

ANÁLISE INTELIGENTE DE DADOS (COB 754)

---

# APRENDIZADO DE MÁQUINA

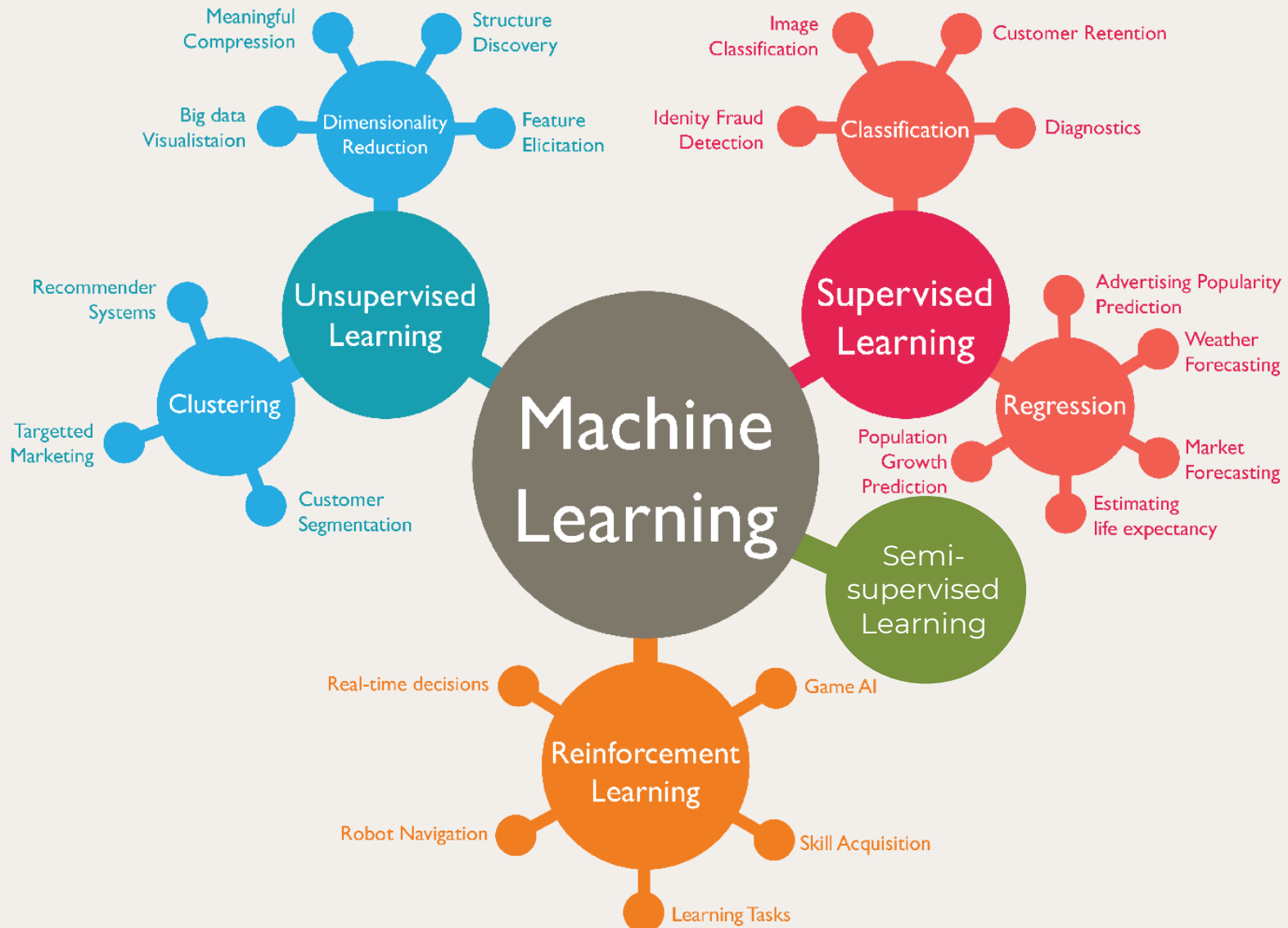
LETÍCIA MARTINS RAPOSO

# DEFINIÇÃO

---



CONJUNTO DE REGRAS  
E PROCEDIMENTOS,  
QUE PERMITE QUE OS  
COMPUTADORES  
POSSAM AGIR E TOMAR  
DECISÕES BASEADOS  
EM DADOS AO INVÉS  
DE SER  
EXPLICITAMENTE  
PROGRAMADOS PARA  
REALIZAR UMA  
DETERMINADA TAREFA.

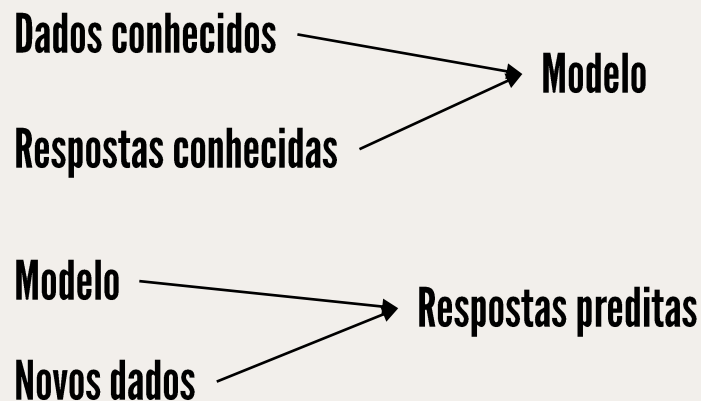


<https://becominghuman.ai/an-introduction-to-machine-learning-7db04da817c4>

# SUPERVISIONADO

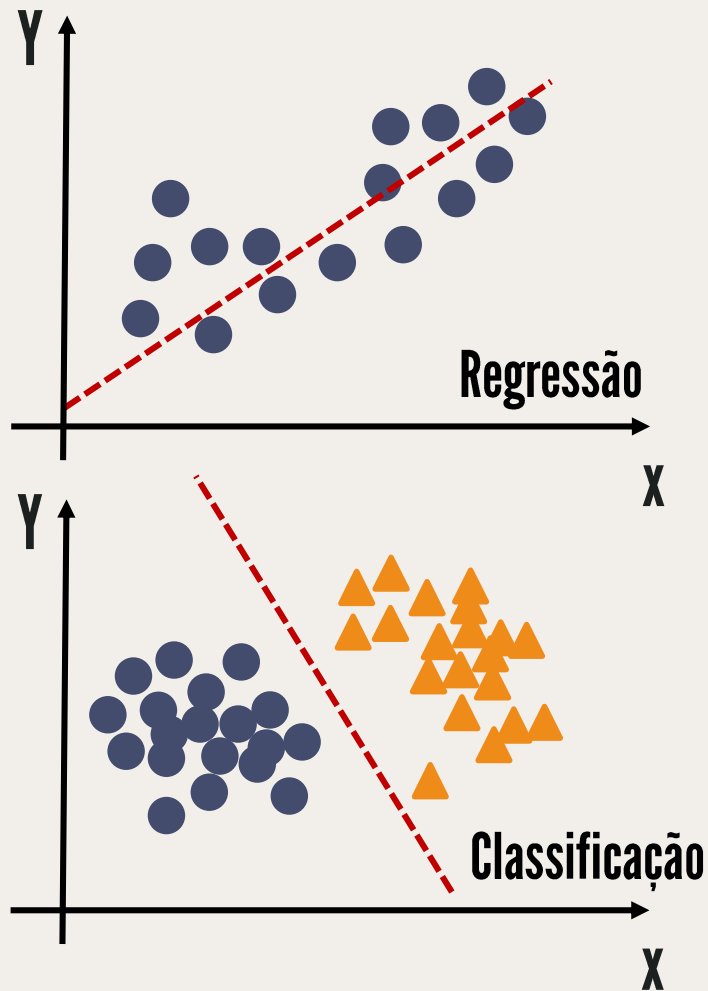
---

O TERMO SUPERVISIONADO VEM DA SIMULAÇÃO DA PRESENÇA DE UM “SUPERVISOR EXTERNO”, QUE CONHECE A SAÍDA (RÓTULO) DESEJADA PARA CADA EXEMPLO.



- TAREFA DE ENCONTRAR UMA FUNÇÃO A PARTIR DE DADOS DE TREINAMENTO ROTULADOS.
- O OBJETIVO É ENCONTRAR OS PARÂMETROS ÓTIMOS QUE AJUSTEM UM MODELO QUE POSSA PREVER RÓTULOS DESCONHECIDOS EM OUTROS OBJETOS (O CONJUNTO DE TESTE).

