

Machine Learning

Aula 03 $\{x_n\}: x_n = \frac{1}{n}; \{y_n\} =$

Evandro J.R. Silva

Uninassau Teresina

Sumário

1 Principais desafios do Aprendizado de Máquina

Quantidade insuficiente de dados de treinamento

Dados de treinamento não representativos

Dados de baixa qualidade

Características irrelevantes

Sobreajuste dos dados de treinamento

Subajuste dos dados de treinamento

2 Teste e Validação

k-fold

Bootstrap

Treinamento, Teste e Validação

3 Algoritmos Meta-Heurísticos

4 FIM

1 Principais desafios do Aprendizado de Máquina

Quantidade insuficiente de dados de treinamento

Dados de treinamento não representativos

Dados de baixa qualidade

Características irrelevantes

Sobreajuste dos dados de treinamento

Subajuste dos dados de treinamento

2 Teste e Validação

k-fold

Bootstrap

Treinamento, Teste e Validação

Algoritmos Meta-Heurísticos

FIM

Principais desafios do Aprendizado de Máquina

- Alguns dos principais desafios ao Aprendizado de Máquina são:

- Quantidade insuficiente de dados de treinamento
- Dados de treinamento não representativos
- Dados de baixa qualidade
- Características irrelevantes
- Sobreajuste dos dados de treinamento
- Subajuste dados de treinamento

Quantidade insuficiente de dados de treinamento

- Nós seres humanos, sem percebermos, conseguimos extrair uma quantidade enorme de dados das coisas ao nosso redor.
- Nós temos a capacidade de **extração de características** de forma **automática e avançada!**
- Para melhorar temos uma quantidade enorme de sensores, cujos dados são todos levados em conta de alguma forma quando registramos alguma observação.
- Para melhorar ainda mais, temos **acesso abundante** a [virtualmente] **infinitas amostras** de variadas coisas.

Quantidade insuficiente de dados de treinamento

Imagine como seria uma foto das costas dessas pessoas, como se tivessem sido tiradas no mesmo momento.

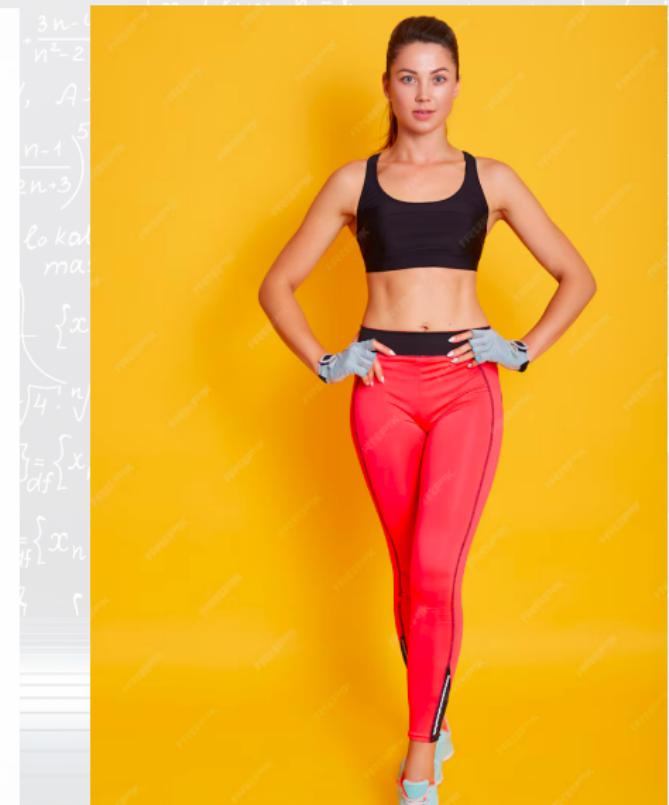
$$\left\{ \frac{1}{n} \right\}_{n=1}^{\infty} = \left\{ \frac{1}{n} \right\}_{n=1}^{\infty}$$

$x_n \leq y_n \leq z_n$

$\downarrow n \rightarrow \infty \quad \downarrow n \rightarrow \infty \quad \downarrow n \rightarrow \infty$

$g \quad g \quad g$

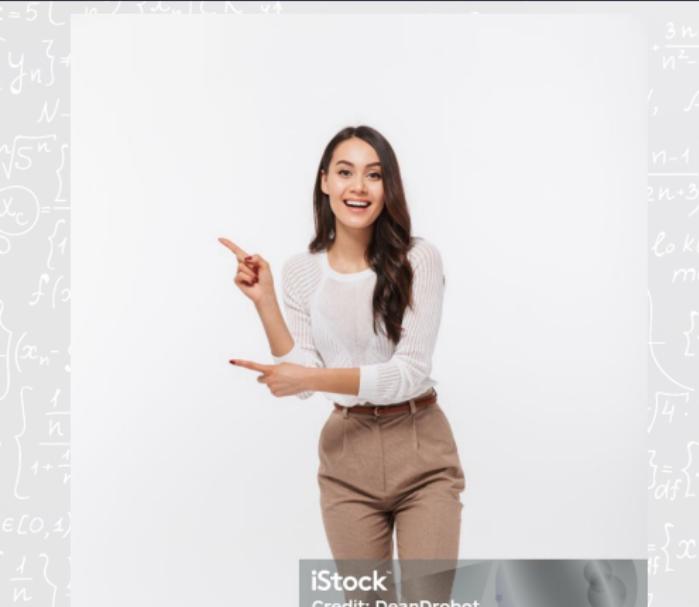
Quantidade insuficiente de dados de treinamento



