



Gradiente Descendente

Aplicado a Regressão Polinomial,
Regressão Logística e Rede Neural

FERNANDO GOMES CRUZ

Objetivos Principais

Demonstrar a aplicação do gradiente descendente nos modelos

Obter os coeficientes do polinômio de grau 17 através do modelo
de Regressão Polinomial

Treinar um modelo de Regressão Logística

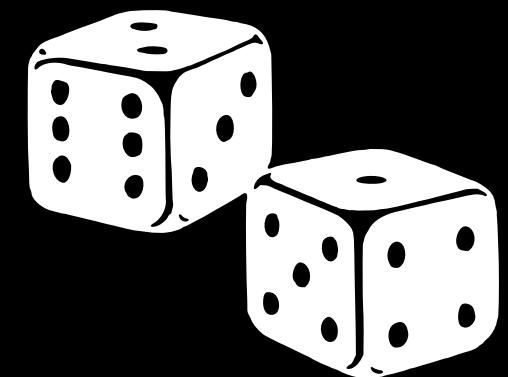
Otimizar os pesos em uma Rede Neural

Dados

O modelo de **Regressão Polinomial** tem como dados de entrada um conjunto de dados gerados a partir de coeficientes aleatórios iniciais.

O modelo **Regressão Logística** e **Rede Neural** tem como base um conjunto de dados tratados disponibilizados em outro projeto do autor, provenientes primariamente do repositório **45/heart+disease** da UC Irvine Machine Learning Repository sob a licença CC BY 4.0.

https://github.com/neocrz/heart_disease
<https://archive.ics.uci.edu/dataset/45/heart+disease>



Var	Nome	Função	Tipo	Descrição	Unidade	Categorias
age	Idade	Variável	Int	Idade do paciente		
sex	Sexo	Variável	Cat	Gênero		0: Feminino, 1: Masculino
cp	Dor no Peito	Variável	Cat	Tipo de dor no peito		1: Angina típica, 2: Angina atípica, 3: Dor não anginosa, 4: Assintomática
trestbps	PA em Repouso	Variável	Int	Pressão arterial em repouso	mm Hg	
chol	Colesterol	Variável	Int	Nível de colesterol no soro	mg/dL	
fbs	Glicemia Jejum	Variável	Cat	Glicemia em jejum > 120 mg/dL		0: Falso, 1: Verdadeiro
restecg	Resultados ECG	Variável	Cat	Resultados do eletrocardiograma em repouso		0: Normal, 1: Anormalidade ST-T, 2: Hipertrofia ventricular esquerda provável
thalach	FC Máxima	Variável	Int	Frequência cardíaca máxima atingida	bpm	
exang	Angina Exercício	Variável	Cat	Angina provocada pelo exercício		0: Não, 1: Sim
oldpeak	Depressão ST	Variável	Int	Depressão do segmento ST pelo exercício	mm	
slope	Inclinação ST	Variável	Cat	Inclinação do segmento ST no pico		1: Ascendente, 2: Plana, 3: Descendente
ca	Vasos	Variável	Int	Número de vasos principais detectados		0: 0, 1: 1, 2: 2, 3: 3
thal	Condição Cardíaca	Variável	Cat	Tipo de condição cardíaca		3: Normal, 6: Defeito fixo, 7: Defeito reversível
num	Diagnóstico	Alvo	Cat	Diagnóstico de doença cardíaca		0: Sem doença, 1: Doença

Regressão Polinomial

A regressão polinomial é uma técnica que busca encontrar uma função polinomial $f(x)$ que melhor se ajusta a um conjunto de dados. Essa abordagem é ideal para modelar relações não-lineares.

Utilizamos o método de **gradiente descendente** para ajustar os coeficientes do polinômio, minimizando o erro entre os valores reais e os valores preditos.

Objetivos do Modelo

Determinar os coeficientes de um polinômio de grau 17 que melhor se ajusta ao conjunto de dados.

Utilizar o método de gradiente descendente com uma taxa de aprendizado e tolerância específicas.

Avaliar o desempenho do modelo por meio do Erro Quadrático Médio e observação gráfica.

Regressão Polinomial

$$f(x) = a_0x^n + a_1x^{n-1} + \dots + a_n$$

Erro Quadrático Médio
(loss function)

$$E(a) = \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i))^2$$

$$\frac{\partial E}{\partial a_k} = -\frac{2}{m} \sum_{i=1}^m (y_i - f(x_i)) \cdot x_i^{n-k}$$

