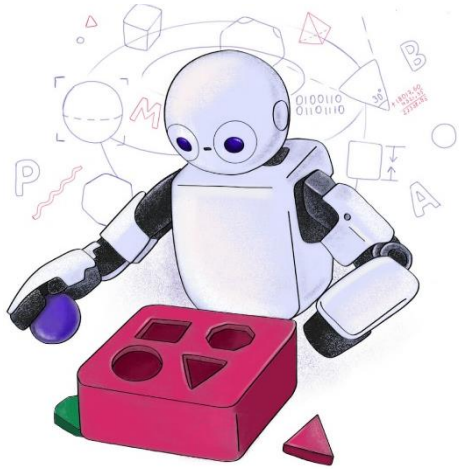
T319 - Introdução ao Aprendizado de Máquina: *Regressão Linear (Parte I)*



Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

Motivação

- **Exemplo 1**: Estimar o preço de casas.
- **Exemplo 2**: Estimar as vendas de sorvete.

500 m^2

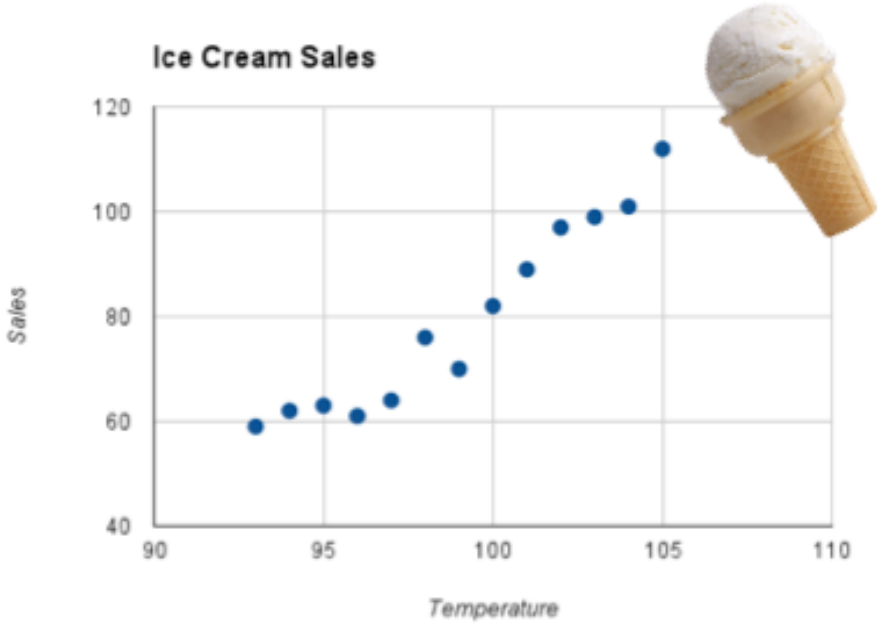
R$ 1.000.000,00

70 m^2

R$ 200.000,00

200 m^2

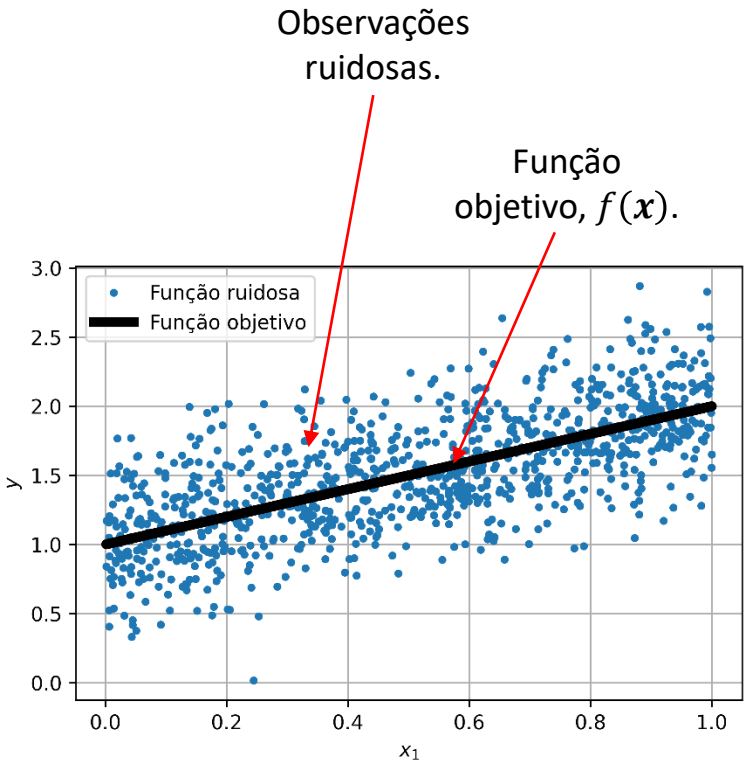
???



- Podemos encontrar uma relação matemática entre a área, localização, n° de quartos de uma casa e seu valor?
- Ou entre a temperatura e a quantidade de sorvetes vendidos?

