ICC204 - Aprendizagem de Máquina e Mineração de Dados

Avaliação de Modelos

Prof. Rafael Giusti
rgiusti@icomp.ufam.edu.br

Motivação

- Considere o seguinte experimento
  - Problema: queremos classificar uma coleção de emails como ***spam*** ou ***ham***
  - Dados: conjunto de emails rotulados por usuários da nossa plataforma
  - Abordagem: treinamos um classificador SVM para o seguinte conceito
    - Qual a relação entre o conteúdo e o remetente de um email e ele ser ou não *spam*?

Motivação

- Uma vez que o modelo esteja treinado, iremos colocá-lo em prática
  - Emails classificados como *spam* serão enviados automaticamente para a caixa de *spam*
  - Os outros emails serão enviados para a caixa de entrada
- Que problemas podem surgir dessa abordagem?

Motivação

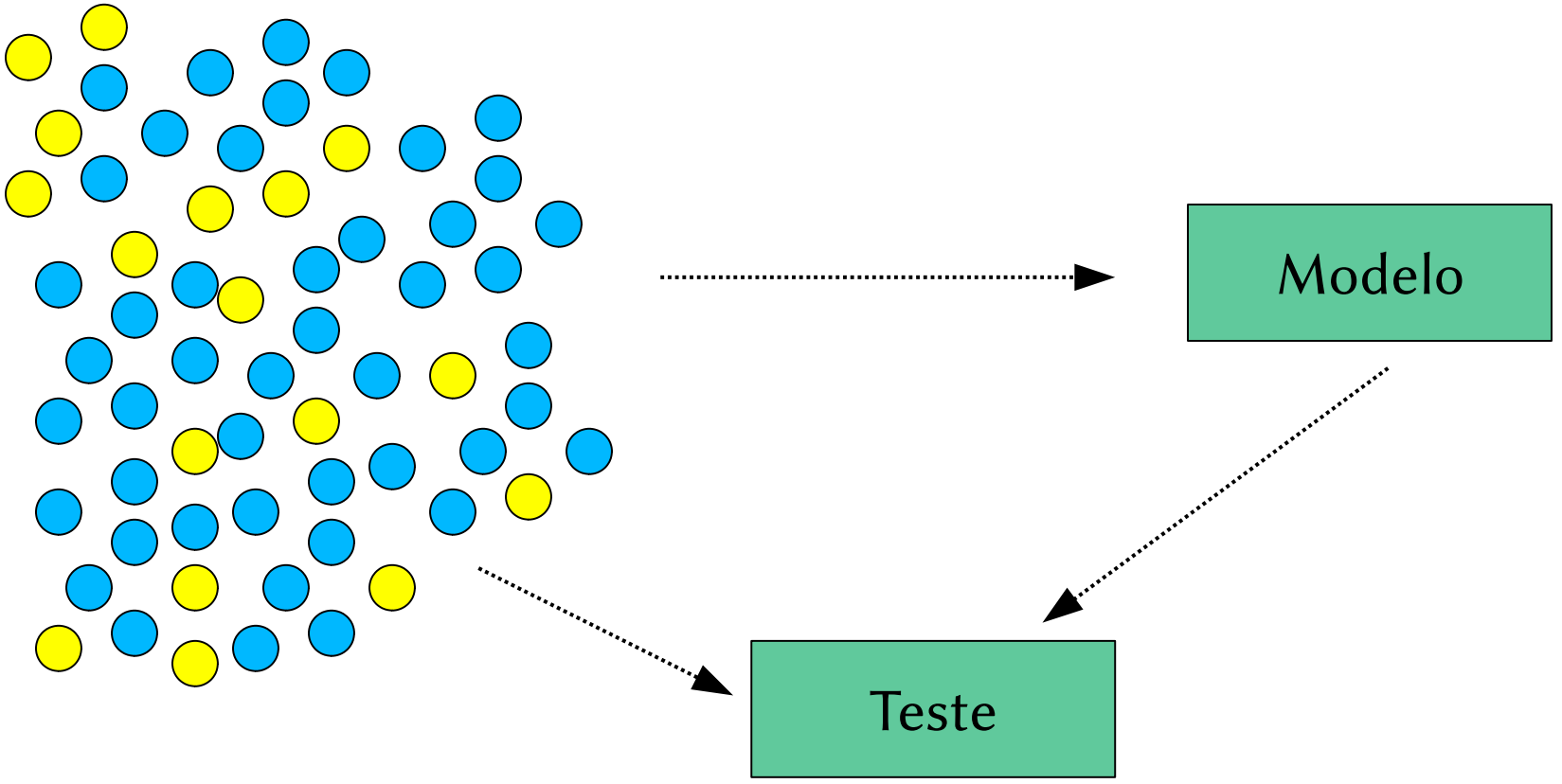
- Antes de colocar o detector de *spam* em produção, fazemos um teste
  - Separamos 70% dos nossos dados para treinamento e 30% para teste
  - Avaliamos o erro cometido pelo detector no conjunto de teste
  - Observamos que o erro é de 0,0001%
  - Podemos colocar o detector de *spam* em prática?

Estimador

- Queremos **estimar** o erro que o modelo cometerá
  - Com base na nossa amostra, o **estimador** é uma estatística do erro
  - Queremos um **estimador não enviesado**, isto é, um estimador cujo valor para a amostra se aproxime da população

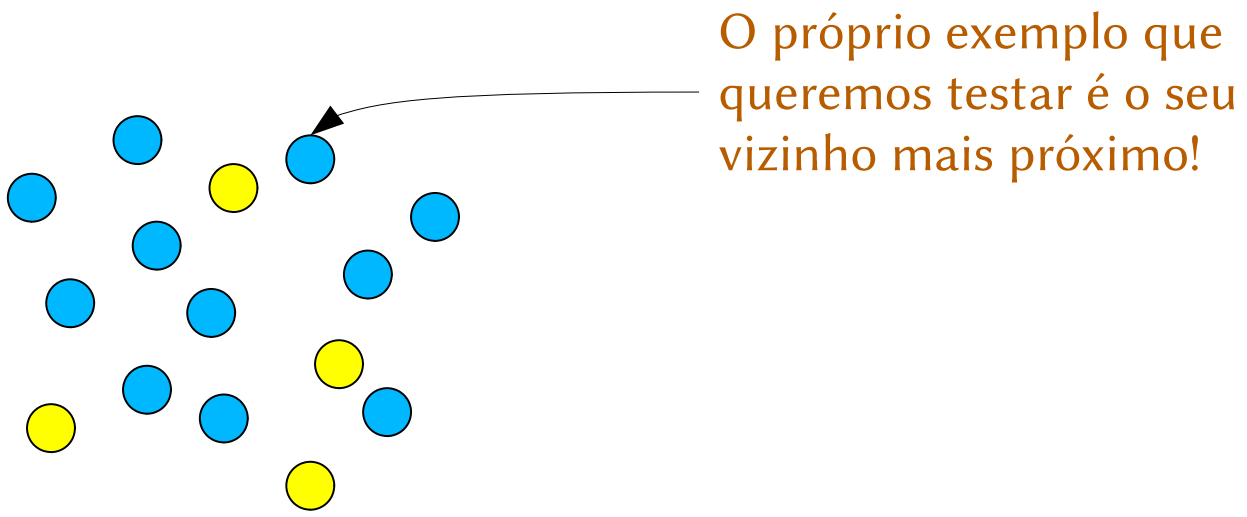
Estimador

- Um simples estimador é o erro empírico
  - Use as amostras de treino para testar o modelo



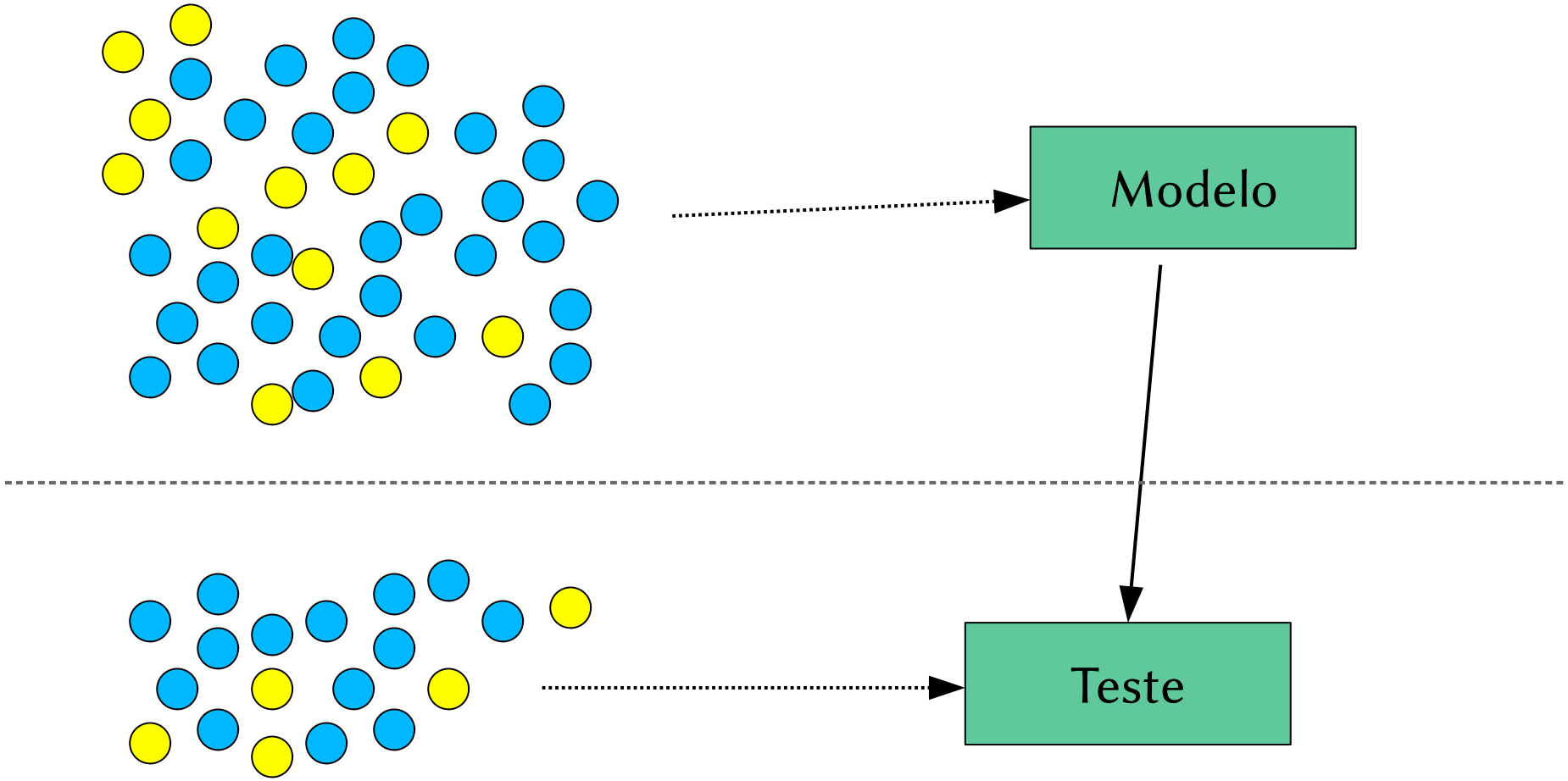
Estimador

- O erro empírico é um estimador muito enviesado
- Para o k-NN, dependendo do valor de *k*, é possível que o erro empírico seja sempre zero



Holdout

- Separe uma porção do conjunto de trenamento para testes

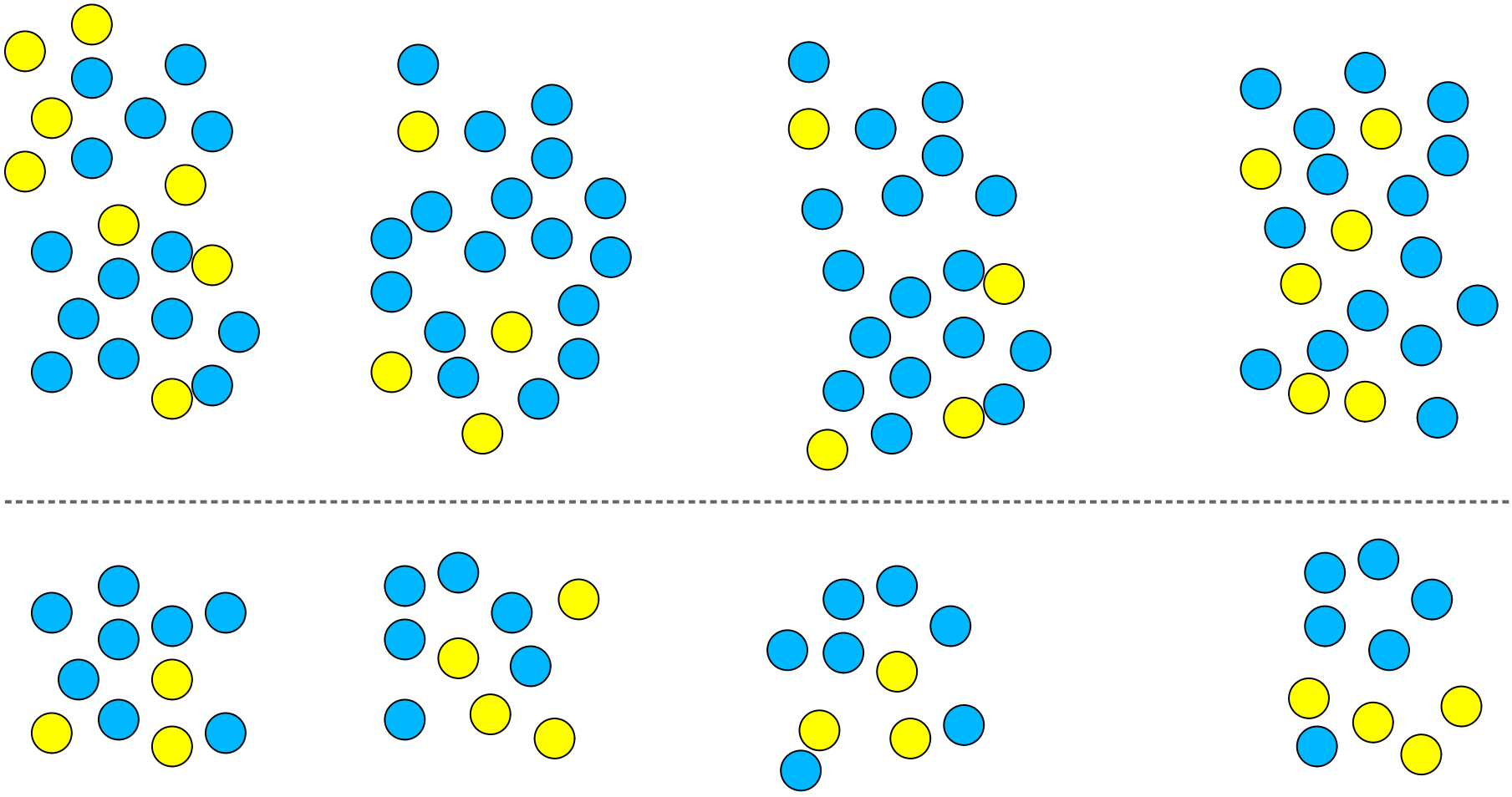


Holdout

- Produz estimadores menos enviesados que o erro empírico
  - Entretanto, o *holdout* ainda é substancialmente enviesado
    - A estimativa do *holdout* depende de uma única amostra retirada da nossa amostra
    - Com conjuntos rotulados muito grandes, esse viés pode ser reduzido

Holdout repetido

- Uma abordagem um pouco melhor que o *holdout* é realizar diversas rodadas do *holdout*



Holdout repetido

- No *holdout* repetido, o conjunto é **reamostrado** n vezes em **partições** de treinamento e teste
  - Em cada particionamento, estimamos um valor de erro e_i
  - O erro do modelo é estimado como a média desses erros individuais

Holdout repetido

- Pontos positivos
  - Menor viés que o *holdout*
- Pontos negativos
  - Existe sobreposições entre as diferentes partições
  - Muitos exemplos podem nunca ser utilizados para testar o modelo

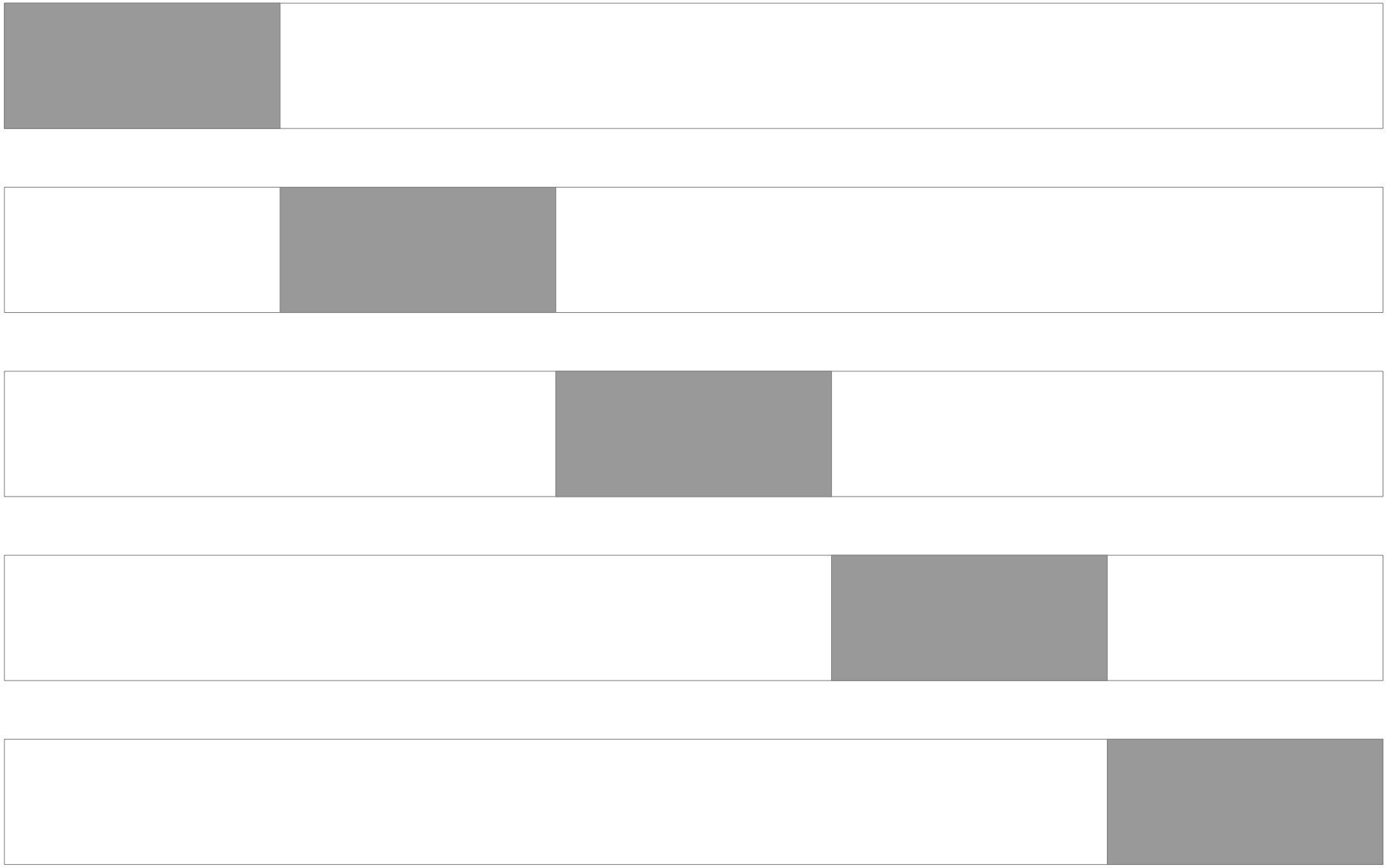
Validação cruzada

- Procedimento de reamostagem no qual todos os exemplos rotulados são utilizados uma única vez para teste
- Não existe sobreposição nos conjuntos de teste

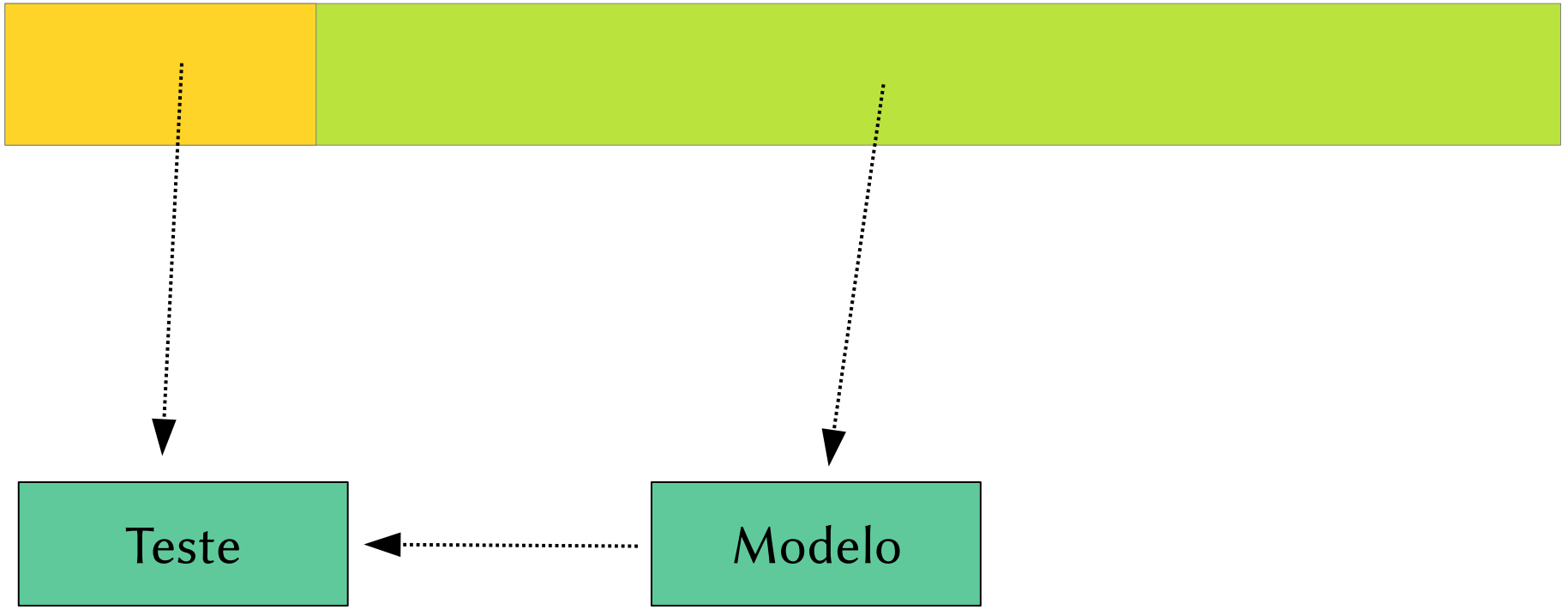
Validação cruzada

- Primeiro passo
  - Dividir as amostras em k subconjuntos de tamanhos idênticos ou aproximadamente idênticos
- Segundo passo
  - Treinar k modelos utilizando cada conjunto (*fold*) para teste e os demais conjuntos para treinamento

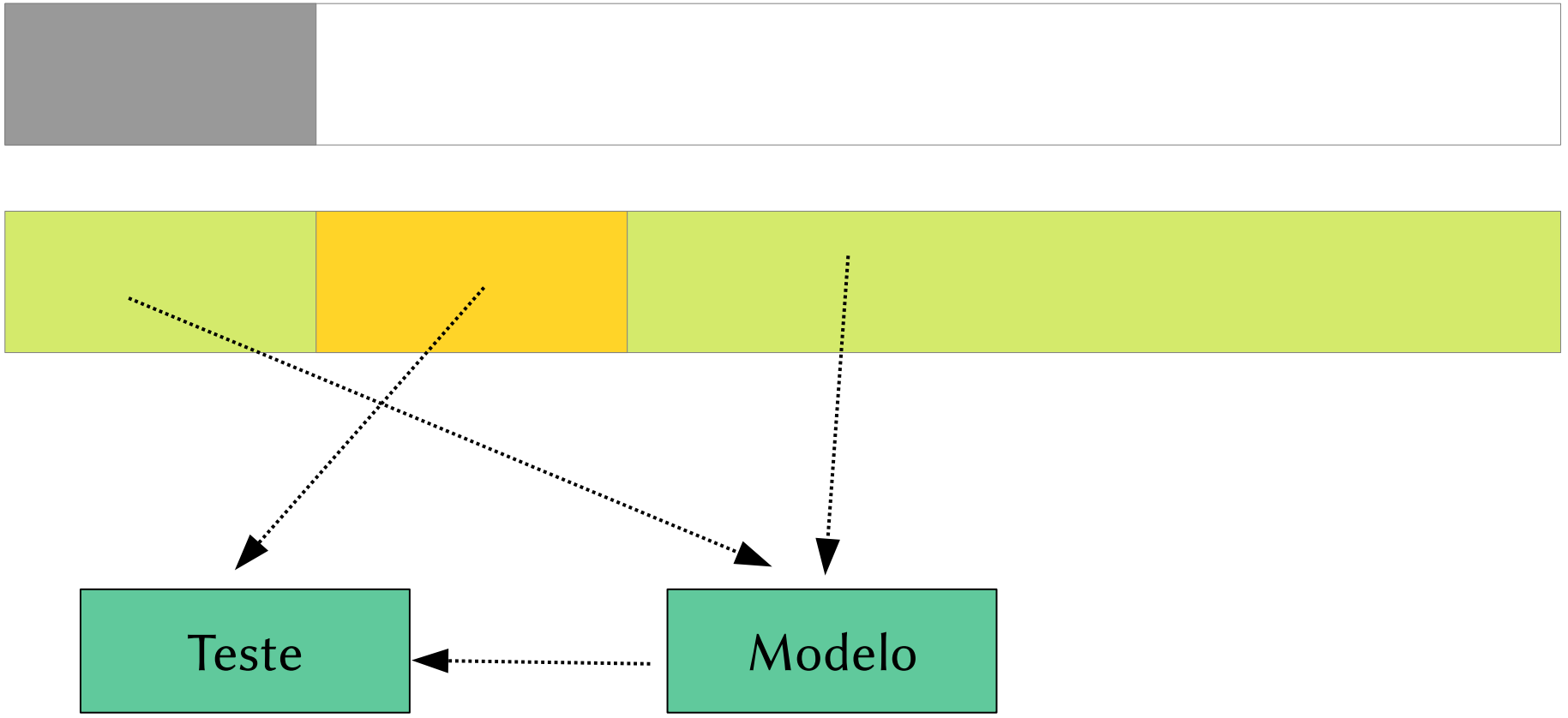
Validação cruzada



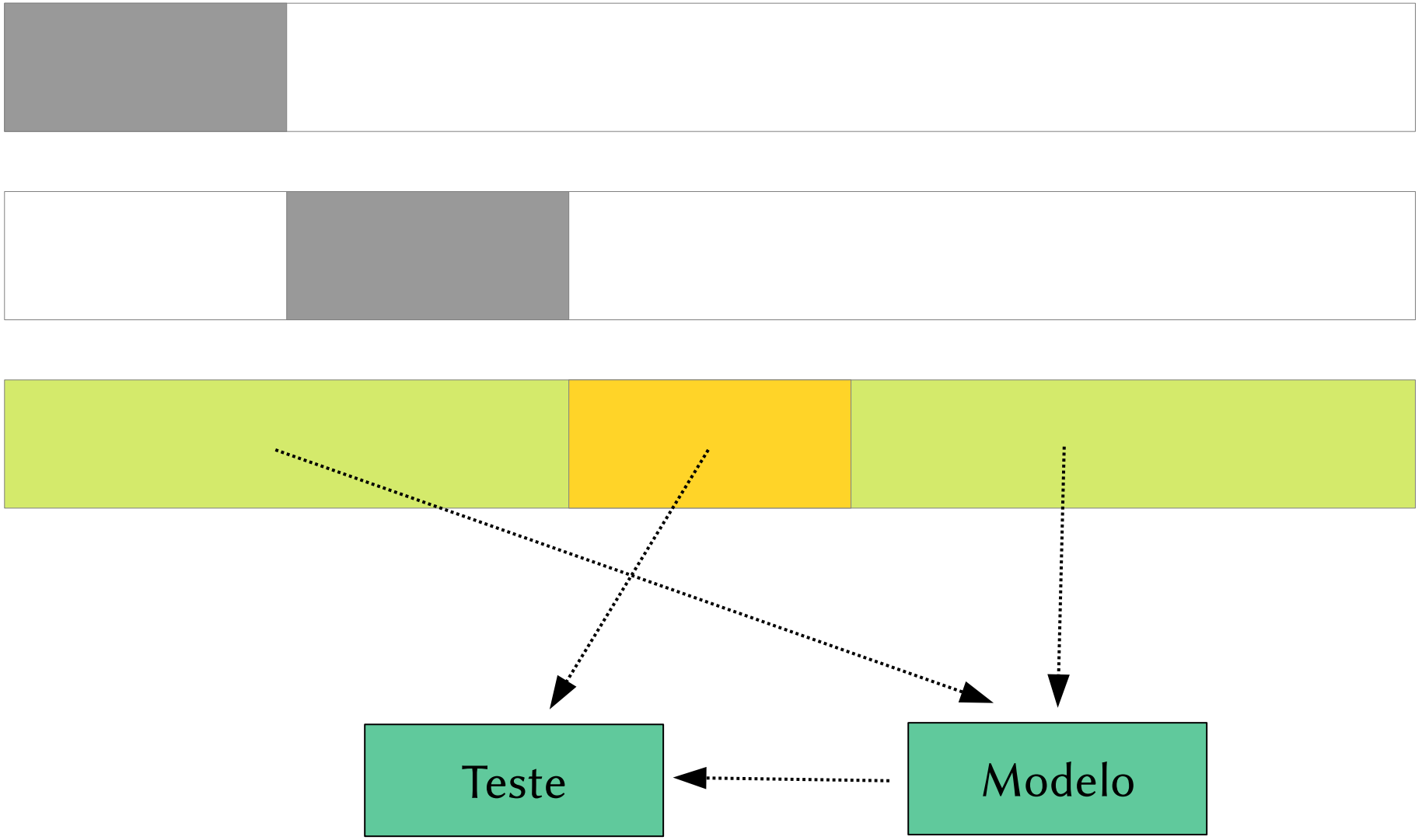
Validação cruzada



Validação cruzada



Validação cruzada

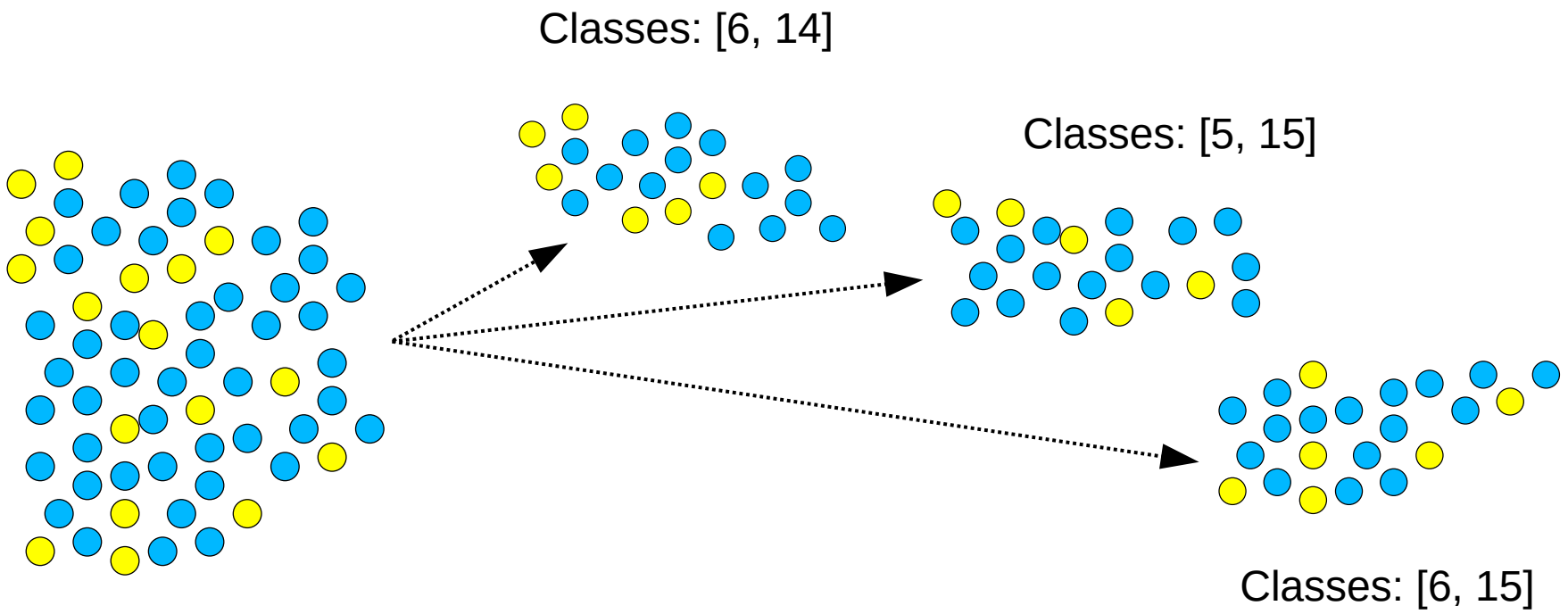


Validação cruzada

- A validação cruzada é normalmente **estratificada**
  - Estratificar significa dividir as amostras em grupos homogêneos antes de reamostrar
    - Exemplo: entre classes
  - Cada subconjunto é reamostrado respeitando as proporções dos estratos
    - Visando, por exemplo, obter *folds* que contenham aproximadamente a distribuição de classes da amostra original

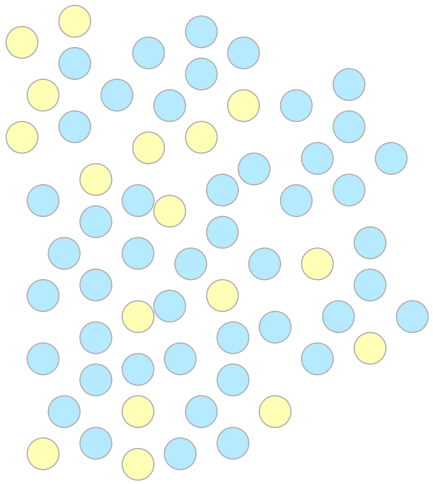
Validação cruzada

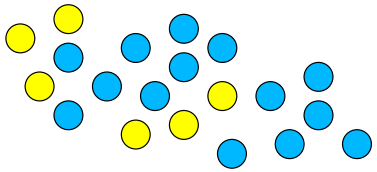
- 3-fold estratificado



Validação cruzada

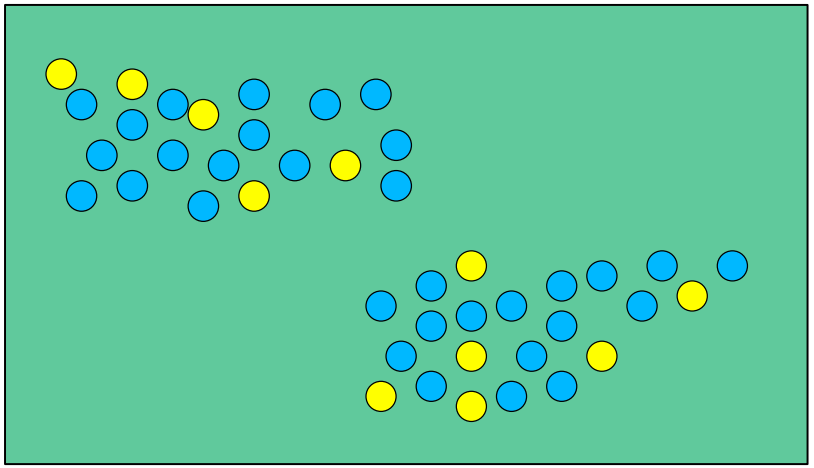
- 3-fold estratificado





Teste

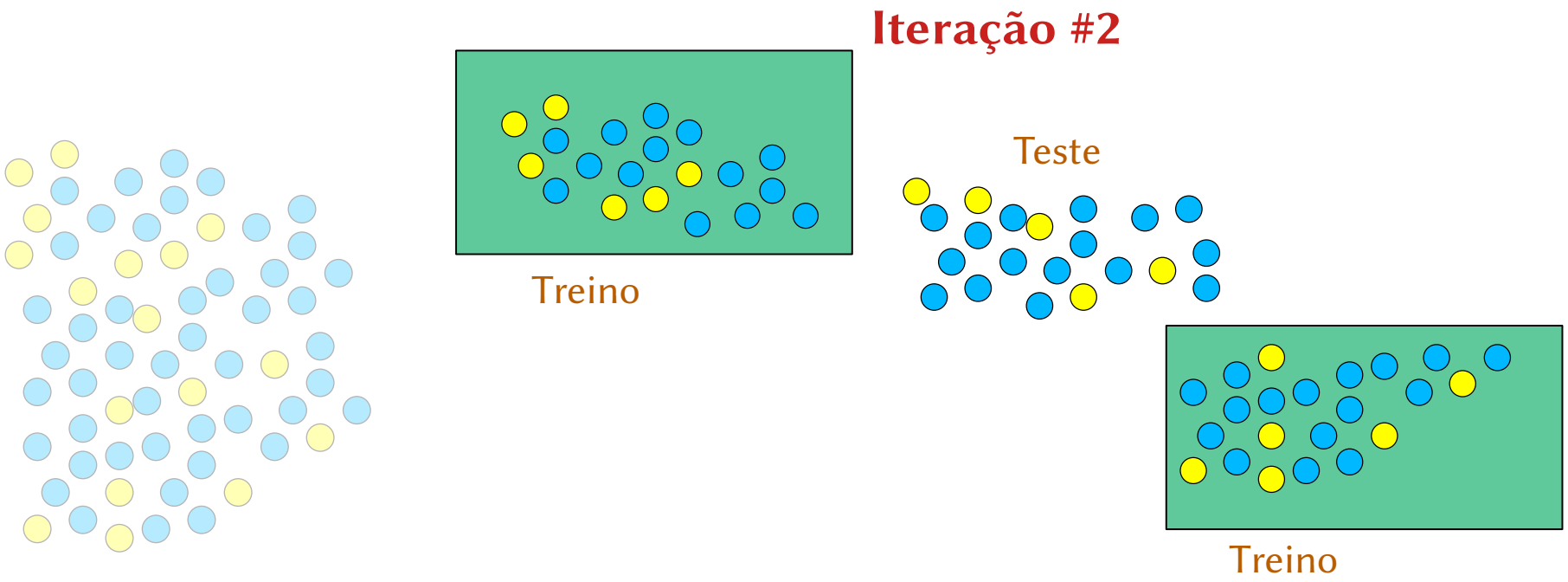
Iteração #1



Treino

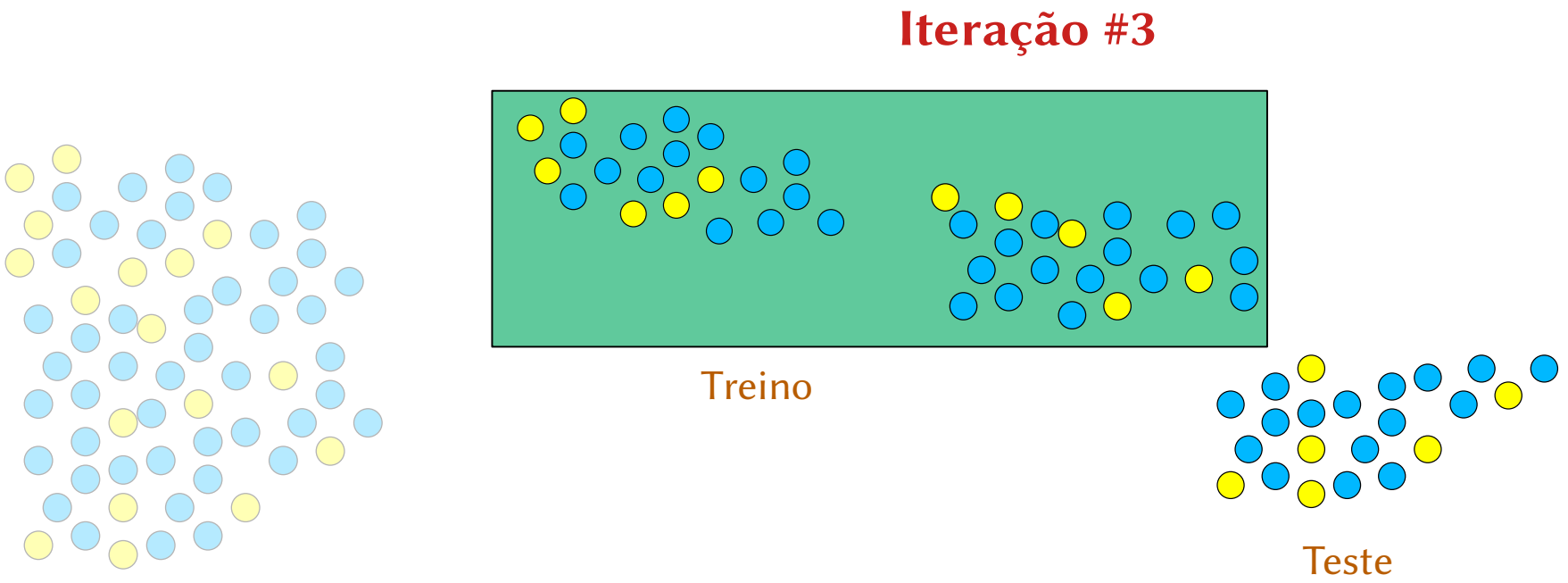
Validação cruzada

- 3-fold estratificado



Validação cruzada

- 3-fold estratificado



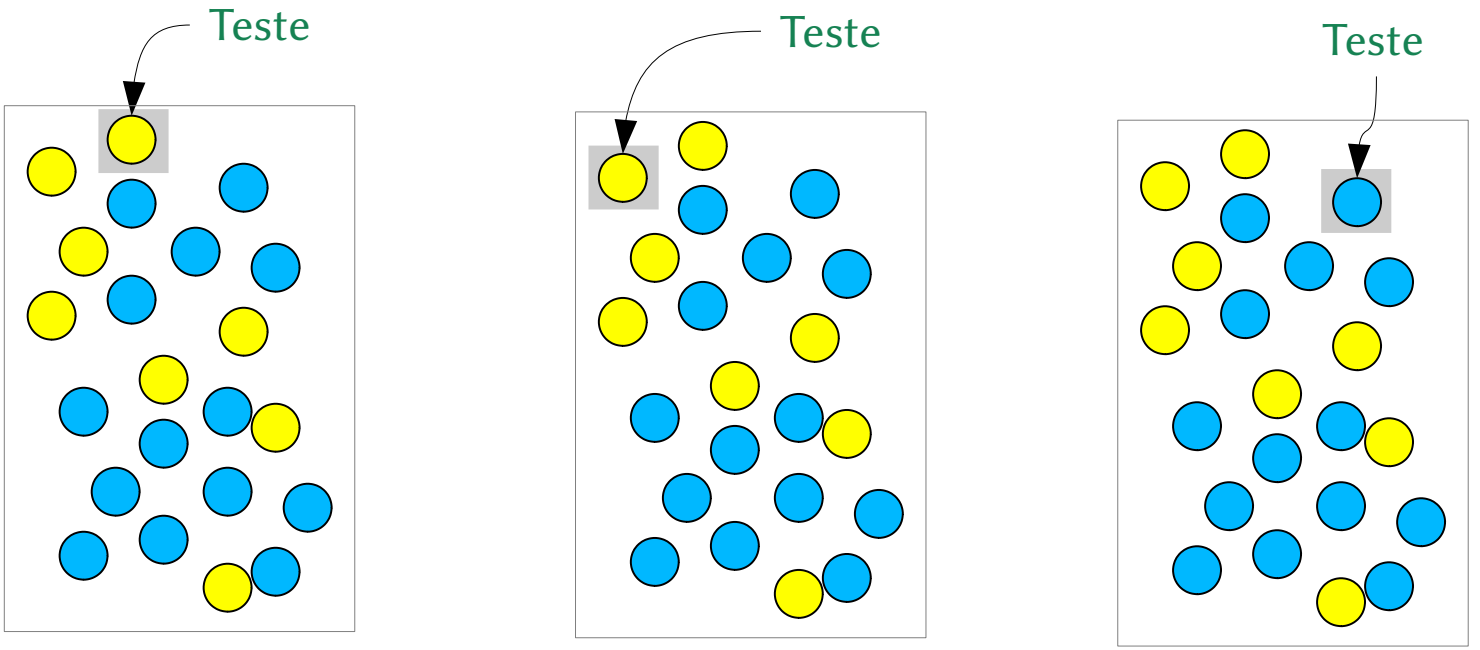
Validação cruzada

- Pontos positivos
  - Todos os exemplos são utilizados para teste
  - Baixo viés do estimador
- Pontos negativos
  - Poucos exemplos de teste em cada *fold*
  - Dificuldade de manter a estratificação em conjuntos com alto desbalanceamento de classes

Validação cruzada

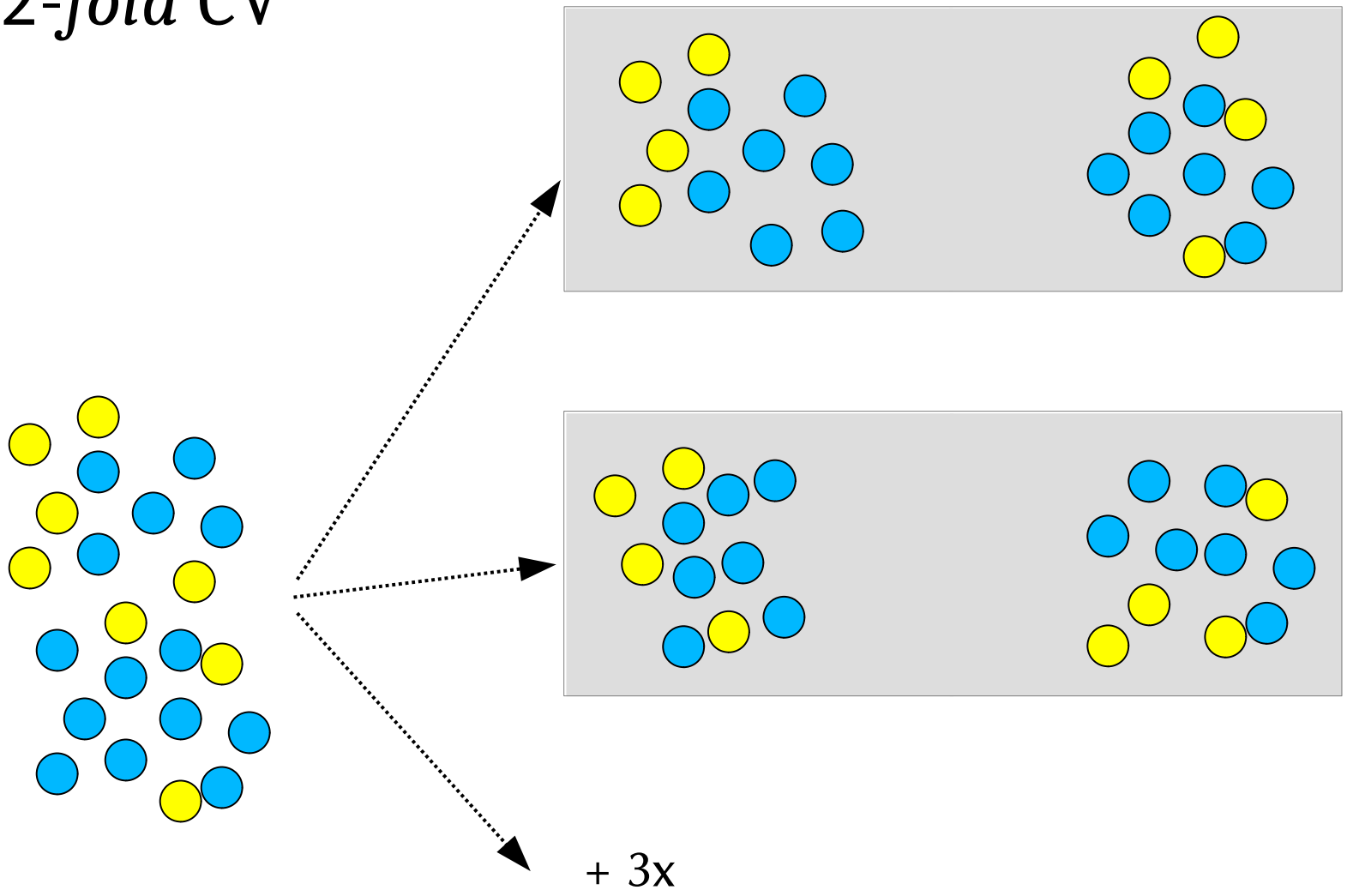
- No caso extremo em que k = N, temos *leave-one out*
  - Deixe um de fora (teste)
  - Treine com todos os restantes
- Outra forma de realizar validação cruzada é através da repetição de um certo k-fold
  - 10x10-CV → realiza 10 vezes 10-*fold* CV
  - 5x2-CV → realiza 5 vezes 2-*fold* CV

Leave-one-out



Validação cruzada

- 5x2-*fold* CV



Leave-one-out

- Pontos positivos
  - Utiliza todos os exemplos para teste
  - Utiliza o máximo disponível para treinamento
  - Adequado para poucos dados
- Pontos negativos
  - Computacionalmente custoso
  - Não existe estratificação

Repetição de k-fold

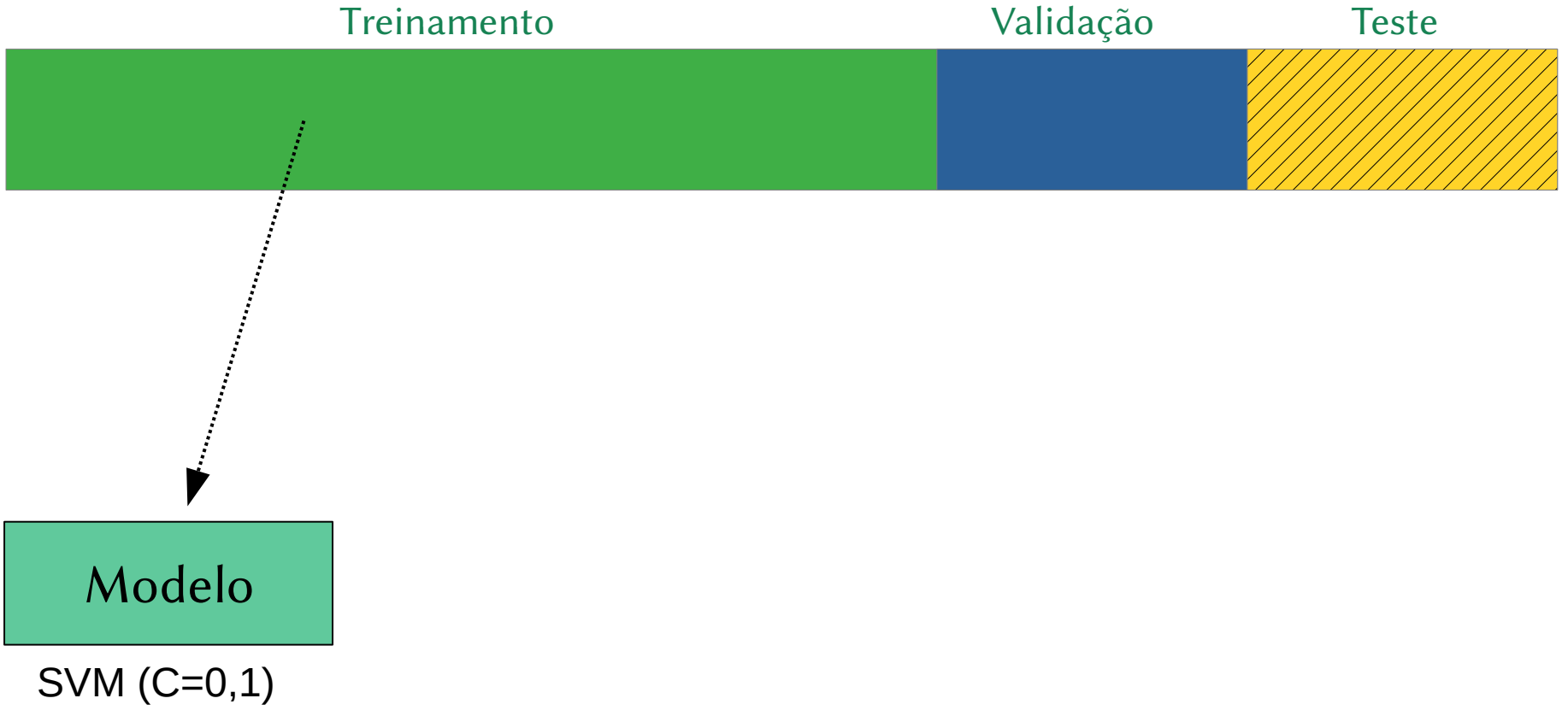
- Pontos positivos
  - Utiliza todos os exemplos para teste
  - Menos custoso que *leave-one-out*
  - Pode ser adequado para menor quantidade de dados
- Pontos negativos
  - Dificuldade de manter o conjunto estratificado em conjuntos desbalanceados

Avaliação e validação

- Conjuntos de treino e teste são adequados para testar modelos
- Mas e se quisermos ajustar hiperparâmetros?
  - Por exemplo, queremos estimar o desempenho do SVM em um conjunto de dados
  - Teremos *overfitting* se testarmos diferentes valores de C diretamente no conjunto de teste
    - $C \in \{0{,}1;\ 1;\ 10\}$

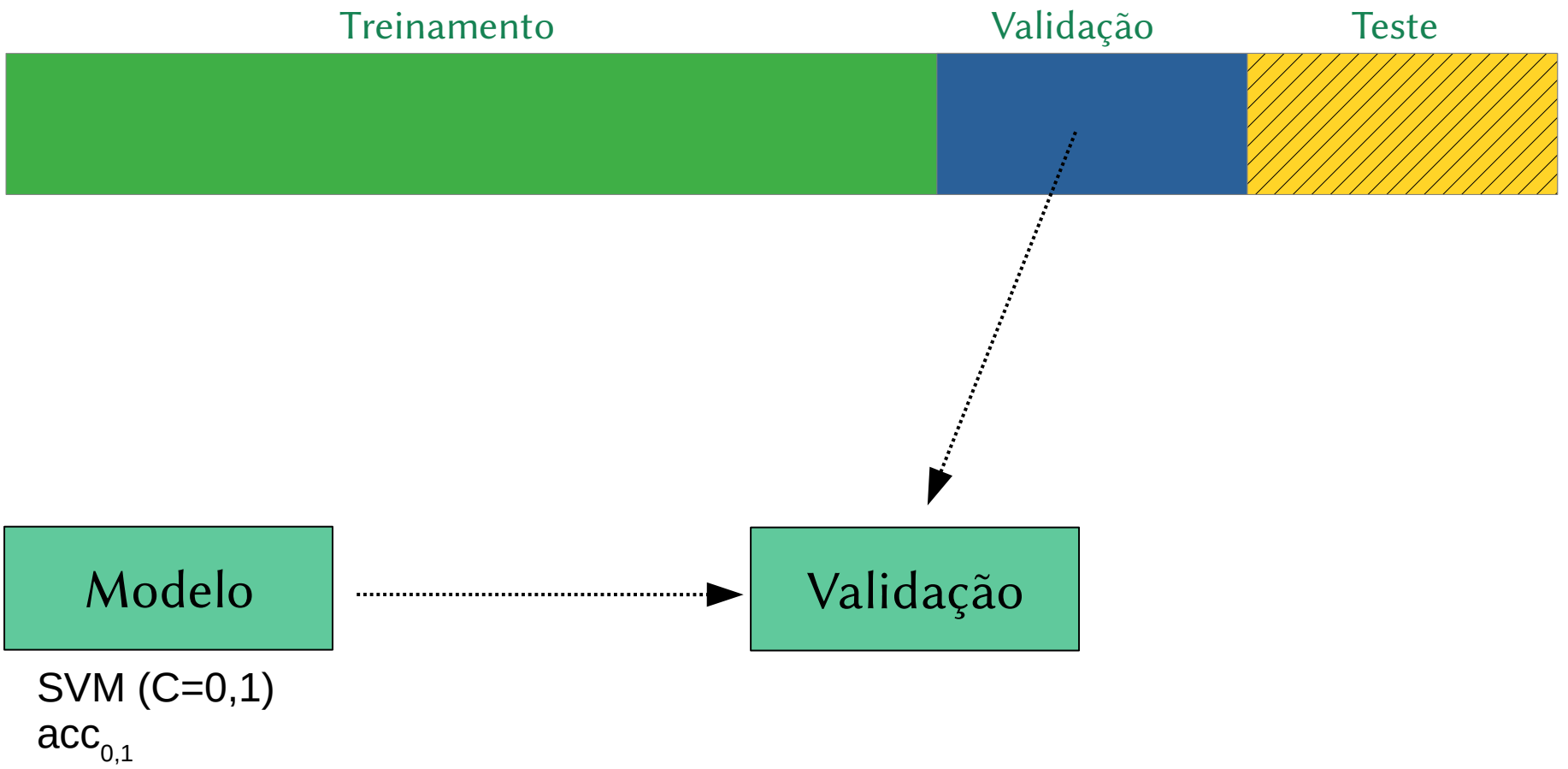
Avaliação e validação

- Empregamos, portanto, um conjunto de validação



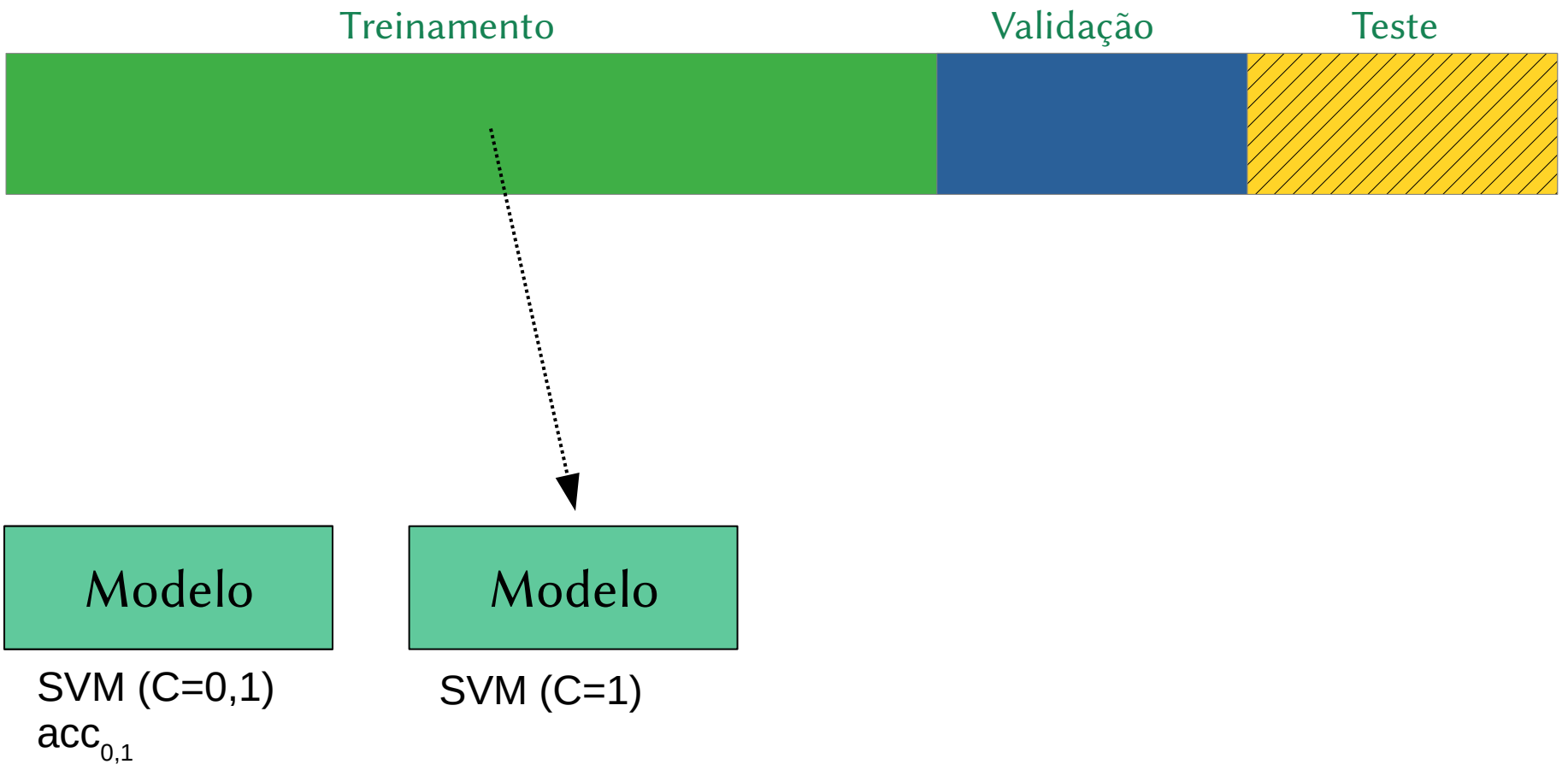
Avaliação e validação

- Empregamos, portanto, um conjunto de validação



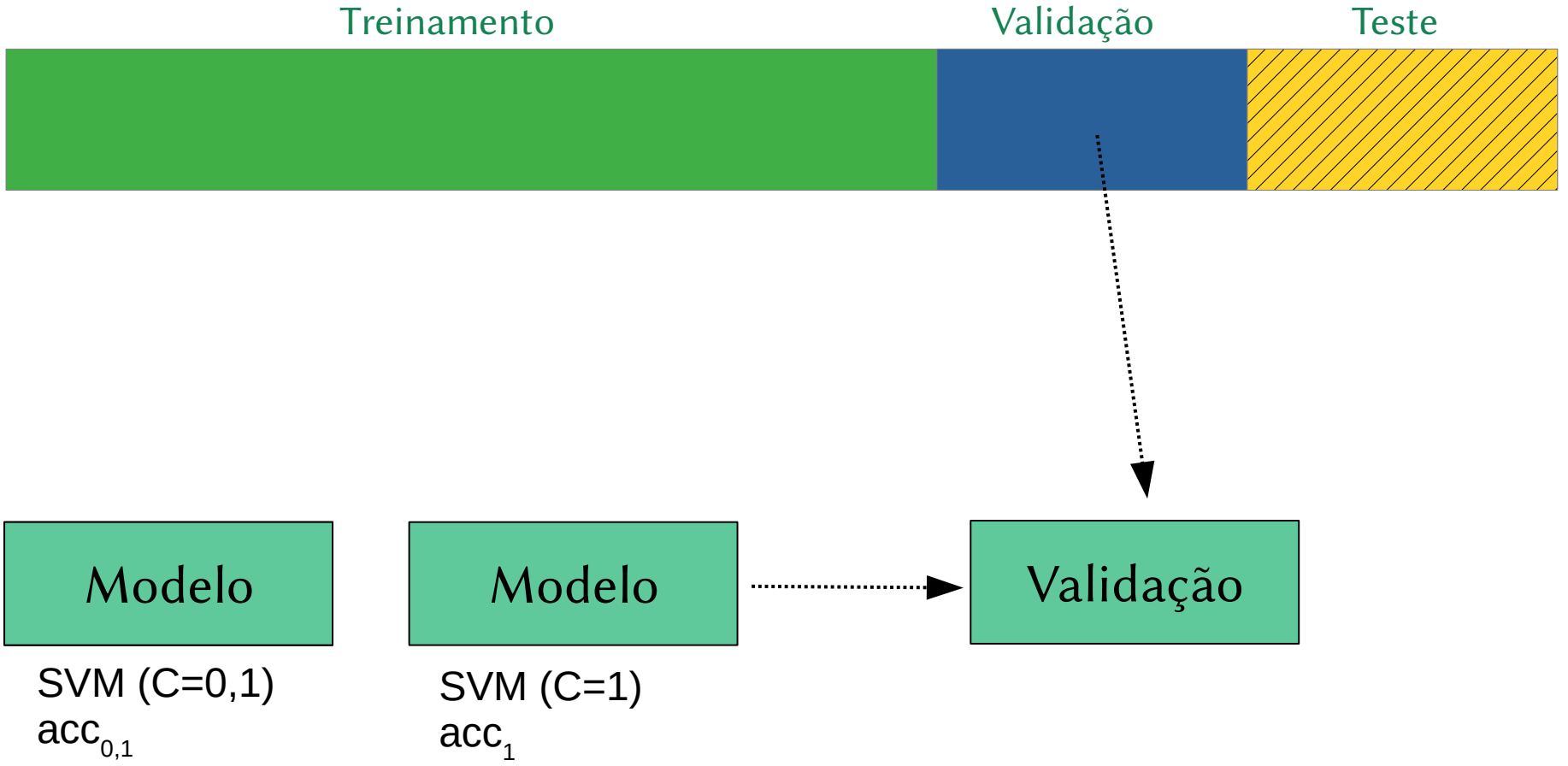
Avaliação e validação

- Empregamos, portanto, um conjunto de validação



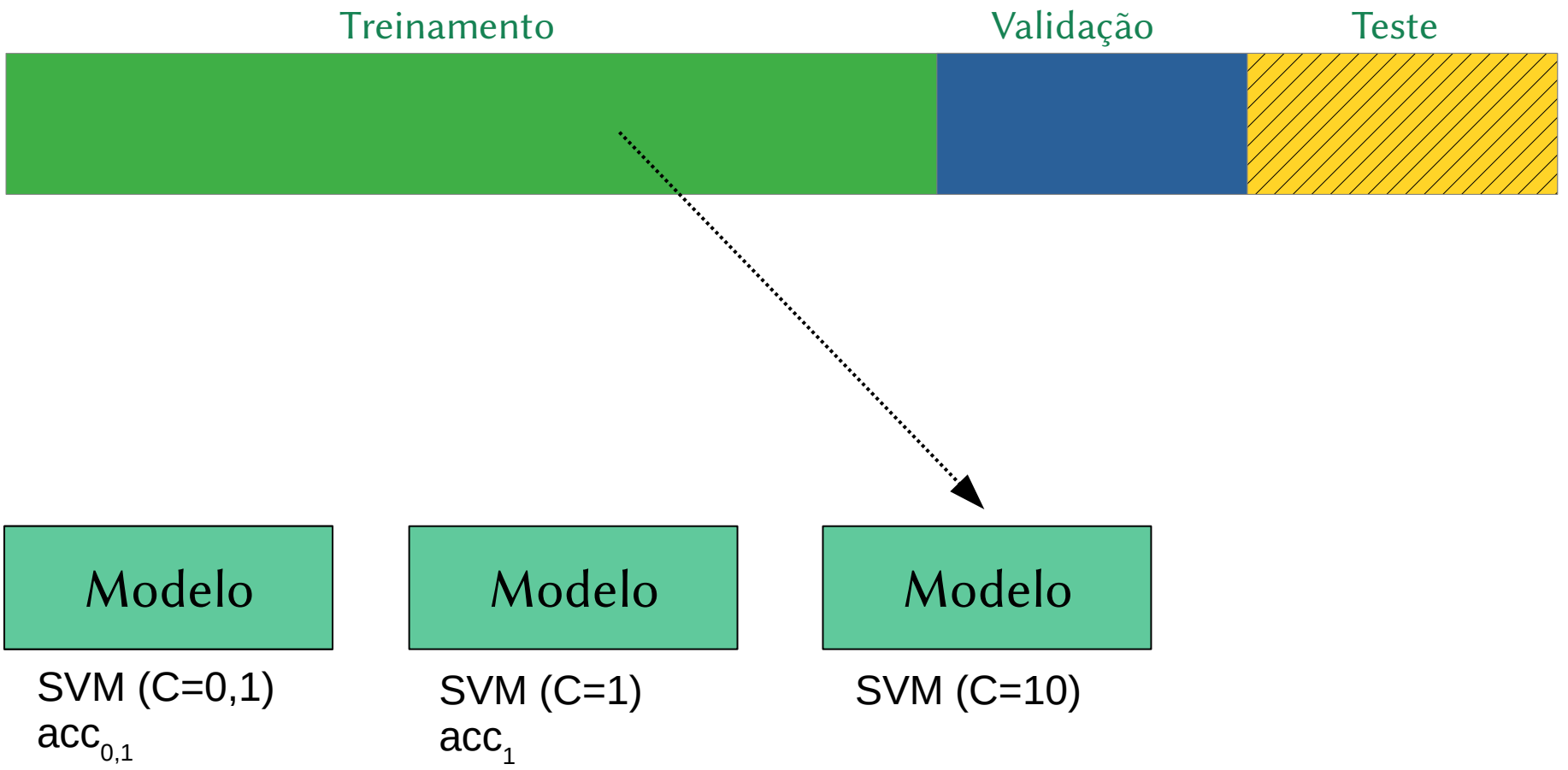
Avaliação e validação

- Empregamos, portanto, um conjunto de validação



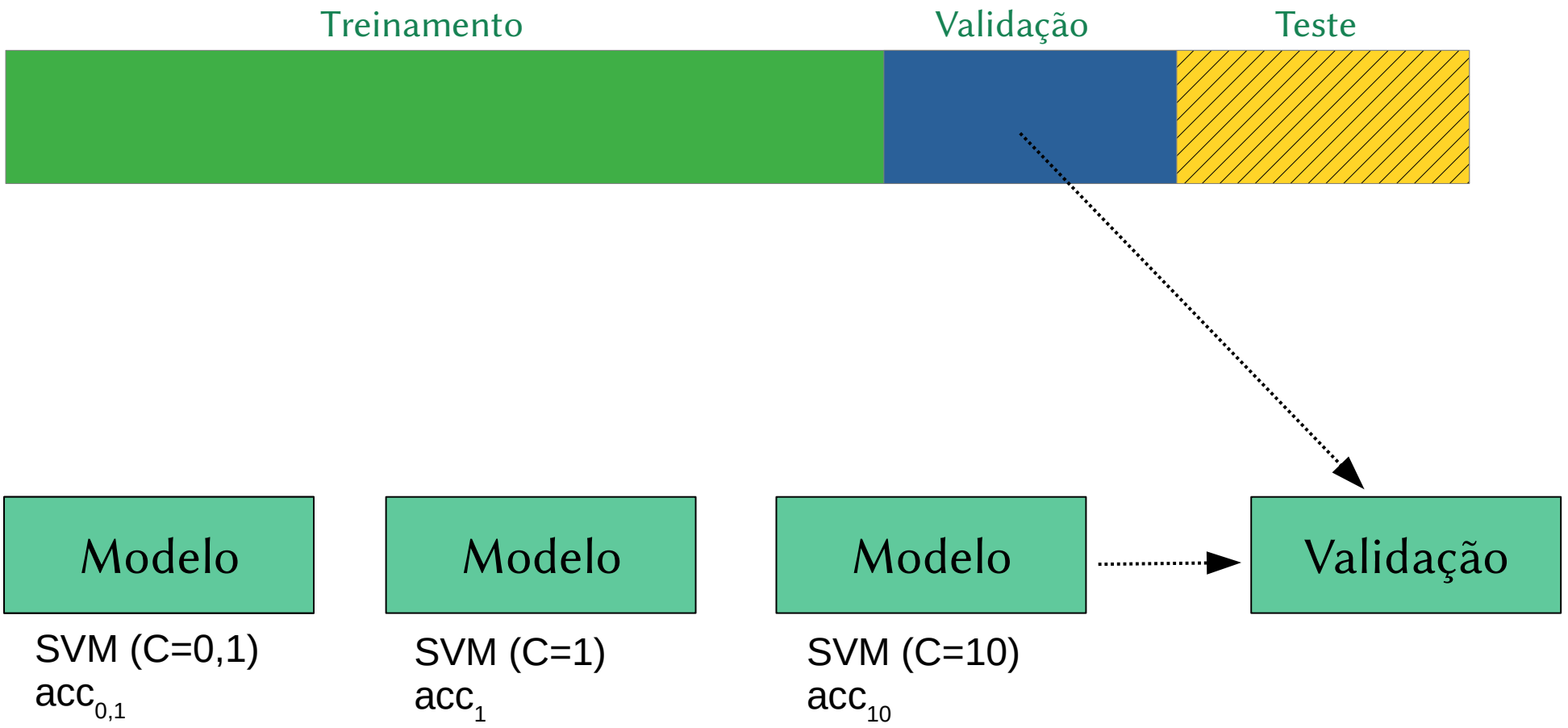
Avaliação e validação

- Empregamos, portanto, um conjunto de validação



Avaliação e validação

- Empregamos, portanto, um conjunto de validação



Avaliação e validação

- Empregamos, portanto, um conjunto de validação

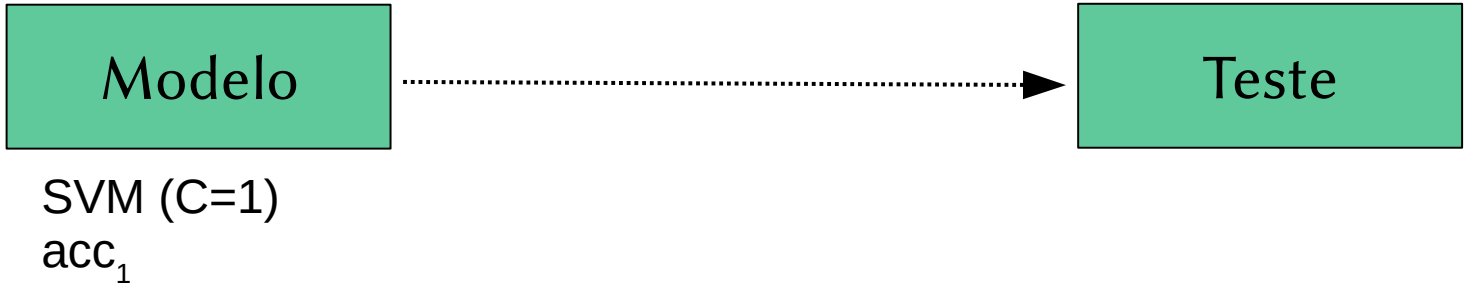
Qual hiperparâmetro gerou o modelo com maior acurácia? Suponha que $acc_1 > acc_{0,1} > acc_{10}$

Modelo	Modelo	Modelo
SVM (C=0,1) $acc_{0,1}$	SVM (C=1) acc_1	SVM (C=10) acc_{10}

Avaliação e validação

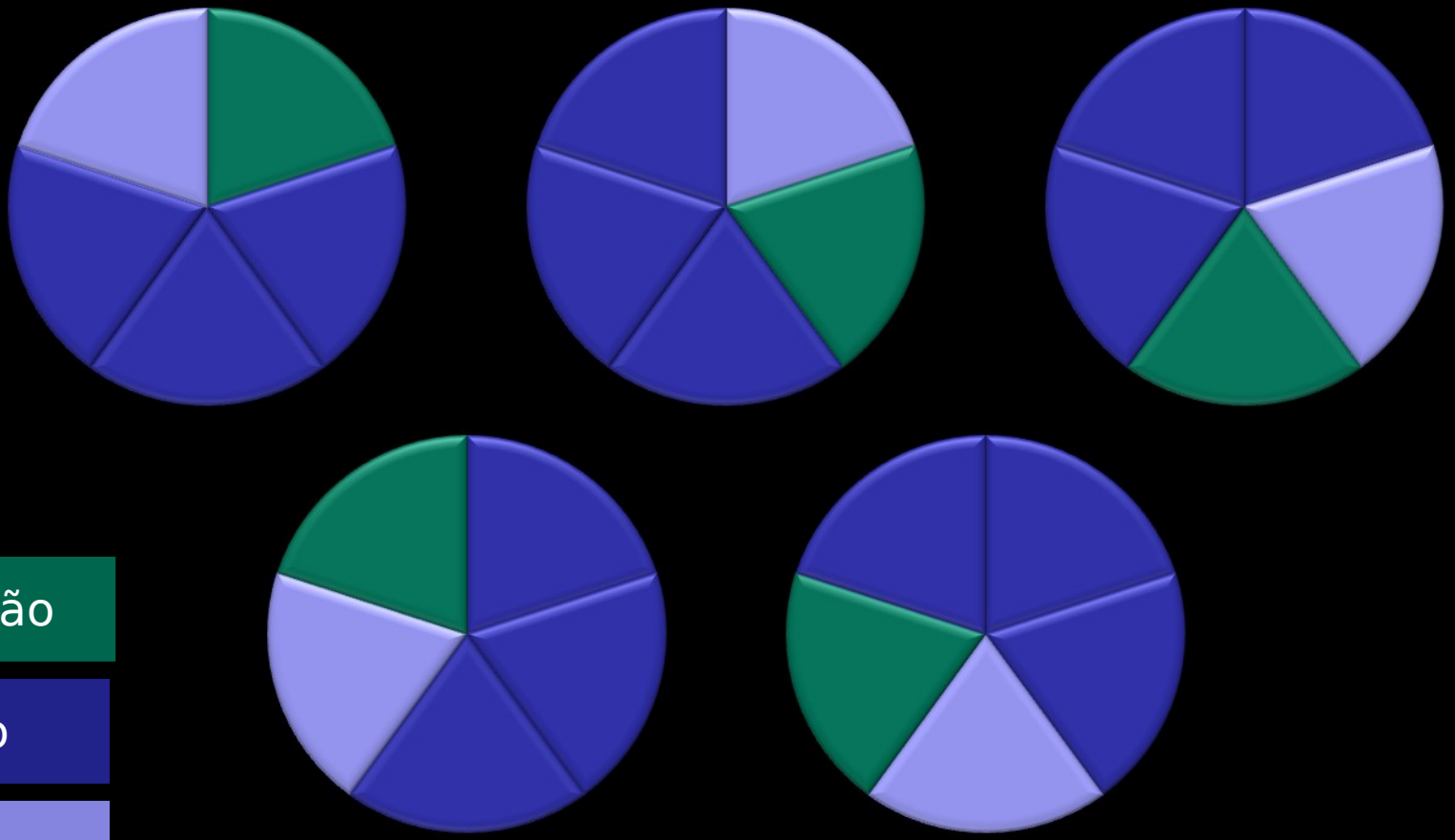
- Empregamos, portanto, um conjunto de validação

Qual hiperparâmetro gerou o modelo com maior acurácia? Suponha que $acc_1 > acc_{0,1} > acc_{10}$



k-Fold com teste e validação

5-*fold cross validation* com *folds* de teste e de validação



Bootstrap

- Método de avaliação baseado em reamostragem com substituição
- Parte do princípio que a amostra que você possui é o conjunto mais representativo do domínio de aplicação
- Reamostra repetidamente desse conjunto
  - **Com substituição**, isto é, pode haver instâncias duplicadas

Bootstrap

- Para uma amostra de N exemplos, reamostre com repetição, N vezes para formar um novo conjunto que contém N exemplos
  - Utilize esses exemplos como conjunto de treinamento
- Verifique no conjunto original os exemplos que não foram amostrados
  - Utilize esses exemplos como conjunto de teste
- Repita várias vezes e tire a média dos erros

Bootstrap 0.632

- Esse procedimento específico é denominado *bootstrap* 0.632
  - Cada exemplo tem probabilidade 1/n de ser selecionada pelo menos uma vez
  - A probabilidade de um exemplo não ser selecionado nenhuma vez é

$$\left(1 - \frac{1}{n}\right)^n \approx e^{-1} \approx 0.368$$

Bootstrap 0.632

- O conjunto de treinamento gerado pelo *bootstrap 0.632* contém, em média, aproximadamente 63,2% dos exemplos do conjunto original
  - Poucos exemplos de treinamento
  - Pode levar a uma estimativa pessimista
  - Recomenda-se ponderar o erro com o erro empírico

$$err = 0.632 \times e_{\text{test instances}} + 0.368 \times e_{\text{training\_instances}}$$

Bootstrap

- Pontos positivos
  - Respeita a distribuição do conjunto original
  - Menos impactante em conjuntos desbalanceados
- Pontos negativos
  - Pode ser inadequado para métodos sensíveis a exemplos repetidos
  - O número de repetições é objeto de debate (alguns defendem 50, 100... 1000 iterações)