#), Conclusão, sintetiza os principais contribuições deste trabalho e sugerindo direções para pesquisas futuras.

## 2. Revisão da Literatura

As ferramentas de AutoML são utilizados justamente na exploração e ajuste de diferentes modelos de Aprendizagem de Máquina de forma automatizada, reduzindo dessa maneira a necessidade de um especialista na área e promovendo praticidade na hora de encontrar o melhor modelo, que será usado

na classificação, em campos de pesquisa, como a área de Análises Clínicas ([Thirunavukarasu et al., 2023](#)).

Uma publicação do AI Institute da Shanghai Jiao Tong University, intitulada MedMNIST Classification Decathlon ([Yang; Shi; Ni, 2020](#)) e conduzida por Yang, Shi e Ni, ressalta a importância e os esforços dedicados ao ajuste de modelos de Aprendizagem de Máquina, na área de análises clínicas. O estudo destaca as ferramentas de AutoML como benéficas, agilizando e facilitando o desenvolvimento de modelos em nessas áreas de pesquisa.

Ainda no contexto da publicação do Institute de Shanghai, foram empregadas ferramentas como Auto-Sklearn, AutoKeras e Google AutoML Vision. Os resultados obtidos destacam a eficácia das ferramentas de AutoML no desenvolvimento de modelos de aprendizado de máquina, simplificando tanto a seleção de características quanto a otimização de hiperparâmetros. Em particular, o Google AutoML Vision demonstrou um bom desempenho ao lidar eficazmente com problemas como *overfitting*, mesmo diante dos desafios impostos pelos conjuntos de dados utilizados na pesquisa. A relevância dessas ferramentas é reafirmada, sugerindo que o uso de conjuntos de dados mais amplos, aliado a um monitoramento e otimização mais rigorosos, pode aprimorar ainda mais os resultados ([Yang; Shi; Ni, 2020](#)).

Um estudo da OpenML ([Gijssbers et al., 2019](#)) investigou 39 conjuntos de dados para classificação binária e multi-classe, realizando múltiplas iterações com amostras usando 10 *folds* em cada iteração. Além disso, utilizando *frameworks* distintos, como Auto-Weka, Auto-Sklearn, TPOT e H2O AutoML, os experimentos foram executados por períodos variáveis, abrangendo uma escala de 1 a 4 horas, totalizando 8000 horas.

Ao analisar o desempenho desses *frameworks*, observou-se que, após 1 hora, houve uma melhoria nas métricas avaliadas; entretanto, a maioria começou a evidenciar *overfitting*. Estendendo a análise para 4 horas, esses *frameworks* mantiveram desempenho competitivo. Contudo, nenhum deles superou a eficácia demonstrada pela *Random Forest* neste estudo. Foram destacados, os desafios enfrentados em contextos de alta dimensionalidade e em tarefas multi-classe para ferramentas de *Automated Machine Learning* (AutoML).

Cada *framework* apresenta particularidades que influenciam seu desempenho. O Auto-Weka, por exemplo, é reconhecido por sua abordagem baseada em otimização bayesiana, explorando eficientemente o espaço de hiperparâmetros para maximizar o desempenho do modelo ([Hutter; Kotthoff; Vanschoren, 2019](#)). Auto-sklearn destaca-se por otimizar pipelines de Aprendizagem de Máquina, incluindo seleção de modelos e pré-processamento de dados. O *Tree-based Pipeline Optimization Tool* (TPOT), utilizando programação genética, procura automaticamente as melhores combinações de transformações e modelos. Já o H2O AutoML é elogiado por sua versatilidade, suportando diversas tarefas de Aprendizagem de Máquina e otimizando automaticamente o processo de modelagem ([Truong et al., 2019](#)).

No contexto das análises clínicas, ferramentas como AutoGluon, AutoKeras e PyCaret foram utilizadas em um estudo para a predição de doenças cardíacas ([Paladino et al., 2023](#)). O AutoGluon foi escolhido por apresentar alto desempenho e facilidade de uso, utilizando diversos algoritmos para encontrar o melhor modelo para os dados.

O AutoKeras, reconhecido por suas previsões em pesquisas anteriores relacionadas à previsão de doenças cardíacas, treina *Deep Neural Network* para seleção de modelo e ajuste de hiperparâmetros com poucas entradas do usuário necessárias.

O PyCaret aplica vários algoritmos sobre os dados, definindo o melhor modelo possível com base em uma variedade de classificadores, como *Linear Discriminant Analysis (LDA)*, *Support Vector Machine (SVM)*, *Naive Bayes (NB)* e outros.

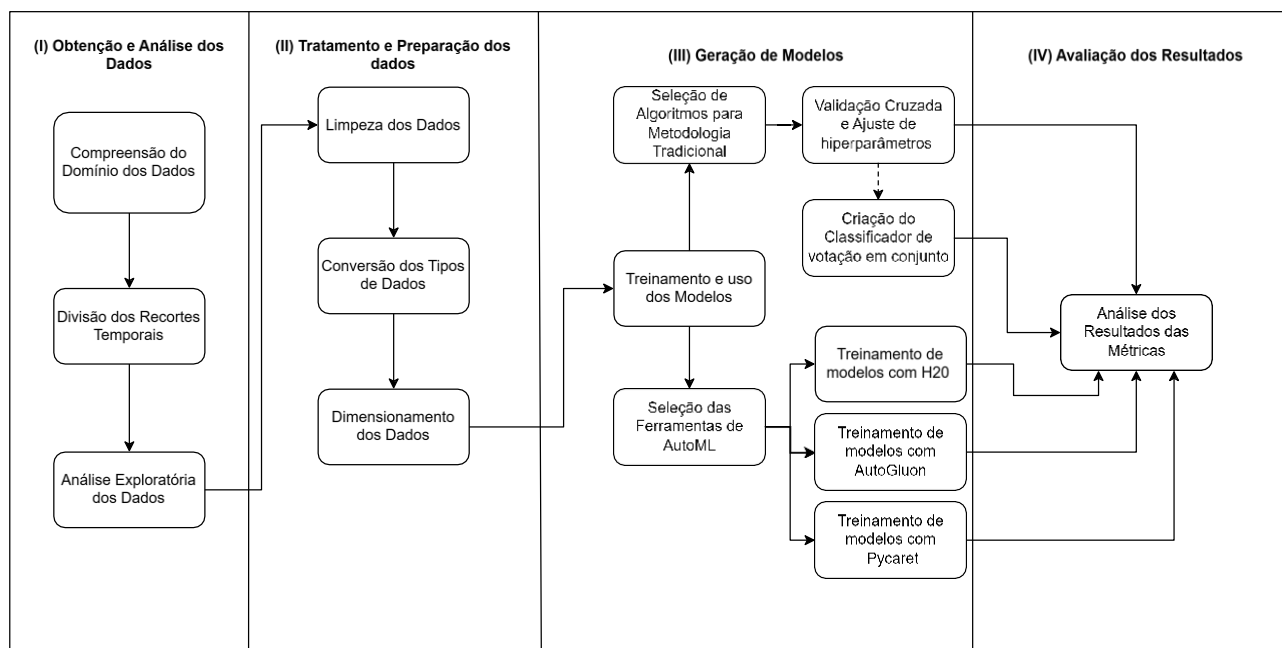
Ainda no contexto do estudo sobre a predição de doenças cardíacas, o AutoGluon demonstrou consistentemente superioridade em relação às demais ferramentas, alcançando taxas de precisão variando de 78% a 86%. Em comparação com métodos tradicionais de aprendizado de máquina, que geraram taxas de precisão de 55% a 60%, a vantagem das ferramentas AutoML ficou evidente (Paladino et al., 2023).

