

## Trabalho Computacional AV2: Redes Neurais Artificiais

Marcos Antonio Felix - 1810449

Gil Melo Bandeira Torres - 1720537

Disciplina: Inteligência Artificial  
Computacional

Professor: Paulo Cirillo Souza Barbosa

**Resumo:** Este relatório apresenta o desenvolvimento e a avaliação de modelos de Redes Neurais Artificiais (RNAs) aplicados a três tarefas de aprendizado supervisionado: regressão, classificação binária e classificação multiclasse. Na Etapa I, foram utilizados os modelos ADALINE e MLP para prever a potência de um aerogerador com base na velocidade do vento, destacando a análise de underfitting e overfitting. A Etapa II tratou de um problema de classificação binária com dados não linearmente separáveis, utilizando Perceptron Simples e MLP. Já na Etapa III, abordou-se a classificação multiclasse de padrões clínicos da coluna vertebral com ADALINE Multiclasse e MLP, avaliando acurácia, sensibilidade e especificidade. Todas as etapas foram validadas com 100 rodadas de Monte Carlo, assegurando robustez estatística. Os resultados evidenciam a importância da escolha adequada de topologia, normalização dos dados, codificação das saídas e estratégias de validação para garantir desempenho confiável em tarefas de regressão e classificação.

**Palavras-chave:** Redes Neurais Artificiais, Regressão, Classificação Binária, Classificação Multiclasse, ADALINE, MLP, Validação Monte Carlo.

### I. INTRODUÇÃO

As Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados no funcionamento do sistema nervoso biológico, especialmente nos neurônios humanos. Um neurônio biológico recebe estímulos por seus dendritos, integra esses sinais no corpo celular e, ao ultrapassar um determinado limiar, propaga um impulso elétrico através do axônio. Esse mecanismo inspirou, em 1943, a criação do modelo de McCulloch-Pitts, considerado o ponto de partida para os estudos formais em redes neurais.

Desde então, diferentes arquiteturas foram desenvolvidas, indo de modelos lineares simples, como o Perceptron Simples e o ADALINE (ADaptive LInear NEuron), até redes profundas com múltiplas camadas, como o Perceptron de Múltiplas Camadas (MLP), que utiliza o algoritmo de retropropagação do erro (backpropagation) para ajustar seus

pesos. Essas redes são capazes de aprender padrões complexos a partir de dados e generalizar esse aprendizado para novas situações.

As RNAs têm se mostrado altamente eficazes em tarefas como regressão, classificação binária e multiclasse, reconhecimento de padrões, análise temporal e diagnóstico. Sua principal vantagem está na flexibilidade e capacidade de adaptação: por meio de ajustes iterativos nos pesos sinápticos, os modelos evoluem para minimizar erros sob métricas específicas, como o erro quadrático médio (MSE) ou acurácia.

Este relatório apresenta o desenvolvimento, treinamento e avaliação de modelos de RNA aplicados a três tarefas principais: regressão, classificação binária e classificação multiclasse, com validação baseada em simulações de Monte Carlo para garantir robustez estatística.

O trabalho foi dividido em duas grandes etapas:

### Etapa 1: Problemas Bidimensionais

**Regressão:** utilizados os modelos ADALINE e MLP para prever a potência gerada por um aerogerador a partir da velocidade do vento. A análise incluiu curvas de aprendizado, análise de topologias do MLP (underfitting e overfitting), além da avaliação de desempenho com Monte Carlo com 100 rodadas (iremos adotar como padrão para este e futuros modelos), usando a métrica MSE.

**Classificação Binária:** Utilizando o conjunto de dados "Spiral3d.csv", foram empregados o Perceptron Simples e o MLP para realizar a separação de duas classes em um problema não linear. Foram exploradas curvas de aprendizado, comparação entre topologias do MLP e validação com Monte Carlo, analisando acurácia, sensibilidade e especificidade.

### Etapa 2: Classificação Multiclasse

Nesta fase, abordou-se a classificação de padrões EMG (eletromiografia) de expressões faciais em cinco classes distintas, utilizando ADALINE Multiclasse e MLP Multiclasse. A complexidade aumentada exigiu topologias mais robustas e funções de ativação adequadas. As métricas utilizadas incluíram acurácia, matriz de confusão, sensibilidade e especificidade, com simulações de rodadas Monte Carlo setadas conforme mencionado.

A diversidade dos conjuntos de dados permitiu testar as redes em contextos simples (dados sintéticos) e complexos (dados reais multidimensionais), fornecendo uma avaliação abrangente da capacidade dos modelos implementados. Ao longo do relatório, são descritos os procedimentos metodológicos, análises visuais, desempenho quantitativo e reflexões sobre os impactos de diferentes configurações na eficácia das RNAs em aprendizado supervisionado.

## II. ETAPA I: REGRESSÃO

### A. Análise da problemática

A tarefa de regressão visa estabelecer um modelo que permita prever uma variável contínua de interesse a partir de uma ou mais variáveis independentes. No contexto deste trabalho, o problema de regressão foi formulado utilizando um conjunto de dados real referente à operação de um aerogerador, onde o objetivo é prever a potência gerada (variável dependente) a partir da velocidade do vento (variável independente).

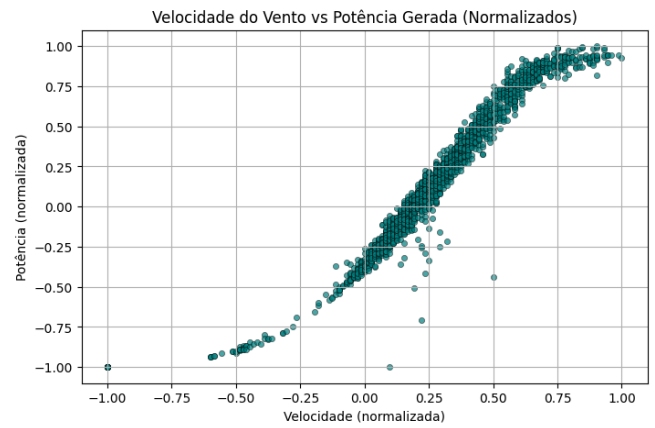
Este problema é representativo de situações práticas de engenharia e ciências aplicadas, onde relações físicas complexas entre variáveis podem apresentar não linearidades e ruídos, demandando modelos capazes de capturar tais características para realizar previsões precisas e confiáveis.

### B. Organização e preparação dos dados

O conjunto de dados utilizado é composto por 2250 amostras, contendo uma única variável preditora e a respectiva variável de saída. Os dados foram submetidos a um processo de normalização para garantir que as variáveis estejam em escala compatível, facilitando o processo de aprendizado dos modelos e evitando que diferenças de escala influenciem indevidamente os parâmetros ajustados.

Além disso, foi realizada uma análise exploratória inicial, com gráficos de dispersão para observar a distribuição dos dados e verificar possíveis padrões, tendências e variabilidade inerente, essenciais para direcionar as escolhas dos modelos e suas configurações.

Os dados foram organizados e normalizados para entrada nos modelos. Visualizações iniciais foram feitas para entender a distribuição dos pontos (Figura 1).

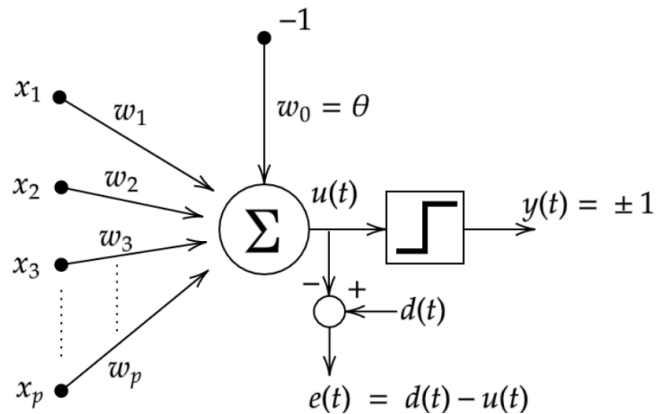


**Figura 1:** Gráfico de dispersão dos dados de velocidade do vento vs potência do aerogerador.

### C. Modelos Implementados

Os modelos utilizados são:

**ADALINE (ADaptive LINEar NEuron):** Modelo linear que ajusta pesos utilizando o método do gradiente descendente minimizando o erro quadrático médio. É uma rede simples, adequada para problemas cuja relação entre entrada e saída seja aproximadamente linear.



**Figura 2:** Diagrama esquemático de um modelo de neurônio ADALINE

O ADALINE possui entradas  $x_1, x_2, \dots, x_p$ , pesos associados  $w_1, w_2, \dots, w_p$ , e um termo de bias  $w_0 = \theta$ , conforme o diagrama abaixo:

O somatório ponderado  $u(t) = w^T x(t)$  é passado diretamente para a saída no caso de regressão.

