trAIn Health: Uma Plataforma para Experimentação Reprodutível em Aprendizagem de Máquina para Dados em Saúde

Thiago Guilherme Bezerra de Aguiar[1,3], **Renan T. Leite**[1,3], **Ana Beatriz S. V. dos Santos**[1,3], **Walter W. C. de Castro**[1,3], **Lourival Gerardo Silva Júnior**[1,2], **Joaquim Celestino Júnior**[1,2,3]

[1]Grupo de Redes e Sistemas Inteligentes (INETSYS)

[2]Programa de Pós-Graduação em Ciência da Computação (PPGCC)

[3]Universidade Estadual do Ceará (UECE) – Fortaleza, CE – Brasil

{thiago.aguiar, renan.teixeira, anab.vasconcelos,}@aluno.uece.br,

{walter.cavalcante, joaquim.celestino}@uece.br, lourival_junior@uvanent.br

Abstract. *Machine learning experimentation in clinical research remains costly due to fragmented workflows and limited methodological guidance. We present trAIn Health, a desktop platform that integrates model experimentation, contextualized scientific literature, and reproducible execution settings. The system supports 15 biomedical algorithms with configurable preprocessing, comparative evaluation, and experiment tracking. In experiments with 10 public datasets, the platform streamlined operational workflows into a unified process completed in minutes, while preserving transparency and technical rigor.*

Resumo. *A seleção de modelos em saúde ainda exige alto esforço técnico e forte suporte metodológico. Este trabalho apresenta o trAIn Health, plataforma desktop que unifica experimentação em aprendizado de máquina, literatura científica contextualizada e reprodutibilidade. A solução oferece 15 algoritmos, pipeline configurável de pré-processamento, comparação de métricas e histórico auditável de experimentos. Em 10 bases públicas, observou-se ganho operacional no fluxo de execução, com manutenção da clareza metodológica para uso em pesquisa clínica.*

1. Introdução

O aprendizado de máquina tem ampliado aplicações clínicas, mas a experimentação reprodutível ainda depende de alto esforço técnico e consulta dispersa à literatura [Goldstein et al. 2017, Johnson et al. 2016]. Na prática, pesquisadores em saúde enfrentam dificuldades para padronizar pré-processamento, comparar modelos e justificar escolhas metodológicas.

Apresentamos o trAIn Health, uma plataforma desktop que integra automação de experimentos, documentação científica contextualizada e controle de reprodutibilidade. A ferramenta implementa 15 algoritmos (classificação e regressão), com pipeline configurável e rastreabilidade das execuções. Entre os modelos suportados estão Random Forest [Breiman 2001] e Gradient Boosting [Friedman 2001].

O artigo descreve arquitetura, processo de desenvolvimento e avaliação em 10 bases públicas, indicando ganhos operacionais sem comprometer o rigor metodológico do fluxo experimental.

2. Trabalhos relacionados

Ferramentas existentes para ML em saúde incluem plataformas generalistas com interface gráfica (Weka [Hall et al. 2009], Orange [Demsar et al. 2013]), ambientes interativos baseados em scripts Python (Jupyter/Anaconda), abordagens AutoML [Feurer et al. 2015] e soluções centradas em dados clínicos [Wang et al. 2020]. Weka e Orange consolidaram o paradigma de experimentação visual com ampla cobertura de algoritmos clássicos, porém pressupõem familiaridade prévia do usuário com técnicas de ML e não incorporam fundamentação científica do domínio ao fluxo experimental. Embora ampliem produtividade, em geral não integram literatura ao momento da decisão experimental nem garantem reprodutibilidade completa do pipeline.

No domínio clínico, essa lacuna compromete validação por equipes multidisciplinares [Goldstein et al. 2017, Rajkomar et al. 2018]. O trAIn Health diferencia-se por combinar automação configurável, transparência metodológica e literatura contextualizada na própria interface, com foco em pesquisadores clínicos não especialistas em programação.

3. trAIn Health

A ferramenta foi projetada para quatro grupos de stakeholders prioritários: pesquisadores clínicos, cientistas de dados iniciantes, médicos colaboradores e estudantes de IA em saúde. A arquitetura segue princípios de modularidade com quatro componentes: `core`, `models` (Registry Pattern [Gamma et al. 1994]), `utils` e `ui` (PyQt6). A stack utiliza scikit-learn [Pedregosa et al. 2011] e XGBoost [Chen and Guestrin 2016], com suporte multiplataforma.