Gradiente
Descendente

$$a_k^{\text{nov}} = a_k - \eta \frac{\partial E}{\partial a_k}$$

$$\text{dist}(a_k^{\text{nov}}, a_k) > \text{tol}$$

Resultados

Entrada

X: sequência de 30 números entre -1 e 1

Coeficientes reais: [4.1, -1.7, -3.1, 0.1, -4.8, -0.5, 4.6, 2.5, -2.9, 2.8, -2.3, 0.7, -4.5, 0.8, -0.2, 2.8, 2.1, 2.9]

tolerância: $10^{(-5)}=0.00001$

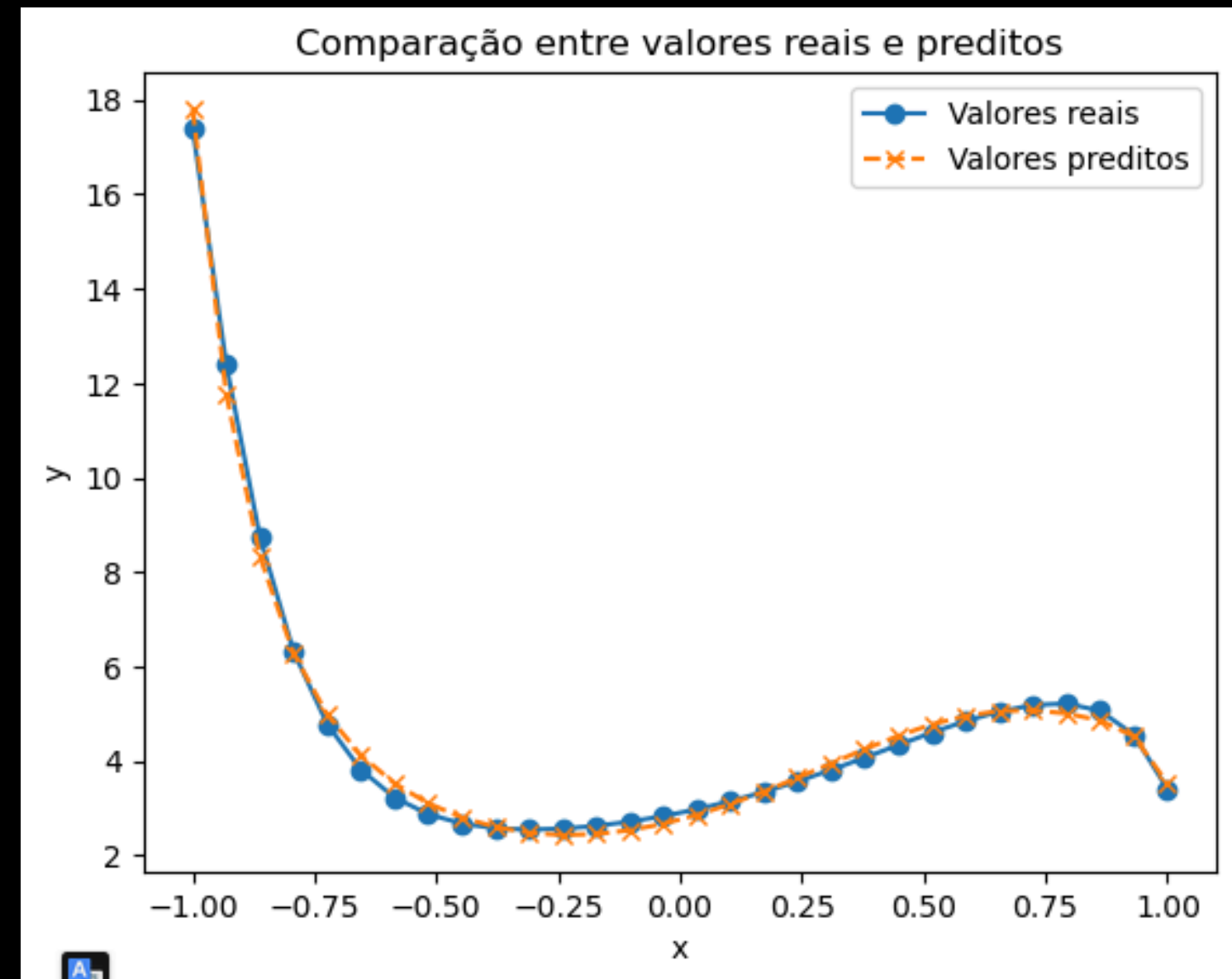
learning rate: $10^{(-5)}=0.00001$

Resultados

Coeficientes obtidos: [-0.1, -2.0, -2.3, 3.2, -2.3, -1.7, 4.2, 2.2, -2.7, 4.9, -1.0, -1.8, -2.2, -1.9, -3.4, 4.9, 2.7, 2.8]

Iterações: 68612

Erro quadrático médio (RMSE): 0.216



Regressão Logística

A regressão logística é uma técnica que busca prever a presença ou ausência de uma característica ou resultado com base em valores de um conjunto de variáveis preditoras

Utilizamos o método de **gradiente descendente** para ajustar os coeficientes da regressão logística, minimizando o erro entre os valores reais e os valores preditos.

Objetivos do Modelo

Determinar os coeficientes do modelo para os dados de entrada.