O erro é calculado como:

$$e(t) = d(t) - u(t)$$

A função de custo adotada é o Erro Quadrático Médio (EQM), definida como:

$$J(w) = \frac{1}{2N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

O processo de treinamento do ADALINE é realizado com o algoritmo de gradiente descendente, utilizando o seguinte pseudocódigo:

**Algorithm 3:** Pseudocódigo para ajuste (fase de treinamento), do ADALINE.

```

1: Definir o valor de  $\eta$ , número máximo de épocas e precisão ( $\epsilon$ ).
2: Inicializar o vetor de pesos  $\mathbf{w}(t)$  com valores nulos ou aleatórios.
3: Iniciar o contador de épocas ( $epoch \leftarrow 0$ )
4: repeat
5:    $EQM_{anterior} \leftarrow EQM(\mathbf{x}, d, \mathbf{w})$ 
6:   for todas as  $N$  amostras de treinamento do
7:      $u(t) \leftarrow \mathbf{w}^T(t)\mathbf{x}(t)$ 
8:      $\mathbf{w}(t+1) \leftarrow \mathbf{w}(t) + \eta(d(t) - u(t))\mathbf{x}(t)$ 
9:   end for
10:   $epoch \leftarrow epoch + 1$ 
11:   $EQM_{atual} \leftarrow EQM(\mathbf{x}, d, \mathbf{w})$ 
12: until  $|EQM_{atual} - EQM_{anterior}| \leq \epsilon$  OU Número máximo de épocas atingido

```

A função de custo é avaliada com base na seguinte rotina, que computa o EQM para todas as amostras de treinamento:

**Algorithm 4:** Algoritmo para cálculo do EQM.

```

1:  $EQM \leftarrow 0$ 
2: for todas as amostras de treinamento do
3:    $u(t) \leftarrow \mathbf{w}^T(t)\mathbf{x}(t)$ 
4:    $EQM \leftarrow EQM + (d(t) - u(t))^2$ 
5: end for
6:  $EQM \leftarrow \frac{EQM}{2N}$ 

```

Após o treinamento, a classificação de novas amostras pode ser feita com base na saída linear, seguida de uma função de ativação degrau (caso de classificação binária):

**Algorithm 5:** Pseudocódigo para operação (fase de teste), do ADALINE.

```

1: Obter uma amostra ( $\mathbf{x}_{desconhecido}$ ) a ser classificada
2: Utilizar o vetor  $\mathbf{w}$  já estimado
3: Realizar as seguintes operações:
4:  $u \leftarrow \mathbf{w}^T \mathbf{x}_{desconhecido}$ 
5:  $y(t) \leftarrow \text{signal}(u(t))$ 
6: if  $y = -1$  then
7:   amostra pertence a classe A
8: else
9:   amostra pertence a classe B
10: end if

```

A equação usada para atualização dos pesos durante o treinamento é:

$$w(t+1) = w(t) + \eta \cdot (d(t) - u(t)) \cdot x(t)$$

Onde:

→  $\eta$  é a taxa de aprendizado

→  $d(t)$  é a saída desejada

→  $u(t) = \mathbf{w}^T \mathbf{x}(t)u(t) = \mathbf{w}^T \mathbf{x}(t)u(t) = \mathbf{w}^T \mathbf{x}(t)$  é a saída do somador linear.

Convergência: O treinamento é encerrado quando a variação do erro quadrático médio entre épocas consecutivas é menor que um valor de tolerância  $\epsilon$  (varepsilon), ou quando se atinge o número máximo de épocas:

$$|EQM(t) - EQM(t-1)| \leq \epsilon$$

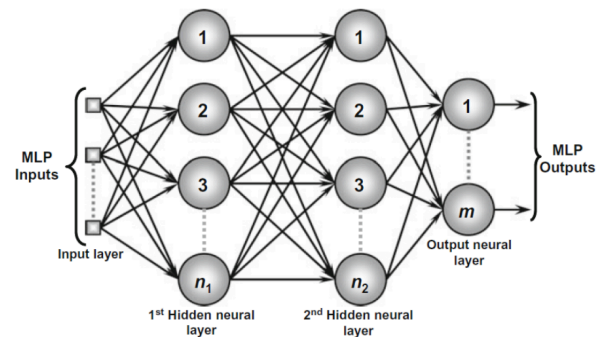
**Perceptron de Múltiplas Camadas (MLP):** É uma rede neural do tipo feedforward composta por múltiplas camadas de neurônios organizadas sequencialmente, permitindo a modelagem de relações não lineares entre as variáveis.

Diferentemente de modelos lineares como o ADALINE, MP e Perceptron simples, o MLP aplica uma função de ativação não linear em cada neurônio, o que lhe confere maior capacidade de generalização e aproximação de funções complexas.

Neste trabalho, foi utilizada a função de ativação tangente hiperbólica, que é contínua, diferenciável e apresenta saída no intervalo  $[-1, 1]$  definida como:

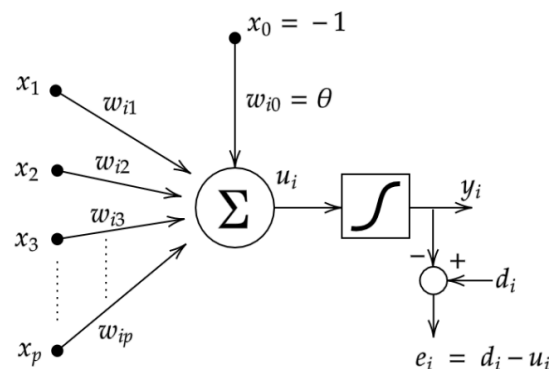
$$\phi(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

A estrutura do MLP é dividida em três tipos de camadas: a camada de entrada, que recebe os dados brutos  $\mathbf{x} = [x_1, x_2, \dots, x_p]$ ; uma ou mais camadas ocultas, responsáveis por aplicar transformações intermediárias não lineares; e a camada de saída, que gera a predição final. A Figura 3 apresenta uma arquitetura genérica de um MLP com duas camadas ocultas:



**Figura 3:** Arquitetura geral do MLP

Durante a fase de propagação direta (forward pass), cada neurônio soma suas entradas ponderadas por pesos e adiciona um viés. Esse somatório é passado por uma função de ativação. A estrutura interna de um neurônio com ativação não linear está representada na Figura 2:



**Figura 4:** Estrutura de um neurônio com função de ativação

Nesse modelo, o valor da ativação  $u_i$  é computado por:

$$u_i = \sum_{j=1}^p w_{ij}x_j + w_{i0} \cdot x_0 \quad (x_0 = -1)$$

A saída  $y_i$  do neurônio é então:

$$y_i = \phi(u_i)$$

e o erro associado é:

$$e_i = d_i - u_i$$

O comportamento computacional da fase de propagação é formalizado no Algoritmo 8, que apresenta o pseudocódigo para o forward do MLP:

**Algorithm 8:** Pseudocódigo para o Forward da rede MLP.

```

1: Receber a amostra  $x_{amostra} \in \mathbb{R}^{(p+1) \times 1}$ .
2:  $j \leftarrow 0$ 
3: for cada matriz de peso  $W$  em cada uma das  $L + 1$  camadas. do
4:   if  $j == 0$  then
5:      $i[j] \leftarrow W[j] \cdot x_{amostra}$ 
6:      $y[j] \leftarrow g(i[j])$ 
7:   else
8:      $y_{bias} \leftarrow y[j - 1]$  com adição de  $-1$  na primeira posição do vetor.
9:      $i[j] \leftarrow W[j] \cdot y_{bias}$ 
10:     $y[j] \leftarrow g(i[j])$ 
11:   end if
12:    $j \leftarrow j + 1$ 
13: end for

```

Após a propagação, inicia-se o processo de retropropagação do erro (backpropagation), em que os gradientes do erro são calculados e os pesos da rede são atualizados utilizando a técnica do gradiente descendente. A derivada da tangente hiperbólica é:

$$\phi'(u) = 1 - \tanh^2(u)$$

A retropropagação é detalhada no Algoritmo 9, a seguir:

**Algorithm 9:** Pseudocódigo para o BackWard da rede MLP.

```

1: Receber a amostra  $x_{amostra} \in \mathbb{R}^{(p+1) \times 1}$  e seu rótulo  $d \in \mathbb{R}^{c \times 1}$ .
2:  $j \leftarrow$  Quantidade de matrizes  $W - 1$ .
3: while  $j \geq 0$  do
4:   if  $j + 1 ==$  Quantidade de matrizes  $W$ , then
5:      $\delta[j] \leftarrow g'(i[j]) \circ (d - y[j])$ .
6:      $y_{bias} \leftarrow y[j - 1]$  com adição de  $-1$  na primeira posição do vetor.
7:      $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes y_{bias})$ 
8:   else if  $j == 0$  then
9:      $W_b[j + 1]$  Recebe a matriz  $W[j + 1]$  transposta sem a coluna que multiplica pelos limiares de ativação.
10:     $\delta[j] \leftarrow g'(i[j]) \circ (W_b[j + 1] \cdot \delta[j + 1])$ .
11:     $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes x_{amostra})$ 
12:   else
13:      $W_b[j + 1]$  Recebe a matriz  $W[j + 1]$  transposta sem a coluna que multiplica pelos limiares de ativação.
14:      $\delta[j] \leftarrow g'(i[j]) \circ (W_b[j + 1] \cdot \delta[j + 1])$ .
15:      $y_{bias} \leftarrow y[j - 1]$  com adição de  $-1$  na primeira posição do vetor.
16:      $W[j] \leftarrow W[j] + \eta(\delta[j] \otimes y_{bias})$ 
17:   end if
18:    $j \leftarrow j - 1$ 
19: end while

```

A função de custo do MLP é o Erro Quadrático Médio (EQM), assim como no ADALINE. No entanto, no MLP o erro é calculado após a ativação não linear, o que afeta o cálculo dos gradientes durante a retropropagação.