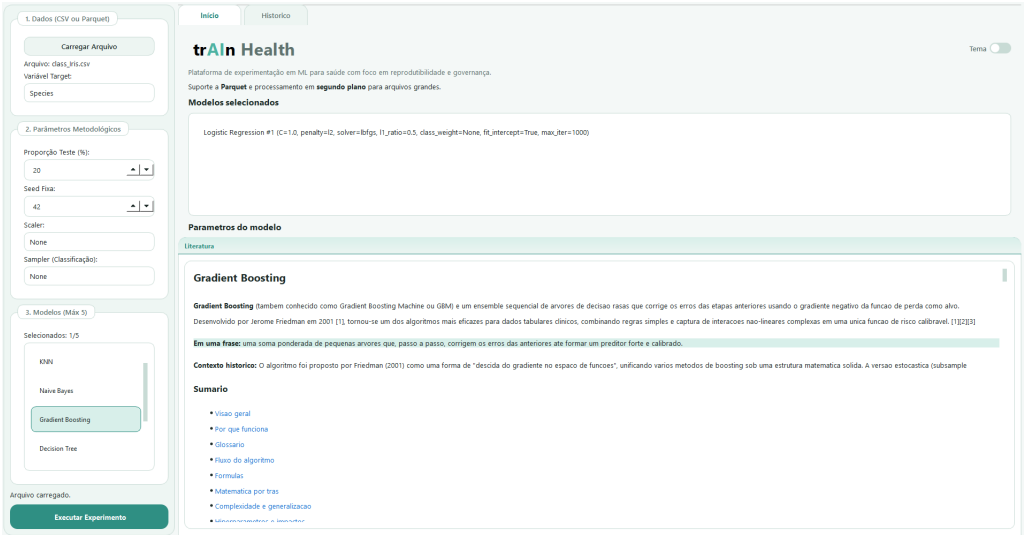

Figura 1. Interface principal do trAIn Health.

Um diferencial central é a **aba de literatura integrada**, que pode ser vista logo abaixo de parâmetros do modelo (Figura 1). Ao selecionar um

modelo, o usuário acessa documentação HTML com fundamentação matemática, orientações de uso, explicação de hiperparâmetros e referências clínicas (Framingham [Framingham Heart Study Collaboration 2026], MIMIC-III [Johnson et al. 2016]). A abordagem reduz a barreira de entrada ao oferecer contexto científico no momento da decisão. Em particular, a aba favorece a comparação fundamentada de diferentes abordagens de modelagem e a escolha informada de hiperparâmetros específicos diretamente na interface, sem necessidade de consultar múltiplas fontes externas.

3.1. Processo de desenvolvimento e requisitos

O desenvolvimento da plataforma seguiu uma abordagem ágil, inspirada no **SCRUM**, com ciclos iterativos curtos de implementação, demonstração e refinamento. Esse processo permitiu incorporar feedback frequente dos stakeholders e adaptar rapidamente decisões de interface, priorização de funcionalidades e fluxos experimentais.

Os requisitos foram elicitados por entrevistas com pesquisadores de IA e baseados em dificuldades relatadas por especialistas em saúde, complementadas por ciclos internos de revisão no laboratório. A especificação foi organizada em histórias de usuário, traduzidas em critérios objetivos de aceitação para cada entrega.

Formalizamos os requisitos em dois grupos. **Requisitos funcionais (RF):** importação/validação de dados, pré-processamento configurável, seleção e comparação de modelos, treinamento reprodutível, visualização/exportação de métricas e rastreabilidade histórica de experimentos. **Requisitos não funcionais (RNF):** segurança e proteção de dados sensíveis, privacidade em conformidade com princípios de LGPD/GDPR, desempenho computacional para execução interativa e usabilidade com interface intuitiva para públicos não especialistas.

3.2. Arquitetura e Implementação

A arquitetura do trAIn Health é modular (Figura 2), organizada em camadas lógicas (`ui`, `core`, `models` e `utils`), com separação de responsabilidades entre interface, processamento experimental e componentes de modelagem. Essa estrutura, aliada ao uso do **Registry Pattern** no módulo de algoritmos, favorece a extensibilidade do sistema.

