UNIVERSIDADE FEDERAL DO MARANHÃO

CAMPUS DE SÃO LUÍS - CIDADE UNIVERSITÁRIA

ENGENHARIA DA COMPUTAÇÃO

REDES NEURAIS

TURMA 1

# **ATIVIDADE DE REGRESSAO LINEAR**

PROFESSOR:THALES LEVI AZEVEDO VALENTE

ALUNOS:

EMANUEL LOPES SILVA - 2021017818

São Luís – MA

29/04/2025

EMANUEL LOPES SILVA - 2021017818

REDES NEURAIS

TURMA 1

# ATIVIDADE DE REGRESSÃO LINEAR

Relatório acadêmico apresentado à disciplina de Redes Neurais, no curso de Engenharia da Computação da Universidade Federal do Maranhão, como parte da avaliação do componente curricular, sob a tutoria do Prof.Dr. Thales Valente

São Luís - MA

29/04/2025

# Sumário

Introdução ........ 5
Contextualização ........ 5
Justificativa ........ **Erro! Indicador não definido.**
Objetivo ........ 6
Fundamentação Teórica ........ 6
Regressão Linear Multivariada ........ 6
Normalização de Features ........ 7
Métodos de Otimização ........ 8
Impactos da Normalização ........ 9
Metodologia ........ 10
Implementações ........ 10
Configurações dos Experimentos ........ 12
Experimentos Realizados ........ 13
Resultados ........ 14
Discussão ........ 19
Conclusão ........ 20

**Resumo:**

**Palavras-Chave:**

**Abstract**

**Keywords**

# Introdução

Este relatório apresenta uma análise comparativa entre os métodos de Gradiente Descendente e Equação Normal para regressão linear multivariada, com base em um conjunto de dados de preços de casas. O fluxo experimental inclui desde o pré-processamento dos dados com diferentes técnicas de normalização (z-score e min-max) até a visualização da convergência do custo e a projeção dos planos de regressão ajustados. A partir da implementação dos modelos, são analisados fatores como custo mínimo obtido, trajetória dos parâmetros, impacto da escala dos dados e tempo de convergência. O objetivo central é compreender como diferentes abordagens influenciam na eficácia e precisão do ajuste dos modelos preditivos, especialmente quando aplicados a dados com múltiplas variáveis correlacionadas.

## Contextualização

A regressão linear multivariada é uma das técnicas fundamentais em modelagem preditiva e análise estatística. Seu princípio consiste em estimar a relação entre uma variável dependente contínua e múltiplas variáveis independentes, assumindo uma relação linear entre elas. Em aplicações práticas como a precificação de imóveis, onde variáveis como área construída e número de quartos influenciam diretamente no valor de mercado de uma propriedade, essa abordagem permite desenvolver modelos capazes de generalizar bem os padrões presentes nos dados históricos.

Contudo, a eficácia dos modelos de regressão depende não apenas da estrutura algébrica adotada, mas também da forma como os dados são tratados antes da modelagem. A normalização das variáveis, por exemplo, desempenha um papel crucial na estabilidade e desempenho computacional do algoritmo de otimização. Em especial, o Gradiente Descendente pode sofrer grandes variações de desempenho dependendo da escala das features, afetando a taxa de convergência e, em casos extremos, levando à não convergência. Nesse sentido, técnicas como a normalização por z-score (padrão) ou por min-max são frequentemente utilizadas para garantir que todas as variáveis estejam em uma mesma ordem de magnitude, evitando que atributos com valores altos dominem o aprendizado.

Além disso, o método de otimização escolhido influencia diretamente na precisão e no custo computacional da modelagem. Enquanto a Equação Normal fornece uma solução exata com base na inversa da matriz de covariância, sendo eficiente para bases pequenas e bem condicionadas, o Gradiente Descendente se mostra mais escalável, especialmente em contextos de alta dimensionalidade ou com grandes volumes de dados. Essa diferença torna relevante a comparação prática entre ambas as abordagens, não apenas em termos de resultados preditivos, mas também em relação ao custo de processamento, tempo de execução e impacto das transformações aplicadas aos dados.

## Objetivo

O código desenvolvido tem como objetivo implementar e comparar duas abordagens para regressão linear multivariada: o Gradiente Descendente (Gradient Descent - GD) e a Equação Normal (Normal Equation - NE). Ambas as técnicas são aplicadas a um conjunto de dados contendo variáveis que influenciam o preço de imóveis, com o intuito de estimar os parâmetros θ que minimizam a função de custo associada. O fluxo completo abrange o carregamento dos dados, normalização das features, treinamento do modelo, avaliação dos custos, realização de predições e geração de visualizações gráficas que permitem analisar o comportamento da função de custo e a trajetória de convergência dos parâmetros.

# Fundamentação Teórica

## Regressão Linear Multivariada

A regressão linear multivariada é uma técnica estatística que busca modelar a relação entre uma variável dependente contínua e duas ou mais variáveis independentes. Trata-se de uma extensão natural da regressão linear simples, que envolve apenas uma variável explicativa. Essa abordagem é amplamente utilizada em problemas de predição, onde se deseja estimar um valor de saída com base em múltiplas características de entrada, como ocorre, por exemplo, na previsão do preço de imóveis a partir de atributos como metragem, número de quartos e localização.