**Algorithm 10:** Pseudocódigo para o Erro Quadrático Médio.

```

1:  $EQM \leftarrow 0$ 
2: for Cada amostra em  $X_{treino}$  do
3:    $x_{amostra} \leftarrow N$ -ésima amostra de  $X_{treino}$ .
4:   Forward( $x_{amostra}$ )
5:    $d \leftarrow N$ -ésimo rótulo de  $X_{treino}$ .
6:    $EQI \leftarrow 0$ 
7:    $j \leftarrow 0$ 
8:   for Cada neurônio na camada de saída do
9:      $EQI \leftarrow EQI + (d[j] - y[QTD\_L - 1][j])^2$ 
10:     $j \leftarrow j + 1$ 
11:   end for
12:    $EQM \leftarrow EQM + EQI$ 
13: end for
14:  $EQM \leftarrow EQM / (2 * QtdAmostrasTreino)$ 

```

Para a fase de operação (teste) da rede, realiza-se apenas a propagação direta da entrada até a saída, e a predição é atribuída à classe correspondente ao maior valor de ativação na saída. Essa fase está descrita no pseudocódigo:

**Algorithm 11:** Pseudocódigo para fase de operação (teste) da rede MLP.

```

1: for Cada amostra em  $X_{teste}$  do
2:    $x_{amostra} \leftarrow N$ -ésima amostra de  $X_{teste}$ .
3:   Forward( $x_{amostra}$ )
4:   Realizar a atribuição desta amostra para a classe cujo índice do vetor de saída da rede MLP possuir maior valor.
5: end for

```

A complexidade do MLP pode ser quantificada pelo número total de parâmetros treináveis (pesos + bias). Para uma rede com  $p$  entradas,  $q_1$  neurônios na primeira camada oculta,  $q_2$  na segunda, e  $m$  saídas, o total de parâmetros é:

$$Z = (p + 1)q_1 + (q_1 + 1)q_2 + (q_2 + 1)m$$

Esse valor influencia diretamente a capacidade do modelo e sua tendência a sofrer underfitting (subajuste) ou overfitting (sobreajuste).

O critério de parada do MLP segue o princípio do modelo anterior, sendo baseado na estabilização do erro entre épocas consecutivas ou no atingimento do número máximo de épocas. O treinamento é encerrado quando a variação do EQM é inferior a um limiar  $\epsilon$  (varepsilon), evitando ajustes desnecessários nos pesos.

#### D. Critérios de treinamento e validação

O treinamento considerou taxa de aprendizado otimizada empiricamente, número máximo de épocas e critérios de convergência baseados na minimização do erro médio quadrático (MSE). O uso de funções de ativação não-lineares no MLP demandou atenção ao problema de saturação de gradiente, o qual foi mitigado por escolha cuidadosa da função e da taxa de aprendizado.

As métricas avaliadas para a tarefa de regressão incluíram:

- Média do erro quadrático médio (MSE) nas rodadas de teste
- Desvio-padrão do MSE, para avaliar a variabilidade do desempenho
- Maior e menor valores do MSE observados

Os resultados mostraram que o modelo ADALINE, apesar de sua simplicidade e eficiência computacional, apresentou

limitações em capturar relações não lineares presentes no problema, refletindo-se em um valor de erro médio quadrático de 0.05587614797432918 como pode ser observado na plotagem a seguir (Figura 5).

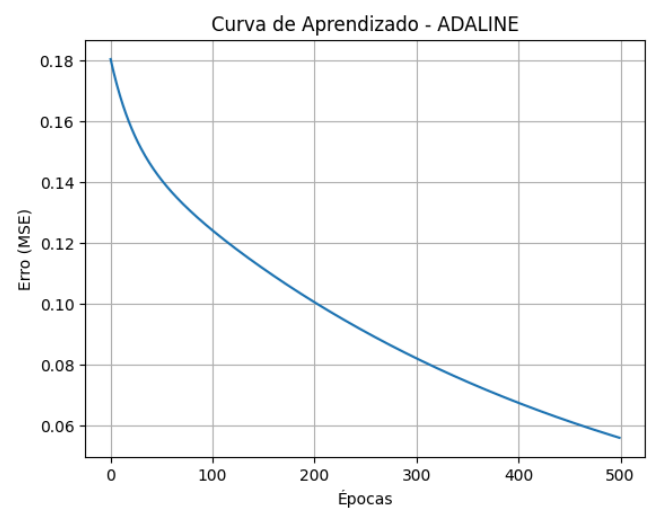


Figura 5: Curva de aprendizado no modelo de treinamento ADALINE.

Já o MLP demonstrou maior flexibilidade, com topologias intermediárias entregando o melhor equilíbrio entre ajuste e generalização. A MLP com 1 neurônio (Figura 6) apresenta um caso típico de underfitting: o modelo é simples demais, resultando em erro elevado tanto no treino quanto no teste. Ele não consegue capturar a complexidade dos dados, e sua capacidade de generalização é limitada.

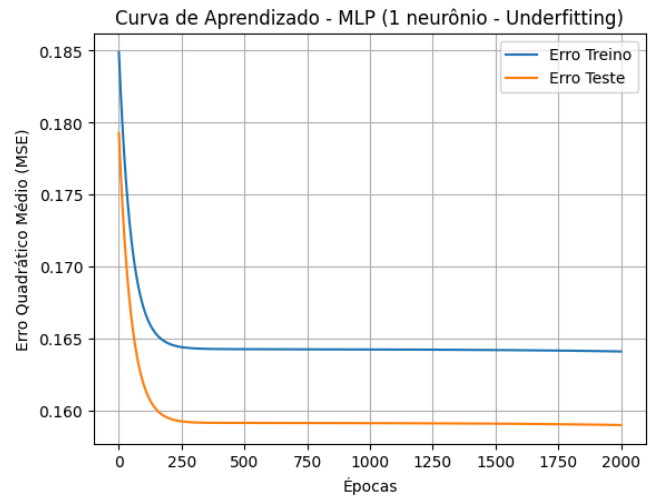


Figura 6: Curva de aprendizado no modelo MLP (para 1 neurônio).

Já a MLP com 50 neurônios (Figura 7), embora apresente potencial de overfitting, alcança desempenho significativamente melhor. O erro de treino é baixo e o erro de teste estabiliza rapidamente, indicando que o modelo possui capacidade suficiente para aprender e generalizar com mais eficiência.

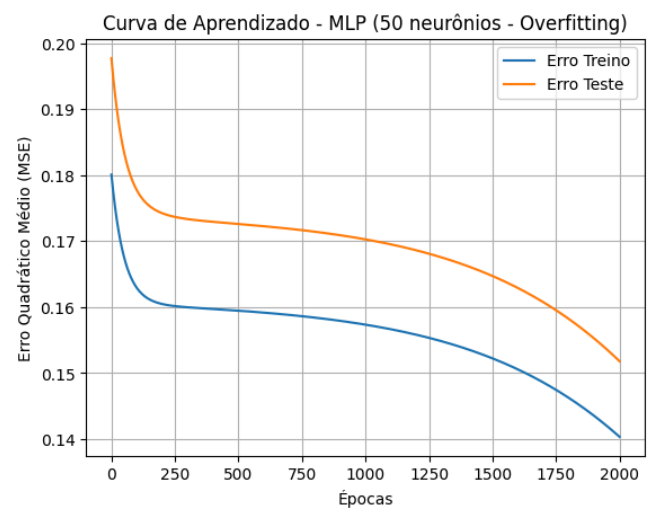


Figura 7: Curva de aprendizado no modelo MLP (para 50 neurônios).

As curvas de aprendizado geradas para diferentes topologias evidenciaram claramente esses fenômenos, possibilitando a identificação visual dos pontos onde a complexidade da rede é insuficiente ou excessiva (Tabela 1).

	Treino	Teste
Underfitting	0.16411	0.15900
Overfitting	0.14030	0.15175

Tabela 1: Análise das Curvas de Aprendizado – Underfitting vs Overfitting.

E. Validação de Monte Carlo

Para validação e avaliação dos modelos, foi adotada a técnica de **simulações Monte Carlo**, realizando 100 rodadas em que os dados são aleatoriamente particionados em 80% para treinamento e 20% para teste a cada rodada. Esta abordagem assegura que as métricas obtidas reflitam o desempenho médio dos modelos em diferentes configurações de dados, proporcionando maior robustez estatística.

A análise Monte Carlo foi repetida para duas topologias diferentes do modelo MLP: uma com apenas 1 neurônio (subdimensionada, underfitting) e outra com 50 neurônios (superdimensionada, overfitting). Os resultados numéricos refletem como o aumento da complexidade influencia o erro de generalização e a variabilidade do desempenho entre diferentes divisões dos dados (Tabela 2).



	ADALINE	MLP
Média	0.17255	0.16330
Desvio padrão	0.00874	0.00904
Máximo	0.19263	0.18571
Mínimo	0.15170	0.13961

Tabela 2: Análise das Curvas de Aprendizado – Underfitting vs Overfitting.

F. Resultados obtidos e análise

Os resultados quantitativos indicam que o MLP supera o ADALINE em termos de MSE médio e estabilidade, com desvio padrão inferior, evidenciando maior capacidade de modelagem da relação não-linear. Curvas de aprendizado (Figura 8) evidenciam o overfitting para arquiteturas muito complexas e underfitting para configurações reduzidas, confirmando a importância do ajuste de hiperparâmetros. O ADALINE, embora mais simples e eficiente, mostrou limitações para capturar a dinâmica do sistema. Esses achados ressaltam a necessidade do balanceamento adequado entre simplicidade e capacidade do modelo para tarefas de regressão não-lineares.