Regressão Linear

- É um dos mais antigos e conhecidos algoritmos de aprendizado de máquina.
- Vai nos dar várias **intuições** importantes para o **entendimento** de outros **algoritmos mais complexos**, como, por exemplo, **classificadores** e **redes neurais**.
- **Objetivo**: encontrar uma função, $h(\boldsymbol{x})$, que **mapeie**, de **forma ótima**, os **atributos**, $\boldsymbol{x}$, em uma variável de saída $\hat{y} = h(\boldsymbol{x})$, de tal forma que $h(\boldsymbol{x})$ seja uma boa aproximação da ***função verdadeira*** ou ***objetivo***, $f(\boldsymbol{x})$.
- $f(\boldsymbol{x})$ é muitas vezes **desconhecida** e temos acesso apenas a **observações ruidosas**.
- Regressão também é conhecida como ***aproximação de funções*** ou ***ajuste de curvas***.
- Como encontramos uma função, $h(\boldsymbol{x})$, que ***aproxime*** $f(\boldsymbol{x})$ de forma ótima a partir de dados ruidosos?

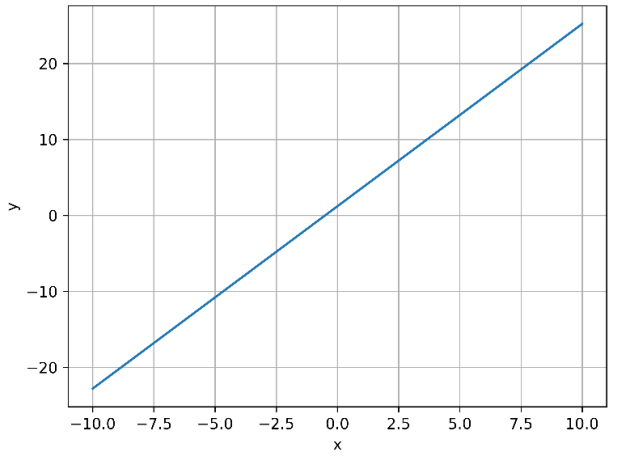


Temos $\boldsymbol{x}$ (atributos) e $\boldsymbol{y}$ (rótulos) e queremos encontrar $\hat{y} = h(\boldsymbol{x})$.

Que tipo de aprendizado?

Regressão Linear

$h(\boldsymbol{x}) = a_0 + a_1 x_1$

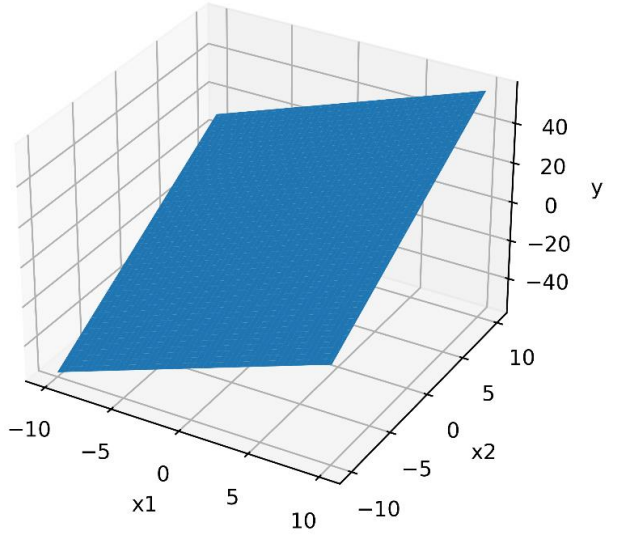


- **Qual forma deve ter a função** $h(\boldsymbol{x})$? Os modelos mais simples são:
  - Com apenas um atributo, x_1, $h(\boldsymbol{x})$ é uma reta, $h(\boldsymbol{x}) = a_0 + a_1 x_1$.
  - Com dois atributos, x_1 e x_2, $h(\boldsymbol{x})$ é uma superfície 2D (ou seja, um plano), $h(\boldsymbol{x}) = a_0 + a_1 x_1 + a_2 x_2$.
  - E assim por diante.
- **Modelo geral**: Equação de um ***hiperplano***

$$h(\boldsymbol{x}) = a_0 + a_1 x_1 + \cdots + a_K x_K = a_0 + \sum_{i=1}^{K} a_i \, x_i.$$

- Existem outros modelos, os quais veremos mais adiante.
- Na literatura, a função $h(\boldsymbol{x})$, é chamada de ***função hipótese***, pois é uma das **possíveis soluções** encontradas no ***espaço de hipóteses***, $\boldsymbol{H}$, para aproximar $f(\boldsymbol{x})$.
- **Espaço de hipóteses**: é conjunto de todas as possíveis ***funções hipótese***.
  - Superfície formada por todos os possíveis valores dos parâmetros, $a_k, \forall k$ quando substituídos em $h(\boldsymbol{x})$.

