Modelagem e Análise de Algoritmos de Machine Learning

Alexandre Alberto Menon[1], Gabriel Rodrigues Estefanes[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)

Curitiba – PR – Brasil

{alexandremenon,gabrielestefanes}@alunos.utfpr.edu.br

Abstract. *The analysis of health data, particularly vital signs, is crucial for patient monitoring, and Machine Learning (ML) offers essential tools for decision support in this area. This paper focuses on the implementation and comparative analysis of ML algorithms for a supervised classification problem using a Vital Signs dataset. Three approaches were implemented and analyzed: two Symbolic Learning algorithms (ID3 and Random Forest) and a Neural Network (Multi-Layer Perceptron - MLP). The models were trained and evaluated following data preprocessing and a train-test split methodology. The results, measured by accuracy on the test set, showed that the Neural Network (MLP), configured with a single hidden layer, achieved the best performance (93.67%). This was followed by the ID3 algorithm (92.66%) and the Random Forest (92.00%). We conclude that the Neural Network demonstrated the best generalization performance for this specific dataset, and the "from scratch" implementation provided a deep understanding of each algorithm's internal mechanisms.*

Resumo. *A análise de dados de saúde, especificamente sinais vitais, é crucial para o monitoramento de pacientes, e a Aprendizagem de Máquina (AM) oferece ferramentas essenciais para o auxílio à decisão nessa área. Este trabalho foca na implementação e análise comparativa de algoritmos de AM para um problema de classificação supervisionada, utilizando um conjunto de dados de Sinais Vitais. Três abordagens foram implementadas e analisadas: duas de Aprendizagem Simbólica (ID3 e Random Forest) e uma Rede Neural (Perceptron de Múltiplas Camadas - MLP). Os modelos foram treinados e avaliados seguindo uma metodologia de pré-processamento e divisão dos dados em conjuntos de treino e teste. Os resultados, medidos pela acurácia no conjunto de teste, mostraram que a Rede Neural (MLP), configurada com uma única camada oculta, obteve o melhor desempenho (93,67%). Esta foi seguida pelo algoritmo ID3 (92,66%) e pelo Random Forest (92,00%). Conclui-se que a Rede Neural apresentou o melhor desempenho de generalização para este conjunto de dados, e a implementação "do zero" proporcionou uma compreensão aprofundada dos mecanismos internos de cada algoritmo.*

1 Introdução

A análise de dados de saúde, especificamente sinais vitais, é um pilar para o monitoramento de pacientes e o diagnóstico precoce de condições críticas. Com o crescente volume desses dados, a Aprendizagem de Máquina (AM) oferece ferramentas para auxiliar profissionais de saúde na tomada de decisão. Este trabalho foca na aplicação e análise comparativa de técnicas de AM para um problema de classificação supervisionada, utilizando um conjunto de dados de Sinais Vitais.

As abordagens escolhidas para o desenvolvimento de soluções neste artigo dividem-se em duas categorias principais: Aprendizagem Simbólica, representada pelos algoritmos ID3 (Iterative Dichotomiser 3) e Random Forest (RF), e uma abordagem utilizando Redes Neurais (RN). O ID3 e o Random Forest geram modelos baseados em regras e árvores, enquanto as Redes Neurais aprendem padrões complexos através de camadas de neurônios interconectados.

O presente artigo está organizado da seguinte forma: as Seções 2 e 3 detalham a fundamentação teórica dos algoritmos de Aprendizagem Simbólica (ID3 e Random Forest) e da Rede Neural, respectivamente. A Seção 4 descreve a metodologia, incluindo a análise do *dataset* e a parametrização de cada um dos três modelos. Na Seção 5, são apresentados e discutidos os resultados obtidos, incluindo uma análise comparativa e uma reflexão sobre os impactos éticos. Por último, as Seções 6, 7 e 8 apresentam as conclusões, a descrição de papéis da equipe e as referências bibliográficas.

2 Aprendizagem Simbólica

A Aprendizagem Simbólica é um paradigma de AM focado em gerar modelos que utilizam representações de alto nível, como regras lógicas e árvores de decisão. Esses modelos são frequentemente valorizados por sua alta interpretabilidade, permitindo que humanos compreendam o processo de tomada de decisão.