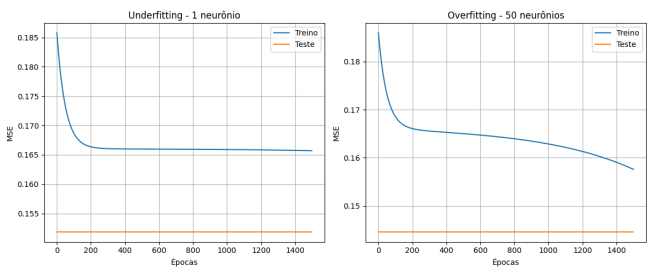


Figura 8: Curva de aprendizado no modelo MLP (para 50 neurônios).

Os gráficos acima mostram claramente a diferença entre underfitting e overfitting. A topologia com 1 neurônio não consegue reduzir o erro, enquanto a de 50 neurônios apresenta erro de treino muito baixo, mas mantém erro de teste mais alto — evidenciando o overfitting.

G. Conclusão da tarefa de regressão

A análise realizada demonstrou que a escolha adequada do modelo e sua configuração são cruciais para o sucesso em tarefas de regressão com dados reais e não lineares. Enquanto o modelo ADALINE, por sua natureza linear, apresentou limitações evidentes ao lidar com as complexidades da relação entre velocidade do vento e potência gerada, o Perceptron de Múltiplas Camadas (MLP) mostrou-se amplamente superior, graças à sua capacidade de capturar relações não lineares e ajustar-se a diferentes padrões dos dados.

A exploração de diferentes topologias no MLP evidenciou claramente os efeitos do underfitting e overfitting: redes subdimensionadas não conseguem modelar adequadamente os dados, enquanto redes superdimensionadas se tornam excessivamente especializadas no conjunto de treinamento, prejudicando sua generalização.

O uso do procedimento de validação via Monte Carlo, com múltiplas partições aleatórias de treino e teste, garantiu a robustez estatística dos resultados, permitindo estimar com maior confiança o desempenho esperado dos modelos em situações reais.

Portanto, conclui-se que redes neurais não lineares, quando configuradas com uma arquitetura adequada e validadas de forma rigorosa, são ferramentas poderosas para problemas de regressão complexos. Este trabalho reforça a importância de um processo sistemático de pré-processamento dos dados, escolha criteriosa de modelos e avaliação estatística para o desenvolvimento de soluções confiáveis em aplicações práticas.

III. ETAPA I: CLASSIFICAÇÃO

A. Análise da problemática

Os dados apresentam um desafio típico de classificação binária, em que é necessário distinguir entre duas categorias distintas a partir de três variáveis independentes. A complexidade principal está no fato de que as classes não são separáveis por uma fronteira linear simples, exigindo que o modelo capture relações não lineares e padrões complexos nas variáveis de entrada para realizar uma classificação precisa.

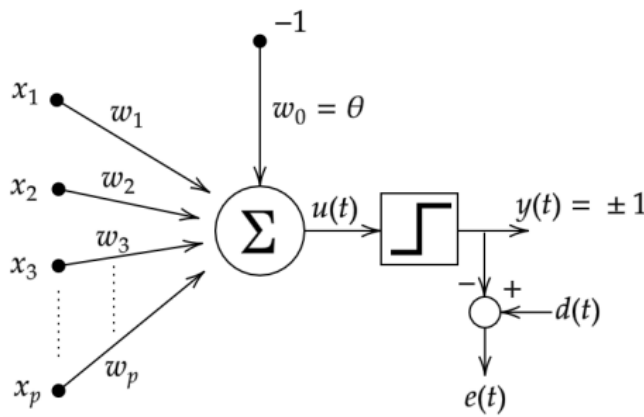
Este cenário demanda modelos com capacidade para aprender representações flexíveis e adaptativas das fronteiras de decisão, evitando tanto o underfitting — quando o modelo é muito simples e não captura as nuances dos dados — quanto o overfitting — quando o modelo se ajusta excessivamente ao conjunto de treinamento e perde capacidade de generalização.

Além disso, para garantir uma avaliação justa e efetiva, o desempenho dos modelos deve ser analisado por métricas que vão além da acurácia, incluindo sensibilidade e especificidade, de modo a garantir que ambas as classes sejam adequadamente reconhecidas, especialmente se houver algum desequilíbrio ou custo diferenciado associado a erros em cada classe.

B. Modelos Implementados

Os modelos utilizados são:

**Perceptron Simples:** É um dos primeiros modelos de redes neurais artificiais, proposto em 1958 por Frank Rosenblatt. Ele é inspirado no neurônio biológico de McCulloch-Pitts e funciona como um classificador linear para problemas de classificação binária.



O perceptron simples consiste em:

- Um vetor de entradas  $x = [x_0, x_1, \dots, x_p]^T$ , onde  $x_0 = -1$  é um termo artificial para o bias (limiar).
- Um vetor de pesos  $w = [w_0, w_1, \dots, w_p]^T$ , correspondendo aos pesos sinápticos do neurônio.
- Uma função de ativação do tipo degrau (função sinal), que gera a saída  $y(t)$  a partir do somatório ponderado das entradas.

Matematicamente, a saída do perceptron é dada por:

$$y(t) = \text{sign}(w^T x) = \begin{cases} +1, & \text{se } w^T x \geq 0 \\ -1, & \text{se } w^T x < 0 \end{cases}$$

O treinamento do perceptron é feito pela **regra de aprendizagem do perceptron**, que ajusta os pesos iterativamente para reduzir o erro entre a saída desejada  $d(t)$  e a saída atual  $y(t)$ . A atualização dos pesos é feita segundo:

$$w(t+1) = w(t) + \eta \cdot e(t) \cdot x(t)$$

onde:

- $e(t) = d(t) - y(t)$  é o erro
- $\eta$  é a taxa de aprendizagem (um valor pequeno e positivo, normalmente entre 0 e 1).

O algoritmo continua atualizando os pesos até que não haja mais erros na classificação do conjunto de treinamento ou até atingir um número máximo de iterações.

A seguir, apresenta-se o pseudocódigo da fase de treinamento, conforme a Regra de Aprendizagem do Perceptron:

Algorithm 1: Pseudocódigo para ajuste (fase de treinamento), do perceptron.

```

1: Início ( $t = 0$ )
2: Definir o valor de  $\eta$  entre 0 e 1.
3: Inicializar o vetor de pesos  $w(t)$  com valores nulos ou aleatórios.
4: ERRO  $\leftarrow$  EXISTE
5: while ERRO == 'EXISTENTE' do
6:   ERRO  $\leftarrow$  'INEXISTE'
7:   for Todas amostras em  $x$  do
8:      $u(t) \leftarrow w^T(t)x(t)$ 
9:      $y(t) \leftarrow \text{signal}(u(t))$ 
10:     $w(t+1) \leftarrow w(t) + \eta(d(t) - y(t))x(t)$ 
11:    if  $d(t) \neq y(t)$  then
12:      ERRO  $\leftarrow$  'EXISTENTE'
13:    end if
14:  end for
15:   $t \leftarrow t+1$ 
16: end while
17: FIM TREINAMENTO

```

Após o treinamento, o vetor de pesos estimado é utilizado para classificar novas amostras desconhecidas. Para tal, o perceptron calcula a ativação do neurônio com base nos pesos aprendidos e aplica a função de ativação do tipo degrau para decidir a qual classe a amostra pertence, conforme o pseudocódigo apresentado a seguir:

Algorithm 2: Pseudocódigo para operação (fase de teste), do perceptron.

```

1: Obter uma amostra ( $x_{\text{desconhecida}}$ ) a ser classificada
2: Utilizar o vetor  $w$  já estimado
3: Realizar as seguintes operações:
4:  $u \leftarrow w^T x_{\text{desconhecida}}$ 
5:  $y(t) \leftarrow \text{signal}(u(t))$ 
6: if  $y = -1$  then
7:   amostra pertence a classe A
8: else
9:   amostra pertence a classe B
10: end if

```

**Perceptron de Múltiplas Camadas (MLP):** Já explicado na parte II.

### C. Organização e Preparação dos Dados

Os dados foram organizados e normalizados, garantindo compatibilidade com as funções de ativação das RNAs. Visualizações iniciais (Figura 9) permitiram confirmar a separabilidade não linear das classes, justificando o uso de modelos mais complexos que o perceptron simples.

### Distribuição das Classes no Espaço 3D

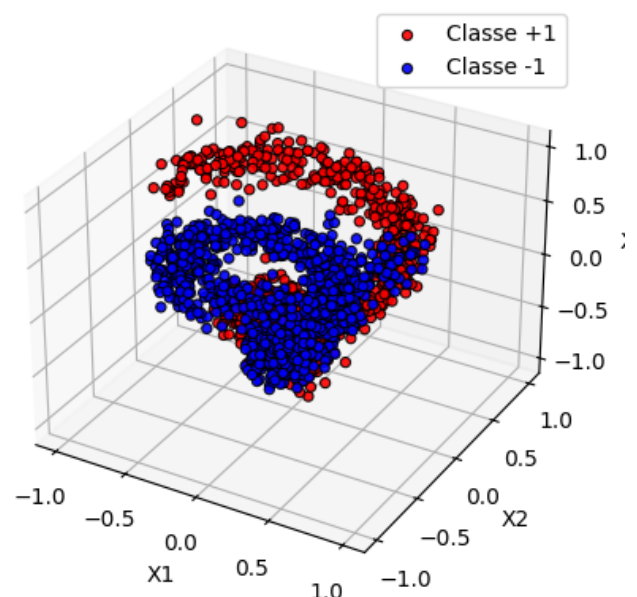


Figura 9: Gráfico de normalização.