$h(\boldsymbol{x}) = a_0 + a_1 x_1 + a_2 x_2$

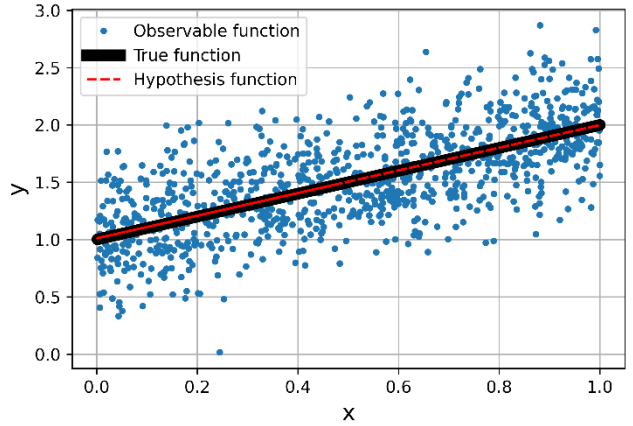


Regressão Linear

- Agora que temos uma forma para $h(\boldsymbol{x})$, podemos refinar o objetivo da regressão um pouco mais.
- **Objetivo**: ***Encontrar*** os ***parâmetros***, também chamados de ***pesos***, a_0, a_1, …, a_K de tal forma que $h(\boldsymbol{x})$ seja uma **aproximação ótima** de $f(\boldsymbol{x})$.
- **Aprendizado supervisionado**: atributos, $\boldsymbol{x}$, mais rótulos/objetivos, y.
- A ***regressão*** é chamada de ***linear*** porque a variável de saída, y, é modelada como sendo uma **combinação linear** dos ***atributos***, $\boldsymbol{x}$.
  - **OBS**.: ***Linear***, nesse contexto, significa "*linear com relação aos pesos*" e não com relação aos ***atributos***, i.e., $\boldsymbol{x}$. Desta forma, os seguintes modelos também são lineares com relação aos pesos:
    - ✓ $y = a_0 + a_1 \log x_1 + a_2 \cos x_2$
    - ✓ $y = a_0 + a_1 e^{x_1}$
    - ✓ $y = a_0 + a_1 x_1^2$
- Exemplo de um modelo não-linear: $y = \dfrac{a_0 x_1}{a_1 + x_1}$.

Não consigo expressar a equação como uma combinação linear dos atributos.

Definição Formal do Problema



O problema da ***regressão linear*** pode ser definido da seguinte forma:

- **Dados disponíveis**:
  - Conjunto de N observações (conjunto de pares de treinamento) : $\{\boldsymbol{x}(i), y(i)\}$, $i = 0, \dots, N-1$, onde
    - $\boldsymbol{x}(i) \in \mathbb{R}^{K+1\times 1}$: *i*-ésimo vetor de atributos de entrada com dimensão $K+1 \times 1$, ou sejam $K+1$ atributos.
    - $y(i) \in \mathbb{R}$: *i*-ésimo valor esperado de saída referente ao vetor de entrada $\boldsymbol{x}(i)$.
- **Modelo**:

Forma vetorial.

$$\hat{y}(i) = h(\boldsymbol{x}(i)) = a_0 + a_1 x_1(i) + \cdots + a_K x_K(i) = \boldsymbol{a}^T \boldsymbol{x}(i),$$

onde $\boldsymbol{a} = [a_0, \dots, a_K]^T$ e $\boldsymbol{x}(i) = [1, x_1(i), \dots, x_K(i)]^T$.

- $\boldsymbol{a}$ é um vetor coluna com dimensão $(K+1 \times 1)$ contendo os ***pesos*** da ***função hipótese*** e $\boldsymbol{x}(i)$ é um vetor coluna com dimensão $(K+1 \times 1)$ contendo os *i*-ésimos valores dos ***atributos***.
- a_0 é o ***coeficiente linear***, ou seja, é o valor de $h(\boldsymbol{x})$ que intercepta o eixo y, a_0 é conhecido também como ***intercept*** ou ***bias***.
- Como a_0 não tem um ***atributo*** relacionado a ele, para facilitar o modelamento matemático, criamos um atributo falso, x_0, com valor constante sempre igual a 1, i.e., $x_0 = 1$.

- **Objetivo do modelo**: encontrar o vetor de pesos $\boldsymbol{a}$ que minimize o ***erro***, dado por uma ***função de erro***, $J_e(\boldsymbol{a})$, entre a aproximação $\hat{y}(i)$ e o valor desejado $y(i)$ para ***todos os exemplos do conjunto***.

$$\min_{\boldsymbol{a}} J_e(\boldsymbol{a})$$

Ou seja, o treinamento do modelo envolve a minimização de uma ***função de erro.***

- Portanto, precisamos definir uma ***função de erro***.

Função de Erro

- **Função de erro**: existem várias possibilidades para se definir a **função de erro** a ser minimizada, porém, geralmente, utiliza-se o **erro quadrático médio**

$$J_e(\boldsymbol{a}) = \frac{1}{N}\sum_{i=0}^{N-1}(y(i) - \hat{y}(i))^2 = \frac{1}{N}\sum_{i=0}^{N-1}\left(y(i) - h(\boldsymbol{x}(i), \boldsymbol{a})\right)^2,$$

Erro entre a saída da função hipótese e a saída esperada.

que nada mais é do que a ***média aritmética do quadrado do erros***.

- Nós veremos mais adiante a razão pela qual o ***erro quadrático médio*** é utilizado.
- A ***função de erro*** pode ser reescrita em forma matricial como

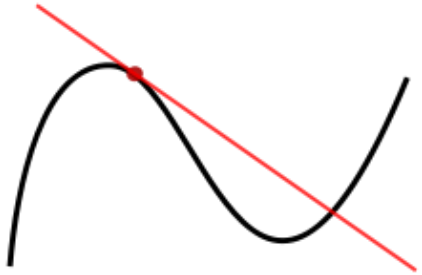
$$J_e(\boldsymbol{a}) = \frac{1}{N}\|\boldsymbol{y} - \boldsymbol{\Phi a}\|^2,$$

onde $\boldsymbol{y} = [y(0), \dots, y(N-1)]^T$ é um vetor $(N \times 1)$, $\boldsymbol{\Phi} = [\boldsymbol{x}(0), \dots, \boldsymbol{x}(N-1)]^T$ é uma matriz $(N \times K+1)$ e N é o número de exemplos ou observações.

