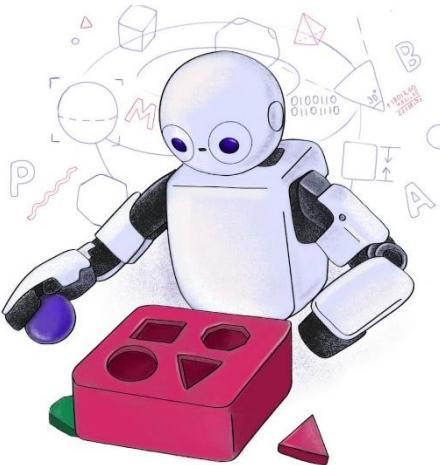


# T319 - Introdução ao Aprendizado de Máquina: *Regressão Linear (Parte V)*



**Inatel**

Felipe Augusto Pereira de Figueiredo  
[felipe.figueiredo@inatel.br](mailto:felipe.figueiredo@inatel.br)

# Recapitulando

- Vimos que o *escalonamento de atributos acelerara o aprendizado do GD* quando os atributos têm intervalos de variação muito diferentes.
- Aprendemos que *funções hipótese polinomiais* podem ser utilizadas para *aproximar comportamentos não-lineares, além de lineares*.
- Porém, precisamos *encontrar o grau ideal da função hipótese polinomial*.
  - Polinômios com *grau muito baixo* podem não ter flexibilidade o suficiente para aproximar os dados, causando *subajuste*.
  - Polinômios com *grau muito alto* podem ser tão flexíveis que acabam memorizando os dados de treinamento, causando *sobreajuste*.
- Na sequência, veremos como *escolher o grau ideal* da *função hipótese polinomial* de *forma quantitativa*, mesmo não conhecendo ou existindo uma função objetivo por trás da geração dos dados coletados.

# Validação cruzada

- A validação cruzada é uma técnica utilizada para *avaliar quantitativamente o desempenho* de um *modelo* e *garantir que ele generalize bem para dados inéditos*, evitando assim problemas de subajuste ou sobreajuste.
- O processo de validação cruzada envolve *dividir o conjunto total de dados em subconjuntos* e realizar *uma ou mais rodadas de treinamento e validação* do modelo com *uma ou mais combinações desses subconjuntos*.
- A validação cruzada é uma ferramenta importante para *comparar e selecionar modelos* e para *ajustar hiperparâmetros* como, por exemplo, o *passo de aprendizagem*, o *grau do polinômio* da função hipótese, etc.

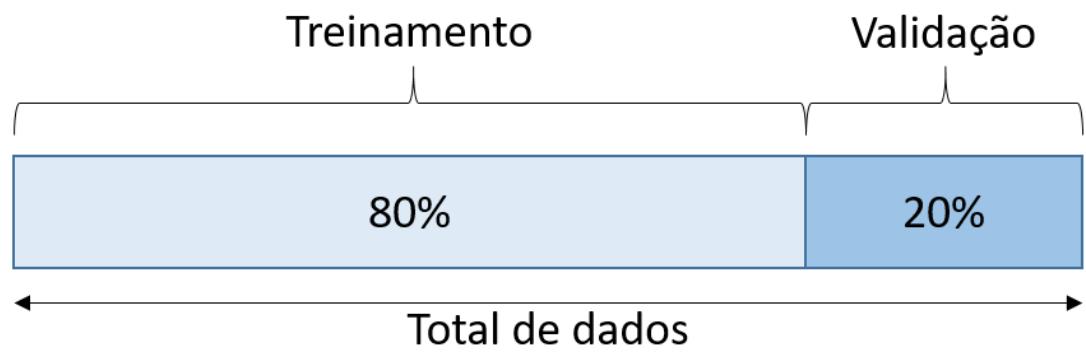
# Validação cruzada

- O **objetivo** da **validação cruzada** é encontrar um **ponto de equilíbrio** entre a **flexibilidade** e a **capacidade de generalização** do modelo (e.g., polinômio).
- Um **modelo equilibrado** é
  - Flexível o suficiente para se ajustar ao **comportamento geral** dos dados.
  - Capaz de prever saídas próximas às esperadas para exemplos não vistos durante seu treinamento.
- A **flexibilidade** de um modelo é **estimada** através do **erro de treinamento** e a **capacidade de generalização** é **estimada** através do **erro de validação ou teste**.
  - Erro de treinamento é calculado com o conjunto usado para treinar o modelo.
  - Erro de validação ou teste é calculado com dados inéditos.

# Validação cruzada

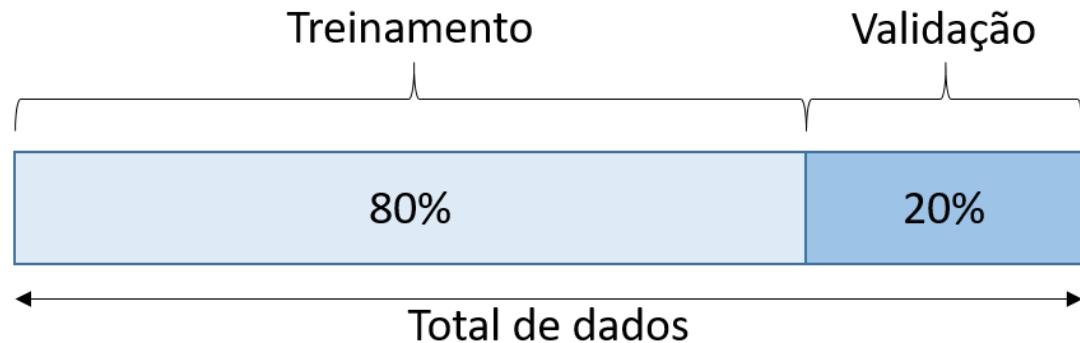
- No caso onde queremos usar a *validação cruzada* para *encontrar o grau ideal da função hipótese polinomial*, o *comportamento destes dois erros* vai nos ajudar a verificar *quais graus fazem o modelo se ajustar demais ou insuficientemente* aos dados de treinamento.
- As estratégias de validação cruzada mais utilizadas e que veremos a seguir são:
  - *Holdout*
  - *k-fold*