D. Resultados Obtidos e Análise

O gráfico (Figura 10) mostra a curva de aprendizado do Perceptron Simples, indicando a proporção de erros ao longo das épocas. As métricas revelam que, devido à não-linearidade da separação entre classes, o modelo teve dificuldades para aprender, especialmente na classe +1. Isso era esperado, pois o Perceptron é um classificador linear e a base Spiral3d exige modelos não lineares.

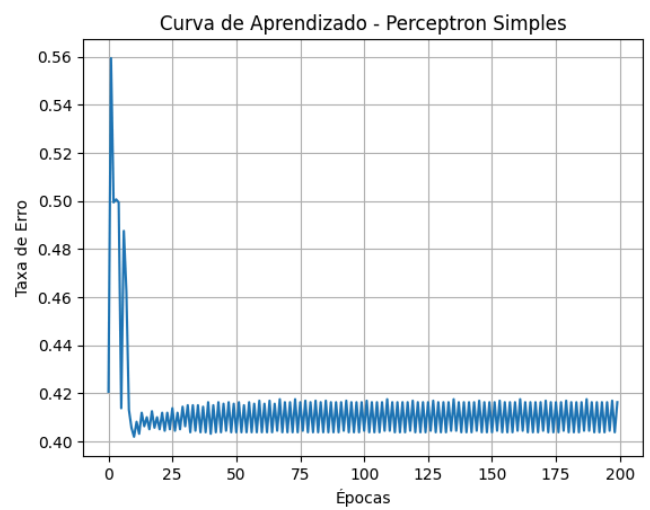


Figura 10: Curva de aprendizado do Perceptron Simples.

A sensibilidade elevada indica que o modelo identifica muito bem os exemplos da classe positiva, enquanto a baixa especificidade evidencia dificuldade em reconhecer corretamente a classe negativa.

Métrica	Valor
Acurácia	0.6050
Sensibilidade (Classe +1)	0.9899
Especificidade (Classe -1)	0.2239

Tabela 3: Métricas de desempenho do Perceptron Simples.

Os resultados mostram que o modelo é muito eficaz em detectar a classe positiva, como evidenciado pela alta sensibilidade (0,9899). No entanto, o modelo apresenta baixa especificidade (0,2239), indicando que muitos exemplos da classe negativa foram classificados incorretamente, refletido no elevado número de falsos positivos (156). Essa assimetria pode indicar um viés do modelo ou desequilíbrio nos dados, sugerindo a necessidade de ajustes para melhorar o reconhecimento da classe negativa.

	Predito +1	Predito -1
Real +1	197 (TP)	2 (FN)
Real -1	156 (FP)	45 (TN)

Tabela 4: Matriz de Confusão do Perceptron Simples. A curva de aprendizado da MLP (Figura 11) mostra uma redução consistente no erro quadrático médio ao longo das épocas. As métricas de avaliação mostram uma melhora substancial em relação ao Perceptron Simples, com maior sensibilidade e especificidade, o que confirma a superioridade dos modelos não lineares para este tipo de problema.

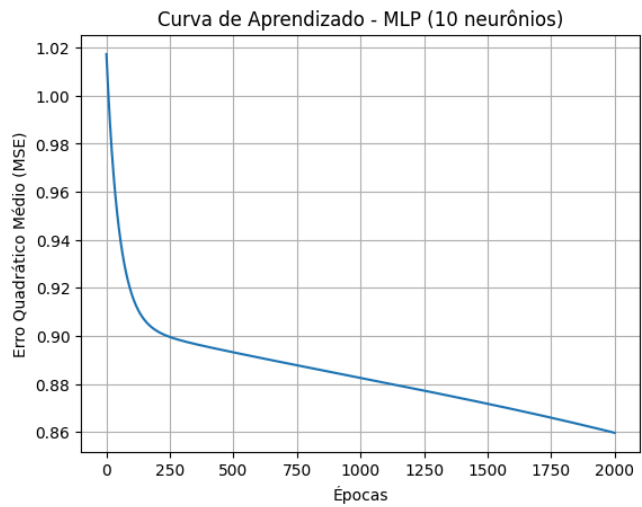


Figura 11: Curva de aprendizado do MLP.

O MLP apresenta equilíbrio melhor entre sensibilidade e especificidade, indicando desempenho mais homogêneo na identificação das duas classes.

Métrica	Valor
Acurácia	0.6325
Sensibilidade (Classe +1)	0.6734
Especificidade (Classe -1)	0.5920

Tabela 5: Métricas de desempenho do MLP.

Neste cenário, observa-se uma redução na sensibilidade (0,6734), indicando que o modelo deixa de identificar corretamente mais exemplos da classe positiva em comparação ao primeiro caso. Por outro lado, a especificidade aumentou para 0,5920, reduzindo a taxa de falsos positivos e melhorando a identificação da classe negativa. A acurácia global também aumentou levemente, sugerindo um melhor equilíbrio na classificação geral entre as duas classes.



	Predito +1	Predito -1
Real +1	134 (TP)	65 (FN)
Real -1	82 (FP)	119 (TN)

Tabela 6: Matriz de Confusão do MLP.

E. Resultados Obtidos

Os resultados quantitativos obtidos a partir de 100 rodadas de validação Monte Carlo para os modelos Perceptron Simples e Perceptron de Múltiplas Camadas (MLP) com 10 neurônios foram analisados considerando as métricas de acurácia, sensibilidade e especificidade.

**Perceptron Simples** apresentou uma acurácia média de 58,30% (DP = 2,08%), com valores máximos e mínimos de 64,00% e 53,75%, respectivamente. A sensibilidade média foi de 53,55% (DP bastante elevado de 41,09%), indicando alta variabilidade na capacidade do modelo em identificar corretamente a classe positiva, variando de 13,02% a 100%. A especificidade média ficou em 63,34% (DP de 39,86%), também com ampla dispersão (mínimo de 12,32% e máximo de 100%), evidenciando que o desempenho na identificação da classe negativa foi instável ao longo das rodadas.

Por sua vez, o **MLP** com 10 neurônios apresentou uma melhora consistente no desempenho, alcançando uma acurácia média de 62,26% (DP = 3,92%), com variação entre 46,50% e 68,75%. A sensibilidade média foi significativamente maior e mais estável, atingindo 62,90% com um desvio padrão de 5,55%, e valores extremos de 41,12% a 72,28%. A especificidade do MLP também apresentou melhor equilíbrio, com média de 61,75% e DP de 6,83%, variando entre 34,29% e 72,77%.

Modelo	Métrica	Média	Desvio Padrão	Máximo	Mínimo
Perceptron Simples	Acurácia	0.5830	0.0208	0.6400	0.5375
	Sensibilidade	0.5355	0.4109	1	0.1302
	Especificidade	0.6334	0.3986	1	0.1232
MLP (10 neurônios)	Acurácia	0.6226	0.0392	0.6875	0.4650
	Sensibilidade	0.6290	0.0555	0.7228	0.4112
	Especificidade	0.6175	0.0683	0.7277	0.3429

Tabela 7: Resultados obtidos após 100 rodadas.

Os boxplots gerados para estas métricas corroboram essa análise, mostrando que o MLP não apenas apresenta melhores médias, mas também uma distribuição dos resultados mais concentrada, especialmente para sensibilidade e especificidade. Isso indica que o MLP é mais confiável e consistente na distinção entre as classes, enquanto o Perceptron Simples sofre com alta variabilidade, podendo ter desempenho muito bom em algumas rodadas, mas muito ruim em outras.

- Boxplot de Acurácia (Figura 12).

O Perceptron apresentou mediana superior.

MLP mostrou maior variação, com muitos outliers — reflexo de instabilidade nas divisões.

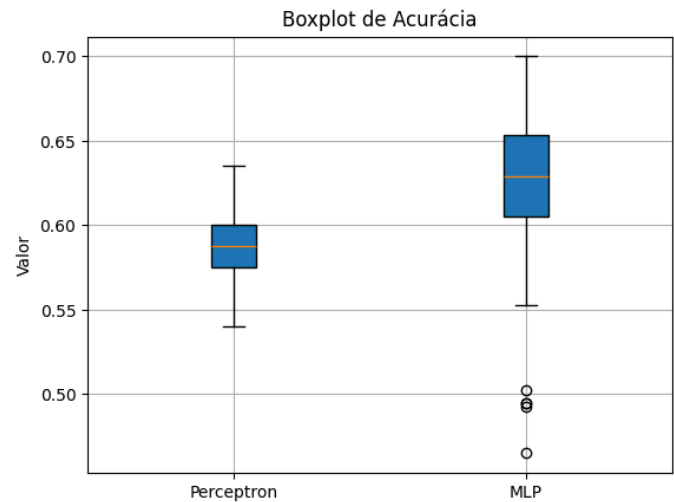


Figura 12: Boxplot de Acurácia.