Matematicamente, o modelo de regressão linear multivariada é representado por:

$$h\theta(x) = \theta 0 + \theta 1x1 + \theta 2x2 + \cdots + \theta nxn$$

ou, em forma matricial:

$$h\theta(x) = X\theta$$

em que:

- $X \in R^{m\times(n+1)}$ é a matriz de atributos com m exemplos e n features (incluindo o termo de bias θ0);
- $\theta \in R^{(n+1)}$ é o vetor de parâmetros do modelo;
- $h\theta(x)$ representa a predição do modelo para as entradas xxx.

Para ajustar os parâmetros θ, define-se uma função de custo que quantifica o erro entre os valores previstos e os valores reais. A função de custo mais comum é o Erro Quadrático Médio (MSE):

$$J(\theta) = \frac{1}{2m} \sum_{\{i=1\}}^{\{m\}} (h_\theta(x^{\{(i)\}}) - y^{\{(i)\}})^2$$

O objetivo do treinamento é encontrar o vetor $\theta$ que minimiza essa função de custo. Para isso, podem ser utilizados dois métodos principais:

- **Gradiente Descendente (Gradient Descent)**: abordagem iterativa que ajusta $\theta$ em pequenas etapas na direção do gradiente negativo da função de custo.
- **Equação Normal (Normal Equation)**: solução analítica que fornece diretamente os parâmetros ótimos por meio da seguinte equação:

$$\theta = (X^T X)^{-1} X^T y$$

A escolha entre essas técnicas depende de fatores como o número de atributos, o tamanho da base de dados e a necessidade de escalabilidade. O Gradiente Descendente é indicado para grandes volumes de dados e permite maior flexibilidade no uso de regularização. Já a Equação Normal é mais simples de implementar em bases pequenas, porém computacionalmente mais custosa quando $n$ é muito grande, devido à inversão da matriz $X^T X$.

Além disso, a normalização das variáveis se mostra essencial para garantir o bom funcionamento do Gradiente Descendente, uma vez que variáveis com escalas muito diferentes podem prejudicar a convergência do algoritmo, tornando-o lento ou até instável. Técnicas como a padronização (z-score) e a normalização min-max são amplamente utilizadas para esse fim.

## Normalização de Features

A normalização de features é uma etapa fundamental no pré-processamento de dados para algoritmos de aprendizado de máquina, especialmente aqueles baseados em métodos numéricos iterativos, como o Gradiente Descendente. Essa etapa consiste em transformar os dados de entrada (variáveis independentes) para que todas as features compartilhem uma escala comum. O principal objetivo dessa transformação é evitar que atributos com magnitudes muito diferentes exerçam influência desproporcional sobre o processo de aprendizado, o que pode comprometer tanto a eficiência quanto a precisão do modelo preditivo.

No contexto da regressão linear multivariada, a normalização impacta diretamente a taxa de convergência do algoritmo de otimização. Quando os atributos não estão na mesma ordem de grandeza, o espaço de busca da função de custo tende a se deformar, resultando em trajetórias de otimização instáveis ou extremamente lentas. Ao transformar os dados para uma escala uniforme, esse problema é mitigado, permitindo que o algoritmo percorra o gradiente de forma mais eficiente.

Duas técnicas de normalização são amplamente utilizadas:

1. **Padronização (Z-score normalization)**
   Essa técnica transforma as features de modo que cada uma passe a ter média zero e desvio padrão igual a um. A fórmula utilizada é:

$$x_j' = \frac{\{x_j - \mu_j\}}{\{\sigma_j\}}$$

onde:

- $x_j$ é o valor original da feature $j$,
- $\mu_j$ é a média da feature $j$,
- $\sigma_j$ é o desvio padrão da feature $j$,
- $x'_j$ é o valor padronizado da feature $j$.

2. **Normalização Min-Max**
Essa técnica mapeia os valores da feature para o intervalo [0,1] com base em seu valor mínimo e máximo. A fórmula é:

$$x_j' = \frac{\{x_j - min(x_j)\}}{\{\max(x_j) - \min(x_j)\}}$$

onde:

- $\max(x_j)$ $e$ $\min(x_j)$ são o menor e o maior valor da feature $j$;
- $x_j'$ é o valor normalizado da feature $j$.

## Métodos de Otimização

Os métodos de otimização são algoritmos matemáticos utilizados para encontrar o conjunto de parâmetros que minimizam (ou maximizam) uma determinada função objetivo. Em contextos de aprendizado de máquina e regressão, essa função geralmente corresponde a uma **função de custo**, que expressa o erro entre as previsões feitas pelo modelo e os valores reais observados. O processo de otimização busca, portanto, ajustar os parâmetros do modelo de forma a reduzir esse erro, garantindo maior acurácia e desempenho preditivo.

Nesta seção, serão explicados os métodos mencionados anteriormente na seção de "Regressão Linear Multivariada", a Gradient Descent, e a Equação Normal. A explicação será iniciada pela lógica da Equação Normal, a seguir:

A Equação Normal é uma técnica analítica que fornece uma **solução fechada** para o problema de regressão linear, ou seja, calcula diretamente o vetor θ que minimiza a função de custo, sem a necessidade de iterações. A fórmula é dada por:

$$\theta = (X^T X)^{-1} X^T y$$

Em que:

- $X$ é a matriz de atributos com termo de bias incluído;

- $y$ é o vetor de valores alvo;
- $(X^T)$ é a transposta de $X$;
- $(X^T X)^{-1}$ é a inversa da matriz de covariância.

Embora eficiente para conjuntos de dados pequenos e moderados, esse método se torna computacionalmente caro e instável em datasets com um número elevado de features, devido à necessidade de inversão de matriz, que tem complexidade $O(n^3)$.