Em outro trabalho encontrado na literatura, conduzido por Richard Ooms e Marco Spruit, do Department of Information and Computing Sciences, Utrecht University, que investiga o uso de ferramentas de AutoML por médicos pesquisadores, destaca-se o potencial benéfico dessas ferramentas para análises médicas. Ferramentas como Hyperoptsklearn, autosklearn, Auto-Weka e TPOT foram utilizadas neste estudo, e os resultados indicaram que TPOT e Auto-Weka apresentaram os melhores desempenhos operando sobre os conjuntos de dados (Ooms; Spruit, 2020).

### 3. Metodologia

Pensando nos objetivos que esse trabalho visa alcançar, foi realizado a divisão de algumas etapas, desde da compreensão a definição do problema até a avaliação dos modelos desenvolvidos. Tradicionalmente, esse processo inclui o detalhamento do problema, a obtenção dos dados, a realização de análises exploratórias, a exploração de diferentes modelos e o ajuste dos mesmos (Domingos, 2012). Essas etapas são importantes para garantir a robustez e a validade dos modelos desenvolvidos.

O desenvolvimento deste trabalho se desdobrou em quatro fases principais: (I) Obtenção e Análise dos Dados, (II) Tratamento e Preparação dos dados, (III) Geração de Modelos, e (IV) Avaliação dos Resultados. A Figura 1 apresenta uma visão geral da metodologia empregada:



**Figura 1. Fases do Desenvolvimento do Trabalho**

Fonte: Elaborada pelo Autor.

Durante a fase de (I) Obtenção e Análise dos Dados, foram selecionados dados da Fiocruz, uma importante instituição de ciência e tecnologia em saúde. Os dados utilizados neste estudo são provenientes de uma base relacionada à COVID-19, fornecida por essa instituição. Esses dados incluem informações clínicas detalhadas, conforme a [Tabela 1](#), que se encontra no Apêndice, [Seção 6](#), bem como resultados de testes laboratoriais e diagnósticos, que indicam se a doença evoluiu para óbito ou não.

Ao longo dos experimentos, foram realizadas análises considerando três recortes temporais distintos da base de dados coletada pela instituição Fiocruz. Esses períodos foram selecionados de forma a representar diferentes fases da circulação do vírus COVID-19:

- Março a outubro de 2020
- Outubro de 2020 a Janeiro de 2021
- Fevereiro a Outubro de 2021

Essa abordagem permite avaliar o desempenho dos modelos em diferentes intervalos de tempo, aplicando-os a conjuntos de dados específicos de cada período. Dessa forma, gera-se uma compreensão a mais sobre as variações no comportamento dos modelos, oferecendo *insights* mais detalhados sobre como eles respondem a mudanças nos diferentes períodos da doença.

Para explorar e compreender melhor os dados, foi realizado uma Análise Exploratória de Dados (EDA), conforme descrito por Tukey ([Beyer, 1981](#)). Esta etapa envolveu a inspeção de estatísticas descritivas, a visualização de distribuições de dados e a identificação de padrões e *outliers*. Trata-se de um valor que se destaca significativamente sendo muito maior ou muito menor do que os outros presente em um conjunto de dados.

Foi realizado também uma análise de correlação para avaliar as relações entre os atributos e a variável alvo, que é a evolução da doença para óbito ou não. Esta análise revelou que nenhum dos atributos apresenta uma correlação alta com a variável alvo, indicando a ausência de uma relação forte e direta, de um dos atributos com a variável de interesse. Detalhes dessas correlações podem ser visualizados na [Figura 2](#), que se encontra no Apêndice, [Seção 6](#).

Foi realizada uma inspeção seguida por um processo (II) Tratamento e Preparação dos dados para lidar com valores ausentes, nulos, inconsistentes e *outliers*. Durante esse procedimento, identificou-se que certos atributos, como DATA ATUALIZAÇÃO, DATA DA NOTIFICAÇÃO, VACINADO, GESTANTE, RAÇA e PROSAÚDE, apresentavam uma quantidade significativa de valores nulos e inconsistentes. Por isso, optou-se pela exclusão desses atributos do conjunto de dados.

Na coluna SEXO, valores codificados como 2, que representam valor ignorado, foram substituídos por *Not a Number (NaN)*, conforme recomendado na documentação oficial da biblioteca NumPy. Essa biblioteca de código aberto para a linguagem Python, é utilizada para trabalhar com arrays multidimensionais e operações matemáticas. Ela utiliza a notação *NaN* para representar valores não numéricos ou ausentes ([Harris et al., 2020](#)). Da mesma forma, registros com valores de -1 também foram convertidos para NaN. Em seguida, todos os registros que tinha valores ausentes em alguma coluna (NaN) foram removidos. Essa decisão foi justificada pela pequena quantidade de registros excluído após todas as etapas realizadas anteriormente ser pequena comparada ao total dos registros presentes nos conjuntos de dados.

Foi realizado também uma conversão dos dados, transformando todos os dados categóricos em numéricos e assegurando que todos os dados estivessem no formato adequado para os modelos de Aprendizagem de Máquina. Foi aplicado o *StandardScaler* para normalizar as características dos

dados, garantindo que todas as variáveis estivessem na mesma escala, com média 0 e desvio padrão de 1. Estas etapas foram necessárias para preparar adequadamente os dados para o treinamento dos modelos (Amorim; Cavalcanti; Cruz, 2023).

Na etapa (III) Geração de Modelos, foi inicialmente adotada uma abordagem tradicional, na qual os modelos de Aprendizado de Máquina foram construídos sem o uso de ferramentas de AutoML. O objetivo dessa abordagem foi estabelecer uma linha de base para comparação, servindo como referência para os resultados obtidos posteriormente com os modelos gerados automaticamente pelas ferramentas de AutoML. Essa metodologia consistiu na seleção de algoritmos amplamente utilizados na área de Aprendizado de Máquina, seguidos por um processo de otimização de hiperparâmetros.

Primeiramente, os dados foram divididos em conjuntos de treino e teste, garantindo uma validação adequada. Em seguida, utilizou-se a técnica de busca de hiperparâmetro *HalvingRandomSearchCV*, que permite economizar recursos ao direcionar gradualmente os esforços para os candidatos mais promissores durante a exploração do espaço de hiperparâmetros, combinado com a validação cruzada para garantir a eficácia das configurações avaliadas (Bergstra; Bengio, 2012). Esse processo incluiu a definição de *Grids* de hiperparâmetros e a avaliação de diferentes combinações para melhorar o desempenho dos modelos, conforme detalhado na Tabela 2, disponível no Apêndice, Seção 6.

Os algoritmos selecionados para esta etapa incluem Random Forest (Breiman, 2001), Gradient Boosting (Friedman, 2001), Logistic Regression (Hosmer; Lemeshow; Sturdivant, 2000), AdaBoost (Freund; Schapire, 1997) e XGBoost (Chen; Guestrin, 2016), escolhidos devido à sua eficácia comprovada em tarefas de classificação. A otimização dos hiperparâmetros foi realizada para garantir que os modelos tivessem a melhor performance possível (Kuncheva, 2014). Utilizamos o Voting Classifier para combinar todos os modelos, cujos resultados, juntamente com os obtidos das ferramentas de AutoML, serão apresentados posteriormente.

A escolha dos hiperparâmetros foi baseada em uma combinação de práticas comuns na literatura para garantir uma exploração eficaz do espaço de hiperparâmetros, balanceando entre generalização e custo computacional.

Para o Logistic Regression, foram testados diferentes níveis de penalização, utilizando os valores 0,1, 1 e 10 para o grau de regularização. Além disso, foram avaliadas duas estratégias para a solução numérica do modelo: uma mais adequada para conjuntos de dados menores (*liblinear*) e outra mais eficiente para volumes maiores (*lbfgs*).

No Random Forest, foi analisado o impacto da quantidade de árvores no modelo, testando os valores 100, 200 e 300. Além disso, a profundidade máxima das árvores foi ajustada para nenhuma limitação, 10 e 20, permitindo avaliar o impacto no controle da complexidade e na prevenção de sobreajuste.

Para o Gradient Boosting, foram considerados diferentes números de estimadores (100, 200 e 300) e taxas de aprendizado de 0,1, 0,05 e 0,01. O objetivo foi encontrar um equilíbrio entre a capacidade do modelo de generalizar bem e o consumo computacional.

No AdaBoost, foram testadas quantidades de estimadores de 50, 100 e 200, avaliando sua influência no desempenho final do modelo. Também foram utilizadas diferentes taxas de aprendizado (1, 0,1 e 0,01) para analisar a relação entre a velocidade de treinamento e a capacidade de generalização do modelo.

Por fim, no XGBoost, foram combinados diferentes números de estimadores (100, 200 e 300) com taxas de aprendizado de 0,1, 0,05 e 0,01. As taxas menores foram exploradas para permitir

ajustes mais refinados no aprendizado do modelo, melhorando sua eficiência na etapa de reforço das previsões.

Essa abordagem possibilitou uma análise do desempenho dos modelos com diferentes configurações de hiperparâmetros para a escolha final dos algoritmo mais adequado para o problema de classificação de casos com potencial risco de óbito entre pacientes com COVID-19.

As ferramentas de AutoML utilizadas foram PyCaret v3.3.2, AutoGluon v1.1.1 e H2O v3.46.0.6, selecionadas devido à sua consolidação na literatura, robustez e capacidade de automatizar o processo de desenvolvimento de modelos. Cada ferramenta possui características distintas que influenciam seu desempenho e aplicabilidade em diferentes contextos de análise de dados dos clínicos, como visto em pesquisas anteriores (Paladino et al., 2023; Gijsbers et al., 2019).

Configuradas com ajustes específicos, como o tempo máximo de convergência dos modelos, a definição de uma semente para garantir a reprodutibilidade dos experimentos e o número de folds na validação cruzada. Além disso, foram mantidas as configurações padrão de cada ferramenta, incluindo suas listas internas de modelos e combinações de hiperparâmetros.

A H2O AutoML é amplamente reconhecida por sua versatilidade e capacidade de suportar uma ampla gama de tarefas de Aprendizagem de Máquina, otimizando automaticamente o processo de modelagem. Esta ferramenta é particularmente eficaz no manejo de grandes volumes de dados, oferecendo uma interface intuitiva que facilita a configuração e a realização de experimentos em ambientes dinâmicos (Ledell; Poirier, 2020).

AutoGluon oferece uma abordagem prática e eficiente para a descoberta do melhor modelo, empregando uma diversidade de algoritmos. Projetado para ser acessível a usuários de todos os níveis de experiência em Aprendizagem de Máquina, AutoGluon facilita a construção de modelos robustos com mínima intervenção manual. Esta ferramenta tem sido elogiada por sua capacidade de proporcionar resultados competitivos com pouco ou nenhum ajuste fino (Erickson et al., 2020).

PyCaret se destaca pela facilidade no processo de implementação de diversos algoritmos de Aprendizagem de Máquina, definindo o melhor modelo com base em uma variedade de classificadores. Como uma biblioteca de código aberto, PyCaret simplifica as tarefas de pré-processamento de dados, seleção de modelos e otimização de hiperparâmetros, tornando-a uma ferramenta valiosa para pesquisadores e profissionais que desejam agilizar esses processos. Sua arquitetura modular e suporte extensivo à comunidade são pontos fortes frequentemente citados na literatura, destacando sua capacidade de se adaptar a diferentes necessidades de pesquisa e desenvolvimento (Pol; Moharekar, 2021).

Neste estudo, foram estabelecidas configurações específicas para assegurar consistência e reprodutibilidade nos experimentos realizados. Uma das configurações mais importantes foi a aplicação do método de validação cruzada, com *fold* igual a 5. Essa abordagem divide os dados em cinco subconjuntos aproximadamente iguais. Em cada iteração, um subconjunto é reservado como conjunto de validação, enquanto os outros quatro são usados para treinamento do modelo. Após cinco iterações, cada subconjunto terá sido usado uma vez como validação, permitindo uma estimativa melhor do desempenho médio do modelo. A Escolha do valor de *fold* igual a 5 reflete um equilíbrio entre estabilidade das avaliações e tempo de processamento (Hastie; Tibshirani; Friedman, 2009).

Além disso, foi estipulado um limite de tempo de 600 segundos para o treinamento de cada ferramenta de AutoML. Esse tempo foi aplicado para controlar a varredura de hiperparâmetros e a geração de modelos em cada cenário experimental. Embora a ferramenta PyCaret não ofereça suporte

nativo para configuração de limites de tempo, os tempos registrados indicaram que todos os modelos convergiram de forma eficiente, sem exceder o limite estipulado.

A reprodutibilidade dos experimentos foi assegurada pela definição de uma semente fixa com valor igual a 42. Essa prática padroniza elementos aleatórios, como divisões de dados e inicializações de modelos, permitindo que os resultados possam ser replicados em execuções futuras.