# Holdout



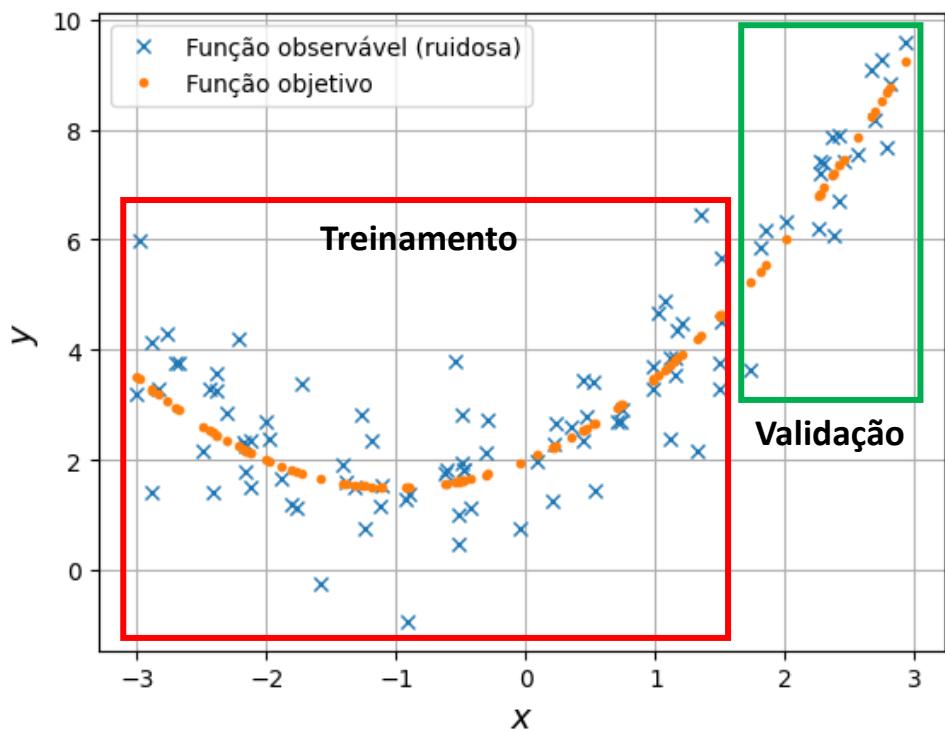
- É a estratégia de validação cruzada *mais simples e rápida*, pois ela **divide** o conjunto total de dados em **apenas dois subconjuntos**, um para **treinamento** e outro para **validação** (ou teste) do modelo.
- Consequentemente, realiza-se **apenas um treinamento e uma validação do modelo**.

# Holdout



- Em geral, mas é opcional, o conjunto total de dados é ***embaralhado de forma aleatória antes da divisão.***
- Normalmente, divide-se o conjunto total de dados em 70 a 80% para treinamento e 30 a 20% para validação.

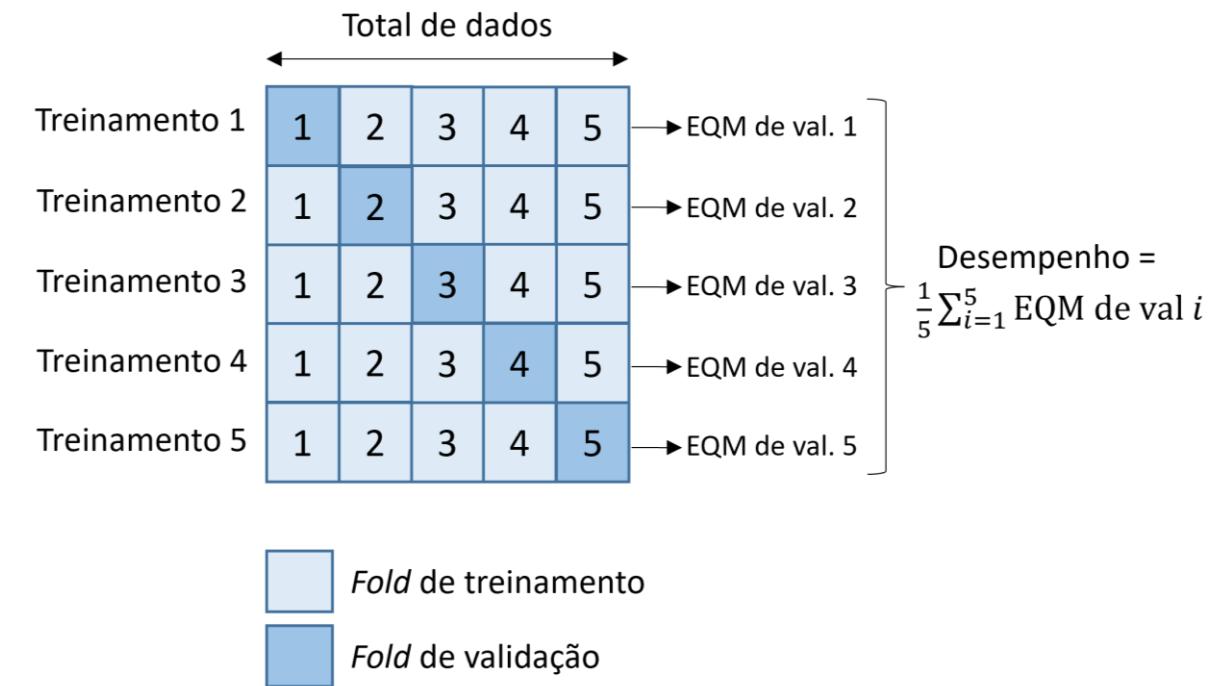
# Holdout



- Entretanto, o modelo treinado e validado com esta estratégia pode **apresentar desempenho ruim se a divisão dos dados não for representativa do padrão presente nos dados.**
  - Problema conhecido como **viés de seleção**.
- O desempenho do modelo pode variar muito dependendo da divisão dos dados.
- Além disso, a **divisão única** pode **não fornecer uma estimativa robusta do desempenho** do modelo.
- Em geral, usa-se o *holdout* quando o **conjunto de dados é muito grande**, o que **minimiza estes problemas**.