Quantidade insuficiente de dados de treinamento



Quantidade insuficiente de dados de treinamento



iStock

Credit: DeanDrobot



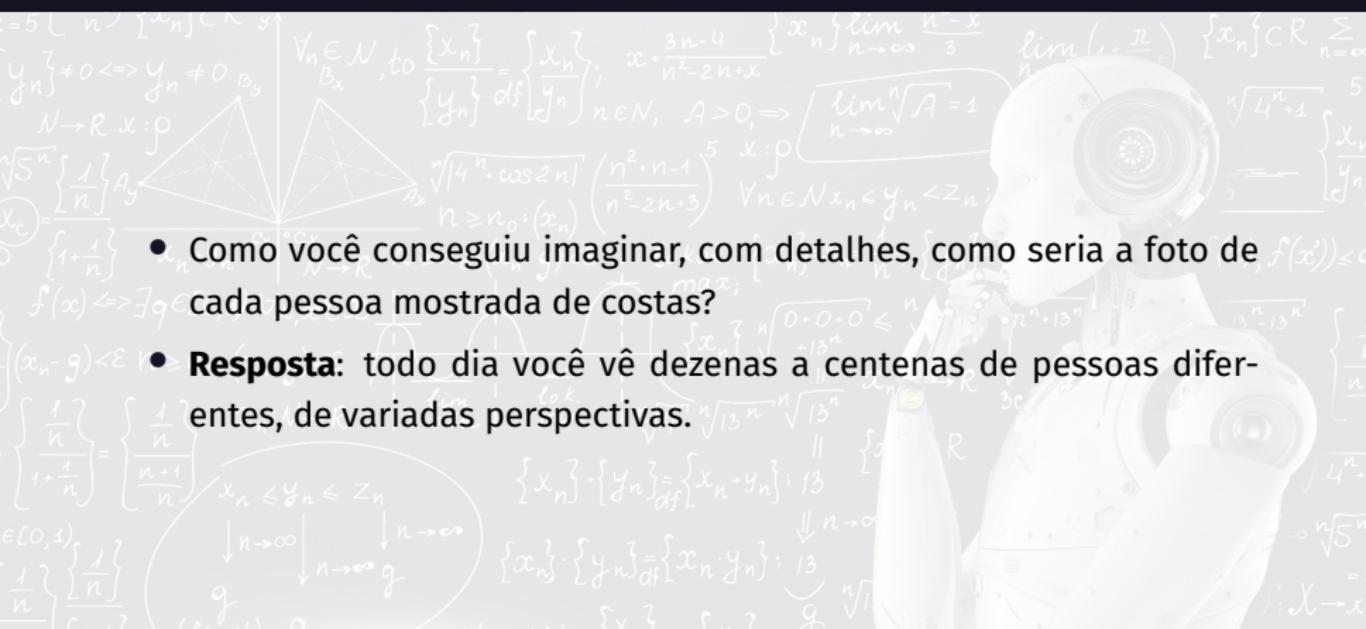
iStock

Credit: Yuri_A

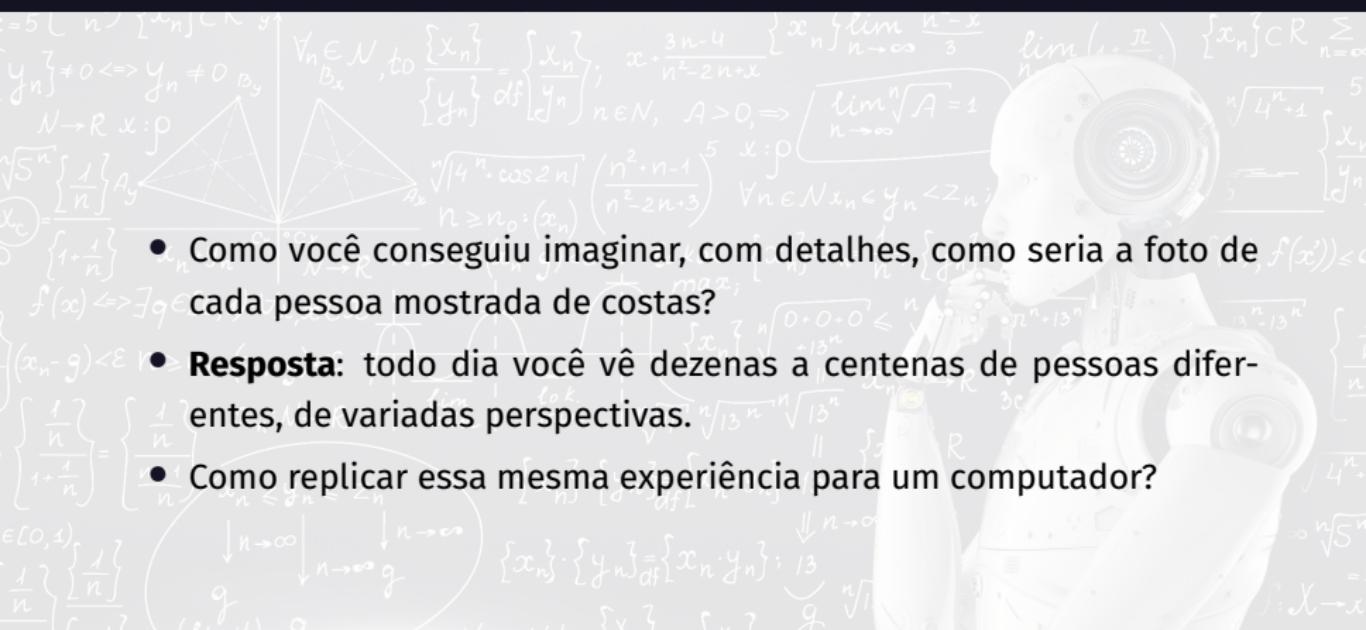
Quantidade insuficiente de dados de treinamento

- Como você conseguiu imaginar, com detalhes, como seria a foto de cada pessoa mostrada de costas?