Cada ferramenta de AutoML utilizou sua lista interna de modelos padrão, com diferentes combinações de hiperparâmetros ajustadas automaticamente. Ao final, foram selecionados os melhores modelos com base na métrica de acurácia. Os resultados gerados retonaram acurácia, *F1-Score* e tempo de treinamento. Para análise, foram considerados os três melhores modelos do PyCaret e do AutoGluon, e dois melhores do H2O de cada uma das rodadas dos experimentais.

Essas ferramentas destacam-se também pela facilidade de integração e execução em ambientes colaborativos, como o Google Colab, o que facilita a colaboração e o compartilhamento de resultados preliminares, particularmente em projetos de pesquisa de grande escala e de natureza interdisciplinar, sendo uma ótima opção para o dinamismo e a agilidade (Amorim; Cavalcanti; Cruz, 2023)

O Google Colab foi selecionado como o ambiente principal para a realização dos experimentos apresentados neste estudo. Essa escolha está alinhada com as necessidades de integração e simplicidade, que contribuíram para a execução eficiente das análises e a obtenção de resultados consistentes. Isso é especialmente útil em projetos de pesquisa de grande escala e de natureza interdisciplinar, onde a agilidade e o dinamismo são fundamentais (Amorim; Cavalcanti; Cruz, 2023).

Tal abordagem reflete as demandas por adaptabilidade e resposta rápida em pesquisas emergentes na área de Análises Clínicas usando modelos de Aprendizagem de Máquina. O Google Colab, amplamente reconhecido como uma ferramenta colaborativa versátil, demonstra ser uma escolha eficaz para esses objetivos (Werth et al., 2022).

As métricas de Acurácia e *F1 Score* foram utilizadas para avaliar o desempenho dos modelos desenvolvidos neste estudo. A acurácia mede a proporção de previsões corretas em relação ao total de previsões, sendo calculada conforme a Equação 1:

$$\text{Acurácia} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

Onde TP (*True Positives* - Verdadeiros Positivos) representa os casos corretamente classificados como positivos, TN (*True Negatives* - Verdadeiros Negativos) refere-se aos casos corretamente identificados como negativos, FP (*False Positives* - Falsos Positivos) indica os casos incorretamente classificados como positivos e FN (*False Negatives* - Falsos Negativos) corresponde aos casos positivos que foram erroneamente previstos como negativos. Essa métrica fornece uma visão geral do desempenho dos modelos, especialmente em cenários onde as classes estão balanceadas.

Por outro lado, o *F1 score*, conforme a Equação 2. Foi escolhido para complementar a avaliação, pois oferece uma medida harmônica entre *Precision*, proporção de previsões positivas corretas em relação ao total de previsões positivas, como na Equação 3 e *Recall*, proporção de verdadeiros positivos em relação ao total de casos positivos reais, conforme Equação 4. Isso é particularmente relevante em problemas onde as classes não são desbalanceadas.

$$\text{F1 score} = 2 \times \frac{(\textit{precision} \times \textit{recall})}{(\textit{precision} + \textit{recall})} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

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

#### 4. Resultados

Os resultados apresentados nesta seção refletem os desempenhos obtidos utilizando abordagens tradicionais de aprendizado de máquina e ferramentas de AutoML. Essa análise oferece uma visão abrangente da aplicação dessas técnicas em diferentes períodos, destacando os pontos fortes e limitações de cada abordagem.

**Tabela 3. Resultado dos modelos Tradicionais no Conjunto de Teste - MAR-OUT 2020**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
Gradient Boosting	0,7757	0,7588	40,72
XGBoost	0,7751	0,7751	11,44
VotingClassifier	0,7736	0,7592	19,52
Random Forest	0,7624	0,7499	59,97
Logistic Regression	0,7622	0,7452	1,20
AdaBoost	0,7597	0,7410	29,38

**Tabela 4. Resultado dos modelos gerados com AutoML no Conjunto de Teste - MAR-OUT 2020**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
NeuralNetTorch (AutoGluon)	0,7797	0,7637	289,13
WeightedEnsemble (AutoGluon)	0,7797	0,7637	290,13
Extreme Gradient Boosting (H2O)	0,7784	0,8627	2,21
Gradient Boosting Classifier (H2O)	0,7782	0,8621	3,58
CatBoost (AutoGluon)	0,7765	0,7585	23,77
Light Gradient Boosting Machine (PyCaret)	0,7669	0,8518	1,77
Gradient Boosting Classifier (PyCaret)	0,7663	0,8506	1,32
Extreme Gradient Boosting (PyCaret)	0,7637	0,8486	0,33

Os resultados obtidos para o período de março a outubro de 2020 destacam a superioridade de alguns modelos gerados por AutoML em relação aos métodos tradicionais, conforme a [Tabela 3](#) e [Tabela 4](#). Entre os modelos tradicionais, o Gradient Boosting e o XGBoost apresentaram os melhores desempenhos, com acurácias de 77,57% e 77,51%, respectivamente, com *F1-Score* 0,7588 e 0,7751, respectivamente. O XGBoost destacou-se com o maior *F1-Score* entre os modelos tradicionais, atingindo 0,7751, embora tenha apresentado um tempo de treinamento mais longo (19,52 segundos).

Nos modelos gerados por AutoML, o NeuralNetTorch e o WeightedEnsemble, ambos do AutoGluon, apresentaram a maior acurácia de 77,97% e *F1-Score* de 0,7637, mas com tempos de treinamento elevados (289,13 e 290,13 segundos). Já os modelos do H2O, como o Extreme Gradient Boosting e o Gradient Boosting Classifier, mostraram um excelente equilíbrio entre desempenho e eficiência, alcançando acurácias de 77,84% e 77,82% e *F1-Score* superiores (0,8627 e 0,8621) em tempos de execução baixos (2,21 e 3,58 segundos). Os modelos do PyCaret também se destacaram,

em eficiência com o Extreme Gradient Boosting alcançando 76,37% de acurácia e um *F1-Score* de 0,8468 em 0,33 segundos.

**Tabela 5. Resultado dos modelos tradicionais no Conjunto de Teste - OUT 2020 - JAN 2021**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
Gradient Boosting	0,7299	0,7128	24,34
VotingClassifier	0,7299	0,7132	6,01
XGBoost	0,7287	0,7120	8,11
Logistic Regression	0,7275	0,7102	2,75
AdaBoost	0,7191	0,6856	28,11
Random Forest	0,7078	0,6952	42,18

**Tabela 6. Resultado dos modelos gerados com AutoML no Conjunto de Teste - OUT 2020 - JAN 2021**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
Gradient Boosting Classifier (H2O)	0,7352	0,8263	0,96
Extreme Gradient Boosting (H2O)	0,7312	0,8248	0,77
Gradient Boosting Classifier (PyCaret)	0,7220	0,8114	0,40
NeuralNetTorch (AutoGluon)	0,7263	0,7098	119,96
Light Gradient Boosting Machine (AutoGluon)	0,7295	0,7074	34,02
CatBoost (AutoGluon)	0,7291	0,7110	19,38
Light Gradient Boosting Machine (PyCaret)	0,7158	0,8030	1,18
Linear Discriminant Analysis (PyCaret)	0,7155	0,8061	0,05

Os resultados obtidos no período de outubro de 2020 a janeiro de 2021 mostram que, entre os modelos tradicionais, o Gradient Boosting e o VotingClassifier se destacaram com uma acurácia de 72,99% e *F1-Score* de 0,7128 e 0,7132, respectivamente, combinando desempenho competitivo e tempos de treinamento relativamente curtos de 24,934 e 6,01 segundos, respectivamente, conforme [Tabela 5](#) e [Tabela 6](#). O XGBoost apresentou resultados próximos (acurácia de 72,87%, *F1-Score* de 0,7120), enquanto modelos como o AdaBoost e o Random Forest obtiveram valores inferiores, com acurácias de 71,91% e 70,78%, e tempos de treinamento mais longos e *F1-Score* abaixo de 0,70.

