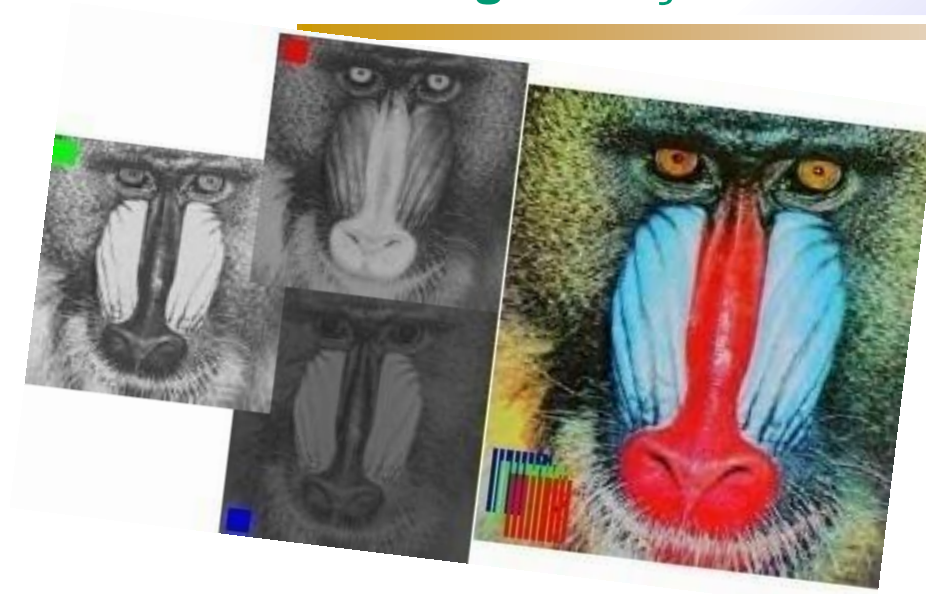


Prof. Dr. Leandro Alves Neves

Pós-graduação em Ciência da Computação



Aula 12

Processamento Digital de
Imagens

Sumário

- Padrões
- Espaço de Características e Modelos para Reconhecimento de Padrões
- Classificação, Métricas, Matriz de Confusão e Espaço ROC
- Generalização: Modelos de Amostragem
- Dimensionalidade e Redução de Dimensionalidade

Etapas

Aquisição



Pré-processamento



Segmentação



Extração de atributos



Reconhecimento e Interpretação

Reconhecimento e Interpretação?

- Descrição:
 - **Definir características** fundamentais do objeto

- Reconhecimento:
 - **Atribuir rótulos** a cada um dos objetos descritos (rotular)

- Interpretação:
 - **Atribuir sentido** à imagem

Reconhecimento e Interpretação?



Refeição (Padrão)



Padrões

- ❑ Descritores de propriedades de objetos ou imagem:
 - Processo nomeado de Extração de Características/features
 - **Padrão**: formado por **um ou mais descritores/features/atributos**
- ❑ **Classe de Padrões**
 - ❑ Família de padrões que **compartilham propriedades** comuns
 - ❑ Denotadas como $w_1, w_2, w_3, \dots, w_M$, onde M é o número de classes

Padrões

■ Arranjos de Padrões

- Cadeias e Árvores (descrições estruturais)
- Vetores (descrições quantitativas)

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix}$$

x_i é um descritor/atributo

n é o número de descritores/
características/atributos

Espaço de Características: Classes

	Caract. 1	Caract. 2	Caract. 3	...	Caract. m
Imagem 1	10	15	98	...	45
Imagem 2	54	26	54	...	56
Imagem 3	32	98	98	...	48
...
Imagem n	54	98	2	...	54



❑ Busca por classes de Padrões

❑ Obter propriedades comuns e definir as classes:

❑ $w_1, w_2, w_3, \dots, w_M$, em que M é o número de classes

Espaço de Características: Classes

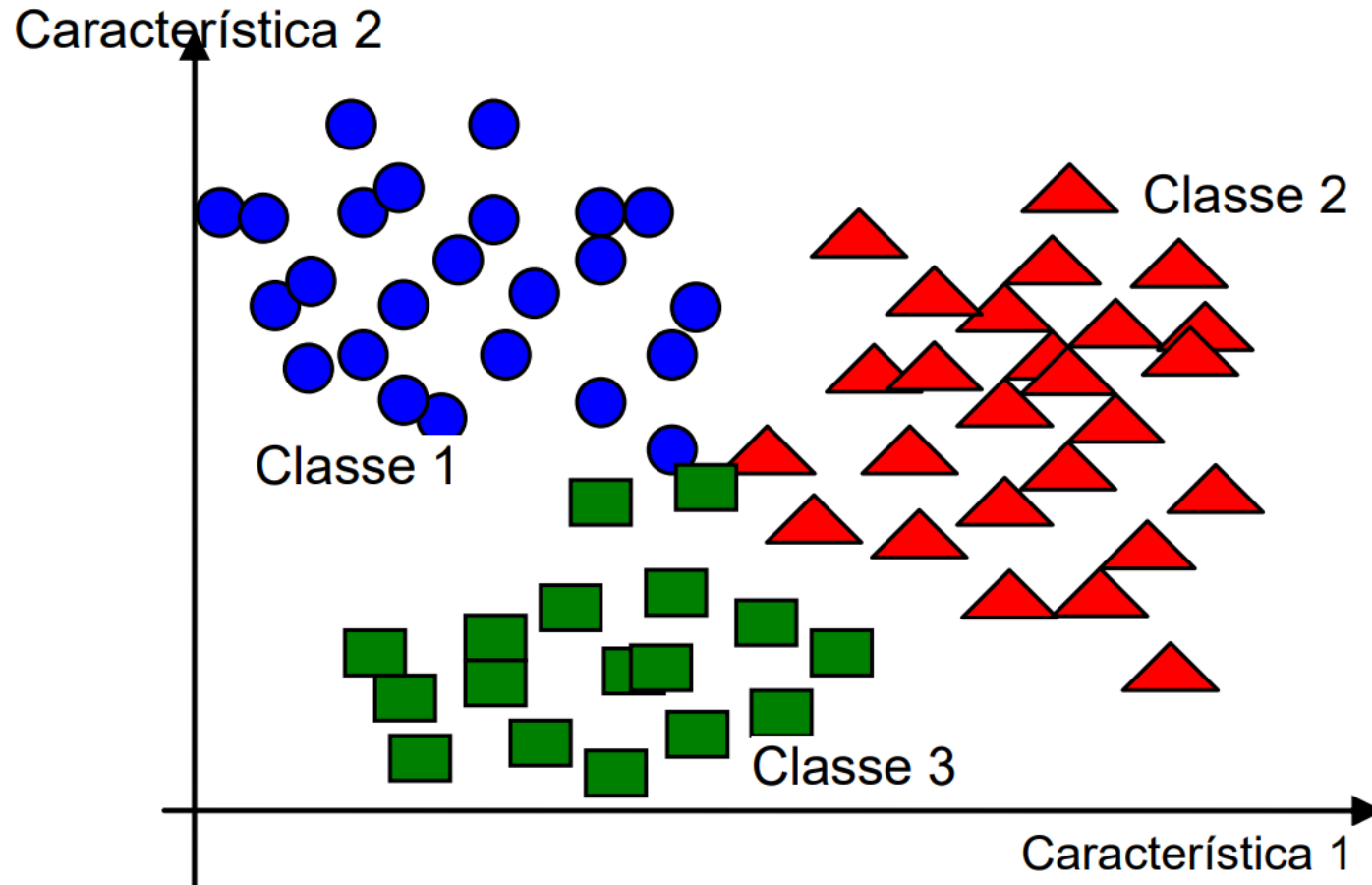
- ❑ Vetores (descrições quantitativas): Exemplo
- Descrever três tipos de flores: Iris setosa, virginia (virginica) e versicolor



3 classes
 W_1, W_2, W_3

- Características:
 - ❑ largura e comprimento de suas pétalas $\longrightarrow x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ x_1 : largura
 x_2 : comprimento
 - ❑ Dado o conjunto de medidas
 - **Vetor de características** torna-se a **representação completa** de cada **amostra física**