2.1 ID3

O ID3 (Iterative Dichotomiser 3) é um algoritmo clássico que constrói uma árvore de decisão de forma gulosa e top-down. O princípio central do ID3 é selecionar, a cada nó, o atributo que melhor divide o conjunto de dados em subconjuntos mais puros.

Para medir a "pureza" e selecionar o melhor atributo, o ID3 utiliza a métrica de **Ganho de Informação** (*Information Gain*), que é baseada no conceito de **Entropia** da teoria da informação. A Entropia **H(S)** de um conjunto de dados **S** (com **c** classes) mede sua incerteza:

$H(S) = -\sum_{i=1}^{c} pi\ log_2(pi)$, onde **pi** é a proporção de amostras em **S** que pertencem à classe **i**.

O Ganho de Informação **IG(S, A)** ao dividir o conjunto **S** usando um atributo **A** é a redução na entropia:

$IG(S,A) = H(S) - \sum_{v \in Valores\ (A)} \frac{|S_v|}{|S|} H(S_v)$, onde **Valores(A)** são os valores possíveis do atributo **A**, e **Sv** é o subconjunto de **S** onde **A** tem o valor **v**. O algoritmo escolhe o atributo com maior **IG** e repete o processo recursivamente até que um critério de parada seja atingido (ex: todos os exemplos pertencem à mesma classe ou a profundidade máxima é atingida).

2.2 Random Forest

O Random Forest (RF) é um método de aprendizado que constrói múltiplas árvores de decisão (uma "floresta") durante o treinamento e define a classe de saída como sendo a moda (votação majoritária) das classes de saída das árvores individuais.

O RF se destaca por sua robustez contra o sobreajuste (*overfitting*), um problema comum em árvores de decisão únicas como o ID3. Isso é alcançado através de duas técnicas principais:

1. **Bagging (Bootstrap Aggregating):** Cada árvore é treinada em uma amostra *bootstrap* diferente do conjunto de dados original (amostras selecionadas aleatoriamente com reposição).
2. **Subespaço Aleatório (Feature Randomness):** Ao dividir um nó, o algoritmo não procura pelo melhor atributo entre *todos* os disponíveis. Em vez disso, ele seleciona um subconjunto aleatório de atributos e busca o melhor apenas dentro desse subconjunto.

Essa dupla aleatoriedade (nas amostras e nos atributos) garante que as árvores da floresta sejam descorrelacionadas, tornando o voto coletivo mais preciso e estável.

3. Rede Neural

Diferente das abordagens simbólicas, as Redes Neurais Artificiais (RNAs) são modelos inspirados na estrutura do cérebro humano. Para este trabalho, foi utilizado um Multi-Layer Perceptron (MLP), que consiste em camadas de neurônios interconectados: uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

Cada neurônio em uma camada aplica uma transformação linear seguida por uma função de ativação não-linear. O treinamento ocorre pelo algoritmo de Backpropagation. A rede processa os dados de entrada, calcula o erro entre a saída prevista e a real usando uma função de perda, e então propaga esse erro de volta, ajustando os pesos de cada conexão através de um algoritmo de otimização para minimizar o erro. Para classificação, a camada de saída utiliza a função Softmax para converter as saídas em um vetor de probabilidades para cada classe.

4. Metodologia

O processo metodológico inicia-se com uma etapa de preparação dos dados, que inclui a limpeza e o tratamento de atributos. Dada a natureza distinta dos algoritmos, etapas como a normalização de atributos são aplicadas seletivamente.

Após o pré-processamento, o conjunto de dados é dividido em subconjuntos de treinamento e teste. O conjunto de treinamento é usado para treinar os modelos, enquanto o conjunto de teste, mantido separado, é usado para a avaliação final, permitindo medir a capacidade de generalização de cada modelo em dados nunca vistos. A seguir, detalha-se a abordagem conceitual da modelagem para cada algoritmo.

4.1 ID3

A modelagem com o ID3 visa a construção de uma estrutura de decisão única e explícita, priorizando a interpretabilidade. O conceito central é particionar recursivamente o espaço de atributos, utilizando o critério de Ganho de Informação (detalhado na Seção 2.1) para selecionar o atributo de divisão em cada nó.