- Então, para encontrar o vetor de pesos $\boldsymbol{a}$ devemos minimizar a função de erro:

$$\min_{\boldsymbol{a} \in \mathbb{R}^{K+1}} \|\boldsymbol{y} - \boldsymbol{\Phi a}\|^2 .$$

Minimizando a Função de Erro



A linha vermelha é **tangencial** à curva no ponto marcado pelo círculo vermelho. Sua **inclinação** é a **derivada** naquele ponto.

Como encontramos o mínimo da função de erro em relação aos pesos?

- Da disciplina de cálculo, sabemos que derivando a ***função de erro***, $\|\boldsymbol{y} - \boldsymbol{\Phi a}\|^2$, em relação a $\boldsymbol{a}$ e igualando a 0 nós encontramos o ***ponto*** onde a ***inclinação*** de uma reta ***tangente*** à ***função de erro*** é nula:

$$\frac{\partial \|\boldsymbol{y} - \boldsymbol{\Phi a}\|^2}{\partial \boldsymbol{a}} = 2\boldsymbol{a}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} - 2\boldsymbol{y}^T \boldsymbol{\Phi} = 0,$$

porém, esse ***ponto*** pode ser tanto um ***mínimo*** quanto um ***máximo*** da ***função de erro***, pois em ambos os pontos a ***inclinação*** da reta tangente é ***nula***.

Então, como sabemos se o ponto encontrado é um mínimo ou um máximo?

- Se a inclinação da tangente é nula e a ***derivada de segunda ordem*** for ***positiva***, então o ponto nos dá o mínimo da função,

$$\frac{\partial^2 \|\boldsymbol{e}\|^2}{\partial^2 \boldsymbol{a}} = 2\boldsymbol{\Phi}^T \boldsymbol{\Phi}.$$

- Se a matriz $\boldsymbol{\Phi}$ tiver ***posto*** igual a $K + 1$, então a matriz $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ é ***positiva semi-definida*** e, portanto, o ponto encontrado acima é realmente o ponto de mínimo da ***função de erro***.
  - **Posto de uma matriz**: é o número de linhas ou colunas linearmente independentes da matriz.
  - Uma matriz quadrada $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ é ***positiva semi-definida*** se $\boldsymbol{x}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi x} = \|\boldsymbol{\Phi x}\|^2 \geq 0, \forall \boldsymbol{x} \neq 0.$

Minimizando a Função de Erro

- Portanto, voltando à equação da derivada parcial de primeira ordem igual a 0, temos

$$\boldsymbol{a}^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{y}^T \boldsymbol{\Phi}.$$

- Após aplicarmos o transposto a ambos os lados e isolando $\boldsymbol{a}$ temos

$$\boldsymbol{a} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{y}.$$

- Essa equação é conhecida como a ***equação normal*** e nos dá a ***solução ótima*** em relação a minimização do ***erro quadrático médio*** para esse sistema de equações.

Observações:

1. O método encontra uma ***solução única*** se e somente se a matriz quadrada, $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$, for ***invertível*** (se ela for ***não-singular***), ou seja, com ***posto*** igual a $K + 1$.
2. O método só funciona para sistemas ***determinados*** ou ***sobredeterminados***, ou seja, quando o número de equações (i.e., pares $\boldsymbol{x}$ e y) é igual ou maior do que o número de incógnitas (i.e., pesos), ou seja, $N \geq K + 1$.
3. Para sistemas ***subdeterminados***, ou seja, que têm menos equações do que incógnitas, a matriz $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ tem ***posto*** menor do que $K + 1$ e, portanto, é ***singular*** (ou seja, a matriz $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ não tem uma inversa). Neste caso, não existe solução ou ela não é única.

Superfície de Erro

- E se plotarmos a função de erro, $J_e(\boldsymbol{a})$, em função dos pesos, $\boldsymbol{a}$?

$$J_e(\boldsymbol{a}) = \frac{1}{N}\sum_{i=0}^{N-1}(y(i) - \hat{y}(i))^2 = \frac{1}{N}\sum_{i=0}^{N-1}\left(y(i) - h(\boldsymbol{x}(i), \boldsymbol{a})\right)^2.$$

- Que forma vocês acham que ela terá?
- $J_e(\boldsymbol{a})$ faz o **mapeamento** entre cada possível valor dos **pesos** do modelo e o erro correspondente:
  - $J_e(\boldsymbol{a})$: $\mathbb{R}^{K+1} \rightarrow \mathbb{R}$. Esse mapeamento define o que conhecemos como ***superfície de erro***.
- $J_e(\boldsymbol{a})$ assume uma forma ***quadrática*** com respeito ao ***vetor de pesos***, $\boldsymbol{a}$.

$$J_e(\boldsymbol{a}) = \|\boldsymbol{y} - \boldsymbol{\Phi a}\|^2 = \boldsymbol{y}^T\,\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{\Phi a}^T - \boldsymbol{a}^T\boldsymbol{\Phi}^T\,\boldsymbol{y} + \boldsymbol{a}^T\boldsymbol{\Phi}^T\boldsymbol{\Phi a}.$$

Termo quadrático.

- Consequentemente, a superfície é ***convexa*** (tem forma de ***tigela***) e, portanto, possui um único ***mínimo*** (***global***), que é encontrado, por exemplo, pela ***equação normal***.
- Este é o motivo de usarmos o ***erro quadrático médio*** como ***função de erro***.
- Isso é provado mostrando-se que $\frac{\partial^2 J_e(\boldsymbol{a})}{\partial^2 \boldsymbol{a}} = 2\boldsymbol{\Phi}^T\boldsymbol{\Phi}$ é uma ***matriz positiva semi-definida***, e portanto, $J_e(\boldsymbol{a})$ sempre será **convexa** com relação ao vetor de pesos, $\boldsymbol{a}$.