Outrossim, O Gradiente Descendente é um método **iterativo** que atualiza os parâmetros do modelo com base no gradiente da função de custo. A cada iteração, o vetor θ é ajustado na direção oposta ao gradiente, reduzindo progressivamente o valor de J(θ). A atualização é realizada segundo a fórmula:

$$\theta := \theta - \alpha \cdot \frac{1}{m} X^T(X\theta - y)$$

Onde:

- $\alpha$ é a taxa de aprendizado, um hiperparâmetro que controla o tamanho do passo na direção do gradiente;
- $X\theta - y$ é o vetor de erros entre as predições e os valores reais.

A principal vantagem do Gradiente Descendente é sua **escalabilidade**, sendo adequado para grandes volumes de dados e situações em que a inversão de matrizes não é computacionalmente viável. No entanto, sua eficácia depende fortemente da **normalização das features** e da escolha adequada da taxa de aprendizado. Uma taxa muito alta pode levar a oscilações ou divergência; uma taxa muito baixa torna o processo lento.

## Impactos da Normalização

A normalização de dados é uma etapa fundamental no pré-processamento de atributos em modelos de aprendizado de máquina, especialmente em algoritmos baseados em otimização iterativa, como o Gradiente Descendente. Seu principal propósito é ajustar a escala das variáveis independentes, tornando-as comparáveis entre si, o que contribui diretamente para a eficiência do treinamento e para a estabilidade numérica do modelo.

Quando os atributos possuem escalas muito distintas, por exemplo, número de quartos variando de 1 a 5 e área construída variando de 500 a 3000, a função de custo assume uma geometria distorcida. Em termos geométricos, isso transforma o espaço de busca do Gradiente Descendente em um vale alongado, dificultando a descida eficiente em direção ao mínimo global. Isso ocorre porque o gradiente da função de custo passa a apontar em direções desalinhadas, fazendo com que o algoritmo execute muitas oscilações antes de convergir.

Ao normalizar os dados, geralmente por padronização (z-score) ou min-max, as variáveis passam a ter magnitude semelhante, fazendo com que os contornos da função de custo se tornem mais simétricos (circulares). Isso permite que o Gradiente Descendente atue com passos mais uniformes em todas as direções, reduzindo o número de iterações necessárias e evitando problemas de divergência. A figura abaixo ilustra essa diferença de comportamento:

- **Sem normalização**: Trajetória irregular e convergência lenta.
- **Com normalização**: Trajetória suave e rápida em direção ao mínimo.

Além de acelerar a convergência, a normalização também melhora a **precisão numérica** dos cálculos matriciais. Modelos como a Equação Normal dependem da inversão da matriz $X^T X$, que pode ser instável ou mal-condicionada caso as features estejam em escalas muito diferentes. Ao normalizar os dados, essa matriz torna-se mais bem condicionada, minimizando erros de arredondamento e aumentando a confiabilidade dos coeficientes estimados.

Por fim, a normalização também contribui para a interpretabilidade dos coeficientes em modelos lineares. Com as variáveis na mesma escala, é possível comparar diretamente os pesos $\theta j$, interpretando quais atributos têm maior influência relativa sobre a variável de saída.

# Metodologia

A metodologia adotada neste trabalho fundamenta-se no desenvolvimento e na execução de um pipeline completo de regressão linear multivariada, com o objetivo de comparar o desempenho entre dois métodos de otimização: o **Gradiente Descendente** (iterativo) e a **Equação Normal** (analítica). Para isso, foram implementadas rotinas em Python que tratam desde a leitura e normalização dos dados até a visualização de resultados e análise de custos. O processo foi cuidadosamente modularizado em arquivos distintos, visando facilitar a organização, reutilização de código e clareza na análise das etapas. A seguir, descrevem-se as responsabilidades e funcionalidades de cada módulo implementado.

## Implementações

Os seguintes arquivos python foram implementados:

- **regressao-multivariada-ex.py**

  É o script principal do experimento. Ele executa todo o fluxo de trabalho de regressão linear multivariada, incluindo:

  - Leitura do conjunto de dados do arquivo ex1data2.txt.

- Aplicação das técnicas de normalização.
- Treinamento do modelo com Gradiente Descendente e Equação Normal.
- Predições com ambos os métodos para uma entrada de teste.
- Comparação entre os custos calculados.
- Geração de gráficos de convergência, superfícies e contornos da função de custo, e visualização tridimensional do plano de regressão ajustado. O script também contém a lógica para calcular os custos ao longo da trajetória do gradiente, além de realizar a conversão de parâmetros para análise visual em escalas normalizadas.

❖ **features_normalize.py**

Contém duas funções de pré-processamento fundamentais:

- features_normalize_by_std(X): aplica padronização por **Z-score**, transformando cada feature para ter média zero e desvio padrão unitário.
- features_normalizes_by_min_max(X): normaliza os dados para o intervalo [0,1], com base nos valores mínimos e máximos de cada coluna.

Ambas as funções retornam a matriz normalizada, o vetor de médias (ou mínimos) e o vetor de desvios padrão (ou máximos), o que permite reverter os valores para escala original caso necessário.

❖ **gradient_descent_multi.py**

Implementa dois métodos relacionados ao Gradiente Descendente:

- gradient_descent_multi(X, y, theta, alpha, num_iters): realiza a otimização iterativa para minimizar o custo J(θ) ao longo de um número fixo de iterações.
- gradient_descent_multi_with_history(...): versão expandida que, além de retornar o vetor final de parâmetros θ, também armazena o histórico completo de θ e dos valores de custo a cada iteração.