# SUPERVISIONADO

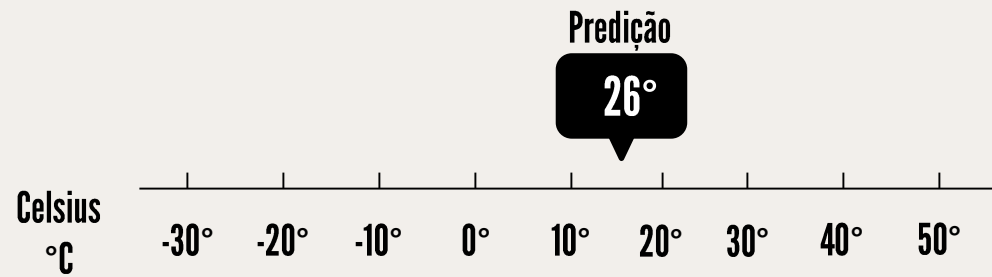


- REGRESSÃO: O OBJETIVO É PREVER UMA MEDIDA CONTÍNUA PARA UMA OBSERVAÇÃO. OU SEJA, AS VARIÁVEIS RESPOSTAS SÃO NÚMEROS REAIS.
- CLASSIFICAÇÃO: O OBJETIVO É ATRIBUIR UMA CLASSE (OU RÓTULO) DE UM CONJUNTO FINITO DE CLASSES A UMA OBSERVAÇÃO. OU SEJA, AS RESPOSTAS SÃO VARIÁVEIS CATEGÓRICAS.



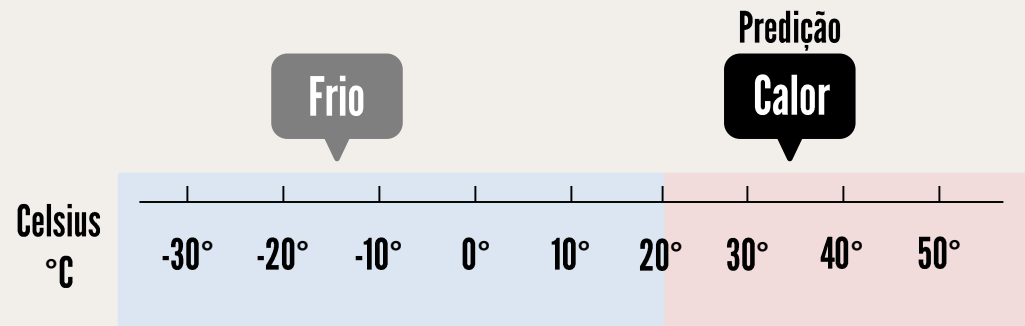
## REGRESSÃO

Qual será a temperatura amanhã?



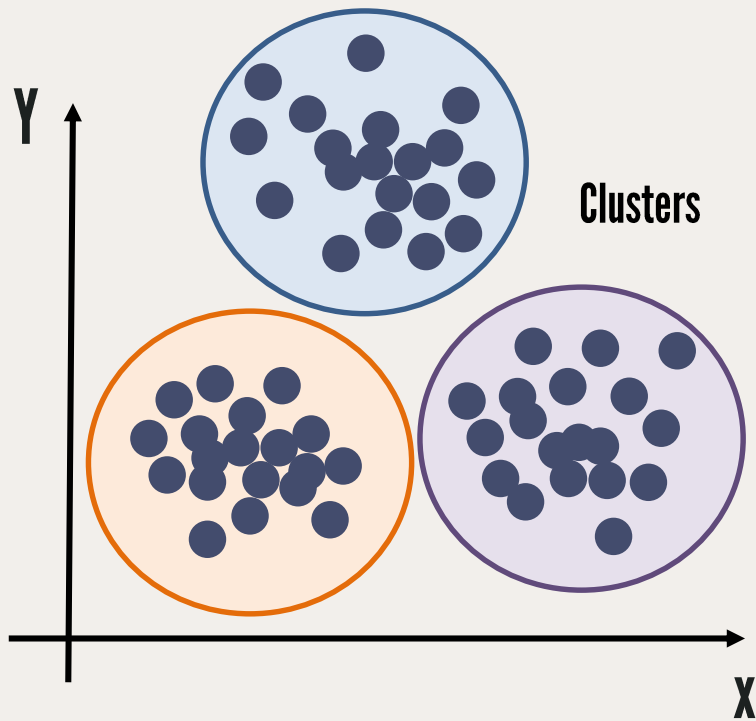
## CLASSIFICAÇÃO

Vai fazer frio ou calor amanhã?



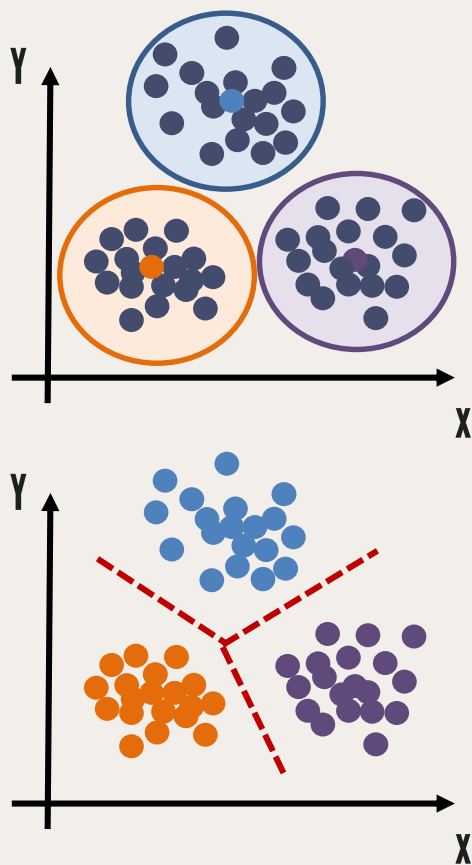
# NÃO SUPERVISIONADO

---



- O CONJUNTO DE TREINAMENTO NÃO É ROTULADO.
- NOSSO OBJETIVO É OBSERVAR ALGUMAS SIMILARIDADES ENTRE OS OBJETOS E INCLUI-LOS EM GRUPOS APROPRIADOS.
- TAMBÉM PODE SER USADA PARA REDUZIR O NÚMERO DE DIMENSÕES EM UM CONJUNTO DE DADOS PARA CONCENTRAR SOMENTE NOS ATRIBUTOS MAIS ÚTEIS.

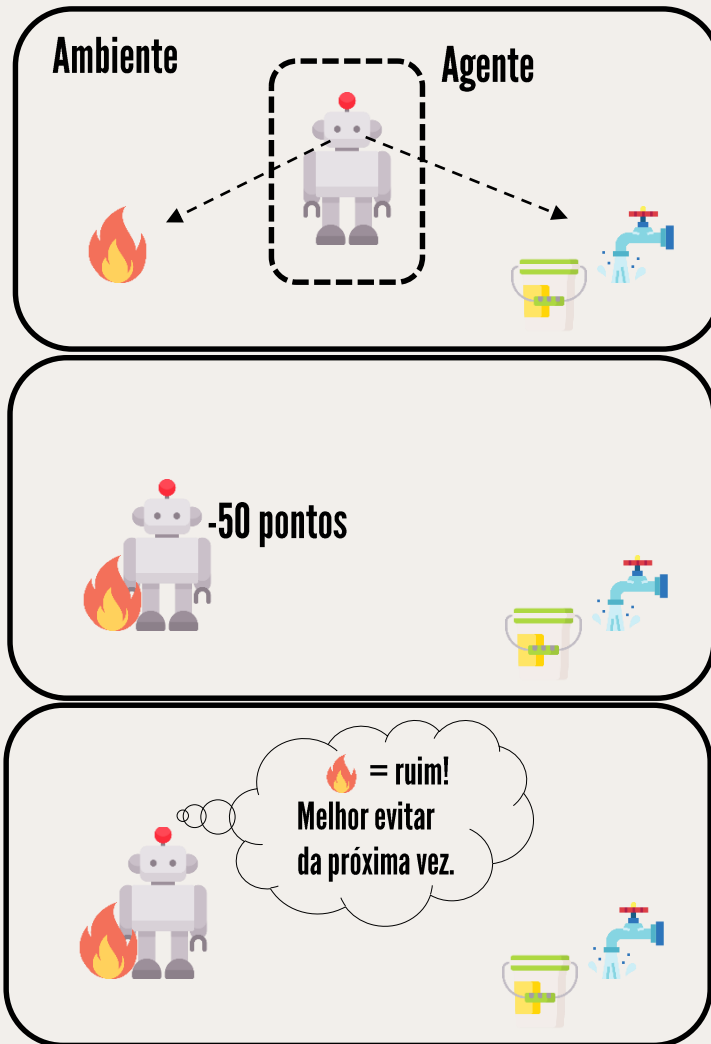
# SEMI-SUPERVISIONADO



- INCLUI AMBOS OS PROBLEMAS DISCUTIDOS ANTIORMENTE: ELA USA DADOS ROTULADOS E NÃO-ROTULADOS.
- A IDEIA É ROTULAR DADOS NÃO ROTULADOS POR UM APRENDIZADO NÃO SUPERVISIONADO E DEPOIS CONSTRUIR UM MODELO SUPERVISIONADO.