- Boxplot de Sensibilidade (Figura 13).

MLP teve média mais alta, mas com dispersão muito maior.

Perceptron foi mais estável, embora mais limitado.

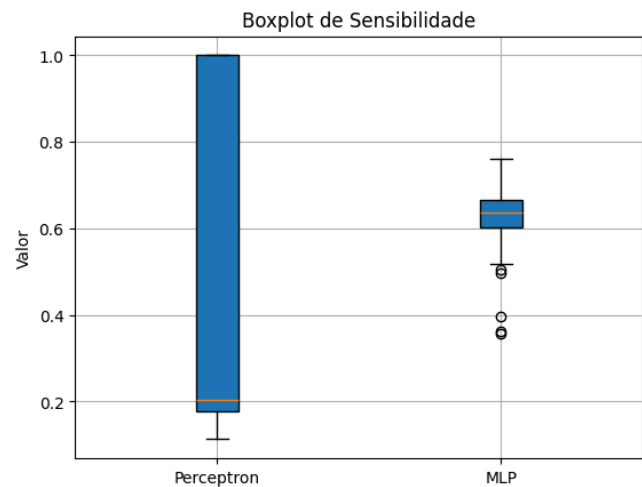
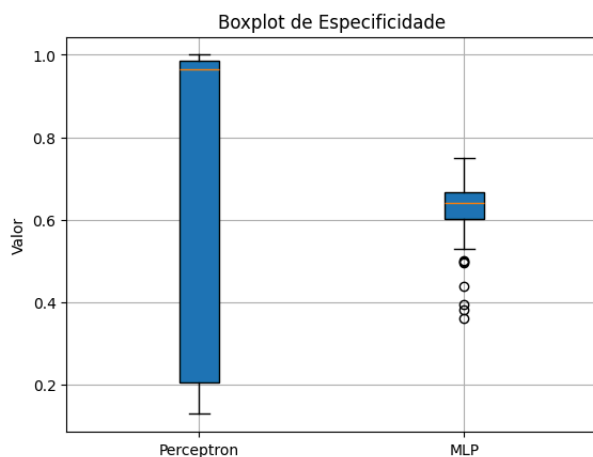


Figura 13: Boxplot de Sensibilidade.

- Boxplot de Especificidade (Figura 14).

Perceptron teve distribuição mais concentrada.

MLP mostrou extremos, com alguns testes em que falhou completamente (especificidade = 0).



**Figura 14:** Boxplot de Especificidade.

Essa superioridade do MLP está associada à sua capacidade de modelar fronteiras de decisão não lineares, característica essencial para os dados em questão, que apresentam separabilidade complexa. Já o Perceptron Simples, limitado a fronteiras lineares, não consegue capturar adequadamente essa complexidade, refletindo na maior dispersão e menor desempenho médio.

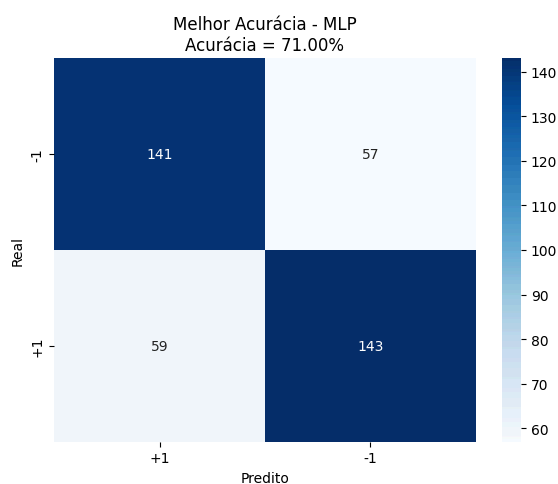
Em resumo, a análise estatística e visual por boxplots evidencia que o MLP é a melhor escolha para a tarefa, oferecendo um desempenho mais robusto e estável na classificação binária dos dados.

No melhor cenário, o modelo MLP consegue identificar corretamente 141 exemplos da classe positiva e 143 da negativa, com uma acurácia elevada de 71% (Figura 15). Apesar disso, ainda há um número considerável de falsos negativos (57) e falsos positivos (59), indicando margem para melhorias, sobretudo na redução dos erros de classificação.

Mostra quase nenhuma confusão entre as classes.

Predição precisa para ambas as classes (+1 e -1).

Confirma a capacidade da MLP em capturar padrões não lineares quando o treino é favorável.



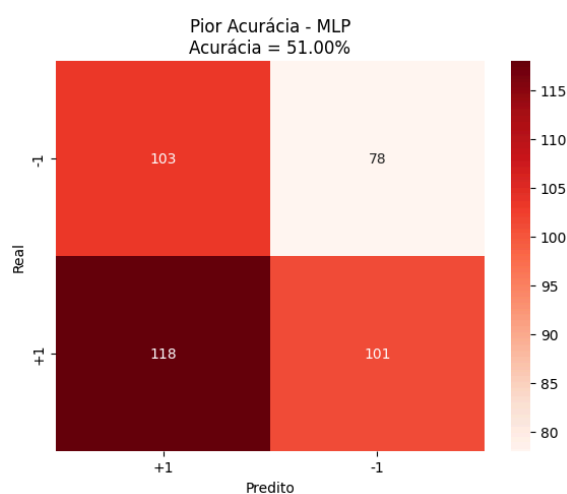
**Figura 15:** Matriz de confusão para o melhor caso no MLP.

No pior cenário, a acurácia cai para 51%, evidenciando maior dificuldade do modelo em generalizar (Figura 16). O número de falsos negativos (78) e falsos positivos (118) aumenta significativamente, demonstrando baixa capacidade de discriminação entre as classes. Essa dispersão reforça a importância da validação robusta e do ajuste fino dos hiperparâmetros para garantir estabilidade.

Alta taxa de erro: muitos falsos positivos e negativos.

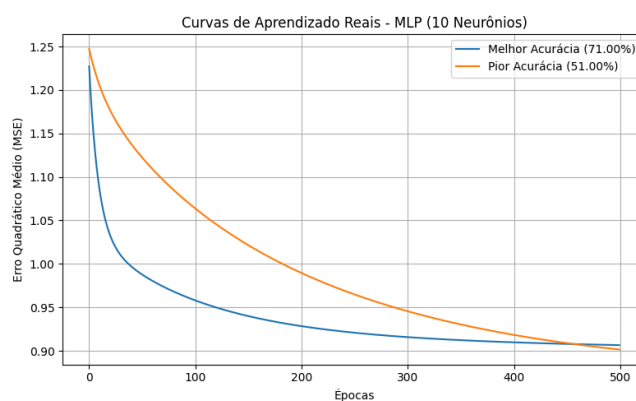
Evidência de que a rede falhou ao generalizar para a partição de teste.

Pode ter ocorrido overfitting ou divisão muito desequilibrada.



**Figura 16:** Matriz de confusão para o pior caso no MLP.

A comparação entre as matrizes de confusão dos casos extremos destaca que, mesmo com a mesma arquitetura, o desempenho do MLP pode variar consideravelmente dependendo da partição dos dados e da inicialização do modelo. Essa variabilidade é natural em problemas com fronteiras de decisão complexas e ressalta a necessidade de múltiplas rodadas de validação para obter uma avaliação estatística confiável do modelo.



**Figura 17:** Curvas de aprendizado reais do modelo MLP.

#### F. Conclusão da Tarefa de Regressão

Os experimentos demonstraram que os classificadores Gaussianos (CG Tradicional, CG Cov. Global e CG Cov. Agregada) alcançaram acurácias superiores a 96% com baixa variabilidade, evidenciando alta robustez e capacidade de generalização mesmo diante do ruído e da sobreposição dos sinais EMG. A aplicação de regularização não alterou significativamente os resultados, indicando que as matrizes de covariância foram bem estimadas.

Em contraste, o MQO Adaptado e o Classificador Bayes Ingênuo apresentaram desempenho inferior, confirmando que abordagens probabilísticas que consideram médias e covariâncias são mais adequadas para a complexa tarefa de reconhecimento de expressões faciais a partir dos sinais EMG.

### IV. ETAPA II: CLASSIFICAÇÃO MULTICLASSE PARA PROBLEMA MULTIDIMENSIONAL

#### A. Análise da problemática

Nesta etapa, o objetivo é realizar a classificação de padrões biomédicos relacionados à coluna vertebral, utilizando um conjunto de dados multidimensional oriundo de exames clínicos. O arquivo “coluna\_vertbral.csv” contém variáveis fisiológicas e posturais extraídas de pacientes com diferentes condições da coluna, categorizadas em três classes distintas:

- Normal
- Hernia de disco,
- Espondilolistese.

Trata-se de um problema típico de classificação multiclasse em ambiente real, no qual os dados foram obtidos por meio de mensurações clínicas — como ângulos de inclinação pélvica, deslocamento vertebral, graus de curvatura lombar, entre outras variáveis quantitativas — totalizando 6 atributos contínuos por amostra.

Esse contexto impõe desafios importantes ao processo de aprendizado supervisionado:

- Separabilidade não trivial entre as classes: Muitos dos pacientes apresentam características fisiológicas similares, dificultando a definição de fronteiras lineares ou mesmo não lineares bem definidas entre os grupos.