Essas implementações são fundamentais para visualizar a trajetória do modelo na superfície de custo e avaliar a convergência do algoritmo.

❖ **normal_eqn.py**

Fornece uma solução analítica para o cálculo dos parâmetros do modelo. A função normal_eqn(X, y) implementa a fórmula clássica da **Equação Normal**:

$$\theta = (X^T X)^{-1} X^T y$$

Utiliza a pseudo-inversa para garantir estabilidade numérica mesmo em casos em que $X^T X$ não é invertível. Essa abordagem é eficiente para datasets de baixa dimensionalidade e serve como referência de comparação para o método iterativo.

- **compute_cost_multi.py**

Contém a função compute_cost_multi(X, y, theta), responsável por calcular a função de custo quadrático médio:

$$J(\theta) = \frac{1}{2m}(X\theta - y)^T(X\theta - y)$$

Ela é usada tanto durante o treinamento por Gradiente Descendente, para acompanhar a convergência, quanto na avaliação final dos modelos obtidos por ambos os métodos.

# Configurações dos Experimentos

Para garantir a reprodutibilidade e a comparabilidade dos resultados obtidos, foram definidas configurações experimentais fixas durante todos os testes. O conjunto de dados utilizado foi o ex1data2.txt, contendo amostras com duas variáveis independentes (metragem do imóvel em pés quadrados e número de quartos) e uma variável dependente (preço da casa, em dólares).

As principais configurações dos experimentos foram:

- **Tamanho do conjunto de dados:** 47 amostras (linhas) com 2 atributos por exemplo.
- **Normalização das features:** Foi aplicada a normalização por **Z-score** (média zero e desvio padrão unitário) como configuração padrão. A normalização min-max foi testada adicionalmente em análises comparativas.
- **Inicialização de parâmetros:** O vetor θ\thetaθ foi inicializado com zeros em todos os experimentos de Gradiente Descendente.
- **Taxa de aprendizado ($\alpha$):** 0,01.
- **Número de iterações:** 400.
- **Ponto de predição de teste:** [1650,3], representando um imóvel de 1650 pés² com 3 quartos.
- **Ferramentas e ambiente:** Python 3.x, NumPy, Matplotlib; sistema local de arquivos para geração de gráficos e resultados.

Para a Equação Normal, os dados foram mantidos em sua escala original, sem normalização, uma vez que esse método não é sensível à escala das variáveis.

## Experimentos Realizados

Os experimentos conduzidos neste trabalho foram estruturados em quatro etapas principais:

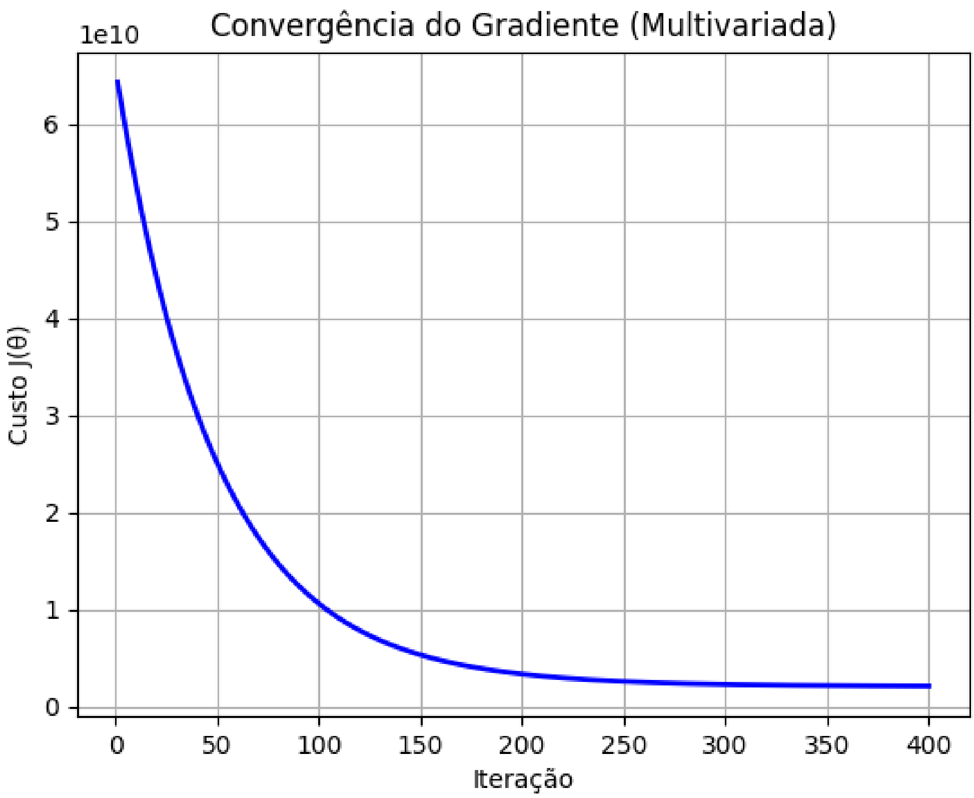
1. **Avaliação do Gradiente Descendente com Dados Normalizados:**
   - O modelo foi treinado usando features normalizadas por Z-score.
   - O vetor de parâmetros θ\thetaθ foi ajustado iterativamente, e o custo J(θ) foi monitorado a cada iteração.
   - Foram gerados gráficos de convergência, mostrando a redução do custo ao longo do tempo.
2. **Predição com o Modelo Treinado:**
   - Utilizou-se o vetor θ obtido para prever o preço de um imóvel com entrada padrão [1650,3], após normalização da entrada com os mesmos parâmetros (μ e σ) do treinamento.
3. **Cálculo de Parâmetros pela Equação Normal:**
   - Os parâmetros θ foram obtidos diretamente por solução fechada.
   - Uma nova predição foi feita com os dados originais (sem normalização) para o mesmo imóvel de teste.
   - Comparou-se o custo obtido por esse método com o do Gradiente Descendente.
4. **Visualizações de Análise e Comparação:**
   - Geração de:
     - Superfície 3D da função de custo J(θ1,θ2);
     - Contornos da função de custo com trajetória do Gradiente Descendente;
     - Plano de regressão ajustado aos dados originais em um gráfico 3D.
   - Essas visualizações permitiram entender geometricamente o processo de convergência e o impacto da normalização.