# POR REFORÇO,



- UM AMBIENTE FORNECERÁ DADOS AOS QUAIS UM “AGENTE” IRÁ TOMAR DECISÕES DE ACORDO ELES.
- ESSE AMBIENTE IRÁ RESPONDER COM UMA RECOMPENSA/PUNIÇÃO QUE DETERMINARÁ A ALTERAÇÃO DE REGRAS DE DECISÃO FEITA PELO “AGENTE” PARA ATUAR MELHOR NO FUTURO.



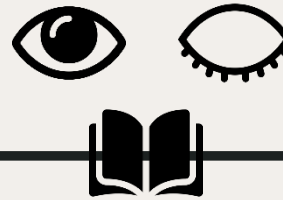
# APRENDIZADOS



SUPERVISIONADO



NÃO  
SUPERVISIONADO



SEMI-  
SUPERVISIONADO



POR REFORÇO

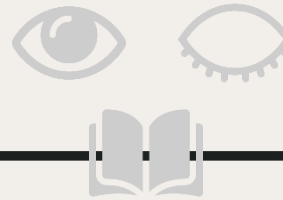
# APRENDIZADOS



SUPERVISIONADO



NÃO  
SUPERVISIONADO



SEMI-  
SUPERVISIONADO

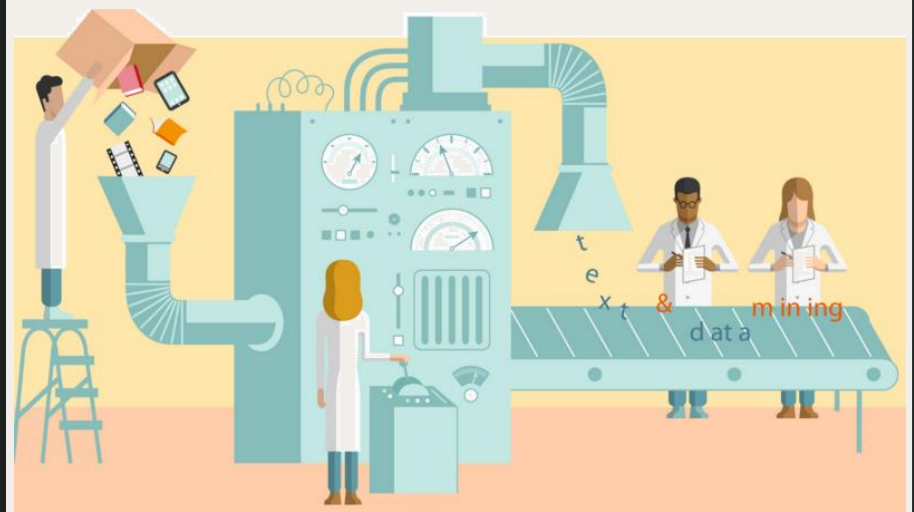


POR REFORÇO

- PREPARAÇÃO DOS DADOS
- ESCOLHA DE UM ALGORITMO
- TREINAMENTO
- AVALIAÇÃO
- AJUSTE DOS PARÂMETROS
- PREDIÇÃO

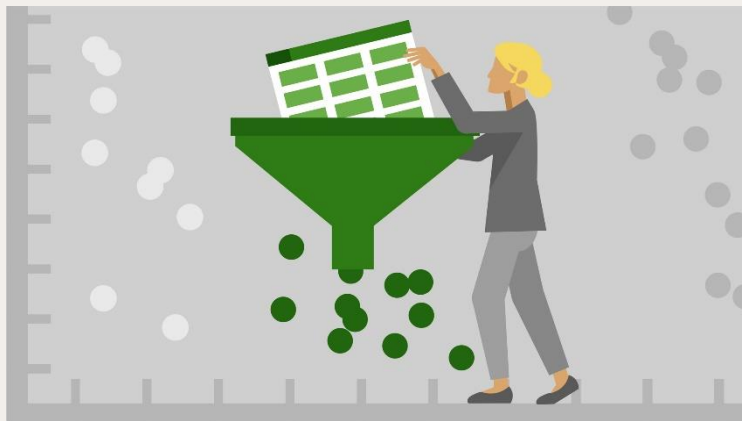
# ETAPAS DO APRENDIZADO SUPERVISIONADO

---



# PREPARAÇÃO DOS DADOS

---



- ELIMINAÇÃO MANUAL DE ATRIBUTOS
- INTEGRAÇÃO DE DADOS
- AMOSTRAGEM DE DADOS
- DADOS DESBALANCEADOS
- LIMPEZA DE DADOS
- TRANSFORMAÇÃO DE DADOS
- REDUÇÃO DE DIMENSIONALIDADE

# ESCOLHA DE UM ALGORITMO

---



EXISTEM *TRADE-OFFS* ENTRE VÁRIAS CARACTERÍSTICAS DE ALGORITMOS, COMO:

- VELOCIDADE DE TREINAMENTO
- USO DE MEMÓRIA
- ACURÁCIA PREDITIVA EM NOVOS DADOS
- TRANSPARÊNCIA OU INTERPRETABILIDADE (FACILIDADE PARA ENTENDER AS RAZÕES PELAS QUAIS UM ALGORITMO FAZ SUAS PREVISÕES)

# **TREINAMENTO**

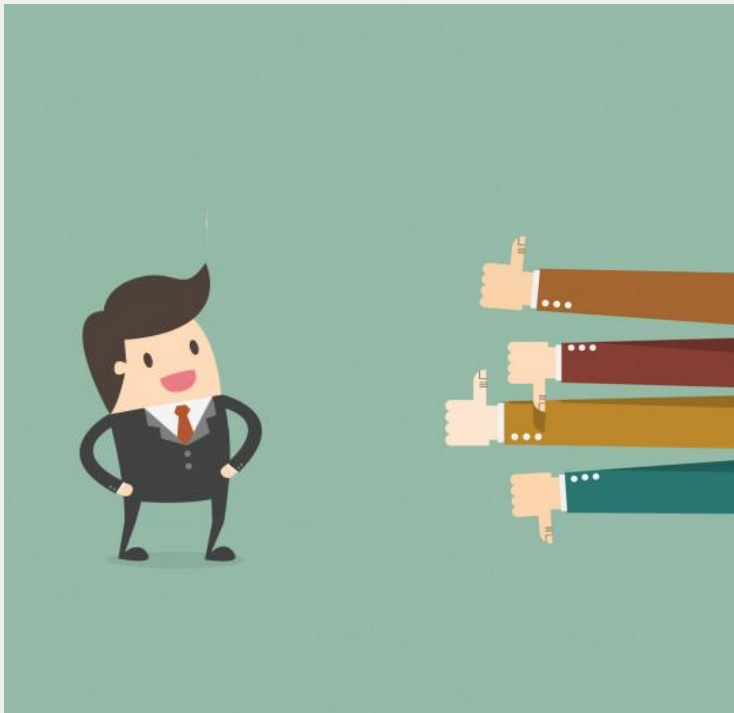
---



- REGRESSÃO LINEAR
- REGRESSÃO LOGÍSTICA
- K-NN
- NAÏVE BAYES
- ÁRVORES DE DECISÃO
- FLORESTA ALEATÓRIA

# AVALIAÇÃO

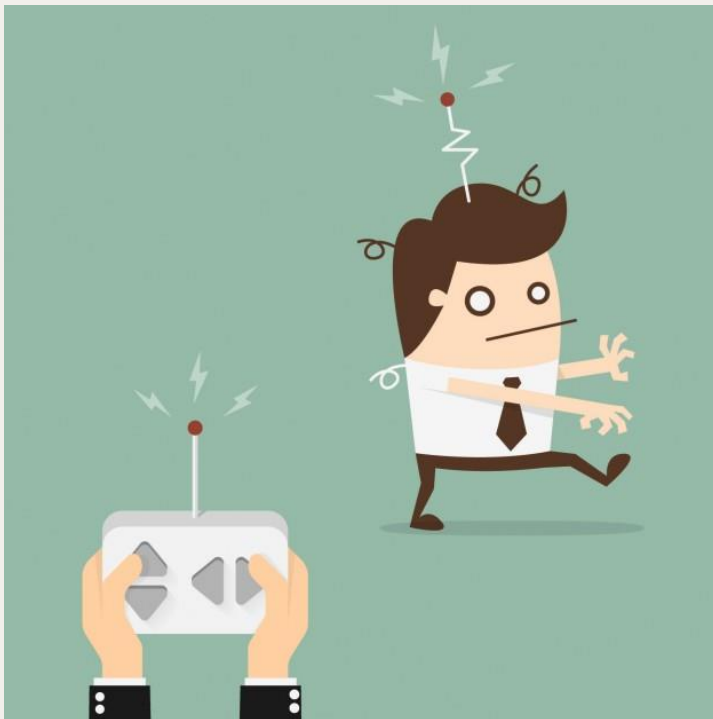
---



- QUANDO O TREINAMENTO ESTIVER CONCLUÍDO, É HORA DE VER SE O MODELO ESTÁ BOM.
- A AVALIAÇÃO NOS PERMITE TESTAR NOSSO MODELO EM RELAÇÃO A DADOS QUE NUNCA FORAM USADOS PARA TREINAMENTO.
- ISSO DEVE SER REPRESENTATIVO DE COMO O MODELO PODE FUNCIONAR NO MUNDO REAL.