Entre os modelos gerados por AutoML, o Gradient Boosting Classifier do H2O liderou o desempenho, alcançando uma acurácia de 73,52% e *F1-Score* de 0,8263 em 0,96 segundos. Modelos como o Extreme Gradient Boosting (H2O) e o Light Gradient Boosting Machine (AutoGluon) também obtiveram bons resultados, com acurácias de 73,12% e 72,95%, respectivamente. Já os modelos do PyCaret e do AutoGluon apresentaram um equilíbrio interessante entre performance e tempo de execução, com destaque para o Linear Discriminant Analysis (PyCaret), que alcançou *F1-Score* de 0,8061 em 0,05 segundos, evidenciando a eficiência em questões de tempo de execução de modelos leves para certas aplicações de classificação.

**Tabela 7. Resultado dos modelos Tradicionais no Conjunto de Teste - FEV-OUT 2021**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
Random Forest	0,7400	0,7096	41,46
XGBoost	0,7375	0,6988	8,76
VotingClassifier	0,7343	0,6953	8,94
Logistic Regression	0,7305	0,7042	6,08
AdaBoost	0,7224	0,6632	26,12
Gradient Boosting	0,7194	0,6452	23,79

**Tabela 8. Resultado dos modelos gerados com AutoML no Conjunto de Teste - FEV-OUT 2021**

Modelo	Acurácia	F1-Score	Tempo de Treinamento(s)
Light Gradient Boosting Machine (PyCaret)	0,7459	0,8317	0,50
Gradient Boosting Classifier (PyCaret)	0,7453	0,8314	0,99
Gradient Boosting Classifier (H2O)	0,7431	0,8340	4,20
Extreme Gradient Boosting (H2O)	0,7430	0,8337	2,08
Linear Discriminant Analysis (PyCaret)	0,7432	0,8308	0,08
Light Gradient Boosting Machine (AutoGluon)	0,7422	0,7145	22,14
CatBoost (AutoGluon)	0,7408	0,7142	26,20
NeuralNetTorch (AutoGluon)	0,7400	0,7133	233,30

No período dos dados de fevereiro a outubro de 2021, os modelos tradicionais demonstraram desempenho moderado, com o Random Forest liderando em acurácia (74,00%) e *F1-Score* (0,7096), apesar de apresentar o maior tempo de treinamento (41,46 segundos), conforme [Tabela 7](#) e [Tabela 8](#). O XGBoost e o VotingClassifier alcançaram acurácias próximas (73,75% e 73,43%, respectivamente) e tempos de execução semelhantes (8,76 e 8,94 segundos), oferecendo boas combinações de eficiência e desempenho. Modelos como Logistic Regression e Gradient Boosting tiveram desempenhos mais baixos, destacando-se pela simplicidade computacional, enquanto o AdaBoost apresentou menor eficácia (72,24% de acurácia e *F1-Score* de 0,6632) em 26,12 segundos.

Nos modelos gerados por AutoML, as soluções do PyCaret foram competitivas, com o Light Gradient Boosting Machine liderando, alcançando uma acurácia de 74,59% e um *F1-Score* de 0,8317 em 0,50 segundos. O Gradient Boosting Classifier do PyCaret também se destacou com métricas muito próximas (acurácia de 74,53% e *F1-Score* de 0,8314) em 0,99 segundos. Os modelos do H2O, como o Gradient Boosting Classifier e o Extreme Gradient Boosting, também exibiram excelente desempenho, com acurácias acima de 74,30% e *F1-Scores* acima de 0,8330, mantendo tempos de treinamento abaixo de 5 segundos. Por outro lado, os modelos do AutoGluon, como o Light Gradient Boosting Machine e o CatBoost, apresentaram acurácias levemente inferiores (74,22% e 74,08%) e *F1-Score* menores (0,7145 e 0,7142), mas com tempos de treinamento significativamente mais elevados, sugerindo menor eficiência nesse contexto.

Na [Tabela 9](#), há um resumo da média de Acurácia e *F1-Score* para as ferramentas de Aprendizagem de Máquina analisadas ao longo de três períodos de distintos, do conjunto de dados. As ferramentas estão agrupadas nas categorias AutoGluon, H2O, PyCaret e Modelos Tradicionais, com o objetivo de fornecer uma visão sobre a aplicação e o desempenho geral de cada ferramenta e a abordagem usando a metodologia tradicional. As médias foram obtidas considerando os três períodos dos conjuntos de dados.

**Tabela 9. Média de Acurácia e F1-Score por Ferramenta**

Ferramenta	Média de Acurácia	Média de F1-Score
H2O	0,7522	0,8091
PyCaret	0,7473	0,7949
AutoGluon	0,7325	0,7483
Modelos Tradicionais	0,7278	0,7294

Os resultados indicam que H2O apresentou o melhor desempenho, com uma acurácia média de 75,22% e um *F1-Score* médio de 0,8091, superando as outras ferramentas analisadas. PyCaret obteve uma acurácia média de 74,73% e *F1-Score* médio de 0,7949, demonstrando bom desempenho também. Por outro lado, AutoGluon teve resultados ligeiramente inferiores, com uma acurácia média de 73,25% e *F1-Score* médio de 0,7483. Os Modelos Tradicionais, embora ainda apresentando um bom desempenho, ficaram atrás das ferramentas de AutoML, com acurácia média de 72,78% e *F1-Score* médio de 0,7294, destacando o potencial uso dos modelos gerados com ferramentas de AutoML em termos de precisão e equilíbrio das métricas.

