



# Introdução a Machine Learning 2

Semana 2:

- Modelos
- Função custo
- Método do gradiente
- Conjunto de treinamento, validação e teste
- O trade-off entre viés e variância
- Não existe almoço grátis

# O que é um modelo?

---



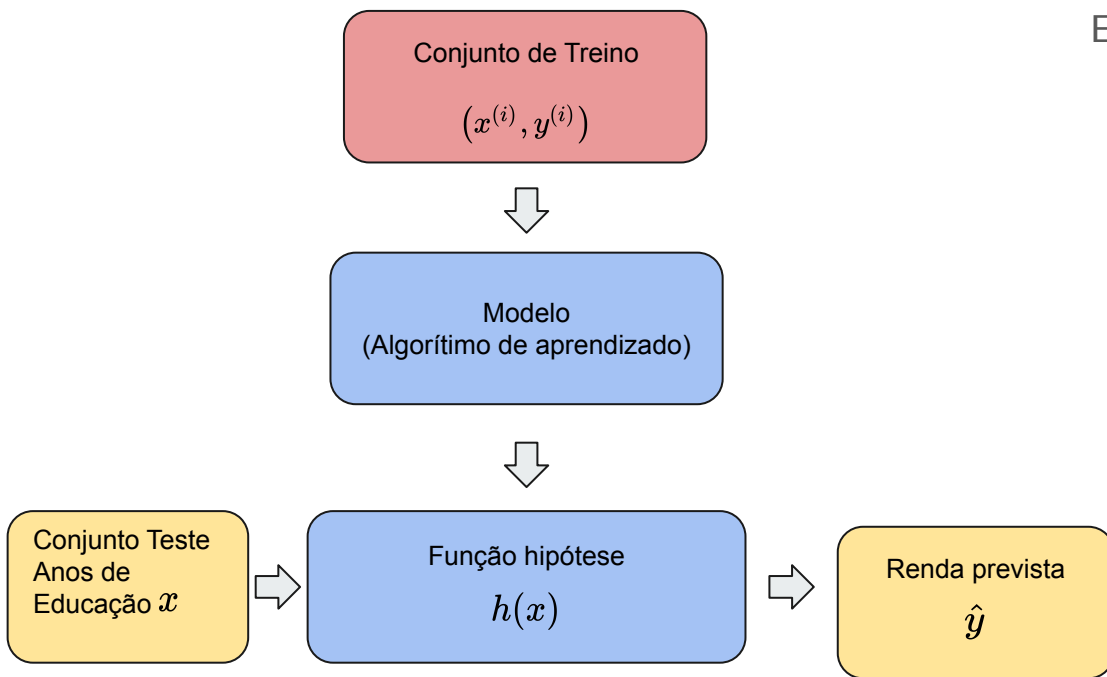
# Representação de um modelo

- Detalhes do modelo:
- Uma variável objetivo,  $y$ ; também chamada de rótulo, ou output (é o que queremos prever).
- Variáveis  $X$ , características ou input (o que vamos usar para explicar)
- $N$  exemplos  $(x^{(i)}, y^{(i)})$
- Valores estimados tem um chapéu:  $\hat{y}$

Observação	Anos de escolaridade (x)	Renda anual (y)
1	8	60,000
2	16	116,000
...	...	...
N	12	97,000



# Exemplo de modelo



Exemplo de regressão linear:

# O que é um bom modelo?



- Campos de estudo diferentes tem noções diferentes do que é um bom modelo.
- Algumas características de bons modelos:
  - Poder explicativo - mapa do metro
  - Poder predictivo - quão boas são as previsões do modelo
  - Falseabilidade - sabemos se o model está errado
  - Simplicidade - navalha de Occam
  - Generalizável - se aplica a situações diferentes
- Em machine learning - bom modelo é aquele que faz boas previsões em uma nova base de dados (out of sample).




# Conjuntos de treino, validação e teste

# Conjuntos de Treino, Validação e teste



Educação (x)	Renda (y)
8	60,000
16	116,000
8	80,000
12	146,000
10	125,000
15	146,000
12	136,000
10	125,000
15	146,000
15	146,000

# Conjuntos de Treino, Validação e teste



Educação (x)	Renda (y)
8	60,000
16	116,000
8	80,000
12	146,000
10	125,000
15	146,000
12	136,000
10	125,000
15	146,000
15	146,000

- O conjunto de treino é usado para estimar os parâmetros do seu modelo.
- O conjunto de validação serve escolhemos os hiperparâmetros e decidir qual modelo vamos usar.
- O conjunto teste funciona como a medida final da performance do modelo.



# Qual a proporção dos dados irá para cada conjunto



Educação (x)	Renda (y)
8	60,000
16	116,000
8	80,000
12	146,000
10	125,000
15	146,000
12	136,000
10	125,000
15	146,000
15	146,000

- Se conjunto de treino for pequeno, a variância dos parâmetros será grande.
- Se o conjunto de validação for pequeno a variância da estatística de seleção de modelos será grande.
- Se co conjunto teste for pequeno a variância do seu teste do modelo será grande.
- Em prática vários manuais recomendam dividir (70,15,15) .
- Se você tiver muitos dados, a proporção de dados no conjunto de treino tende a aumentar.

# Como dividir os dados em 3 partes



## Observações independentes

Educação (x)	Renda (y)
8	60,000
16	116,000
8	80,000
12	146,000
10	125,000
15	146,000
12	136,000
10	125,000

## Séries temporais

Tempo (x)	Preço (y)
1	60
2	62
3	65
4	64
5	68
6	66
7	69
8	71



# Função Custo e como avaliar um modelo

# Função custo: motivação

$N$  = número de observações

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

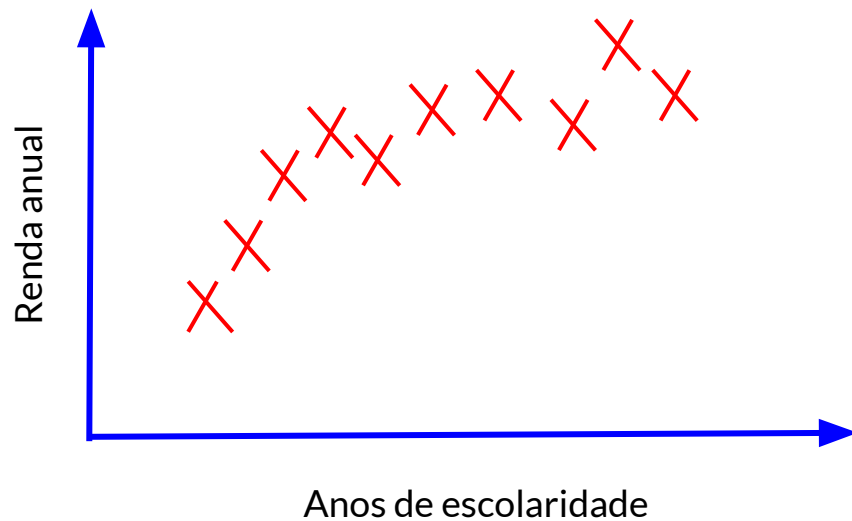
$\theta'_i$  : parâmetros

Como parâmetros diferentes geram modelos diferentes?

Como escolher parâmetros?



# Erro quadrático médio (EQM)

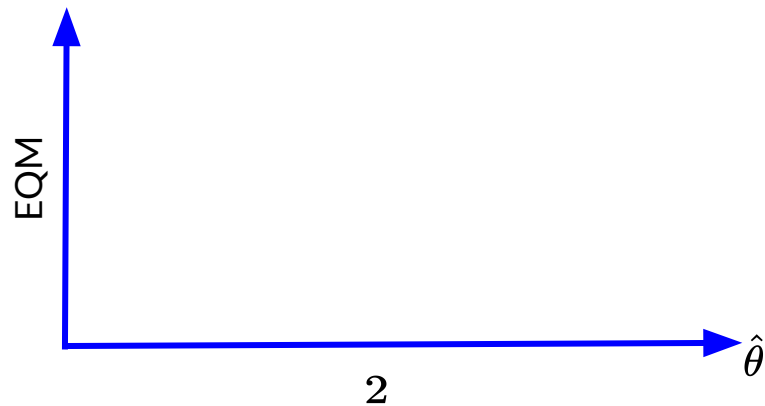
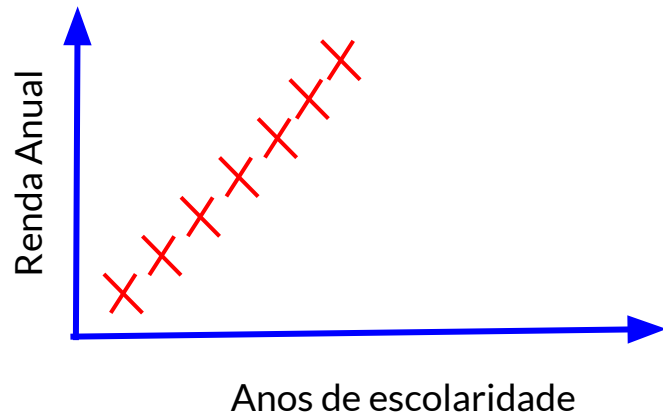


# Função custo Intuição

Suponha  $y = 2 * x$

Só há um parâmetro, a inclinação da curva, que nesse caso é igual a 2.

Como a função custo (EQM) muda de acordo com o nosso parâmetro estimado  $\hat{\theta}$ ?

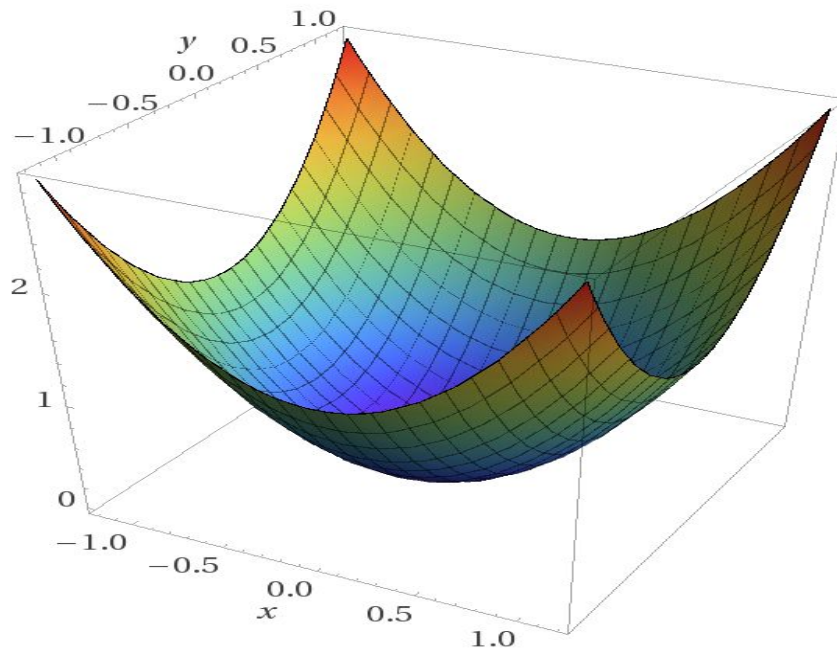




**Método do gradiente ajuda a  
minimizar a função custo.**

# O que é método do gradiente?

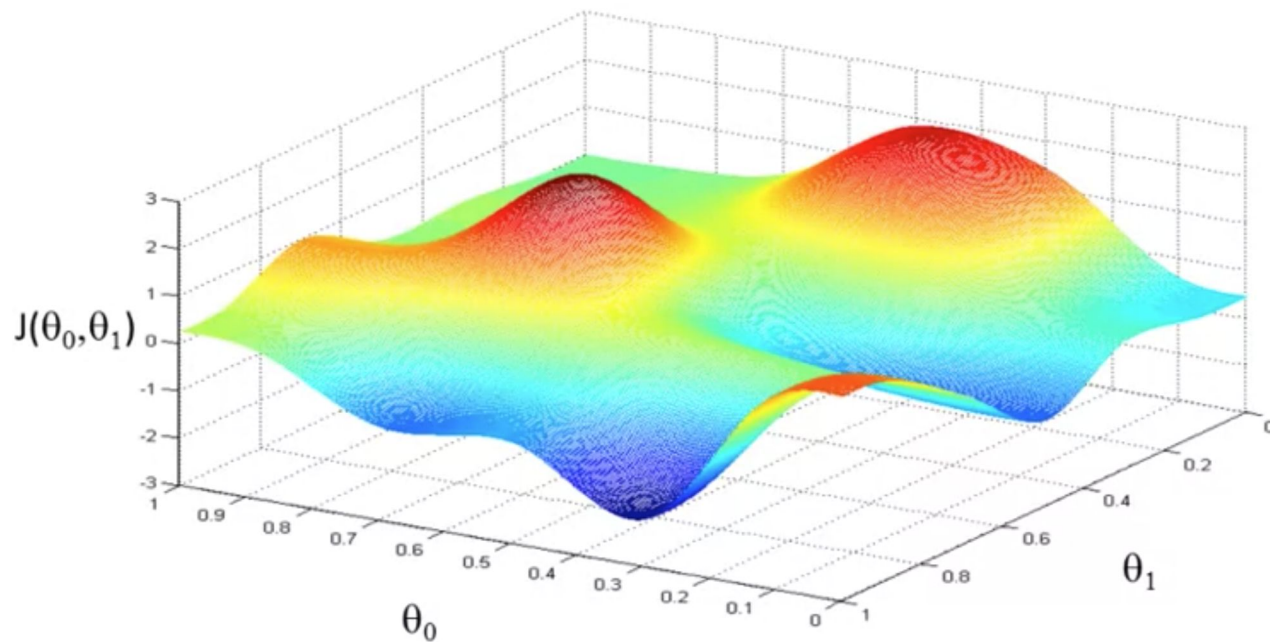
- Podemos minimizar a função custo usando o método do gradiente.
- Relembrando: gradiente é um vetor com a derivada da função com respeito a cada uma das variáveis.
- O gradiente sempre aponta na direção de maior aumento da função (logo a gente sempre se move na direção oposta)



Computed by Wolfram|Alpha



# Um exemplo mais complicado



## Método do gradiente pseudo código



Objetivo:  $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

Comece com valores aleatórios:  $\theta_0, \theta_1$

Enquanto não estiver em um mínimo:

calcule o gradiente de  $J(\theta_0, \theta_1)$

atualize  $\theta_0, \theta_1$  conforme sua derivada

# Matemática do método do gradiente



Atualize os parâmetros simultaneamente até convergir:

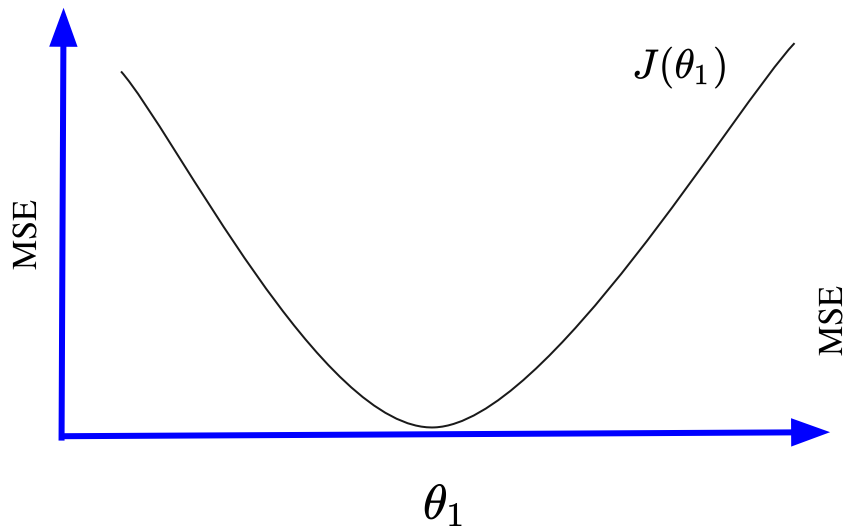
$$\theta_0 := \theta_0 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0}$$

$$\theta_1 := \theta_1 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1}$$

O que é alpha?

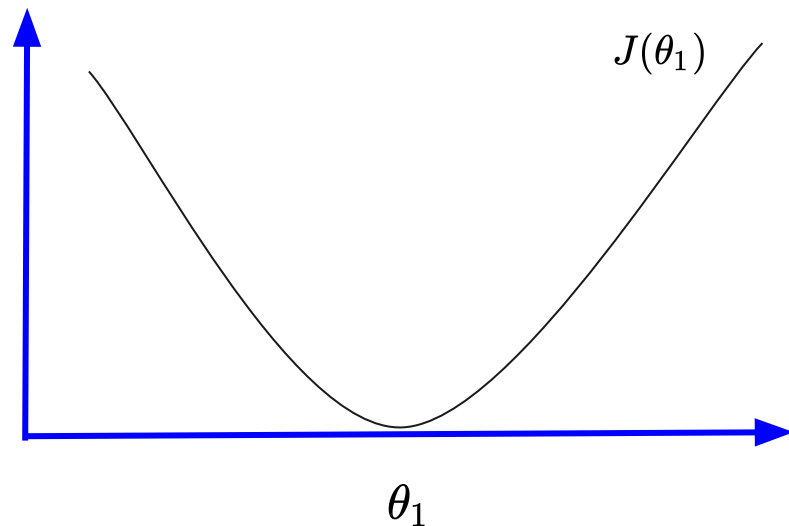
# Hyperparâmetro: velocidade do aprendizado

Velocidade de aprendizado pequena demais



gradiente é muito lento

Velocidade de aprendizado muito grande



gradiente pode saltar o mínimo

# Método do gradiente na regressão linear



$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

$$J(\theta_0, \theta_1) = \frac{1}{2N} \sum_{i=1}^N (h_{\theta}(x^i) - y^i)^2$$

# Tipos diferentes do método do gradiente



1. Gradiente em lotes (batches)Batch Gradient: Em cada iteração o algoritmo usa todas as  $N$  observações do conjunto de treinamento (baixo custo computacional, alto custo de memória).
2. Gradiente em mini-lotes (mini batches): em cada iteração usamos um subconjunto de  $K$  observações para atualizar os parâmetros.
3. Gradiente estocástico: calcula o gradiente e atualiza o modelo depois de cada observação (custo computacional alto, baixo custo de memória).



# Usando a função custo para escolher o modelo

# Relembrando sobre-adequação e sub-adequação



- Sobre-adequação (overfitting) ocorre quando o modelo é muito complexo, ele explica o conjunto de treinamento bem demais e não extrapola bem para outras bases de dados.
- Sub-adequação (underfitting) ocorre quando o modelo é muito simples e nem mesmo explica bem o conjunto de treinamento.





# Usando a função custo para escolher o modelo



$$\text{Treino EQM} = \frac{1}{2N_{treino}} \sum_{i=1}^{N_{treino}} (\hat{y}_i - y_i)^2$$

$$\text{Val EQM} = \frac{1}{2N_{val}} \sum_{j=1}^{N_{val}} (\hat{y}_j - y_j)^2$$

$$\text{Teste EQM} = \frac{1}{2N_{teste}} \sum_{k=1}^{N_{teste}} (\hat{y}_k - y_k)^2$$

Podemos usar o custo no conjunto de treino e de validação para julgar se o modelo está sobre ou sub adequado.

Se o modelo tem uma performance ruim no conjunto de treino, ele provavelmente está sub-adequado.

Se a diferença entre o custo no conjunto treino e validação for grande, o modelo está sobre-adequado.

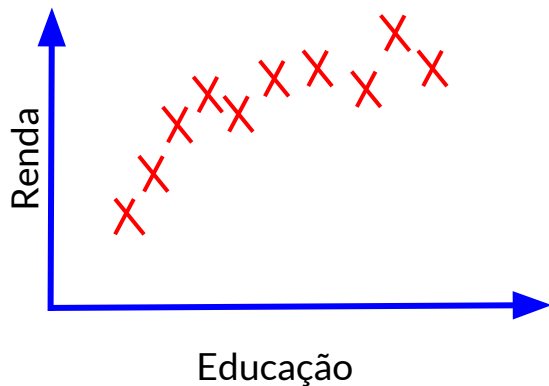


# Dicotomia entre viés e variância

# Viés e Variância

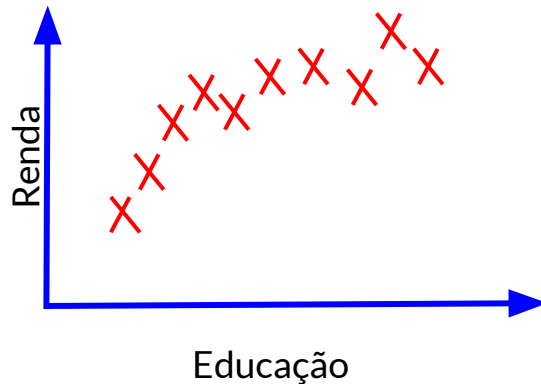


Vies (Sub-adequação)

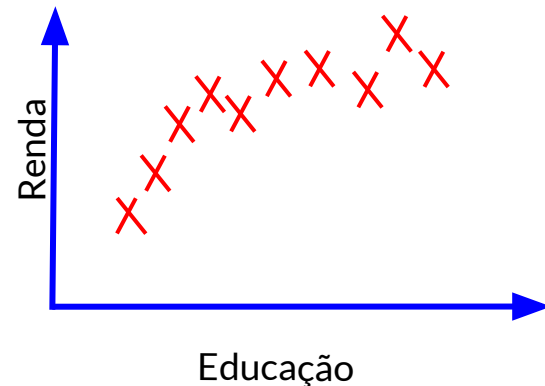


Modelo menos complexo

“Caso ideal”

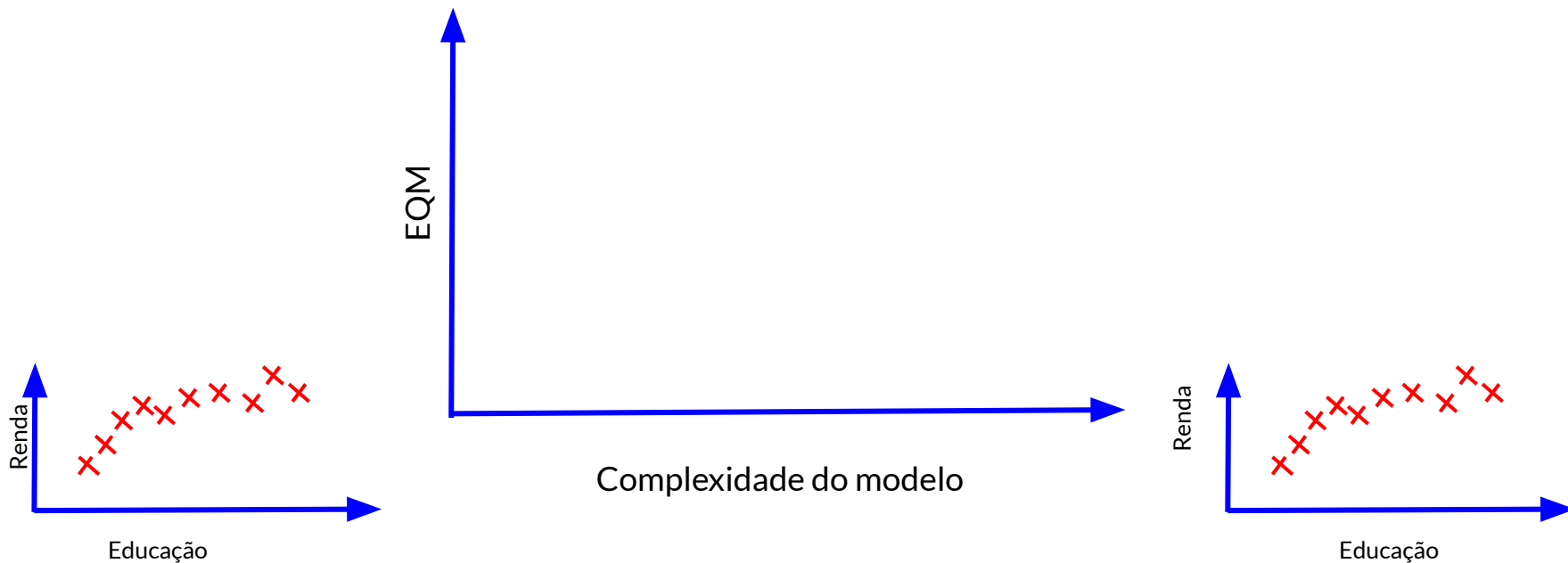


Variância (Sobre-adequação)



Modelo mais complexo

# Dicotomia entre viés e variância





**Não há almoço grátis teorema  
(No free lunch)**

# Não há almoço grátis teorema



- [Não há almoço grátis teorema.](#)
- Em economia, não há almoço grátis se refere ao custo de oportunidade
- Paper: “The Lack of A Priori Distinctions Between Learning Algorithms”.
- Em ML: sem nenhuma suposição sobre a base de dados, não há razão alguma para preferir um modelo ao outro.

# Não há almoço grátis

- Para alguma base de dados, o melhor modelo é uma regressão linear, para outra é uma floresta aleatória.
- Não há um modelo que podemos garantir que sempre funcionará melhor
- Quão pertinente esse teorema é na prática? Há modelos que são melhores que outros na maioria dos problemas que estamos interessados?