Superfície de Erro: Exemplo #1

- A figura ao lado mostra a ***superfície de erro*** para a seguinte ***função observável***:

$$y_{\text{noisy}}(n) = y(n) + w(n),$$

onde $w(n) \sim N(0,0.1)$ e $y(n)$ é a ***função objetivo***.

- Neste exemplo, a ***função objetivo*** (ou ***modelo gerador***) é dada por:

$$y(n) = a_1 x_1(n),$$

onde $x_1(n) \sim U(0,1)$ e $a_1 = 1$.

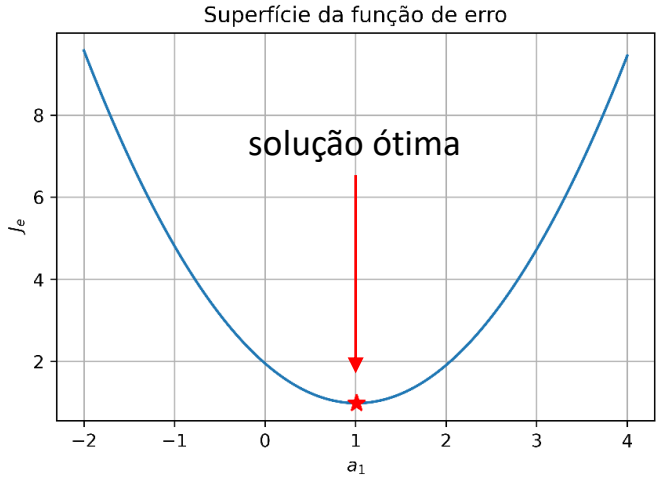
- A ***função hipótese***, $h(\boldsymbol{x})$, é dada por

$$h(\boldsymbol{x}) = \hat{y} = \widehat{a_1} x_1(n).$$

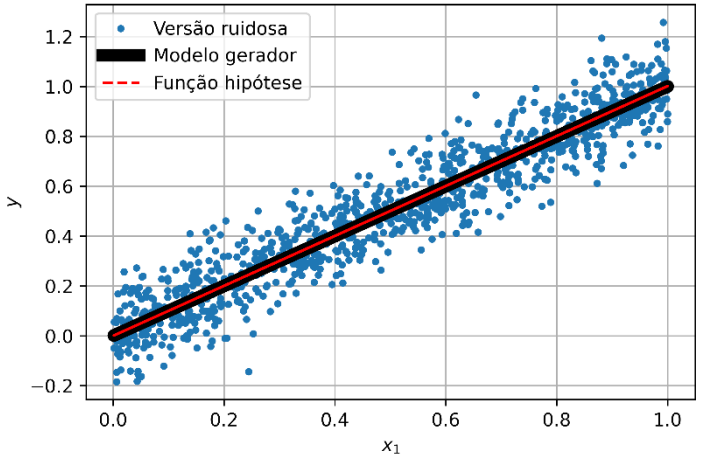
- O erro ***erro quadrático médio*** é calculado como

$$J_e(a_1) = \frac{1}{N}\sum_{n=0}^{N-1}\left(y_{\text{noisy}}(n) - \hat{y}(n)\right)^2 = \frac{1}{N}\sum_{n=0}^{N-1}\left(y_{\text{noisy}}(n) - \widehat{a_1} x_1(n)\right)^2.$$

Observem que temos **apenas um ponto de mínimo**. Isto se deve à superfície ser **convexa**.



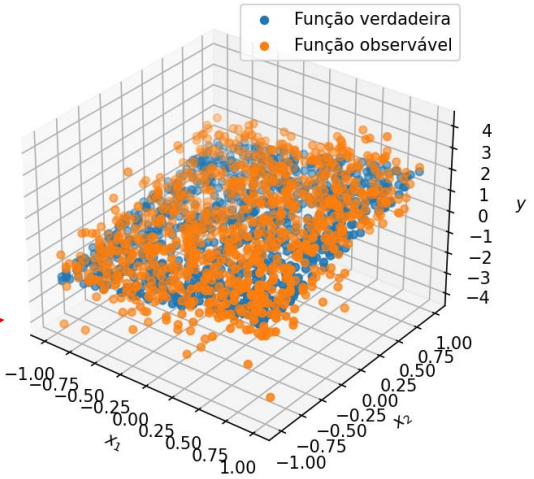
Comparação do **modelo gerador** com a **versão ruidosa** e a **aproximação** obtida com a equação normal.



IMPORTANTE: Os valores de erro para plotarmos a superfície de erro são calculados variando-se a_1 na equação do EQM.

Exemplo: error_surface_example1.ipynb

Superfície de Erro: Exemplo #2



- A figura ao lado mostra a ***superfície de erro*** para a seguinte ***função observável***:

$$y_{\text{noisy}}(n) = y(n) + w(n),$$

onde $w(n) \sim N(0,1)$ e $y(n)$ é a ***função objetivo***.

- Neste exemplo, a ***função objetivo*** (ou ***modelo gerador***) é dada por:

$$y(n) = a_1 x_1(n) + a_2 x_2(n),$$

onde $x_1(n)$ e $x_2(n) \sim U(-1,1)$ e $a_1 = a_2 = 1$.