O módulo `core` centraliza a preparação e a construção de pipelines; `models` fornece interface unificada e extensível; `utils` agrega avaliação e geração de relatórios; e `ui` implementa a interação em PyQt6, incluindo a aba de literatura com conteúdo técnico em HTML.

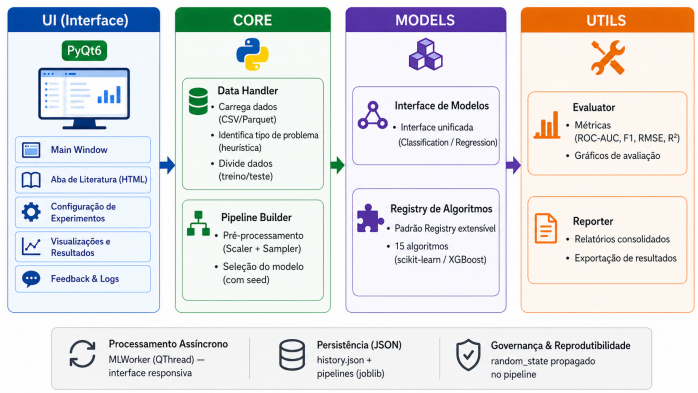

Figura 2. Arquitetura da aplicação.

Implementamos processamento assíncrono com `QThread` para manter a interface responsiva durante experimentos longos. A persistência em JSON registra histórico de execuções (timestamp, modelo, parâmetros, tempo e métricas), apoiando auditoria e comparação longitudinal de resultados.

3.3. Funcionalidades

A plataforma organiza o fluxo experimental em quatro blocos funcionais. **(i) Dados e pré-processamento:** importação CSV/Parquet, identificação automática do tipo de problema a partir da variável alvo, normalização (Standard, MinMax, Robust ou nenhuma) e balanceamento (SMOTE, over/under-sampling ou nenhum). **(ii) Modelagem:** seleção simultânea de até cinco algoritmos com configuração guiada de hiperparâmetros por formulários e apoio contextual pela aba de literatura. **(iii) Treinamento reprodutível:** execução assíncrona com controle de seed no pipeline completo e registro histórico em JSON. **(iv) Análise e governança:** geração de métricas/visualizações comparativas (p.ex., ROC-AUC, F1, RMSE, R^2, matriz de confusão, resíduos), exportação de resultados e histórico auditável para comparação longitudinal.

4. Avaliação de desempenho

A avaliação combinou três frentes: **scripts automatizados de validação** dos módulos centrais (`core`, `models`, `utils`) e da integração com `ui`; **análise quantitativa do fluxo unificado** sobre 10 bases públicas (tempos end-to-end e métricas preditivas); e **avaliação exploratória de usabilidade** conduzida internamente (equipe de desenvolvimento e colaboradores do laboratório). Um estudo formal com usuários externos está previsto como trabalho futuro.

As avaliações internas indicaram **percepção positiva preliminar**: intuitividade do fluxo, facilidade de comparação entre modelos e redução da barreira para pesquisadores não programadores; a literatura integrada foi apontada como diferencial para justificativa de hiperparâmetros.

Para avaliar o desempenho, executamos os testes sobre 10 bases públicas de tamanhos diferentes: 5 de classificação (Heart Disease, Adult Income, Default of Credit Card Clients, Iris, Wine Quality) e 5 de regressão (Bike Sharing, California Housing Prices, Medical Cost Personal, NYC Taxi Trip Duration, Online News Popularity). Cada experimento foi disparado em uma única ação na interface, com configuração padrão (sem normalização e sem balanceamento, `seed=42`, split 80/20), em ambiente Windows 11, processador Intel Core i5-10400F (6 núcleos / 12 threads, 2.90 GHz), 16 GB de RAM e Python 3.11. A Tabela 1 reporta o tempo total *end-to-end* do fluxo unificado por base, encadeando pré-processamento, treino e avaliação dos cinco modelos sem intervenção do usuário.

Tabela 1. Tempo total do fluxo unificado por base – Classificação.

Base	Tempo total	Gargalo
Heart Disease	2.3236s	SVM
Adult Income	592.8938s	SVM
Default of Credit Card Clients	150.8543s	SVM
Iris	0.9893s	RF
Wine Quality	2.8296s	XGB

Para evidenciar que os tempos da Tabela 1 correspondem a execuções com resultados preditivos coerentes, a Tabela 2 ilustra as métricas obtidas pelos cinco modelos sobre o caso *Heart Disease*, extraídas diretamente do histórico em JSON registrado pela própria plataforma.