**Tabela 10. Média de Acurácia dos Modelos Tradicionais**

Modelo	OUT 2020 - JAN 2021	MAR-OUT 2020	FEV-OUT 2021	Média Acurácia
XGBoost	0,7287	0,7751	0,7375	0,7471
VotingClassifier	0,7299	0,7736	0,7343	0,7459
Gradient Boosting	0,7299	0,7757	0,7194	0,7417
Logistic Regression	0,7275	0,7622	0,7305	0,7401
Random Forest	0,7078	0,7624	0,7400	0,7367
AdaBoost	0,7191	0,7597	0,7224	0,7337

**Tabela 11. Média de Tempo dos Modelos Tradicionais**

Modelo	OUT 2020 - JAN 2021	MAR-OUT 2020	FEV-OUT 2021	Média Tempo (s)
Logistic Regression	2,75	6,08	6,08	4,64
VotingClassifier	6,01	8,94	8,94	7,63
XGBoost	8,11	8,76	8,76	8,54
Gradient Boosting	24,34	23,79	23,79	24,31
AdaBoost	28,11	26,12	26,12	26,45
Random Forest	42,18	41,46	41,46	41,70

Na [Tabela 10](#), referente aos modelos tradicionais, o XGBoost foi o modelo mais consistente, com média de 74,71%, seguido de perto pelo VotingClassifier, com média de 74,59%. O VotingClassifier também apresentou um bom equilíbrio entre acurácia e desempenho ao longo dos três períodos. Modelos como o Logistic Regression e o Random Forest mantiveram-se com médias razoáveis, mas não conseguiram superar os modelos mais complexos. Já o AdaBoost apresentou o desempenho mais modesto entre os modelos tradicionais, com uma média de 73,37%, refletindo sua limitação para as características específicas dos dados.

Em relação ao tempo dos modelos tradicionais dentro dos períodos dos dados, apresentado na [Tabela 11](#). É possível notar a eficiência em destaque do Logistic Regression, apresentando um resultado com médias razoáveis de acurácia e um tempo significativamente bom em relação aos modelos tradicionais.

**Tabela 12. Média de Acurácia dos Modelos Gerados com AutoML**

Modelo	OUT 2020 - JAN 2021	MAR-OUT 2020	FEV-OUT 2021	Média Acurácia
Gradient Boosting Classifier (H2O)	0,7352	0,7782	0,7431	0,7522
Extreme Gradient Boosting (H2O)	0,7312	0,7637	0,7430	0,7460
NeuralNetTorch (AutoGluon)	0,7263	0,7797	0,7400	0,7487
CatBoost (AutoGluon)	0,7291	0,7765	0,7408	0,7488
Gradient Boosting Classifier (PyCaret)	0,7220	0,7663	0,7453	0,7445
Light Gradient Boosting Machine (AutoGluon)	0,7295	0,7669	0,7422	0,7462
Light Gradient Boosting Machine (PyCaret)	0,7158	0,7641	0,7459	0,7419
Linear Discriminant Analysis (PyCaret)	0,7155	0,7452	0,7432	0,7413

**Tabela 13. Média de Tempo dos Modelos Gerados com AutoML**

Modelo	OUT 2020 - JAN 2021	MAR-OUT 2020	FEV-OUT 2021	Média Tempo (s)
Linear Discriminant Analysis (PyCaret)	0,05	0,08	0,08	0,07
Light Gradient Boosting Machine (PyCaret)	1,18	0,50	0,50	0,73
Gradient Boosting Classifier (PyCaret)	0,40	0,99	0,99	0,79
Extreme Gradient Boosting (H2O)	0,77	2,08	2,08	1,64
Gradient Boosting Classifier (H2O)	0,96	4,20	4,20	3,12
CatBoost (AutoGluon)	19,38	26,20	26,20	23,26
Light Gradient Boosting Machine (AutoGluon)	34,02	22,14	22,14	26,77
NeuralNetTorch (AutoGluon)	119,96	233,30	233,30	195,19

Na [Tabela 12](#), referente aos modelos gerados com AutoML, é possível observar que, de forma geral, os modelos do H2O e do PyCaret se destacaram pela consistência e desempenho ao longo dos períodos. O Gradient Boosting Classifier do H2O se manteve como o modelo mais preciso, alcançando uma média de 75,22%. O Extreme Gradient Boosting (H2O) também teve bom desempenho com uma média de 74,60%, seguido de perto pelo NeuralNetTorch (AutoGluon), com média de 72,63%. Os Modelos como o NeuralNetTorch (AutoGluon) e o CatBoost (AutoGluon) apresentaram desempenho competitivo, embora com tempos de treinamento mais altos, especialmente o NeuralNetTorch.

Na [Tabela 13](#), os modelos do PyCaret, especialmente o Light Gradient Boosting Machine, se mostraram eficientes, com média de tempo 0,73 segundos, com o valor acurácia de 74,19% exibido anteriormente, mostra uma boa relação entre acurácia e tempo de treinamento presente nos modelos gerados por essa ferramenta.

H2O e PyCaret demonstraram o melhor equilíbrio entre acurácia e tempo de processamento, com H2O se destacando em acurácia e PyCaret sendo o mais rápido. AutoGluon, embora ofereça boas acurácias, é significativamente mais lento. Os Modelos Tradicionais, apesar de fornecerem resultados melhores ao AutoGluon em termos de tempo, são muito mais lentos, em relação as outras duas ferramentas de AutoML como PyCaret e o H2O. Isso reforça a ideia de que as ferramentas de AutoML podem ser vantajosas em termos de desempenho e eficiência, com bons resultados de acurácia e tempo.

## 5. Conclusão

Este estudo analisou o desempenho de diferentes abordagens de aprendizado de máquina, comparando métodos tradicionais e ferramentas de AutoML na classificação de dados relacionados à COVID-19. Os resultados evidenciam que as ferramentas de AutoML, como PyCaret, H2O e AutoGluon, apresentaram um média de resultados melhores que dos métodos tradicionais em termos de acurácia e *F1-Score*, conforme a [Tabela 10](#), mostrando-se mais adaptáveis às mudanças nos dados ao

longo dos períodos analisados. Modelos como o Gradient Boosting e o XGBoost, embora competitivos, foram superados pelas soluções de AutoML, que se destacaram pela automação e pela habilidade de otimizar múltiplas métricas de desempenho.

As ferramentas de AutoML como o Gradient Boosting Classifier (H2O) e o Extreme Gradient Boosting (H2O) se destacaram, alcançando médias de acurácia 75,22% e 74,60%, evidenciando sua eficiência e consistência. Além disso, O NeuralNetTorch (AutoGluon) apresentou uma acurácia média, alcançando até 74,87% nos períodos analisados, enquanto os modelos tradicionais chegaram no máximo a 74,71% com o XGBoost. Apesar dos desafios relacionados ao tempo de treinamento, especialmente no caso de modelos como o NeuralNetTorch do AutoGluon, que apresentou uma média de tempo de 197,19 segundos, essas ferramentas mostraram-se vantajosas em contextos que exigem alta acurácia e podem tolerar maior tempo de processamento. Já os modelos tradicionais, como o AdaBoost, enfrentaram dificuldades em atingir bons resultados, destacando a necessidade de abordagens mais eficientes e escaláveis.

Os modelos gerados pelo PyCaret, embora não tenham apresentado a maior acurácia média entre todas as abordagens, obtiveram resultados expressivos tanto em acurácia quanto em F1-Score, especialmente quando comparados à metodologia tradicional. Além disso, destacaram-se pelos menores tempos de convergência. Um bom exemplo é o Linear Discriminant Analysis (PyCaret), que apresentou um tempo médio de convergência de 0,07 segundos ao longo dos três períodos analisados.

