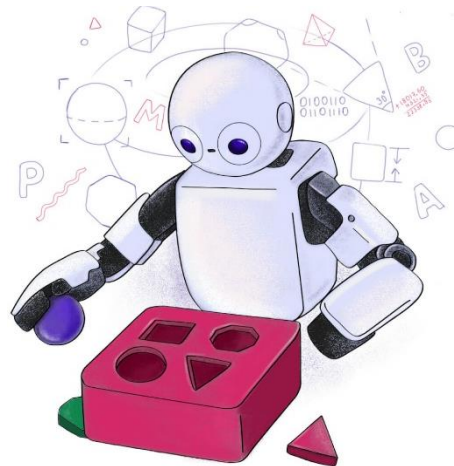


T319 - Introdução ao Aprendizado de Máquina: *Regressão Linear (Parte I)*



Inatel

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

Motivação

- **Exemplo 1:** Estimar o preço de casas.

5 quartos, 3 vagas na garagem, 1 piscina, cond. de luxo, 500 m^2



R\$ 1.000.000,00

1 quarto, sem vaga na garagem, bairro afastado, 70 m^2



R\$ 200.000,00

2 quartos, 1 vaga na garagem, centro, 200 m^2

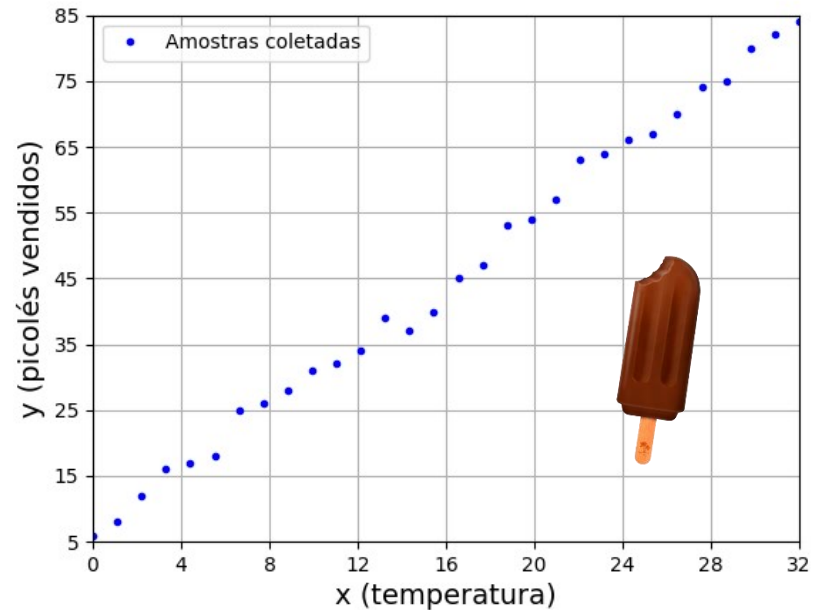


???

- Podemos usar **regressão** para encontrar uma **relação matemática**, $h(x)$, entre o n° de quartos, n° de vagas na garagem, se tem piscina, localização, área, etc. de uma casa e seu valor.
- **Objetivo:** agilizar o processo de avaliação de propriedades.

Motivação

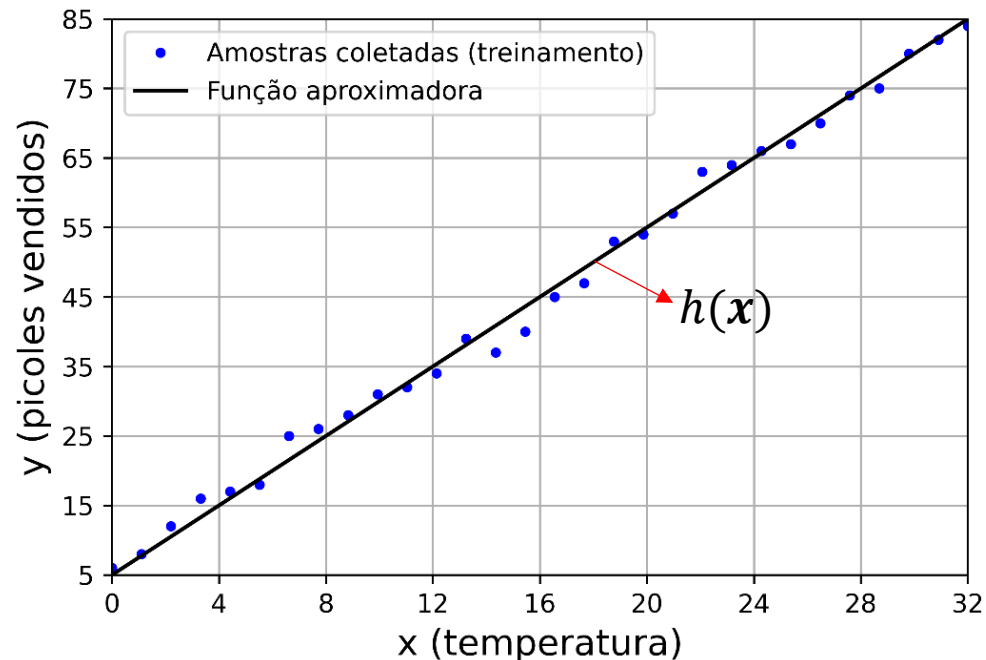
- **Exemplo 2:** Estimar as vendas de picolés.



- Podemos usar regressão para encontrar um **mapeamento**, $h(x)$, entre a temperatura média de um dia e a quantidade de picolés vendidos.
- **Objetivo:** reduzir o desperdício e aumentar os lucros.

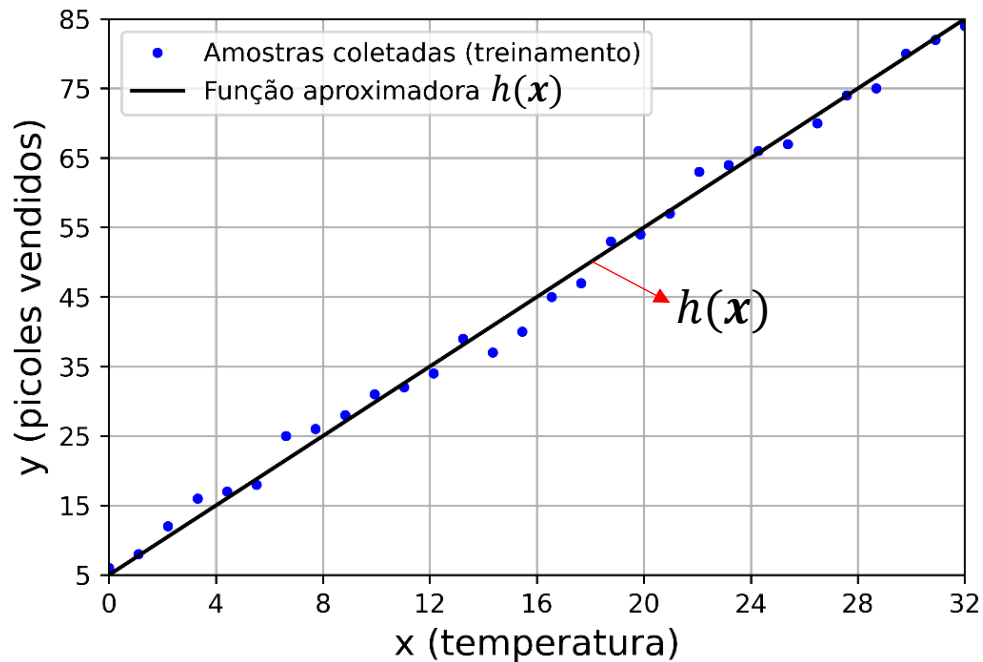
*Como podemos resolver esses problemas
de estimação?*

Regressão linear



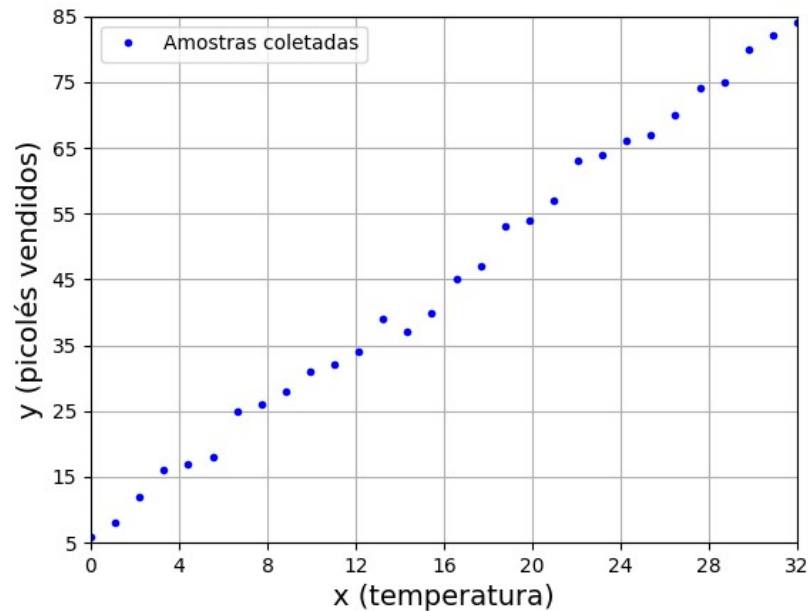
- É um dos mais simples, antigos e usados algoritmos de aprendizado de máquina.
- Regressão linear também é conhecida como ***aproximação de funções*** ou ***ajuste de curvas***.
 - Por exemplo, ajustar uma curva (i.e., uma função) aos dados de vendas de picolés.
- Vai nos dar ***intuições*** importantes para o ***entendimento*** de ***algoritmos mais complexos***, como ***classificadores*** e ***redes neurais***.

Qual o objetivo da regressão linear?



- **Objetivo:** encontrar uma função, $h(x)$, que **mapeie** as entradas, x , em uma variável de saída $\hat{y} = h(x)$, de tal forma que $h(x)$ se ajuste aos dados coletados de **forma ótima**.
 - Ótimo no sentido da **minimização da diferença** entre as previsões feitas por $h(x)$, (i.e., os valores de \hat{y}), e os valores esperados de saída, y .
- No contexto da regressão linear, nós chamamos as entradas, x , de **atributos** e os valores esperados de saída, y , de **rótulos**.
 - Vários pares de x e y formam o **conjunto de treinamento**.

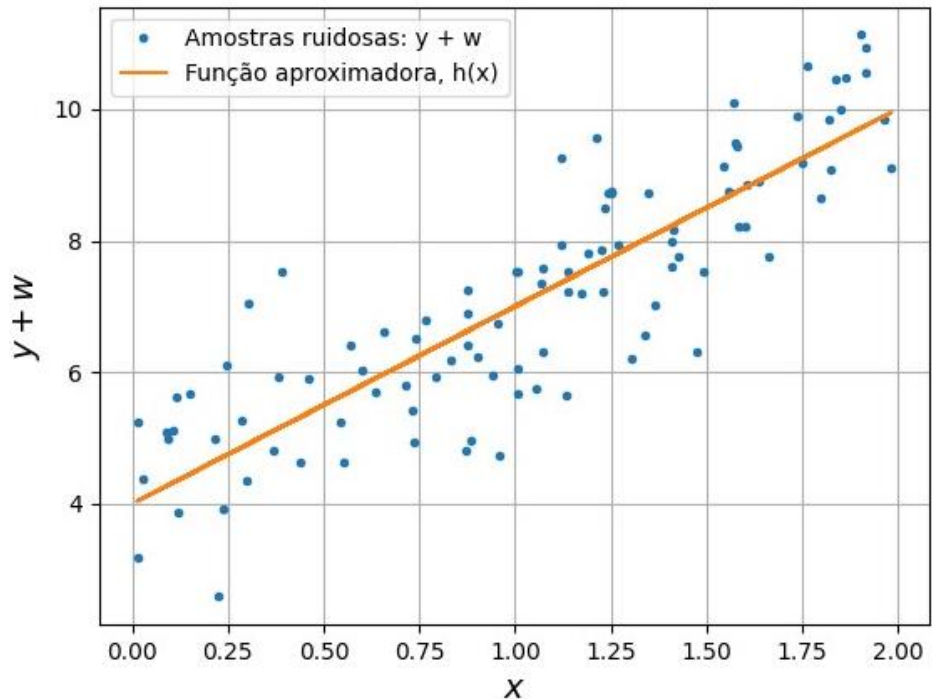
Relação matemática real entre x e y



Não existe realmente uma função por trás desse comportamento da venda de picolés, mas podemos encontrar uma função que o aproxime usando regressão.

- Na grande maioria dos casos, **não existe uma relação matemática**, i.e., uma função, que **realmente descreve a relação entre as entradas, x , e as saídas esperadas, y** , dos dados em um conjunto.
- Porém, mesmo nestes casos, ainda podemos **modelar** a relação entre x e y da **melhor forma possível** usando regressão.
- Nos casos onde essa função existe, a chamamos de **função verdadeira** ou **objetivo**, a qual é, normalmente, denotada por $f(x)$.

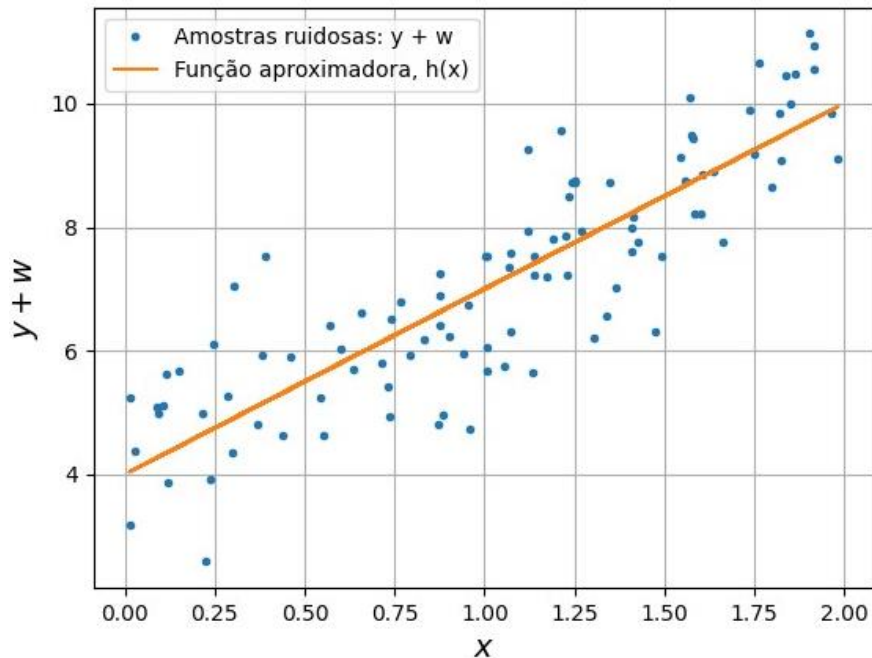
Dados ruidosos



OBS.: w representa o ruído contaminando os dados.

- Em geral, em problemas de regressão, lidamos com ***dados que estão sujeitos a ruído***.
- Exemplos de introdução de ruído aos dados:
 - **Previsão de preços de imóveis:** os preços de imóveis podem ser influenciados por mudanças na taxa de juros, inflação, preferências do comprador, valor sentimental, etc.
 - **Previsão de preços de ações:** os preços de ações podem ser influenciados por notícias, mudanças econômicas, eventos globais, etc.
 - **Deteção de símbolos:** os símbolos transmitidos passam através de um canal ruidoso.
- Nos nossos exemplos, modelaremos o ruído como uma variável aleatória Gaussiana.

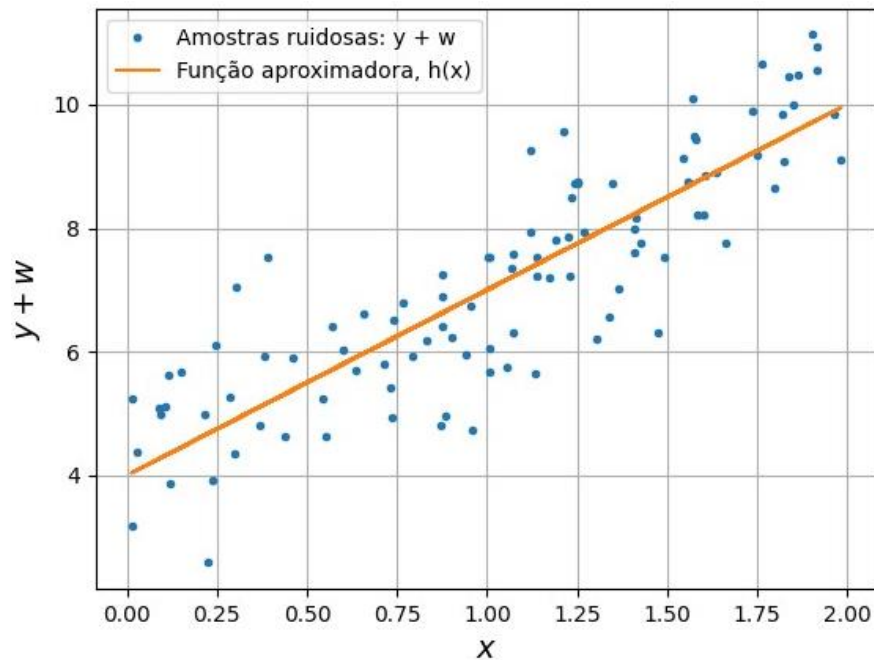
O objetivo geral da regressão linear



OBS.: w representa o ruído contaminando os dados.

- Assim, o **objetivo geral** da regressão linear é encontrar uma função $h(x)$ que descreva a relação entre x e y , **mesmo que os dados coletados estejam corrompidos por ruído**, i.e., $y + w$.
- Em outras palavras, a regressão busca **capturar** um **padrão geral por trás dos dados, mesmo que ruidosos, de forma a se obter uma boa capacidade de generalização**.
- Esse conjunto de **valores coletados**, ou **experiências prévias**, será chamado de **conjunto de treinamento**.

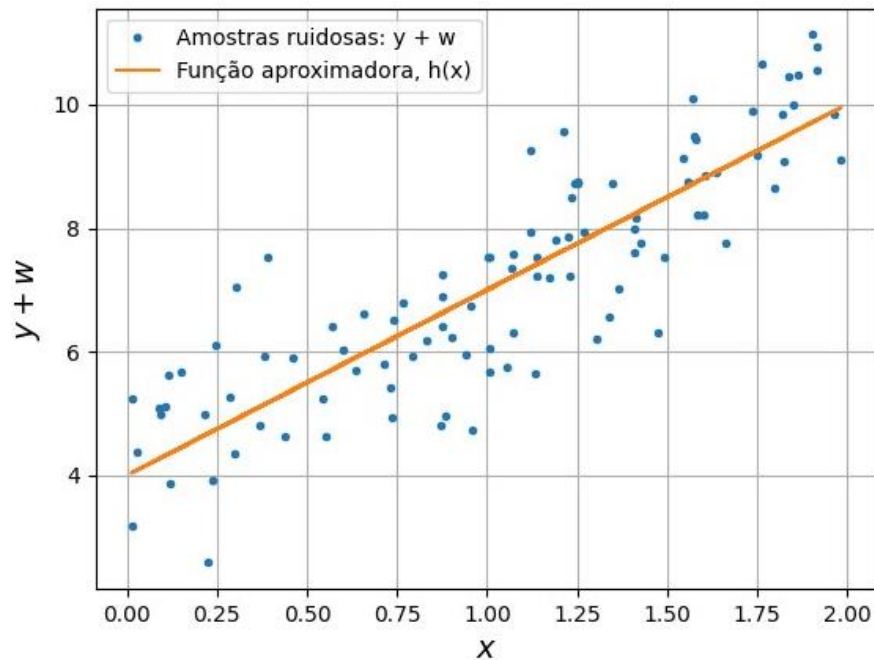
Qual é o paradigma de aprendizado?



OBS.: w representa o ruído contaminando os dados.

- Dado que possuímos um **conjunto** de valores de entrada, x (os **atributos**), e os seus respectivos valores esperados de saída, y (os **rótulos**), **que tipo de aprendizado é realizado pela regressão linear?**

Qual é o paradigma de aprendizado?



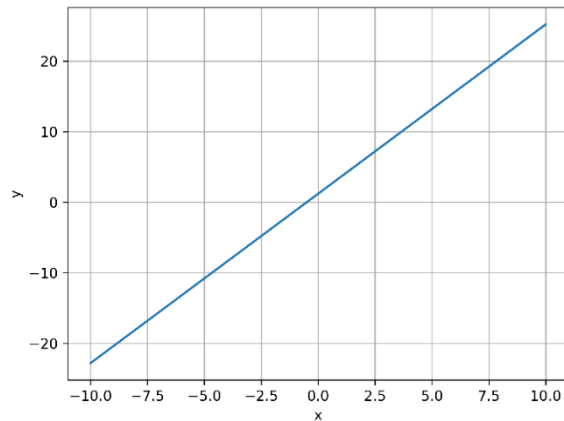
OBS.: w representa o ruído contaminando os dados.

- Dado que possuímos um **conjunto** de valores de entrada, x (os **atributos**), e os seus respectivos valores esperados de saída, y (os **rótulos**), **que tipo de aprendizado é realizado pela regressão linear?**

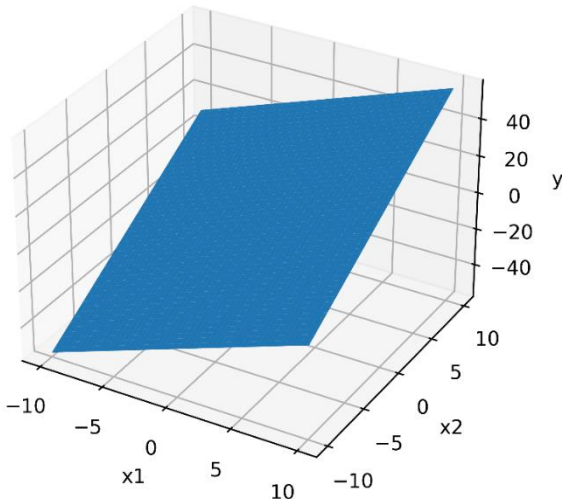
**Aprendizado
supervisionado**

Por que linear?

$$h(x_1) = a_0 + a_1x_1 \text{ (reta)}$$



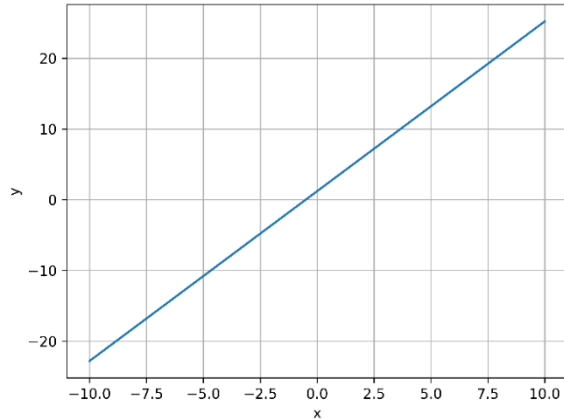
$$h(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2 \text{ (plano)}$$



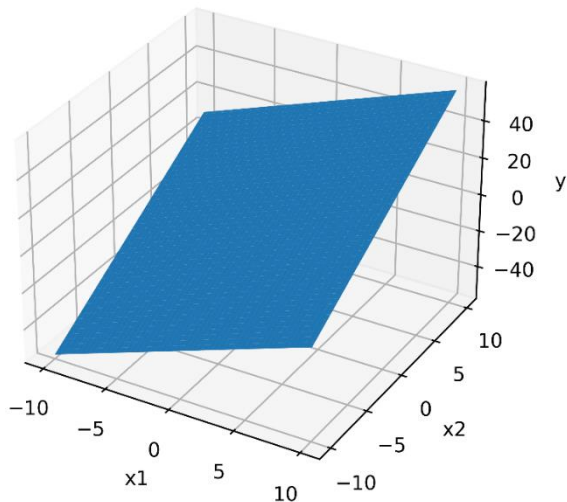
- A **regressão** é chamada de **linear** porque a saída da **função aproximadora**, $\hat{y} = h(\mathbf{x})$, é **modelada** como sendo uma **combinação linear dos atributos**, \mathbf{x} .
- Ou seja, usamos uma **função que é uma combinação linear dos atributos** para modelar o relacionamento entre \mathbf{x} e y .
- Por exemplo, com **um atributo**, x_1 , temos a equação de uma **reta**:
$$\hat{y} = h(\mathbf{x}) = a_0 + a_1x_1.$$
- Já com **dois atributos**, x_1 e x_2 , temos a equação de um **plano**:
$$\hat{y} = h(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2.$$

Por que linear?

$$h(x_1) = a_0 + a_1 x_1 \text{ (reta)}$$



$$h(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_2 \text{ (plano)}$$



- Uma **generalização** para qualquer **número de atributos**, K , é dada pela equação do **hiperplano**:
$$\hat{y} = h(\mathbf{x}) = a_0 + a_1 x_1 + \cdots + a_K x_K = a_0 + \sum_{i=1}^K a_i x_i.$$

Número de atributos
- O **hiperplano** será o **primeiro formato** de **função aproximadora** (também, chamado de **modelo**) que iremos utilizar.
- O **modelo** é definido pela **equação** e pelos **pesos**, $a_i, \forall i$.
- Existem outros formatos de funções (i.e., modelos), como os **polinômios**, os quais veremos mais adiante, que modelarão relações não lineares entre $x_k, \forall k$ e \hat{y} .

Por que linear?

Observação

- A palavra **linear** no contexto da regressão significa “**linear com relação aos pesos**” e não com relação aos **atributos**, x .
- Desta forma, os seguintes **modelos** também são lineares com relação aos pesos:

$$\left. \begin{array}{l} \checkmark \hat{y} = h(\mathbf{x}) = a_0 + a_1 \log x_1 + a_2 \cos x_2 \\ \checkmark \hat{y} = h(\mathbf{x}) = a_0 + a_1 e^{x_1} \\ \checkmark \hat{y} = h(\mathbf{x}) = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 \end{array} \right\}$$

Mesmo não tendo um mapeamento linear das entradas na saída, continuamos tendo uma combinação linear dos atributos em relação aos pesos.

- Exemplo de um modelo não-linear: $\hat{y} = h(\mathbf{x}) = \frac{a_0 x_1}{a_1 + x_1}$.

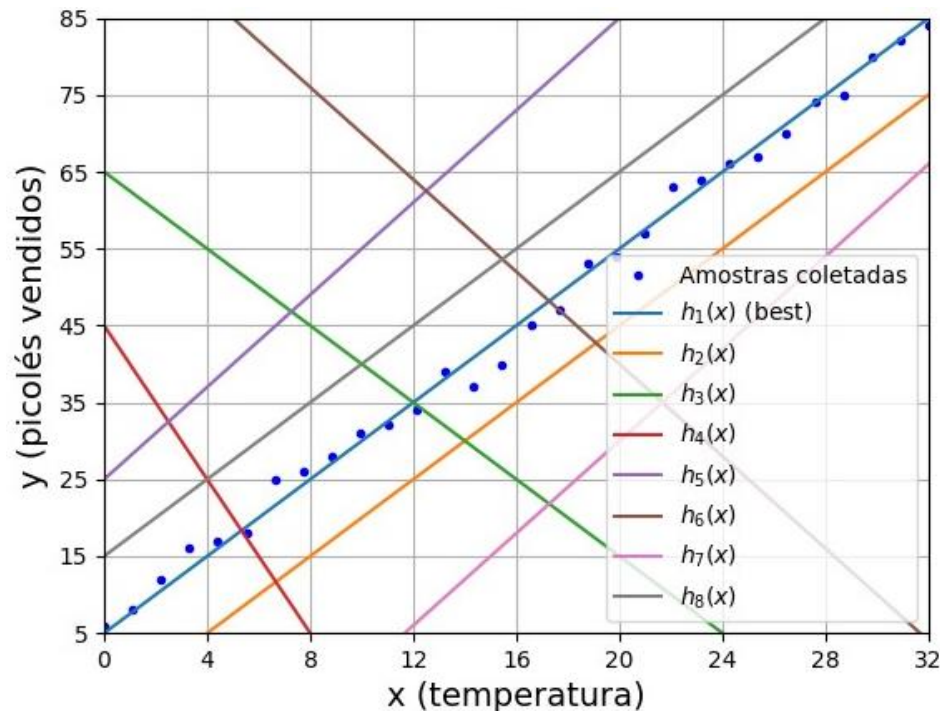
Não é possível expressar a equação como uma combinação linear dos atributos.

- Em geral, não é possível aplicar algumas das técnicas de regressão que veremos em breve a modelos não-lineares.

Função hipótese

- No contexto da regressão, a **função aproximadora**, $h(x)$, é chamada de **função hipótese**, pois refere-se à **explicação proposta para a relação entre as entradas, x e a saída esperada, y** .
 - Por exemplo, anteriormente, **hipotetizamos** que uma **reta** explicaria bem a relação entre a temperatura média de um dia e o número de picolés vendidos.
- O conjunto de todas as possíveis **funções hipótese** é chamado de **espaço de hipóteses, H** .
- Em outras palavras, é o **espaço** criado por **todas as combinações possíveis de atributos e valores dos pesos**.
- A solução que **melhor mapeia** x em y encontra-se dentro do **espaço de hipóteses** e **nosso objetivo é encontra-la**.

Espaço de hipóteses



A figura mostra algumas possíveis funções hipótese que modelam o relacionamento entre x e y .

- No exemplo da previsão da quantidade de picolés vendidos, onde usamos a **equação de uma reta** como **função hipótese**,

$$\hat{y} = h(x) = a_0 + a_1 x_1,$$

- o **(sub)espaço de hipóteses contém todas as possíveis retas** (restringimos o espaço) que podem ser usadas para mapear os dados.
- Porém, claro, **apenas uma é a melhor segundo o critério de minimização da diferença entre a saída do modelo e o valor esperado**.

Refinando o objetivo da regressão linear

- Agora que temos um formato definido para $h(x)$, ou seja, o **modelo**, podemos refinar o objetivo da regressão um pouco mais.
- **Objetivo:** **Encontrar** os **pesos**, a_0, a_1, \dots, a_K , que façam com que $h(x)$ se **ajuste de forma ótima** ao conjunto de dados coletado.
 - **Ótima** no sentido em que $h(x)$ minimiza o erro (i.e., diferença) entre os valores de saída do modelo, \hat{y} , e os esperados, y .
- Na sequência, definiremos uma **função de erro** que nos ajudará a encontrar os **pesos ótimos** do modelo.
 - **Pesos ótimos:** conjunto de pesos da função hipótese, $h(x)$, que fazem com que o erro seja minimizado.

Definição formal do problema

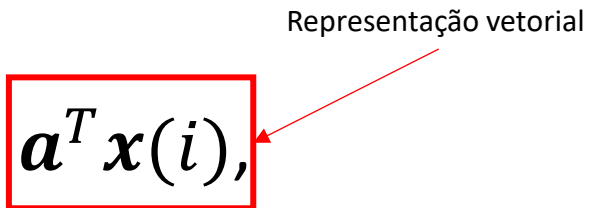
Informação disponível

- Conjunto de N observações ou ***pares de treinamento***: $\{x(i), y(i)\}$, $i = 0, \dots, N - 1$, onde
 - $x(i) = [1, x_1(i), \dots, x_K(i)]^T \in \mathbb{R}^{K+1 \times 1}$ é o ***vetor de atributos*** (vetor coluna) com dimensão $(K + 1 \times 1)$ contendo os i -ésimos valores dos ***atributos***.
 - $y(i) \in \mathbb{R}$ é o i -ésimo ***rótulo*** (valor esperado de saída) referente ao vetor de atributos, $x(i)$.
- As N observações formam o ***conjunto de treinamento***, ou seja, o conjunto de amostras coletadas que utilizaremos para ***treinar o modelo***.
- Notem que o ***vetor de atributos*** tem sempre um elemento a mais do que o número real de atributos, K .
- Como veremos, esse elemento, o primeiro do vetor, tem valor constante igual a 1 para que possamos representar o ***modelo*** em ***formato vetorial***.

Definição formal do problema

Modelo

- Função do hiperplano

$$\hat{y}(i) = h(\mathbf{x}(i)) = a_0 + \sum_{i=1}^K a_i x_i = \mathbf{a}^T \mathbf{x}(i),$$


onde $\mathbf{a} = [a_0, \dots, a_K]^T$ é um vetor **coluna** com dimensão $(K + 1 \times 1)$ contendo os **pesos** da **função hipótese**.

Observações

- a_0 é o **coeficiente linear**, ou seja, é o valor de $h(\mathbf{x})$ que intercepta o eixo das ordenadas, y , a_0 é conhecido também como **intercept** ou **bias**.
- Como a_0 não tem um **atributo** relacionado a ele, para **facilitar a representação matricial**, criamos um **atributo falso**, x_0 , chamado de **atributos de bias**, com valor constante igual a $x_0 = 1$.

Definição formal do problema

Objetivo

- Encontrar o **vetor de pesos**, \mathbf{a} , que minimize o **erro**, dado por uma **função de erro**, $J_e(\mathbf{a})$, entre a aproximação, $\hat{y}(i)$, e o valor esperado, $y(i)$, para **todos os exemplos do conjunto de treinamento**.
- Ou seja, o **treinamento** do modelo **envolve** a **minimização de uma função de erro**.
- Isso pode ser escrito como um problema de otimização

$$\min_{\mathbf{a} \in \mathbb{R}^{K+1 \times 1}} J_e(\mathbf{a}).$$

- Portanto, para que consigamos treinar o modelo, precisamos **definir** uma **função de erro**.

Função de erro

- Existem várias possibilidades para se definir a **função de erro**.
- Porém, em problemas de regressão, é comum utilizar-se o **erro quadrático médio (EQM)**

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

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

onde N é o número de exemplos ou observações.

- O **EQM** nada mais é do que a **média aritmética do quadrado dos erros**.
- Veremos mais adiante a razão pela qual o **EQM** é utilizado.
- O vetor de pesos foi colocado em evidência na função hipótese para deixar claro que o **erro varia com o valor dos pesos**.
 - Variando os pesos, ficamos mais próximos ou distantes da função hipótese que minimiza o erro, i.e., da **função hipótese ótima**. Estamos caminhando no espaço de hipóteses.

Função de erro

- A **função de erro** pode ser reescrita em **formato matricial** como

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

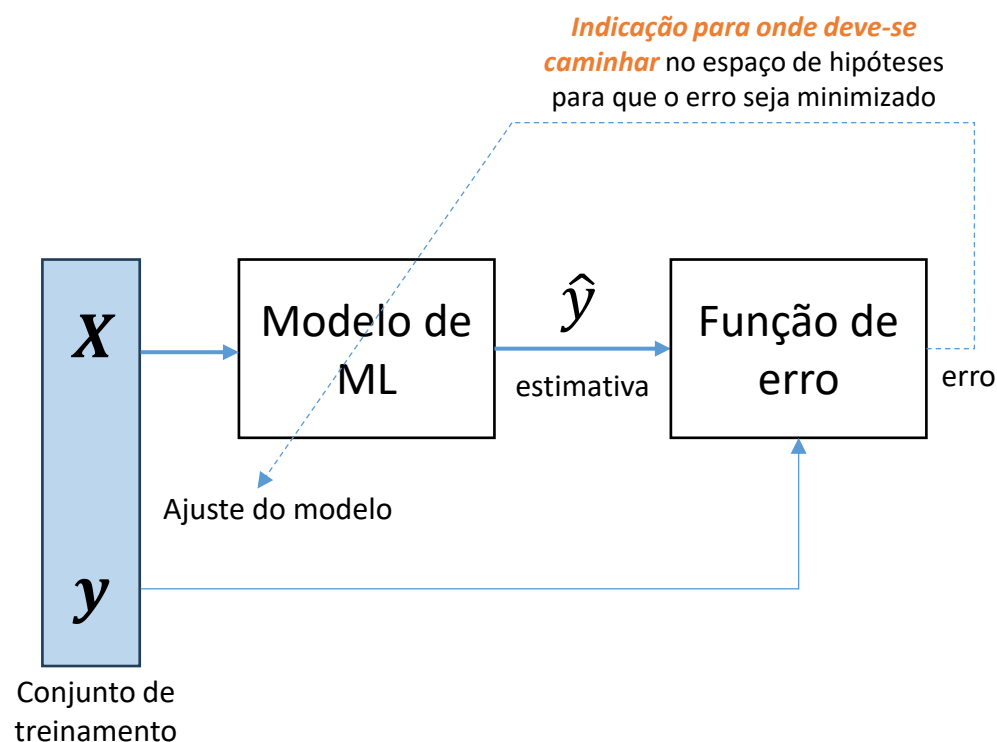
onde $\mathbf{y} = [y(0), \dots, y(N-1)]^T$ é um **vetor coluna** com dimensão $(N \times 1)$ contendo todos os N valores esperados e $\mathbf{X} = [\mathbf{x}(0), \dots, \mathbf{x}(N-1)]^T$ é uma **matriz** com dimensão $(N \times K + 1)$ contendo todos os vetores de atributo.

- Então, para encontrarmos o **vetor de pesos**, \mathbf{a} , devemos **minimizar a função de erro**

$$\min_{\mathbf{a} \in \mathbb{R}^{K+1 \times 1}} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2.$$

OBS.: Por ser constante, o termo $1/N$ não influencia na minimização e, portanto, pode ser omitido na equação acima.

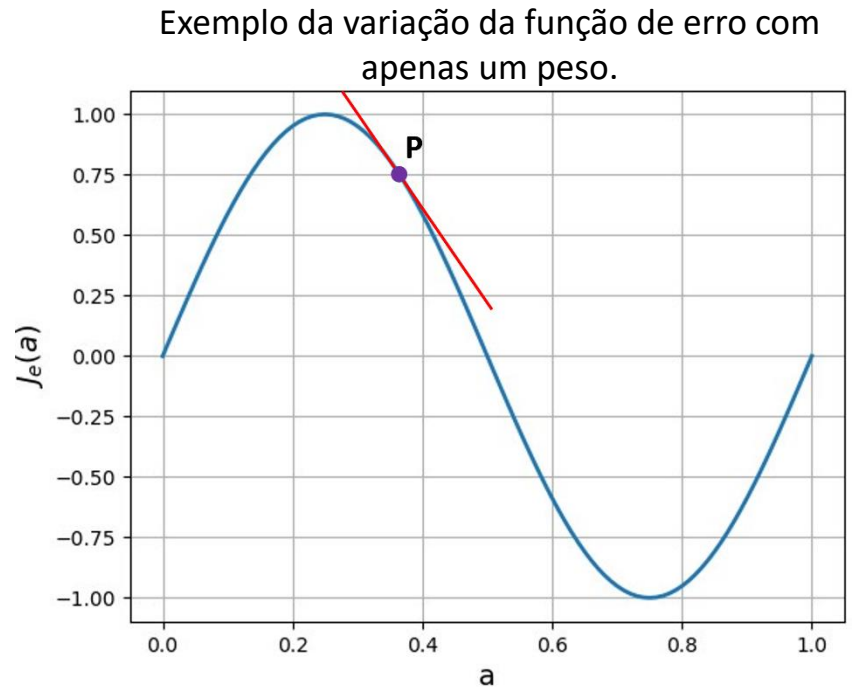
Resumo do problema da regressão



- Portanto, resumindo, nosso objetivo é **treinar** um **modelo**, a partir de um **conjunto de treinamento**, que **minimize o erro de aproximação**.
- Assim, para **treinar** o modelo, precisamos encontrar **maneiras** de **ajustar seus pesos** de forma que o erro seja minimizado.
- Na sequência veremos duas formas de se encontrar o **conjunto de pesos ótimo** através da **minimização da função de erro**.

***Como encontramos os pesos que
minimizam a função de erro?***

Como minimizar a função de erro?

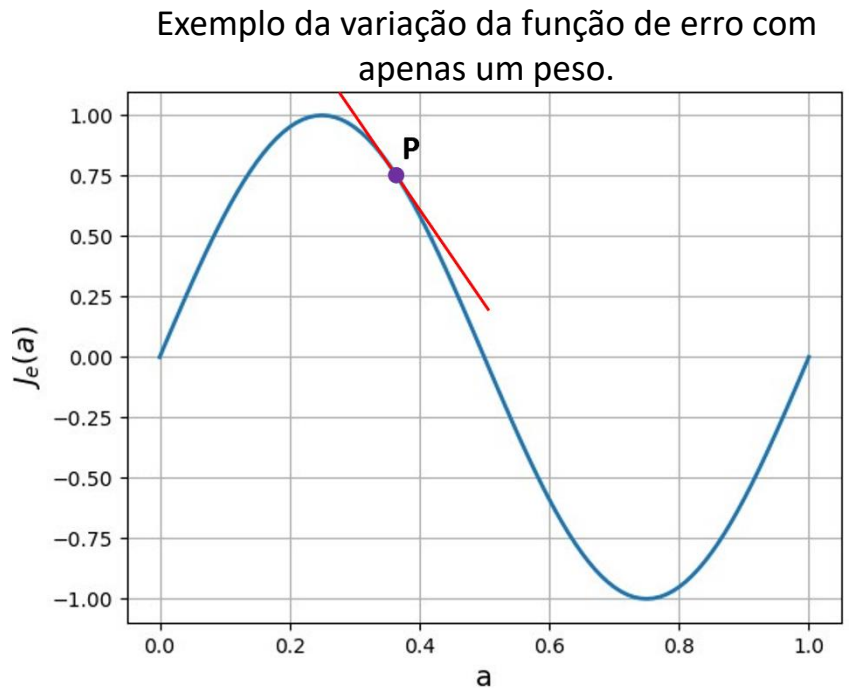


A reta vermelha é **tangente** à curva no ponto **P** (roxo). Sua **inclinação** é dada pela **derivada parcial** da função naquele ponto.

- Da disciplina de cálculo, sabemos que derivando a **função de erro** em relação ao vetor \mathbf{a} e igualando a derivada à 0, nós encontrar o **ponto** (i.e., os valores dos pesos) onde a **taxa de variação da função é nula** (i.e., não cresce ou decresce mais)

$$\frac{\partial J_e(\mathbf{a})}{\partial \mathbf{a}} = \frac{\partial \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2}{\partial \mathbf{a}} = 0.$$

Como minimizar a função de erro?



A reta vermelha é **tangente** à curva no ponto **P** (roxo). Sua **inclinação** é dada pela **derivada parcial** da função naquele ponto.

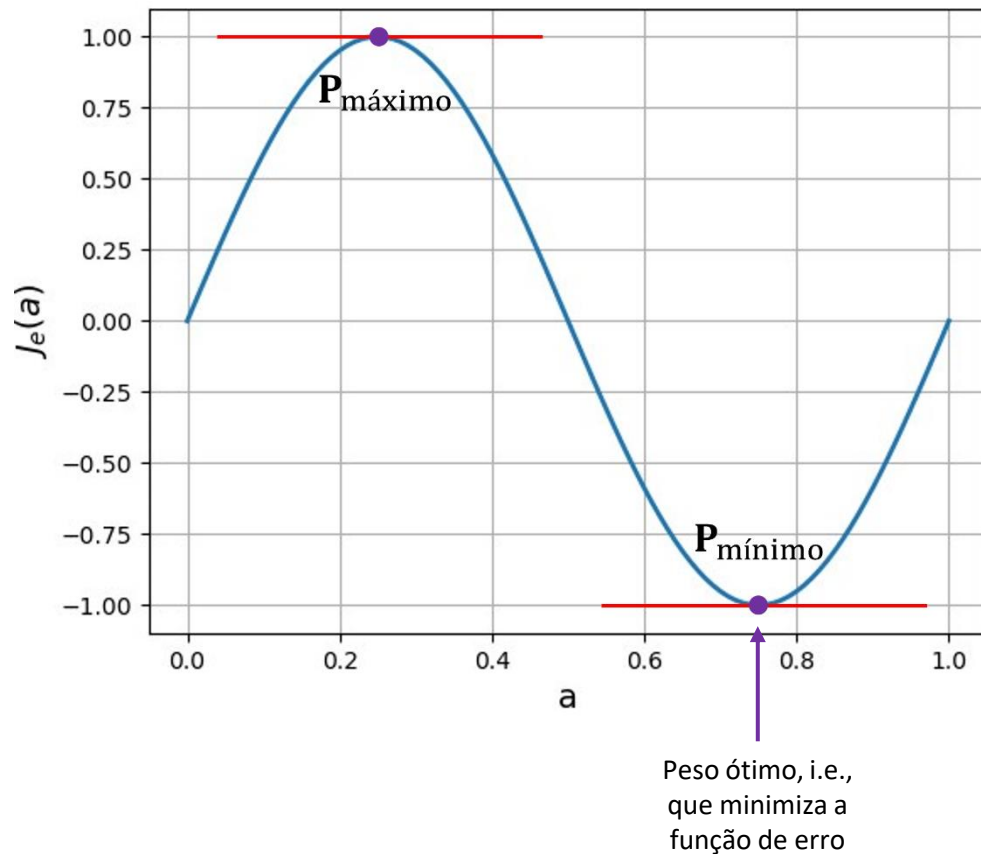
- A resolução da derivada também é interpretada como sendo o **ponto** onde a **inclinação de um hiperplano** (e.g., **reta na figura ao lado**) **tangente à função de erro é nula**.

$$\frac{\partial J_e(\mathbf{a})}{\partial \mathbf{a}} = \frac{\partial \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2}{\partial \mathbf{a}} = 0.$$

- Após aplicarmos a derivada parcial, chegamos à seguinte equação

$$2\mathbf{a}^T \mathbf{X}^T \mathbf{X} - 2\mathbf{y}^T \mathbf{X} = 0.$$

Como minimizar a função de erro?



- Porém, o **ponto** onde a **taxa de variação da função de erro é nula** pode ocorrer tanto em um **mínimo** quanto em um **máximo** da **função**, pois a inclinação da reta tangente é nula em ambos os casos.
- Entretanto, no nosso caso, estamos interessados no **ponto** onde a **função de erro tem o seu menor valor**.
 - Ponto de mínimo.
- É nesse ponto que encontramos os pesos que nos dão a **função hipótese ótima**.

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

Como minimizar a função de erro?

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

$$\frac{\partial^2 \|y - Xa\|^2}{\partial^2 a} = 2X^T X.$$

- Se a **matriz de atributos**, X , tiver **posto** igual a $K + 1$, então a matriz $X^T X$ 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 $X^T X$ é **positiva semi-definida** se $z^T X^T X z = \|Xz\|_2^2 \geq 0, \forall z \neq 0$.

Equação normal

- Agora que sabemos que se a matriz de atributos, X , tiver $K + 1$ **colunas linearmente independentes** encontramos o **ponto de mínimo da função** através da sua **derivada parcial**, nós podemos resolver a equação da derivada parcial e encontrar o vetor de pesos ótimo.

- Assim, resolvendo a equação em função de \mathbf{a} , temos

Conjunto de pesos
correspondente ao ponto
mais baixo da função de erro

$$2\mathbf{a}^T \mathbf{X}^T \mathbf{X} - 2\mathbf{y}^T \mathbf{X} = 0,$$

$$\mathbf{a}_{\text{opt}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

- Essa equação é conhecida como **equação normal** e nos dá a **solução ótima** em relação a minimização do **erro quadrático médio** para esse **sistema de equações lineares**.
- Notem que só precisamos do conjunto de amostras coletadas (i.e., de treinamento), X e y , para encontrar o **vetor de pesos ótimo**, \mathbf{a}_{opt} .

Observações quanto a equação normal

1. A **equação normal** encontra uma **solução única** se e somente se a matriz, $X^T X$, for **invertível** (i.e., **não-singular**).
 - Para que a matriz seja invertível, ela deve ter **posto** igual a $K + 1$.
2. A **equação normal** só funciona para sistemas **determinados ou sobredeterminados**, ou seja, quando o número de equações (i.e., pares 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 $X^T X$ tem **posto** menor do que $K + 1$ e, portanto, é **singular** (i.e., a matriz $X^T X$ não é invertível).
 - Neste caso, não existe solução ou ela não é única.

Superfície de erro

$$J_e(\mathbf{a}) = \frac{1}{N} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2$$

- Notem que o EQM é **função do vetor de pesos**, \mathbf{a} , pois o **conjunto de exemplos é estático**.
- Ou seja, o erro varia conforme os valores dos pesos mudam.
- $J_e(\mathbf{a})$ faz o **mapeamento** entre o **vetor de pesos** e o **erro correspondente**:
 - $J_e(\mathbf{a}): \mathbb{R}^{K+1 \times 1} \rightarrow \mathbb{R}$
- Se nós calcularmos o EQM para diversos valores de \mathbf{a} e **plotarmos esses erros em função dos pesos**, nos obtemos uma superfície chamada de **superfície de erro**.

***Que formato vocês acham que essa
superfície tem?***

Superfície de erro

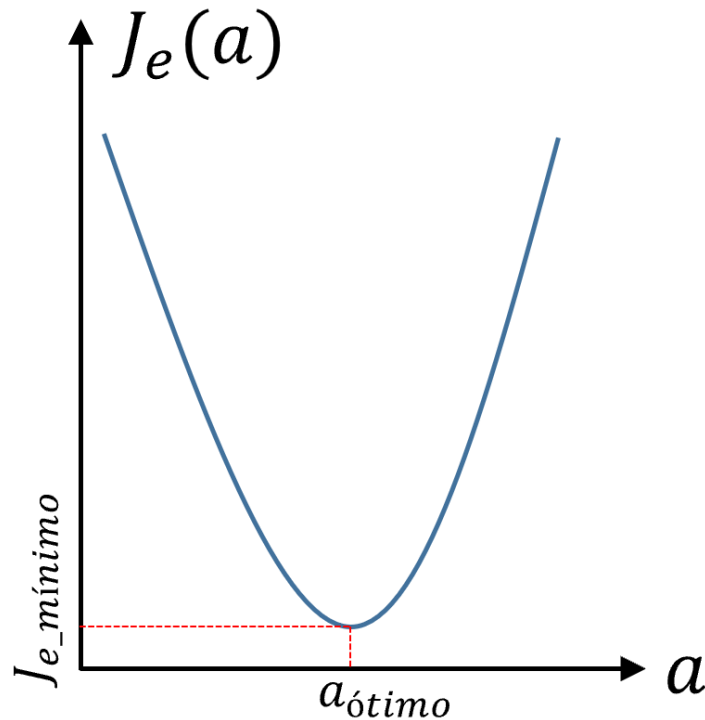
- É fácil descobrir se analisarmos a equação do EQM.
- Se expandirmos $J_e(\mathbf{a})$ notamos que ela possui forma **quadrática** com relação ao **vetor de pesos, \mathbf{a}** .

$$J_e(\mathbf{a}) = \frac{1}{N} \|\mathbf{y} - \mathbf{X}\mathbf{a}\|^2 = \frac{1}{N} (\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\mathbf{a}^T - \mathbf{a}^T \mathbf{X}^T \mathbf{y} + \boxed{\mathbf{a}^T \mathbf{X}^T \mathbf{X} \mathbf{a}}).$$

Termo
quadrático,
 $\|\mathbf{X}\mathbf{a}\|^2$.

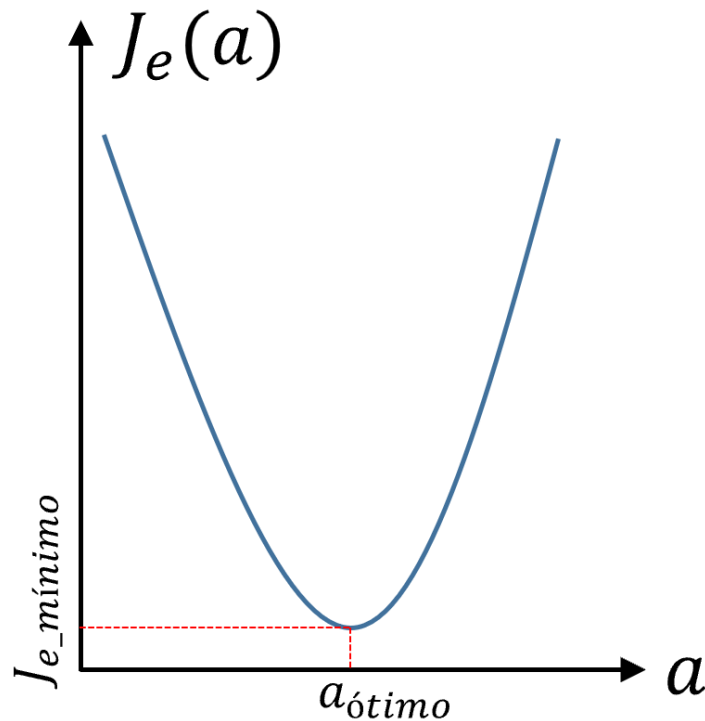
- O gráfico de uma função quadrática é uma **parábola**.
- Essa parábola tem a **concavidade voltada para cima**, pois $\|\mathbf{X}\mathbf{a}\|^2 \geq 0$.
- O que isso significa?

Superfície de erro



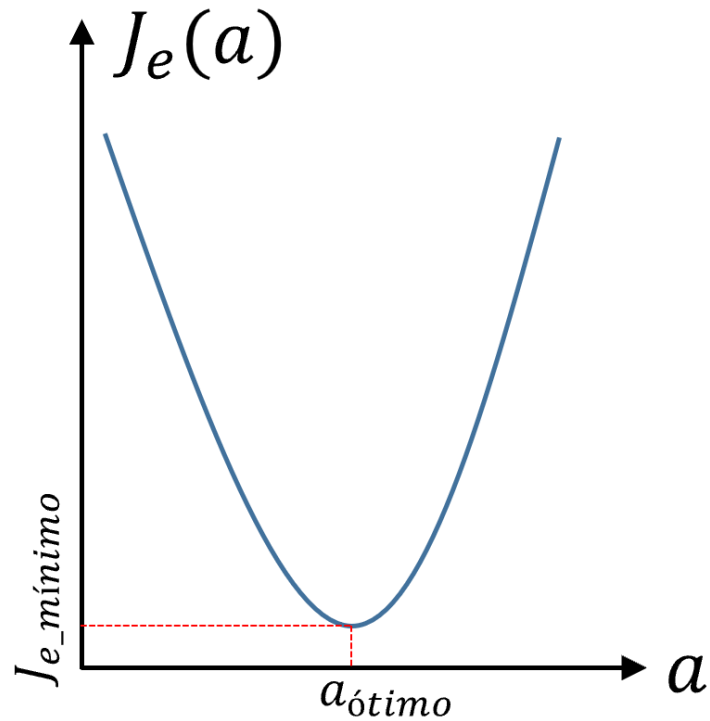
- Isso significa que a **superfície de erro** tem o formato de uma **parábola convexa** (i.e., tem formato de **tigela** ou **vale**).
- A figura ao lado mostra o formato da **superfície de erro para uma função hipótese com apenas um peso**.
- Para funções hipótese com mais pesos, teremos superfícies em 3 ou mais dimensões.

Superfície de erro



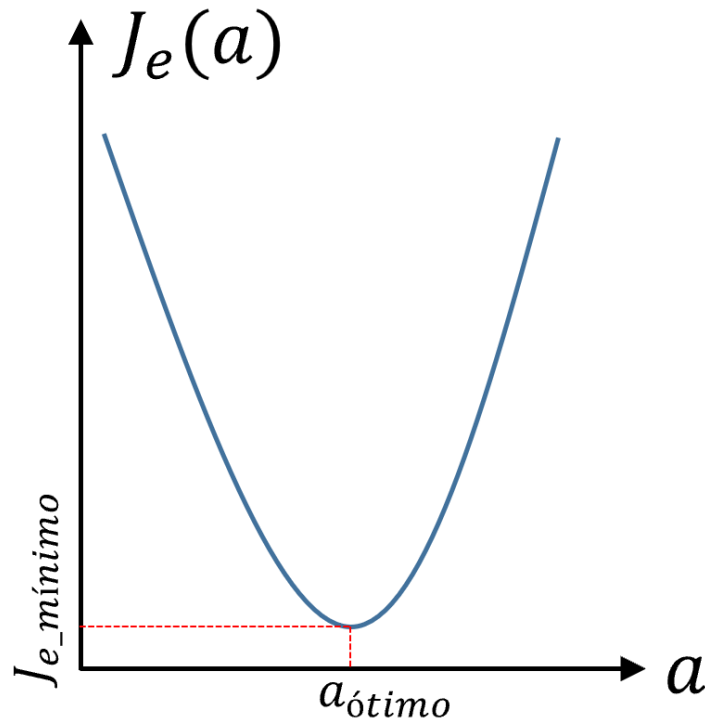
- Vejam que a superfície possui um **único valor mínimo** (i.e., o **vértice** da parábola).
- Esse ponto onde a superfície (i.e., a função de erro) tem o seu menor valor é chamado de **ponto de mínimo**.
- Ele também é chamado de **mínimo global**, pois não há outro valor do peso a que resulte em um erro menor.
- Todos os outros valores têm erro maior.

Superfície de erro



- Este é o motivo de usarmos o **erro quadrático médio** como **função de erro**, pois **só existe um único vetor de pesos, a , que minimiza a função de erro.**
- O ponto de mínimo nos dá o **valor ótimo** dos pesos da função hipótese.
- Ele é encontrado
 - através da **equação normal**.
 - **visualmente**, em alguns casos, se plotarmos a superfície, pois basta **localizar seu ponto mais baixo**.
 - ou através de um **algoritmo de otimização iterativo** que caminha sempre em direção à parte mais baixa da superfície.

Superfície de erro



- O entendimento do **formato** e da questão da superfície ter **apenas um ponto de mínimo** nos ajudará a entender como o **algoritmo de otimização iterativa** que vamos aprender em breve funciona.
- Na sequência, plotamos a superfície de erro para uma função hipótese.

Plotando a superfície de erro

- Vamos supor 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** (ou **modelo gerador**) dada por

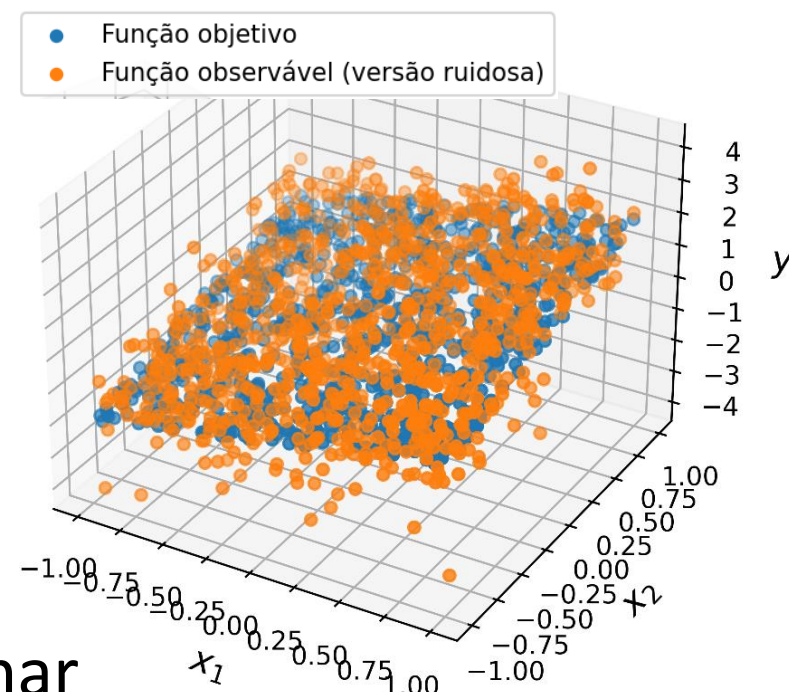
$$y(n) = a_1 x_1(n) + a_2 x_2(n),$$

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

- Agora, suponhamos que nós quiséssemos aproximar a **função objetivo** a partir, apenas, de suas **amostras ruidosas** com a seguinte **função hipótese**

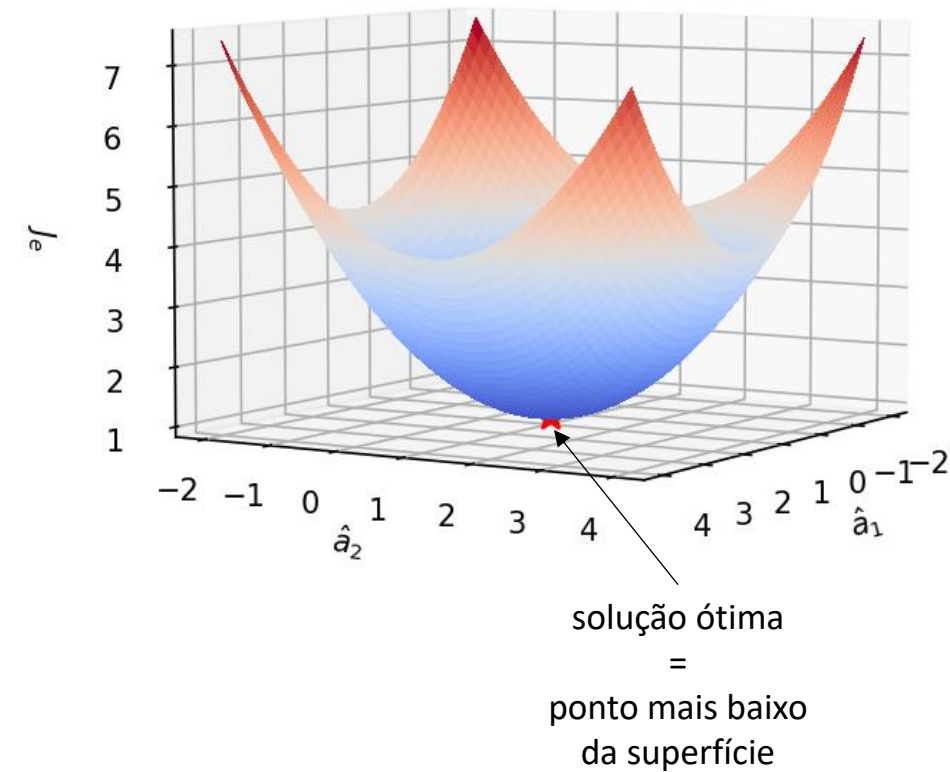
$$h(\mathbf{x}(n), \hat{\mathbf{a}}) = \hat{y}(n) = \hat{a}_1 x_1(n) + \hat{a}_2 x_2(n).$$

- Como encontraríamos os valores de \hat{a}_1 e \hat{a}_2 ?



OBS.: para não confundirmos com os pesos originais, a_1 e a_2 , vamos usar \hat{a}_1 e \hat{a}_2 para nos referir aos pesos da função hipótese, i.e., as estimativas de a_1 e a_2 .

Plotando a superfície de erro

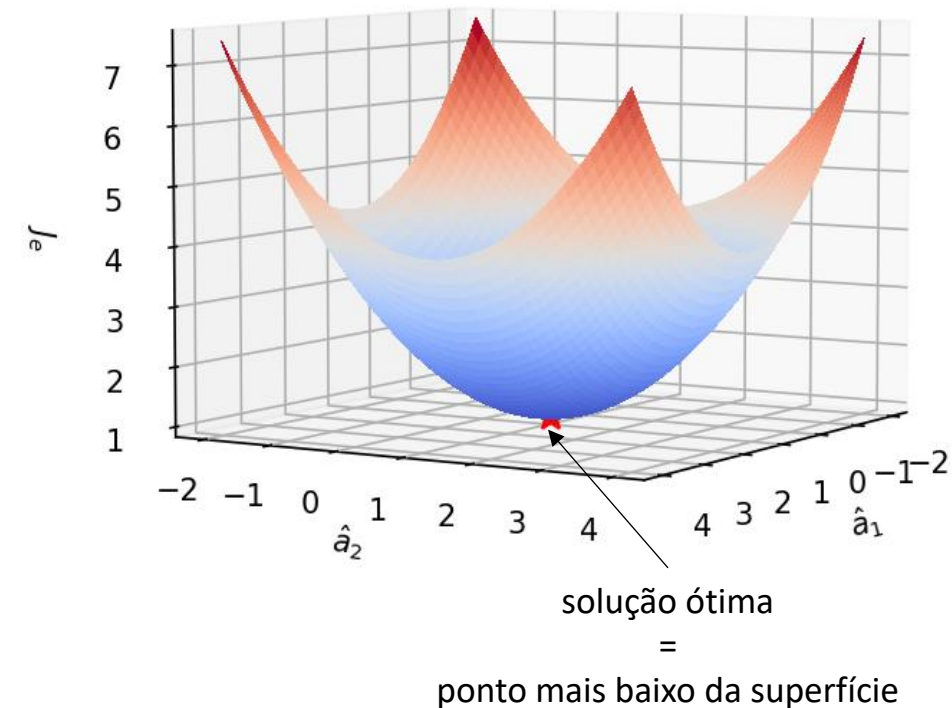


- Até o momento, conseguiríamos encontrar o **ponto de mínimo** com a **equação normal** ou **visualmente**, plotando a **superfície de erro** a partir da função do EQM

$$\begin{aligned} J_e(\hat{a}_1, \hat{a}_2) &= \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - \hat{y}(n) \right)^2 \\ &= \frac{1}{N} \sum_{n=0}^{N-1} \left(y_{\text{noisy}}(n) - (\hat{a}_1 x_1(n) + \hat{a}_2 x_2(n)) \right)^2. \end{aligned}$$

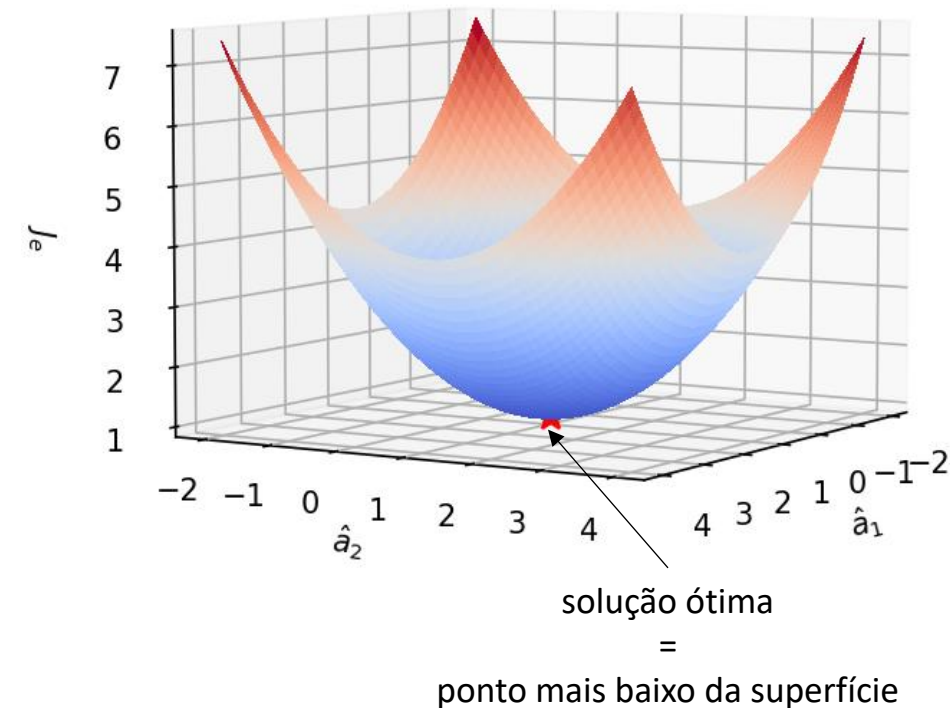
e **procurando por sua parte mais baixa**.

Plotando a superfície de erro



- Os valores de erro, $J_e(\hat{a}_1, \hat{a}_2)$, para plotarmos a **superfície de erro** são obtidos variando-se \hat{a}_1 e \hat{a}_2 na equação do EQM.
- Para este exemplo, onde temos dois atributos, a **superfície de erro** é **representada por uma figura em 3 dimensões**, onde cada par de valores \hat{a}_1 e \hat{a}_2 corresponde a um erro, $J_e(\hat{a}_1, \hat{a}_2)$.

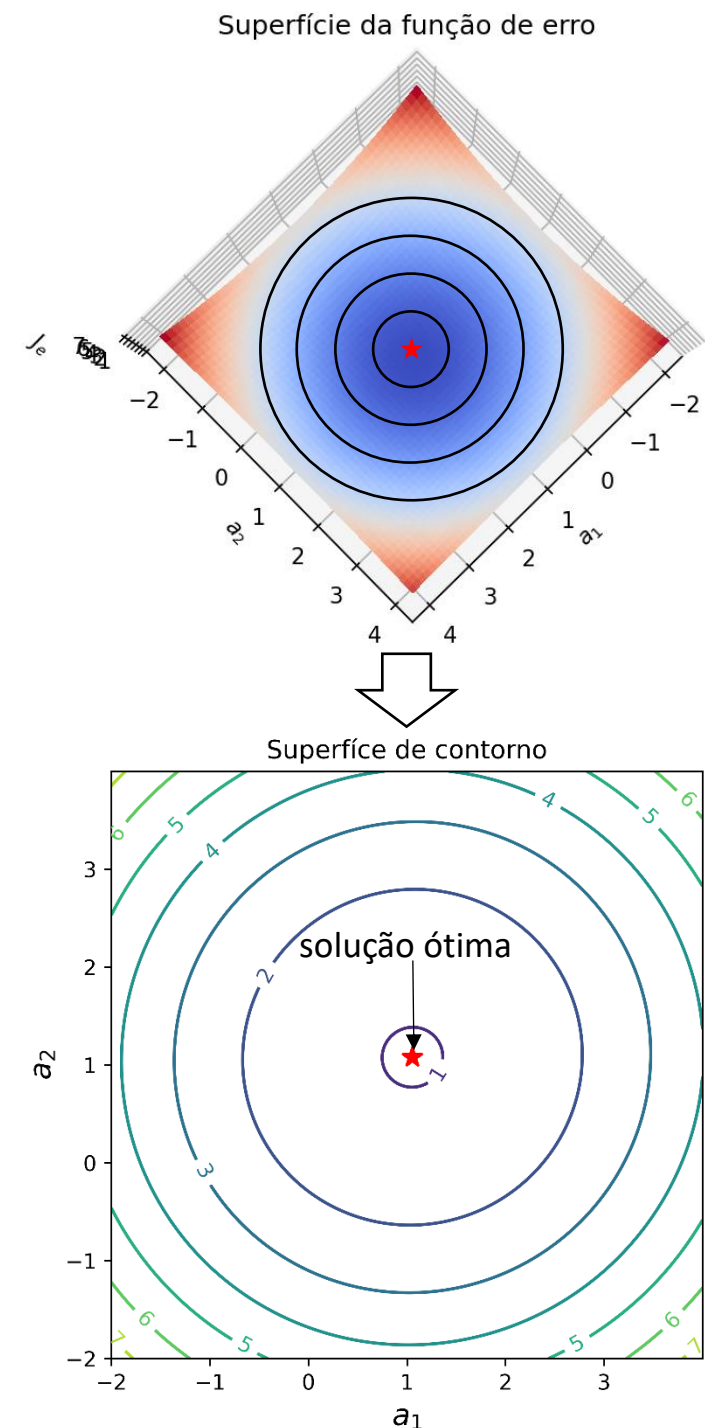
Plotando a superfície de erro



- Devido a superfície ser **convexa**, temos apenas **um ponto de mínimo**, o **mínimo global**.
- O **mínimo global** é ponto mais baixo da superfície e nos dá a solução ótima.
- **Observações**
 - Só conseguimos encontrar o ponto de mínimo de **forma visual** se tivermos funções hipóteses com até 2 atributos.
 - Além disso, é uma **solução com altíssima complexidade computacional**, pois usa a força bruta para encontrar o ponto ótimo.
 - Assim, em geral, usamos a forma fechada.

Plotando a superfície de contorno

- Outra figura importante que plotamos a partir dos resultados obtidos para plotar a superfície de erro é chamada de **superfície de contorno**.
- Uma superfície de contorno contém várias **linhas de contorno**.
 - Uma **linha de contorno** é uma **curva ao longo da qual uma função tem um valor constante**.
- Na superfície de erro, cada linha indica uma curva ao longo da qual o **erro é constante**.
- Ou seja, **qualquer par de valores \hat{a}_1 e \hat{a}_2 ao longo de uma mesma curva terá o mesmo valor de erro**.



***Porém, nem tudo são flores com a
equação normal...***

Desvantagens da equação normal

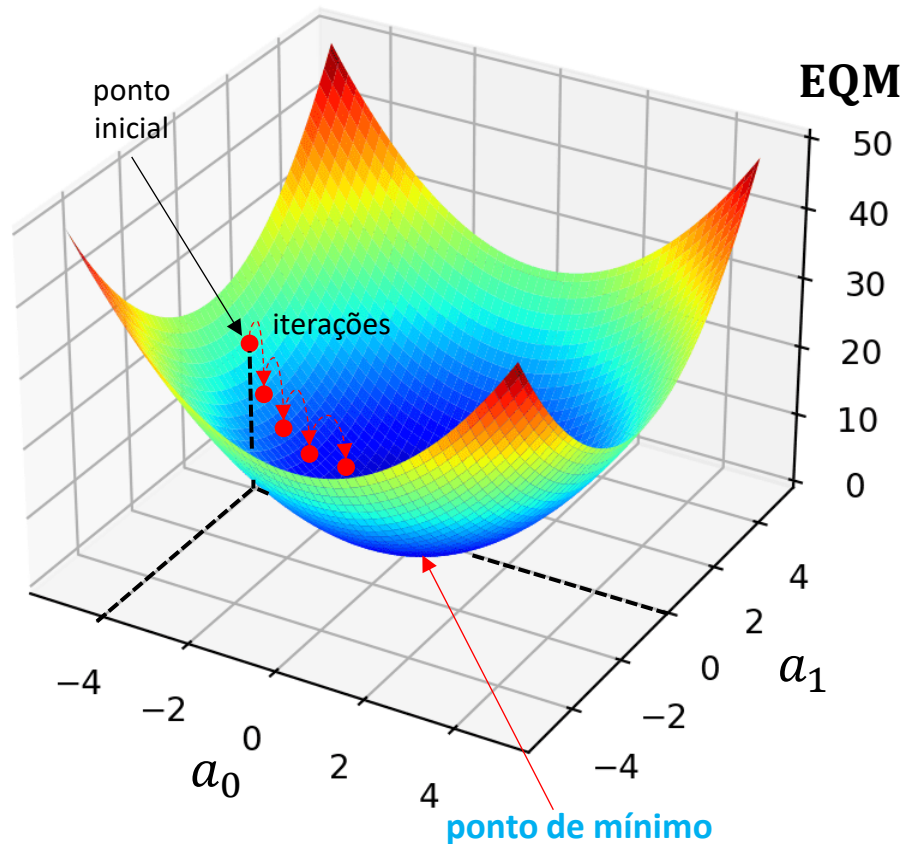
- **Alta complexidade computacional:** a solução da *equação normal* envolve o *cálculo da inversa* de $X^T X$.
- Esse cálculo tem complexidade computacional que varia de $O(K^{2.4})$ a $O(K^3)$, onde K é o número de atributos.
 - “Big O” é uma notação usada para analisar como o *tempo de execução de um algoritmo cresce à medida que o número de entradas aumenta*.
 - **Exemplo:** Se o número de *atributos*, K , passar de 1 para 2, o tempo para cálculo aumenta de $2^{2.4} = 5.3$ a $2^3 = 8$ vezes.
- Alguns conjuntos de dados podem ter centenas ou até milhares de atributos!
 - **House Prices** é um dataset com informações de casas que possui 79 atributos.
 - **ImageNet** é um dataset com imagens, onde cada imagem tem em média $256 \times 256 = 65536$ *pixels* e cada *pixel* é um atributo.

