CEFET/RJ

# Pré-processamento de Dados: DAL Toolbox

**Eduardo Ogasawara**

eduardo.ogasawara@cefet-rj.br

https://eic.cefet-rj.br/~eogasawara

# Estrutura

## Limpeza de Dados

- Tratamento de valores ausentes
- Detecção de outliers

## Codificação Categórica

- Mapeamento Categórico
-

## Normalização de Dados

- Min-Max
- Z-Score

## Discretização e Suavização

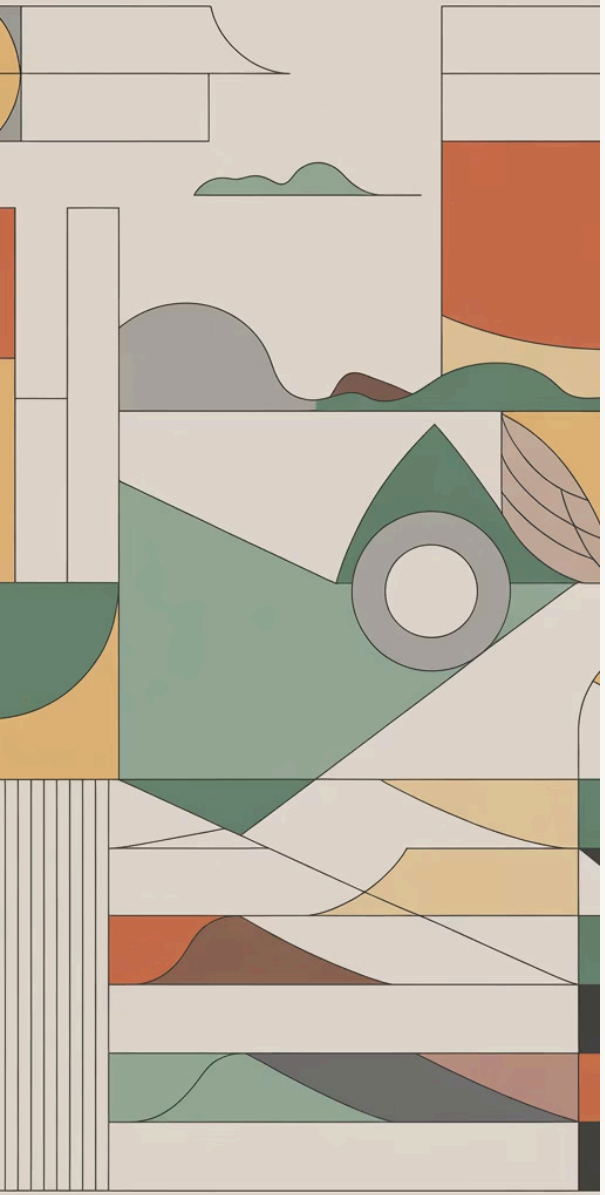
- Métodos baseados em intervalos
- Métodos baseados em frequência
- Agrupamento (Clustering)

## Redução de Dimensionalidade

- PCA
- Análise de curvatura

## Amostragem de Dados

- Amostragem aleatória
- Amostragem estratificada



# Limpeza de Dados: Valores Ausentes e Outliers

A qualidade dos dados é crucial para a robustez de qualquer modelo de Machine Learning. A limpeza de dados foca em resolver problemas comuns que podem comprometer a análise.

## Valores Ausentes (Missing Values)

- **O que são:** Dados não registrados ou indisponíveis em uma observação específica.
- **Por que ocorrem:** Erros de entrada, falhas em sensores, recusa de resposta ou dados não aplicáveis.
- **Importância do tratamento:** Ignorá-los pode levar a modelos enviesados, análises imprecisas e perda de informações valiosas.

## Outliers (Valores Atípicos)

- **O que são:** Pontos de dados que se desviam significativamente do padrão geral do conjunto de dados.
- **Por que são problemáticos:** Podem distorcer estatísticas, influenciar negativamente o treinamento de modelos e levar a conclusões errôneas.
- **Importância da detecção:** Identificá-los e tratá-los adequadamente é essencial para garantir a validade e a generalização dos modelos.

# Valores Ausentes: Causas e Riscos

## Causas Comuns

Erros na coleta, pesquisas incompletas, falhas de sensores ou dados indisponíveis.

## Riscos de Viés

Ausência não aleatória distorce análises e leva a conclusões incorretas.

## Falhas de Algoritmos

Algoritmos de ML podem falhar ou gerar resultados incorretos com valores NA. Pré-processamento é crucial.

Lidar com dados ausentes envolve duas estratégias: **remoção** (excluir NAs) e **imputação** (preencher valores). Cada uma tem prós e contras, dependendo do caso de uso.

# Prática: Remoção de Valores Ausentes

## Exemplo de Implementação

Demonstração prática em R com o dataset `iris`. Introduzimos um valor ausente e usamos `na.omit()` para remover linhas com NAs.