O desafio metodológico do ID3 é encontrar um equilíbrio entre a pureza dos nós-folha (onde idealmente todas as amostras pertencem à mesma classe) e o risco de sobreajuste (*overfitting*). Um modelo que se ajusta perfeitamente aos dados de treinamento pode falhar ao generalizar. Para mitigar isso, a modelagem conceitual inclui a aplicação de restrições de

"poda", como a definição de uma profundidade máxima (max_depth), que impede o crescimento excessivo da árvore. Por ser um modelo baseado em regras de limiar, a normalização dos atributos de entrada é conceitualmente desnecessária.

4.2 Random Forest

A modelagem com Random Forest abandona o conceito de um único modelo ótimo em favor da "sabedoria da multidão" (*wisdom of the crowd*), uma abordagem de *conjunto*. O objetivo metodológico é construir uma "floresta" de múltiplos classificadores (árvores de decisão) e agregar seus resultados através de uma votação majoritária.

O ponto-chave desta metodologia é garantir que as árvores sejam **descorrelacionadas** (diversas). Isso é alcançado conceitualmente por duas técnicas de aleatorização: **Bagging**, onde cada árvore é treinada em uma subamostra com reposição dos dados de treinamento; e **Subespaço Aleatório**, onde, em cada divisão de nó, o algoritmo seleciona o melhor atributo não do conjunto total, mas de um subconjunto aleatório de atributos. A modelagem foca, portanto, em definir o número de árvores (n_trees) e o grau de aleatoriedade para otimizar o desempenho do *conjunto*.

4.3 Rede Neural

A modelagem da Rede Neural focou na implementação de um MLP. Diferentemente das abordagens simbólicas, o MLP exige uma normalização dos atributos de entrada para garantir uma convergência estável. Nesta implementação, os dados foram normalizados antes de serem apresentados à rede.

A arquitetura do MLP foi definida com três estágios:

1. Camada de Entrada: Com neurônios correspondentes ao número de atributos de entrada após o pré-processamento.
2. Camada Oculta: Foi utilizada uma única camada oculta contendo 64 neurônios. A função de ativação empregada nesta camada foi a ReLU (Rectified Linear Unit), escolhida por sua eficiência computacional e capacidade de mitigar o problema do desaparecimento do gradiente.
3. Camada de Saída: Utilizou a função Softmax, ideal para problemas de classificação multiclasse, pois converte as saídas da rede em um vetor de probabilidades.

O treinamento da rede consiste em um processo iterativo de otimização, onde os pesos são ajustados para minimizar uma função de perda que mede o erro do modelo. Utiliza-se um algoritmo de otimização baseado em gradiente para guiar esse ajuste, buscando encontrar os parâmetros que melhor generalizam a classificação para os dados. O controle desse processo depende de uma série de hiperparâmetros que definem como o modelo aprende ao longo do tempo.

5. Resultados

Após a definição metodológica e o treinamento dos modelos nos dados de treino, seus desempenhos de generalização foram avaliados no conjunto de testes. Os hiperparâmetros finais de cada modelo e a acurácia obtida são consolidados na Tabela 1.

Tabela 1: Comparativo de Desempenho dos Modelos no Conjunto de Teste

Modelo	**Parâmetros Principais**	**Acurácia**
ID3	max_depth = 10	92,66%
Random Forest	N_trees = 10, max_depth = 10	92,00%
Rede Neural	1 camada oculta (64 neurônios)	93,67%

A Rede Neural alcançou o melhor desempenho, indicando que sua arquitetura, mesmo com apenas uma camada oculta, foi a mais eficaz em capturar os padrões não-lineares e as interações complexas presentes nos dados dos sinais vitais.

O algoritmo ID3, embora conceitualmente mais simples, obteve um desempenho notavelmente competitivo. Isso sugere que o conjunto de dados possui regras de decisão relativamente claras que podem ser bem representadas por uma única árvore de decisão, tornando o ID3 uma alternativa viável onde a interpretabilidade do modelo é um requisito crítico.

O resultado mais surpreendente foi o do Random Forest, que apresentou um desempenho ligeiramente inferior ao da árvore de decisão única (ID3). Em teoria, um algoritmo como o RF é projetado para superar árvores únicas, reduzindo o sobreajuste e melhorando a generalização. Uma hipótese principal para este resultado é o baixo número de árvores utilizadas. Um número tão pequeno pode não ser suficiente para que o algoritmo convirja para uma solução robusta. É provável que o desempenho do RF aumentasse significativamente com um número maior de árvores, embora isso também aumentasse o custo computacional.