Utilizar o método de gradiente descendente com uma taxa de aprendizado e tolerância específicas.

Avaliar o desempenho do modelo.

Regressão Logística

Função de Ativação
(sigmoid)

$$h_{\theta}(x) = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n$$

Função de Erro (log loss)

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$$

Atualização dos
Coeficientes

$$\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

Resultados

tolerância: 10^{-3}

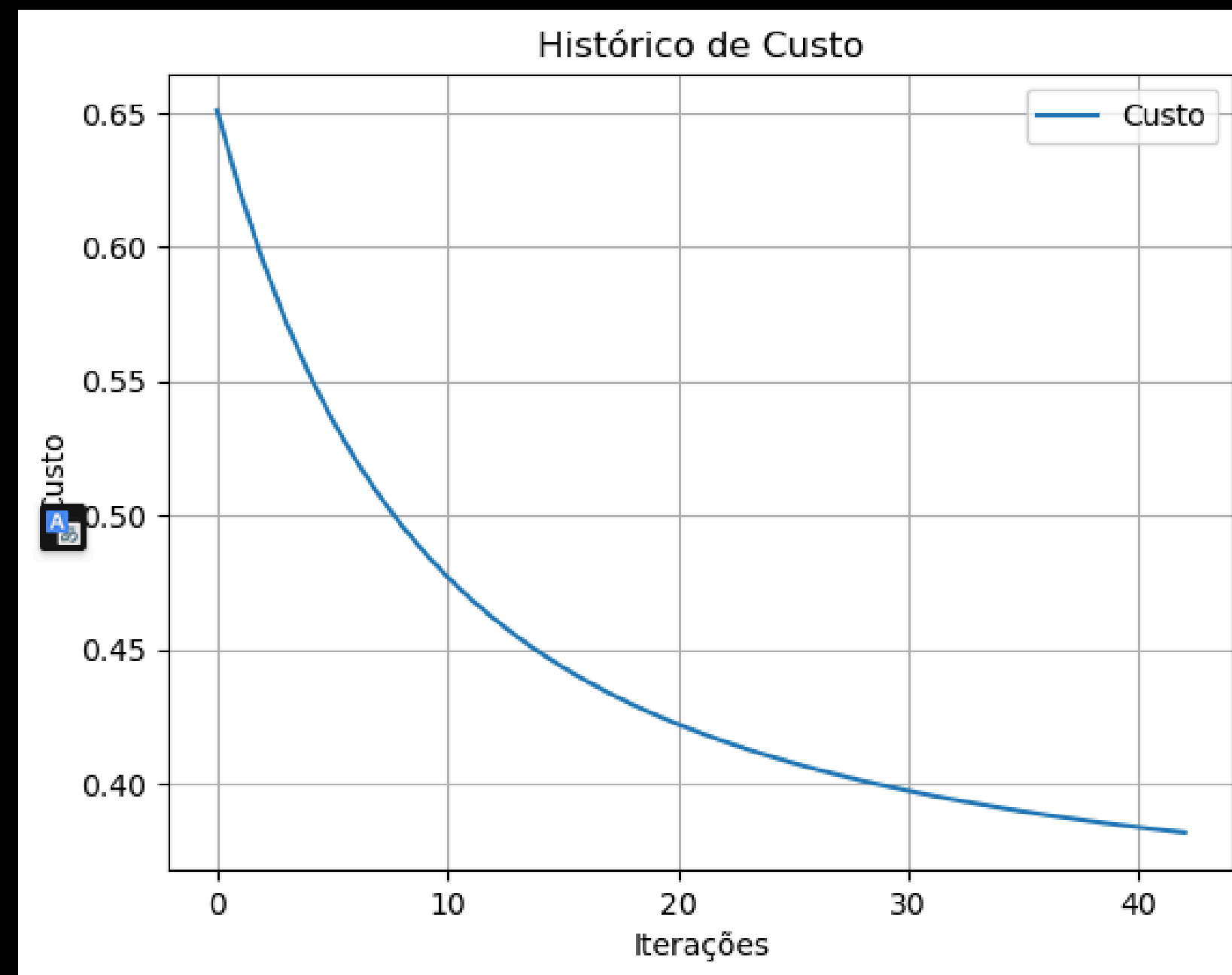
learning rate: 0.1

Resultados

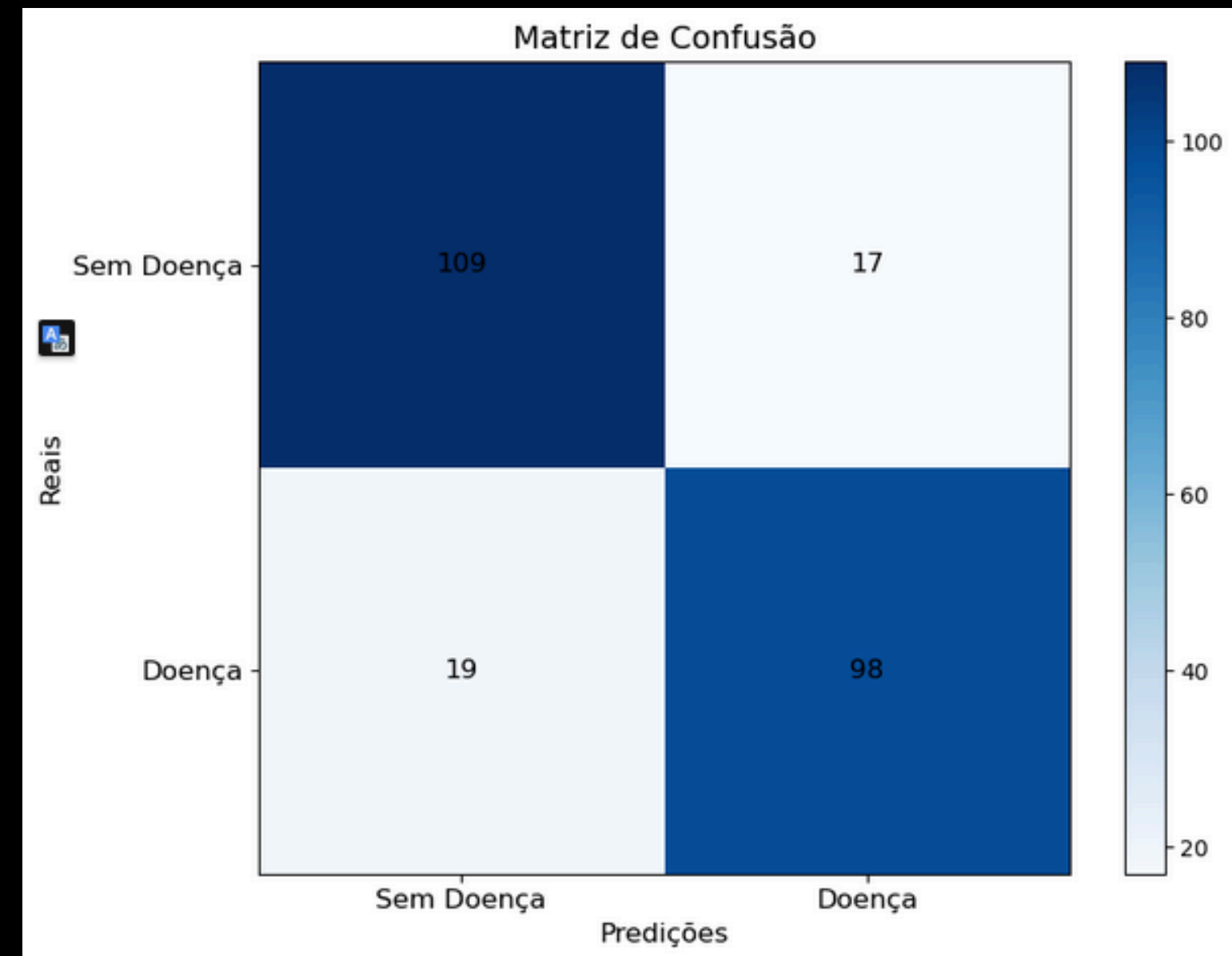
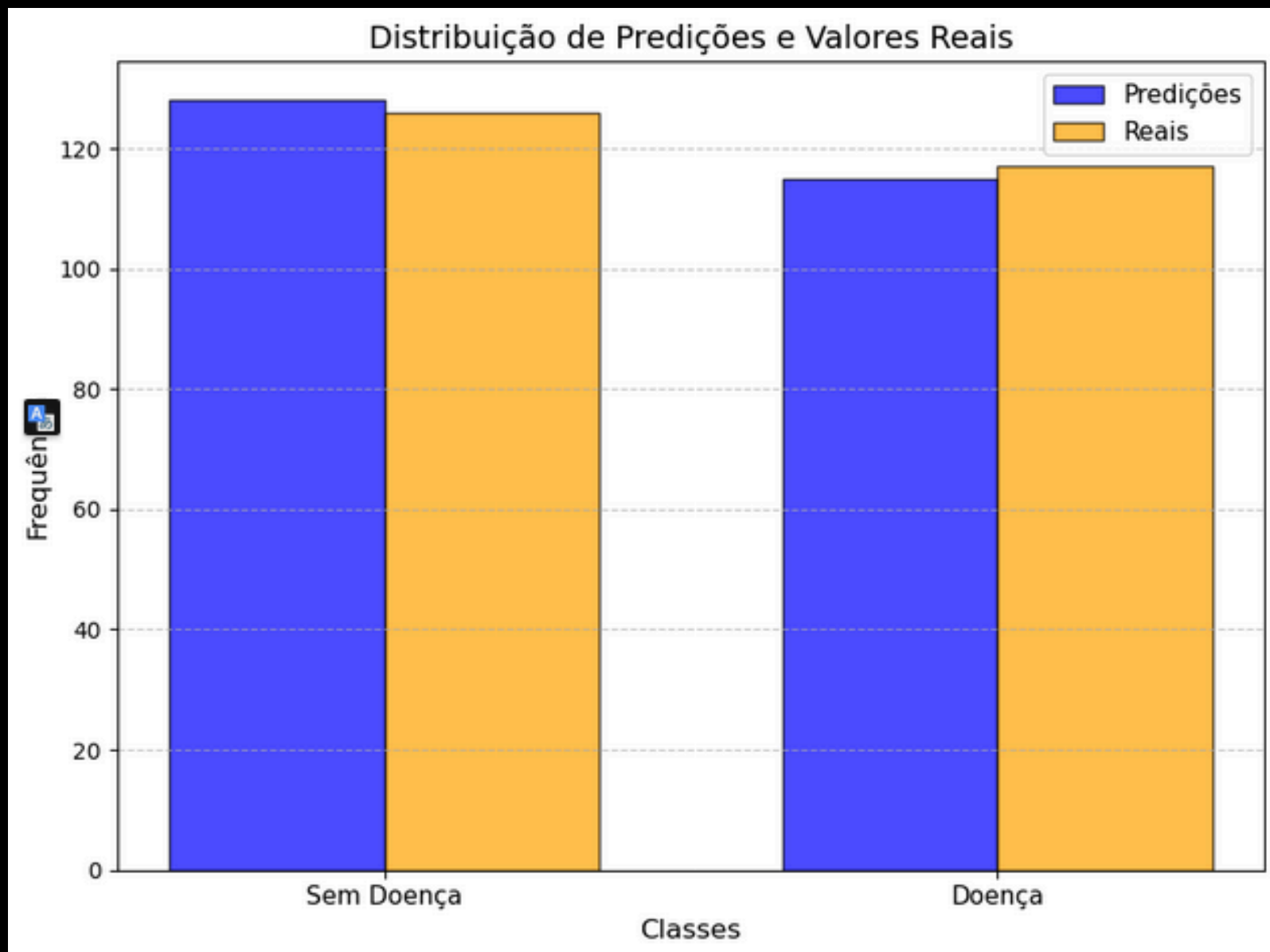
Coeficientes otimizados: [0.1, 0.2, 0.4, 0.1, 0.1, 0.0, 0.2, -0.3, 0.3, 0.3, 0.3, 0.4, 0.4]