Quantidade insuficiente de dados de treinamento

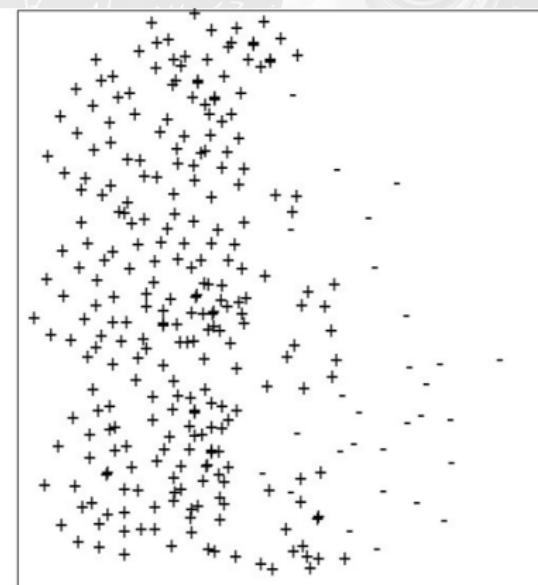
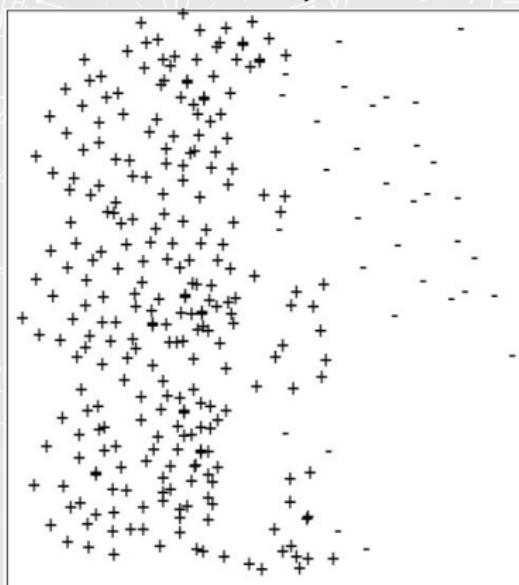


Quantidade insuficiente de dados de treinamento

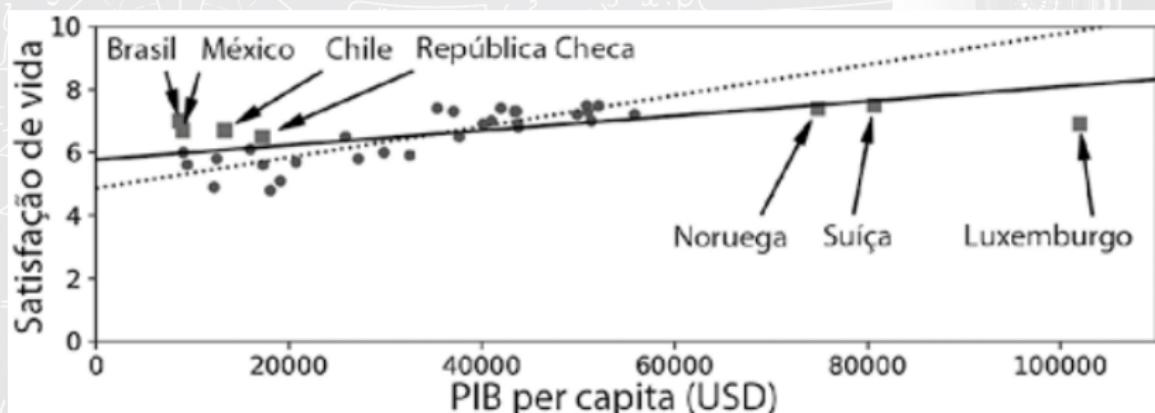


Dados de treinamento não representativos

- Data set shift e viés de amostragem (*sampling bias*) – o problema pode estar na própria base, ou em como as instâncias foram selecionadas para treinamento e teste.



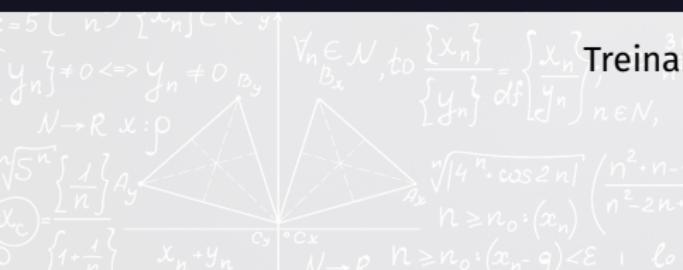
Dados de treinamento não representativos



Quantidade insuficiente de dados de treinamento

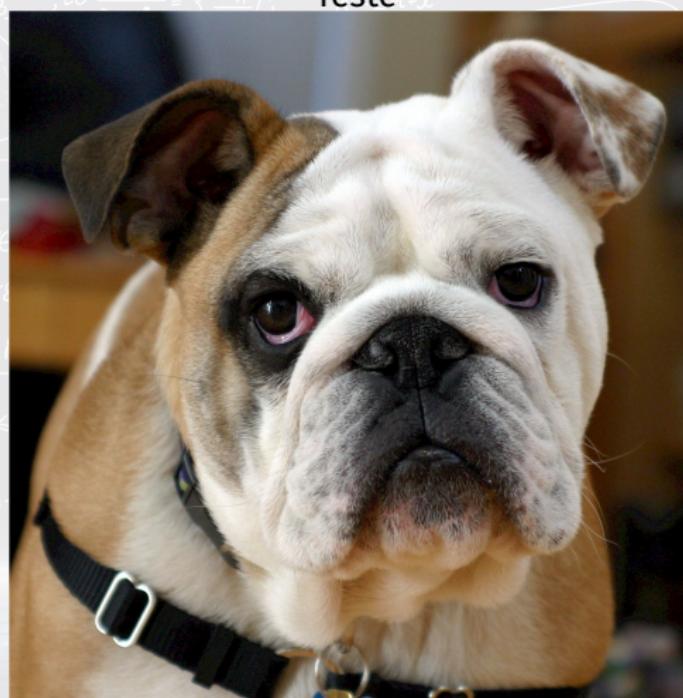


Quantidade insuficiente de dados de treinamento



Quantidade insuficiente de dados de treinamento

$$\begin{aligned} & \{x_n\}_{n=1}^{\infty} \subset \mathbb{R}^2, \quad \{y_n\}_{n=1}^{\infty} \subset \mathbb{R}, \\ & \{y_n\}_{n=1}^{\infty} \neq 0 \Leftrightarrow y_n \neq 0, \quad \forall n \in \mathbb{N}, \\ & N \rightarrow R, \quad x: p, \\ & \sqrt{5^n} \left\{ \frac{1}{n} \right\} A_y, \quad C_y = \left\{ 1 + \frac{1}{n} \right\} N, \quad C_x = \left\{ 1 + \frac{1}{n} \right\} N, \\ & f(x) \Leftrightarrow \exists q \in [0, 1]: \forall x, x' \in \\ & \{(x_n - g) < \varepsilon \mid n \geq n_0\}: (x_n - g) < \varepsilon, \\ & \left\{ \frac{1}{n} \right\}_{1+1/n}^{1/n} = \left\{ \frac{1}{n} \right\}_{n+1/n}^{n}, \quad x_n \rightarrow 1, \quad x_n \leq y_n \leq \\ & \varepsilon \in [0, 1], \quad \left\{ \frac{1}{n} \right\}_{1+1/n}^{1/n} \left(\frac{1}{n}, \frac{1}{n+1} \right) \end{aligned}$$



$$\begin{aligned} & \forall n \in \mathbb{N}, \text{ to } \{x_n\} = \{x_n\}, \quad x \in \text{Teste}_{n+1}, \quad \{x_n\}_{n=1}^{\infty} \text{ s.t. } \lim_{n \rightarrow \infty} \frac{x_n - x}{n} = 0, \\ & \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n} \right)^n = e, \quad \{x_n\} \subset \mathbb{R}, \quad n \geq \infty, \\ & \sqrt[n]{4^n + 1} = \sqrt[n]{4^n} + \sqrt[n]{1} = 4 + 1 = 5, \quad f(x), f'(x') \leq 0, \\ & n+13^n, \quad 13^n+13^n, \quad \left\{ \frac{1}{n} \right\}_{1+1/n}^{1/n} \end{aligned}$$

Dados de baixa qualidade

- Dados com muitos erros, **outliers** ou **ruidos** dificultam, e até atrapalham a aprendizagem.
- Seres humanos são tendentes ao erro, principalmente devido à fadiga. Muitos dados foram levantados por pessoas se esforçando fisicamente para coletá-los.
- Solução: investir tempo no pré-processamento.

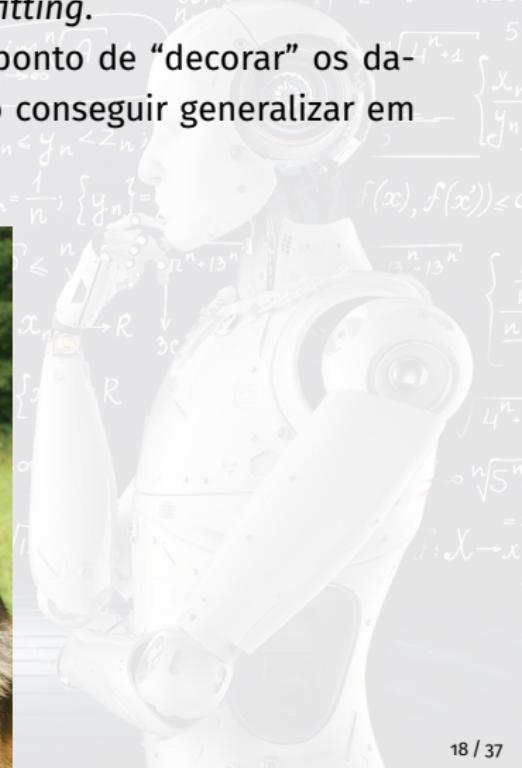
Características irrelevantes

- Quanto mais discriminantes forem as características, melhor.
- Porém, é difícil saber durante a construção de uma base quais as características serão, de fato, discriminantes. Há, inclusive, a possibilidade de características que seriam muito discriminantes não serem coletadas.
- Ex.:
 - A quantidade de portas, ou a quantidade de rodas é inútil para aprendermos a diferença entre os carros a seguir:



Sobreajuste dos dados de treinamento

- O termo é mais conhecido em Inglês: *overfitting*.
- Um determinado modelo é treinado ao ponto de “decorar” os dados de treinamento e, devido a isso, não conseguir generalizar em seguida.
- Ex.: $f(x) \Leftrightarrow \exists g \in [0,1] : \forall x, x \in \mathcal{X} : |f(x) - g| < \epsilon$



Sobreajuste dos dados de treinamento

- O termo é mais conhecido em Inglês: *overfitting*.
- Um determinado modelo é treinado ao ponto de “decorar” os dados de treinamento e, devido a isso, não conseguir generalizar em seguida.
- Ex.: $f(x) \Leftrightarrow \exists g \in [0,1] : \forall x, x' \in \mathcal{X} : |f(x) - g| < \epsilon$



Sobreajuste dos dados de treinamento

- Possíveis soluções

- Escolher algum modelo mais simples (algoritmo mais simples, ou menos parâmetros).
 - Menos tempo de treinamento.
 - Disponibilizar mais dados de treinamento.
 - Reducir mais os ruídos durante o pré-processamento.

Subajuste dos dados de treinamento

- Termo em Inglês: *underfitting*
- É o problema oposto ao sobreajuste. Ou seja, por mais que treine, o modelo não consegue aprender algum conceito.
- Possíveis soluções
 - Selecionar um modelo mais poderoso.
 - Permitir um maior tempo de treinamento.
 - Alimentar o algoritmo de aprendizado com melhores (\neq mais) características.
 - Minimizar restrições no modelo.

Principais desafios do Aprendizado de Máquina

Quantidade insuficiente de dados de treinamento

Dados de treinamento não representativos

Dados de baixa qualidade

Características irrelevantes

Sobreajuste dos dados de treinamento

Subajuste dos dados de treinamento

2 Teste e Validação

k-fold

Bootstrap

Treinamento, Teste e Validação

$\frac{1}{n}$

Algoritmos Meta-Heurísticos

4 FIM

Teste e Validação

- Como mensurar o quanto bom é um modelo de AM?

Teste e Validação

- Como mensurar o quanto bom é um modelo de AM?

- Treina o modelo, disponibiliza para os usuários, e espera os feedbacks
— método **confia no pai!**

Teste e Validação

• Como mensurar o quanto bom é um modelo de AM?

Treinar e validar o modelo antes de ser disponibilizado: método **cross-validation (CV)**.

Teste e Validação

• Como mensurar o quanto bom é um modelo de AM?

• Duas abordagens principais: *k-fold* e *bootstrap*

Teste e Validação

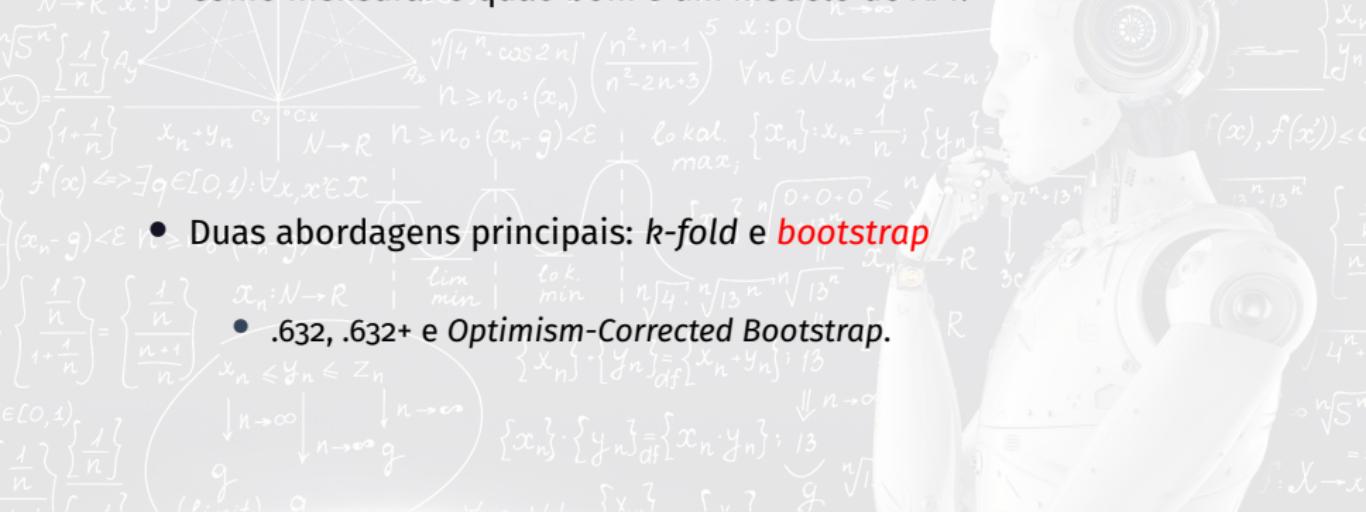
• Como mensurar o quanto bom é um modelo de AM?