# $k$ -fold

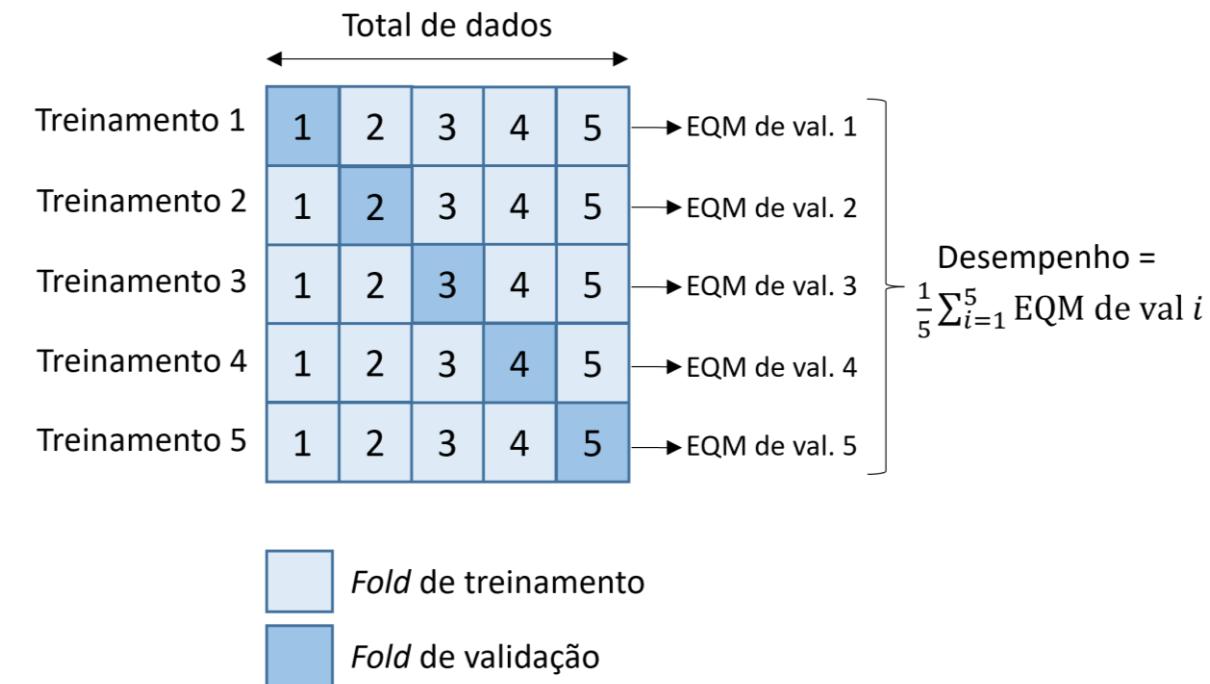
$k = 5$



- É uma estratégia mais elaborada do que a do *holdout*.
- A estratégia consiste em embaralhar (opcional) e **dividir o conjunto total de dados em  $k$  partes** (ou *folds*) **iguais**.
- O **modelo é treinado  $k$  vezes**, cada vez usando  **$k-1$**  partes como conjunto de treinamento e a parte restante como conjunto de validação.
- O **EQM com cada conjunto de validação** é calculado **ao final de cada treinamento**.

# $k$ -fold

$k = 5$

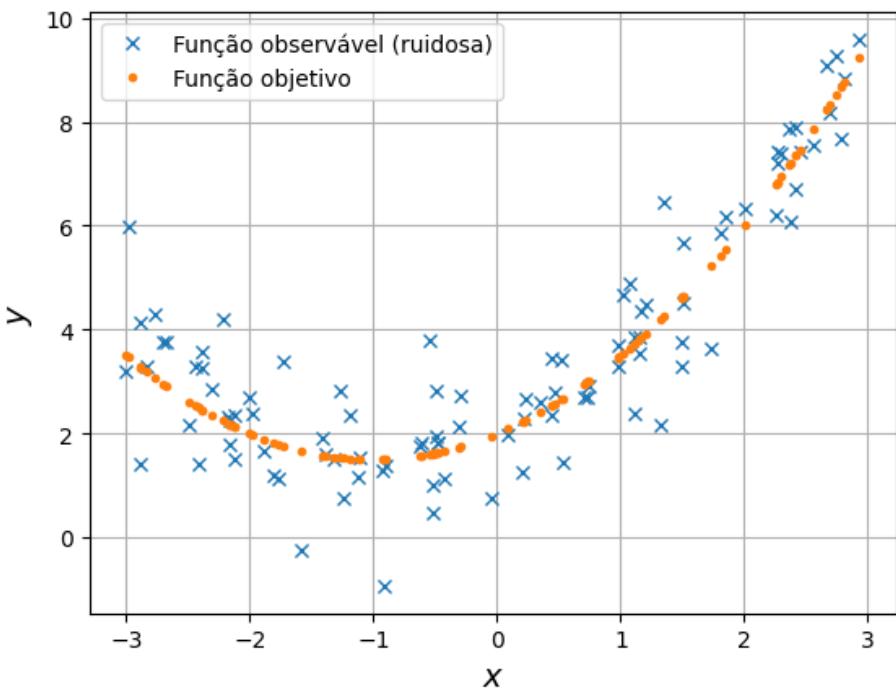


- Ao final dos  $k$  treinamentos, calcula-se a **média** e o **desvio padrão** dos  $k$  EQMs de validação para fornecer uma **avaliação geral do desempenho do modelo**.
- Em geral, utiliza-se  $k = 5$  ou  $10$ .
- Porém,  $k$  deve ser escolhido de forma que os *folds* sejam **representativos do padrão presente nos dados**.

# *k*-fold

- O *k-fold* é a estratégia de validação cruzada mais usada por fornecer ***indicações mais claras*** sobre desempenho do modelo, devido à média tomada.
- Essa avaliação ***minimiza os possíveis efeitos provocados pelo viés de seleção***, pois o modelo é treinado e validado *k* vezes, cada vez com uma divisão diferente dos dados.
  - Isso faz com que a ***avaliação do modelo se torne menos sensível à divisão dos dados***.
- Entretanto, em relação ao *holdout*, o *k-fold* tem um ***tempo de validação maior (cerca de k vezes)***, pois deve-se realizar *k* treinamentos e validações, enquanto que com o *holdout*, realiza-se apenas um treinamento e validação.

# Validação cruzada para encontrar o grau do polinômio aproximador



- Para exemplificar o uso das estratégias de validação cruzada para encontrar o grau ideal do polinômio aproximador, vamos usar a seguinte **função observável**

$$y_{noisy} = y + w,$$

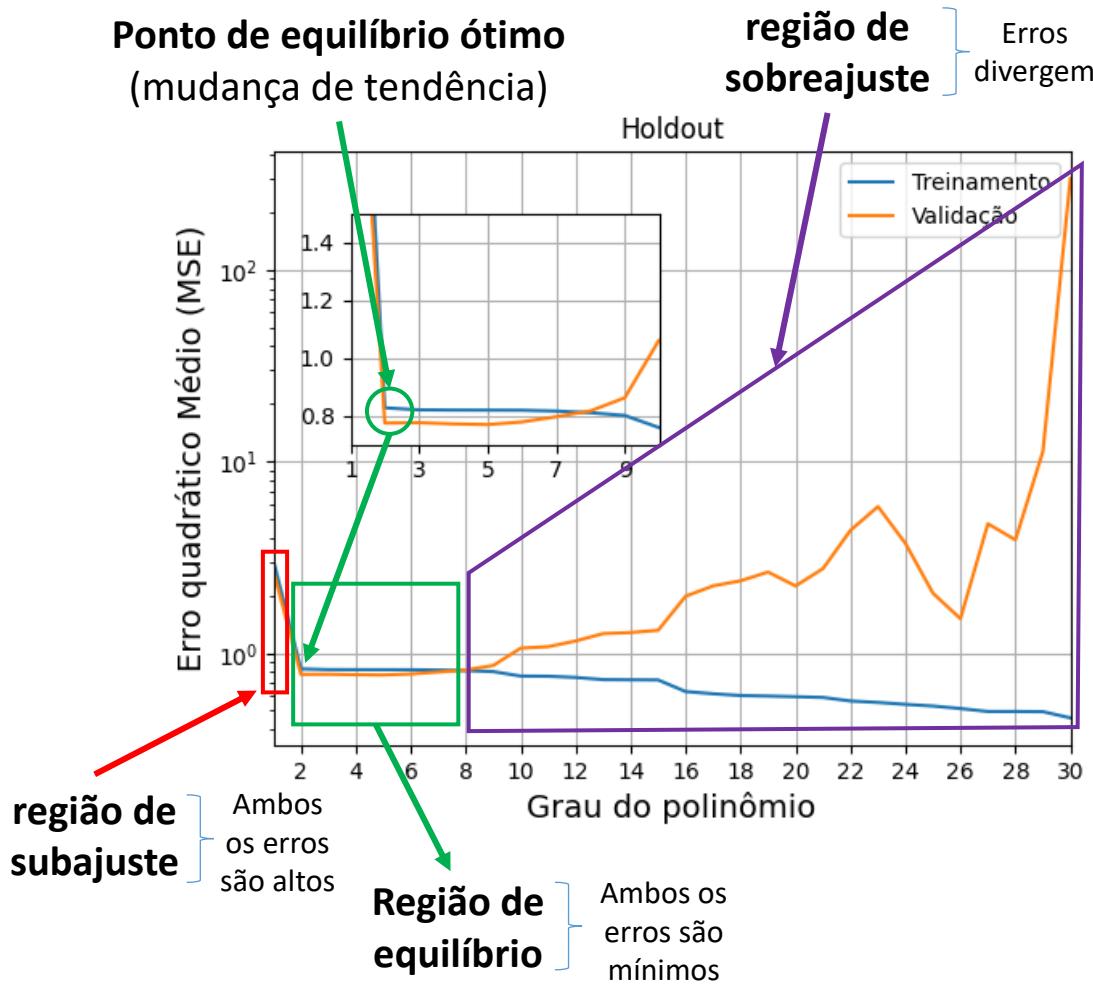
onde  $y$  é a função objetivo e  $w$  é o ruído, o qual tem amostras retiradas de uma distribuição Gaussiana com média zero e variância unitária.

- A **função objetivo** é um polinômio de segundo grau definido como

$$y = 2 + x + 0.5x^2,$$

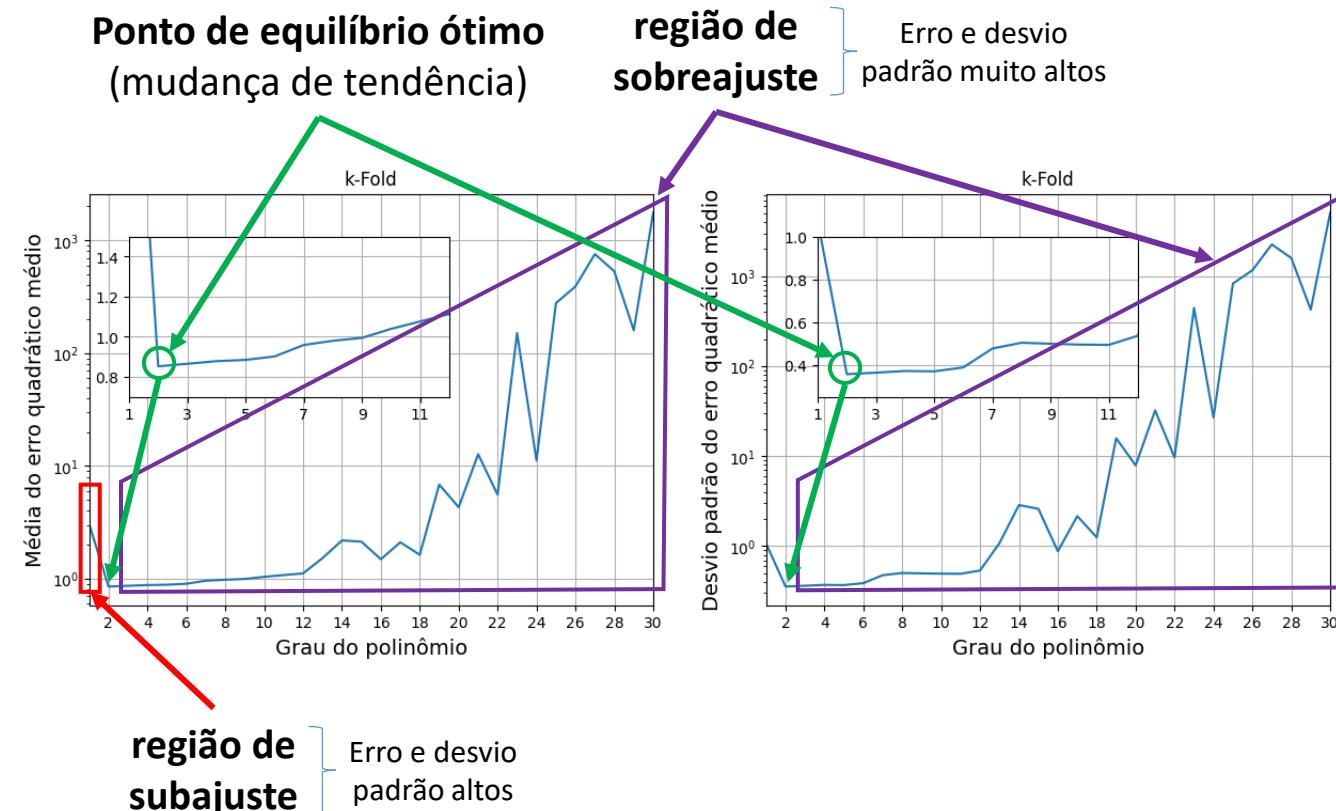
onde  $x$  é o atributo, o qual varia entre -3 a 3.

# Usando holdout para encontrar o grau do polinômio aproximador



- Divisão: 70% para o conjunto de treinamento e 30% para o conjunto de validação.
- Tempo médio para validação cruzada *holdout* com  $N = 100$  é de  $\approx 150$  [ms].
- Erro de treinamento **diminui** conforme o grau do polinômio aumenta.
- Erro de validação **aumenta** conforme o grau do polinômio aumenta.
- Qual grau escolher?
  - Valor para o qual **ambos os erros sejam mínimos** (balanço entre flexibilidade e capacidade de generalização) e que tenha **menor complexidade computacional**.

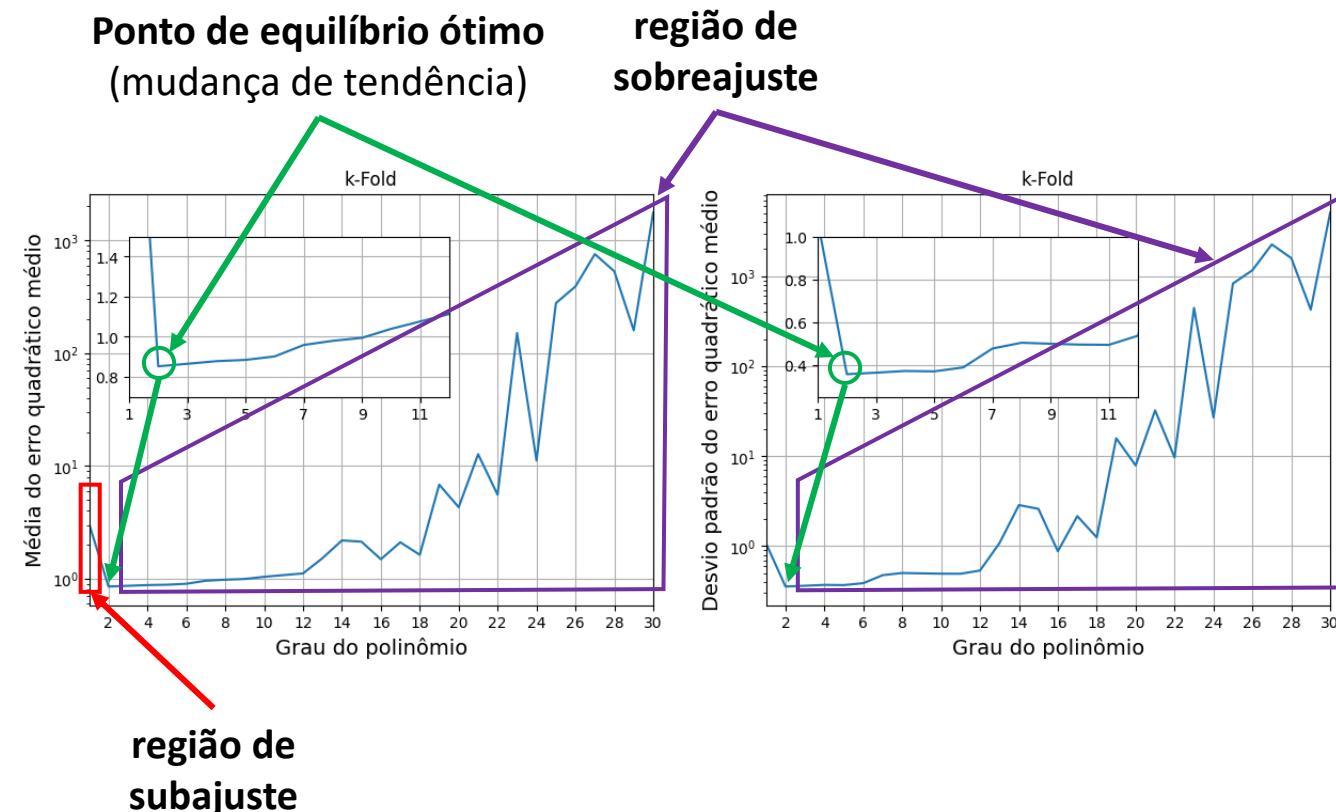
# Usando $k$ -fold para encontrar o grau do polinômio aproximador



- $k = 10 \text{ folds}$ : 10 treinamentos com 9 *folds* para treinamento e 1 para validação.
- Os gráficos mostram a média e o desvio padrão do EQM de validação para as 10 etapas de treinamento para cada grau avaliado.
- A média e o desvio padrão do EQM diminuem, passando pelo ponto de equilíbrio, e depois aumentam com o grau do polinômio.
- Qual grau escolher?
  - Valor onde *ambos, média e desvio padrão do EQM, sejam mínimos* e que tenha *menor complexidade computacional*.

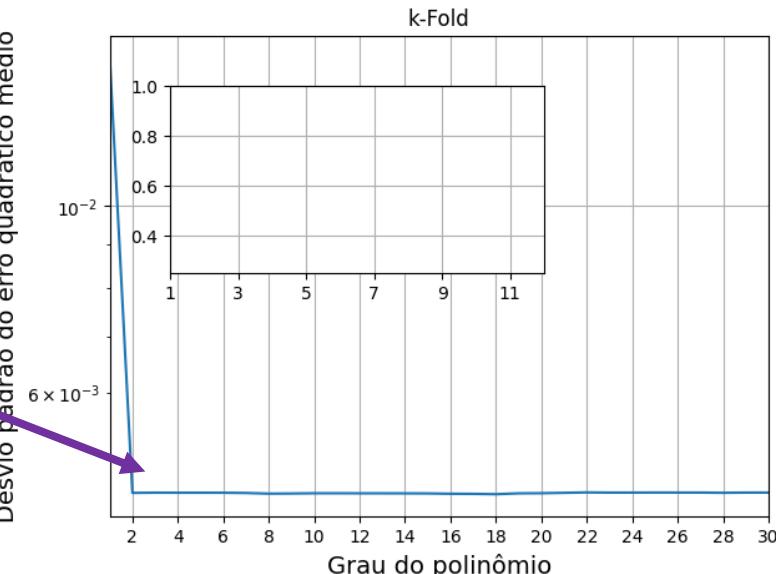
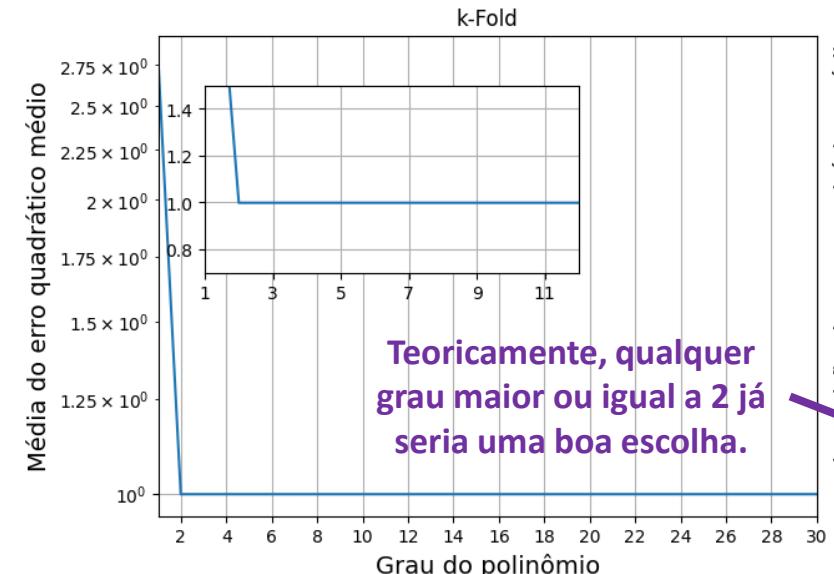
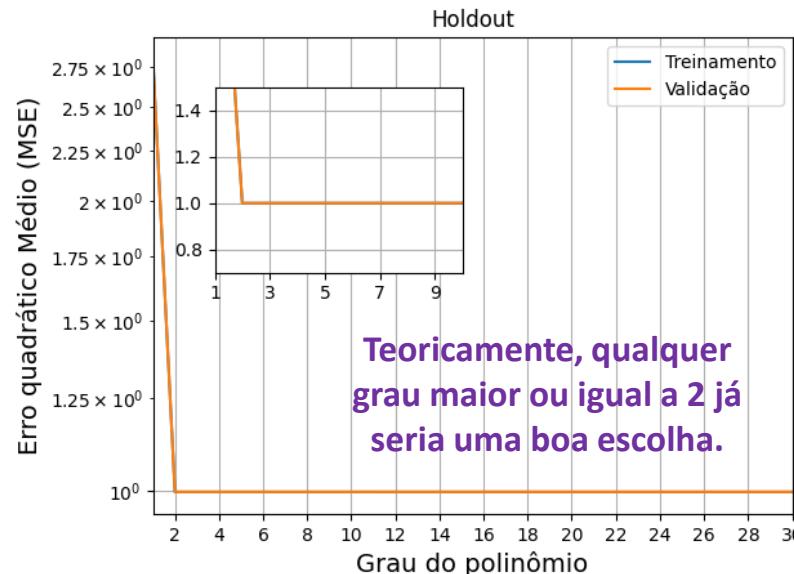
\* Tempo médio para validação cruzada  $k$ -fold com  $N = 100$  e  $k = 10$  exemplos é de  $\approx 1.5$  [s].

# Usando $k$ -fold para encontrar o grau do polinômio aproximador



- Conforme o modelo se **sobreajusta** aos dados de treinamento, o **desvio padrão do erro de validação aumenta** devido à **redução de sua capacidade de generalização**.
  - Ou seja, o modelo aprendido se distancia muito do comportamento geral dos dados.
- Modelos **muito flexíveis** (mais do que o necessário) apresentam desvios padrão do erro de treinamento muito baixo e do erro de validação muito alto, indicando **sobreajuste**.
- Modelos pouco flexíveis (menos do que o necessário) têm ambos os desvios padrão dos erros altos, indicando **subajuste**.

# Qual grau escolher quando vários são possíveis?



- Observem as figuras.
- *Qual grau devemos escolher quando os erros (holdout) ou a média e o desvio padrão dos erros (k-fold) são mínimos e praticamente constantes para vários graus de polinômio?*
  - Isso ocorre quando o número de exemplos é muito maior do que a flexibilidade (i.e., grau) dos modelos testados.