Desvantagens da equação normal

- Além disso, dependendo do número de ***exemplos do conjunto de treinamento***, N , e de ***atributos***, x , a matriz de atributos, X , pode se tornar ***muito grande***.
- Consequentemente, a inversão de matrizes e os cálculos associados podem ***consumir muita CPU e memória***.
 - O que pode ser um limitante para dispositivos com recursos restritos (e.g., IoT).
- Adicionalmente, para irmos ***além dos modelos lineares*** (i.e., modelos não-lineares como classificadores e redes neurais) precisamos lidar com o fato de que ***nem sempre existem formas fechadas*** como a ***equação normal***.
- **Portanto, essa abordagem não é escalonável e muito menos flexível!**

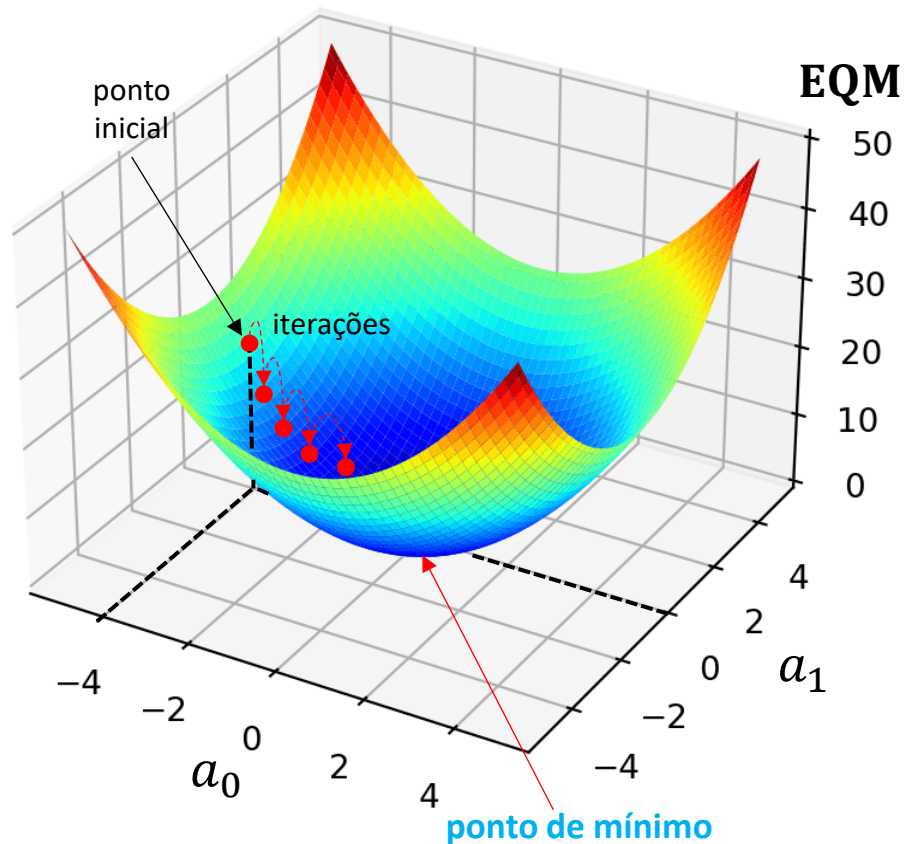
Então o que pode ser feito?

Abordagens iterativas de otimização



- Podemos usar *algoritmos de otimização iterativos*.
- Eles têm *complexidade menor e ajustável* e são *flexíveis* o suficiente para trabalhar com modelos lineares e não lineares.
- Esses algoritmos “*procuram*” de forma *iterativa* o *ponto mais baixo da superfície de erro*, ou seja, o ponto que nos dá os *pesos ótimos*.

Abordagens iterativas de otimização

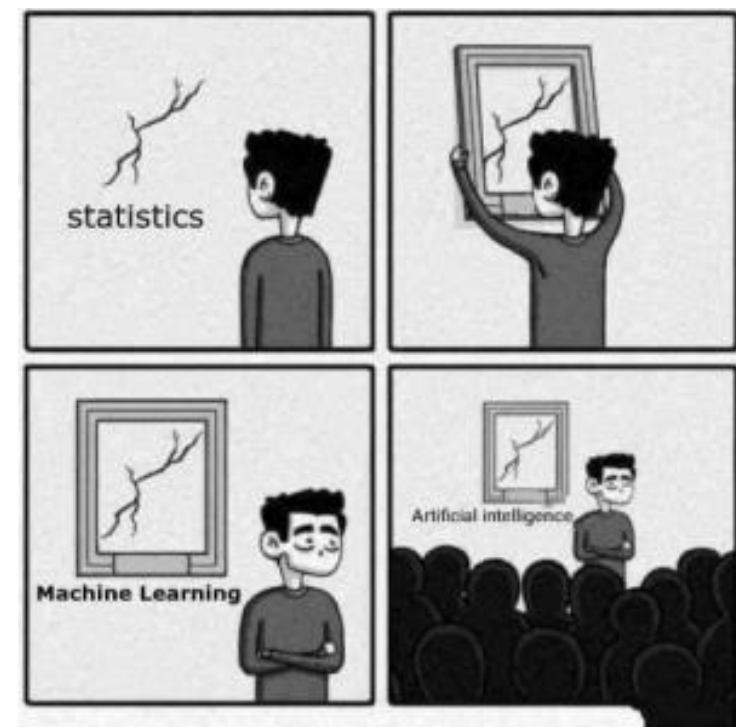
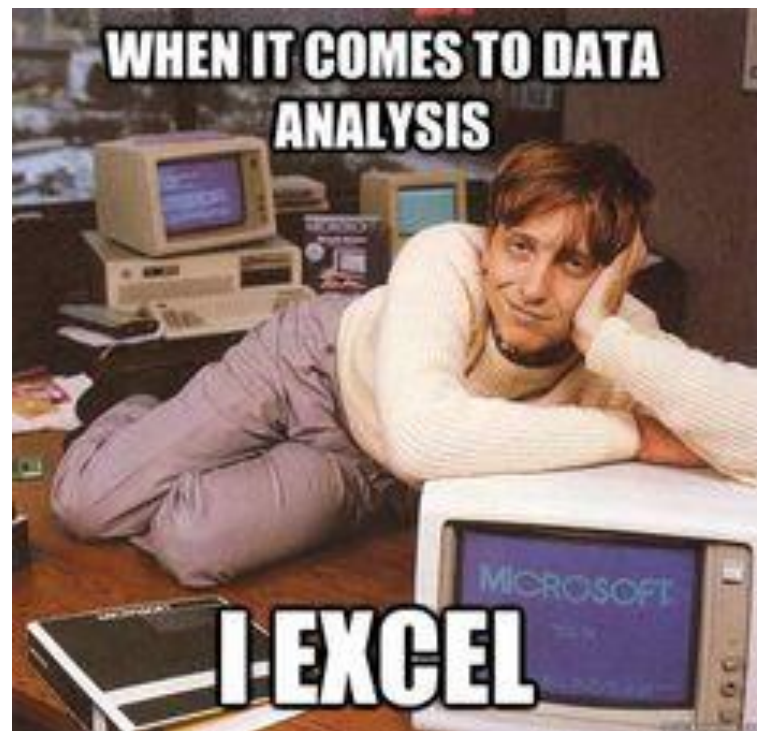
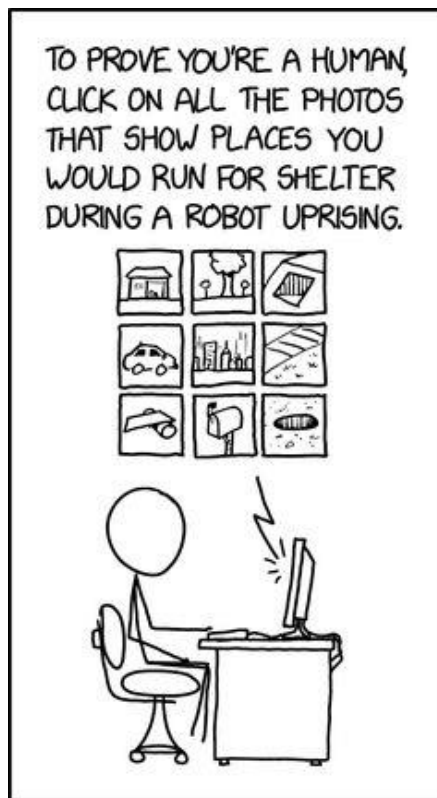


- A **cada iteração**, esses algoritmos **refinam** os valores dos pesos, ficando mais e mais próximos do **ponto de mínimo**.
- Um exemplo de abordagem iterativa é o algoritmo do **gradiente descendente (GD)**.
- O GD busca **iterativamente**, a partir de um ponto inicial, o **ponto mais baixo da superfície de erro**.

Tarefas

- **Quiz:** “*T319 - Quiz - Regressão: Parte I*” que se encontra no MS Teams.
- **Exercício Prático:** [Laboratório #2](#).
 - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
 - Vídeo explicando o laboratório: Arquivos -> Recordings -> Laboratório #2
 - Se atentem aos prazos de entrega.
 - [Instruções para resolução e entrega dos laboratórios](#).

Obrigado!



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

Machine learning:



