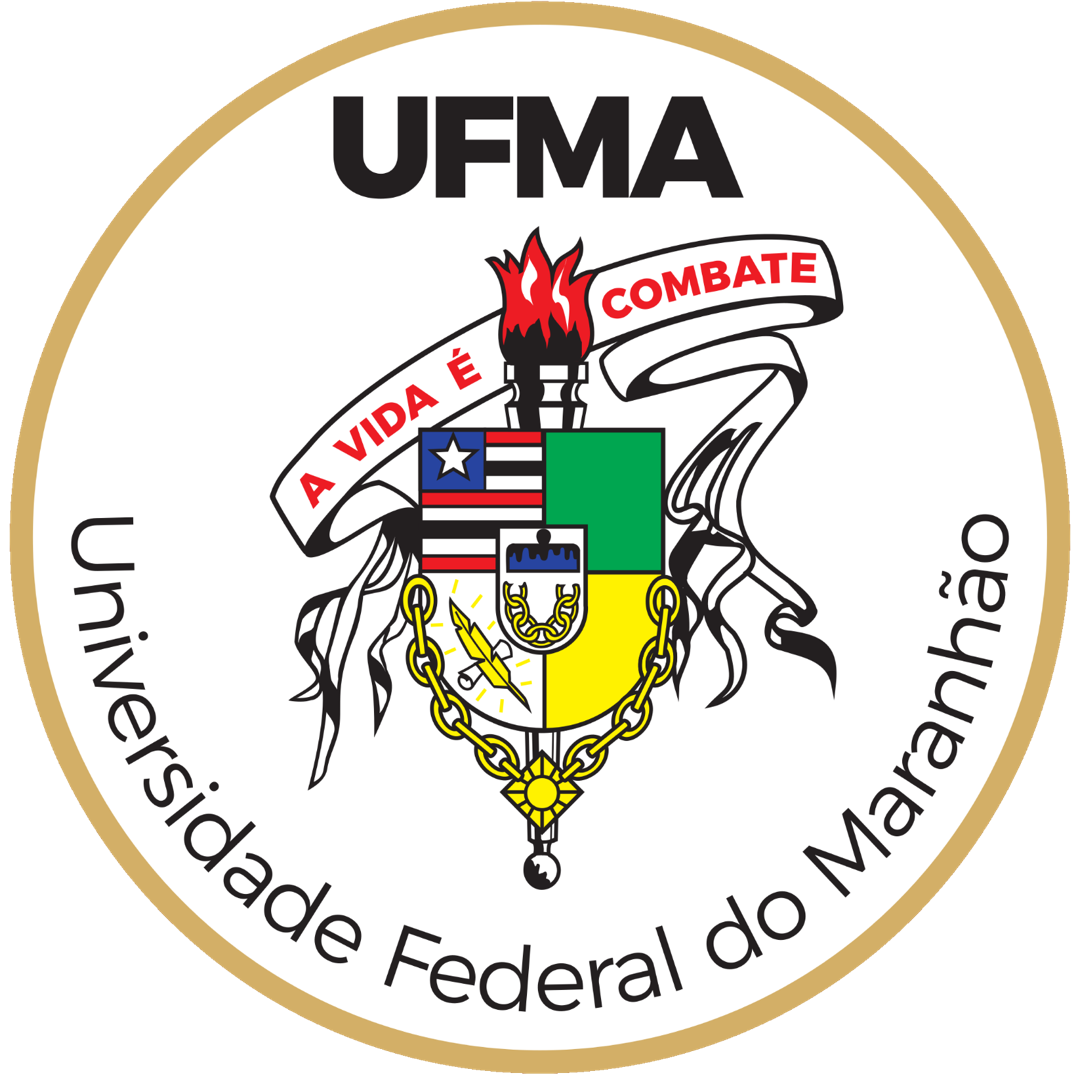
****

**UNIVERSIDADE FEDERAL DO MARANHÃO**

**ENGENHARIA DA COMPUTAÇÃO**

**Trabalho Individual: Regressão Linear**

**EECP0053 - TÓPICOS EM ENGENHARIA DA COMPUTAÇÃO II - FUNDAMENTOS DE REDES NEURAIS**

Discente: Guilherme Roberto Matos Silva

São Luís - MA.

2025.

**INTRODUÇÃO**

Neste trabalho, foi desenvolvido um script de regressão linear simples com o objetivo de proporcionar uma compreensão prática das etapas fundamentais de um modelo de aprendizado supervisionado. O processo envolveu desde a preparação dos dados até a visualização da função de custo em 3D, incluindo a implementação da função de custo e do algoritmo de descida do gradiente.

**DESENVOLVIMENTO DO SCRIPT**

1. Exercícios de Aquecimento (Warm-up)

O script foi iniciado com a execução de exercícios de aquecimento (warm-up exercises), fundamentais para a familiarização com operações básicas de álgebra linear utilizando o NumPy. As atividades realizadas incluíram:

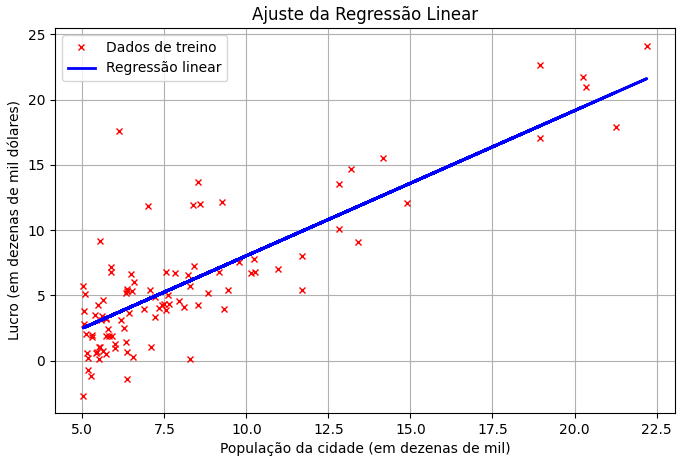
* Impressão de uma matriz identidade 5x5.
* Criação de vetores de 1s.
* Adição de bias aos dados (coluna 1s).
* Produto matricial entre X e theta,
* Cálculo de erros quadráticos e do custo médio.

Esses exercícios preparam a base para a manipulação correta dos dados nas etapas seguintes.

1. **CARREGAMENTO E VISUALIZAÇÃO DOS DADOS**

Após o aquecimento, foi realizado o carregamento do conjunto de dados ex1data1.txt, que contém informações sobre a população de cidades e seus respectivos lucros.

Figura: lucros versus cresciemento populacional.



Fonte: acervo do autor.

Com a utilização da função *plot\_data()*, foi gerado um gráfico de dispersão para visualizar e distribuição dos pontos. Essa visualização foi fundamental para identificar a tendência linear dos dados, justificando a escolha da regressão linear como modelo.

1. **PREPARAÇÃO DOS DADOS**

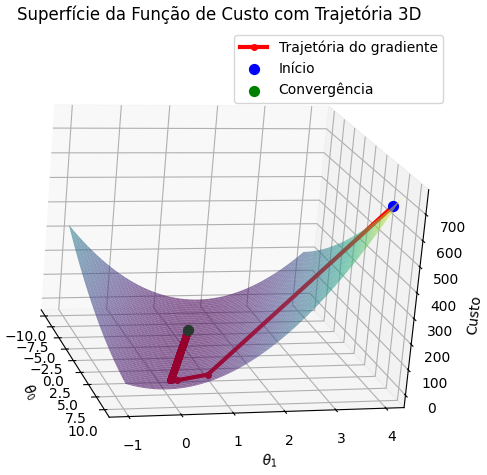
Antes da aplicação do algoritmo de descida do gradiente, foi adicionado uma coluna de 1s aos dados de entrada, criando a matriz x\_aug, necessária para a inclusão do termo de interceptação (θ₀) no modelo. O vetor theta foi inicializado com valores iguais a zero, conforme o padrão em problemas de regressão linear.

1. **TESTE DA FUNÇÃO DE CUSTO**

A função de custo *compute\_cost()* foi testada para validação de seu funcionamento, utilizando dois vetores theta diferentes:

* theta = [0,0] - Custo esperado: aproximadamente 32,07.
* theta= [-1,2] - Custo esperado: aproximadamente 54,24.

Figura: superfície da função de custo.



Fonte: acervo do autor.