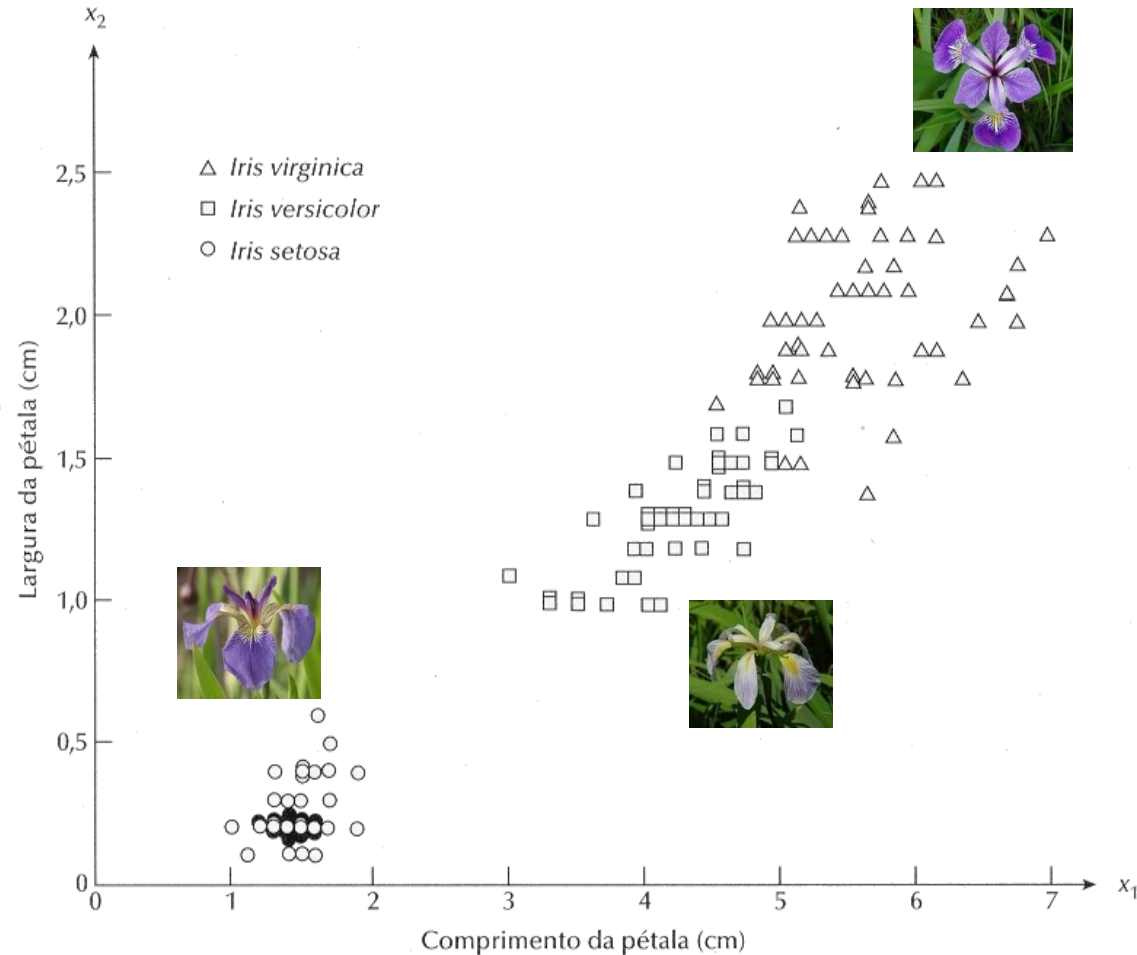
Espaço de Características: Classes



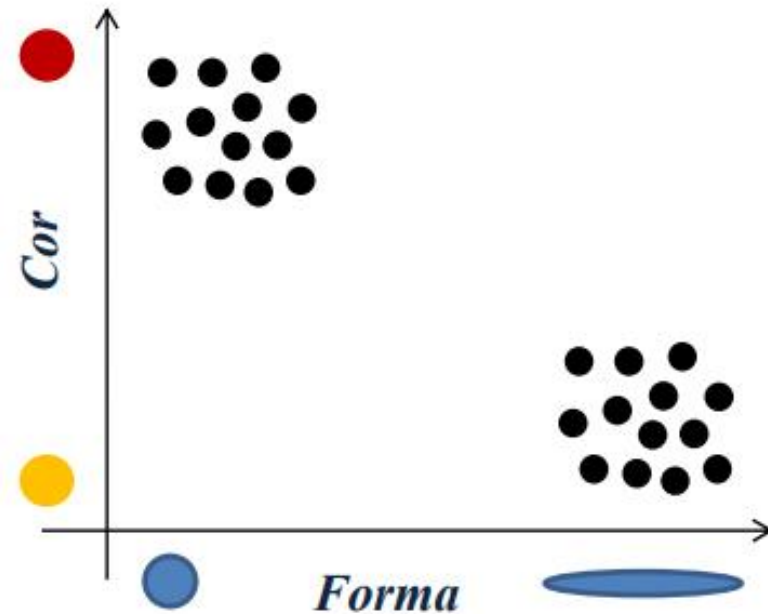
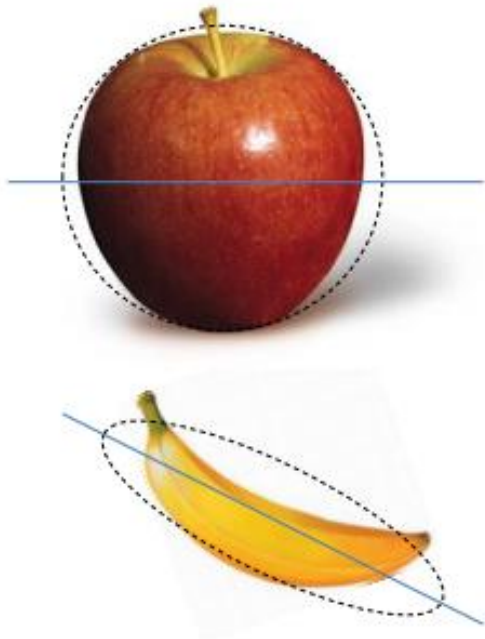
Espaço de Características: Classes

- Vetores (descrições quantitativas):
- **Cada flor do conjunto de amostras de flores é um ponto no espaço euclidiano bi-dimensional**
- **Referência**

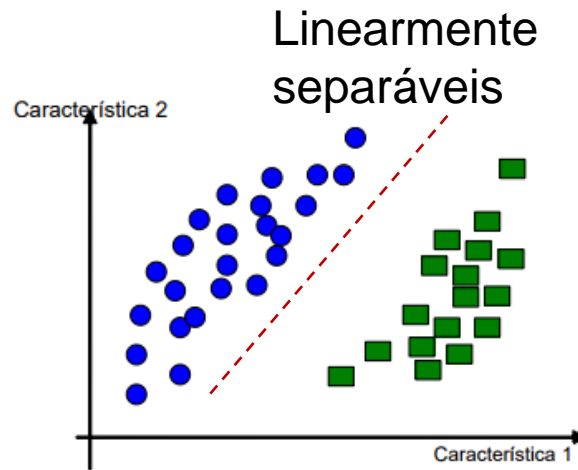
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \begin{array}{l} x_1: \text{largura} \\ x_2: \text{comprimento} \end{array}$$



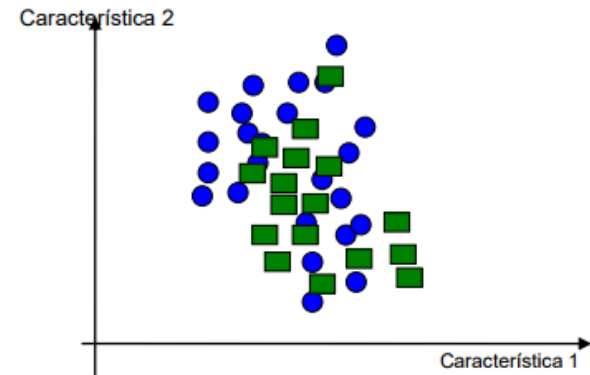
Espaço de Características: Classificação



Espaço de Características: Classificação

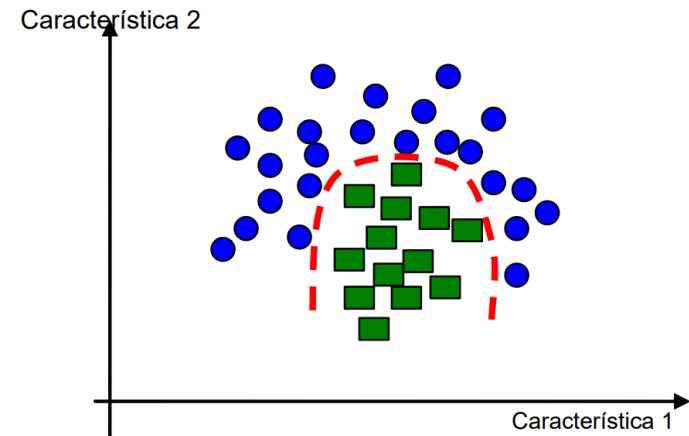


Bom



Ruim

Relevância dos
descritores



Não-linearmente separável

■ Classificação requer:

□ **Descritores**

- Característica (s) de um objeto a ser classificado

□ **Atributo Classe/Categoria**

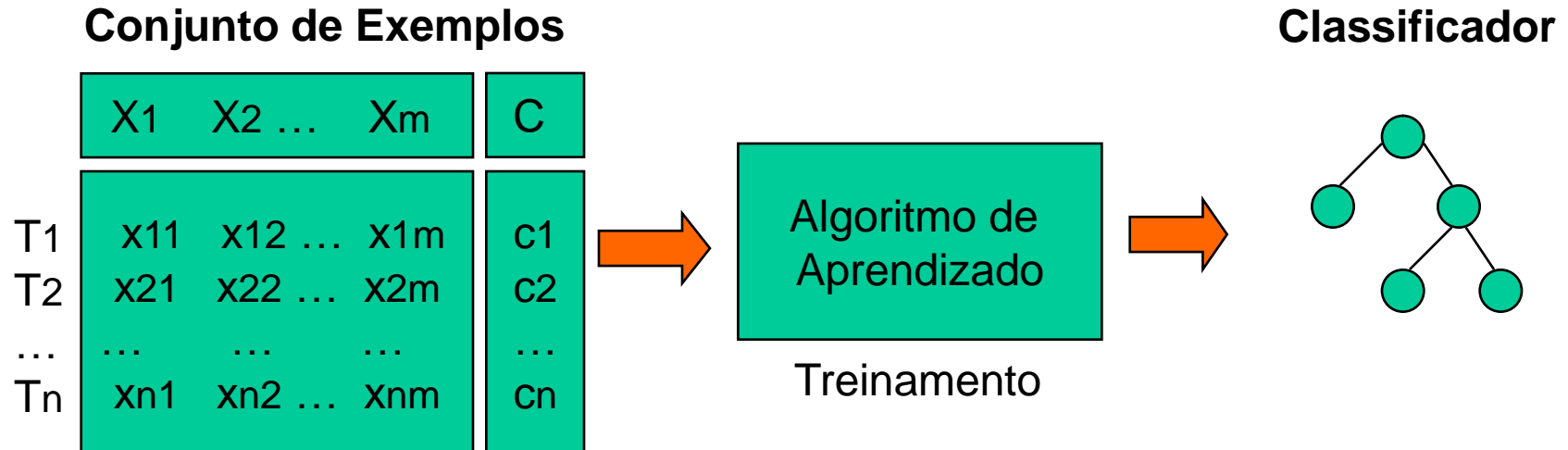
- Atributo alvo da Predição

□ **Exemplos (Treinamento e Teste)**

- Atributos que descrevem um objeto de interesse + classe do exemplo
 - Dados de um paciente e indicação da doença

■ Método de aprendizado

- Algoritmo que adquire conhecimento a partir de um conjunto de exemplos



- Diversidade de algoritmos de aprendizado

- Exemplos

- Aprendizado Baseado em Instâncias
- Árvores de Decisão e Regras
- Redes Neurais Artificiais
- Máquinas de Vetores Suporte
- Aprendizado Bayesiano

Métricas para Avaliação de um Classificador

■ Visualização do Desempenho de um classificador

Matriz de Confusão

		Classe Predita	
		Positivo	Negativo
Classe Verdadeira	Positivo	Verdadeiro + (TP)	Falso - (FN)
	Negativo	Falso + (FP)	Verdadeiro - (TN)

Acertos

Métricas para Avaliação de um Classificador

■ Visualização do Desempenho de um classificador

Matriz de Confusão: Exemplos

		Classe Preditada		
		Positivo	Negativo	
Classe Verdadeira	Positivo	Verdadeiro + (TP)	Falso - (FN)	
	Negativo	Falso + (FP)	Verdadeiro - (TN)	

Acertos

Classe Verdadeira	Classe Preditada			Total Amostras:
	1	2	3	
1	40	7	3	50
2	5	25	5	35
3	2	5	8	15

Acertos

Classe Verdadeira	Classe Preditada		Total Amostras:
	Positivo	Negativo	
Positivo	24	6	30
Negativo	2	18	20

Acertos

Métricas para Avaliação de um Classificador

■ Visualização do Desempenho de um classificador

Matriz de Confusão: Exemplos

Classe Verdadeira	Classe Predita			
	1	2	3	
1	40	7	3	10
2	5	25	5	10
3	2	5	8	8
	7	12	8	

FN (False Negative) is indicated by the red text "FN" above the matrix.
FP (False Positive) is indicated by the red text "FP" below the matrix.
Acertos (73) (Correct Predictions) is indicated by the blue text "Acertos (73)" with an arrow pointing to the diagonal elements (40, 25, 8).

$$\text{Acurácia} = (73 / 100) * 100 = 73\%$$

$$\text{Precisão} = 73 / (73 + 27) = 0,73$$

$$\text{Revocação} = 73 / (73 + 28) = 0,7227$$

$$\text{F-score} = 2 \times (0,73 \times 0,7227) / (0,73 + 0,7227) = 0,7263$$

$$\text{Especificidade} = 73 / (73 + 27) = 0,73$$

Classe Verdadeira	Classe Predita		Total Amostras:50
	Positivo	Negativo	
Positivo	24	6 FN	30
Negativo	2 FP	18	20

Acertos (42) (Correct Predictions) is indicated by the blue text "Acertos (42)" with an arrow pointing to the diagonal elements (24, 18).

$$\text{Acurácia} = (42 / 50) * 100 = 84\%$$

$$\text{Precisão} = 24 / (24 + 2) = 0,9231$$

$$\text{Revocação} = 24 / (24 + 6) = 0,8$$

$$\text{F-score} = 2 \times (0,9231 \times 0,8) / (0,9231 + 0,8) = 0,8572$$

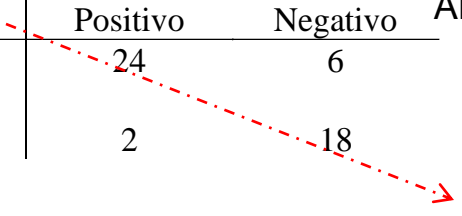
$$\text{Especificidade} = 18 / (18 + 2) = 0,9$$

Acurácia (Taxa de Acerto)

- Indica quanto o modelo acertou das previsões possíveis

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} = \frac{\text{previsões corretas}}{\text{todas as previsões}}$$

		Classe Predit		Total Amostras:50
		Positivo	Negativo	
Classe Verdadeira	Positivo	24	6	
	Negativo	2	18	

 **Acertos**

Exemplo: **Acurácia** = $(42 / 50) * 100 = 84\%$

Sensibilidade (S) ou Revocação/Recall (R)

- Taxa de **acerto na classe positiva**: $recall = \frac{TP}{TP + FN}$

- Medida que permite responder:

		Classe Predita		Total Amostras:50
		Positivo	Negativo	
Classe Verdadeira	Positivo	24	6	→ Acertos
	Negativo	2	18	

- Qual a proporção de positivos que foi identificada corretamente?

Exemplo: **Revocação** = $24 / (24 + 6) = 0,8$

- Quão bom meu modelo é para prever positivos, sendo positivo entendido como a classe verdadeira (real)

- Quanto mais sensível é um teste, maior sua capacidade de detectar o caso (doença, por exemplo)

Especificidade (E)

- Taxa de acerto na classe negativa: $E = \frac{TN}{TN + FP}$

- Permite responder:

		Classe Predita		Total Amostras:50
		Positivo	Negativo	
Classe Verdadeira	Positivo	24	6	
	Negativo	2	18	

→ Acertos

- Qual a capacidade do sistema em prever corretamente a ausência da condição em casos negativos (real)?

Exemplo: **Especificidade** = $18 / (18 + 2) = 0,9$

- Testes específicos são úteis para:
 - Confirmar a ausência da condição (doença, por exemplo)
 - Evitar tratamento desnecessário, no exemplo da doença

Precisão (Pr)

- **Proporção de exemplos positivos classificados corretamente entre todos aqueles preditos como positivos**

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

		Classe Predita		Total Amostras:50
		Positivo	Negativo	
Classe Verdadeira	Positivo	24	6	Acertos
	Negativo	2	18	

Exemplo: Precisão = $24 / (24 + 2) = 0,9231$

- **Permite responder (exatidão do modelo):**
 - Qual a proporção de identificações positivas que está realmente correta?

Medida F_1 ou F-score

■ Média harmônica entre precisão e revocação

$$\text{Medida-F} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

	Classe Predita		Total Amostras:50
	Positivo	Negativo	
Classe Verdadeira	Positivo	24	6
	Negativo	2	18

Acertos

Exemplo: F-score = $2 \times (0,9231 \times 0,8) / (0,9231 + 0,8) = 0,8572$

Precisão = $24 / (24 + 2) = 0,9231$: Revocação = $24 / (24 + 6) = 0,8$

■ Medida de confiabilidade da Acurácia

- Valor alto de Medida F_1 indica que a **acurácia é relevante**
- Valores de TP, TN, FP e FN **não apresentam distorções significativas**

Espaço ROC ou Curva ROC (*Receiver Operator Characteristic Curve*)

- Permite representar graficamente a **relação entre falso positivos e verdadeiros positivos**

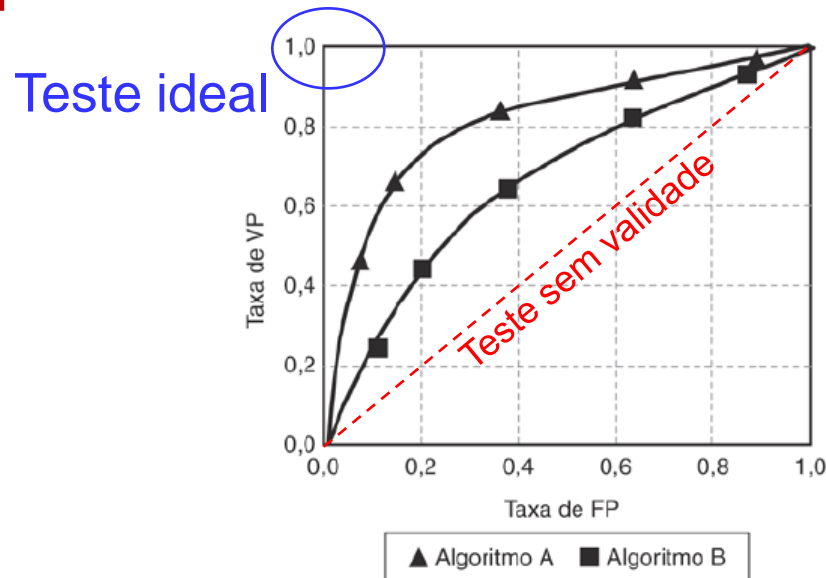


Figura 9.4 Exemplos de curva ROC.

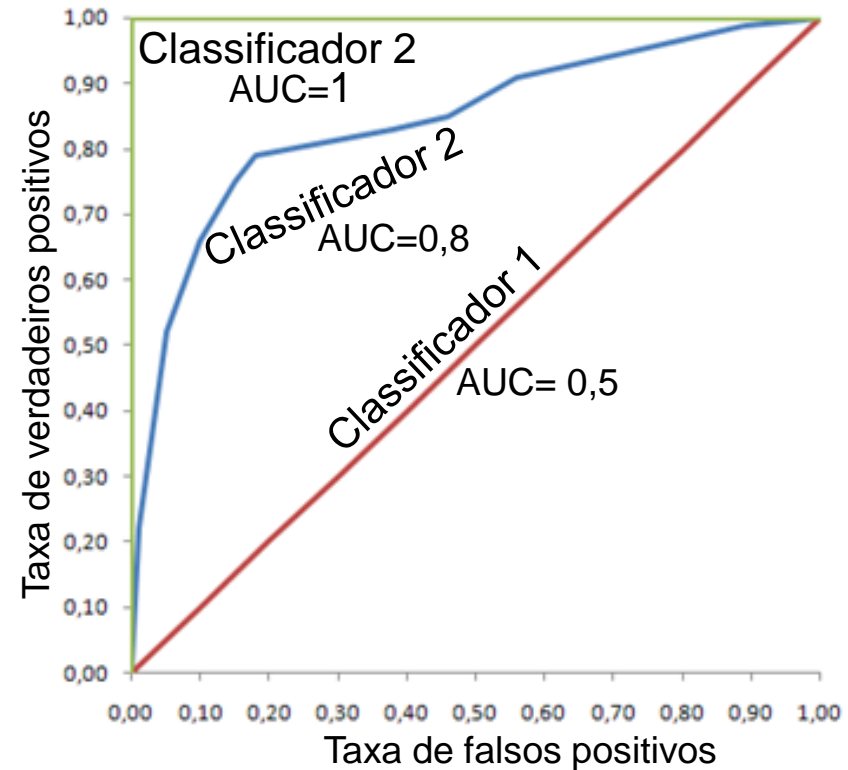
- Dada por:
 - Eixo X: Taxa de falsos positivos (TFP ou 1-Especificidade)
 - Eixo Y: Taxa de verdadeiros positivos (TVP ou Sensibilidade)

Espaço ROC ou Curva ROC (*Receiver Operator Characteristic Curve*)

■ Ponto de corte

- Pode ser selecionado arbitrariamente pelo analista
- Para cada ponto escolhido, calcular TVP e TFP

■ AUC evita a escolha de um ponto específico



Melhor Ponto de corte:
Maiores Taxas de verdadeiros e falsos positivos

AUC: (Area Under ROC Curve), via integração numérica

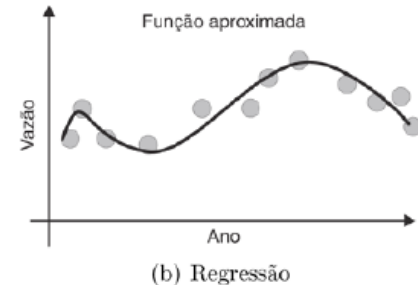
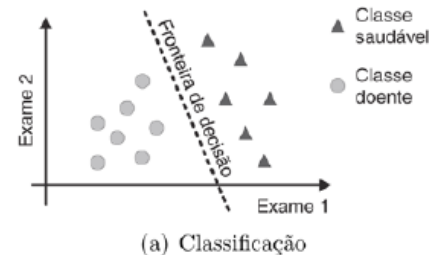
https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-9863-7_209

Modelos Preditivos

■ Algoritmo de Aprendizado de máquina preditivo:

- **Função** que considera um conjunto de dados **rotulados**

- Produz um **estimador**



- Domínio com valores nominais: Problema de Classificação

- Estimador é um **classificador**

- Domínio é um conjunto infinito e ordenado de valores: Problema de regressão

- Induz um **regressor**

Figura 3.1 Gráfico ilustrativo das tarefas.

Modelos Preditivos

Algoritmo de Aprendizado de máquina preditivo:

- **Classificação:** $y_i = f(x_i) \in \{c_1, \dots, c_m\}$
 - Assume valores em um conjunto discreto

- **Regressão:** $y_i = f(x_i) \in \mathbb{R}$
 - Assume valores em um conjunto infinito e ordenado de valores

Tabela 3.12 Exemplo de conjunto de dados para problema de classificação

Tamanho (P)	Largura (P)	Tamanho (S)	Largura (S)	Espécie
5,1	3,5	1,4	0,2	Setosa
4,9	3,0	1,4	0,2	Setosa
7,0	3,2	4,7	1,4	Versicolor
6,4	3,2	4,5	1,5	Versicolor
6,3	3,3	6,0	2,5	Virgínica
5,8	2,7	5,1	1,9	Virgínica

Tabela 3.13 Exemplo de conjunto de dados para problema de regressão

Fertilidade	Agricultura	Educação	Renda	Mortalidade
80,2	17,0	12	9,9	22,2
83,1	45,1	9	84,8	22,2
92,5	39,7	5	93,4	20,2
85,8	36,5	7	33,7	20,3
76,9	43,5	15	5,2	20,6

Generalização e Overfitting

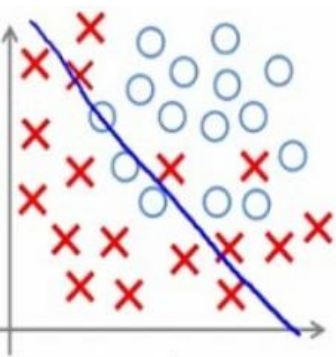
- **Objetivo da Classificação:** definir corretamente novos casos
 - Casos diferentes dos utilizados no treinamento
- **Problemas: Baixa capacidade Generalização** (hipóteses)
 - **Overfitting** aos dados de treinamento, alta taxa de acerto
 - **Underfitting** baixa taxa de acerto, mesmo no subconjunto de treinamento

Causa: exemplos não são representativos ou modelo simples

Generalização e Overfitting

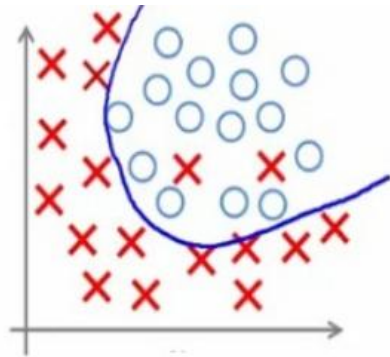
- ❑ **Modelo** construído via **memorização** de exemplos
 - Logo, processo que busca
 - ❑ **Overfitting controlado:** Alguns modelos mais do que outros
- ❑ **Dilema:**
 - Usar um procedimento que não realiza **overfitting**?
 - ❑ Todos utilizam
 - **Complexidade do Modelo x Possibilidade de Overfitting**

Generalização e Overfitting

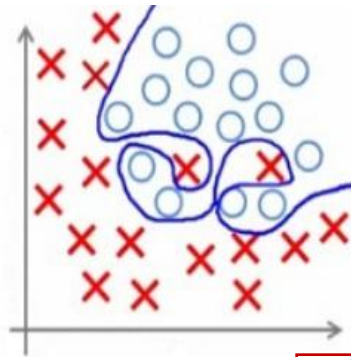


Underfitting

(too simple to explain the variance)

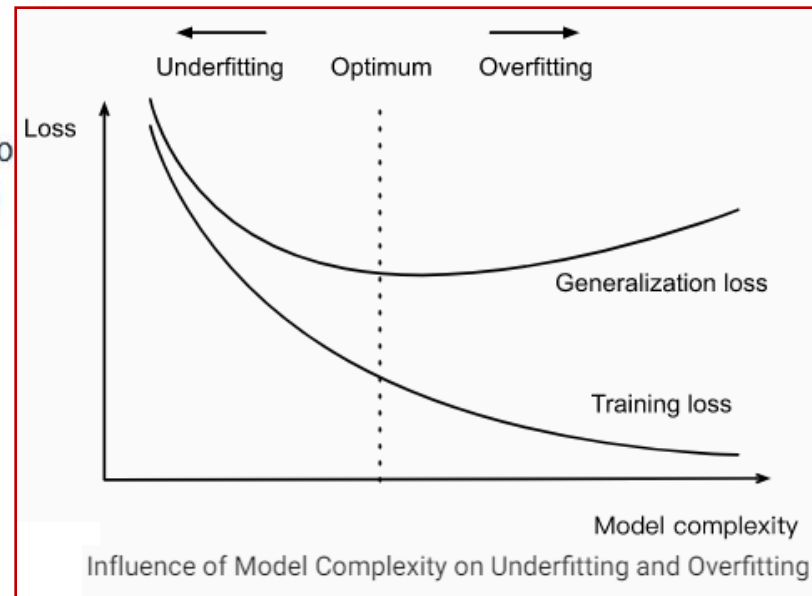


Appropriate-fitting



Overfitting

(forcefitting -- too good to be true)



Complexidade do modelo: Definida por diferentes maneiras

http://d2l.ai/chapter_multilayer-perceptrons/underfit-overfit.html

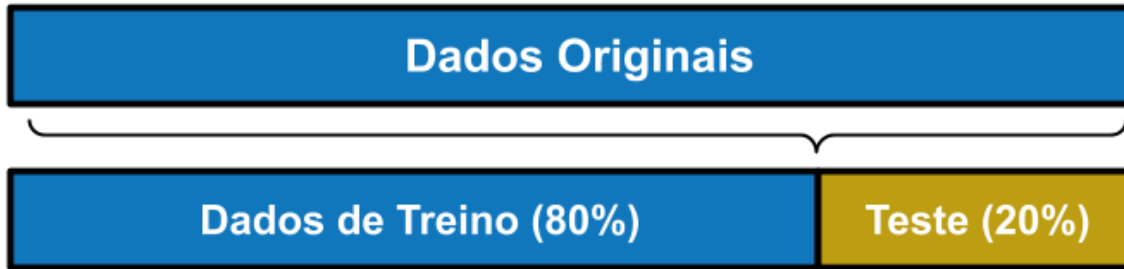
Generalização: Métodos de Amostragem

■ Estratégias

- Permite aferir a qualidade do classificador e do espaço de características
 - Hold-out
 - Repeated Hold-out
 - Validação Cruzada k-folds (cross-validation k-folds) e Validação Cruzada Estratificada
 - Leave-one-out

Generalização: Métodos de Amostragem

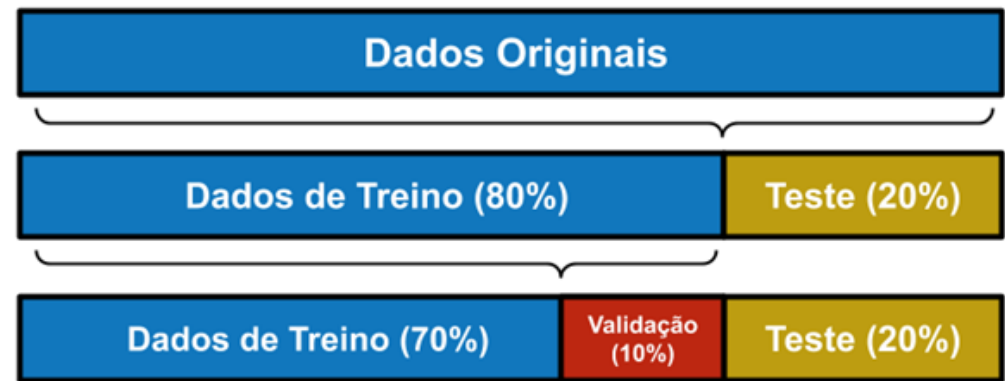
Divisão Treino e Teste simples (Hold-out)



Distribuições:

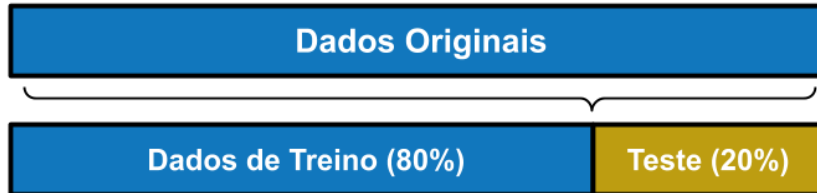
- 80/20
- 70/30

Otimizar hiperparâmetros de um modelo
Comparar diferentes modelos



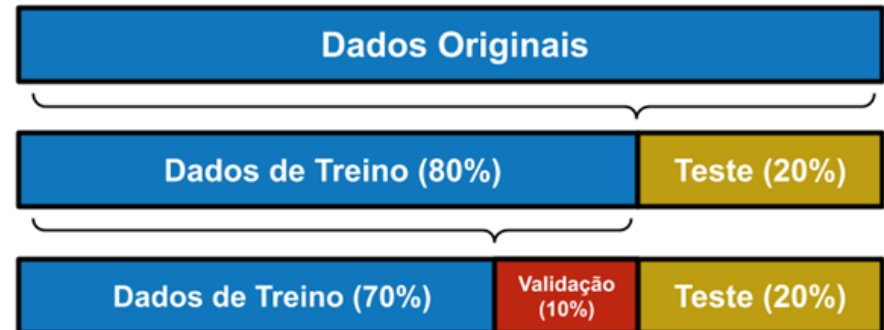
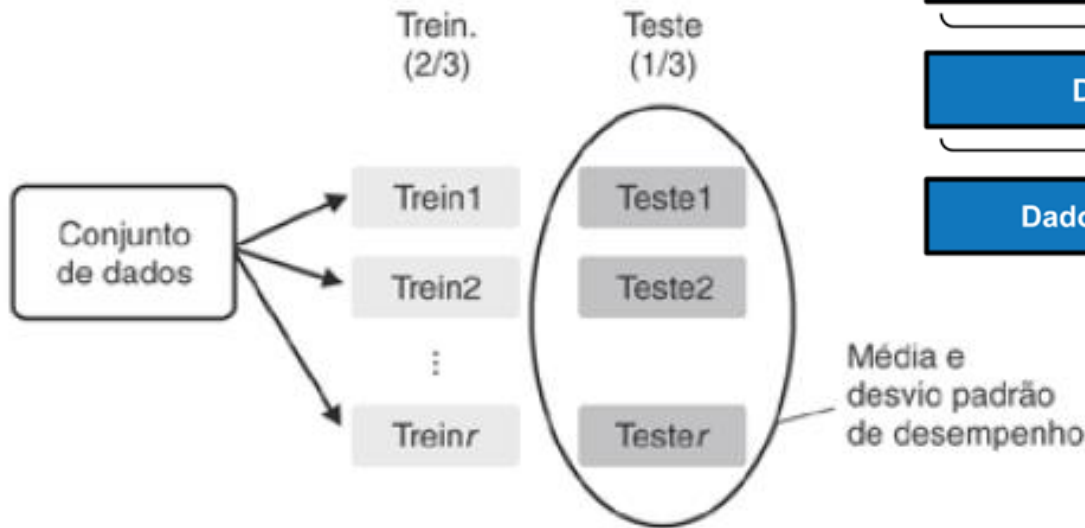
Generalização: Métodos de Amostragem

Divisão Treino e Teste simples (Hold-out)



Distribuições:

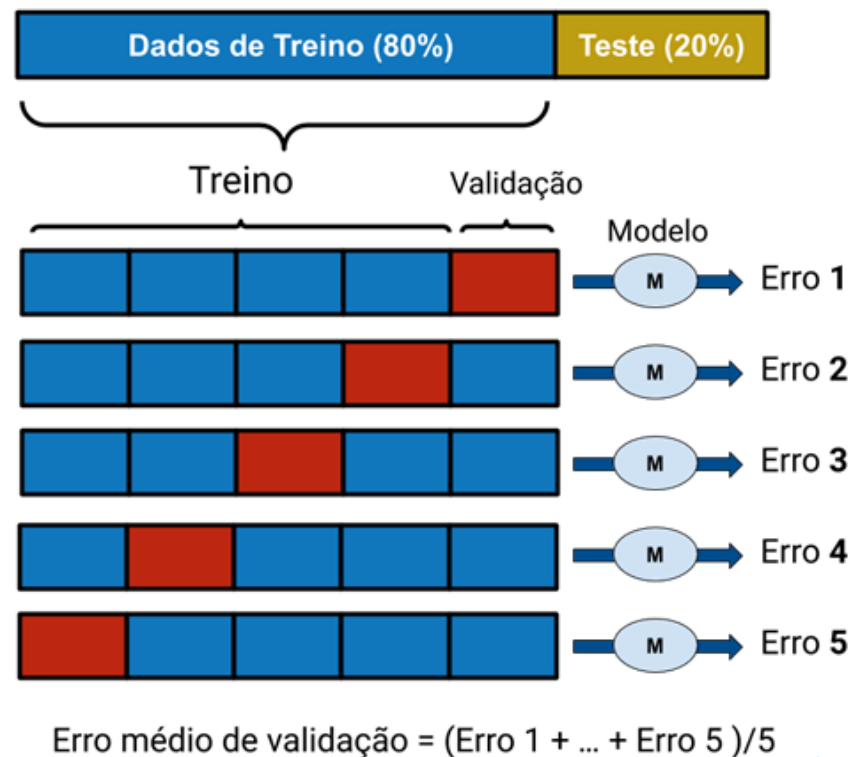
- 80/20
- 70/30



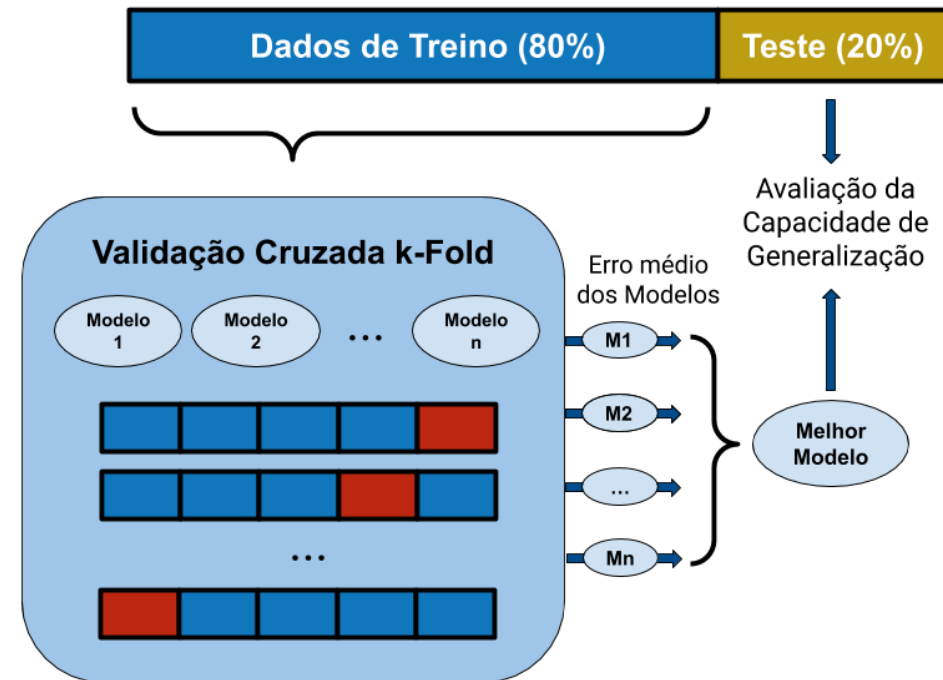
Amostragem aleatória
Repeated Holdout Method

Generalização: Métodos de Amostragem

Validação Cruzada (K-fold Validation)



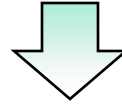
Comparação de Modelos



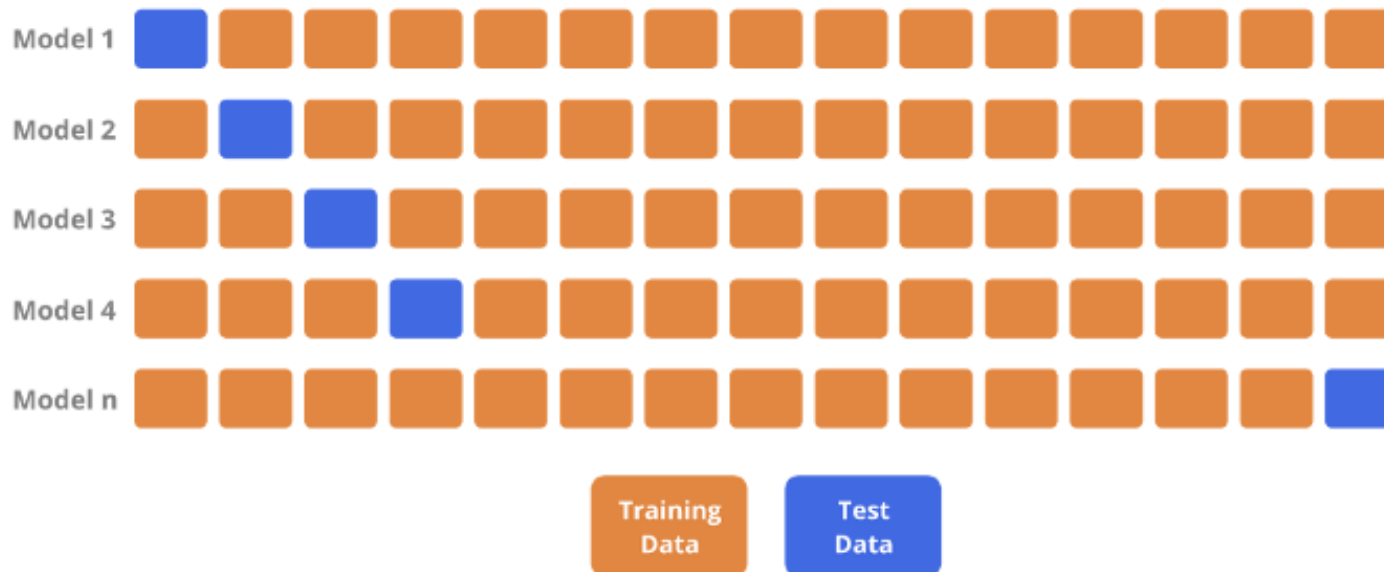
* **Validação Cruzada Estratificada**, mantém em cada partição a proporção de exemplos de cada classe semelhante à proporção do conjunto original

Generalização: Métodos de Amostragem

K-fold Validation com $K = n$, em que n representa o número de amostras



Leave-One-Out Cross Validation



Generalização e Overfitting

■ Número de Descritores

- ❑ Quanto mais, melhor?
- ❑ O desempenho **começa melhorando**, entretanto, **piora** conforme o **número de características cresce**
- ❑ **Limitar o espaço de características**

Generalização e Overfitting

■ Amostragem dos Dados: Cuidados

□ Dados desbalanceados

- Problemas com a Indução do Modelo para uma classe

□ Limpeza de Dados

- Dados Inconsistentes, Incompletos, com Ruídos ou Redundantes

□ Normalização

□ Dimensionalidade

PDI Normalização

■ Normalização dos Dados:

- Limites de valores de atributos distintos são muito diferentes

- Magnitude maior

- Pode ocorrer **preponderância na classificação**

- Quando recomendada, normalizar:

- Amplitude

- Reescala e padronização

$M =$	0.11	1001	Classe 1
	0.12	1005	
	0.07	1003	
	0.09	1004	
	0.10	1005	
	0.99	1001	Classe 2
	0.97	1007	
	0.96	1004	
	0.98	1006	
	0.95	1002	

PDI Normalização

■ Amplitude: Reescala ou min-max

$$v_{novo} = \mathbf{min} + \frac{v_{atual} - \mathbf{menor}}{\mathbf{maior} - \mathbf{menor}} (\mathbf{max} - \mathbf{min})$$

em que, *menor*, menor valor do atributo; *maior*, maior valor do atributo;

min, menor valor da nova escala; *max*, maior valor da nova escala;

v_{atual} , valor a ser normalizado

Considerando 0.11, temos $= -1 + (0,11 - 0,07) / (0,99 - 0,07) * (1 - (-1))$

$$M = \begin{bmatrix} 0.11 & 1001 \\ 0.12 & 1005 \\ 0.07 & 1003 \\ 0.09 & 1004 \\ 0.10 & 1005 \\ 0.99 & 1001 \\ 0.97 & 1007 \\ 0.96 & 1004 \\ 0.98 & 1006 \\ 0.95 & 1002 \end{bmatrix} \xrightarrow{\text{Considerando } \mathbf{min} = -1 \text{ e } \mathbf{max} = 1, \text{ temos: } M_{norm} =} \begin{bmatrix} -0.9130 & -1.0000 \\ -0.8913 & 0.3333 \\ -1.0000 & -0.3333 \\ -0.9565 & 0 \\ -0.9348 & 0.3333 \\ 1.0000 & -1.0000 \\ 0.9665 & 1.0000 \\ 0.9348 & 0 \\ 0.9783 & 0.6667 \\ 0.9130 & -0.6667 \end{bmatrix}$$

PDI Normalização

■ Amplitude: Padronização (zscore)

- Considera uma medida de localização (μ) e uma medida de escala (σ):

$$v_{novo} = \frac{v_{atual} - \mu}{\sigma}$$

em que, μ é a média; σ é o desvio padrão; v_{atual} é o valor a ser normalizado.

$M =$	0.11	1001	$\xrightarrow{\text{Considerando } \mu=0,534 \text{ e } \sigma=0,4598, \text{ temos:}}$	$M_{norm} =$	-0.9220	-1.3699
	0.12	1005			-0.9002	0.5871
	0.07	1003			-1.0089	-0.3914
	0.09	1004			-0.9655	0.0978
	0.10	1005			-0.9437	0.5871
	0.99	1001			0.9916	-1.3699
	0.97	1007			0.9481	1.5656
	0.96	1004			0.9263	0.0978
	0.98	1006			0.9698	1.0763
	0.95	1002			0.9046	-0.8806

Feature 1

Feature 2

Padronização é melhor para casos com outliers

Dimensionalidade

■ Dimensionalidade

- Se cada descritor é uma coordenada em um **espaço d -dimensional** $\Rightarrow d$ é o número de atributos
 - Pode ocorrer um **espaço de característica altamente complexo**
 - **Maldição da Dimensionalidade**

Dimensionalidade



Figure 2. A single feature does not result in a perfect separation of our training data.

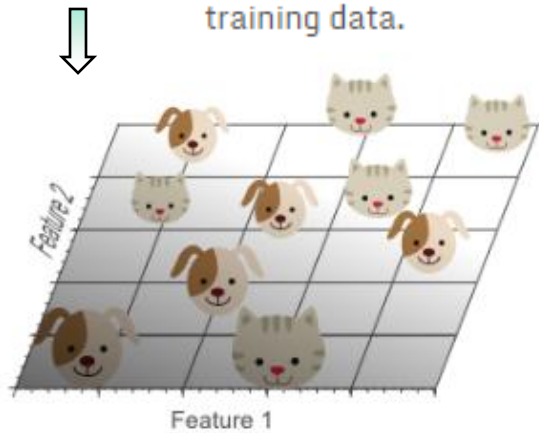


Figure 3. Adding a second feature still does not result in a linearly separable classification problem: No single line can separate all cats from all dogs in this example.

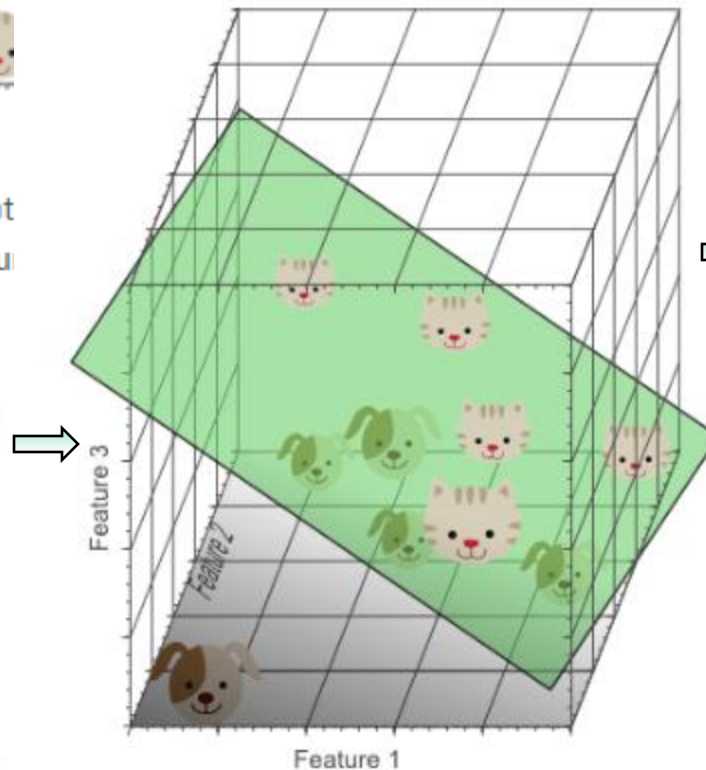


Figure 5. The more features we use, the higher the likelihood that we can successfully separate the classes perfectly.

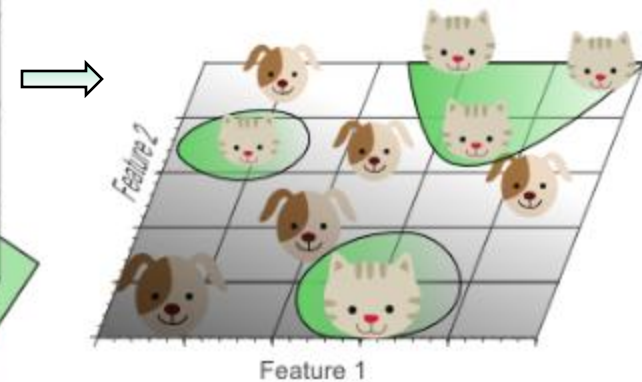


Figure 6. Using too many features results in overfitting. The classifier starts learning exceptions that are specific to the training data and do not generalize well when new data is encountered.

Dimensionalidade

- Considere um conjunto de Dados
 - Cada objeto do conjunto tem um atributo
 - O atributo pode assumir um dentre 10 valores
 - Logo, $10^1 \implies$ Conjunto pode ter 10 objetos diferentes (número de amostras)
 - E se o número de atributos for igual a 5? Aumento exponencial: 10^5
 - É necessário garantir modelos representativos:
 - Número de amostras deve crescer exponencialmente

Dimensionalidade

- Problema:

- Teorema do “**patinho feito**”

- Watanabe, 1985:

- “...dois padrões arbitrários podem ficar similares se esses forem codificados com um número suficientemente grande de características similares.”

Dimensionalidade

- ❑ Número de exemplos não aumenta na mesma proporção que aumenta o número de características
- Chamado de **“Fenômeno de Máximo”**
 - ❑ *“...a maioria dos classificadores paramétricos geralmente estima parâmetros não conhecidos e liga-os com parâmetros verdadeiros nas densidades de classe-condicional. Em uma amostra de tamanho fixo, quando o número de características cresce (à medida que aumenta o número de parâmetros desconhecidos) a confiança dos parâmetros estimados decresce. Consequentemente, o desempenho dos classificadores, para uma amostra de tamanho fixo, pode degradar com o aumento do número de características...”*

SUNG, K.-K.; POGGIO, Tomaso. Example-based learning for view-based human face detection. **IEEE Transactions on pattern analysis and machine intelligence**, v. 20, n. 1, p. 39-51, 1998.

BIANCHI, Marcelo Franceschi de. **Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA**. 2006. Tese de Doutorado. Universidade de São Paulo.

Dimensionalidade

- ❑ Jain et al., 2000:
 - ❑ “No mínimo, deve-se utilizar um número de exemplos de treinamento por classe dez vezes maior que a dimensionalidade”

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37.

Dimensionalidade

- ❑ Aumentar o número de exemplos é uma tarefa fácil?
- ❑ Quais os problemas?

Dimensionalidade

- **Outro problema: Relevância do descritor**
 - Um conjunto descrito por 20 atributos
 - **Apenas 2 atributos são relevantes**
 - Os demais são atributos ruins ou correlacionados
 - **Resultado: a classificação é afetada**

Redução da Dimensionalidade

- ❑ Redução do número de atributos
- ❑ Aumento do número de exemplos

Redução da Dimensionalidade

- Espera-se que todos os atributos sejam relevantes
 - Mas, nem sempre é possível garantir isso
- Alguns atributos são redundantes e podem ser eliminados
- Objetivo:
 - Definir conjunto de atributos que sejam relevantes e não-redundantes

Redução da Dimensionalidade

■ Abordagens

- Avaliação (Seleção) X Agregação

■ Seleção de atributos:

- Escolha de um subconjunto de atributos relevantes dentre os atributos disponíveis
 - Abordagens Embutidas (Embedded), Filtros e Wrappers

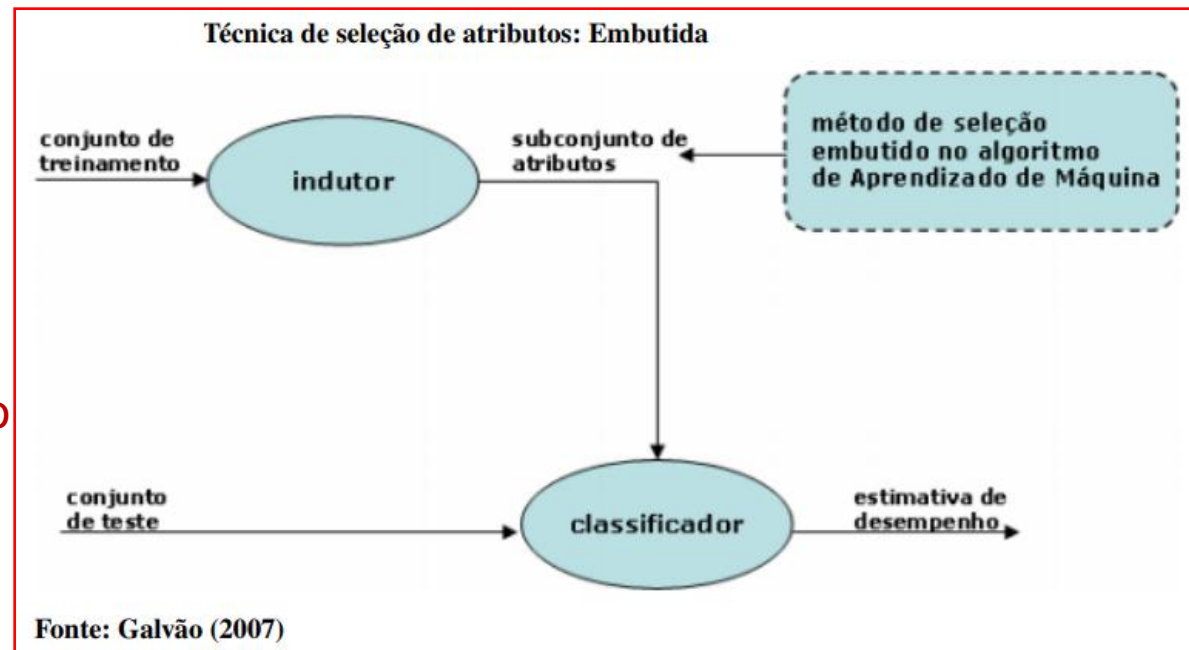
■ Agregação:

- Combinação dos atributos existentes por meio de funções lineares ou não lineares
 - Eliminação de redundâncias
- Exemplo, Análise de Componentes Principais
 - Levam à perda dos valores originais

Redução da Dimensionalidade

■ Métodos Embutidos (Embedded)

- Subconjunto de atributos é selecionado no próprio processo de construção do modelo de classificação
 - Ocorre durante a etapa de treinamento
 - Geralmente são específicos para um dado algoritmo de classificação



Exemplo:

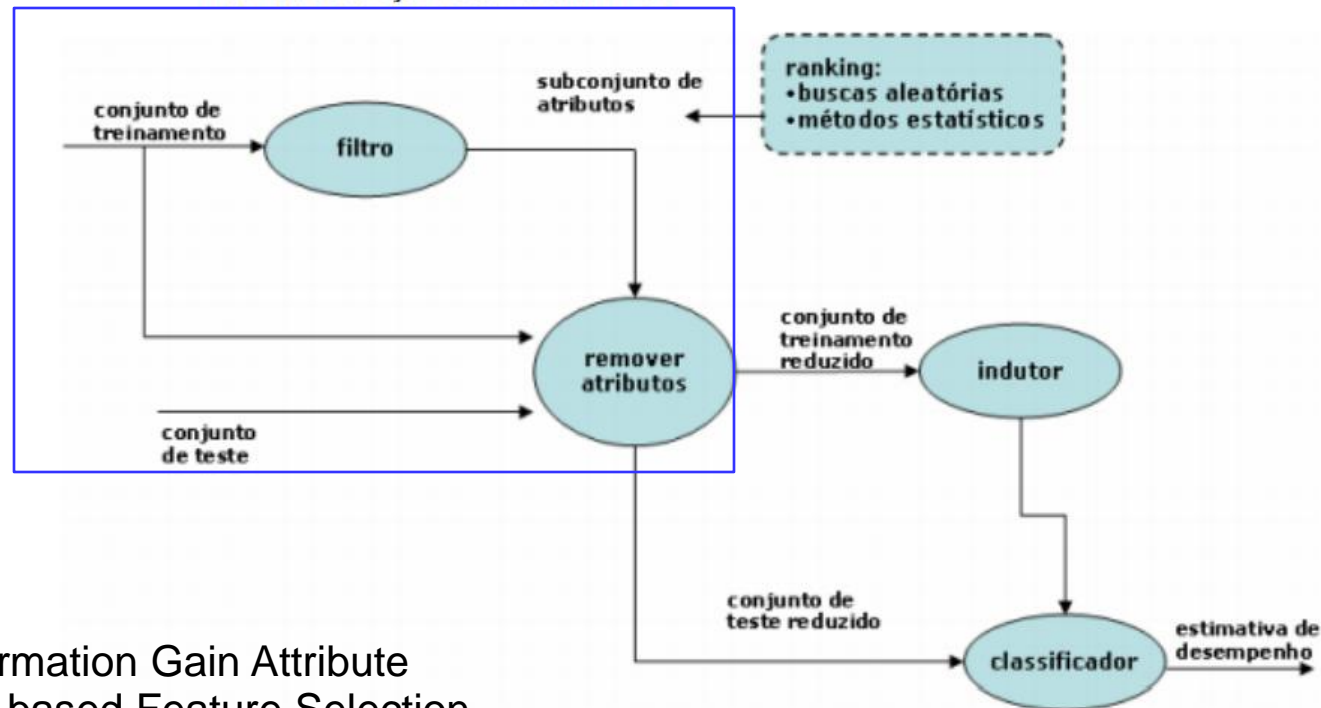
- indução de árvores de decisão

Redução da Dimensionalidade

■ Filtros

- Atributos são **ordenados** com base em **métricas** de relevância e redundância

Técnica de seleção de atributos: Filtro



- Retorna os atributos mais relevantes

- Não depende do classificador

- Exemplos: Relief, Information Gain Attribute Ranking e Correlation-based Feature Selection

Fonte: Adaptado de Galvão (2007)

- http://repositorio.utfpr.edu.br/jspui/bitstream/1/3334/1/PG_PPGCC_M_Almeida%2C%20Thissiany%20Beatriz_2018.pdf

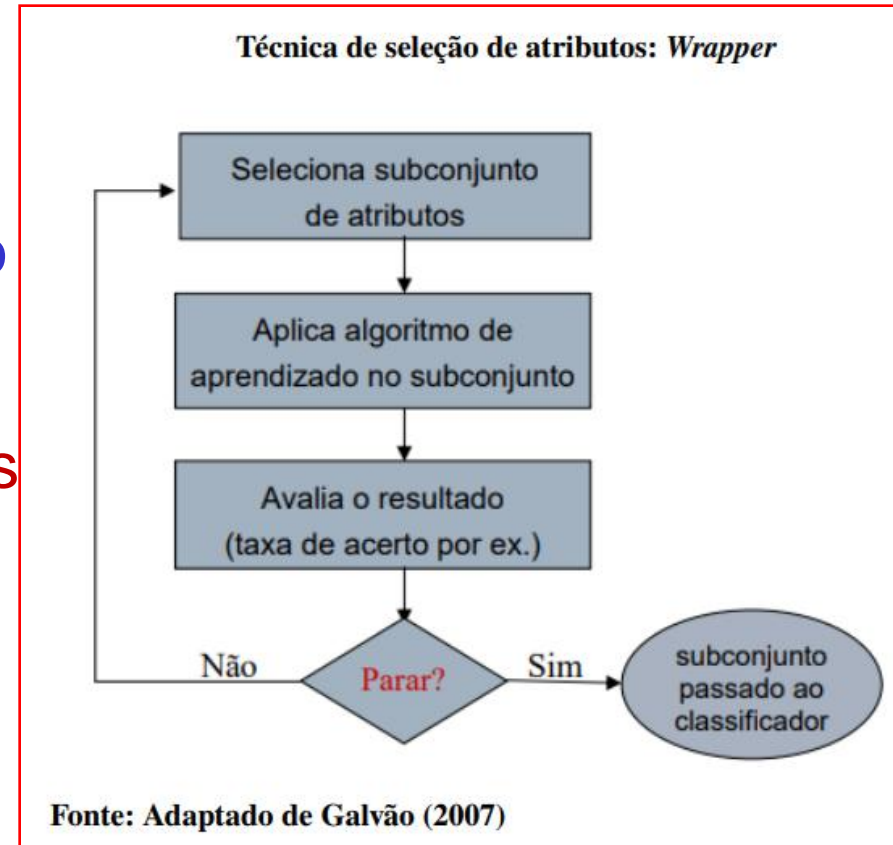
Redução da Dimensionalidade

- **Filtros** de atributos: **baixo custo computacional**
- Porém, apresenta dificuldades:
 - Nem sempre é fácil **definir quantos atributos devem ser descartados**
 - Na prática, isso é **definido por tentativa-e-erro ou heurística**
 - Não considera o classificador

Redução da Dimensionalidade

■ Wrappers

- ❑ O desempenho (uso de uma métrica, por exemplo **acurácia**) do **algoritmo** é avaliado para diferentes **sub-conjuntos de atributos**
- ❑ O melhor subconjunto encontrado é retornado
 - Depende do Classificador
 - Elevado custo computacional



Uso de meta-heurísticas para eliminar buscas exaustivas

Redução de Dimensionalidade - Exemplo



Expert Systems with Applications
Volume 120, 15 April 2019, Pages 262-278



Classification of colorectal cancer based on the association of multidimensional and multiresolution features

Matheus Gonçalves Ribeiro ^a, Leandro Alves Neves ^a, Marcelo Zanchetta do Nascimento ^b, Alessandro Santana Martins ^c, Thaína Aparecida Azevedo Tosta ^d

[Show more](#)

<https://doi.org/10.1016/j.eswa.2018.11.034>

Existe um número máximo de características a partir do qual o desempenho do classificador irá degradar

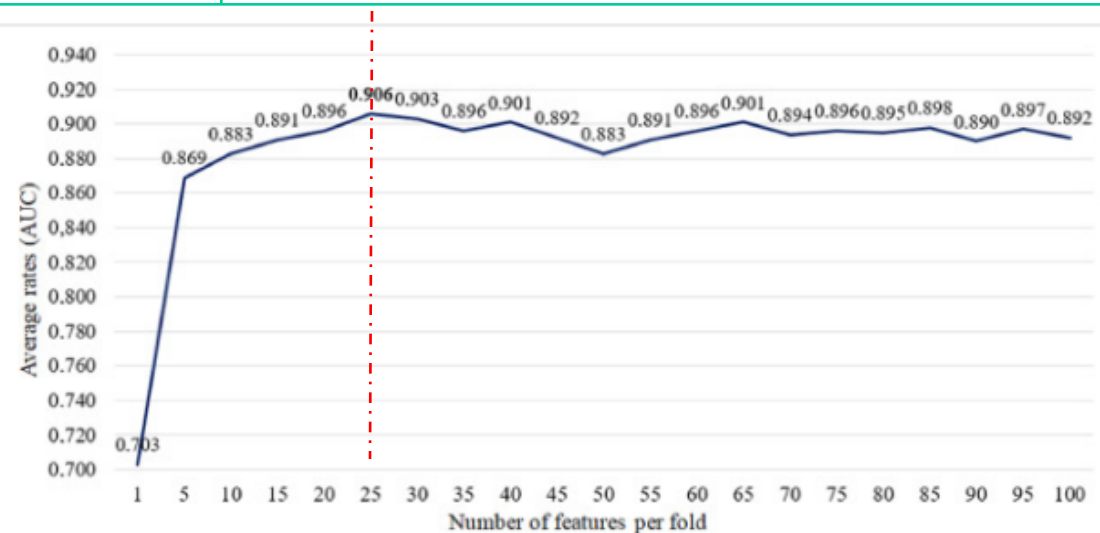
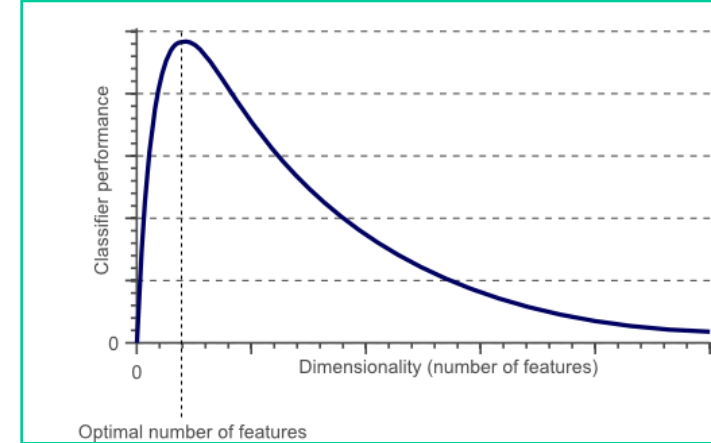


Fig. 9. Average rates for AUC calculated from the folds and applying the methods DT, SVM, NB, RaF, K* and PL. The values were obtained with subsets of 1 to 100 features, limits applied to each fold.

Redução de Dimensionalidade - Exemplo



Expert Systems with Applications
Volume 120, 15 April 2019, Pages 262-278



Classification of colorectal cancer based on the association of multidimensional and multiresolution features

Matheus Gonçalves Ribeiro ^a, Leandro Alves Neves ^a, Marcelo Zanchetta do Nascimento ^b, Guilherme Freire Roberto ^b, Alessandro Santana Martins ^c, Thaina Aparecida Azevedo Tosta ^d

Show more

<https://doi.org/10.1016/j.eswa.2018.11.034>

Get rights and content

Exemplo de Estratégia de Validação: Generalização

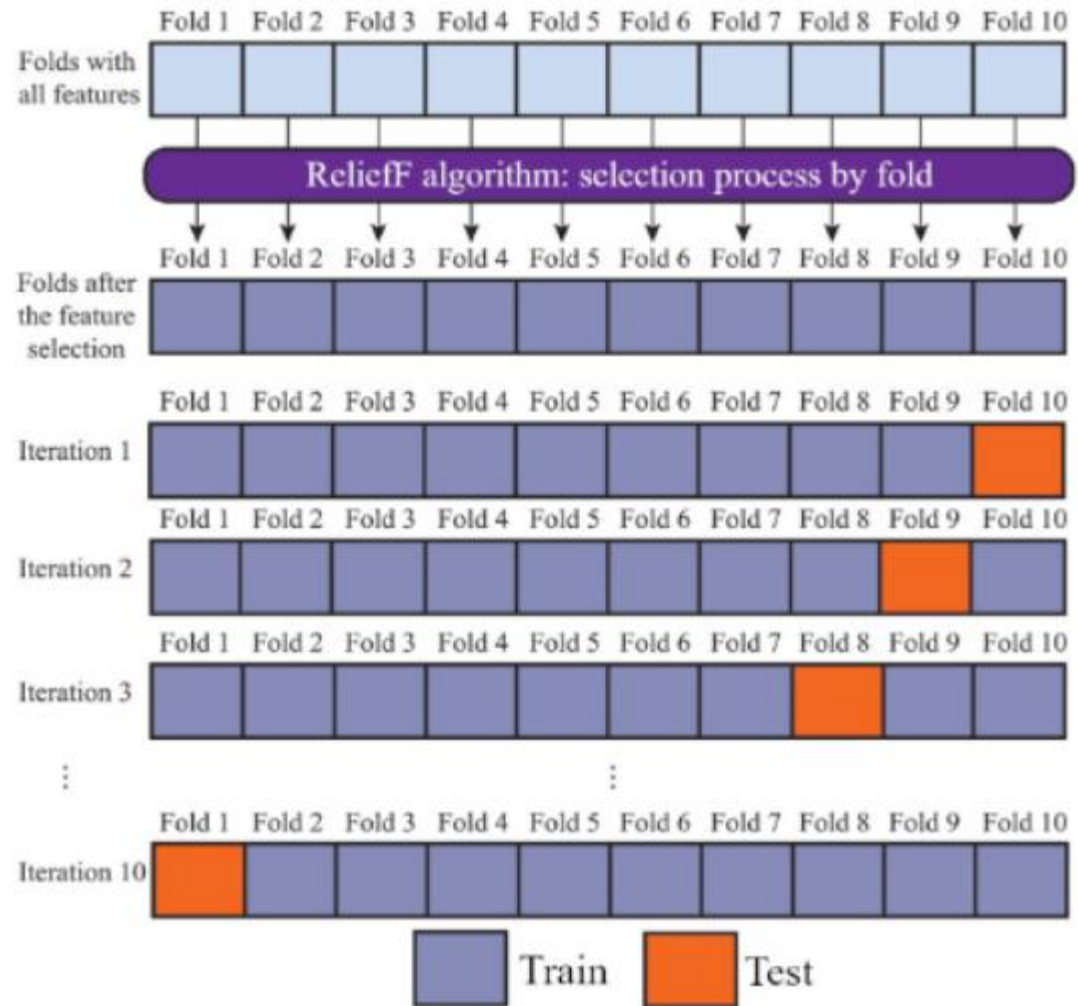




Fig. 3. Illustration of the cross-validation and selection processes by folds applied in the proposed method.

Redução de Dimensionalidade - Exemplo

 Expert Systems with Applications
Volume 120, 15 April 2019, Pages 262-278



Classification of colorectal cancer based on the association of multidimensional and multiresolution features

Matheus Gonçalves Ribeiro ^a, Leandro Alves Neves ^a, Marcelo Zanchetta do Nascimento ^b, Guilherme Freire Roberto ^b, Alessandro Santana Martins ^c, Thaina Aparecida Azevedo Tosta ^d

[Show more](#)

<https://doi.org/10.1016/j.eswa.2018.11.034>

[Get rights and content](#)

Exemplo de Estratégia de Validação: Generalização

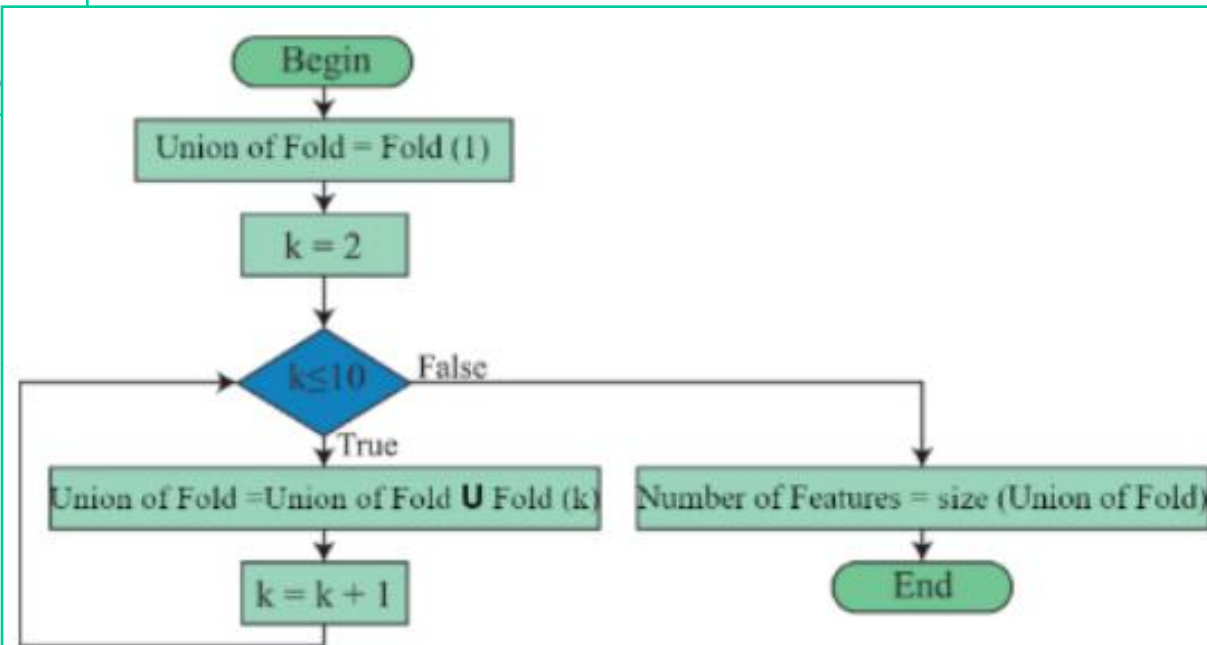




Fig. 4. Flowchart of the feature counting without repetitions.

Redução de Dimensionalidade - Exemplo

 Expert Systems with Applications
Volume 120, 15 April 2019, Pages 262-278



Classification of colorectal cancer based on the association of multidimensional and multiresolution features

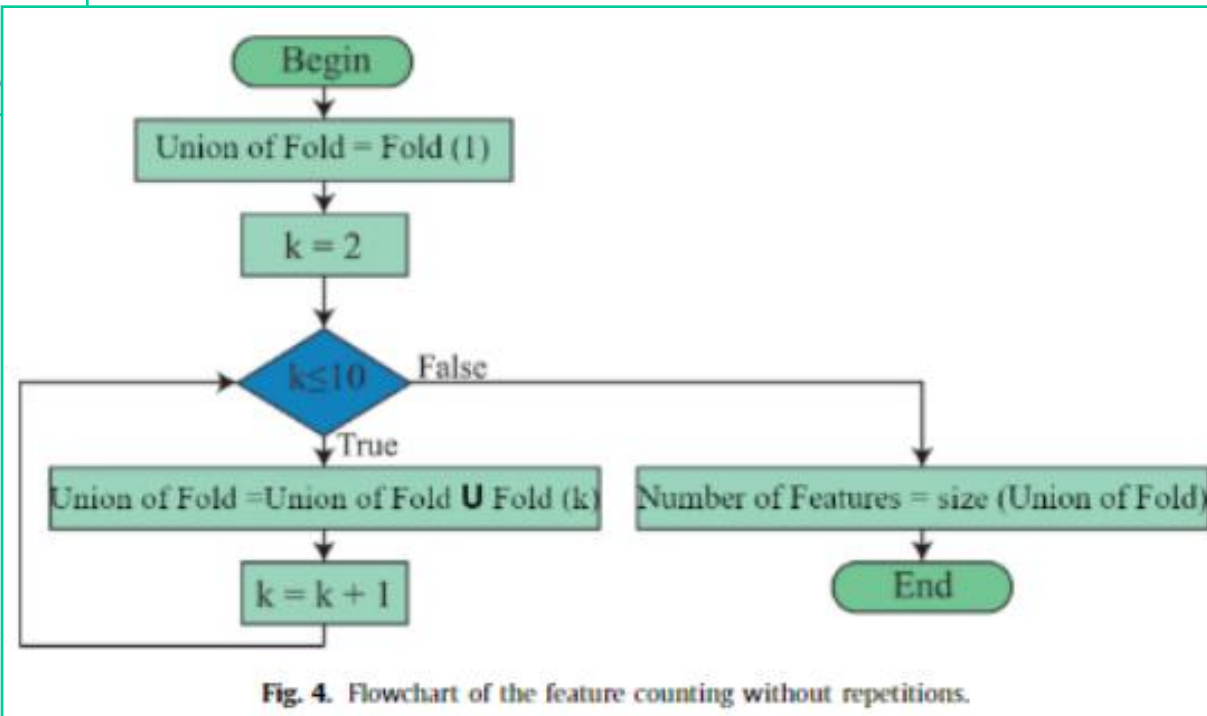
Matheus Gonçalves Ribeiro ^a, Leandro Alves Neves ^a, Marcelo Zanchetta do Nascimento ^b, Guilherme Freire Roberto ^b, Alessandro Santana Martins ^c, Thaina Aparecida Azevedo Tosta ^d

[Show more](#)

<https://doi.org/10.1016/j.eswa.2018.11.034>

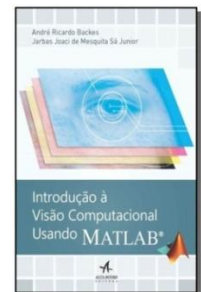
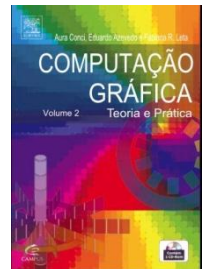
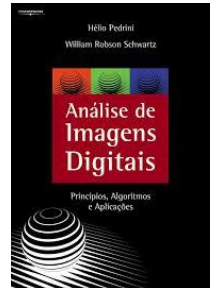
[Get rights and content](#)

Exemplo de Estratégia de Validação: Generalização



Referências

1. Pedrini, H., Schwartz, W. R. *Análise de Imagens Digitais: Princípios Algoritmos e Aplicações*. São Paulo: Thomson Learning, 2008.
2. Conci, A., Azevedo, E., Leta, F. R. *Computação Gráfica: Teoria e Prática*. Rio de Janeiro: Elsevier, vol. 2, 2008.
3. Backes, A. R., Sá Junior, J. J. De M. *Introdução à Visão Computacional Usando MatLab*. Rio de Janeiro: Alta Books, 2016.
4. Faceli, K., Lorena, A. C., Gama, J., de Carvalho, A. C. P. L. F. *Inteligência Artificial: Uma abordagem de Aprendizado de Máquina*. Rio de Janeiro: LTC, 2017.



Anexo: Exemplos Código Matlab

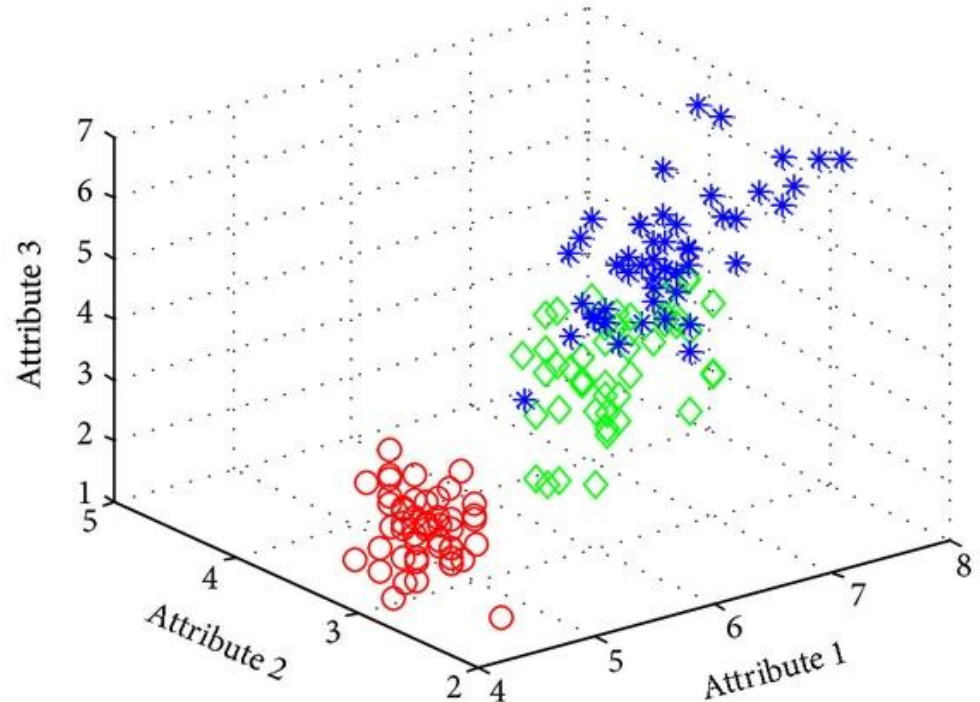
Exemplo de Estratégia de Validação: Generalização

Features (comp e larg. sépala, comp. e larg. pétala)

150x4 double

	1	2	3	4
1	5.1000	3.5000	1.4000	0.2000
2	4.9000	3	1.4000	0.2000
3	4.7000	3.2000	1.3000	0.2000
4	4.6000	3.1000	1.5000	0.2000
5	5	3.6000	1.4000	0.2000
6	5.4000	3.9000	1.7000	0.4000
7	4.6000	3.4000	1.4000	0.3000
8	5	3.4000	1.5000	0.2000
9	4.4000	2.9000	1.4000	0.2000
10	4.9000	3.1000	1.5000	0.1000
11	5.4000	3.7000	1.5000	0.2000
12	4.8000	3.4000	1.6000	0.2000
13	4.8000	3	1.4000	0.1000

fisheriris dataset (150 amostras)



Anexo: Códigos Matlab

K-folds

```
classify_kfold.m*  X  +  
1  function classify_kfold()  
2  - clear  
3  - load fisheriris;  
4  
5  %classes=[ones(1,50); 2*ones(1,50); 3*ones(1,50)]; %Três classes  
6  - classes=[ones(1,50); 2*ones(1,50)]; %Duas classes  
7  - k=5;  
8  - indices=crossvalind('Kfold', 100,k);  
9  - vet_acuracia=zeros(1,k);  
10  
11  - for i=1:k  
12  -     ind_train=find(indices~=i);  
13  -     ind_test=find(indices==i);  
14  -     base_train=meas(ind_train,:);  
15  -     base_test=meas(ind_test,:);  
16  -     classes_train=classes(ind_train);  
17  -     classes_test=classes(ind_test);  
18  -     class=classify(base_test, base_train, classes_train, 'linear');  
19  
20  -     vet_acuracia(i)=sum(class==classes_test)/length(class);  
21  
22  - end  
23  - fprintf('Acurácia média - %f\n',mean(vet_acuracia));
```


Anexo: Códigos Matlab

Holdout

```
classify_holdout.m  x  +
1  function classify_holdout()
2  -  clear
3  -  load fisheriris;
4  -  [train,test]=crossvalind('HoldOut', 150,0.25);
5  -  classes=[ones(1,50); 2*ones(1,50); 3*ones(1,50)];
6  -  ind_train=find(train==1);
7  -  ind_test=find(test==1);
8  -  base_train=meas(ind_train,:);
9  -  base_test=meas(ind_test,:);
10 -  classes_train=classes(ind_train);
11 -  classes_test=classes(ind_test);
12 -  class=classify(base_test, base_train, classes_train,'linear');
13 -  acuracia=sum(class==classes_test)/length(class);
14 -  fprintf('Acurácia - %f\n',acuracia);
15 -  end
```


Anexo: Códigos Matlab

Knn

```
knn_exemplo.m  X  +
1  function knn_exemplo()
2  -  clear
3  %amostras para teste
4  -  Teste=[2.2 4.4
5  -         -4.2 2.3];
6  %amostras para treinamento
7  -  Treino=[ 1.0  2.0;  1.1  2.4;  1.3  2.1  %classe 1
8  -           -4.0  5.0; -4.1  4.6; -4.2  4.2  %classe 2
9  -           -3.0 -4.1; -3.1 -4.4; -3.3  -4.2]; %classe 3
10
11 -  hold on;
12  %plot das amostras de teste
13
14 -  plot(Teste(:,1) ,Teste(:,2), 'k^');
15  %plot das amostras de treinamento
16 -  plot(Treino(1:3,1) , Treino(1:3,2), 'go');%classe 1
17 -  plot(Treino(1:3,1) , Treino(1:3,2), 'rx');%classe 2
18 -  plot(Treino(4:6,1) , Treino(7:9,2), 'bd');%classe 2
19 -  axis([-6 5 -6 6]);
20
21  %grupos de cada uma das amostras de treinamento
22 -  Grupo=[1; 1; 1; 2; 2; 2; 3; 3; 3];
23 -  K=2;%número de vizinhos mais próximos
24  %clusterização
25 -  Classes=knnclassify(Teste,Treino,Grupo,K);
26 -  disp(Classes);
```