- A ***função hipótese***, $h(\boldsymbol{x})$, é dada por

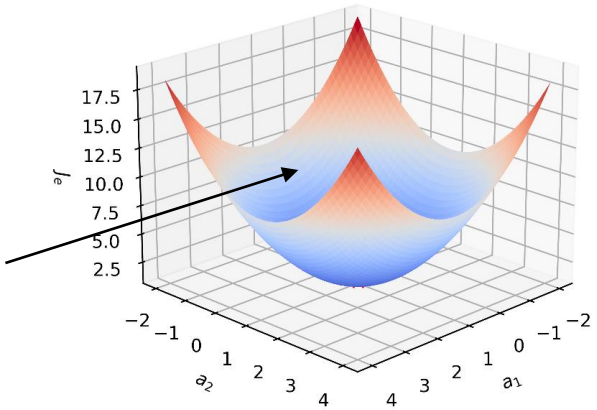
$$h(\boldsymbol{x}) = \hat{y} = \widehat{a_1} x_1(n) + \widehat{a_2} x_2(n).$$

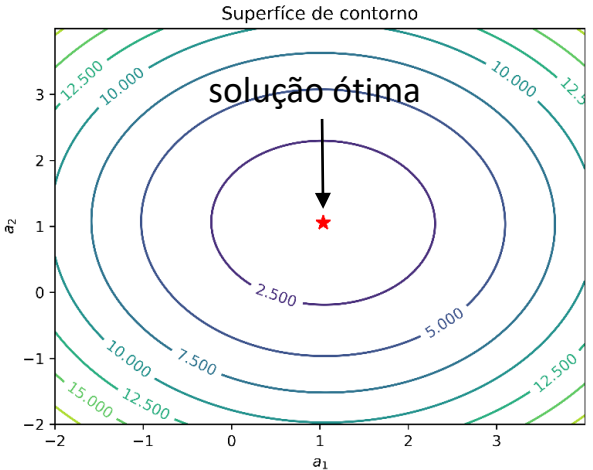
- O erro ***erro quadrático médio*** é calculado como

$$J_e(a_1, a_2) = \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - \hat{y}(n)\right)^2$$

$$= \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - (\widehat{a_1} x_1(n) + \widehat{a_2} x_2(n))\right)^2$$

O erro é calculado variando-se a_1 e a_2 na equação do EQM. Neste caso, a superfície será representada por uma matriz de 3 dimensões. Cada par de valores a_1 e a_2 corresponde a um erro.





- A segunda figura mostra a **superfície de contorno**.
  - Uma linha de contorno de uma função de duas variáveis é uma curva ao longo da qual a função tem um valor constante. Ou seja, no nosso caso, cada uma das linhas indica uma curva que têm o mesmo erro.

Exemplo: error_surface_example2.ipynb

Formatos diferentes para a superfície de erro

Sejam a seguinte função objetivo:

$$y(n) = a_1 x_1(n) + a_2 x_2(n),$$

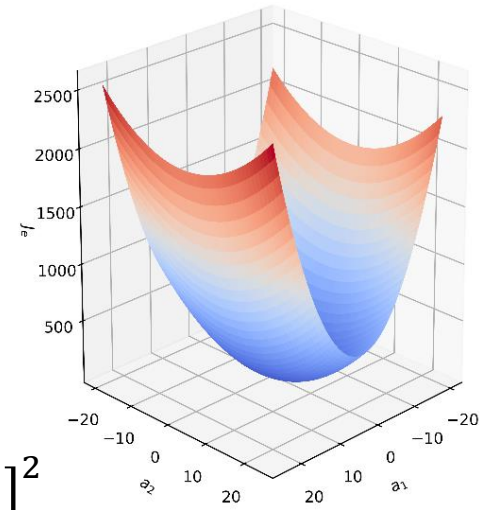
onde $a_1 = 1$, $a_2 = 1$ e função observável

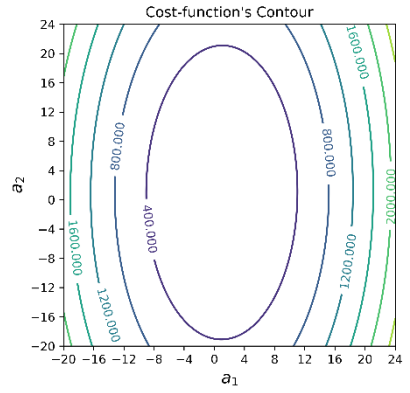
$$y_{\text{noisy}}(n) = y(n) + w(n).$$

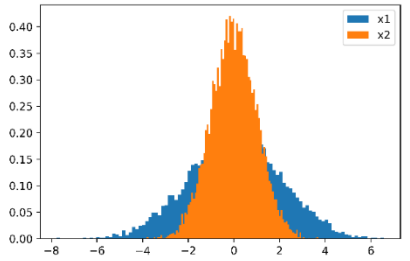
Para plotar a superfície de erro usamos:

$$J_e(\boldsymbol{a}) = \frac{1}{N}\sum_{n=0}^{N-1}\left[y_{\text{noisy}}(n) - (\widehat{a_1}x_1(n) + \widehat{a_2}x_2(n))\right]^2$$