Em pesquisas futuras pode-se ser explorado a combinação de ferramentas de AutoML com outras estratégias avançadas de engenharia de características e otimização de hiperparâmetros, isso pode apresentar resultados ainda melhores em pesquisas futuras. Além disso, seria relevante investigar sua aplicação em outros contextos epidemiológicos e de saúde pública. Este aprofundamento pode ampliar a aplicabilidade dos modelos em análises clínicas, auxiliando no desenvolvimento de soluções para o enfrentamento de desafios médicos, minimizando a necessidade de um especialista de um especialista em Ciência de dados para geração de modelos de Aprendizagem Máquina.

## **Agradecimentos**

Agradecemos à Dra. Lousiana Regadas de Macedo Quinino e a FIOCRUZ-PE pela disponibilização da base de dados utilizada neste trabalho.

## Referências

- AMORIM, Lucas B.V. de; CAVALCANTI, George D.C.; CRUZ, Rafael M.O. The choice of scaling technique matters for classification performance. **Applied Soft Computing**, v. 133, p. 109924, 2023. ISSN 1568-4946. DOI: 10.1016/j.asoc.2022.109924. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S1568494622009735>. Acesso em: 24 fev. 2025
- BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, p. 281–305, fev. 2012. Disponível em: <https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf> . Acesso em: 24 fev. 2025.
- BEYER, H. Tukey, John W.: Exploratory Data Analysis. Addison-Wesley Publishing Company Reading, Mass. — Menlo Park, Cal., London, Amsterdam, Don Mills, Ontario, Sydney 1977, XVI, 688 S. **Biometrical Journal**, v. 23, n. 4, p. 413–414, 1981. DOI: 10.1002/bimj.4710230408. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/bimj.4710230408>. Acesso em: 24 fev. 2025.
- BREIMAN, Leo. Random Forests. **Machine Learning**, v. 45, p. 5–32, out. 2001. DOI: 10.1023/A:1010950718922. Disponível em: [https://www.researchgate.net/publication/275342330\\_Random\\_Forests](https://www.researchgate.net/publication/275342330_Random_Forests). Acesso em: 24 fev. 2025.
- CHEN, Tianqi; GUESTRIN, Carlos. Xgboost: A scalable tree boosting system. In: PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [S.l.: s.n.], 2016. P. 785–794. Disponível em : <https://arxiv.org/abs/1603.02754>. Acesso em: 24 fev. 2025.
- DOMINGOS, Pedro. A few useful things to know about machine learning. **Communications of the ACM**, v. 55, n. 10, p. 78–87, 2012. Disponível em: <https://dl.acm.org/doi/10.1145/2347736.2347755>. Acesso em: 24 fev. 2025.
- ERICKSON, Nick et al. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. **arXiv preprint arXiv:2003.06505**, 2020. Disponível em: <https://arxiv.org/abs/2003.06505>. Acesso em: 24 fev. 2025.
- FEURER, Matthias et al. Efficient and Robust Automated Machine Learning. In \_\_\_\_\_ . **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2015. v. 28.
- FREUND, Yoav; SCHAPIRE, Robert E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. **J. Comput. Syst. Sci.**, v. 55, n. 1, p. 119–139, 1997. DOI: 10.1006/jcss.1997.1504. Disponível em: <https://doi.org/10.1006/jcss.1997.1504>. Acesso em: 24 fev. 2025.
- FRIEDMAN, Jerome H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, v. 29, n. 5, p. 1189–1232, 2001. DOI: 10.1214/aos/1013203451. Disponível em: <https://doi.org/10.1214/aos/1013203451>. Acesso em: 24 fev. 2025.
- GIJSBERS, Pieter et al. An Open Source AutoML Benchmark. **CoRR**, v. 1907.00909, 2019. arXiv: 1907.00909. Disponível em: : <http://arxiv.org/abs/1907.00909>. Acesso em: 24 fev. 2025.
- Instituto Federal de Educação, Ciências e Tecnologia de Pernambuco. *Campus Paulista*. Curso de 15 Análise e Desenvolvimento de Sistemas. 31 de Janeiro de 2025.

HARRIS, Charles R. et al. Array programming with NumPy. **Nature**, v. 585, n. 7825, p. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2. Disponível em: <https://doi.org/10.1038/s41586-020-2649-2>. Acesso em: 24 fev. 2025.

HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. [S.l.]: Springer, 2009. (Springer Series in Statistics). DOI: 10.1007/978-0-387-84858-7. Disponível em: <https://doi.org/10.1007/978-0-387-84858-7>. Acesso em: 24 fev. 2025.

HE, Xin; ZHAO, Kaiyong; CHU, Xiaowen. AutoML: A survey of the state-of-the-art. **Knowledge-Based Systems**, Elsevier BV, v. 212, p. 106622, jan. 2021. ISSN 0950-7051. DOI: 10.1016/j.knosys.2020.106622. Disponível em: <http://dx.doi.org/10.1016/j.knosys.2020.106622> . Acesso em: 24 fev. 2025.

HOSMER, David W.; LEMESHOW, Stanley; STURDIVANT, Rodney X. **Applied Logistic Regression**. Edição: Frank Hutter, Lars Kotthoff e J. Vanschoren. [S.l.]: John Wiley & Sons, Inc, 2000. ISBN 978-0-471-72214-4. DOI: 10.1002/0471722146.

HUTTER, Frank; KOTTHOFF, Lars; VANSCHOREN, J. (Ed.). **Automatic machine learning: methods, systems, challenges**. Alemanha: Springer, 2019. (Challenges in Machine Learning). ISBN 978-3-030-05317-8. DOI: 10.1007/978-3-030-05318-5.

KUNCHEVA, Ludmila. **Combining Pattern Classifiers: Methods and Algorithms: Second Edition**. [S.l.]: John Wiley & Sons, ago. 2014. v. 47. ISBN 978-1-118-91456-4. DOI: 10.1002/0471660264.

LEDELL, Erin; POIRIER, Sebastien. **H2o automl: scalable automatic machine learning**. In: ICML SAN DIEGO, CA, USA. PROCEEDINGS of the AutoML Workshop at ICML. [S.l.: s.n.], 2020. Disponível em: [https://www.automl.org/wp-content/uploads/2020/07/AutoML\\_2020\\_paper\\_61.pdf](https://www.automl.org/wp-content/uploads/2020/07/AutoML_2020_paper_61.pdf). Acesso em: 24 fev. 2025.

OOMS, Richard; SPRUIT, Marco. Self-Service Data Science in Healthcare with Automated Machine Learning. **Applied Sciences**, v. 10, p. 2992, abr. 2020. DOI: 10.3390/app10092992. Disponível em: : <https://www.mdpi.com/2076-3417/10/9/2992>. Acesso em: 24 fev. 2025.