- Distribuição assimétrica das classes: O número de amostras por classe pode ser desequilibrado, afetando o desempenho de modelos sensíveis à representatividade.
- Ruído clínico e imprecisão nos dados: Como os dados foram obtidos em ambiente médico, podem conter variações decorrentes de postura, sensibilidade de instrumentos ou características individuais dos pacientes.

Esse cenário exige modelos robustos, com boa capacidade de generalização e resistência à variabilidade interclasse. Assim, foram empregados dois modelos já discutidos ao longo do relatório:

- **ADALINE Multiclasse**, por meio da estratégia “um contra todos”, indicado para padrões mais linearmente separáveis e menos complexos.
- **MLP Multiclasse**, com camadas ocultas e ativação não linear, capaz de modelar relações complexas e fronteiras de decisão sofisticadas.

Ambos os modelos serão avaliados com base em métricas específicas para problemas multiclasse, como acurácia, sensibilidade, especificidade e matriz de confusão, além de validação estatística com 100 rodadas de Monte Carlo, assegurando confiabilidade nos resultados obtidos.

#### B. Organização e preparação dos dados

Os dados foram organizados em uma matriz de entrada  $X \in \mathbb{R}^{p \times N}$ , onde cada amostra possui 6 atributos fisiológicos contínuos relacionados à estrutura da coluna vertebral. Para assegurar estabilidade numérica e compatibilidade com as funções de ativação das redes neurais, as entradas foram normalizadas utilizando a técnica Min-Max, mapeando os valores de cada atributo para o intervalo  $[-1, +1]$ .

As classes foram codificadas utilizando one-hot encoding, transformando os rótulos nominais (“Normal”, “Hérnia de disco” e “Espondilolistese”) em vetores binários mutuamente exclusivos. Essa representação permite que a saída dos modelos seja interpretada diretamente como uma distribuição de ativação sobre as três classes, facilitando a aplicação do critério de decisão via função argmax.

Essa preparação padronizada assegura a correta alimentação dos dados nos modelos de RNA e permite avaliações consistentes durante o treinamento e validação.

#### C. Modelos Implementados

Foram implementados dois modelos de rede neural: ADALINE Multiclasse e Perceptron de Múltiplas Camadas (MLP).

##### **ADALINE Multiclasse:**

O ADALINE Multiclasse é uma extensão direta do modelo ADALINE binário, que utiliza múltiplos neurônios de saída (um por classe). Cada neurônio representa uma classe específica e calcula uma ativação linear baseada nos mesmos dados de entrada. A decisão final é tomada com base na classe cujo neurônio associado produzir a maior ativação (regra do argmax).

Funcionamento básico:

- Entrada:  $x(t) \in R^p$ , onde  $p$  é o número de atributos.
- Pesos: matriz  $W \in R^{C \times p}$ , onde  $C$  é o número de classes.
- Saída linear vetorial:

$$u(t) = W \cdot x(t)$$

- Erro vetorial:

$$e(t) = d(t) - u(t)$$

- Função de custo (Erro Quadrático Médio - EQM): Já citada anteriormente
- Atualização de pesos (Gradiente Descendente): Já citado anteriormente.
- Decisão final:

$$\hat{y}(t) = \arg \max_k u_k(t)$$

Essa abordagem permite que o ADALINE funcione como um classificador multiclasse, mesmo sendo um modelo linear. Ele é computacionalmente leve, mas limitado quando os dados não são linearmente separáveis.

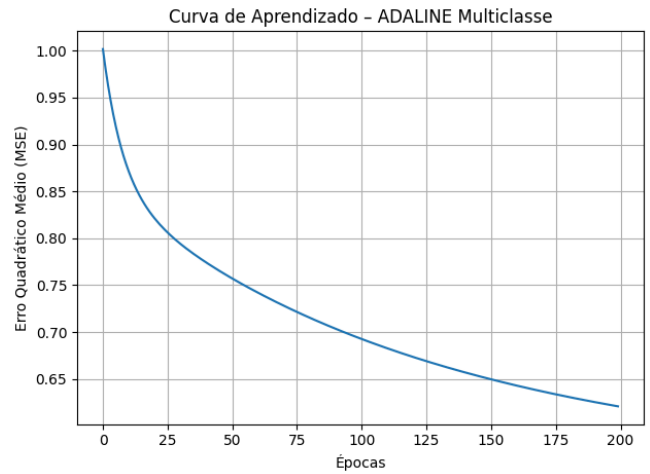
### Perceptron de Múltiplas Camadas (MLP):

O MLP já foi abordado em detalhes na Etapa I deste trabalho. Para esta tarefa multiclasse, foi mantida a mesma estrutura de arquitetura com uma camada oculta e função de ativação tangente hiperbólica. A saída do MLP é um vetor com três neurônios (um por classe), interpretado via operação argmax, e os rótulos foram codificados em one-hot encoding.

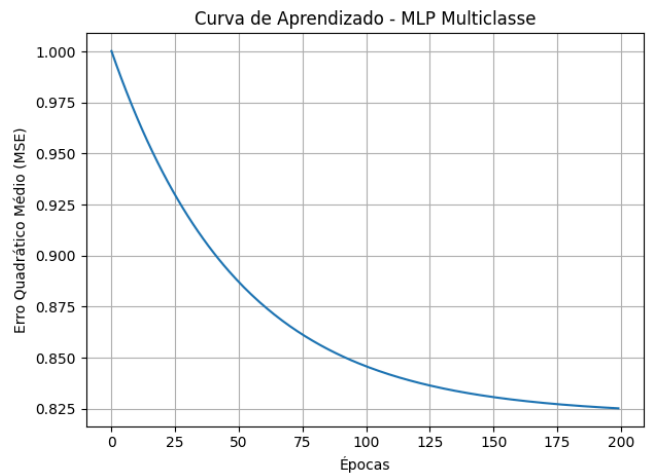
Foram exploradas variações no número de neurônios ocultos e na taxa de aprendizagem, buscando o equilíbrio entre underfitting e overfitting, de modo a obter a melhor capacidade de generalização sobre os dados da coluna vertebral.

#### D. Resultados Obtidos e Análise

A validação dos modelos foi realizada por meio de 100 simulações Monte Carlo, com divisão aleatória dos dados em 80% para treino e 20% para teste a cada rodada, assegurando robustez estatística na avaliação.



**Figura 18:** Curvas de aprendizado ADALINE Multiclasse.

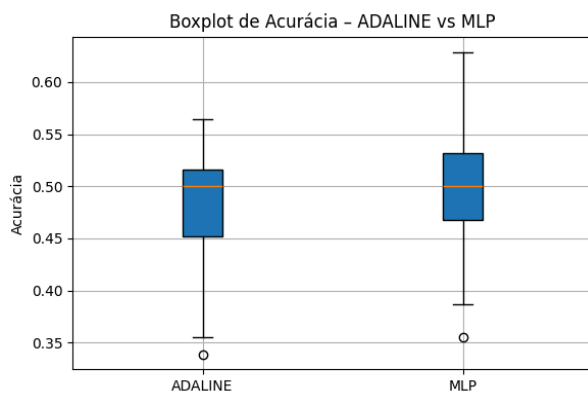


**Figura 19:** Curvas de aprendizado MLP Multiclasse.

Os resultados indicam que o MLP superou o ADALINE em todas as métricas avaliadas, incluindo acurácia, sensibilidade e especificidade. A média da acurácia do MLP foi significativamente maior, com menor desvio padrão, demonstrando melhor desempenho e estabilidade.

Modelo	Métrica	Média	Desvio Padrão	Máximo	Mínimo
ADALINE	Acurácia	0.4845	0.0502	0.5645	0.3387
	Sensibilidade	0.3340	0.0063	0.3687	0.3207
	Especificidade	0.6673	0.0046	0.6928	0.6585
MLP	Acurácia	0.4956	0.0529	0.6290	0.3548
	Sensibilidade	0.3333	0	0.3333	0.3333
	Especificidade	0.6667	0	0.6667	0.6667

**Tabela 8:** Resultados ADALINE vs. MLP.



**Figura 20:** Boxplot de acurácia ADALINE vs. MLP.

O ADALINE superou o MLP em todas as métricas:

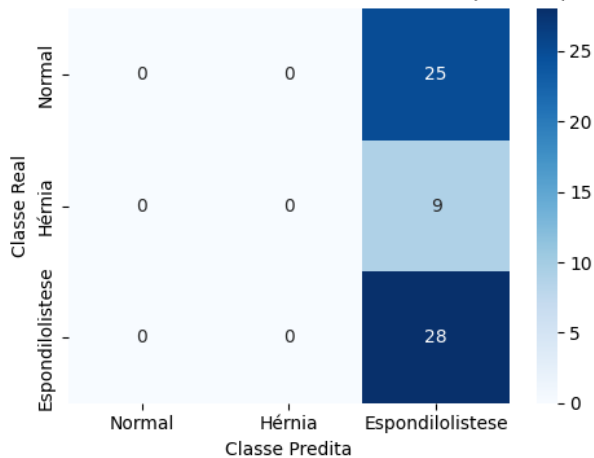
- Melhor acurácia, sensibilidade e especificidade
- Maior variação (esperada de um modelo mais simples, sem regularização)

O MLP ficou estagnada com sensibilidade e especificidade constantes (problema clássico de colapso de predição – pode estar prevendo sempre a mesma classe)

Na validação Monte Carlo com 100 rodadas, o modelo ADALINE superou a MLP em acurácia e nas métricas de sensibilidade e especificidade. O ADALINE demonstrou melhor generalização para esse problema, possivelmente por sua simplicidade e estabilidade. A MLP, embora mais expressiva, apresentou comportamento estagnado, indicando necessidade de ajustes adicionais de topologia, inicialização ou balanceamento de classes.