**Início Rápido:** O código completo e exemplos adicionais estão disponíveis no repositório GitHub do DAL Toolbox em examples/transf/na_removal.md

```
# Load the iris dataset
iris <- datasets::iris

# Introduce a missing value
iris$Sepal.Length[2] <- NA

# Check original size
nrow(iris)  # 150

# Remove rows with NAs
iris.clean <- na.omit(iris)

# Verify cleaned size
nrow(iris.clean)  # 149
```

**Importante:** a primitiva dedicada ao tratamento de valores ausentes no DAL Toolbox ainda não está disponível.

# Detecção de Outliers: Regra do Boxplot (Tukey)

A Regra do Boxplot, desenvolvida por John Tukey, é um método estatístico amplamente utilizado para detecção de outliers.

- Define outliers com base no **Intervalo Interquartil (IQR = Q3 - Q1)**.
- Um ponto é classificado como outlier se estiver:
  - Abaixo de **Q1 - 1.5×IQR**
  - Acima de **Q3 + 1.5×IQR**
- **Vantagens:**
  - Robusto para distribuições assimétricas, utilizando quartis em vez de média.
  - Altamente interpretável, devido à natureza visual dos box plots.

**Referências:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools e Techniques (4th Ed.)

# Prática: Boxplot de Outliers (DAL Toolbox)

A DAL Toolbox simplifica a implementação da Regra do Boxplot em um processo intuitivo de três etapas, seguindo o paradigma padrão `fit-transform`:

- Crie um objeto de detecção de outliers com `outliers_boxplot()`.
- Ajuste (`fit`) o objeto ao seu conjunto de dados para calcular Q1, Q3 e IQR.
- Aplique a transformação (`transform`) para remover ou sinalizar outliers.

Aqui está um exemplo prático usando o conjunto de dados Iris:

```
out <- outliers_boxplot()
out <- fit(out, datasets::iris)
iris.clean <- transform(out, datasets::iris)
attr(iris.clean, "idx")
```

Os índices dos outliers removidos são armazenados como um atributo, permitindo fácil inspeção.

**Código completo disponível em:** https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/outliers_boxplot.md

# Detecção de Outliers: Regra 3-Sigma Gaussiana

A Regra 3-Sigma Gaussiana, ou regra 68-95-99.7, é um método estatístico para detecção de outliers, baseado na distribuição normal.

## Pontos Chave

- Outliers definidos como pontos fora de **média ± 3×desvio padrão**.
- Baseia-se na propriedade de que ~99.7% dos dados normais caem dentro de 3 desvios padrão.
- Mais conservadora que a Regra do Boxplot (detecta menos outliers).
- Ideal para dados com distribuição normal ou para identificar apenas os outliers mais extremos.
- Simples de calcular e amplamente compreendida.

## Limitações

- Dependência da média e desvio padrão a torna sensível aos próprios outliers.
- Requer que os dados sigam uma distribuição normal para ser precisa.
- Pode ignorar outliers em distribuições não normais.

**References:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Prática: Outliers Gaussianos (DAL Toolbox)

A Regra Gaussiana 3-Sigma no DAL Toolbox segue o padrão "fit-transform", consistente com o método Boxplot.

A implementação calcula média e desvio padrão para cada característica numérica, aplicando o limiar de 3-sigma para identificar outliers:

```
out <- outliers_gaussian()
out <- fit(out, datasets::iris)
iris.clean <- transform(out, datasets::iris)
attr(iris.clean, "idx")
```

- `fit()`: Calcula as estatísticas necessárias.
- `transform()`: Aplica a regra para detectar outliers.
- Os índices dos outliers são acessíveis para inspeção.

A API permite alternar e comparar facilmente diferentes métodos de detecção de outliers, escolhendo o mais adequado para seus dados.

**Código completo disponível em:** https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/outliers_gaussian.md

# Comparação de Ambas as Abordagens

### Regra do Boxplot

Detecta mais outliers

Robusto à assimetria

Sem suposição de normalidade

### Gaussiana 3-Sigma

Detecta menos outliers

Assume distribuição normal

Abordagem mais conservadora

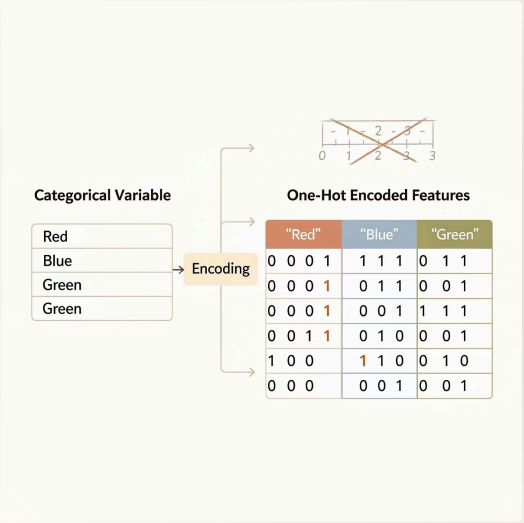
A escolha do método depende das características dos seus dados e dos seus objetivos.