PALADINO, L. M. et al. Evaluating the performance of automated machine learning (AutoML) tools for heart disease diagnosis and prediction. **AI**, v. 4, p. 1036–1058, 2023. DOI: 10.3390/ai4040053. Disponível em: <https://www.mdpi.com/2673-2688/4/4/53>. Acesso em: 24 fev. 2025.

POL, Urmila; MOHAREKAR, Tejashree. Automl: Building An Classification Model With Pycaret. **Ymer**, v. 20, p. 547–552, dez. 2021. Disponível em: [https://www.researchgate.net/publication/356749319\\_Automl\\_Building\\_An\\_Classification\\_Model\\_With\\_Pycaret](https://www.researchgate.net/publication/356749319_Automl_Building_An_Classification_Model_With_Pycaret) . Acesso em: 24 fev. 2025.

THIRUNAVUKARASU, Arun James et al. Clinical performance of automated machine learning: a systematic review. **medRxiv**, Cold Spring Harbor Laboratory Press, 2023. DOI: 10.1101/2023.10.26.23297599. Disponível em:

<https://www.medrxiv.org/content/10.1101/2023.10.26.23297599v1> . Acesso em: 24 fev. 2025.

TRUONG, Anh et al. Towards Automated Machine Learning: Evaluation and Comparison of AutoML Approaches and Tools. **CoRR**, v. 1908.05557, 2019. arXiv: 1908.05557. Disponível em:

<http://arxiv.org/abs/1908.05557>. Acesso em: 24 fev. 2025.

WERTH, Alexandra et al. Engagement in collaboration and teamwork using Google Colaboratory. In \_\_\_\_\_ . **2022 Physics Education Research Conference Proceedings**. [S.l.: s.n.], set. 2022.

P. 481–487. DOI: 10.1119/perc.2022.pr.Werth. Disponível em:

<https://ui.adsabs.harvard.edu/abs/2022perc.conf.481W>. Acesso em: 24 fev. 2025.

WYNANTS, L. et al. Prediction models for diagnosis and prognosis of COVID-19: systematic review and critical appraisal. **BMJ**, v. 369, p. m1328, 2020. Disponível em:

<https://www.bmj.com/content/369/bmj.m1328>. Acesso em: 24 fev. 2025.

YANG, J.; SHI, R.; NI, B. MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis. **CoRR**, v. 2010.14925, 2020. arXiv: 2010.14925. Disponível em: <https://arxiv.org/abs/2010.14925>. Acesso em: 24 fev. 2025.

ZÖLLER, Marc-André; HUBER, Marco F. Benchmark and Survey of Automated Machine Learning Frameworks. **Journal of Artificial Intelligence Research**, v. v. 1904.12054, p. 409–472, 2021.

Disponível em: : <http://arxiv.org/abs/1904.12054>. Acesso em: 24 fev. 2025.

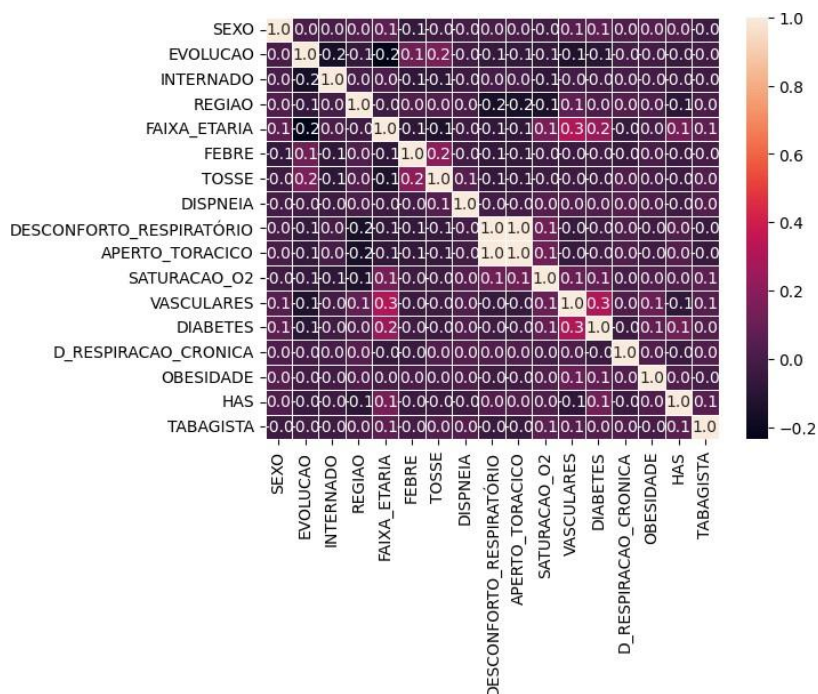
## 6. Apêndice I

Tabela 1. Descrição dos Atributos

Nome do Atributo	Descrição
SEXO	0- MASCULINO, 1- FEMININO, 2- IGNORADO
PRO SAÚDE	(-1) NaN, 0- Sim, 1- Não
GESTANTE	(-1) NaN, 0- Não, 1- Sim
VACINADO	(-1) NaN, 0- Não, 1- Sim
RAÇA	(-1) NaN, 0- Parda, 1- Branca, 2- Ignorado, 3- Preta, 4- Amarela, 5- Indígena
EVOLUÇÃO	(-1) NaN, 0- Óbito, 1- Não óbito
INTERNADO	(-1) NaN, 0- Sim, 1- Não
REGIAO	0- Interior, 1- RMR
FAIXA ETARIA	0- [0,9] F1, 1- [9,17] F2, 2- [17,39] F3, 3- [39,65] F4, 4- [65,75] F5, 5- (75, INF) F6
FEBRE	0- N Ã O, 1- SIM
TOSSE	0- N Ã O, 1- SIM
DISPNEIA	0- Não, 1- Sim
DESCONFORTO RESPIRATÓRIO	0- Não, 1- Sim
APERTO TORACICO	0- Não, 1- Sim
SATURACAO_O2	0- Não, 1- Sim
VASCULARES	0- Não (Doenças cardíacas ou Vasculares), 1- Sim
DIABETES	0- Não, 1- Sim
D.RESPIRACAO.CRONICA	0- Não, 1- Sim
OBESIDADE	0- Não, 1- Sim
HAS	0- Não, 1- Sim
TABAGISTA	0- Não, 1- Sim

**Tabela 2. Hiperparâmetros dos Classificadores *HalvingRandomSearchCV***

Classificadores	Hiperparâmetros
Logistic Regression	C: [0.1, 1, 10] Solver: [liblinear, lbfgs]
Random Forest	n_estimators: [100, 200, 300] max_depth: [None, 10, 20]
Gradient Boosting	n_estimators: [100, 200, 300] learning_rate: [0.1, 0.05, 0.01]
AdaBoost	n_estimators: [50, 100, 200] learning_rate: [1, 0.1, 0.01]
XGBoost	n_estimators: [100, 200, 300] learning_rate: [0.1, 0.05, 0.01]



**Figura 2. Análise de Correlação dos Atributos**  
 Fonte: Elaborada pelo Autor (Google Colab)