1. **EXECUÇÃO DA DESCIDA DO GRADIENTE**

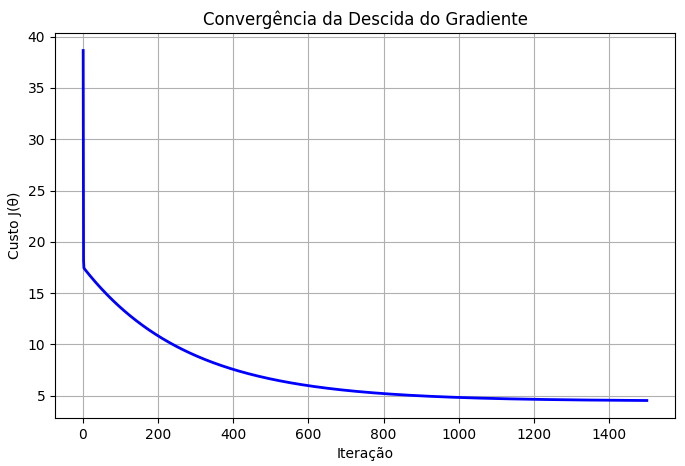
Em seguida, o algoritmo de descida do gradiente foi aplicado utilizando os seguintes parâmetros:

* Taxa de aprendizado (alpha): 0,01
* Número de iterações:1500
* Theta inicial: [8.5,4.0]

Após a execução, foram obtidos valores finais para theta próximos dos valores esperados: aproximadamente [-3.63, 1.16].

Foi gerado um gráfico mostrando a convergência da função de custo ao longo das iterações, evidenciando a diminuição progressiva do custo e validando a efetividade do treinamento.

Figura: convergência da descida do gradiente.



Fonte: acervo do autor.

1. **RESULTADOS**

Foram realizados experimentos para testar o comportamento da função de custo e da descida do gradiente utilizando diferentes valores iniciais de theta [0, 0], [5, 5], [-5,5]. Para cada caso, foi avaliado o custo inicial, a evolução dos parâmetros após o treinamento, e as previsões de lucro para diferentes valores de população.

No primeiro experimento, com theta inicial [0, 0] o produto X @ theta foi novamente [0, 0, 0], e os erros quadráticos foram [25, 81, 169], semelhantes ao segundo experimento. Segundo experimento: o custo médio inicial obtido foi de 45,83. Para [5, 5] também ocorreu um valor de 45,83 e vetor theta de [0, 0, 0], além de erros quadráticos também iguais [25, 81, 169].

Para o terceiro experimento [-5, 5], o produto X @ theta foi [5, 15, 25], erros quadráticos de [0, 36, 144]. O custo médio inicial foi de 30,00 (valor inferior ao dois outros custos).

Apesar das diferenças nos custos iniciais, todas as execuções da descida do gradiente resultaram em parâmetros convergentes para [-3.09, 1.11], assim como as previsões de lucro para populações de 35.000 e 70.000 habitantes, que também se mantiveram em $7.991,24 e $46.931,38, respectivamente.

* Para alpha = 0.01, os parâmetros theta encontrados foram [-3.09, 1.11] (esperado aproximadamente [-3.63, 1.16]). Previsões de lucro: $7991.24 e $46931.38.
* Para alpha = 0.001, theta = [5.25545545 0.27369269] (esperado aproximadamente [-3.63, 1.16]).
* Para alpha = 0.1, apareceram warnings de overflow (overflow encountered in reduce, overflow encountered in square) e depois invalid value encountered in subtract). Parâmetros theta = [nan, nan].

Foi construído um gráfico de superfície 3D, demonstrando como o custo varia em função dos parâmetros.Também foi gerado um gráfico de contorno evidenciando o caminho percorrido pela descida do gradiente até encontrar o mínimo. Essas visualizações proporcionaram uma compreensão mais aprofundada sobre o comportamento da função de custo e o processo de otimização.

OBSERVAÇÕES E SUGESTÕES DE MELHORIAS:

O repositório não estava no padrão de estrutura PEP8 (nomenclatura de diretórios e arquivos de código), o que foi corrigido. Tudo isso pode ser visto no github: <https://github.com/guilherme-rms-cv/regressao-linear-ex1_guilherme>.