Iterações: 42

Precisão no conjunto de teste: 83.33%



Resultados



Rede Neural

Uma Rede Neural é composta por camadas de neurônios artificiais e utiliza funções de ativação, como a sigmoide, para transformar entradas em saídas

A rede neural consiste em três camadas: uma **camada de entrada**, uma **camada oculta** e uma **camada de saída**.

Rede Neural

Entrada

$$X \in \mathbb{R}^{m \times n}$$

Ativação

$$\hat{y} = A^{[2]} = \sigma(Z^{[2]})$$

Função de Erro: Log loss

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right]$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

<https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a>

<https://mariofilho.com/guia-completo-da-log-loss-perda-logaritmica-em-machine-learning/>

Rede Neural

Camada Oculta

$$W^{[1]} \in \mathbb{R}^{h \times n}$$

$$b^{[1]} \in \mathbb{R}^h$$

$$Z^{[1]} = W^{[1]}X^T + b^{[1]}$$

$$A^{[1]} = \sigma(Z^{[1]}) = \frac{1}{1 + e^{-Z^{[1]}}}$$

Erro propagado

$$\delta^{[1]} = \left(W^{[2]}\right)^T \delta^{[2]} \odot \sigma'(Z^{[1]})$$

Gradientes

$$\frac{\partial J}{\partial W^{[1]}} = \frac{1}{m} \delta^{[1]} X \quad \text{e} \quad \frac{\partial J}{\partial b^{[1]}} = \frac{1}{m} \sum \delta^{[1]}$$

Rede Neural

Camada Saída

$$W^{[2]} \in \mathbb{R}^{1 \times h}$$

$$b^{[2]} \in \mathbb{R}$$

$$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$

$$\hat{y} = A^{[2]} = \sigma(Z^{[2]})$$

Erro

$$\delta^{[2]} = A^{[2]} - Y$$

Gradientes

$$\frac{\partial J}{\partial W^{[2]}} = \frac{1}{m} \delta^{[2]} A^{[1]} \quad \text{e} \quad \frac{\partial J}{\partial b^{[2]}} = \frac{1}{m} \sum \delta^{[2]}$$