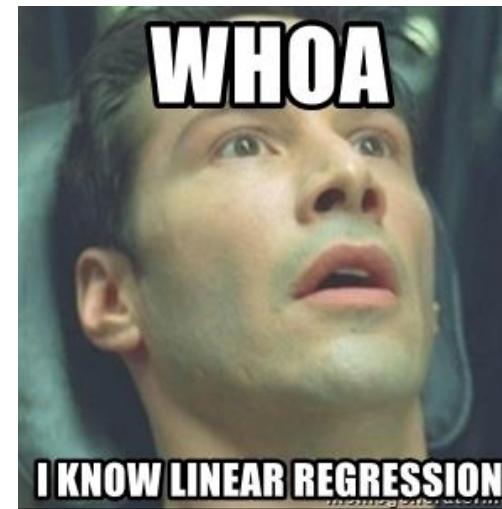
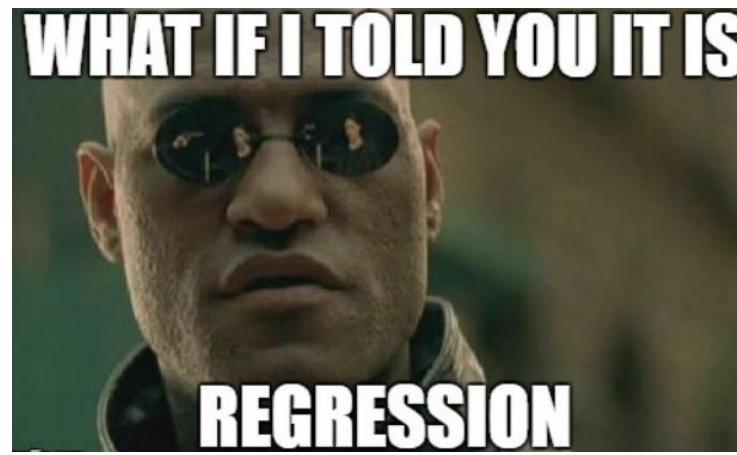
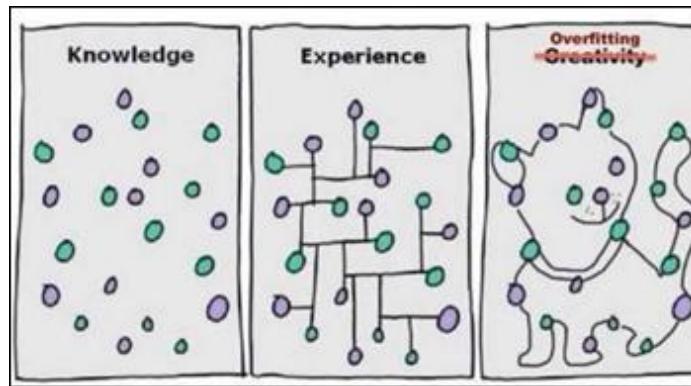
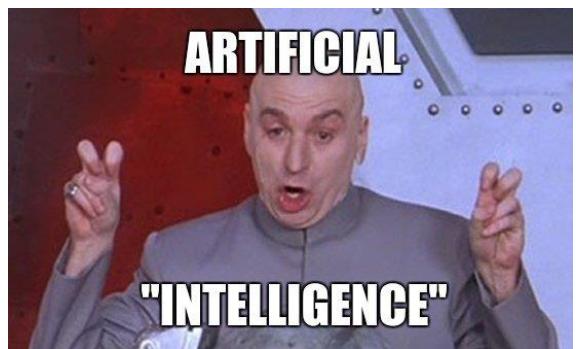
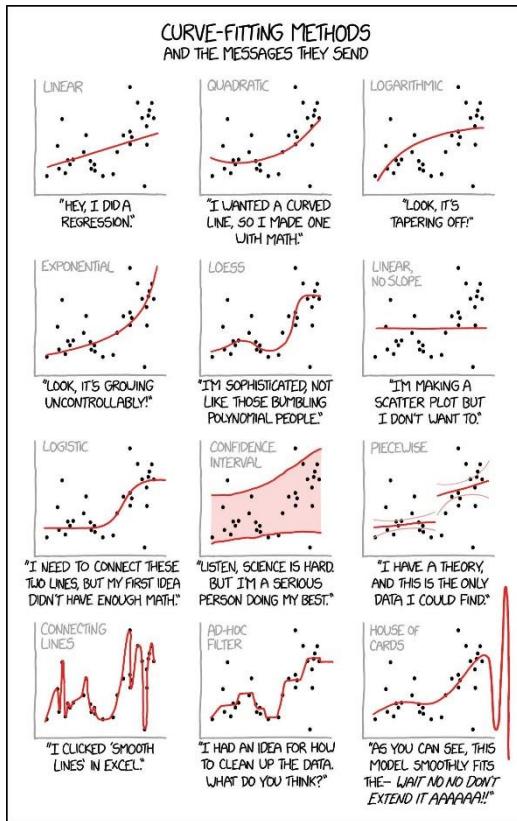
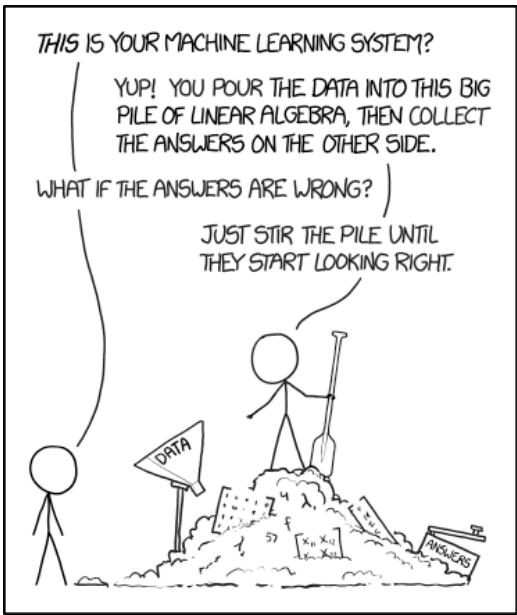
# Qual grau escolher quando vários são possíveis?

- A resposta é aplicar a *navalha de Occam*.
- A *navalha de Occam* é um princípio lógico que sugere que, *entre várias explicações igualmente plausíveis* para um conjunto de observações, a *mais simples deve ser preferida*.
  - Ou seja, deve-se *preferir explicações mais simples às mais complexas*.
- Portanto, usando a *navalha de Occam* escolhemos a *função hipótese polinomial com menor grau* (i.e., menos complexa), *mas que se ajusta bem ao comportamento geral dos dados*.
  - Ou seja, escolhemos o modelo mais simples em termos de quantidade de cálculos, mas que possua uma boa capacidade de generalização.

