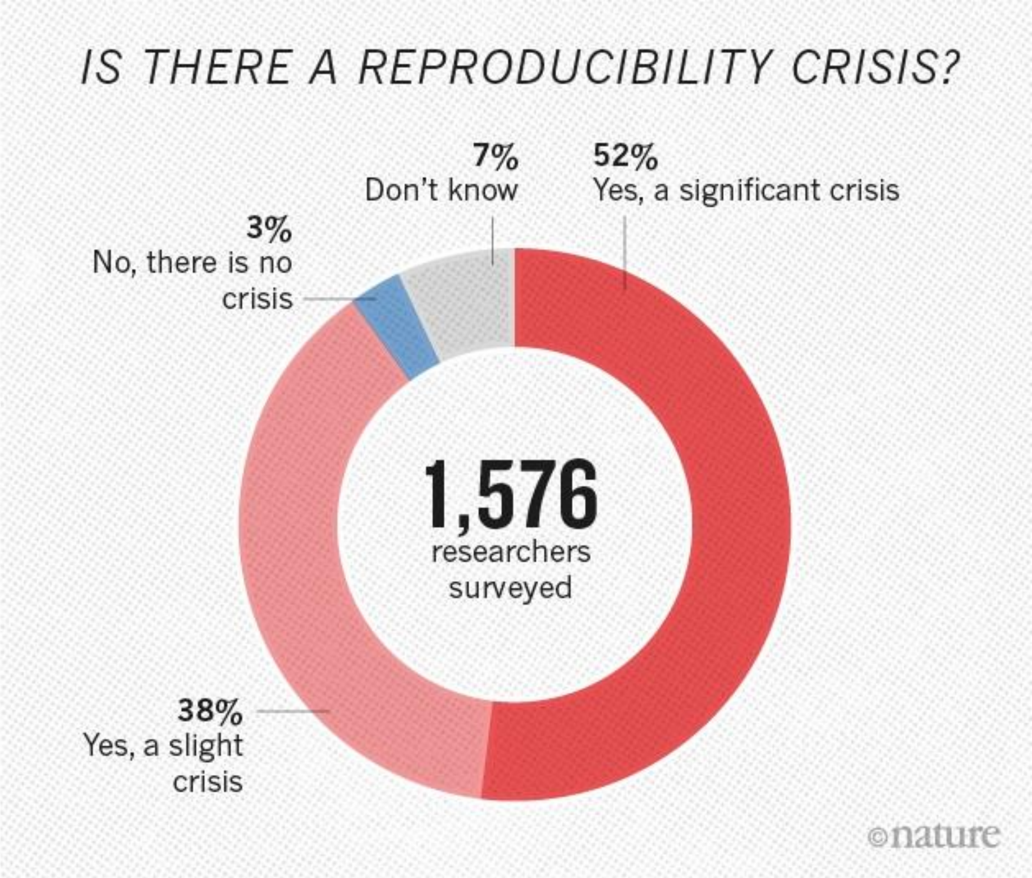
Avaliação de modelos, seleção de modelos e seleção de algoritmos

Cleber Zanchettin
UFPE - Universidade Federal de Pernambuco
CIn - Centro de Informática

Avaliação dos Classificadores

- Existem poucos estudos analíticos sobre o comportamento de algoritmos de aprendizagem
- A análise de classificadores é fundamentalmente experimental
- Dimensões de análise:
  - Taxa de erro
  - Complexidade dos modelos
  - Tempo de aprendizagem
  - …



- Pesquisadores da Bayer só conseguem reproduzir 25% dos paper examinados
- Percentual parecido nos papers de ML em uma pesquisa do MIT

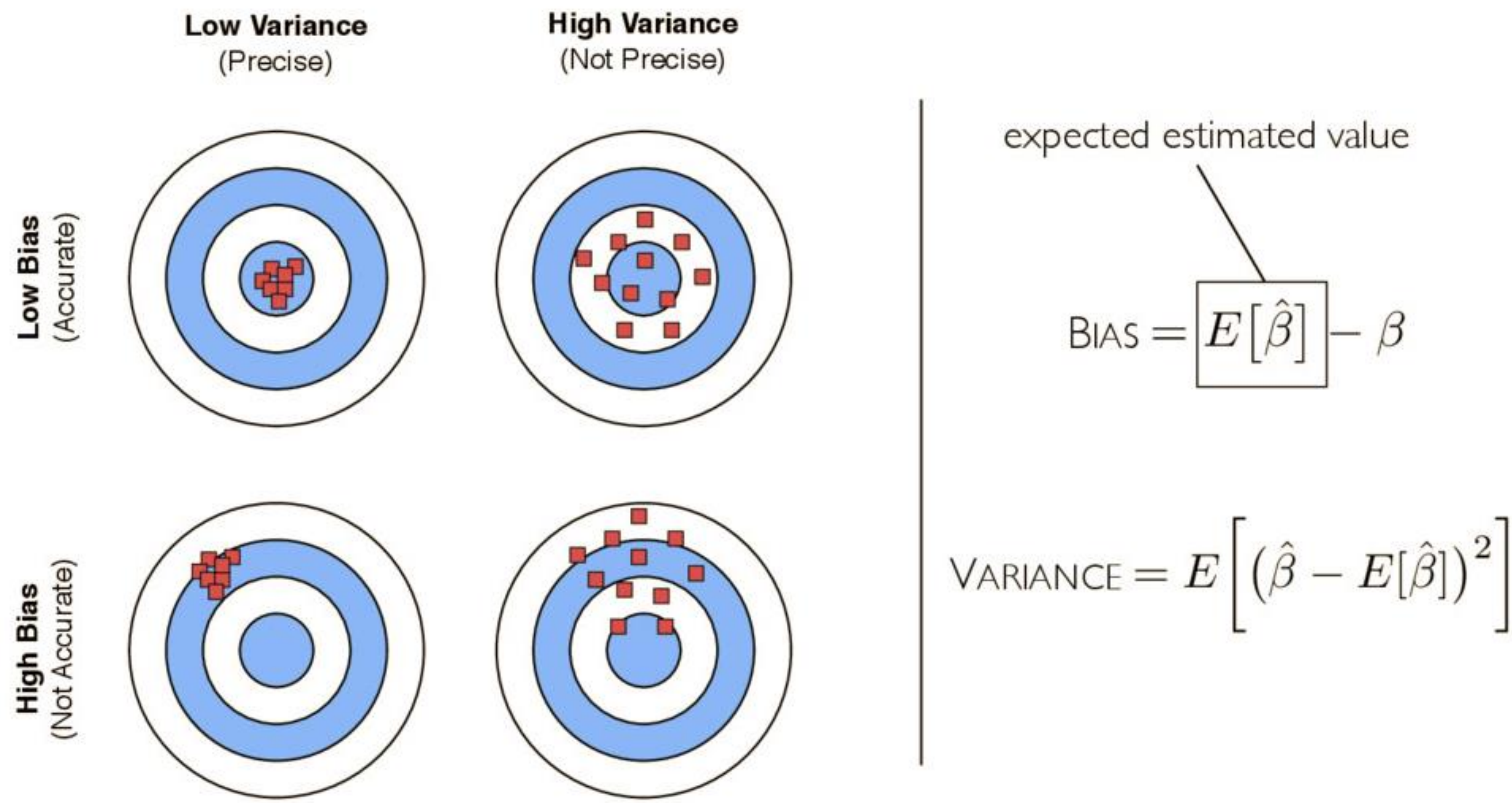
1. M. Baker, "1,500 scientists lift the lid on reproducibility," Nature, no. 533, pp. 452-454, 2016.
2. Florian Prinz, Thomas Schlange, Khusru Abdallah, "Believe it or not: how much can we rely on published data on potential drug targets," Nature Reviews, no. 712.
3. Open Scien. Collaboration, "Estimating the reproducibility of psychological science," Science, vol. 349, no. 6251, 2015.
4. M. Baker, "Over half of psychology studies fail reproducibility test," Nature, 2015.

Avaliar a performance do modelo

- Estimar o desempenho de **generalização** em dados futuros (não vistos)

- **Aumentar** o desempenho ajustando o algoritmo de aprendizagem e **selecionar** o modelo de melhor desempenho a partir de um determinado espaço de hipóteses

- **Identificar** o algoritmo de ML mais adequado
  - Comparar diferentes algoritmos
  - Selecionando o melhor desempenho

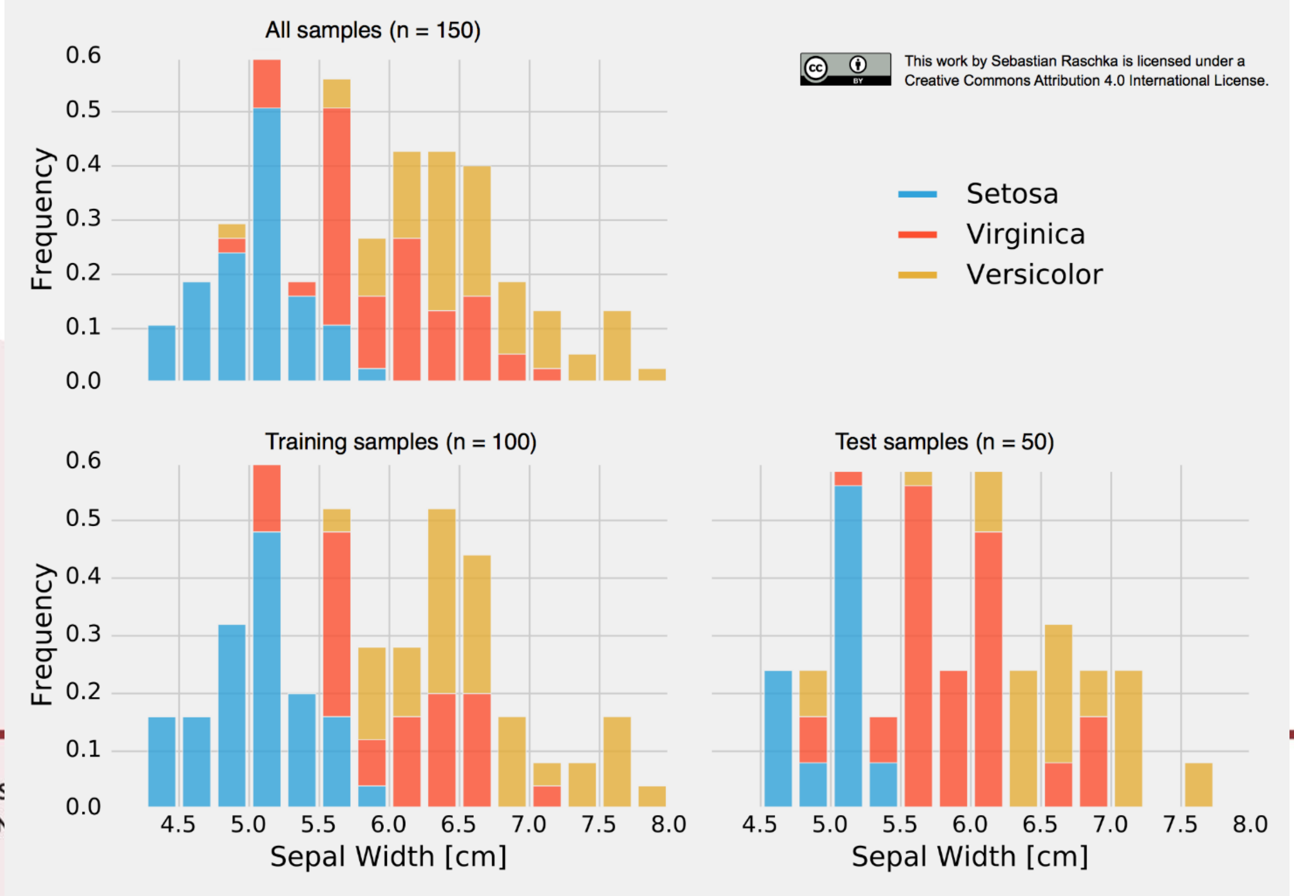


https://sebastianraschka.com/pdf/manuscripts/model-eval.pdf

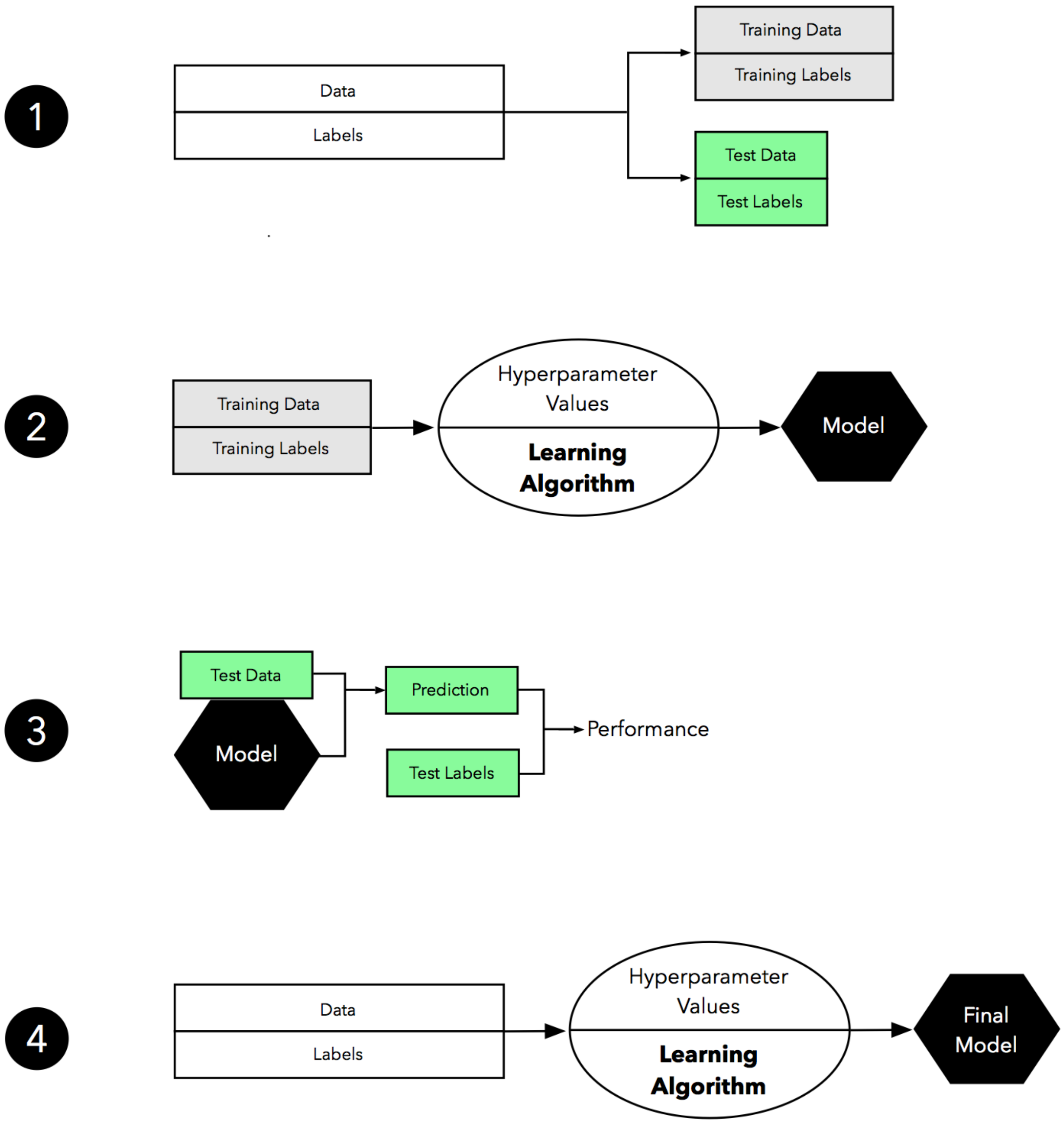
Avaliação do modelo

Amostragem

- Independência estatística
  - Média, proporção e variância
- Estratificação

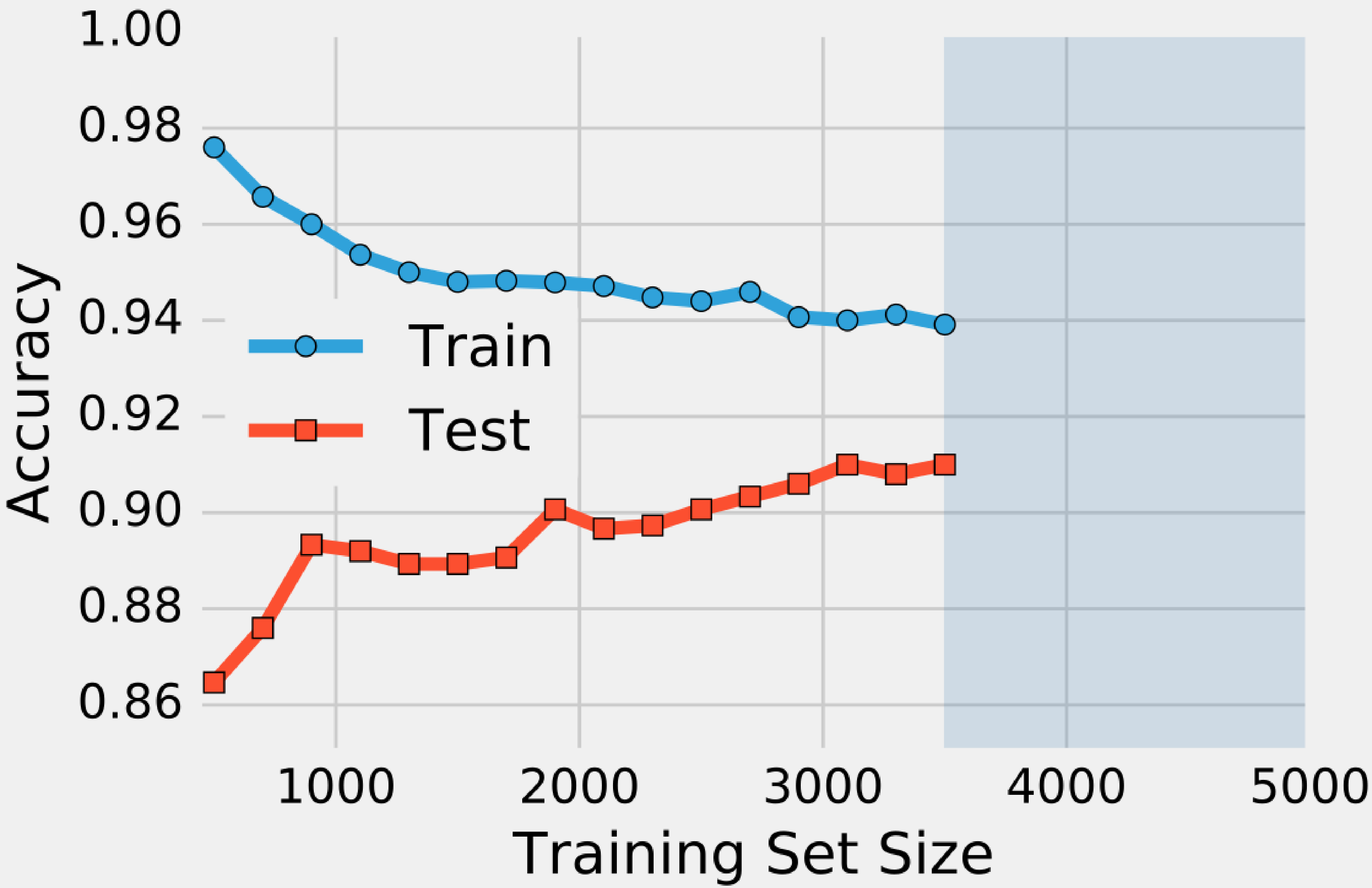


Holdout

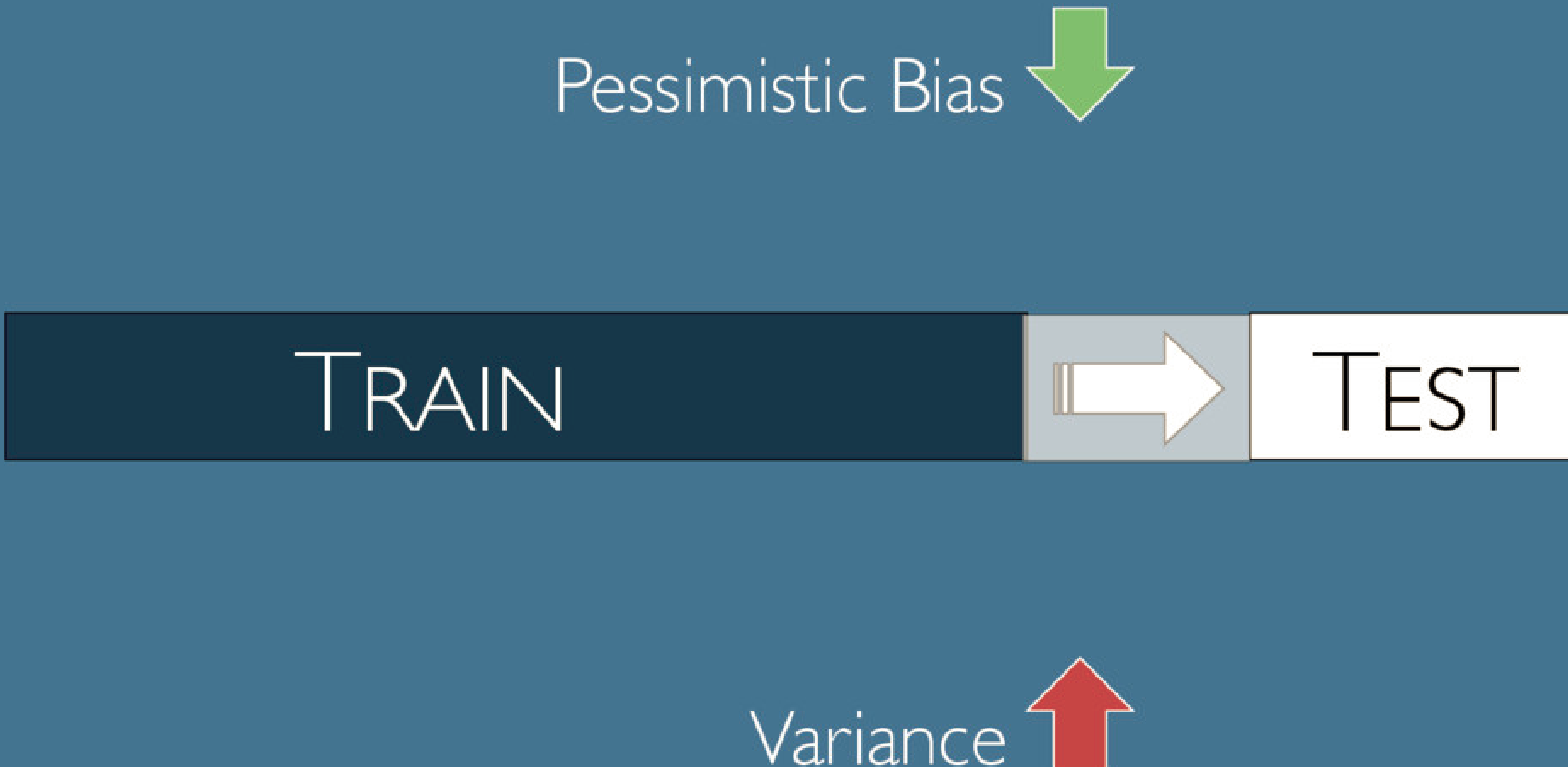


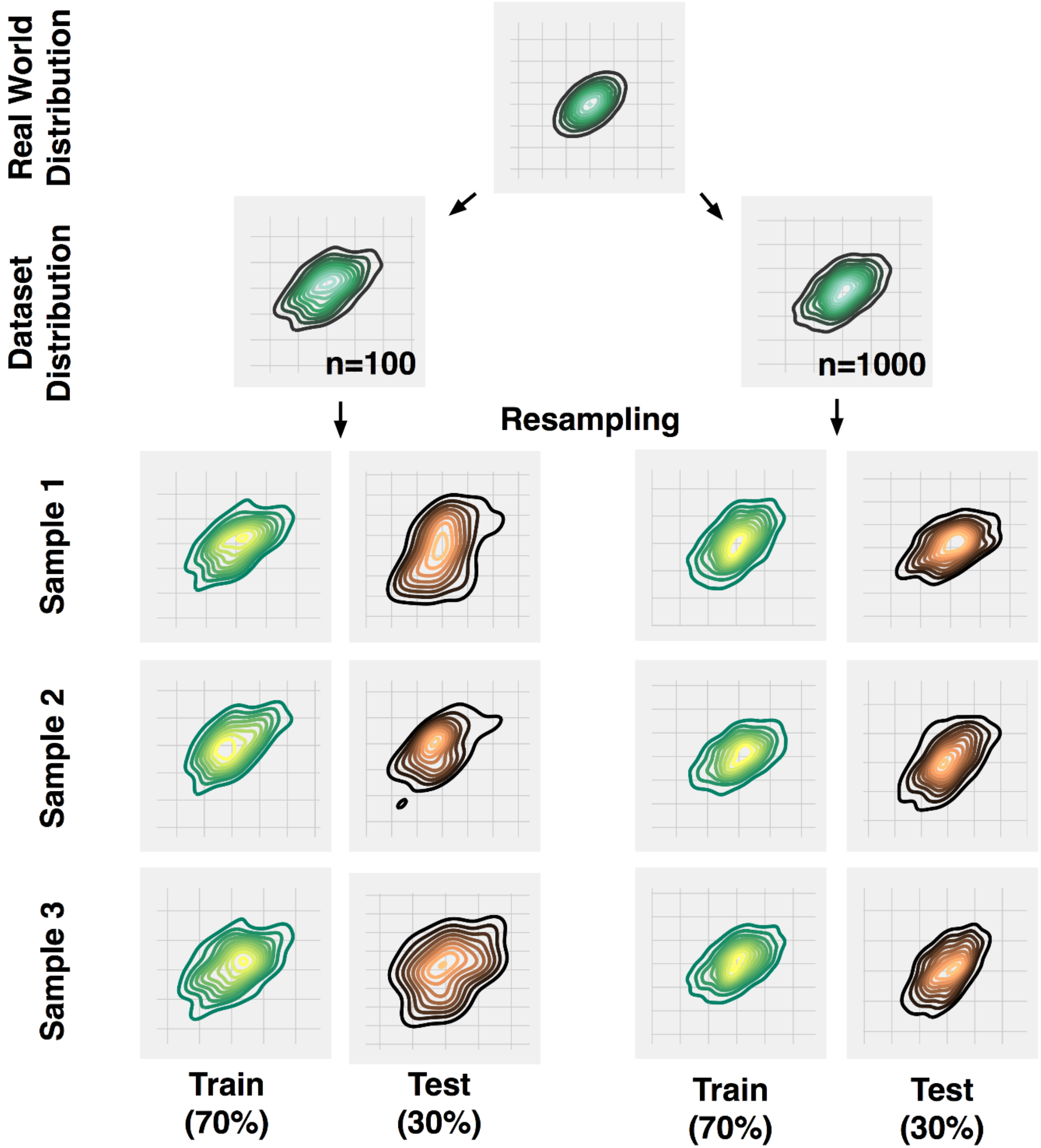
1
Data
Labels
Training Data
Training Labels
Test Data
Test Labels
2
Training Data
Training Labels
Hyperparameter Values
Learning Algorithm
Model
3
Test Data
Model
Prediction
Test Labels
Performance
4
Data
Labels
Hyperparameter Values
Learning Algorithm
Final Model


Pessimistic Bias
TRAIN
TEST

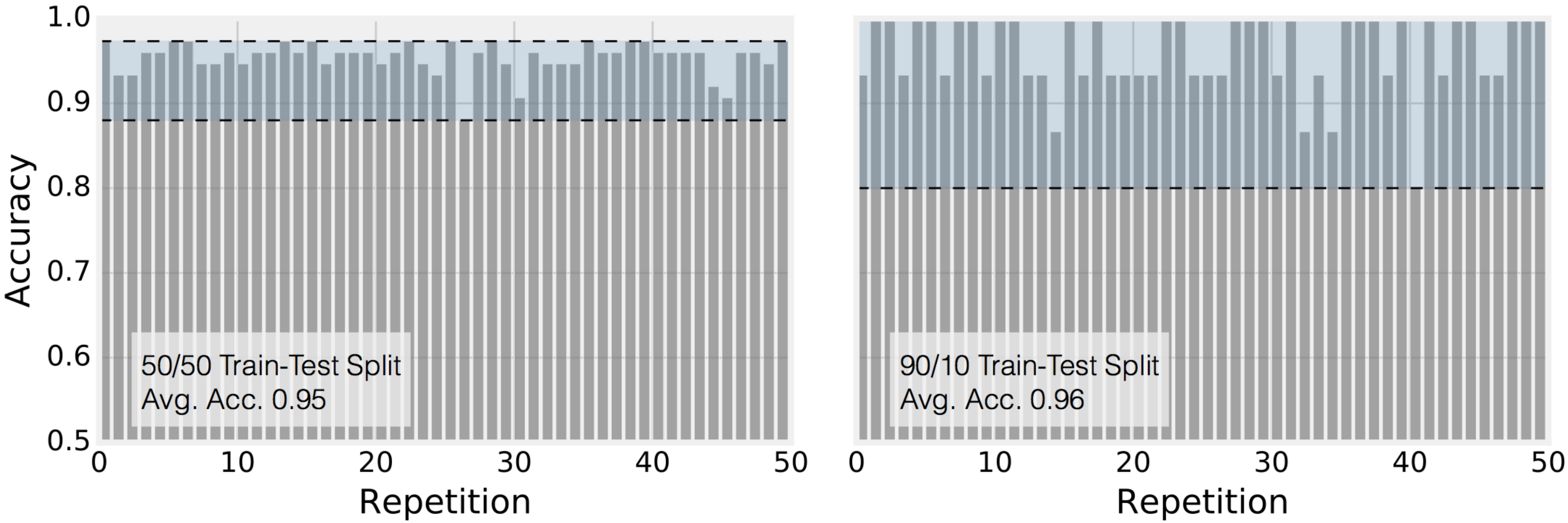


MNIST, 500 exemplos estratificados das 9 classes
3.500 – treinamento (variando)
1.500 – teste (fixo)


Pessimistic Bias
TRAIN
TEST
Variance


Real World Distribution
Dataset Distribution
n=100
n=1000
Resampling
Sample 1
Sample 2
Sample 3
Train (70%)
Test (30%)
Train (70%)
Test (30%)

Repeated Holdout Validation



Random Subsampling

Este estimador **é significativamente melhor** que o obtido com holdout

Executa **K experiências, uma por cada partição** sobre o conjunto de dados:

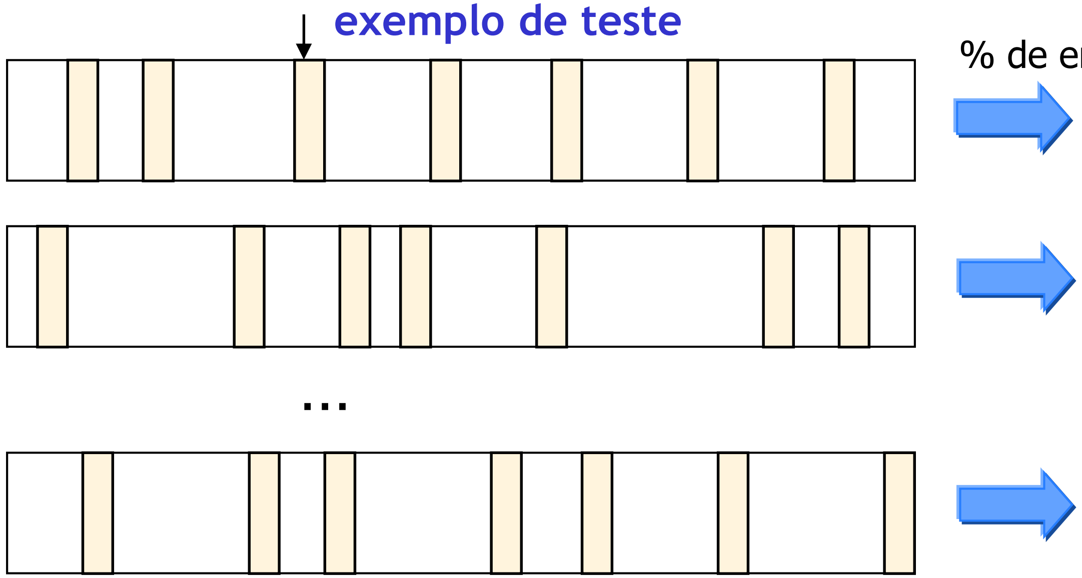
✓ em cada partição é <u>seleccionado aleatoriamente um nº (fixo) de exemplos</u> de teste

✓ o classificador é induzido nos exemplos de treinamento e avaliado no teste

Taxa de Erro

% de erros no conjunto de teste

exemplo de teste



Experiência nº 1 usa 1ª partição → E_1

Experiência nº 2 usa 2ª partição → E_2

...

Experiência nº K usa kª partição → E_k

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

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

Bootstrapping ou Bagging

Método de estimação baseado em re-amostragem com reposição (*sampling with replacement*) usado quando dispormos de poucos exemplos

- ✓ Dado um conjunto de dados **D** com **N exemplos** é gerado um número **B de amostras (bootstraps)** de tamanho **N:**
  - ✓ cada amostra é gerada usando **amostragem com reposição**

    cada vez que um exemplo é adicionado aleatoriamente este é logo **reposto**

    ⇒ alguns exemplos podem aparecer mais do que uma vez, enquanto outros podem nunca aparecer
- ✓ Em cada experiência (*trial*): **(no total são efetuadas B experiências)**
  - ✓ uma amostra bootstrap é gerada e usada como **conjunto de treino**
  - ✓ os exemplos do conjunto **D** que não pertencem à amostra são usados como **conjunto de teste,** e é obtida uma estimativa da taxa de erro
- ✓ A **estimativa da taxa de erro** é a **média das taxa de erros** obtidas por cada uma das amostras

Bootstrap

- Efron, Bradley, and Robert Tibshirani. 1994. *An Introduction to the Bootstrap*. Chapman & Hall.
- Efron, Bradley, and Robert Tibshirani. 1997. “Improvements on Cross-Validation: The .632+ Bootstrap Method.” *Journal of the American Statistical Association* 92 (438): 548. doi:10.2307/2965703

Bootstrap 0.632

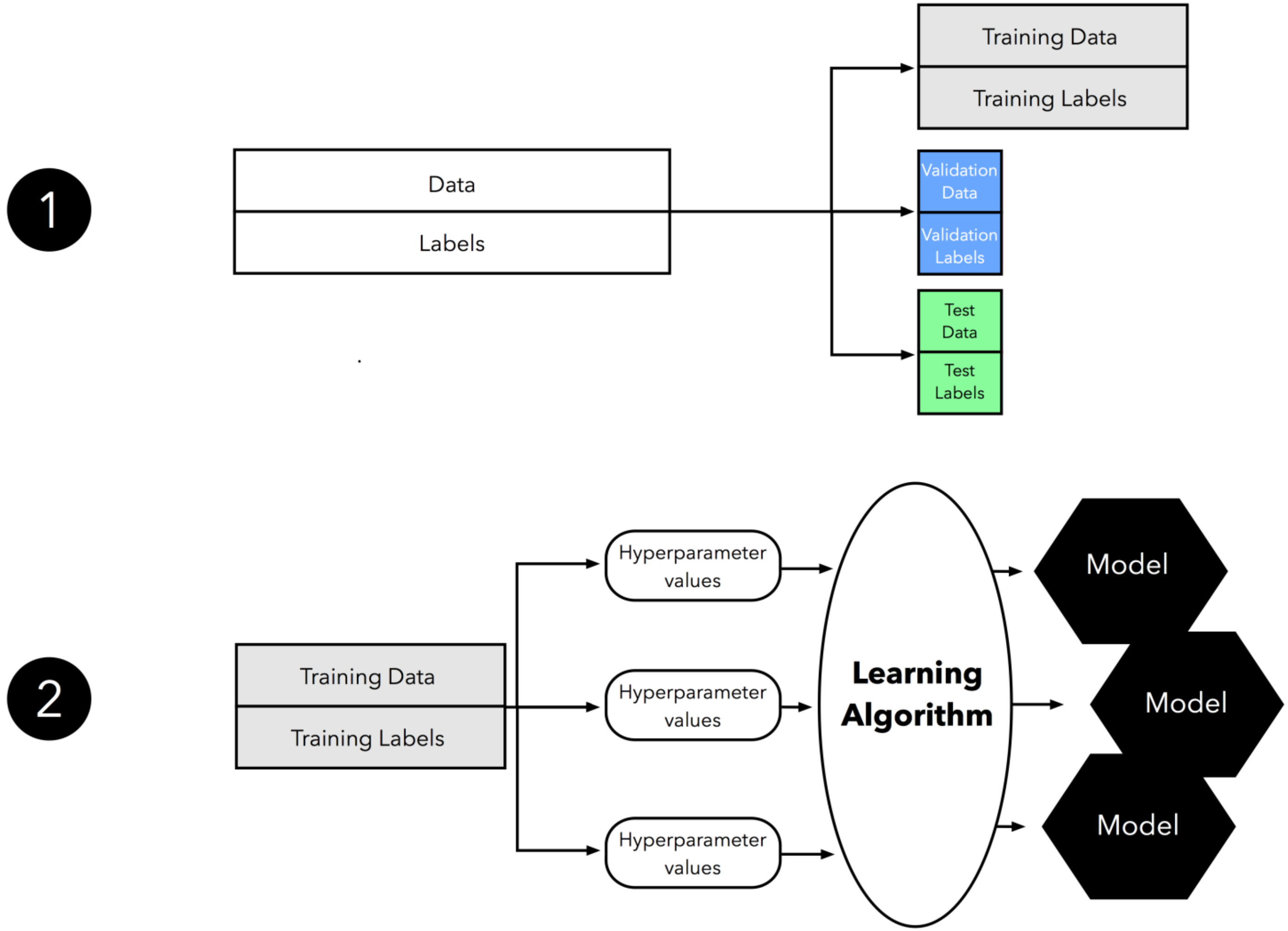
Um exemplo tem uma probabilidade de **1-1/N** de não ser selecionado
⇒ a probabilidade que fique no conjunto de teste é de $(1-1/N)^N \approx 1- e^{-1} = 0.368$
⇒ o conjunto de treino vai conter aproximadamente **63.2% dos exemplos** de **D**
⇒ a taxa de erro E_{teste} é um estimador **muito pessimista** (usa 36.8% dos exemplos)
⇒ solução: usar também a taxa do erro E_{treino} obtida no conjunto de treino

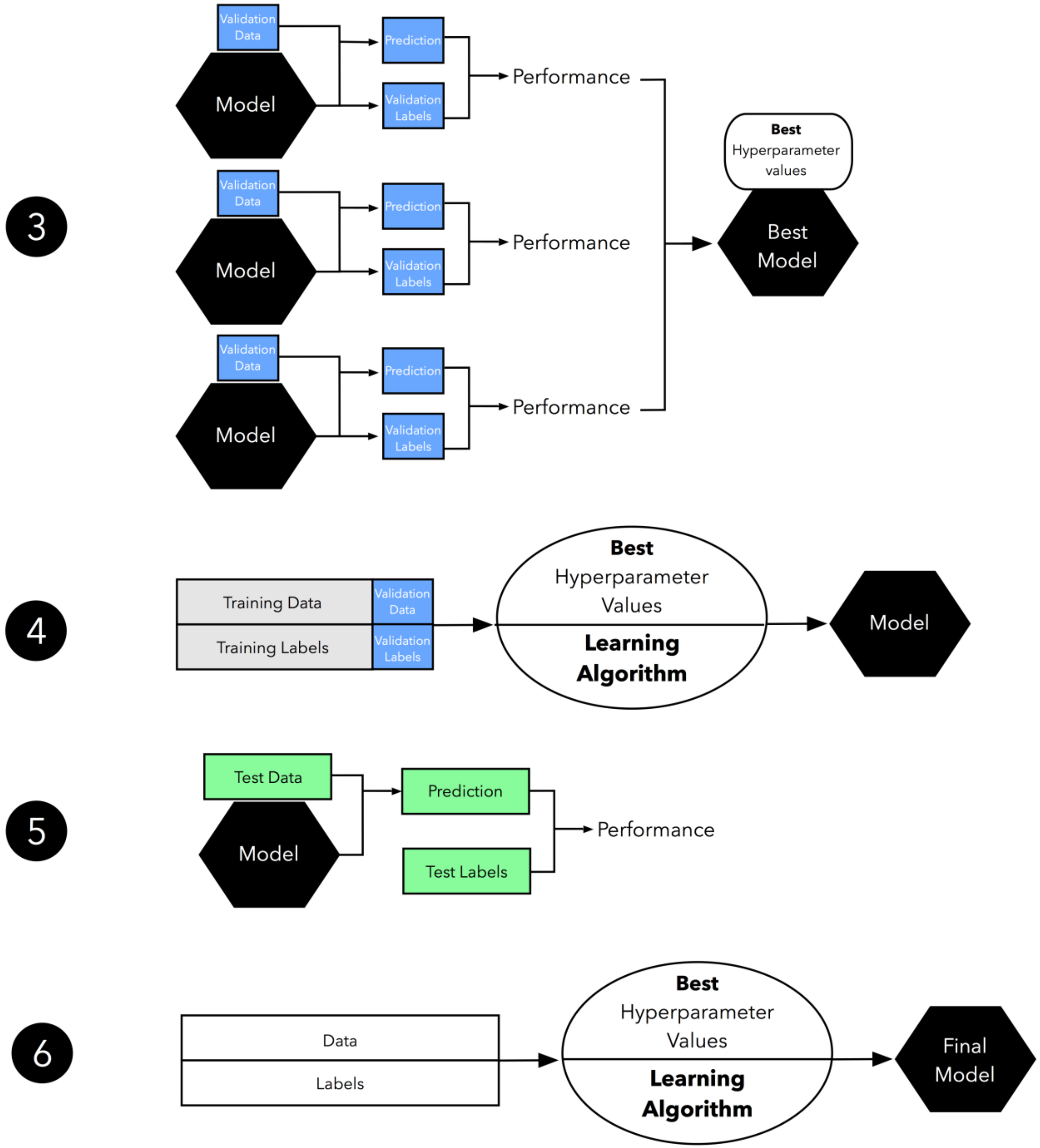
	≈ 63.2 % dos exemplos de D	≈ 37 % de D	Estimativas da Taxa de Erro
Experiência nº 1 1º bootstrap sample	Conjunto de Treino: D_1	Conjunto de Teste: $D \setminus D_1$	$E_1 = 0.632\, E_{teste} + 0.368\, E_{treino}$
Experiência nº 2 2º bootstrap sample	Conjunto de Treino: D_2	Conjunto de Teste: $D \setminus D_2$	$E_2 = 0.632\, E_{teste} + 0.368\, E_{treino}$
	…, etc.		
Experiência nº B Bº bootstrap sample	Conjunto de Treino: D_B	Conjunto de Teste: $D \setminus D_B$	$E_B = 0.632\, E_{teste} + 0.368\, E_{treino}$

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

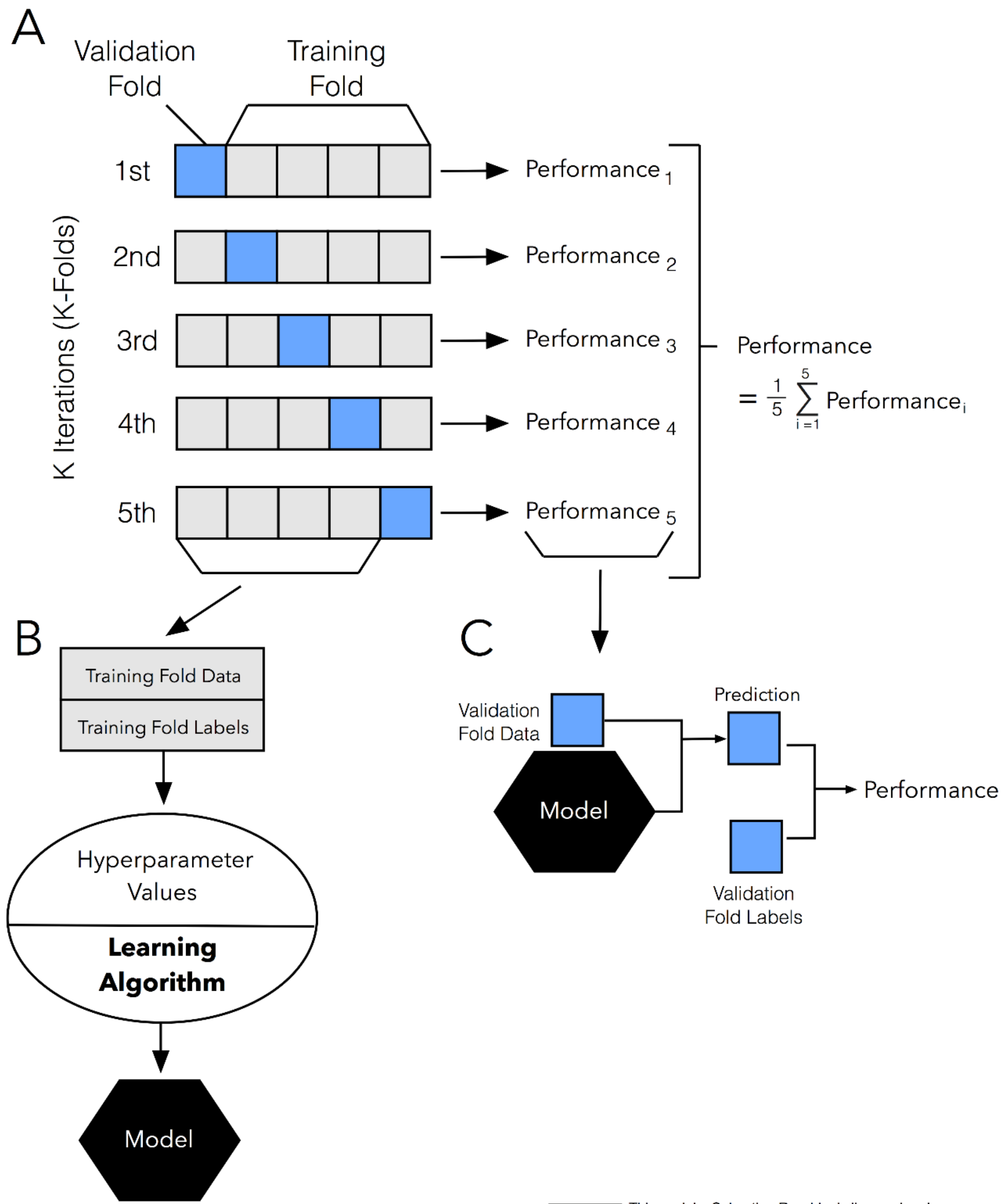
$$\boxed{E = \frac{1}{B}\sum_{i=1}^{B} E_i}$$

Three-Way Holdout Method



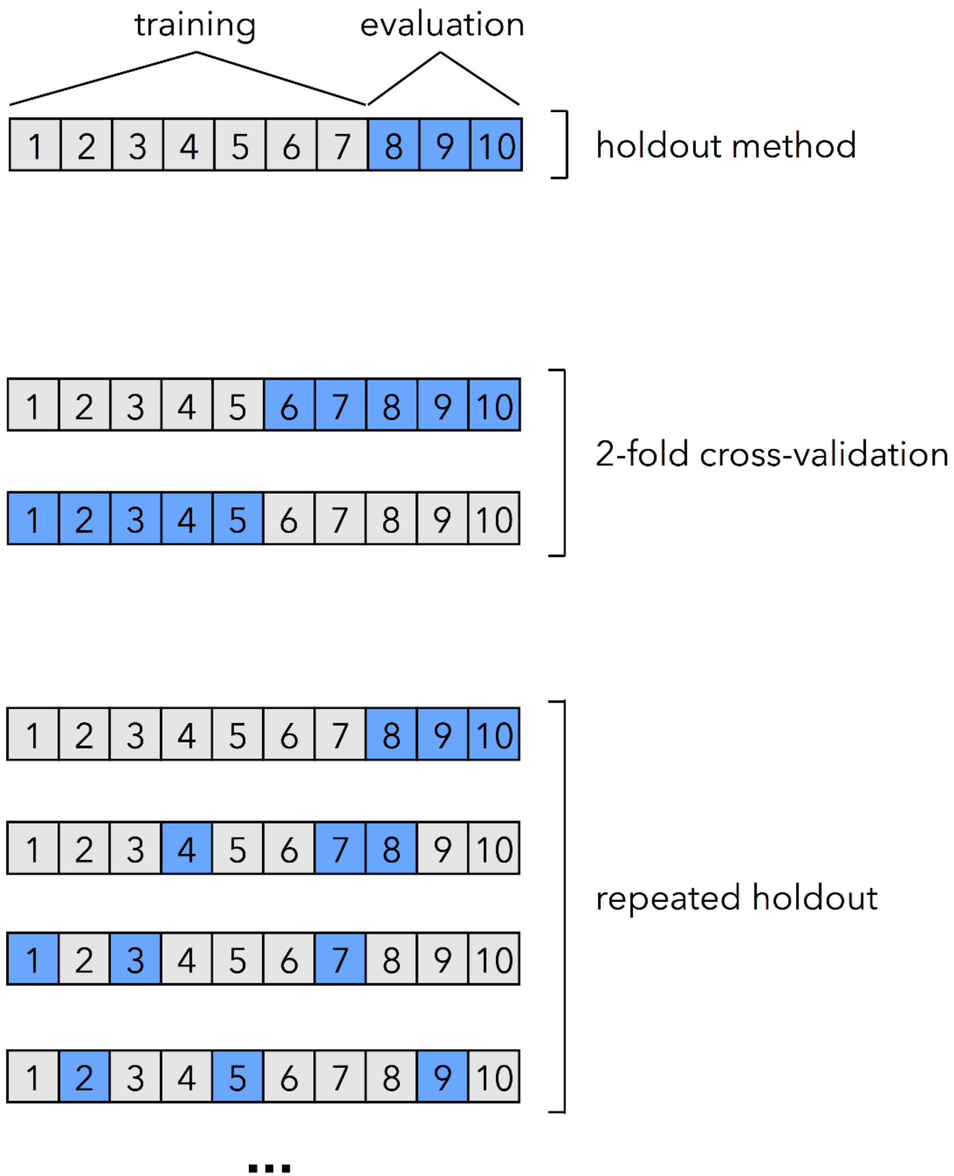

3
Validation Data
Model
Prediction
Validation Labels
Performance
Best Hyperparameter values
Best Model
4
Training Data
Training Labels
Validation Data
Validation Labels
Best Hyperparameter Values
Learning Algorithm
Model
5
Test Data
Model
Prediction
Test Labels
Performance
6
Data
Labels
Best Hyperparameter Values
Learning Algorithm
Final Model

K-Fold Cross-Validation



- $k=?$

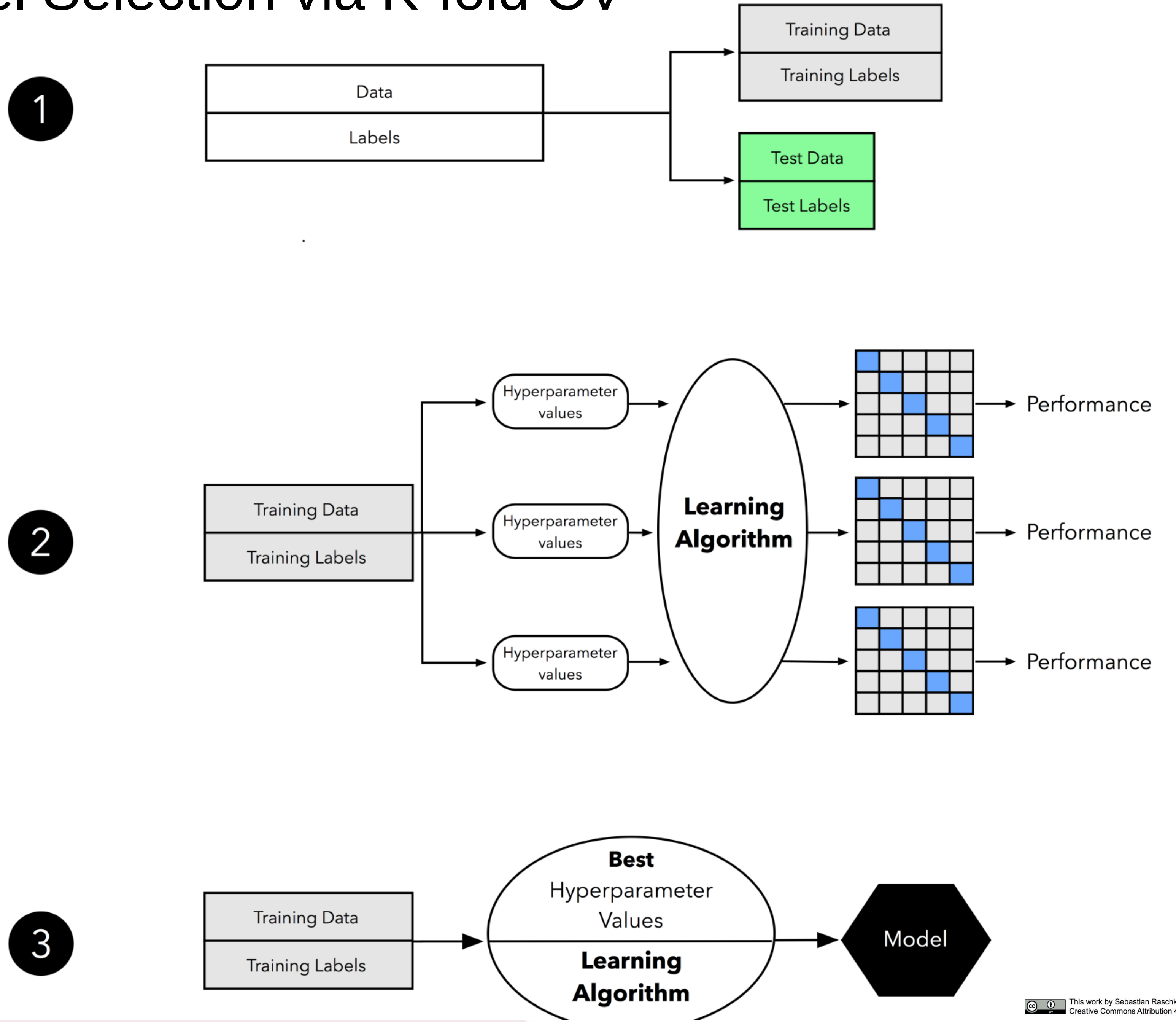
- $k=2$
- $k=n$

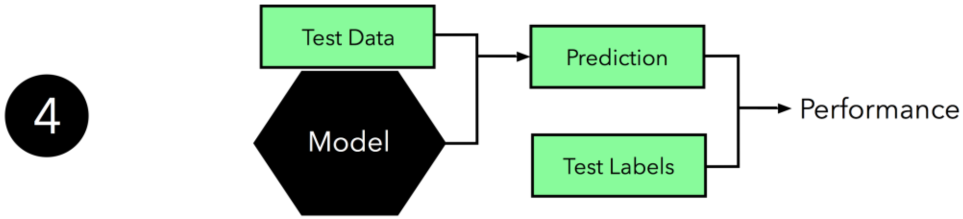


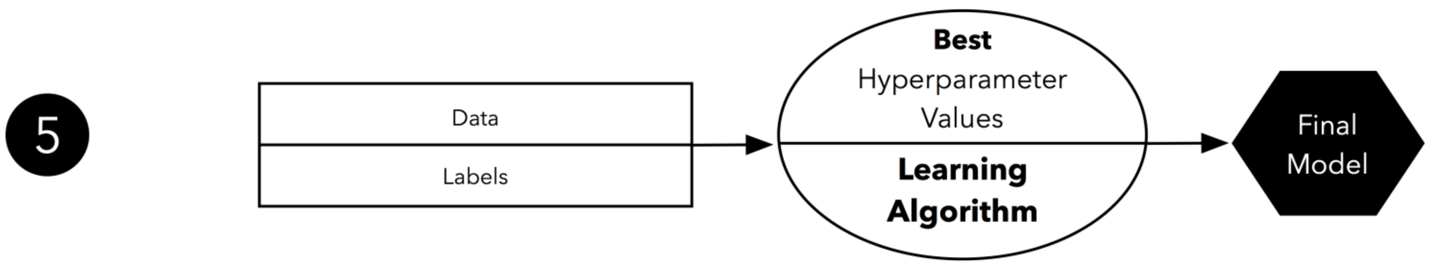
training
evaluation
1 2 3 4 5 6 7 8 9 10
holdout method
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
2-fold cross-validation
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10
repeated holdout
...

Leave-one-out cross-validation

Model Selection via K-fold CV

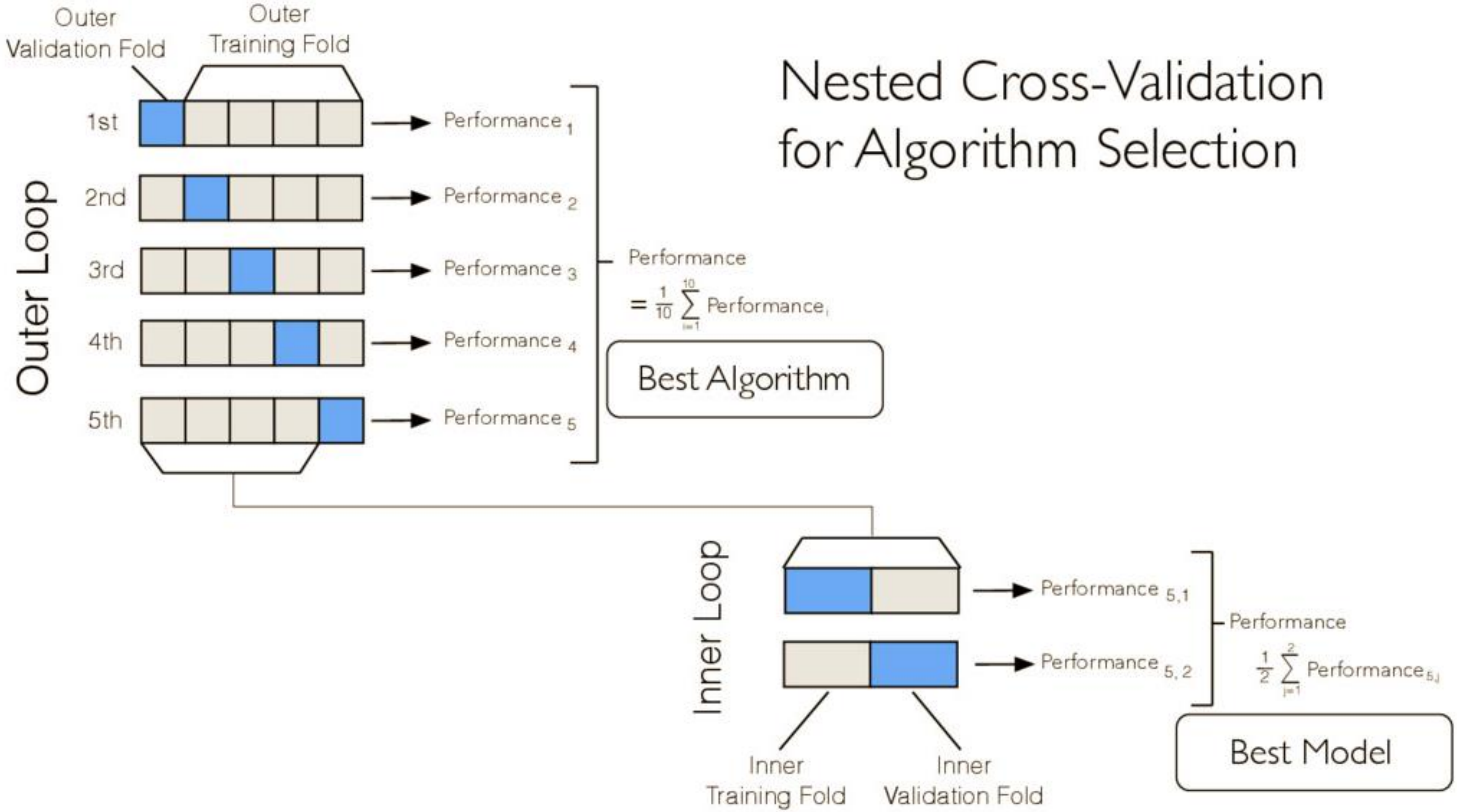






http://www.ijcai.org/Proceedings/95-2/Papers/016.pdf
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.3582

Nested Cross-Validation for Algorithm Selection



Outros fatores que afetam o desempenho do classificador

O desempenho de um classificador não depende apenas do algoritmo de aprendizagem; este depende também de outros fatores:

- A distribuição da classe
- Esparsidade do conjunto de dados
- Custo associado à *misclassification* (ter classificado incorretamente um exemplo)
- Dimensão de conjunto de treinamento e de teste

Quantos folds escolher?

- Maior número de folds:
  - Mais exata (*accurate*) a estimativa do erro verdadeiro
    menor bias (desvio), maior variância
  - Maior o número de runs ⇒ maior tempo de processamento

- Menor número de folds:
  - Menos exato (*accurate*) a estimativa do erro verdadeiro
    ⇒ maior bias, menor variância
  - Menor o número de runs ⇒ menor tempo de processamento

- <u>Na prática:</u> a escolha de *k* depende de *N* (nº de exemplos)
  - se tamanho muito grande ⇒ com 3-fold cross-validation podemos obter uma estimativa *accurate*
  - se dados muito esparsos ⇒ usar one-leave-out para poder obter o maior número possível de exemplos de treino

  - Como regra, **k=10** ou k=30

"The main theorem shows that there exists no universal (valid under all distributions) unbiased estimator of the variance of K-fold cross-validation" (*Bengio and Grandvalet, 2004*) *http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.3582*

Resumindo ...

- **Holdout**: para *N* grande
- **Random Subsampling**: melhora a estimativa obtida com holdout mas não existe controle sobre os exemplos usados para treinamento e para teste
- **k-fold Cross Validation**: para *N* intermediário

  estimação "*unbiased*" do erro verdadeiro, mas com elevada variância
- **0,632 booststraping**: para *N* pequena

  estimação "*unbiased*" no limite e com pouca variância

Outras Medidas de Avaliação

- *Speed*:
  - tempo para criação do modelo (*training time*)
  - tempo para o uso (*classification/prediction time*)
- *Robustness*: desempenho sobre ruído e dados faltantes
- *Scalability*: eficiência em bases de dados grandes
- *Interpretability*
  - Entendimento e conhecimento fornecido pelo modelo
- Outras medidas
  - qualidade das regras, como o tamanho das árvores de decisão ou poder de síntese das regras de classificação

Avaliação de Algoritmos de Aprendizagem

- Qual o desempenho do classificador $h(\mathbf{x})$ aprendido?
  - **Medida natural de desempenho:** **taxa de erro** (inversamente **taxa de acerto**)

$$Err(h(\mathbf{x}), D) = \frac{\#\,exemplos\ incorrectamente\ classificados}{total\ exemplos\ avaliados}$$

$$Acc(h(\mathbf{x}), D) = 1 - Err(h(\mathbf{x}), D)$$

- Como estimar a taxa de erro do algoritmo?
  - O erro de re-substituição (usando o conjunto de treinamento também como conjunto de teste) é um **estimador otimista**
  - Usar métodos de estimação:
    - **hold-out:** dividir conjunto de dados em treino(70%)-teste(30%)
    - métodos de reamostragem
      - **validação cruzada k-fold**
      - **bootstrap**

Erro de classificação

O principal objetivo é classificar corretamente exemplos nunca vistos

Errar o mínimo possível

- Minimizar taxa de erro para **exemplos nunca vistos**
- A isso se dá o nome de **Generalização**!

Geralmente não é possível medir com exatidão essa taxa de erro

- Ela deve ser estimada

Medidas para classificação binária

Dois tipos de erro:

- Classificação de um exemplo *N* (negativo) como *P* (positivo)
  - Falso positivo (alarme falso)
    - Ex.: Diagnosticado como doente, mas está saudável

- Classificação de um exemplo *P* como *N*
  - Falso negativo
    - Ex.: Diagnosticado como saudável, mas está doente

Medidas para classificação binária

- **Matriz de confusão** para 200 exemplos divididos em 2 classes
  - Pode ser utilizada para mais de duas classes

Classe predita

Classe verdadeira

	P	N
P	70	30
N	40	60

Classe predita

Classe verdadeira

	P	N
P	VP	FN
N	FP	VN

Estimativa de erro de classificação

- Acurácia
  - Trata as classes igualmente
  - Pode não ser adequada para dados desbalanceados
    - Classe rara pode ser mais interessante que a majoritária
    - Pode prejudicar o desempenho da classe minoritária

$$\text{Acurácia} = \frac{VP+VN}{VP+VN+FP+FN}$$

- Taxa do total de acertos (VP + VN) sobre o total de tentativas

Classe verdadeira \ Classe predita	P	N
P	VP	FN
N	FP	VN

Precisão versus Revocação

- Revocação (*recall*)
  - Taxa de exemplos positivos considerando aqueles que foram erroneamente classificados como negativos

$$\frac{VP}{VP+FN}$$ → Total de exemplos cuja classe verdadeira é positiva

- Precisão
  - Taxa de exemplos positivos considerando aqueles que foram erroneamente classificados como positivos

Total de exemplos classificados como positivos, mas que nem sempre são ← $$\frac{VP}{VP+FP}$$

		Classe predita	
		P	N
Classe verdadeira	P	VP	FN
	N	FP	VN

Sensibilidade *versus* Especificidade

- Sensibilidade
  - Taxa de exemplos positivos considerando aqueles que foram erroneamente classificados como negativos

$$\frac{VP}{VP + FN} \longrightarrow$$ Total de exemplos cuja classe verdadeira é positiva

Classe verdadeira \ Classe predita	P	N
P	VP	FN
N	FP	VN

- Especificidade
  - Taxa de exemplos negativos considerando todos os exemplos que deveriam ser classificados como negativos

$$\frac{VN}{VN + FP} \longrightarrow$$ Total de exemplos cuja classe verdadeira é negativa

Avaliação de desempenho

- *Medida-F*
  - Média harmônica ponderada da precisão e da revocação

$$\frac{(1+\alpha)\times(prec\times rev)}{\alpha\times prec+rev}$$

  - *Medida-F1*
    - Precisão e revocação têm o mesmo peso

$$\frac{2\times(prec\times rev)}{prec+rev} = \frac{2}{1/prec+1/rev}$$

Exemplo

- Seja um classificador com a seguinte matriz de confusão, definir:
  - Acurácia
  - Precisão x Revocação
  - Sensibilidade x Especificidade

		Classe predita	
		P	N
Classe verdadeira	P	70	30
	N	40	60

Exemplo

Acurácia = $\dfrac{VP+VN}{VP+VN+FP+FN}$

Precisão = $\dfrac{VP}{VP+FP}$

Revocação =
ou Sensibilidade $\dfrac{VP}{VP+FN}$

Especii cidade = $\dfrac{VN}{VN+FP}$

Predito

Verdadeiro		P	N
	P	VP	FN
	N	FP	VN

Verdadeiro		P	N
	P	70	30
	N	40	60

Exemplo

$$\text{Acurácia} = \frac{VP+VN}{VP+VN+FP+FN} = (70 + 60) / (70 + 30 + 40 + 60) = 0.65$$

$$\text{Precisão} = \frac{VP}{VP+FP} = 70/(70+40) = 0.64$$

$$\text{Revocação} = \frac{VP}{VP+FN} = 7/(70+30) = 0.70$$

ou Sensibilidade

$$\text{Especificidade} = \frac{VN}{VN+FP} = 60/(40+60) = 0.60$$

Predito

Verdadeiro

	P	N
P	VP	FN
N	FP	VN

	P	N
P	70	30
N	40	60

Quanto maior, melhor!!! Para todas essas medidas!!!

Avaliação de Algoritmos de Aprendizagem

Dois problemas distintos:

- Dados um algoritmo e um conjunto de dados
  - Quanta confiança podemos ter na taxa de erro (acerto) estimada?

    ⇒ calcular **intervalos de confiança**

- Dados dois algoritmos e um conjunto de dados
  - Qual algoritmo tem melhor desempenho (capacidade de generalização) ?

    ⇒ realizar **testes de significância**

Intervalo de Confiança

Um intervalo de confiança para um parâmetro θ, a um grau de confiança $1-\alpha$, é uma concretização de um intervalo aleatório (L_{inf}, L_{sup}) para o qual se tem:

$$\boxed{P(L_{inf} < \theta < L_{sup}) = 1-\alpha\ ,\ \alpha \in (0,1)}$$

onde α deve ser um valor muito pequeno para termos confianças elevadas

Valores usuais para o grau de confiança: **95%**, **99%** e **90%**

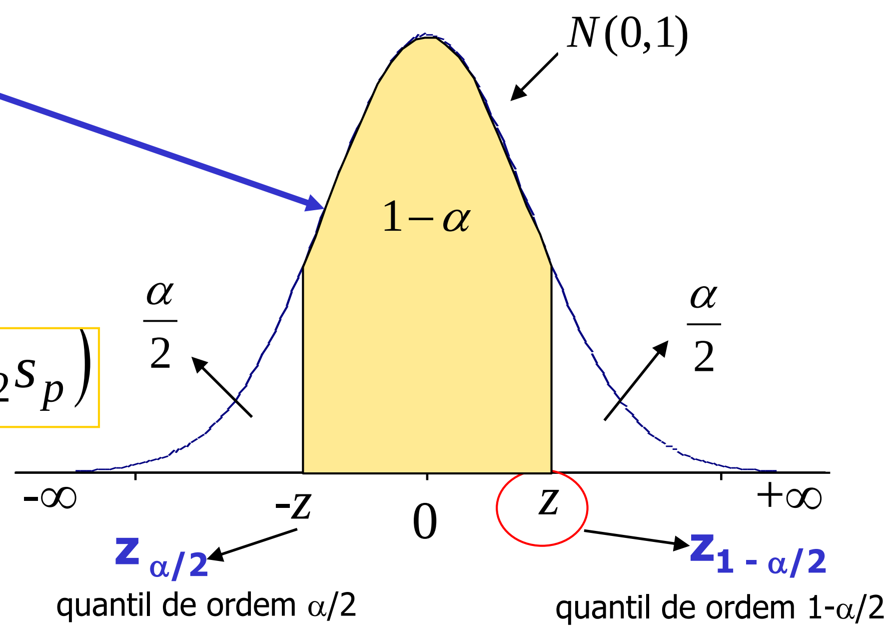
Intervalo de Confiança para a Taxa de Acerto (grandes amostras)

- Para conjunto de teste com N > 30, a taxa de acerto pode ser aproximada, pelo TLC, com uma distribuição Normal de **média** *p* e **variância** *p(1-p)/N*

$$acc = \frac{X}{N} \underset{aprox}{\sim} N\left(p, \frac{p(1-p)}{N}\right) \overset{centrando\ e\ reduzindo}{\Leftrightarrow} Z = \frac{acc - p}{\sqrt{p(1-p)/N}} \underset{aprox}{\sim} N(0,1)$$

- Intervalo de confiança para a taxa de acerto verdadeira *p* (desconhecida):

$$\boxed{P\left(z_{\alpha/2} < \frac{acc - p}{\sqrt{p(1-p)/N}} < z_{1-\alpha/2}\right) = 1 - \alpha}$$



Um IC aproximado para *p*, com um grau de confiança 1-α é dado por:

$$\boxed{IC_{(1-\alpha)}(p) \approx \left(\hat{a}cc - z_{1-\alpha/2}\, s_p,\ \hat{a}cc + z_{1-\alpha/2}\, s_p\right)}$$

onde $\boxed{\hat{a}cc = \frac{x}{N}}$ e $\boxed{s_p = \sqrt{\frac{\hat{a}cc(1-\hat{a}cc)}{N}}}$

usamos $\hat{a}cc$ como uma estimativa pontual de *p para calcular o desvio padrão amostral e* a letra minúscula *x* pois estamos representando uma concretização da v.a. *X*

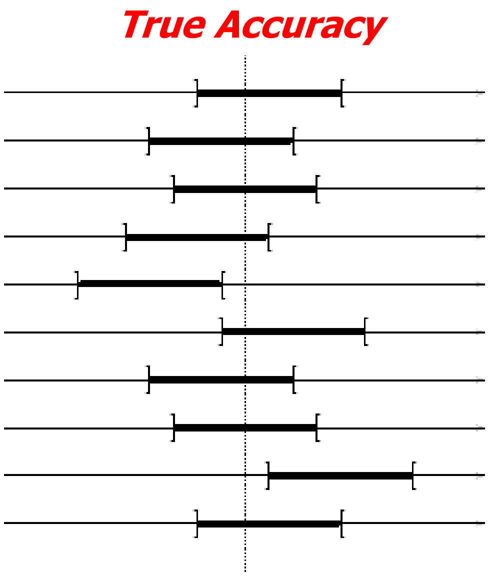
Interpretação do IC

Determinando um IC aproximado para a taxa de acerto verdadeira a 95%:

$$\boxed{IC_{(95\%)}(p) \approx \left(\hat{a}cc - 1.96 \times s_p , \hat{a}cc + 1.96 \times s_p\right)}$$

onde $\boxed{\hat{a}cc = \dfrac{x}{N}}$, $\boxed{s_p = \sqrt{\dfrac{\hat{a}cc(1-\hat{a}cc)}{N}}}$

Interpretação: 95% dos possíveis ICs obtidos a partir de uma amostra de tamanho *N*, conterão de fato o verdadeiro valor da *accuracy*



Comparação do Desempenho (dois algoritmos e um conjunto de dados)

Dado dois algoritmos e um conjunto de dados qual algoritmo tem melhor desempenho ? $\Rightarrow$ realizar **teste de hipótese (TH)**

Procedimento estatístico que permite averiguar a "sustentação" de **uma hipótese**

- Existem duas hipóteses: H_0 vs H_1
  - **Hipótese Nula** – H_0 (usar sempre sinal =)
  - **Hipótese Alternativa** – H_1
- Dois tipos de testes:
  - **unilateral:** H_1 apenas contempla possibilidades à direita ou à esquerda de H_0

    $H_0 : \mu = 1$ vs $H_1 : \mu > 1$ (*unilateral à direita*) $H_0 : \mu = 1$ vs $H_1 : \mu < 1$ (*unilateral à esquerda*)
  - **bilateral:** H_1 contempla possibilidades à direita ou à esquerda de H_0

    $H_0 : \mu = 1$ vs $H_1 : \mu \neq 1$ (*bilateral*)
- Dois tipos de decisão:
  - Rejeitar a hipótese nula H_0
  - Não rejeitar a hipótese nula H_0

Testes de Hipóteses (TH)
Definições básicas

- Estatística de teste **T**: estatística calculada a partir da amostra e usada para tomar a decisão
- Região de rejeição ou região crítica **RC**: conjunto de valores da estatística de teste que nos levam a rejeitar H_0
- Nível de significância ou tamanho do teste **α**:

  α = P(Erro de tipo I) = P(rejeitar $H_0 | H_0$ verdadeiro)

  normalmente α=0.1, α=0.05 ou α=0.01
- Potência do teste **1 – β**:

  1 – β = 1 – P(Erro de tipo II) = P(não rejeitar $H_0 | H_1$ verdadeiro)
- p-value: a probabilidade de observar um valor da estatística de teste tanto ou mais afastado que o valor observado na amostra, assumindo que H_0 é verdadeira

TH para Comparação do Desempenho (dois algoritmos e um conjunto de dados)

- Ambos os algoritmos devem:
  - aprender nos mesmos conjuntos de treino
  - avaliar os modelos induzidos nos mesmos conjuntos de teste
- Testar: existe uma diferença significativa no desempenho ?
  - **Hipótese nula H_0** : não há diferença significativa
  - **Hipótese alternativa H_1** : há diferença significativa
- Teste de Hipóteses: H_0 vs. H_1

  (deve medir a evidencia que existe em favor da rejeição da hipótese nula)
- usar o teste *t* para amostras emparelhadas (*paired t-test*)
  - permite inferir sobre a igualdade das médias de duas amostras emparelhadas
  - se as amostras têm dimensão inferior a 30

    $\Rightarrow$ as amostras devem provir de populações **normalmente distribuídas**
  - se é violada a normalidade dos dados

    $\Rightarrow$ usar **testes não paramétricos**
    - **teste de Wilcoxon (signed-ranks)** ou **teste dos sinais (sign test)**

Teste T para Amostras Emparelhadas (dois algoritmos e um conjunto de dados)

Amostras emparelhadas: se pares de observações (x_i, y_i) são dependentes sendo todos os restantes pares (x_i, y_j), $i \neq j$ independentes

Para obter duas amostras emparelhadas usar **validação cruzada k-fold:**

Para cada fold j (j=1,..,k) :

1) estimar valor de medida de desempenho c_{ij} para cada algoritmo i (i=1,2)

(taxa de erro, taxa de acerto, precisão, sensibilidade, área AUC, etc.)

2) calcular as diferenças no desempenho: $d_j = c_{1j} - c_{2j}$

para k=10

Fold	1	2	3	4	5	6	7	8	9	10
Algoritmo1	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{110}
Algoritmo2	c_{21}	c_{22}	c_{23}	c_{24}	c_{25}	c_{26}	c_{27}	c_{28}	c_{29}	c_{210}
Diferenças	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}

Teste T para Amostras Emparelhadas (dois algoritmos e um conjunto de dados)

Testar se a diferença no desempenho é estatisticamente significante

1. Identificar o parâmetro de interesse e especificar H_0 e H_1

   $H_0 : \mu_D = 0$ vs. $H_1 : \mu_D \neq 0$ (a média da diferença na população)

   $D = M_1 - M_2 \sim N\left(\mu_D, \sigma_D^2\right)$ D - v.a. que representa a diferença entre as v.a. que representam as medidas de desempenho M_1 e M_2 obtidas pelos algoritmos 1 e 2

2. Calcular t_{obs} usando a estatística do teste T

   $$t_{obs} = \frac{\bar{d}}{s_{c_D}\sqrt{k}}$$

   média amostral

   $$\bar{d} = \frac{1}{k}\sum_{j=1}^{k} d_j$$

   Desvio padrão amostral corrigido

   $$s_{c_D} = \sqrt{\sum_{j=1}^{k}(d_j - \bar{d})^2 \Big/ (k-1)}$$

   se H_0 é verdadeira: **T** tem distribuição t-student com ***k*-1 graus de liberdade**

3. Determinar o p-value usando a tabela de distribuição t-student:
   - $2P(T < t_{obs} | H_0)$ se $\bar{d} < 0$ ou $2P(T > t_{obs} | H_0)$ se $\bar{d} > 0$

4. Tomar decisão: **rejeitar H_0 se p-value $\leq \alpha$**
   - α é o nível de significância, usualmente α=0.05 ou α=0.01

k-fold cross-validated paired t-test
Problemas na implementação

- O teste T pressupõe que as diferenças no desempenho $d_j = c_{1j} - c_{2j}$
  - provenham de uma distribuição Normal ⇒ difícil de provar pois há poucos dados (se k=10, a amostra apenas contêm 10 elementos)
- Os conjuntos de testes são independentes, porém os conjuntos de treino não (se k=10, dois conjuntos de treino partilham o 80% dos dados)

  ⇒ Elevada probabilidade de ocorrência do erro de Tipo I

  erro de Tipo I ≡ $P(\text{rejeitar } H_0 | H_0 \text{ true})$

  incorretamente detecta que existe diferença significativa no desempenho dos dois algoritmos quando realmente esta diferença não existe

- **Alternativas:**
  - Testes não paramétricos: Wilcoxon (signed ranks) ou dos sinais — **o mais recomendado**
  - 10x10 cross validation = 10 iterações de 10-fold CV

    ⇒ gera amostra de tamanho 100 (pelo TLC aproxima-se à Normal)
  - 5x2 cross validation

    Dietterich (1998) provou que este teste reduz o erro de Tipo I

Tomada de decisão. Podemos errar?

Um classificador pode auxiliar à tomada de decisões entre diferentes ações. Podemos permitir decisões incorretas?

Tomada de decisão numa central nuclear: Um classificador ***h*** prediz se **abrir** ou **fechar** a válvula do módulo de refrigeração num dado momento

- Avaliamos o desempenho num conjunto de teste = **100 000 dados** acumulados no último mês; a classe é o resultado da decisão tomada por um operário (esperto) em cada momento
  - Número de exemplos da classe "**fechar**": **99 500**
  - Número de exemplos da classe "**abrir**": **500**
- Suponhamos h prediz sempre "**fechar**" (classe maioritária). A taxa de erro é muito pequena: $$\mathrm{Err} = \frac{500}{100000} \times 100 = 0.5\%$$

É h um bom clasificador?

© Exemplo adaptado de Cesar Martines & José Orallo

Qual Classificador é melhor?

Exemplo: (conjunto de teste com 100.000 instâncias)

Predita

Real

h_1	abrir	fechar
abrir	300	200
fechar	500	99000

Predita

h_2	abrir	fechar
abrir	0	500
fechar	0	99500

Predita

h_3	abrir	fechar
abrir	400	100
fechar	5400	94100

Sensibilidade

Recall

Especificidade

Precision

ERRO: 0,7%

TPR= 300 / 500 = 60%
FNR= 200 / 500 = 40%
TNR= 99000 / 99500 = 99,5%
FPR= 500 / 99500 = 0,05%
Precision = 300 / 800 = 37,5%

ERRO: 0,5%

TPR= 0 / 500 = 0%
FNR= 500 / 500 = 100%
TNR= 99500 / 99500 = 100%
FPR= 0 / 99500 = 0%
Precision = 0 / 0 = INDEFINIDO

ERRO: 5,5%

TPR= 400 / 500 = 80%
FNR= 100 / 500 = 20%
TNR= 94100 / 99500 = 94,6%
FPR= 5400 / 99500 = 5,4%
Precision = 400 / 5800 = 6,9%

Matriz de Custos

Em muitas situações todos os erros produzidos por um modelo preditivo não têm as mesmas consequências

Tomada de decisão numa central nuclear: Deixar fechada uma válvula quando é necessário abri-la pode provocar uma **explosão**, enquanto abrir uma válvula quando pode se manter fechada pode provocar uma **parada**

- Matriz de custos

Predita

Real

	abrir	fechar
abrir	**0**	**2000€**
fechar	100€	**0**

O importante não é obter um classificador que erre o menos possível senão aquele que tenha um **menor custo**

- A partir da matriz de custo avalia-se cada classificador e seleccionamos o classificador com menor custo

Problema de Decisão Central Nuclear

Matrizes de confusão para 3 classificadores

Pred (colunas) / Real (linhas)

h_1	abrir	fechar
abrir	300	200
fechar	500	99000

h_2	abrir	fechar
abrir	0	500
fechar	0	99500

h_3	abrir	fechar
abrir	400	100
fechar	5400	94100

Predita (colunas) / Real (linhas)

Matriz de custo

	abrir	fechar
abrir	0	2000€
fechar	100€	0

Matrizes resultado

h_1	abrir	fechar
abrir	0€	400.000€
fechar	50.000€	0€

h_2	abrir	fechar
abrir	0€	1.000.000€
fechar	0€	0€

h_3	abrir	fechar
abrir	0€	200.000€
fechar	540.000€	0€

CUSTO TOTAL: 450.000€

CUSTO TOTAL: 1.000.000€

CUSTO TOTAL: 740.000€

De que depende o custo final?

- Para problemas de duas classes depende de um contexto (o ***skew***):
  - proporção do custo dos FP e FN
  - proporção de exemplos negativos e positivos
- Para o exemplo anterior calculamos o "***slope***":

Proporção dos custos dos erros

$$\frac{FPcost}{FNcost} = \frac{100}{2000} = \frac{1}{20}$$

Proporção das classes

$$\frac{Neg}{Pos} = \frac{99500}{500} = 199$$

$$slope = \frac{1}{20} \times 199 = 9{,}95$$

- o "slope" é suficiente para determinar qual classificador é o melhor:

h_1: FNR= 40%, FPR= 0,5%
Custo unitário =
1 x 0,40 + 9,95 x 0,005 = 0,45

h_2: FNR= 100%, FPR= 0%
Custo Unitário =
1 x 1 + 9,95 x 0 = 1

h_3: FNR= 20%, FPR= 5,4%
Custo Unitário =
1 x 0,20 + 9,95 x 0,054 = 0,74

Menor custo unitário = melhor classificador

Desempenho

- O classificador com menor erro não é obrigatoriamente o melhor.

  O desempenho de um classificador também depende:

  1. **do contexto:**
     - **distribuição das classes** (não sempre todas as classes têm a mesma proporção, podem não estar balanceadas, i.e. 1/1 =50 % de cada)
     - **custos de cada tipo de erro**
  2. **tamanho dos conjuntos de treino e teste**

- PROBLEMA: Em muitas aplicações não se conhece a priori a distribuição das classes no conjunto de teste ⇒ desta forma é difícil estimar a matriz de custos ⇒ comparar classificadores usando **análises ROC**

Curvas ROC

Curva ROC

- ROC = **R**eceiver **O**perating **C**haracteristic Curve

- Enfoque gráfico que mostra um *trade-off* entre as taxas de TP (TPR) e FP (FPR) de um classificador.

- TPR = TP/(TP + FN) (= recall)
- FPR = FP/(TN + FP)
- **Ideal : TPR = 1 e FPR = 0**

Curva ROC de um classificador ?

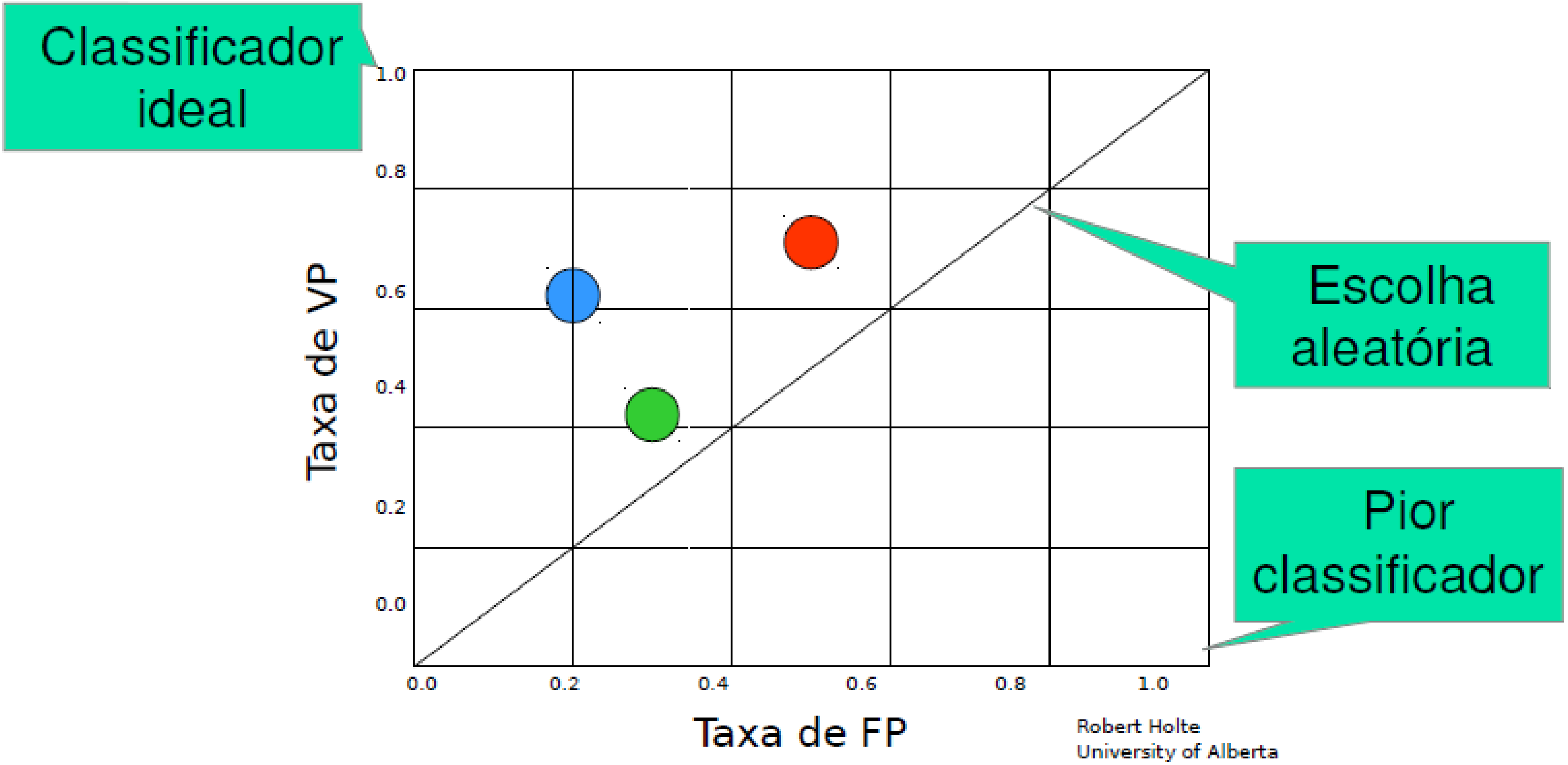
- O classificador precisa produzir, para cada exemplo X, a probabilidade do exemplo X ser classificada na classe **Positiva**.
- Classificadores como redes neurais artificiais e redes bayesianas produzem tais probabilidades.
- Para outros tipos de classificadores, é preciso calcular esta probabilidade.
  - Exemplo: KNN, árvore de decisão ?

Exemplo

- Plotar Gráfico ROC para 3 classificadores:
  - Considerando um caso Binário (duas classes) e com conjuntos de treinamento e teste estratificados

Classificador 1	Classificador2	Classificador 3
TFP = 0.3	TFP = 0.5	TFP = 0.2
TVP = 0.4	TVP = 0.7	TVP = 0.6

Gráfico ROC para os três classificadores



Curva Roc

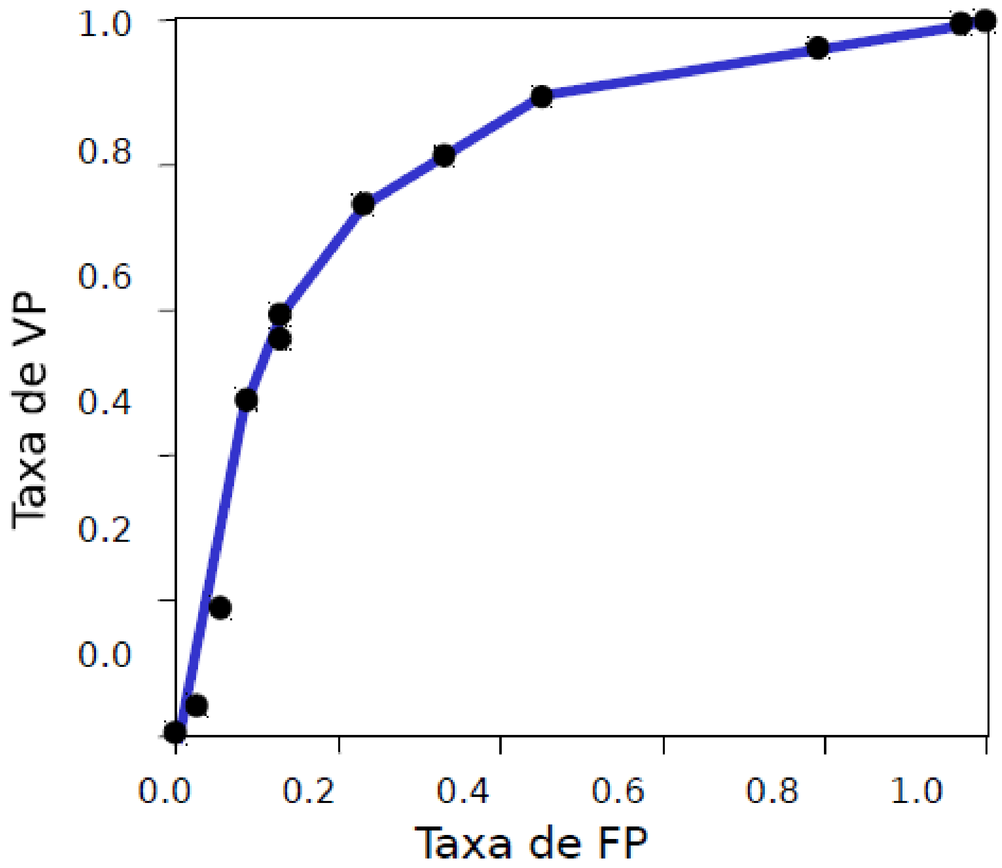
Dependendo da forma de conduzir experimentos:

- Classificadores podem produzir um simples ponto no gráfico ROC

- Ou produzir curvas de desempenho que permitem analisar o quão sensível é o classificador em relação aos eixos do Gráfico ROC

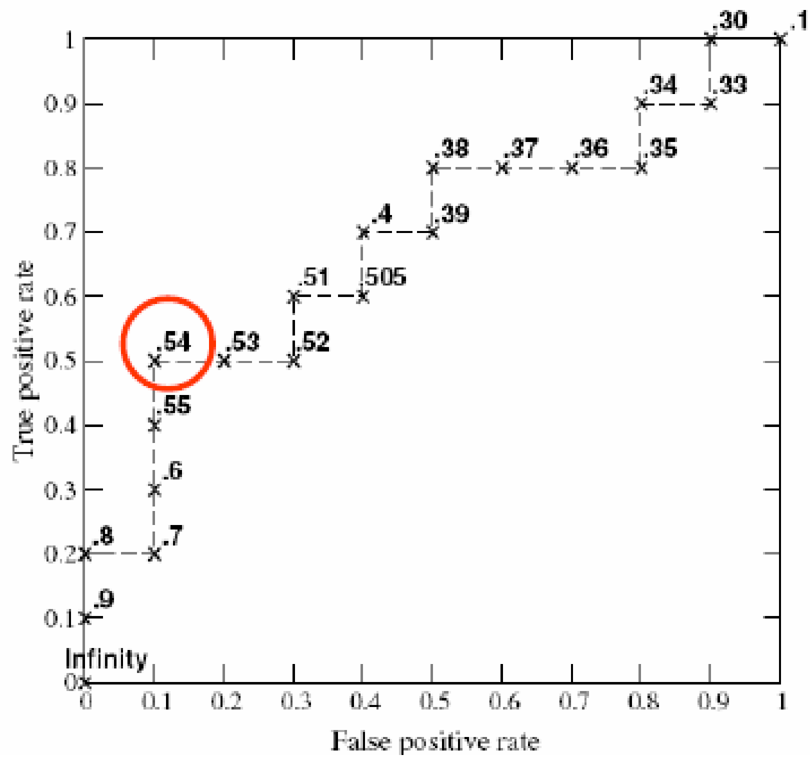
Curvas ROC

Vários pontos resultantes do desempenho de um Classificador são unidos para formar uma curva



Curva ROC – Alternativa a definição de limiares

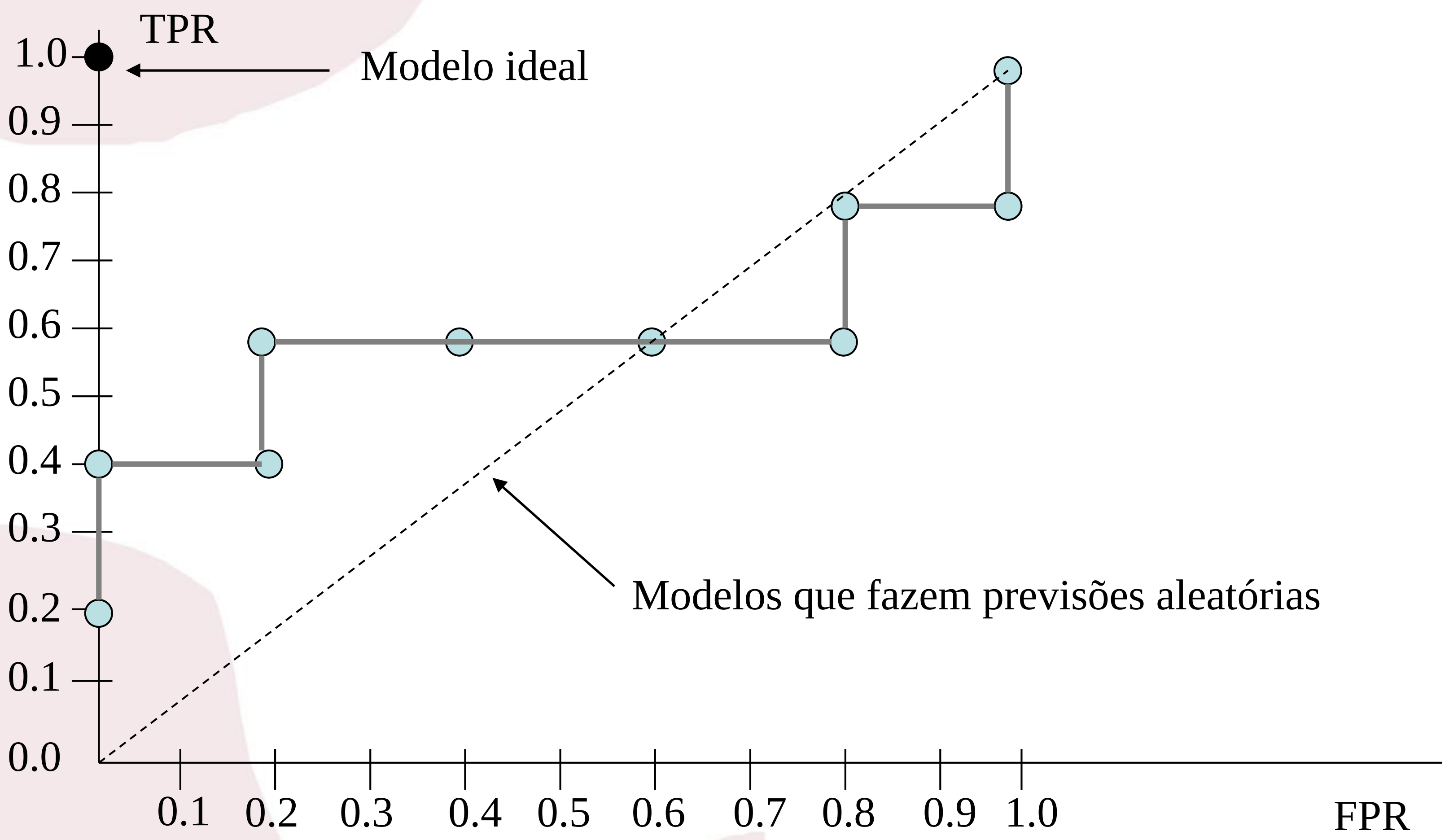
Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1



Curva Roc

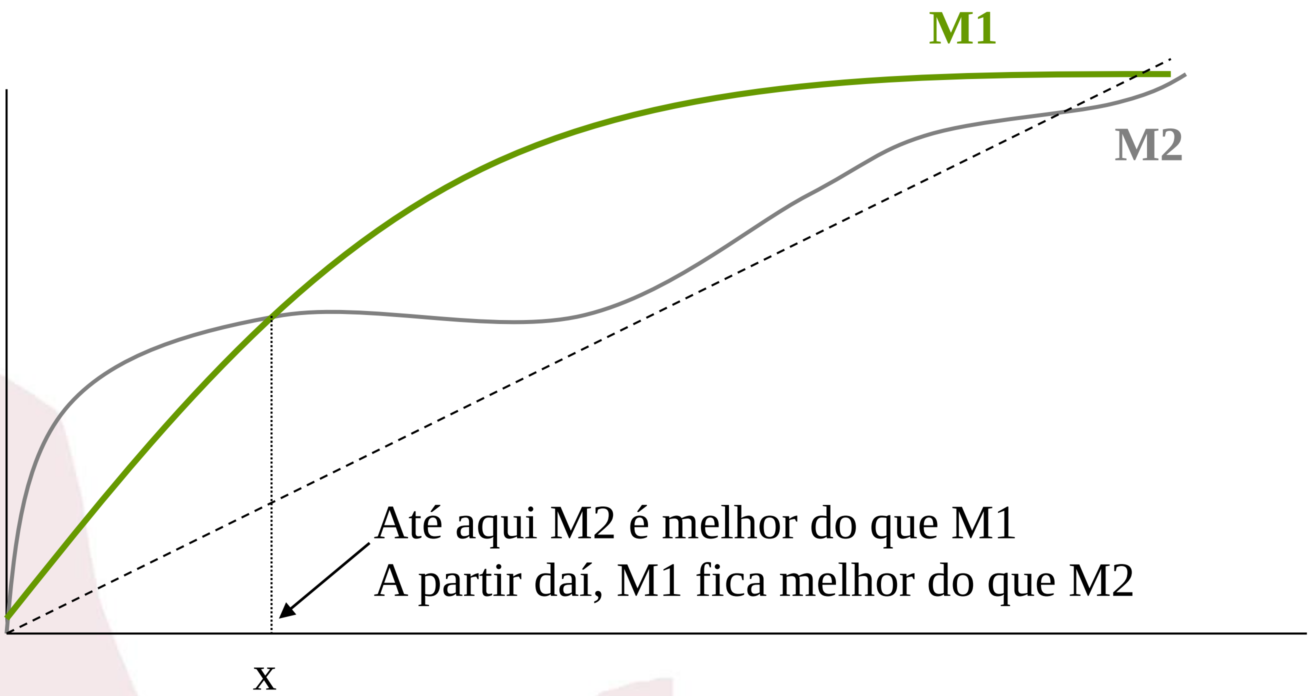
- Cada ponto na curva corresponde a um dos modelos induzidos pelo classificador
- Um bom modelo deve estar localizado próximo do ponto (0,1)
- Modelos localizados na diagonal são modelos aleatórios – TPR = FPR
- Modelos localizados acima da diagonal são melhores do que modelos abaixo da diagonal.

Exemplo


TPR
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
Modelo ideal
Modelos que fazem previsões aleatórias
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
FPR

Comparando performance

- **Curvas Roc** são utilizadas para se medir a performance relativa de diferentes classificadores.



Área abaixo da curva ROC (AUC)

- A área abaixo da curva ROC fornece medida para comparar performances de classificadores.
- Quanto maior a área AUC melhor a performance global do classificador.
- Classificador ótimo: área =1
- Classificador randômico: área = 0.5

Curva ROC vs. *Precision-Recall* Classes Desbalanceadas

O que acontecer se houver mudanças na proporção de exemplos positivos e negativos no conjunto de teste?

- A curva ROC <u>não é sensível às mudanças</u> - baseia-se nas taxas TPR e FPR, as quais não dependem da distribuição das classes

atual \ predita	+	-
+	TP	FN
-	FP	TN

atual \ predita	+	-
+	TP	FN
-	FP	TN

Proporções pelas linhas – para cada classe a sua taxa

- O gráfico de precision vs. recall <u>é sensível às mudanças</u> – a medida de precisão depende da distribuição das classes

$$\text{precision} = \frac{TP}{TP + FP}$$

atual \ predita	+	-
+	TP	FN
-	FP	TN

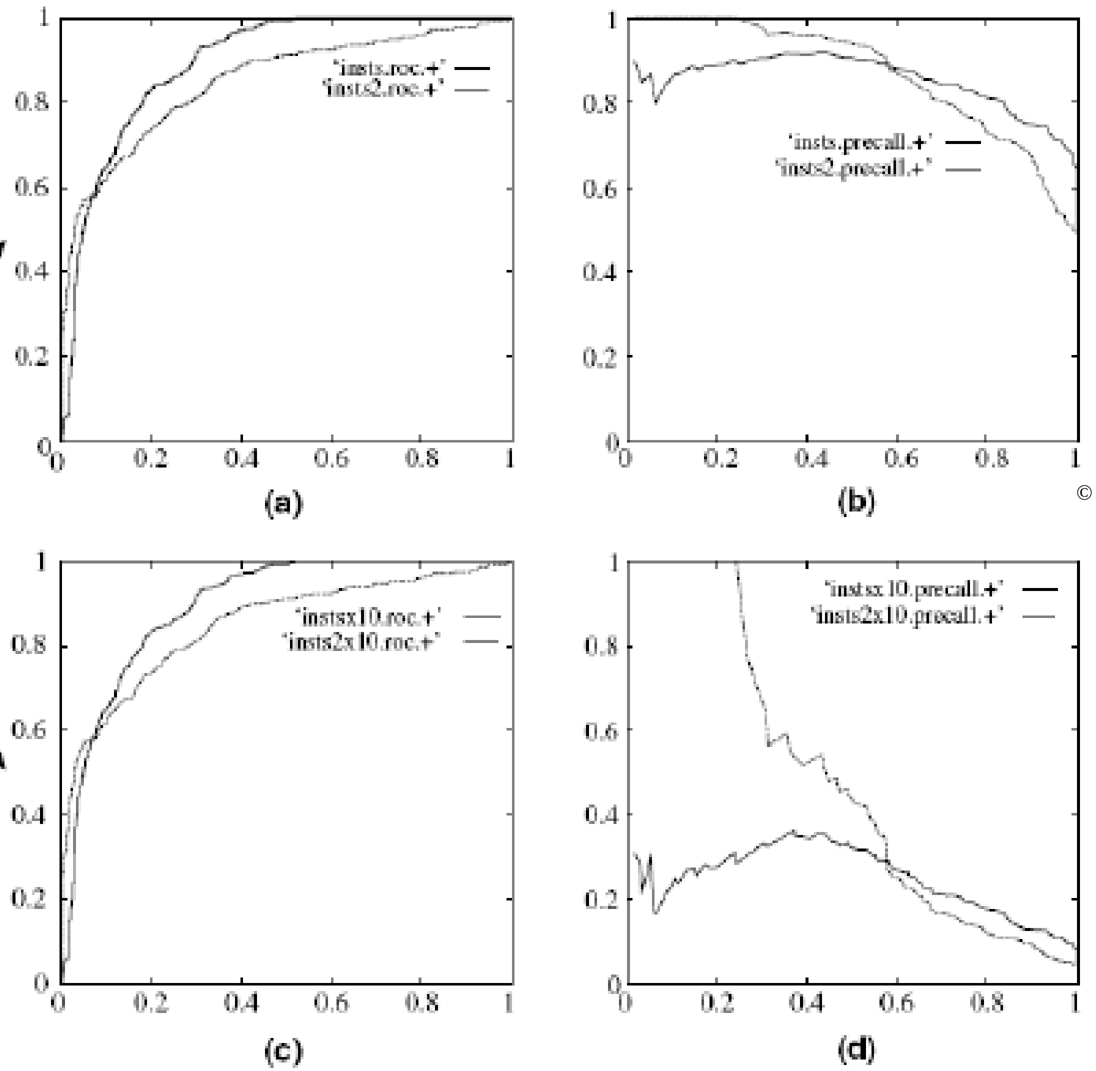
Proporção pela coluna- tem em conta nº exemplos "+" e "-"

Curva ROC vs. Precision-Recall Classes Desbalançadas

Curva ROC 1:1

Constante

Curva ROC 1:10



Curva *Precision-recall* 1:1

© Tom Fawcett

Curva *Precision-recall* 1:10

Referência

Journal of Machine Learning Research 7 (2006) 1–30 Submitted 8/04; Revised 4/05; Published 1/06

Statistical Comparisons of Classifiers over Multiple Data Sets

Janez Demšar JANEZ.DEMSAR@FRI.UNI-LJ.SI
Faculty of Computer and Information Science
Tržaška 25
Ljubljana, Slovenia

Editor: Dale Schuurmans

Abstract

While methods for comparing two learning algorithms on a single data set have been scrutinized for quite some time already, the issue of statistical tests for comparisons of more algorithms on multiple data sets, which is even more essential to typical machine learning studies, has been all but ignored. This article reviews the current practice and then theoretically and empirically examines several suitable tests. Based on that, we recommend a set of simple, yet safe and robust non-parametric tests for statistical comparisons of classifiers: the Wilcoxon signed ranks test for comparison of two classifiers and the Friedman test with the corresponding post-hoc tests for comparison of more classifiers over multiple data sets. Results of the latter can also be neatly presented with the newly introduced CD (critical difference) diagrams.

Keywords: comparative studies, statistical methods, Wilcoxon signed ranks test, Friedman test, multiple comparisons tests

Referência

Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis

Alessio Benavoli[†] ALESSIO@IDSIA.CH
Giorgio Corani[†] GIORGIO@IDSIA.CH
Janez Demšar[♮] JANEZ.DEMSAR@FRI.UNI-LJ.SI
Marco Zaffalon[†] ZAFFALON@IDSIA.CH

[♮]*Faculty of Computer and Information Science, University of Ljubljana, Vecna pot 113, SI-1000 Ljubljana, Slovenia*
[†]*Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) Galleria 2, 6928 Manno, Switzerland*

.04316v3 [stat.ML] 15 Jul 2017

Editor:

Abstract

The machine learning community adopted the use of null hypothesis significance testing (NHST) in order to ensure the statistical validity of results. Many scientific fields however realized the shortcomings of frequentist reasoning and in the most radical cases even banned its use in publications. We should do the same: just as we have embraced the Bayesian paradigm in the development of new machine learning methods, so we should also use it in the analysis of our own results. We argue for abandonment of NHST by exposing its fallacies and, more importantly, offer better—more sound and useful—alternatives for it.

Keywords: comparing classifiers, null hypothesis significance testing, pitfalls of p-values, Bayesian hypothesis tests, Bayesian correlated t-test, Bayesian hierarchical correlated t-test, Bayesian signed-rank test

http://repositorio.icmc.usp.br/handle/RIICMC/6563?show=full

Referência

- "The NHST computes the probability of getting the observed (or a larger) difference between classifiers if the null hypothesis of equivalence was true, which is not the probability of one classifier being more accurate than another, given the observed empirical results."

- Já o teste Bayesiano funciona ao contrário (pensando de forma bem geral), ele retorna as probabilidades da comparação entre dois algoritmos dados os resultados experimentais: P(algA > algB), P(algA = algB), P(algA < algB). Com base nesses valores, podem ser usados pontos de corte, por exemplo, se P(algA > algB) > 0.95, então o algA realmente ganhou do algB. Não precisa ser 0.95, o autor chega a mencionar 0.90, 0.80, isso vai depender do contexto.

http://repositorio.icmc.usp.br/handle/RIICMC/6563?show=full

Referência

MIC2003: The Fifth Metaheuristics International Conference 110-1

A Statistical Test for Comparing Success Rates

Éric D. Taillard*

*EIVD, University of Applied Sciences of Western Switzerland
Route de Cheseaux 1, Case postale
CH-1401 Yverdon-les-Bains, Switzerland

Eric.Taillard@eivd.ch

Referência

1

Curvas ROC para avaliação de classificadores

R. C. Prati, G. E. A. P. A. Batista e M. C. Monard

Resumo — **Gráficos ROC foram recentemente introduzidos como uma poderosa ferramenta para a avaliação de algoritmos de aprendizado. Apesar de gráficos ROC serem conceitualmente simples, existem algumas interpretações errôneas a seu respeito. Neste artigo, é feita uma introdução à análise ROC dentro do escopo de aprendizado de máquina e mineração de dados, ressaltando as vantagens de sua utilização bem como apontando os erros mais comuns quanto à sua interpretação e utilização.**

Palavras-chave — **Curvas ROC (*ROC graphs*), Aprendizado de Máquina (*Machine Learning*), Mineração de Dados (*Data Mining*), Avaliação de Modelos (*Model Evaluation*).**

conjunta e condicional para a avaliação de modelos de classificação. Na Seção III, é discutida a avaliação de modelos, ressaltando a diferença entre classificação e ordenação. Na Seção IV, é apresentado o gráfico ROC propriamente dito, destacando a sua utilização em AM e MD. Finalmente, na Seção V são apresentadas as considerações finais.

II. Probabilidade Conjunta e Condicional

Para induzir um classificador, um algoritmo de aprendizado supervisionado utiliza uma amostra de casos para os quais se

Referência

Machine Learning, 45, 171–186, 2001

A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems

DAVID J. HAND d.j.hand@ic.ac.uk
ROBERT J. TILL r.till@ic.ac.uk
Department of Mathematics, Imperial College, Huxley Building, 180 Queen's Gate, London SW7 2BZ, UK

Editor: David W. Aha

Abstract. The area under the ROC curve, or the equivalent Gini index, is a widely used measure of performance of supervised classification rules. It has the attractive property that it side-steps the need to specify the costs of the different kinds of misclassification. However, the simple form is only applicable to the case of two classes. We extend the definition to the case of more than two classes by averaging pairwise comparisons. This measure reduces to the standard form in the two class case. We compare its properties with the standard measure of proportion correct and an alternative definition of proportion correct based on pairwise comparison of classes for a simple artificial case and illustrate its application on eight data sets. On the data sets we examined, the measures produced similar, but not identical results, reflecting the different aspects of performance that they were measuring. Like the area under the ROC curve, the measure we propose is useful in those many situations where it is impossible to give costs for the different kinds of misclassification.

Keywords: receiver operating characteristic, ROC curve, AUC, Gini index, error rate

Considerações

- Tabelas de resultados
  - Apresentar sempre médias e desvio-padrão
  - Realizar testes de distribuição
  - Testes não paramétricos
    - Wilcoxon Signed-ranks Test
    - Counts of Wins, Losses and Ties: Sign Test
    - ANOVA
- Número de bases de teste?
- Análise dos resultados ao contrário de descrever as tabelas