Resultados

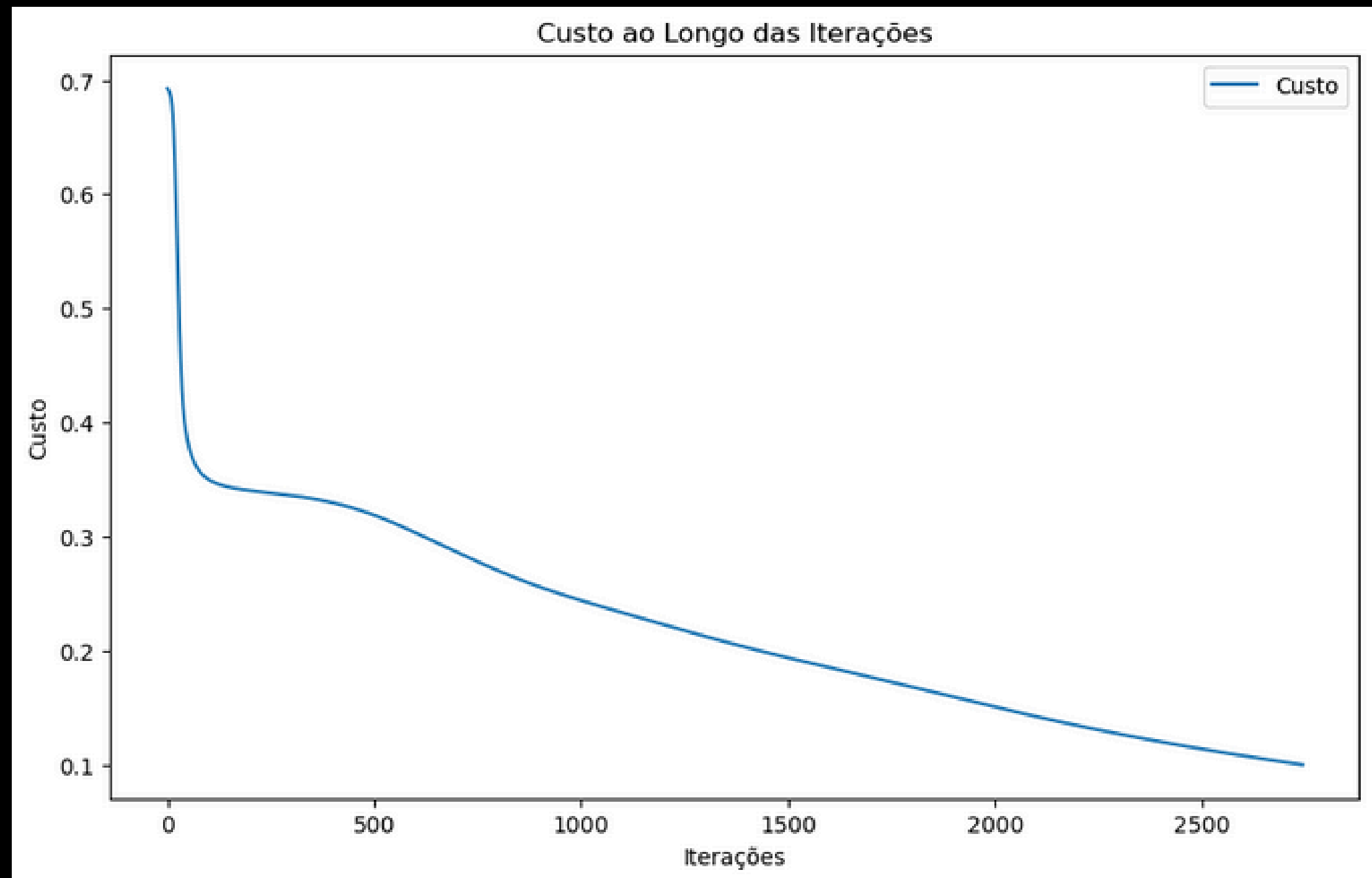
tolerância: 0.1
learning rate: 0.01
neurônios (oculta): 8