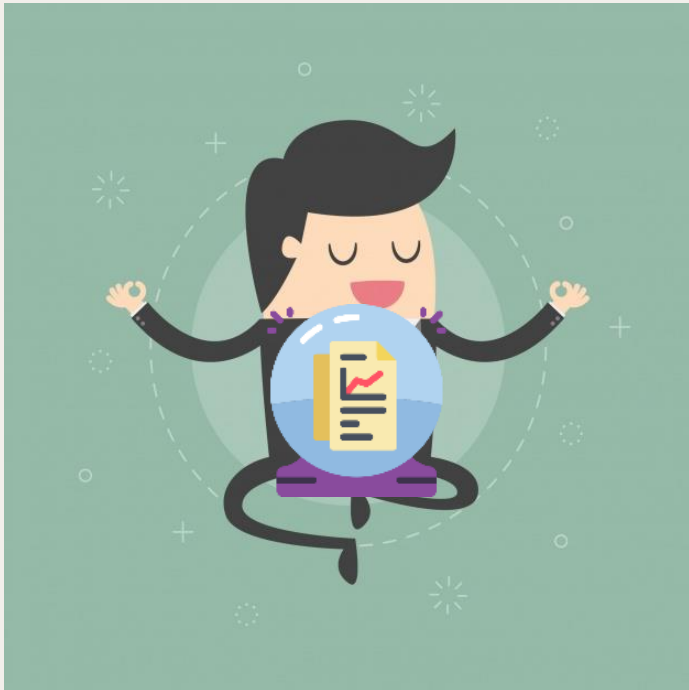
# AJUSTE DOS PARÂMETROS



- TENTAR MELHORAR AINDA MAIS O MODELO.
- É IMPORTANTE DEFINIR O QUE TORNA UM MODELO “BOM O SUFICIENTE”, CASO CONTRÁRIO, VOCÊ PODE SE ENCONTRAR AJUSTANDO PARÂMETROS POR UM TEMPO MUITO LONGO.
- ESSES PARÂMETROS SÃO GERALMENTE CHAMADOS DE “HIPERPARÂMETROS”.

# PREDIÇÃO

---



- ETAPA EM QUE O MODELO IRÁ PREDIZER A RESPOSTA A NOVOS CASOS.
- TODO O TRABALHO REALIZADO TEM COMO OBJETIVO CHEGAR NESTA ETAPA.



---

# AVALIAÇÃO DE MODELOS PREDITIVOS

# AMOSTRAGEM

---

- Importante a divisão entre conjunto de treinamento e teste.
- Uso do mesmo conjunto de treino na avaliação: estimativas otimistas.
- Treinamento: indução e ajuste do modelo.
- Teste: simulam a apresentação de objetos novos ao preditor que não foram vistos em sua indução.
- Treinamento e teste: conjuntos disjuntos.

# MÉTODO HOLDOUT

---



The diagram illustrates the Holdout method for data partitioning. It consists of two horizontal bars. The top bar is a single yellow rectangle labeled 'Dados'. The bottom bar is divided into two segments: a light orange segment on the left labeled 'Treinamento' and a darker orange segment on the right labeled 'Teste'. This visualizes the process of splitting a dataset into training and testing subsets.

**Dados**

**Treinamento**

**Teste**

Dividimos, aleatoriamente, os dados em conjuntos de treinamento e teste (geralmente  $2/3$  para o treinamento e  $1/3$  para o teste)

# MÉTODO HOLDOUT

---



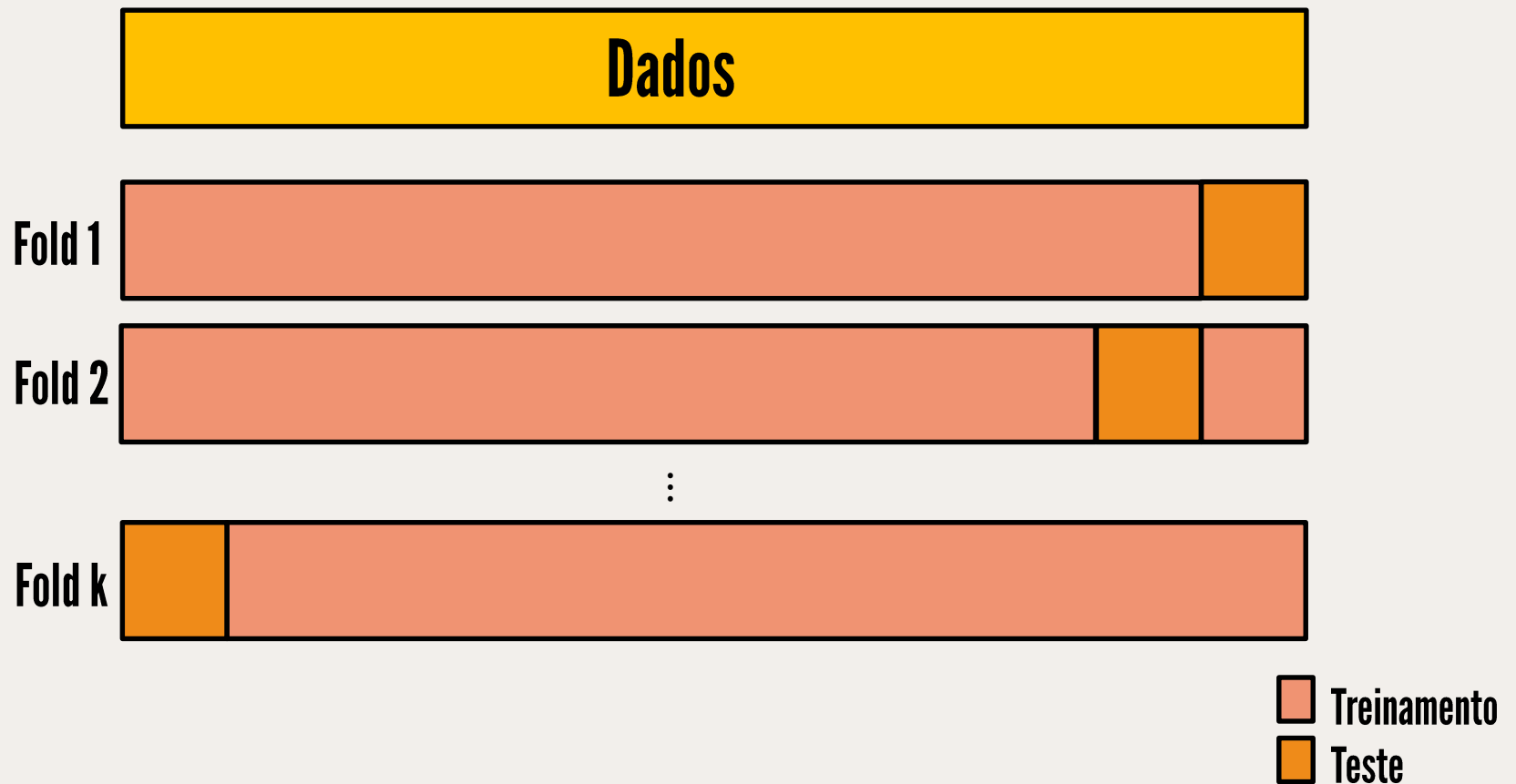
Vantagens: dados totalmente independentes; só precisa ser executado uma vez, portanto, tem custos computacionais mais baixos.



Desvantagens: a avaliação de desempenho está sujeita a maior variância, dado o tamanho menor dos dados; difícil calcular qualquer informação de variação ou intervalos de confiança em um único conjunto de dados.

# VALIDAÇÃO CRUZADA K-FOLD

---



# VALIDAÇÃO CRUZADA K-FOLD

---

- Os dados são normalmente estratificados antes de serem divididos em subconjuntos:
  - A estratificação é o processo de reorganização dos dados para garantir que cada subconjunto seja um bom representante do todo.
  - P. ex., em um problema de classificação binária em que cada classe compreende 50% dos dados, o ideal é organizar os dados de modo que em cada subconjunto, cada classe inclui cerca de metade das instâncias.
- A estimativa do erro é obtida pela média dos erros de cada rodada.



# VALIDAÇÃO CRUZADA K-FOLD

---



Vantagens: propenso a menor variação porque usa todo o conjunto de treinamento.



Desvantagens: maiores custos computacionais - o modelo precisa ser treinado K vezes na etapa de avaliação e mais uma para gerar o modelo final.