• Duas abordagens principais: **k-fold** e **bootstrap**

• 2-fold, 5-fold, 10-fold, leave-one-out e hold-out.

Teste e Validação

- Como mensurar o quanto bom é um modelo de AM?



Teste e Validação

- Como mensurar o quanto bom é um modelo de AM?

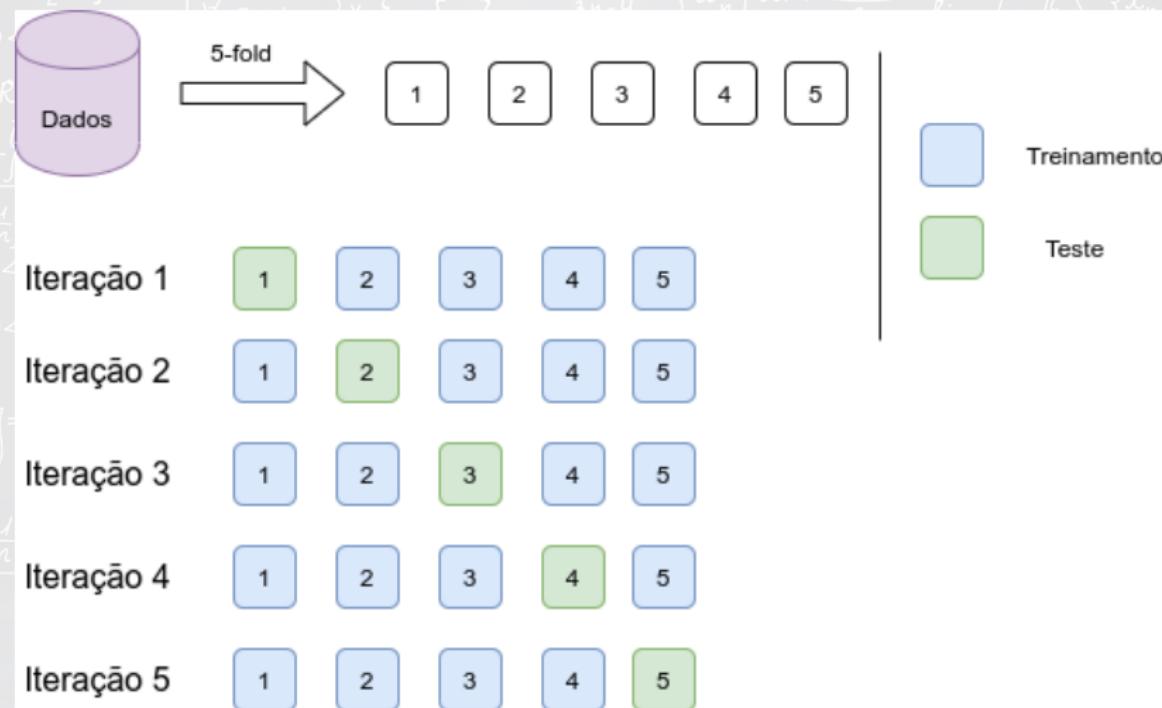
- Duas abordagens principais: *k-fold* e *bootstrap*

- O objetivo é dividir, de alguma forma, os dados disponíveis em dois conjuntos, um para treino e outro para teste.
- A repetição dessa separação, treinamento e teste provê resultados confiáveis sobre o desempenho de um modelo.

k-fold

- **Funcionamento:** $\forall n \geq n_0: (x_n)$
- Esta técnica divide a base de dados aleatoriamente em k partes iguais (ou o mais igual possível). A primeira parte é usada como conjunto de teste e as restantes $k - 1$ partes são usadas para treinamento.
- O treinamento e teste acontecem k vezes, onde em cada nova iteração, os conjuntos de treinamento e teste são diferentes.
- A validação se dá a partir de cálculos estatísticos sobre os k resultados.

k-fold



k-fold

- **2-fold:** base de dados dividida aleatoriamente ao meio.
- **5-fold:** base de dados dividida aleatoriamente em 5 partes.
- **10-fold:** base de dados dividida aleatoriamente em 10 partes.
- **10× 10-fold:** 10-fold repetido 10 vezes (ou seja, o modelo é treinado e testado 100 vezes).
 - **hold-out 50%:** é o 2-fold mas sem repetição.
 - **hold-out 33%:** 1/3 dos dados são selecionados aleatoriamente para o conjunto de teste, e o restante vai para o conjunto de treinamento.
 - **leave-one-out:** apenas uma instância é usada para teste, enquanto todas as outras são usadas para treinamento. Suponha uma base com 100 instâncias; o *leave-one-out* seria um 100-fold.

Bootstrap

- **Funcionamento**

- Seleciona, aleatoriamente e com reposição, o equivalente a 2/3 das amostras como conjunto de treinamento.

Original Dataset

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------

Bootstrap 1

x_8	x_6	x_2	x_9	x_5	x_8	x_1	x_4	x_8	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_{10}
-------	-------	----------

Bootstrap 2

x_{10}	x_1	x_3	x_5	x_1	x_7	x_4	x_2	x_1	x_8
----------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_6	x_9
-------	-------

Bootstrap 3

x_6	x_5	x_4	x_1	x_2	x_4	x_2	x_6	x_9	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_8	x_{10}
-------	-------	-------	----------

Training Sets

Test Sets

- O conjunto de treinamento tende a possuir aproximadamente 63,2% das amostras totais.