Resultados

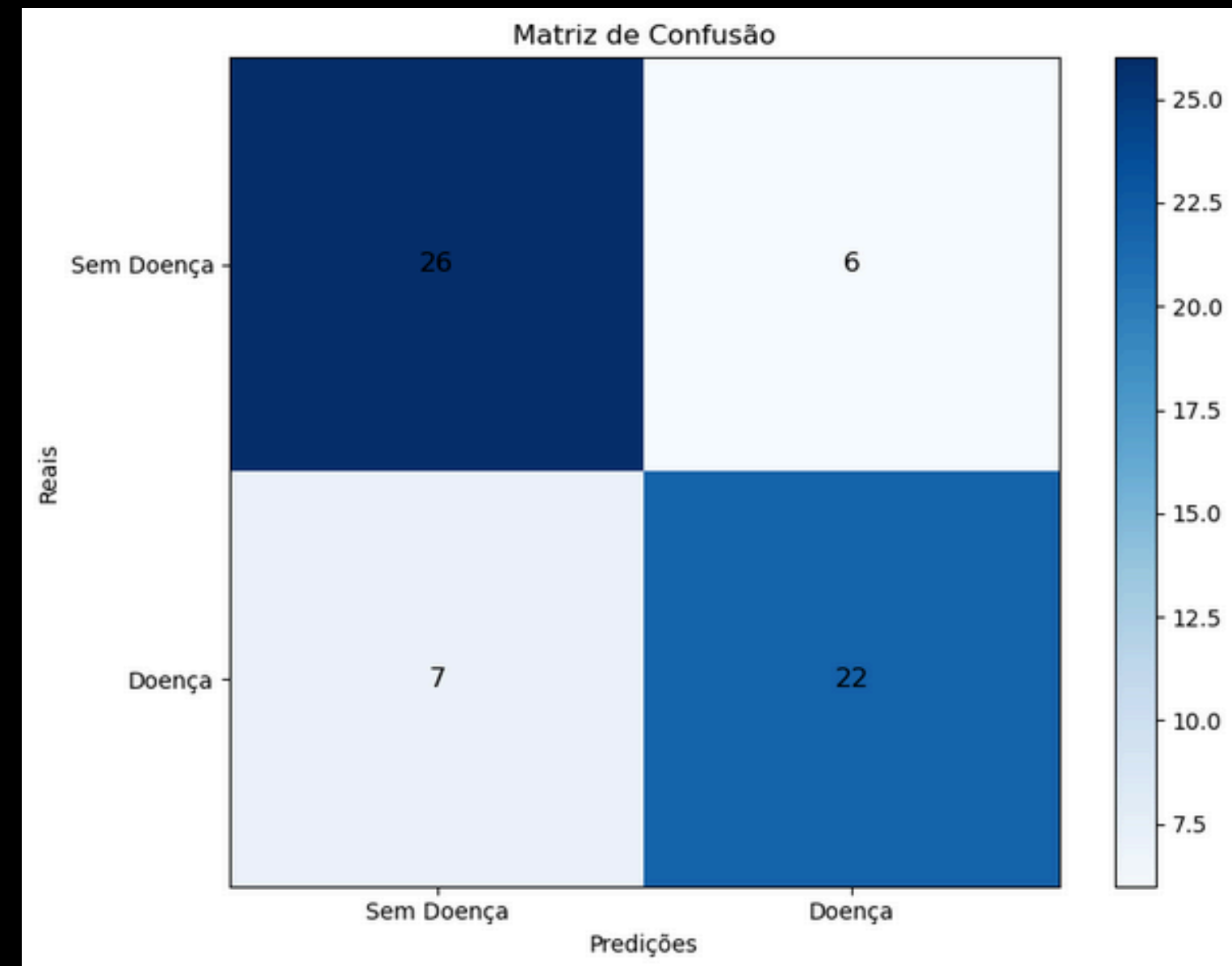
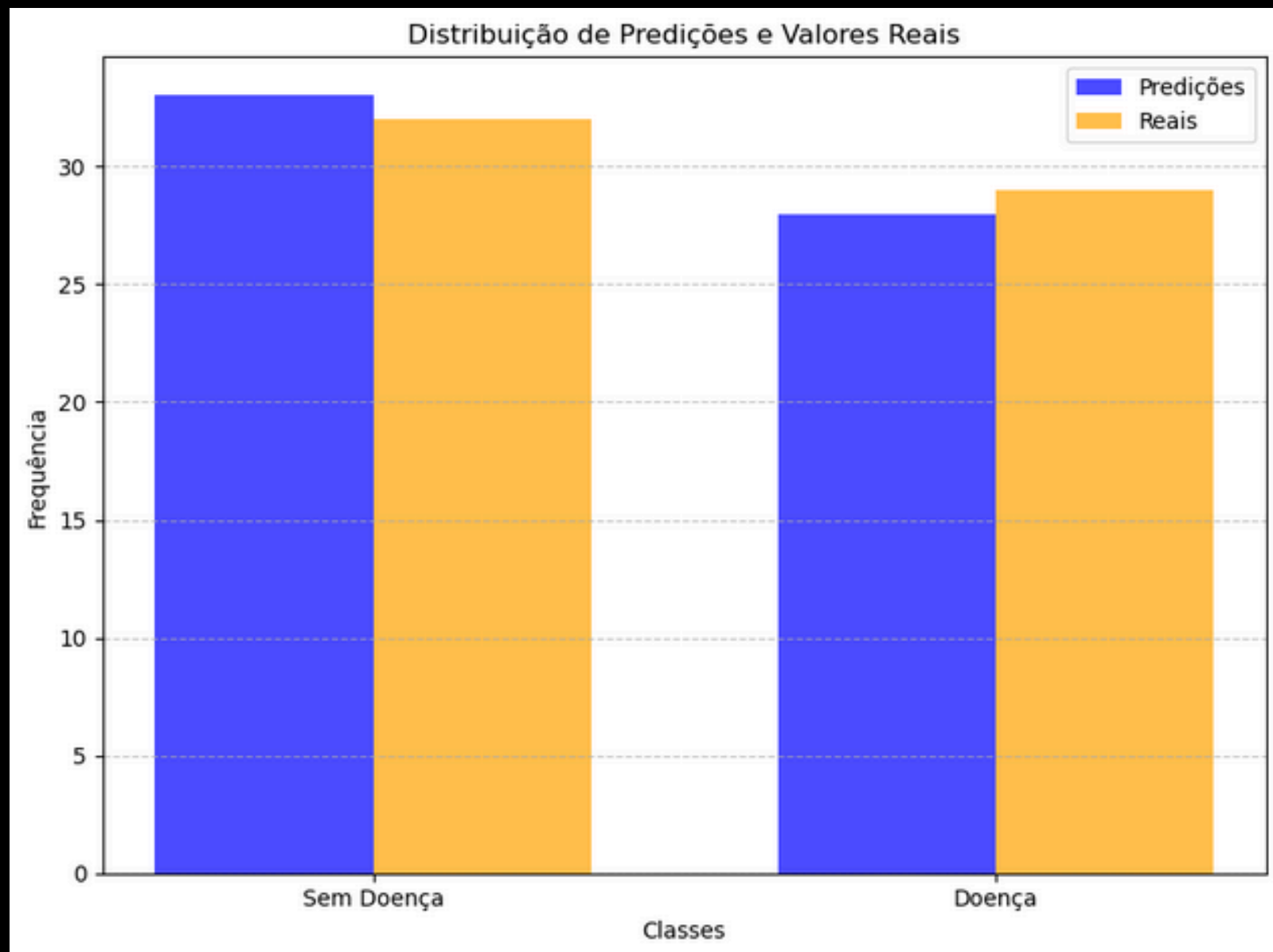
Iterações: 2743