Adicionalmente, foi testada a aplicação indevida de parâmetros calculados via Equação Normal em dados normalizados, para evidenciar a importância da coerência entre os dados e os coeficientes utilizados. Isso resultou em um custo incorretamente elevado, confirmando a necessidade de atenção à escala dos dados.

# Resultados

A presente seção apresenta e discute os resultados obtidos a partir da aplicação dos métodos de Regressão Linear Multivariada com Gradiente Descendente e Equação Normal. São analisados os coeficientes estimados, a acurácia das previsões, a evolução do custo ao longo das iterações e a visualização geométrica da função de custo. Além disso, são exibidas comparações entre as abordagens implementadas, destacando-se os impactos da normalização das variáveis, a convergência dos algoritmos e a qualidade do ajuste do modelo em relação aos dados originais. Cada resultado é acompanhado por gráficos gerados durante os experimentos, os quais auxiliam na interpretação e validação dos métodos utilizados.

Gráfico 1- Descida do Gradiente



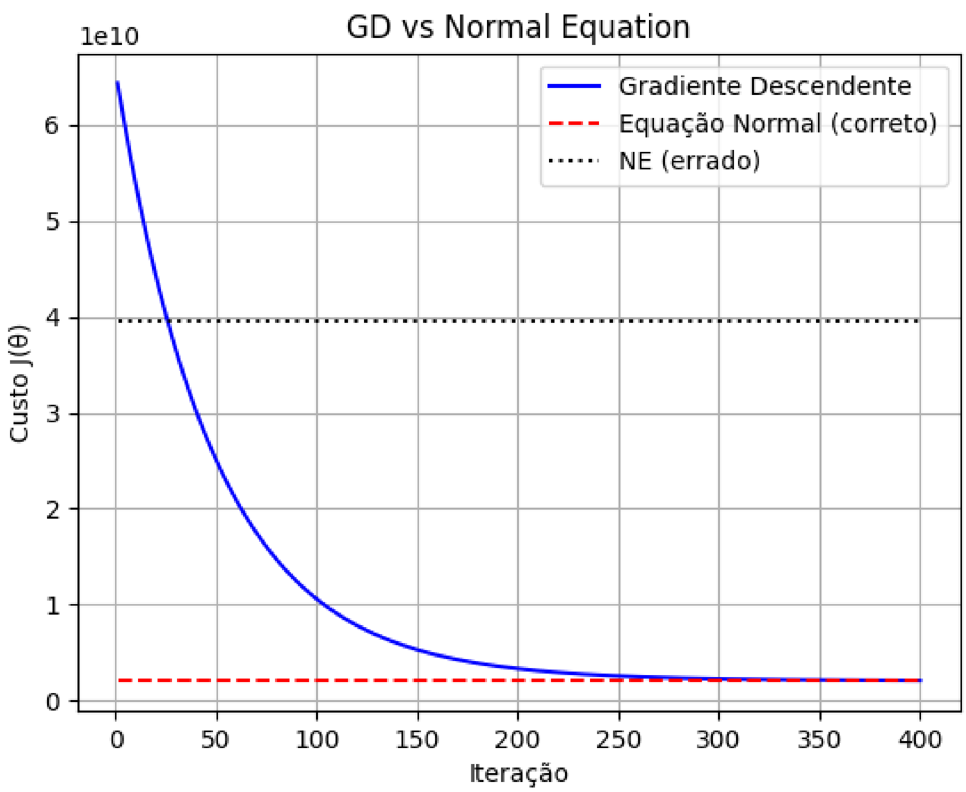

Autoria Própria

O gráfico 1 intitulado *"Convergência do Gradiente (Multivariada)"* representa a evolução do valor da função de custo J(θ) ao longo de 400 iterações do algoritmo de Gradiente Descendente aplicado à regressão linear multivariada. A curva decrescente indica que, à medida que as iterações avançam, o custo é progressivamente reduzido, o que evidencia a efetividade do processo de otimização na busca pelos parâmetros θ\thetaθ que minimizam os erros quadráticos entre as previsões do modelo e os valores reais. No início, observa-se uma queda abrupta, característica de uma boa taxa de aprendizado, que permite uma rápida aproximação da região de ótimo.

Após cerca da centésima iteração, o ritmo de redução desacelera, indicando que o algoritmo está se aproximando da convergência. O valor final de custo é suficientemente baixo, o que demonstra que o modelo ajustado é eficaz para representar os dados de treinamento. Esse comportamento é típico de um processo de aprendizado bem calibrado, onde a escolha da taxa de aprendizado ($\alpha$=0,01) e a normalização das features contribuíram para uma convergência estável e eficiente.