- Para validar seu modelo, você pode criar 5, 10, 20, 100 ou mais conjuntos bootstrap. Porém, não é garantido que seu modelo treinará e será testado com todas as instâncias possíveis.

Bootstrap

- O bootstrap .632 é uma forma estatística de se calcular o desempenho do seu modelo como se todas as instâncias tivessem sido utilizadas para treinamento ou teste. Porém, seus valores são “pesimistas”.

- O .632+ tenta calcular o desempenho de forma mais otimista.

Aqui tem uma explicação mais detalhada com exemplo em Python.

- O Optimism-Corrected Bootstrap é outra forma de se calcular o desempenho, porém com o objetivo de evitar estimativas muito otimistas.

Aqui tem uma explicação mais detalhada, com código em R.

Treinamento, Teste e Validação

- Também existem os casos onde os dados são separados em 3 conjuntos: treinamento, validação e teste.

Treinamento, Teste e Validação

- Também existem os casos onde os dados são separados em 3 conjuntos: **treinamento**, validação e teste.
- Continua o mesmo, ou seja, é o conjunto usado para treinar um modelo.

Treinamento, Teste e Validação

- Também existem os casos onde os dados são separados em 3 conjuntos: treinamento, **validação** e teste.

- Conjunto separado do de treino, utilizado para verificar o desempenho de treino, porém, sem os dados utilizados para o treino em si.

Treinamento, Teste e Validação

- Também existem os casos onde os dados são separados em 3 conjuntos: treinamento, validação e teste.

- Continua o mesmo, ou seja, é o conjunto de dados até então não vistos pelo modelo, e utilizado para validar o desempenho após o treinamento.

1 Principais desafios do Aprendizado de Máquina

Quantidade insuficiente de dados de treinamento

Dados de treinamento não representativos

Dados de baixa qualidade

Características irrelevantes

Sobreajuste dos dados de treinamento

Subajuste dos dados de treinamento

2 Teste e Validação

k-fold

Bootstrap

Treinamento, Teste e Validação

g

3 Algoritmos Meta-Heurísticos



Algoritmos Meta-Heurísticos

- Antes de definirmos o que é uma meta-heurística, vamos entender o que é uma heurística.

Heurística

É uma estratégia prática que diminui o tempo de tomada de decisão [fonte].

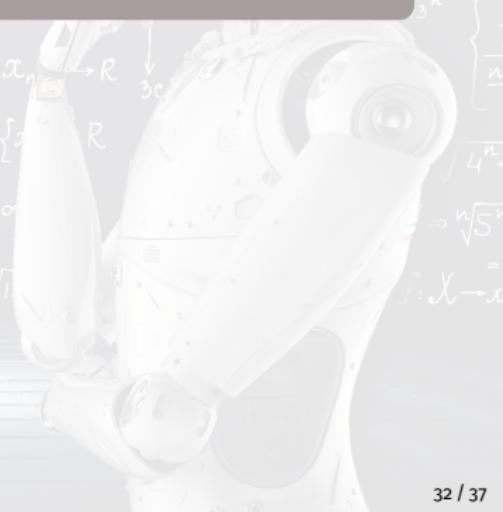
- É, basicamente, o desenvolvimento de alguma estratégia, a partir de alguma informação, que ajude a encontrar uma solução de forma mais rápida e prática.

Algoritmos Meta-Heurísticos

Meta-heurística

Método solução que coordena procedimentos de busca locais com estratégias de mais alto nível, de modo a criar um processo capaz de escapar de mínimos locais e realizar uma busca robusta no espaço de soluções de um problema.

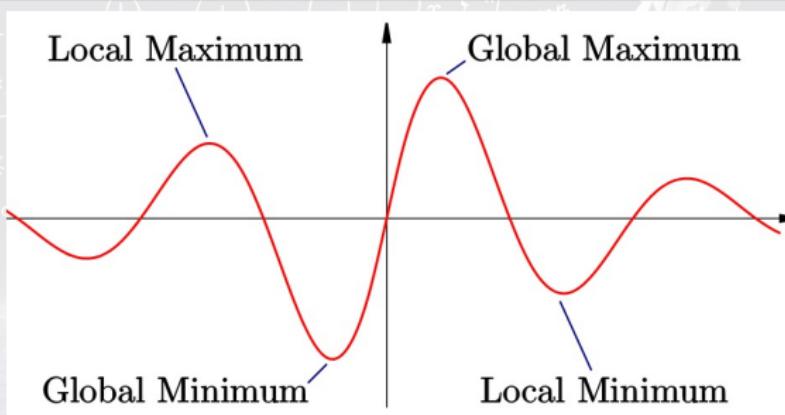
$$\begin{aligned}
 & (x_n - g) < \varepsilon \quad n \geq n_0 \quad (x_n - g) < \varepsilon \\
 & \left\{ \frac{1}{n} \right\} = \left\{ \frac{1}{n+1} \right\} \quad x_n: N \rightarrow R \\
 & \lim_{n \rightarrow \infty} \frac{1}{n} = 0 \quad x_n \leq y_n \leq z_n \\
 & \lim_{n \rightarrow \infty} \frac{1}{n+1} = 0 \quad y_n \rightarrow g \\
 & \lim_{n \rightarrow \infty} (x_n - g) = 0 \quad z_n \rightarrow g
 \end{aligned}$$



Algoritmos Meta-Heurísticos

Meta-heurística

Método solução que coordena procedimentos de busca locais com estratégias de mais alto nível, de modo a criar um processo capaz de escapar de **mínimos locais** e realizar uma busca robusta no espaço de soluções de um problema.