Precisão no Treinamento: 98.35%

Precisão no Teste: 78.69%

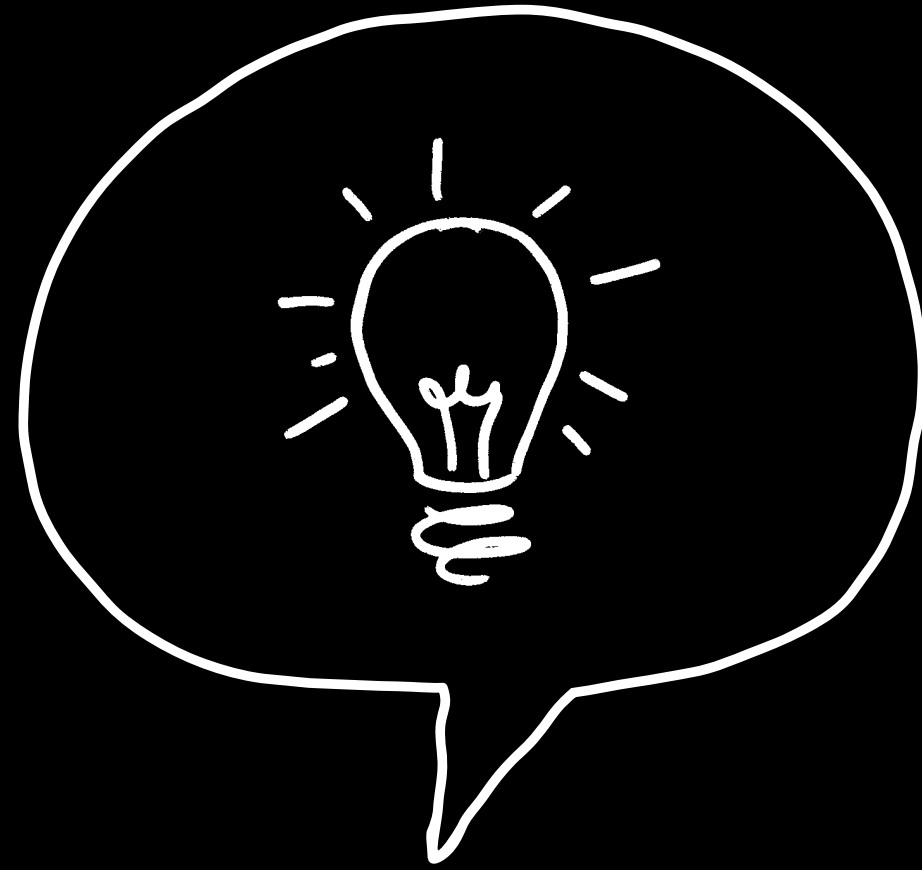


Resultados



Conclusão

O gradiente descendente é um método robusto e sólido para otimizar funções de custo em diferentes tipos de problemas de regressão e classificação, destacando-se como um pilar na construção de modelos de machine learning eficientes.



Perguntas?

Obrigado!