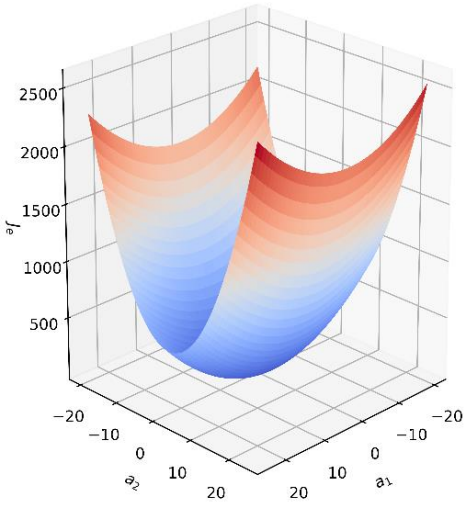
- Se x_1 varia entre um intervalo muito grande de valores, então o ***peso*** da variação de $\widehat{a_1}$ no erro é maior, ou seja, o erro varia mais rapidamente com variações de $\widehat{a_1}$, resultando num vale.
- A mesma coisa pode ser dita para x_2 e $\widehat{a_2}$ (vale).
- Quando x_1 e x_2 têm intervalos semelhantes, então, a variação tanto de $\widehat{a_1}$ quanto de $\widehat{a_2}$ tem ***pesos*** semelhante na variação do erro (tigela).

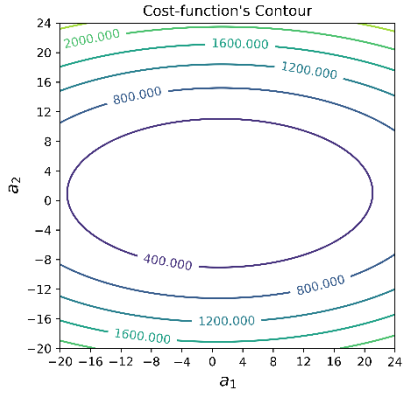


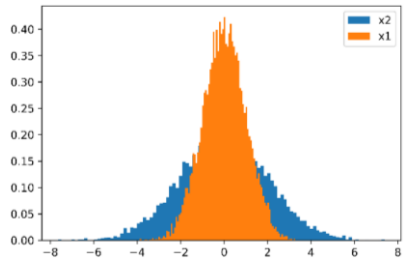




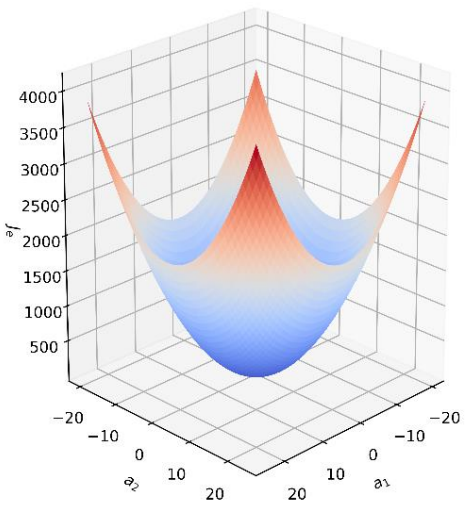
$x_1 = 2 * \text{randn(N, 1)}$
$x_2 = \text{randn(N, 1)}$

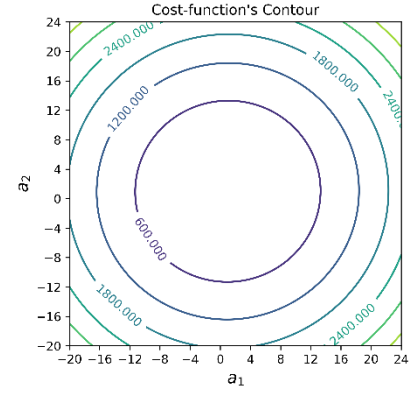


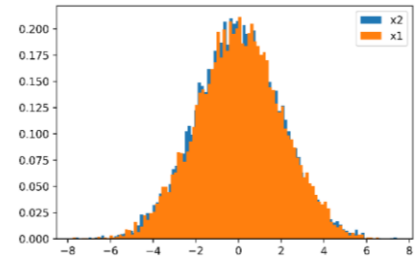




$x_1 = \text{randn(N, 1)}$
$x_2 = 2 * \text{randn(N, 1)}$







$x_1 = 2 * \text{randn(N, 1)}$
$x_2 = 2 * \text{randn(N, 1)}$

Exemplo: formatos_diferentes_da_superfície_de_erro.ipynb

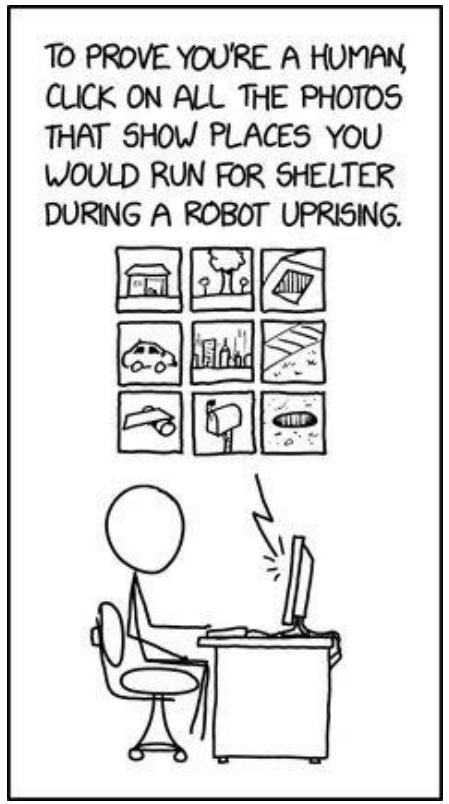
Desvantagens da forma fechada (Eq. Normal)