A execução do algoritmo de Gradiente Descendente utilizando uma taxa de aprendizado $\alpha$=0,01 durante 400 iterações resultou em uma estimativa dos parâmetros $\theta$ igual a [334302,06; 99411,45; 3267,01]. Com esses coeficientes, a predição do preço de uma casa com 1650 pés quadrados e 3 quartos foi de aproximadamente **US$289.221,55**, o que demonstra a capacidade do modelo em gerar estimativas plausíveis após o treinamento. O gráfico de convergência correspondente reforça a eficiência do método, ao apresentar uma redução acentuada no valor do custo nas primeiras iterações, seguida de uma estabilização gradual até atingir um mínimo. Esse comportamento é característico de uma convergência bem-sucedida, sugerindo que os parâmetros ajustados proporcionam um bom ajuste ao conjunto de dados.

Gráfico 2- Descida do Gradiente Versus Equação Normal



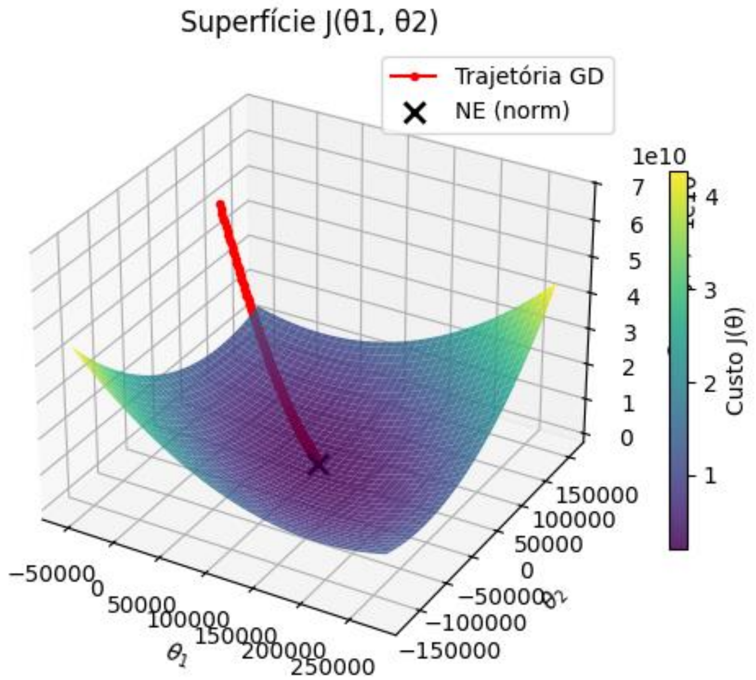

Autoria Própria

O gráfico 2 chamado "Descida do Gradiente Versus Equação Normal" compara a evolução do custo J($\theta$) ao longo das iterações do Gradiente Descendente com os custos obtidos pela aplicação da Equação Normal. A curva azul representa a convergência do

Gradiente Descendente, iniciando com um custo elevado e decaindo exponencialmente até estabilizar próximo ao valor mínimo. A linha vermelha tracejada indica o custo obtido ao aplicar corretamente a Equação Normal nos dados originais (sem normalização), evidenciando um valor constante e baixo, o que reforça a precisão do método analítico.

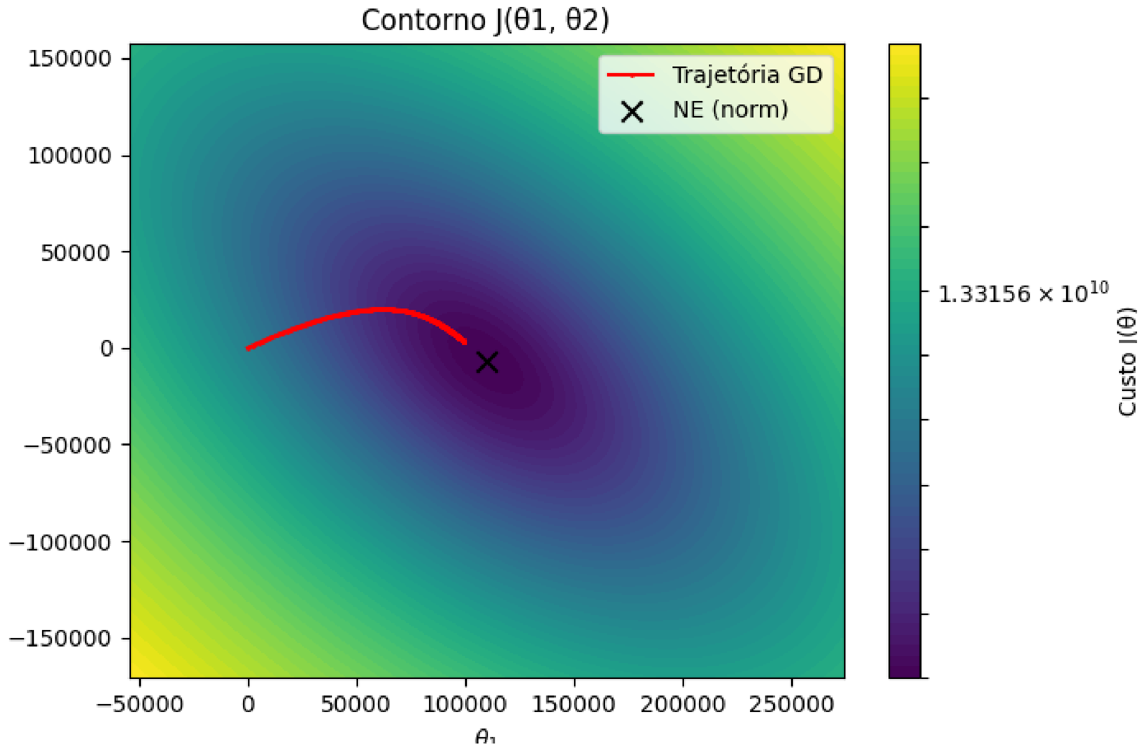
Já a linha preta pontilhada representa o custo incorretamente calculado ao aplicar os parâmetros da Equação Normal em dados normalizados, resultando em um custo elevado devido à incompatibilidade de escala entre os dados e os coeficientes. Os coeficientes obtidos via Gradiente Descendente foram [334302.06,99411.45,3267.01, enquanto pela Equação Normal foram [89597.91,139.21,−8738.02]. As previsões de preço para uma casa com 1650 pés² e 3 quartos foram próximas: **$289.221,55** com Gradiente Descendente e **$293.081,46** com Equação Normal, o que demonstra que ambos os métodos, quando aplicados corretamente, convergem para soluções similares, ainda que por abordagens distintas.