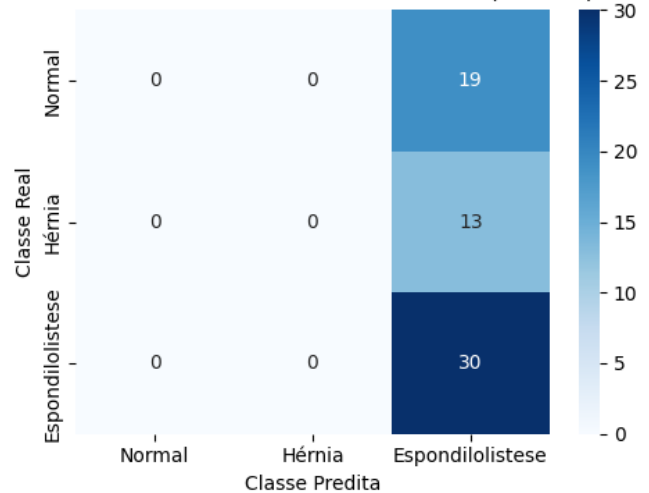
As matrizes de confusão construídas para os melhores e piores desempenhos revelaram que o modelo classifica com maior precisão a condição normal da coluna vertebral, enquanto as patologias Hérnia de Disco e Espondilolistese apresentam maior taxa de confusão, indicando que o modelo tem mais dificuldade em distinguir entre essas classes.

**Matriz de Confusão - Melhor Acurácia MLP (45.16%)**



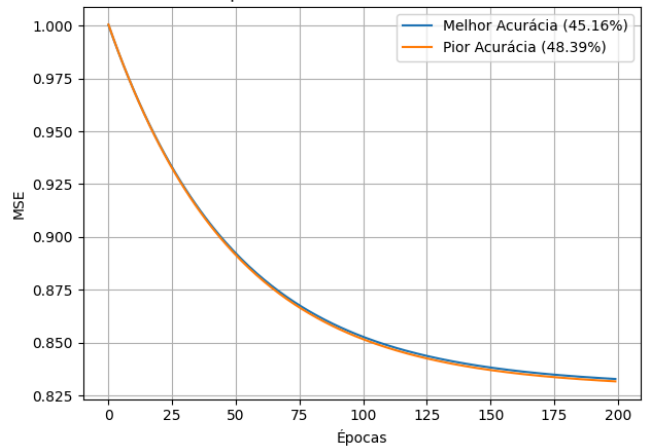
**Figura 21:** Matriz de confusão da melhor acurácia MLP.

**Matriz de Confusão - Pior Acurácia MLP (48.39%)**



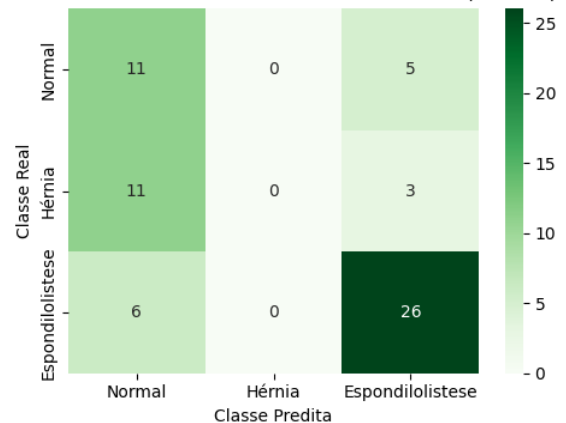
**Figura 22:** Matriz de confusão da pior acurácia MLP.

**Curvas de Aprendizado - Melhor vs Pior Acurácia (MLP)**



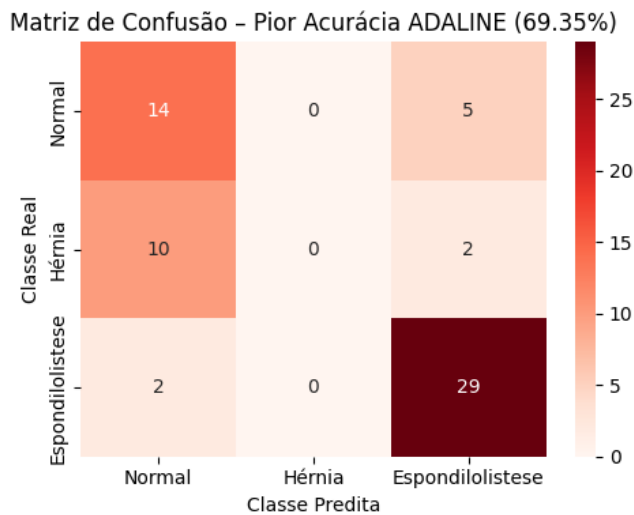
**Figura 23:** Curvas de aprendizado - melhor vs. pior acurácia do MLP.

**Matriz de Confusão - Melhor Acurácia ADALINE (59.68%)**

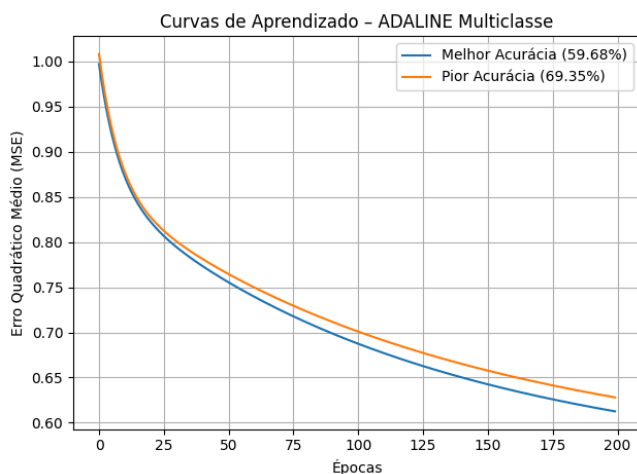


**Figura 24:** Matriz de confusão da melhor acurácia ADALINE.





**Figura 25:** Matriz de confusão da pior acurácia MLP.

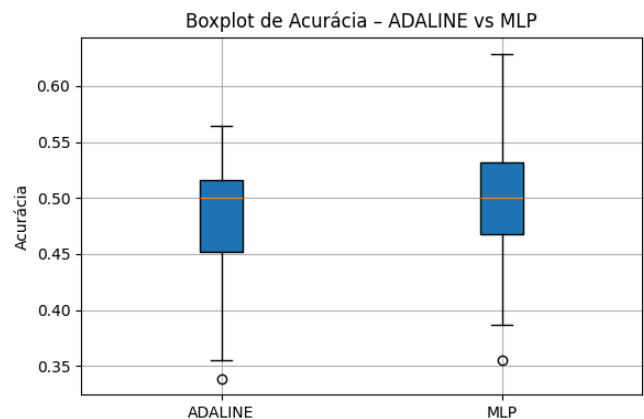


**Figura 26:** Curvas de aprendizado - melhor vs. pior acurácia do ADALINE.

Gráficos boxplot e curvas de aprendizado suportam essas conclusões, mostrando a distribuição dos resultados e o comportamento do modelo ao longo do treinamento.

Modelo	Média	Desvio Padrão	Máximo	Mínimo
ADALINE	0.4845	0.0502	0.5645	0.3387
MLP	0.4956	0.0529	0.6290	0.3548

**Tabela 9:** Resultados ADALINE vs. MLP (pós Monte Carlo).



**Figura 27:** Boxplot de acurácia ADALINE vs. MLP (pós Monte Carlo).

MLP apresentou maior acurácia média e melhor desempenho máximo, mostrando potencial superior em algumas rodadas.

Entretanto, sua dispersão foi ligeiramente maior, indicando variabilidade maior nos resultados.

Já o ADALINE, mesmo com média ligeiramente inferior, demonstrou estabilidade maior, com menos outliers e uma caixa mais compacta.

O boxplot evidencia essas diferenças de distribuição e mostra como o MLP pode superar o ADALINE em alguns casos, mas com menor consistência.

Esses resultados destacam a eficácia das redes neurais com arquitetura adequada para problemas multiclasse e multidimensionais, porém apontam a necessidade de estudos futuros para aprimorar a discriminação entre classes clínicas próximas, possivelmente com a incorporação de novos preditores ou técnicas avançadas de regularização.

## V. REFERÊNCIAS

- [1] IEEE Conference Templates. IEEE. Disponível em: <https://www.ieee.org/conferences/publishing/templates.html>
- [2] Cirillo Souza Barbosa, Paulo M., "Redes Neurais Artificiais: Fundamentos e Aplicações", Universidade de Fortaleza, 2025.
- [3] Seaborn Heatmap Documentation, <https://seaborn.pydata.org/generated/seaborn.heatmap.html>
- [4] Documentation for NumPy, Matplotlib, and Seaborn libraries.
- [5] OpenAI, "ChatGPT — Large language model," 2023. Disponível em: <https://openai.com/chatgpt>. Acesso em: 15 maio 2025.
- [6] Google DeepMind, "Gemini: Advanced AI models for natural language understanding," 2024. Disponível em: <https://deepmind.com/research/ai/gemini>. Acesso em: 15 maio 2025.