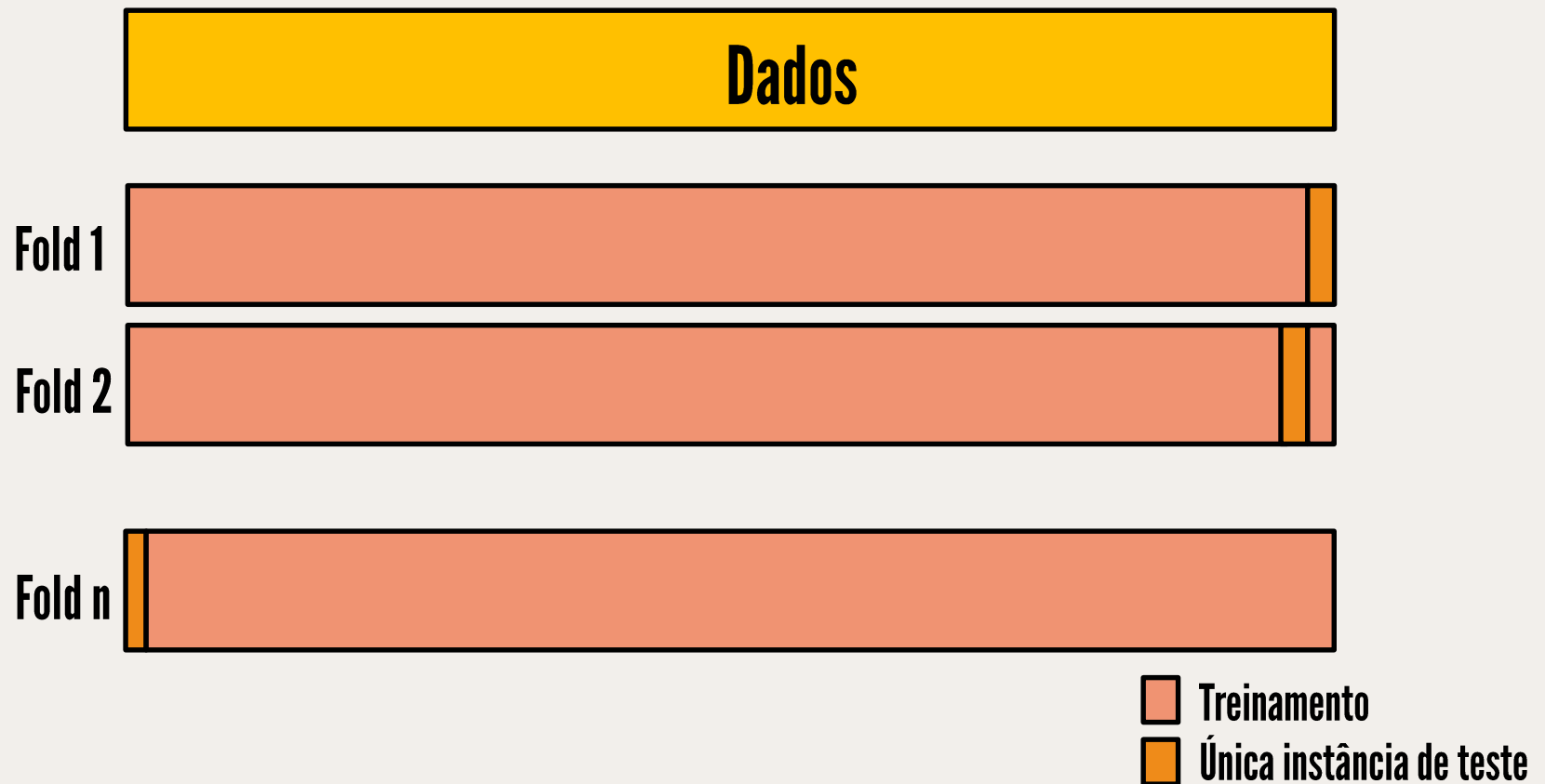
# VALIDAÇÃO CRUZADA LEAVE-ONE-OUT

---

- Caso especial da *k-fold*, com k igual ao número de instâncias nos dados.
- Em cada iteração, quase todos os dados, exceto uma única observação, são usados para treinamento e o modelo é testado nessa única observação.
- Amplamente utilizada quando os dados disponíveis são muito raros.
- O desempenho é dado pela soma dos desempenhos verificados para cada exemplo de teste individual.

# VALIDAÇÃO CRUZADA LEAVE-ONE-OUT

---



# VALIDAÇÃO CRUZADA LEAVE-ONE-OUT

---



Vantagens: propenso a menos variação porque usa todo o conjunto de treinamento.



Desvantagens: a estratificação não é possível.

# OBSERVAÇÕES

---

- Na prática, a escolha do número de subconjuntos depende do tamanho do conjunto de dados.
- Para conjuntos grandes,  $k$  igual a 3 já fornecerá bons resultados.
- Para conjuntos mais esparsos, talvez a melhor opção seja o *leave-one-out* para treinar com um maior número possível.
- Uma escolha comum para  $k$  é 10.

# OBSERVAÇÕES

---

- O propósito da validação cruzada não é chegar ao modelo final, mas sim avaliar o(s) modelo(s).
- Para construir o modelo final, devemos usar todos os dados que temos para chegar ao melhor modelo possível.
- P. ex., digamos que temos dois modelos, um modelo de regressão linear e uma rede neural. Como podemos dizer qual modelo é melhor?
  - Podemos fazer validação cruzada *k-fold* e ver qual deles se mostra melhor na previsão dos exemplos de teste. Uma vez que usamos a validação cruzada para selecionar o modelo com melhor desempenho, treinamos esse modelo (quer seja a regressão linear ou a rede neural) com todos os dados.

# BOOTSTRAP

---

- Método de reamostragem proposto por Bradley Efron em 1979.
- Técnica de reamostragem com reposição ( $\neq$  validação cruzada).
- A partir de um conjunto de dados com  $N$  exemplos, selecionamos, com reposição,  $N$  exemplos e utilizamos esse subconjunto para treino.
- Os exemplos restantes que não foram selecionados serão usados no teste.
- Repetimos esse processo  $k$  vezes (normalmente  $k \geq 100$ ).
- O erro estimado também será dado pela média dos erros de cada experimento.

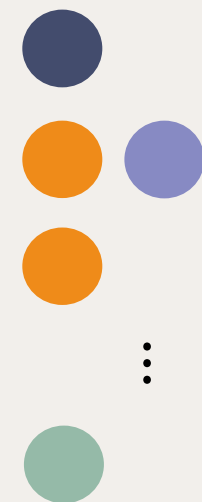
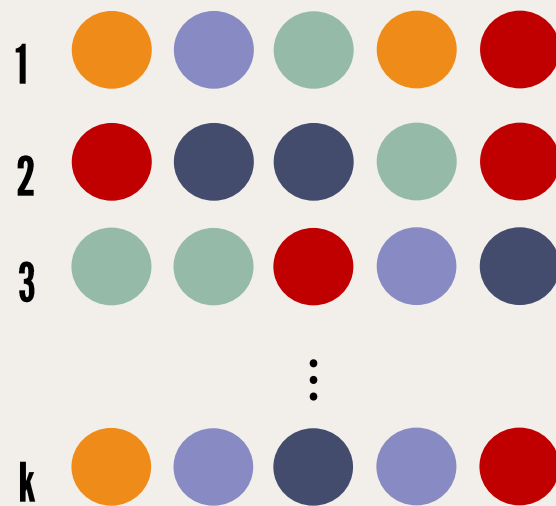
# BOOTSTRAP

---



Treinamento

Teste





# BOOTSTRAP 0.632

---

- Um exemplo tem uma probabilidade de  $1 - \frac{1}{N}$  de não ser escolhido.
- Assim, sua probabilidade de terminar nos dados do teste é:  $\left(1 - \frac{1}{N}\right)^N$ . Para N grande, essa probabilidade é aproximadamente  $\frac{1}{e} \approx 0,368$ .
- O conjunto de treinamento vai conter aproximadamente 63,2% dos exemplos.
- A taxa de erro é um estimador muito pessimista (usa 36,8% dos exemplos).
- Solução: usar também a taxa de erro obtida no conjunto de treinamento.

# BOOTSTRAP 0.632

	$\approx 63\%$ dos exemplos de $D$	$\approx 37\%$ de $D$	Estimativas da Taxa de Erro
Experiência nº 1	Conjunto de Treino: $D_1$	Conjunto de Teste: $D \setminus D_1$	$E_1 = 0.632 E_{\text{teste}} + 0.368 E_{\text{treino}}$
Experiência nº 2	Conjunto de Treino: $D_2$	Conjunto de Teste: $D \setminus D_2$	$E_2 = 0.632 E_{\text{teste}} + 0.368 E_{\text{treino}}$
	...		
Experiência nº $k$	Conjunto de Treino: $D_k$	Conjunto de Teste: $D \setminus D_k$	$E_k = 0.632 E_{\text{teste}} + 0.368 E_{\text{treino}}$

A estimativa do erro verdadeiro é obtida  
como a média dos erros de cada  
experiência.

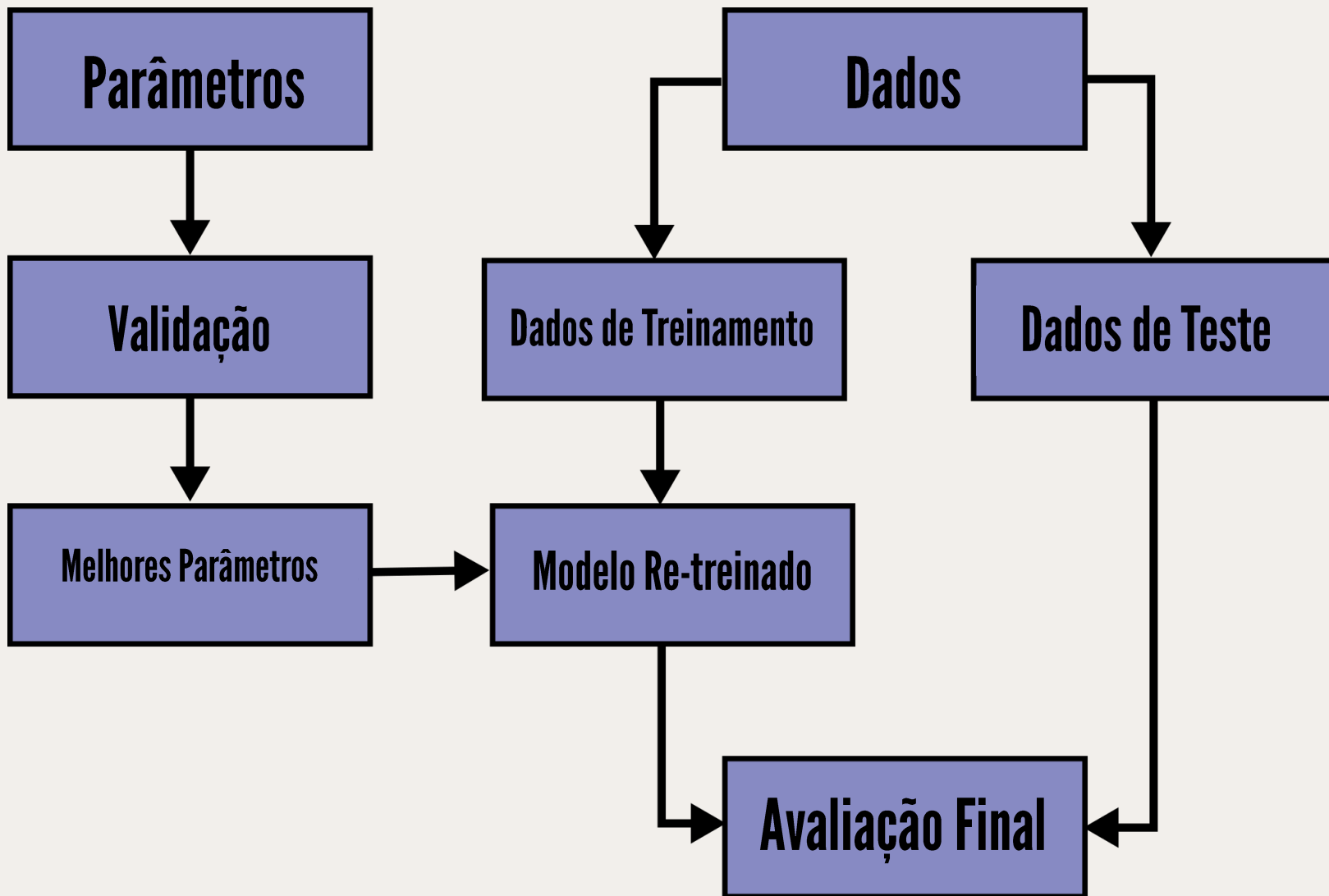
$$E = \frac{1}{k} \sum_{i=1}^k E_i$$

# AJUSTE DE PARÂMETROS

---

Na maioria dos modelos, torna-se necessário realizar um ajuste de parâmetros.

- Nesses casos, é necessário reservar uma parte dos exemplos para ajustar os parâmetros e outra parte para teste.



# AJUSTE DE PARÂMETROS

## HOLDOUT

Para um dado modelo

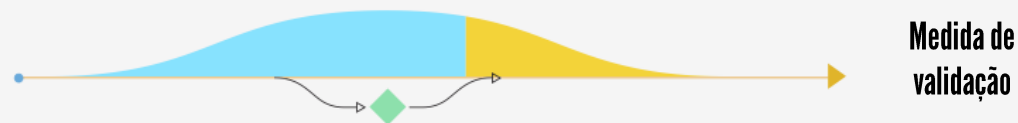
1

Divida os dados em treinamento/validação/teste



2

Para cada combinação de parâmetros



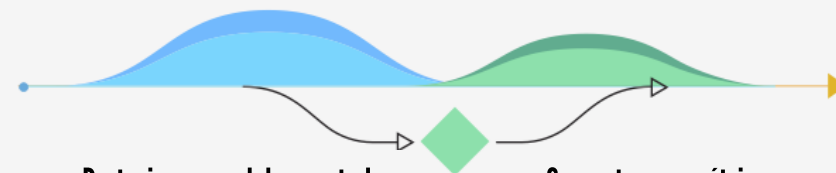
Treinar um modelo

Computar uma métrica na conjunto de validação

Medida de validação

3

Escolha a combinação de parâmetros com a melhor métrica



Re-treinar o modelo com todos os dados de treinamento

Computar um métrica no conjunto de teste

Medida de teste  
(pode comparar com outros modelos)

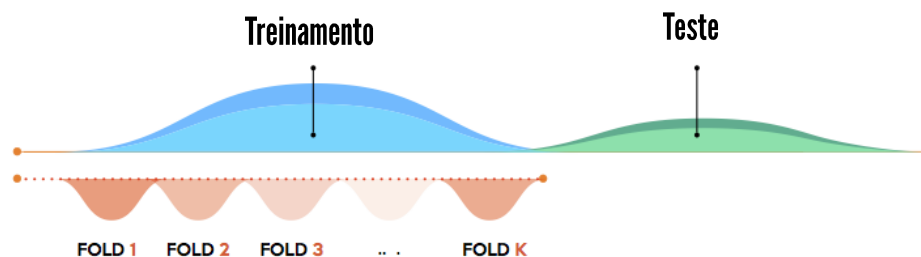
# AJUSTE DE PARÂMETROS

## VALIDAÇÃO CRUZADA

Para um dado modelo

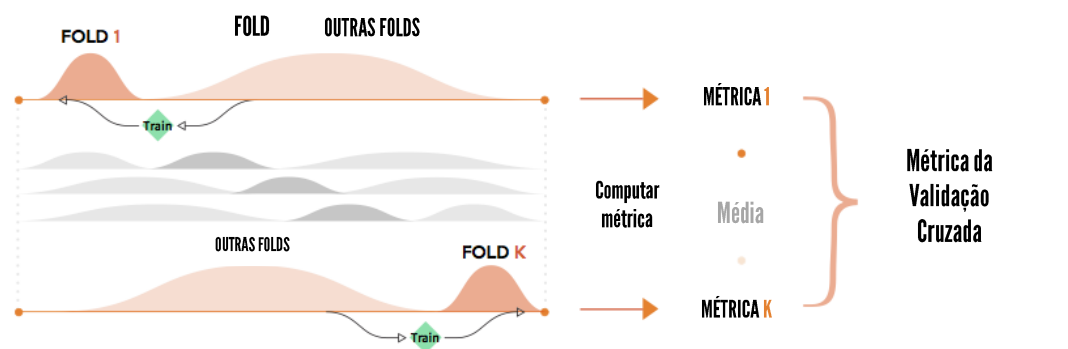
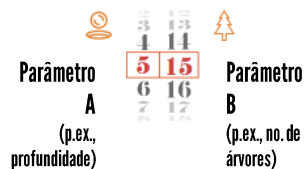
1

Separe o conjunto de teste e divida os dados de treinamento em  $k$  partições



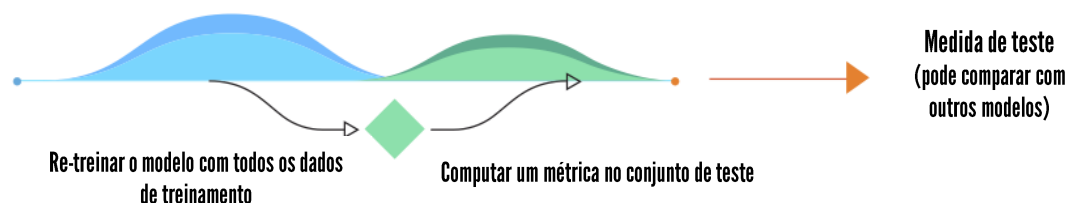
2

Para cada combinação de parâmetros



3

Escolha a combinação de parâmetros com a melhor métrica



# MÉTRICAS PARA REGRESSÃO

---

- Erro quadrático médio (MSE – *mean squared error*):

$$MSE(\hat{f}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i))^2$$

- Distância absoluta média (MAD – *mean absolute distance*):

$$MAD(\hat{f}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{f}(\mathbf{x}_i)|$$

- Para ambas as medidas, valores mais baixos correspondem a melhores modelos.