Gráfico 3 - Gráfico de Contorno da Função J em 3D




Autoria Própria

Gráfico 4- Gráfico de Contorno da Função J em 2D




Autoria Própria

Os dois gráficos 3 e 4 acima representam visualizações complementares da função de custo J(θ) em relação aos parâmetros $\theta_1$ e $\theta_2$ no processo de regressão linear multivariada. O primeiro gráfico, tridimensional, mostra a superfície da função de custo, com a trajetória do Gradiente Descendente (GD) em vermelho e o ponto de mínimo global identificado pela Equação Normal (NE) em preto. A curva vermelha descreve como o algoritmo GD ajusta iterativamente os valores dos coeficientes em direção ao mínimo. O segundo gráfico, em vista de contorno (2D), projeta essa superfície, permitindo uma análise mais clara da convergência dos parâmetros, evidenciada pela aproximação do traçado vermelho até o ponto central escuro, que representa o menor custo alcançado.

O primeiro gráfico apresentado mostra o plano de regressão ajustado sobre os dados originais em um espaço tridimensional, onde os eixos representam respectivamente o tamanho do imóvel (em pés²), o número de quartos e o preço previsto (em dólares). A superfície plana colorida foi construída a partir dos coeficientes θ obtidos via Gradiente Descendente (GD), sendo previamente convertidos da escala normalizada de volta para a escala original dos dados. Os pontos vermelhos no gráfico representam as amostras reais de treinamento, e a boa aderência visual entre o plano e os dados indica que o modelo conseguiu capturar bem a tendência linear existente entre as variáveis explicativas e o preço do imóvel.

Esse gráfico é particularmente importante porque traduz visualmente a função de hipótese aprendida pelo modelo e demonstra que a regressão linear multivariada, mesmo sendo um modelo simples, foi capaz de representar a estrutura dos dados de maneira coerente. A superfície gerada corresponde à equação:

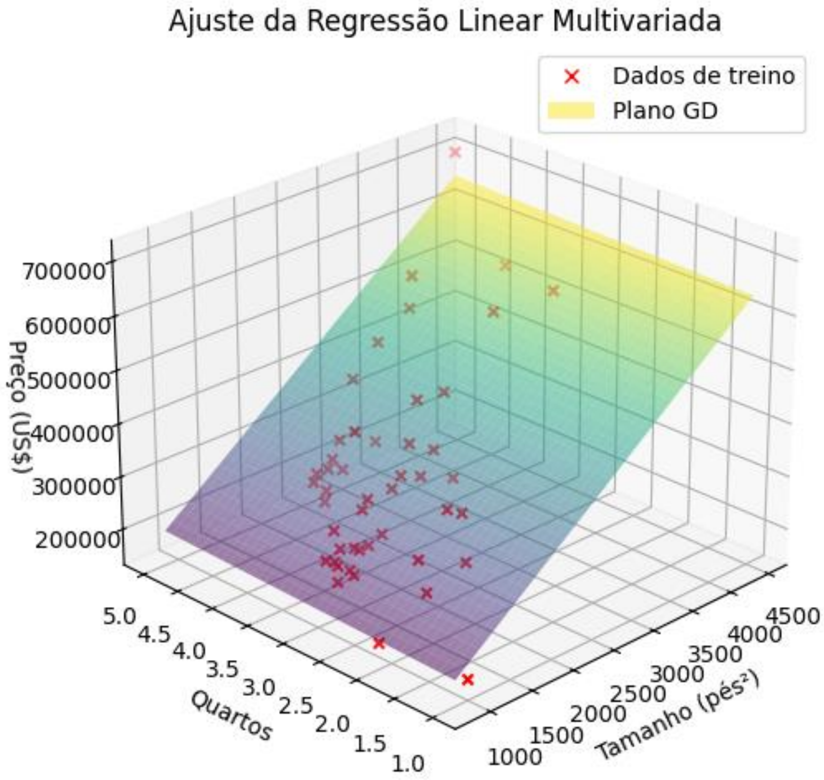
$$\hat{y} = \theta_0 + \theta_1 \cdot tamanho + \theta_2 \cdot quartos$$

A inclinação e orientação do plano refletem os pesos atribuídos a cada feature, e mostram que o tamanho do imóvel exerce maior influência sobre o preço final do que o número de quartos, o que está de acordo com os valores de θ encontrados e com a intuição do problema.

O segundo gráfico, o Gráfico 4, chamado de gráfico de contorno da função de custo J(θ1,θ2) fornece uma visão bidimensional da superfície de otimização no espaço dos parâmetros do modelo. As curvas coloridas representam linhas de mesmo valor de custo, com as cores mais escuras indicando regiões de menor custo. O ponto marcado com um X preto corresponde à solução encontrada pela Equação Normal (NE) com parâmetros convertidos para a escala normalizada, enquanto a linha vermelha mostra a trajetória do algoritmo de Gradiente Descendente (GD) ao longo das iterações.

Esse gráfico evidencia como o gradiente descendente aproxima-se progressivamente do ponto de mínimo global da função de custo, partindo de uma inicialização arbitrária até convergir para a região escura central. A trajetória curva do GD é consequência da escala diferente entre as features, mesmo após a normalização, e da topologia elíptica da função de custo, típica da regressão linear com múltiplas variáveis. A proximidade final entre o ponto de chegada do GD e o ponto estimado pela NE reforça a consistência entre os dois métodos e valida a implementação realizada.

Gráfico 5 – Ajuste da Regressão Linear Multivariada




Autoria Própria

O gráfico 5 acima apresenta o ajuste da Regressão Linear Multivariada em um espaço tridimensional. Os pontos vermelhos (x) representam os dados reais de treinamento, ou seja, observações compostas por número de quartos, tamanho do imóvel (em pés²) e o respectivo preço. Já o plano colorido corresponde à superfície de predição gerada a partir dos coeficientes $\theta$ obtidos via Gradiente Descendente, projetado no espaço tridimensional.

Esse gráfico é crucial para visualizar como o modelo ajusta-se aos dados. A proximidade dos pontos em relação à superfície indica que o modelo foi capaz de capturar a tendência geral do comportamento do preço em função das variáveis explicativas. Pequenos desvios verticais entre os pontos e o plano indicam os erros de predição, enquanto a orientação inclinada da superfície reflete a sensibilidade do preço com relação ao aumento do tamanho e da quantidade de quartos.

# Discussão

Ambos os métodos demonstraram capacidade de estimar coeficientes realistas e gerar previsões coerentes com os dados, sendo suas principais distinções observadas na forma de convergência, sensibilidade à escala dos dados e custo computacional.

A execução do algoritmo de Gradiente Descendente mostrou-se eficiente, especialmente quando precedida pela normalização das variáveis, fator essencial para garantir a estabilidade numérica e a convergência adequada. Isso é evidenciado pelo gráfico de convergência, no qual o custo $J(\theta)$ apresentou uma queda exponencial nas primeiras iterações e estabilizou após aproximadamente 300 passos. Esse comportamento indica que a taxa de aprendizado ($\alpha$=0,01) foi bem escolhida e que a normalização foi eficaz em tornar o problema de otimização bem condicionado.

Ao comparar os métodos GD e NE, observou-se que ambos convergem para soluções próximas, conforme indicam os valores similares de $\theta$ e as previsões para o mesmo exemplo de entrada. No entanto, a Equação Normal se destacou pela obtenção direta de uma solução analítica com custo significativamente menor, sem necessidade de múltiplas iterações. Ainda assim, sua sensibilidade à escala dos dados ficou evidente no experimento de custo calculado incorretamente com variáveis normalizadas, resultando em um erro elevado. Isso reforça a importância de manter a consistência entre os dados e os coeficientes ao aplicar o modelo.

As visualizações tridimensionais e de contorno da função de custo $J(\theta1,\theta2)$ enriqueceram a análise ao ilustrar geometricamente o processo de otimização. A trajetória do Gradiente Descendente seguiu um caminho curvo até o mínimo global, o que é característico de funções com curvatura elíptica, e sua aproximação ao ponto obtido pela Equação Normal validou empiricamente a implementação de ambos os métodos.

Por fim, a visualização do plano de regressão ajustado sobre os dados reais revelou um bom ajuste do modelo linear às tendências dos dados. A inclinação do plano refletiu corretamente o maior impacto do tamanho da casa no preço, em comparação ao número de quartos, como também indicado pelos coeficientes aprendidos.

# Conclusão

A aplicação da Regressão Linear Multivariada utilizando as abordagens de Gradiente Descendente e Equação Normal permitiu compreender, na prática, as diferenças metodológicas, computacionais e conceituais entre métodos iterativos e soluções analíticas. Ambos se mostraram eficazes na tarefa de prever o preço de imóveis a partir de variáveis como tamanho e número de quartos, com resultados similares em termos de previsão e ajuste do modelo.

O Gradiente Descendente destacou-se por sua flexibilidade e por possibilitar uma análise detalhada da evolução do processo de otimização, sendo sensível à escolha da taxa de aprendizado e à escala das features. Já a Equação Normal apresentou desempenho superior em termos de velocidade de execução e simplicidade computacional, desde que aplicada diretamente aos dados em sua escala original.

Os experimentos confirmaram a importância da normalização de variáveis para algoritmos baseados em otimização iterativa, e reforçaram a necessidade de atenção à consistência entre escala dos dados e parâmetros. As visualizações tridimensionais e bidimensionais da função de custo e do plano de regressão contribuíram significativamente para a interpretação dos resultados e validação dos modelos.