Algoritmos Meta-Heurísticos

- A definição de **meta-heurística** passou a abranger quaisquer procedimentos que empregassem estratégias para escapar de mínimos locais em espaços de busca de soluções complexas.
- Em especial, foram incorporados procedimentos que utilizam o conceito de vizinhança para estabelecer meios de fugir dos mínimos locais.
- Uma metaheurística, portanto, visa produzir um resultado satisfatório para um problema, porém sem qualquer garantia de optimalidade.

Algoritmos Meta-Heurísticos

- São algoritmos de IA, mas não de Aprendizado de Máquina. Contudo, podem e são usados com certa frequência em modelos de AM. Ex.: encontrar um bom conjunto de valores de parâmetros.
- Algoritmos básicos mais conhecidos:

Best-first Search; Greedy Search ou A*

Hill Climbing;

Tabu Search;

Simulated Annealing;

Ant Colony Optimization;

GRASP;

Memetic Algorithms;

Genetic Algorithms;

Particle Swarm Optimization;

Harmony Search.

Algoritmos Meta-Heurísticos

Inteligência Artificial destruindo no dinossauro da Google! (Rede Neural)

1 Principais desafios do Aprendizado de Máquina

Quantidade insuficiente de dados de treinamento

Dados de treinamento não representativos

Dados de baixa qualidade

Características irrelevantes

Sobreajuste dos dados de treinamento

Subajuste dos dados de treinamento

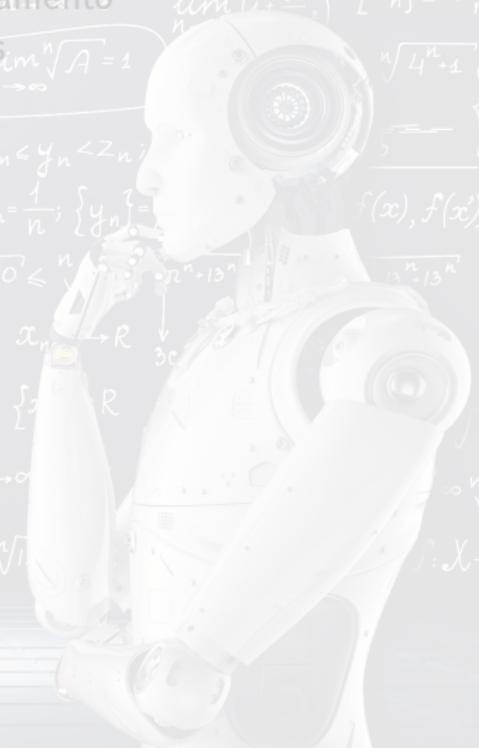
2 Teste e Validação

k-fold

Bootstrap

Teste e Validação

3 Algoritmos Meta-Heurísticos



$\lim_{n \rightarrow \infty} \left(\frac{n+1}{n} \right) \{x_n\} \subset R$
 $\{y_n\} \neq 0 \Leftrightarrow y_n \neq 0$
 $N \rightarrow R$ $x: p$
 $\sqrt{5^n} \left\{ \frac{1}{n} \right\} A_y$
 $x_c = \left\{ 1 + \frac{1}{n} \right\}$
 $f(x) \Leftrightarrow \exists q \in [0, 1] : \forall x, x' \in \mathcal{X}$
 $(x_n - g) < \varepsilon \quad n \geq n_0 : (x_n - g) < \varepsilon$
 $\left\{ \frac{1}{n} \right\} = \left\{ \frac{1}{n} \right\}$
 $\epsilon \in [0, 1] \quad \left\{ \frac{1}{n} \right\}$
 $\lim_{n \rightarrow \infty} \sigma^n \sigma^* = \lim_{n \rightarrow \infty} \sqrt{1 + e^{i\pi} + j\omega + 13} \{x_n\} \lim_{n \rightarrow \infty} \frac{x_n}{3}$
 $\forall n \in N, \text{ to } \frac{\{x_n\}}{\{y_n\}} = \frac{\{x_n\}}{\{y_n\}}$
 $n \geq n_0 : (x_n - g) < \varepsilon$
 $\lim_{n \rightarrow \infty} \sqrt[4]{4^n \cdot \cos 2\pi} \left(\frac{n^2 + n - 1}{n^2 - 2n + 3} \right)^5$
 $\forall n \in N \quad x_n < y_n < z_n$
 $\lim_{n \rightarrow \infty} \sqrt[n]{A} = 1$
 $\lim_{n \rightarrow \infty} \left(1 + \frac{\pi}{n} \right)$
 $\sqrt[4]{4^n + 1} \sum_{n=1}^{\infty} \{y_n\}$
 $f(x), f(x') \leq c$
FIM:
 $\{x_n\} : x_n = \frac{1}{n}; \{y_n\} =$
 $\{x_n\} : \sqrt[4]{D + O + O} \leq \sqrt[4]{13^n}$
 $x_n \rightarrow R$
 $\{x_n\} \cdot \{y_n\} = \{x_n + y_n\}; 13$
 $\{x_n\} \cdot \{y_n\} = \{x_n \cdot y_n\}; 13$
 $\downarrow n \rightarrow \infty$
 $\{x_n\} \quad ? \quad g \quad \sqrt[4]{13^n}$