# MÉTRICAS PARA CLASSIFICAÇÃO

---

- Taxa de erro ou de classificações incorretas:

$$err(\hat{f}) = \frac{1}{n} \sum_{i=1}^n I(y_i \neq \hat{f}(x_i))$$

em que  $I(a) = 1$  se  $a$  é verdadeiro e é 0 em caso contrário.

- A taxa de erro varia entre 0 e 1, e valores próximos ao extremo 0 são melhores.
- O complemento dessa taxa corresponde à taxa de acerto ou acurácia do classificador:

$$ac(\hat{f}) = 1 - err(\hat{f})$$

- Valores próximos de 1 são considerados melhores.



# MÉTRICAS PARA CLASSIFICAÇÃO

Matriz de confusão:

		Verdadeira Classe	
		Positivo	Negativo
Classe Preditada	Positivo	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
	Negativo	Falsos Negativos (FN)	Verdadeiros Negativos (VN)

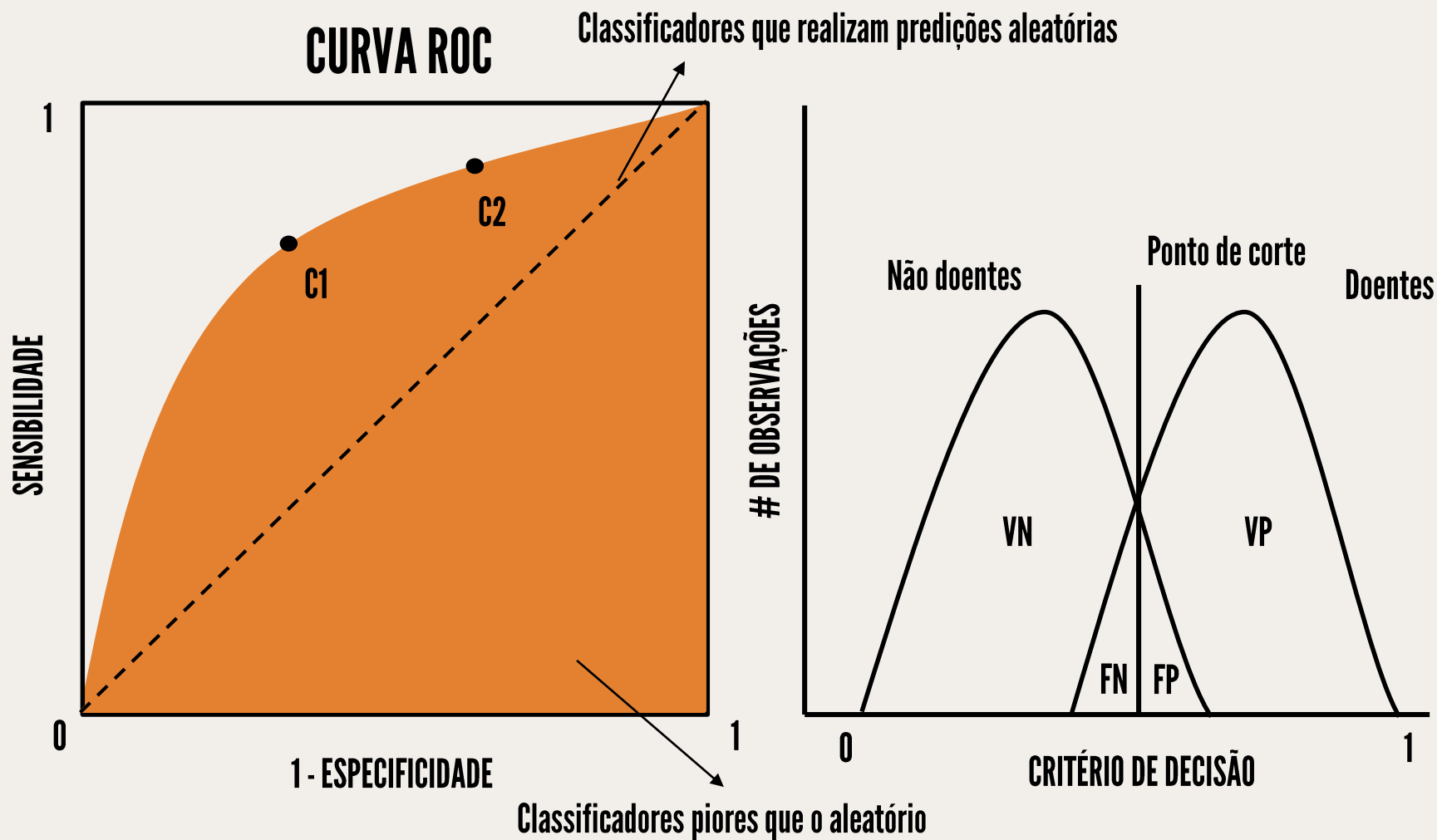
		Verdadeira Classe		
		A	B	C
Classe Preditada	A	4	1	0
	B	0	5	0
	C	0	2	8

# PROBLEMAS DE DUAS CLASSES

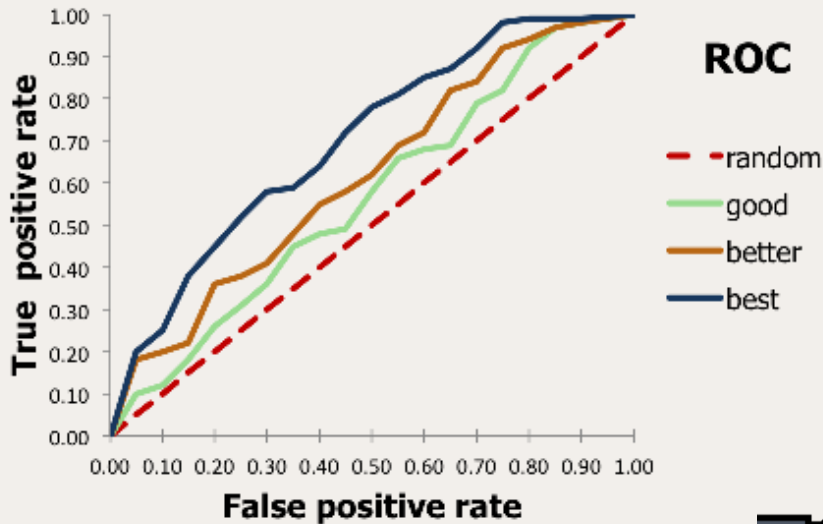
		Verdadeira Classe	
		Positivo	Negativo
Classe Predita	Positivo	Verdadeiros Positivos (VP)	Falsos Positivos (FP)
	Negativo	Falsos Negativos (FN)	Verdadeiros Negativos (VN)

- SENSIBILIDADE =  $VP/(VP+FN)$
- ESPECIFICIDADE =  $VN/(VN+FP)$
- VALOR PREDITIVO POSITIVO =  $VP/(VP+FP)$
- VALOR PREDITIVO NEGATIVO =  $VN/(VN+FN)$
- ACURÁCIA =  $(VP+VN)/(VP+VN+FP+FN)$
- MEDIDA-F =  $2X(S+VPP)/(S+VPP)$

# ANÁLISE ROC



# ANÁLISE ROC



AUC: área sob a curva ROC

- Maior AUC, classificador superior aos demais.
- Entretanto, melhor comparar estatisticamente.



Vantagem: medida única, boa para dados desbalanceados.



Desvantagem: limitada a problemas binários.

# ÍNDICE KAPPA

---

- Mede o grau de concordância entre duas diferentes técnicas além do que seria esperado pelo acaso.
- Calculado pela divisão da diferença entre a concordância esperada e a concordância observada e a diferença entre a concordância absoluta e a concordância esperada.