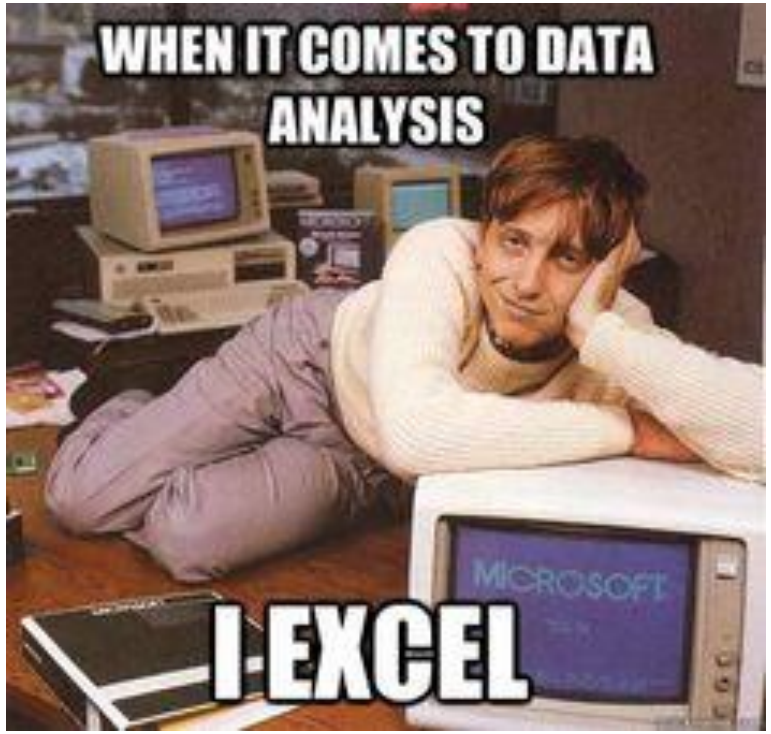
- **Alta complexidade computacional**: a solução da ***equação normal*** envolve o cálculo da inversa de $\boldsymbol{\Phi}^T\boldsymbol{\Phi}$, o qual tem complexidade computacional que varia de $O(K^{2.4})$ a $O(K^3)$.
  - **Exemplo**: Se o número de ***atributos***, K, dobrar, o tempo para cálculo aumenta de $2^{2.4} = 5.3$ a $2^3 = 8$ vezes.
- Dependendo do número de ***exemplos***, N, e de ***atributos***, $\boldsymbol{x}$, a matriz $\boldsymbol{\Phi}$ pode consumir muita memória.
- **Portanto, essa abordagem não é escalonável!**
- Adicionalmente, para irmos além dos modelos lineares (i.e., modelos não-lineares) precisamos lidar com o fato de que não existem formas fechadas como a **equação normal**.
- **Solução**: abordagens iterativas
  - Métodos iterativos de busca que façam a atualização dos parâmetros, $\boldsymbol{\alpha}$, à medida que os dados são apresentados ao modelo.
  - Por exemplo, o algoritmo do ***gradiente descendente***, o qual veremos a seguir.

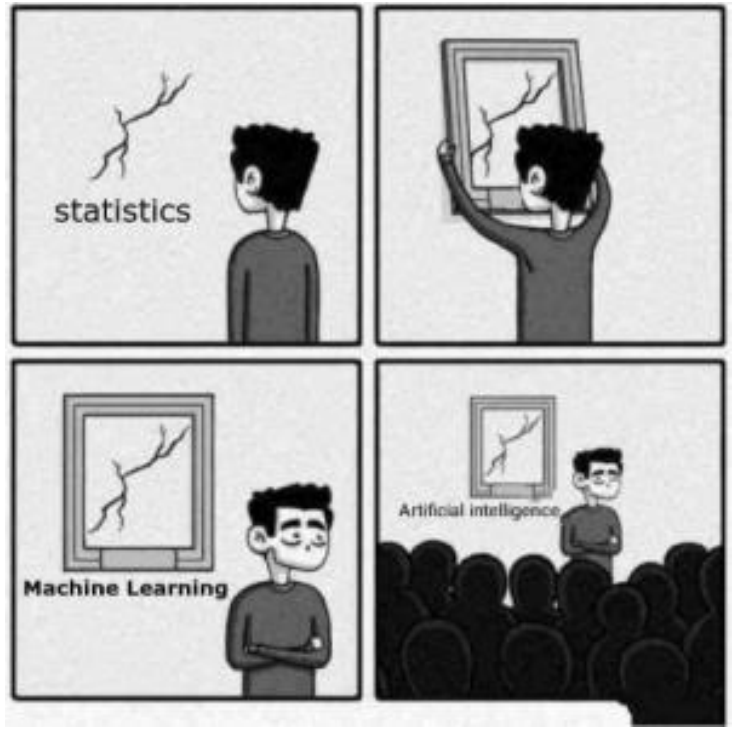
Tarefas

- **Quiz**: “*T319 - Quiz - Regressão: Parte I*” que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #2**.
  - Pode ser baixado do MS Teams ou do GitHub.
  - Pode ser respondido através do link acima (na nuvem) ou localmente.
  - Instruções para resolução e entrega dos laboratórios.
  - **Laboratórios podem ser feitos em grupo, mas as entregas devem ser individuais.**

Obrigado!







Albert Einstein: Insanity Is Doing the Same Thing Over and Over Again and Expecting Different Results

Machine learning: