

# Cross-Validation

ESTAT0016 – Tópicos Especiais em Estatística (Introdução à Aprendizagem de Máquina)

Prof. Dr. Sadraque E.F. Lucena

# Introdução

- Um dos pontos principais para o sucesso de uma modelagem de dados é a capacidade de avaliar adequadamente a performance dos modelos desenvolvidos.
- Uma métrica muito usada para isso é o *desempenho preditivo*, ou seja, a capacidade de um modelo fazer previsões precisas sobre dados não vistos anteriormente.
- Até agora vimos o *método holdout*, que consiste em particionar os dados em um conjunto de treino e outro de teste.

**Passo 1**

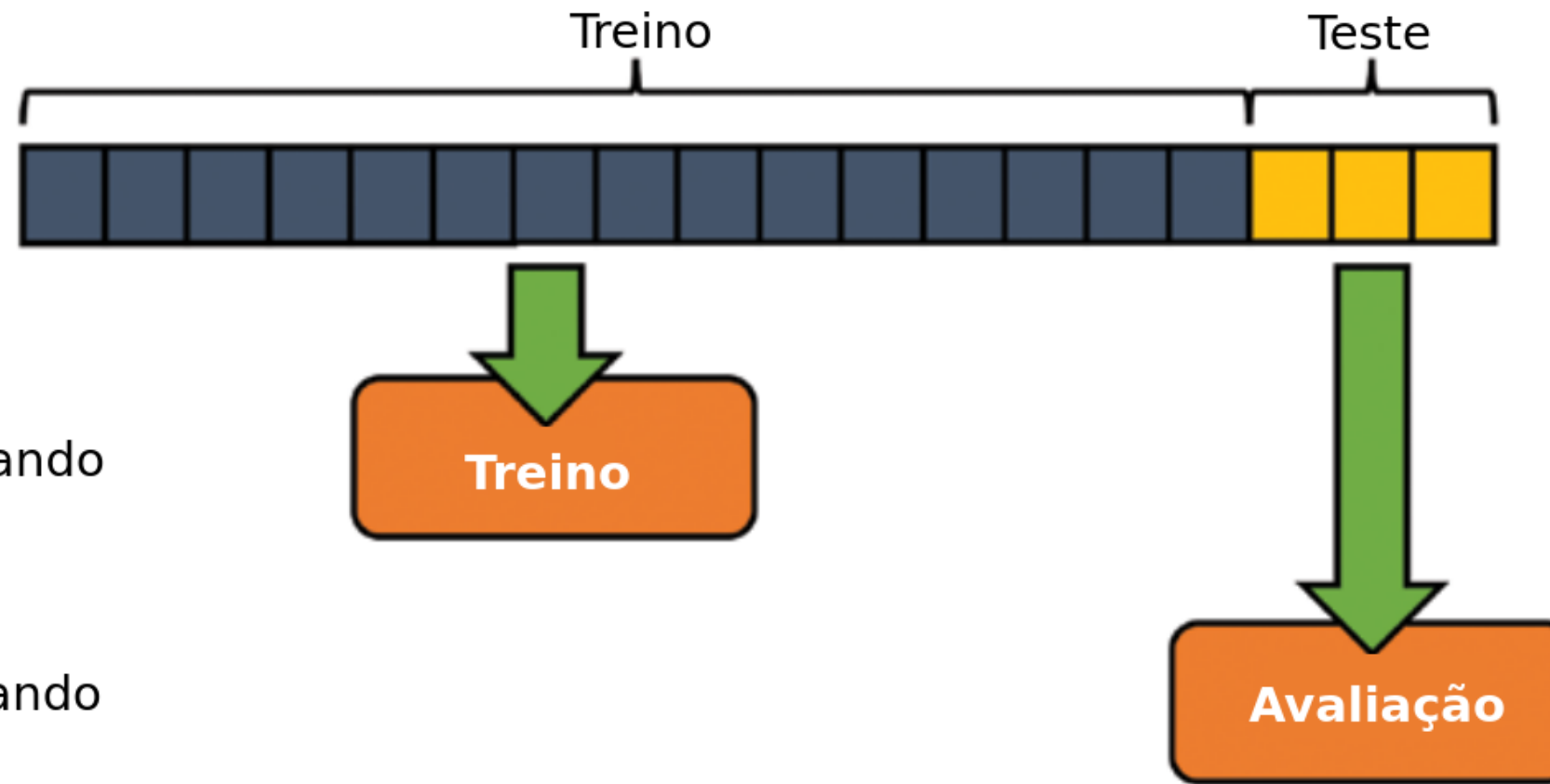
Dividir os dados em treino e teste.

**Passo 2**

Treinar o modelo usando dados de treino.

**Passo 3**

Avaliar o modelo usando dados de teste.



# Introdução

- Além dos dados de treino e teste, é comum ser usado um conjunto de *dados de validação*.
- Esses dados são usados no processo de construção do modelo com objetivo de refiná-lo antes de usá-lo nos dados de teste.

## Passo 1

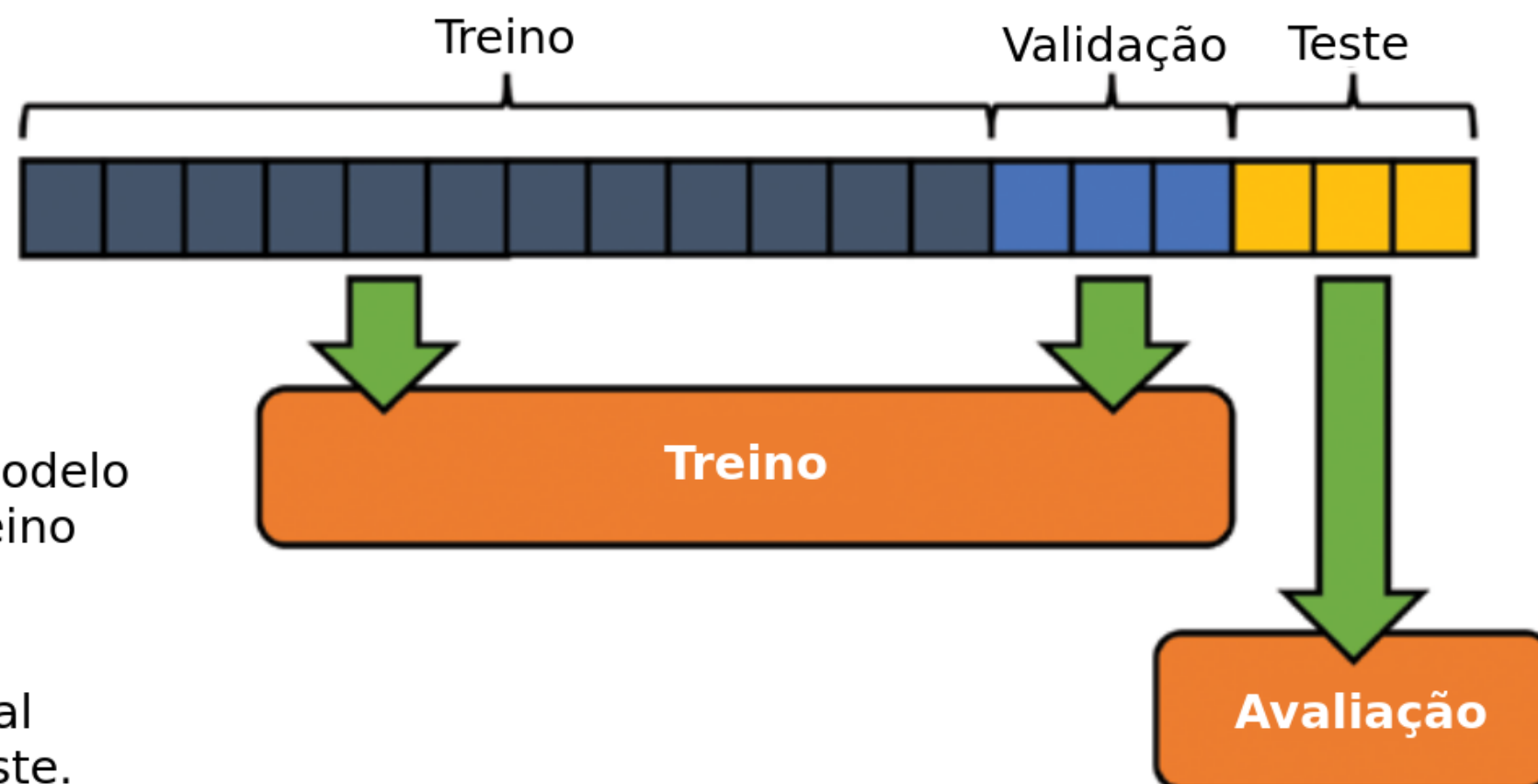
Dividir os dados em treino, validação e teste.

## Passo 2

Treinar e refinar o modelo usando dados de treino e validação.

## Passo 3

Avaliar o modelo final usando dados de teste.



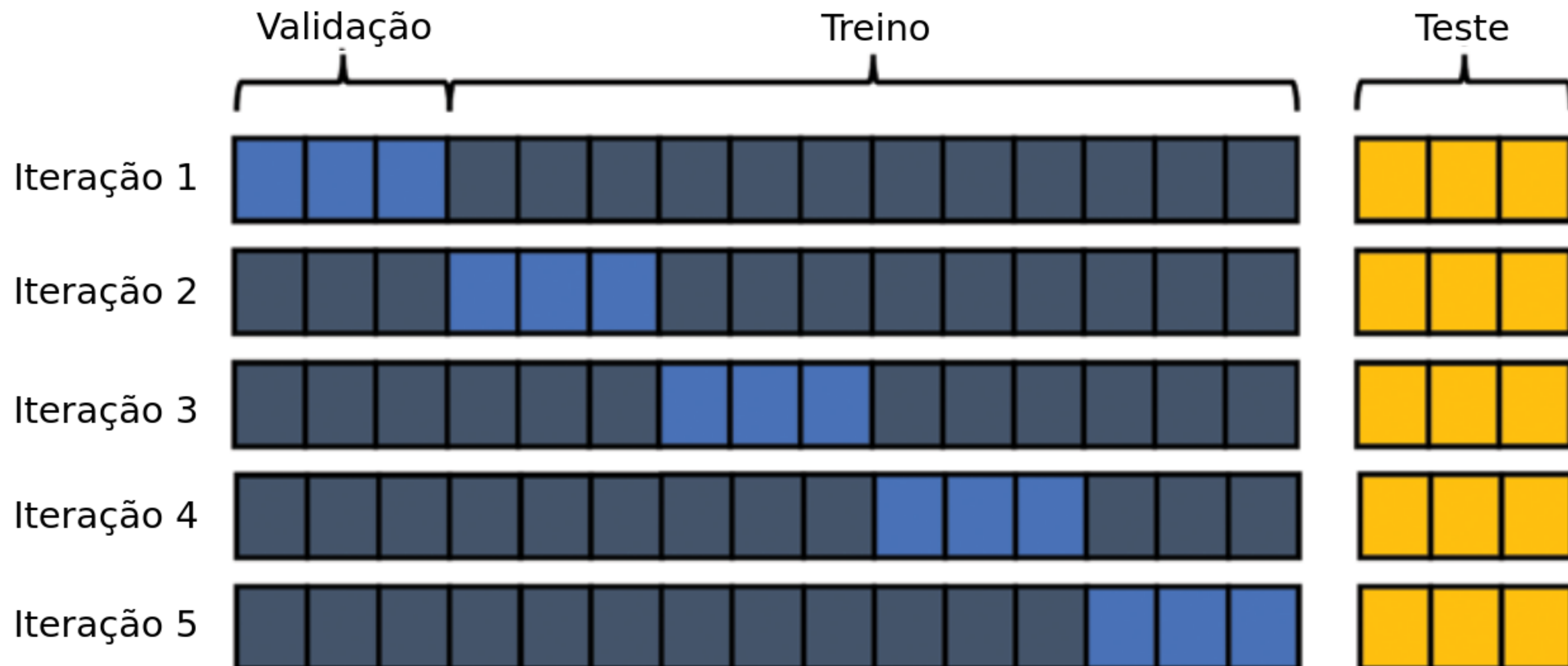
# Validação Cruzada (*Cross-Validation*)

- O problema dessa abordagem é que quando não temos grandes quantidades de dados, as partições podem não ser representativas dos dados originais.
- Para resolver esse problema do *método holdout*, usamos técnicas de validação cruzada.
- Ela consiste em usar diferentes amostras dos dados originais para treinar e validar o modelo.
- As abordagens mais comuns são:
  - *k-Fold Cross-Validation*
  - *Leave-One-Out Cross-Validation*
  - *Random Cross-Validation*

# k-Fold Cross-Validation

- Esta é a técnica de *cross-validation* mais usada.
- Ela consiste em:
  1. Dividir o conjunto de dados em  $k$  subconjuntos (ou *folds*) aproximadamente iguais.
  2. A cada rodada:
    - 2.1. Reter um dos  $k$  subconjuntos como conjunto de teste.
    - 2.2. Treinar o modelo com os subconjuntos restantes.
    - 2.3. Avaliar o desempenho nos dados retidos para teste.
  3. Repetir o passo 2 retendo um subconjunto por vez.
  4. Calcular a média das métricas de desempenho obtidas para obter uma estimativa mais robusta do desempenho do modelo.
- Em geral usa-se  $k$  entre 5 e 10.

# k-Fold Cross-Validation



# Leave-One-Out Cross-Validation

- Esta é outra técnica de *cross-validation* muito usada, sendo uma variação do *k-fold cross-validation*.
- Ela consiste em:
  1. A cada rodada separe uma única observação para teste e todas as demais para treino.
  2. Ajuste o modelo com os dados de treino e teste na observação deixada de fora.
  3. Repita o passo 2 até que todas as observações sejam usadas como teste uma vez.
  4. Calcule a média das métricas de desempenho utilizadas.



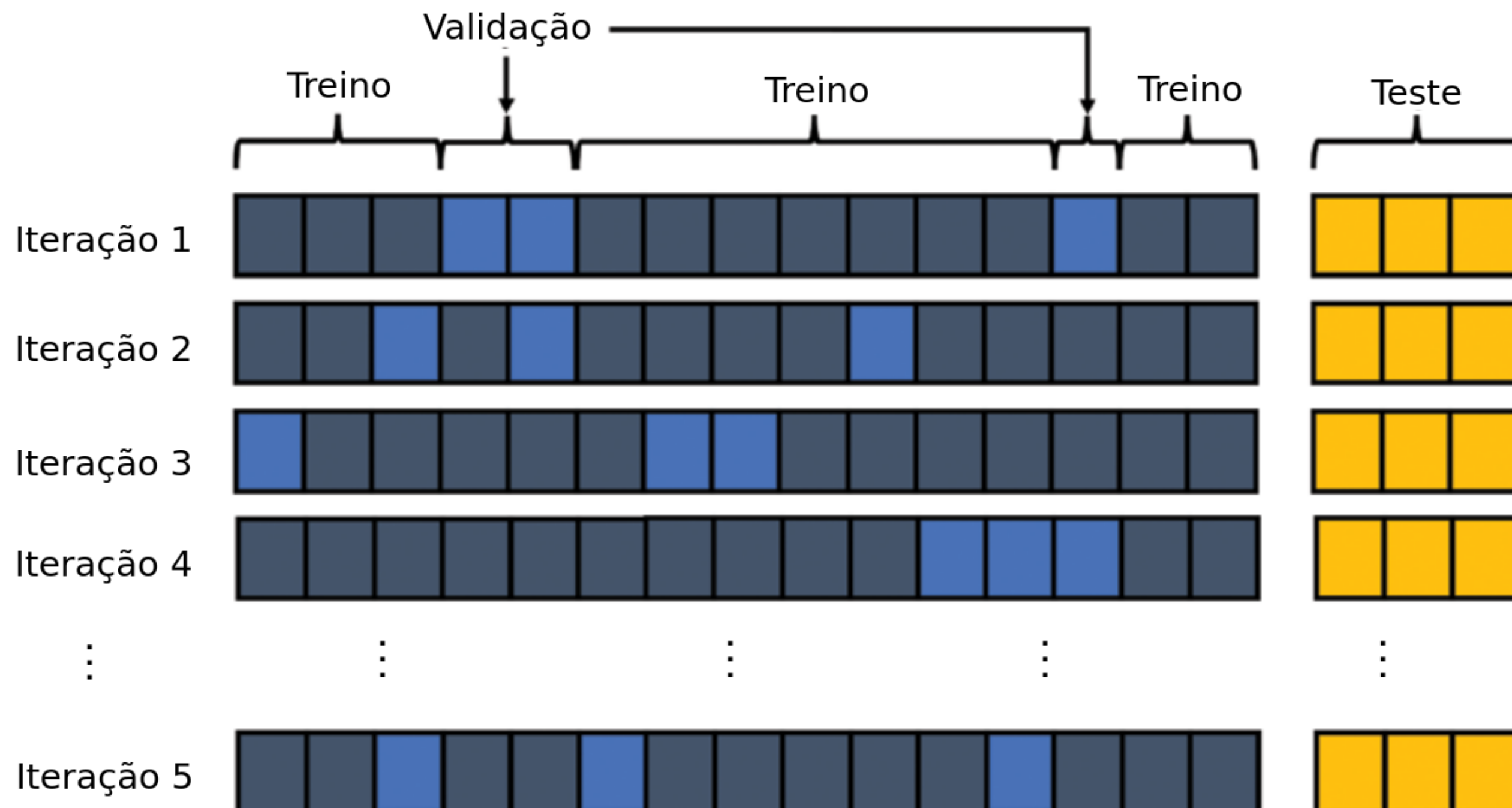
# Leave-One-Out Cross-Validation



# Random Cross-Validation

- O método *Random cross-validation* ou *Monte Carlo cross-validation* é similar ao método *k-fold cross-validation*.
- A diferença é que, ao invés de criar os  $k$  subconjuntos no início do processo, os dados de teste são selecionados aleatoriamente sem reposição durante cada iteração.

# Random Cross-Validation



# FIM