Kappa	Interpretação
<0	Discordantes
0 - 0,19	Concordância fraca
0,20 - 0,39	Concordância razoável
0,40 - 0,59	Concordância moderada
0,60 - 0,79	Concordância substancial
0,80 - 1,00	Concordância quase perfeita

# ÍNDICE KAPPA

---

- Acurácia observada:

$$((22 + 13) / 51) = 0,69$$

- Acurácia esperada:

$$((29 * 31 / 51) + (22 * 20 / 51)) / 51 = 0,51$$

- Kappa:

$$(0,69 - 0,51) / (1 - 0,51) = 0,37$$

	Referência		
Classificador		Classe A	Classe B
	Classe A	22	9
	Classe B	7	13

# TESTES ESTATÍSTICOS

---

- Comparar dois ou mais algoritmos na solução de um ou mais problemas práticos.
- Utilizar uma estratégia de amostragem, p. ex., validação cruzada *k-fold*.
- A cada iteração, todos os algoritmos usam a mesma partição de treinamento e de teste para obter seus resultados e, dessa forma, a média de desempenho obtida por todos é calculada sobre os mesmos objetos.
- Mais recomendado: teste de hipóteses para a comparação dos desempenhos dos modelos em investigação.

LEITURA RECOMENDADA: DEMŠAR, Janez. Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research, v. 7, n. Jan, p. 1-30, 2006.

# TESTES ESTATÍSTICOS

---

Testes comumente utilizados:

- Wilcoxon signed-rank;
- Friedman.

Pareados e não paramétricos.

Baseados em ranqueamentos, permitem comparar outras medidas de desempenho além das tradicionais, estabelecidas em algum tipo de erro/acerto preditivo.



# COMPARANDO DOIS MODELOS

---

Teste recomendado: *Wilcoxon signed-rank*

1. Calculam-se inicialmente as diferenças nas medidas de desempenho dos algoritmos.
2. Os valores absolutos dessas diferenças são ranqueados (menores diferenças assumem primeiras posições e assim sucessivamente).
  - No caso de empates, atribuem-se valores médios das posições de ordenação.
3. Pelo teste, comparam-se as posições das diferenças positivas e negativas entre os algoritmos.
  - $A \text{ e } B \rightarrow B - A$ : maiores valores são melhores (taxa de acerto, AUC, precisão...), então +, melhor desempenho de B, -, melhor desempenho de A.

# COMPARANDO DOIS MODELOS

MODELO A	MODELO B	DIF (B-A)	DIF_ABS	POSIÇÃO
77,98	77,91	-0,07	0,07	3
72,26	72,27	0,01	0,01	1
76,95	76,97	0,02	0,02	2
77,94	76,57	-1,37	1,37	5
72,23	71,63	-0,60	0,60	4
76,90	75,48	-1,42	1,42	6
77,93	75,75	-2,18	2,18	7

Seja  $R^+$  a soma das posições (*rank*s) de conjuntos de dados em que o algoritmo B é melhor que o algoritmo A e  $R^-$  a soma de posições oposta. As posições das diferenças nulas são repartidas igualmente entre as duas somas. Se há um número ímpar de diferenças nulas, uma é ignorada.

$$Z = \frac{S - \frac{1}{4}N(N-1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}$$

S: menor das somas.

$$R^+ = \sum_{d_i > 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i)$$

$$R^- = \sum_{d_i < 0} \text{rank}(d_i) + \frac{1}{2} \sum_{d_i = 0} \text{rank}(d_i)$$

# COMPARANDO MAIS MODELOS

---

Teste recomendado: teste de Friedman

- Também é baseado na comparação de ranqueamentos de desempenhos.
- O valor absoluto da medida de desempenho de cada algoritmo individualmente em cada conjunto de dados é considerado para realizar o ranqueamento.
- Para cada conjunto de dados, realiza-se o ranqueamento dos algoritmos de acordo com seu desempenho (dos melhores para os piores).
  - Em caso de empates, valores médios de posição são atribuídos.

# COMPARANDO MAIS MODELOS

---

MODELO A	MODELO B	MODELO C
77,98	77,91	75,89
72,26	72,27	74,56
76,95	76,97	77,32
77,94	76,57	76,98
72,23	71,63	72,65
76,90	75,48	75,32
77,93	75,75	76,90

MODELO A	MODELO B	MODELO C
1	2	3
3	2	1
3	2	1
1	3	2
2	3	1
1	2	3
1	3	2
1,71	2,43	1,86

# COMPARANDO MAIS MODELOS

---

- Seja  $r_j^i$  a posição do desempenho do algoritmo  $j$  (dentre  $A$  algoritmos) no conjunto de dados  $i$  (dentre  $N$  conjuntos de dados).
- O teste de Friedman irá comparar os ranqueamentos médios  $R_j$  dos diferentes algoritmos.
- A  $H_0$  afirma que todos os algoritmos são equivalentes e que suas posições de ranqueamento são iguais.

$$F_F = \frac{(N - 1)\chi_F^2}{N(A - 1)\chi_F^2}, \quad \chi_F^2 = \frac{12N}{A(A + 1)} \left[ \sum_j R_j^2 - \frac{A(A + 1)^2}{4} \right]$$

Se a estatística calculada for maior que  $F_{A-1,(A-1)(N-1)}$ , rejeita-se a  $H_0$  de que todos os algoritmos têm o mesmo desempenho.

# COMPARANDO MAIS MODELOS

---

- Se  $H_0$  for rejeitada, precisamos descobrir quais algoritmos possuem diferença de desempenho → pós-teste.
- No pós-teste, o desempenho de dois algoritmos em particular é estatisticamente diferente caso a diferença entre os seus valores médios de posição no ranqueamento seja maior ou igual ao valor da diferença crítica  $CD$  (*Critical Difference*):

$$CD = q_{\alpha} \sqrt{\frac{A(A + 1)}{6N}}$$

# COMPARANDO MAIS MODELOS

---

- Se todos os algoritmos estão sendo comparados entre si em pares, os valores de  $q_\alpha$  podem ser fornecidos pela estatística de Nemenyi.
  - Se a diferença entre os ranqueamentos médios de dois algoritmos é maior que CD, então a  $H_0$  de que os algoritmos têm o mesmo desempenho é rejeitada.
- Quando a comparação é de vários algoritmos em relação a um único algoritmo (p. ex., o desempenho de várias modificações de um algoritmo é comparado ao do algoritmo base),  $q_\alpha$  pode ser menos restritivo, pois menos comparações são realizadas.
  - A estatística de Bonferroni-Dunn pode ser empregada.

# COMPARANDO MAIS MODELOS

#classifiers	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.343	2.569	2.728	2.850	2.949	3.031	3.102	3.164
$q_{0.10}$	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

(a) Critical values for the two-tailed Nemenyi test

#classifiers	2	3	4	5	6	7	8	9	10
$q_{0.05}$	1.960	2.241	2.394	2.498	2.576	2.638	2.690	2.724	2.773
$q_{0.10}$	1.645	1.960	2.128	2.241	2.326	2.394	2.450	2.498	2.539

(b) Critical values for the two-tailed Bonferroni-Dunn test; the number of classifiers include the control classifier.



# COMPARANDO MAIS MODELOS

---

Teste de Holm: o modelo com a melhor colocação é selecionado como modelo-controle e uma comparação dois a dois dos demais modelos com o controle é realizada. A hipótese nula afirma que o modelo-controle e o segundo utilizado na comparação pareada possuem a mesma colocação média.

1. Os valores  $p$  das comparações são ordenados de forma crescente com suas hipóteses associadas. Para um dado nível de significância  $\alpha$ , tome  $k$  como índice mínimo tal que  $p_{(k)} > \frac{\alpha}{m+1-k}$ , sendo  $m$  o número de comparações.
2. Conforme tomamos um novo valor  $p$ , incrementamos o valor de  $k$ .
3. Se  $p < p_{(k)}$ , então rejeitamos a hipótese. Quando alguma hipótese não é rejeitada, o método para e as demais hipóteses não são verificadas (já considera não rejeição).

# COMPARANDO MAIS MODELOS

$$\left. \begin{array}{l} H_1 \rightarrow p_1 = 0,01 \\ H_2 \rightarrow p_2 = 0,04 \\ H_3 \rightarrow p_3 = 0,03 \\ H_4 \rightarrow p_4 = 0,005 \end{array} \right\}$$

Valores p não ajustados  
 $\alpha = 0,05$

$$p_{(k)} > \frac{\alpha}{m + 1 - k}$$

Se  $p < p_{(k)}$ ,  
então rejeitamos  
a hipótese.

$$\begin{array}{l} H_4 = H_{(1)} \\ p_4 = p_{(1)} = 0,005 \\ p_{(1)} = \frac{0,05}{4 + 1 - 1} = 0,0125 \rightarrow \text{Rejeitamos} \end{array}$$

$$\begin{array}{l} H_1 = H_{(2)} \\ p_1 = p_{(2)} = 0,01 \\ p_{(2)} = \frac{0,05}{4 + 1 - 2} = 0,0167 \rightarrow \text{Rejeitamos} \end{array}$$

$$\begin{array}{l} H_3 = H_{(3)} \\ p_3 = p_{(3)} = 0,03 \\ p_{(2)} = \frac{0,05}{4 + 1 - 3} = 0,025 \rightarrow \text{Não rejeitamos} \end{array}$$

$H_2 \rightarrow \text{Não rejeitamos}$