6. Conclusões

Este trabalho implementou e analisou comparativamente três algoritmos de Aprendizagem de Máquina - ID3, Random Forest e MLP - para um problema de classificação de Sinais Vitais, com foco em implementações "do zero" para aprofundar a compreensão de seus mecanismos.

A implementação "do zero" foi fundamental para compreender os desafios de cada abordagem, desde o critério de divisão de nós no ID3 até a complexa mecânica do *backpropagation* no MLP. A principal limitação observada foi o desempenho abaixo do esperado do Random Forest, provavelmente devido a uma parametrização subótima (baixo número de árvores), indicando que esses tipos de algoritmos exigem uma calibração cuidadosa para demonstrar sua superioridade.

Como trabalhos futuros, sugere-se: A realização de uma otimização de hiperparâmetros mais sistemática; A reavaliação do Random Forest com um número substancialmente maior de árvores e uma otimização maior voltada para o quesito de tempo de execução; A implementação de validação cruzada para uma estimativa mais robusta do

desempenho dos modelos, em vez de uma única divisão treino-teste; E a exploração de arquiteturas de MLP mais profundas.

7. Descrição de papéis e estimativa de horas

O desenvolvimento deste trabalho foi dividido equitativamente entre os membros da equipe, abrangendo desde a pesquisa teórica inicial e documentação até a implementação prática dos algoritmos de Aprendizagem de Máquina.

Ambos os autores, Alexandre Alberto Menon e Gabriel Rodrigues Estefanes, colaboraram ativamente em todas as fases do projeto. As responsabilidades foram compartilhadas da seguinte forma:

- Pesquisa e Fundamentação Teórica: Ambos os membros realizaram a pesquisa bibliográfica sobre os algoritmos ID3, Random Forest (Aprendizagem Simbólica) e Redes Neurais (MLP), bem como sobre as métricas de avaliação e pré-processamento de dados.
- Desenvolvimento e Codificação: A implementação dos três algoritmos ("do zero") foi um esforço conjunto. Ambos os autores participaram da codificação, depuração e otimização dos modelos.
- Metodologia e Experimentação: A definição da metodologia, incluindo a divisão dos dados, a parametrização dos modelos (como definição de profundidade máxima, número de árvores e arquitetura da rede) e a execução dos testes, foi realizada em conjunto.
- Análise de Resultados e Documentação: A análise comparativa dos resultados de acurácia, a discussão sobre os achados e a redação deste artigo (incluindo introdução, seções teóricas, metodologia, resultados e conclusões) foram igualmente divididas.

Estima-se que o tempo total dedicado ao projeto, incluindo pesquisa, codificação, experimentação e documentação, foi de 48 horas de trabalho combinado da equipe.

8. Referências

MENON, A.A.; ESTEFANES, G.R. **machine-learning**. 2025. Disponível em: https://github.com/AleMenon/machine-learning. Acesso em: 28 out. 2025.

RUSSELL, S.; NORVIG, P. Inteligência Artificial. 3. ed. Rio de Janeiro: Elsevier, 2013.

DATACAMP. *Decision Tree Classification in Python*. Disponível em: https://www.datacamp.com/pt/tutorial/decision-tree-classification-python. Acesso em: 27 out. 2025.

BAHETY, A.; KRUK, M. S.; O'LEARY, D. P. (2008). **GPU acceleration of the sum-product algorithm**. Relatório Técnico UMIACS-TR-2008-11 / CS-TR-4911, University of Maryland, College Park. Disponível em: https://www.cs.umd.edu/sites/default/files/scholarly_papers/Bahety_1.pdf. Acesso em: 27 out. 2025.

IACOMCAFÉ. *Implementando Random Forest em Python*. Disponível em: https://iacomcafe.com.br/implementando-random-forest-em-python/. Acesso em: 27 out. 2025.

IBM. [s.d.]. What Is Random Forest? Disponível em: https://www.ibm.com/think/topics/random-forest. Acesso em: 27 out. 2025.

DISCOVERYDATALAB. *Neural Networks Introduction*. Disponível em: https://discoverydatalab.ufop.br/blog/2024/10/10/neural_networks_introduction/. Acesso em: 27 out. 2025.

NATURE OF CODE. *Neural Networks*. Disponível em: https://natureofcode.com/neural-networks/. Acesso em: 27 out. 2025.