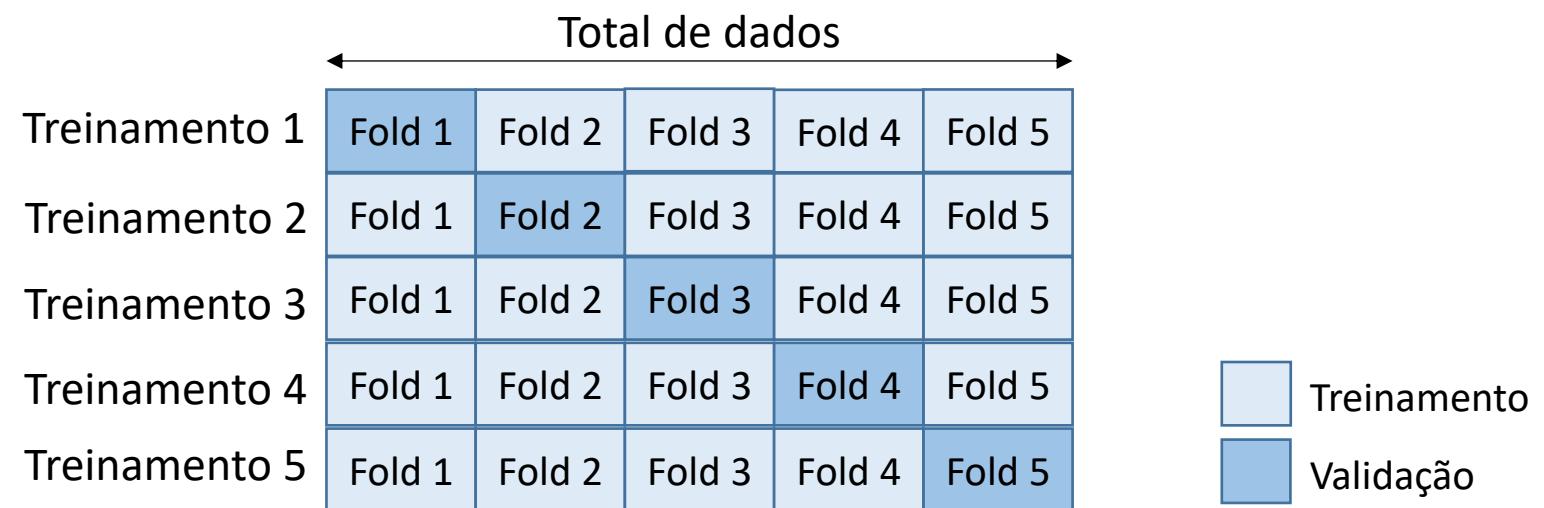
# Tarefas

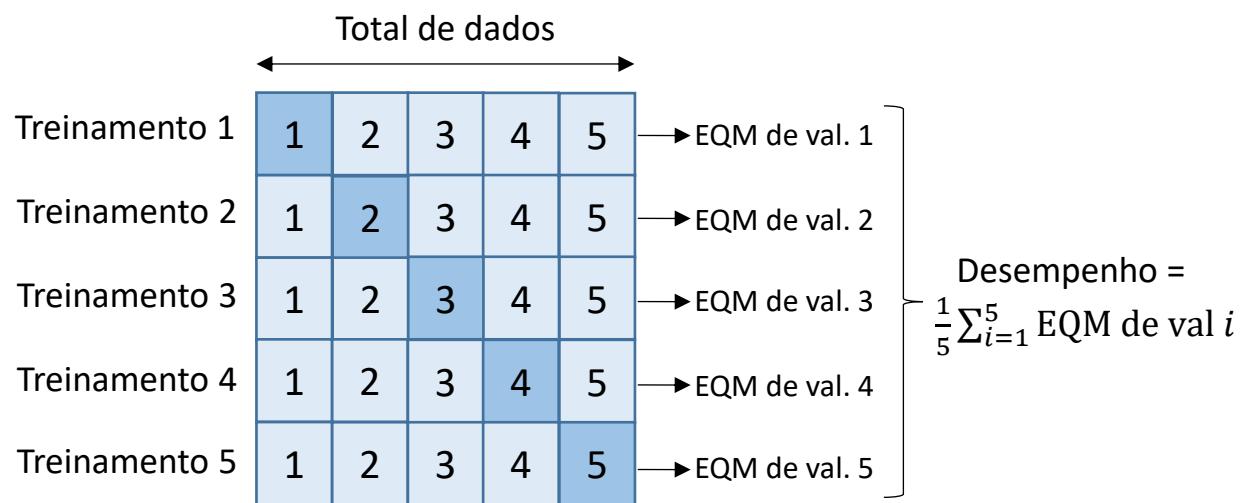
- **Quiz:** “*T319 - Quiz - Regressão: Parte V*” que se encontra no MS Teams.
- **Projeto Final**
  - Projeto pode ser feito em grupos de no máximo 3 alunos.
  - **Entrega: 28/11/2025 até às 23:59.**
  - Leiam os enunciados do trabalho atentamente.

Obrigado!



# FIGURAS





*Fold de treinamento*



*Fold de validação*

Total de dados

	Total de dados									
Treinamento 1	1	2	3	4	5	6	7	8	9	10
Treinamento 2	1	2	3	4	5	6	7	8	9	10
Treinamento 3	1	2	3	4	5	6	7	8	9	10
Treinamento 4	1	2	3	4	5	6	7	8	9	10
Treinamento 5	1	2	3	4	5	6	7	8	9	10
Treinamento 6	1	2	3	4	5	6	7	8	9	10
Treinamento 7	1	2	3	4	5	6	7	8	9	10
Treinamento 8	1	2	3	4	5	6	7	8	9	10
Treinamento 9	1	2	3	4	5	6	7	8	9	10
Treinamento 10	1	2	3	4	5	6	7	8	9	10



*Fold de treinamento*



*Fold de validação*