Tabela 2. Métricas preditivas dos cinco modelos na base *Heart Disease*

Modelo	Accuracy	F1 (weighted)	ROC-AUC
Logistic Regression	0.840	0.850	0.905
Naive Bayes	0.840	0.847	0.903
Random Forest	0.929	0.933	0.973
SVM	0.735	0.745	0.811
XGBoost	0.920	0.924	0.969

Padrão análogo de tempos e qualidade preditiva é observado nas demais bases de classificação e nos cenários de regressão (avaliados por R^2 e RMSE), com o detalhamento completo por modelo e por base disponível como material suplementar no repositório da ferramenta.

Os tempos caracterizam o *throughput end-to-end* da ferramenta e não a velocidade isolada dos algoritmos, substituindo etapas que, em fluxo manual, exigiriam codificação e tabulação separadas por modelo. O custo total escala com o tamanho amostral e com o algoritmo dominante (SVM/SVR em bases médias, Random Forest em bases muito grandes), enquanto o XGBoost mantém desempenho competitivo sob restrição de tempo. Além do ganho operacional, o histórico em JSON, do qual a Tabela 2 foi extraída, garante reprodutibilidade e comparação metodologicamente consistente entre hipóteses.

5. Lições aprendidas

Uma lição central foi que, em IA clínica, **automação sem explicitação de premissas reduz confiança** e dificulta adoção em ambientes multi-disciplinares. A integração de literatura científica no fluxo de modelagem não apenas informa tecnicamente, mas **contribui para aumentar confiança** entre usuários (pesquisadores, médicos) e a ferramenta, podendo facilitar sua adoção em contextos aplicados onde rigor metodológico é crítico.

Enfrentamos desafios na integração da documentação HTML com a interface PyQt6, especialmente para manter legibilidade e navegação fluida entre algoritmos. As soluções incluíram separação modular da documentação por algoritmo, padronização visual e processamento assíncrono dos experimentos para preservar a responsividade da interface.

Por fim, o **feedback iterativo no laboratório** mostrou-se essencial: a colaboração contínua entre desenvolvedores e pesquisadores permitiu refinar interface, priorização de funcionalidades e fluxos de trabalho. Essa validação interna por especialistas resultou em uma ferramenta **mais alinhada** às necessidades práticas, reduzindo parcialmente o hiato entre protótipos acadêmicos e demandas de experimentação aplicada.

6. Conclusão e trabalhos futuros

Os resultados do trAIn Health indicam que automação de experimentação clínica pode ser combinada com transparência metodológica e reprodutibilidade. A plataforma reduz

esforço operacional no cenário avaliado e mantém suporte científico contextualizado para tomada de decisão.

Como trabalhos futuros, planejamos: (1) suporte a modelos de deep learning; (2) expansão da aplicação para abranger outras áreas; (3) mais opções de pre-processamento; (4) adicionar suporte a bases de dados não tratadas; (5) aumentar o nível de integração à clínica; e (6) avaliação com pesquisadores externos.

Disponibilidade: Código-fonte, vídeo de demonstração e datasets de validação estão disponíveis em repositório do projeto. Documentação técnica completa e guia de uso acompanham o material.

Referências

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

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Demsar, J. et al. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353.

Feurer, M. et al. (2015). Efficient and robust automated machine learning. In *Advances in Neural Information Processing Systems*, volume 28, pages 2962–2970.

Framingham Heart Study Collaboration (2026). Framingham heart study: Public access data sets. https://www.framinghamheartstudy.org/. Acesso em: 08 mar. 2026.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

Gamma, E. et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

Goldstein, B. A., Navar, A. M., and Pencina, M. J. (2017). Opportunities and challenges in developing risk prediction models with electronic health records data: A systematic review. *Journal of the American Medical Informatics Association*, 24(1):198–208.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Johnson, A. E. W. et al. (2016). Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035.

Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Rajkomar, A. et al. (2018). Scalable and accurate deep learning with electronic health records. *npj Digital Medicine*, 1(18):1–10.

Wang, S. et al. (2020). Mimic-extract: A data extraction, preprocessing, and representation pipeline for mimic-iii. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 222–235.