- **Boxplot:** Ideal para análise exploratória e dados assimétricos, sem suposições de normalidade.
- **Gaussiana 3-Sigma:** Melhor para dados normais e outliers extremos. É mais conservadora, com menos falsos positivos.

Recomenda-se aplicar ambos os métodos e comparar os resultados. Isso oferece insights sobre a sensibilidade da detecção e valida os outliers.

**Referências:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Variáveis Categóricas: Por Que a Codificação Importa




## A Exigência Numérica

- Algoritmos de ML só processam números.
- Categorias (ex: "vermelho") devem ser convertidas.

## O Problema da Ordem Falsa

- Codificar como inteiros (vermelho=1, azul=2) cria ordem artificial.
- Algoritmos podem assumir relações matemáticas inexistentes.

### Categorias Brutas

Rótulos de texto (ex: "gato")

### One-Hot Encoding

Colunas binárias independentes (ex: gato=1/0)

### Pronto para Algoritmo

Sem ordenação falsa, entradas válidas

One-hot encoding resolve isso com colunas binárias para cada categoria. É essencial para modelos lineares, SVMs e Deep Learning, evitando ordens falsas.

# Prática: Mapeamento Categórico

## Usando a DAL Toolbox para Transformação

```
# Create categorical mapping object
cm <- categ_mapping("Species")

# Transform the iris dataset
iris_cm <- transform(cm, datasets::iris)

# View the results
head(iris_cm)
```

**Saiba Mais:** Documentação completa e exemplos avançados disponíveis em github.com/cefet-rj-dal/daltoolbox/examples/transf/categorical_mapping.md

A função `categ_mapping()` da DAL Toolbox simplifica a codificação categórica. Ela garante consistência entre dados de treinamento e teste, crucial para a robustez do modelo.

01

### Criar Mapeamento

Definir qual coluna codificar

02

### Aplicar Transformação

Converter categorias para representação numérica

03

### Verificar Resultados

Inspecionar colunas codificadas para correção

# Normalização de Dados em Machine Learning

## O Desafio

Dados possuem escalas diversas (ex: idade vs. renda).

Algoritmos sensíveis à escala podem dar peso indevido a valores maiores.

Isso prejudica o desempenho do modelo, impactando algoritmos de distância e gradiente.

## A Solução

Normalização padroniza escalas para contribuição justa e equilibrada.

Métodos principais:

- **Min-Max:** Reescalona para um intervalo fixo.
- **Z-Score:** Centra na média, com variância unitária.

A DAL Toolbox oferece ambos com sintaxe simples.

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Normalização Min-Max

## Conceito Central

Redimensiona valores para o intervalo [0,1], mantendo a forma da distribuição original.

## A Fórmula

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Ajusta cada valor em relação ao mínimo e máximo.

## Benefícios Chave

- Lida com diferentes unidades
- Ideal para redes neurais
- Intervalo de saída limitado
- Preserva valores zero

**Importante:** Min-Max é sensível a outliers, que podem comprimir os dados.

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Prática: Min-Max no DAL Toolbox

## Passos de Implementação

A normalização Min-Max no DAL Toolbox segue 3 etapas:

Este padrão segue o familiar paradigma fit-transform, intuitivo para quem usa scikit-learn.

**Exemplo completo:**

github.com/cefet-rj-dal/daltoolbox

```
# Step 1: Create normalizer object
norm <- minmax()

# Step 2: Fit to your data
norm <- fit(norm, datasets::iris)

# Step 3: Transform the dataset
idata <- transform(norm, datasets::iris)

# Verify results
summary(idata)
```

# Padronização Z-Score

## 1 Fundamentação Estatística

Transforma dados para terem média 0 e desvio padrão 1.

$$z = \frac{x - \mu}{\sigma}$$

Onde μ é a média e σ é o desvio padrão da feature.

## 2 Vantagens sobre Min-Max

- Menos sensível a outliers
- Sem intervalo delimitado
- Preserva informações de outliers
- Lida com unidades diferentes

## 3 Aplicações Ideais

- Essencial para PCA e modelos que assumem distribuição normal.
- Útil para features com escalas variadas.
- Mantém informações de outliers.

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Conceitos e Técnicas (3ª Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Ferramentas e Técnicas Práticas de Machine Learning (4ª Ed.)

# Z-Score na Prática & Escalonamento Personalizado

1

## Z-Score Padrão

```
# Create standardizer
norm <- zscore()

# Fit to data
norm <- fit(norm, datasets::iris)

# Transform
zdata <- transform(norm, datasets::iris)

summary(zdata)
```

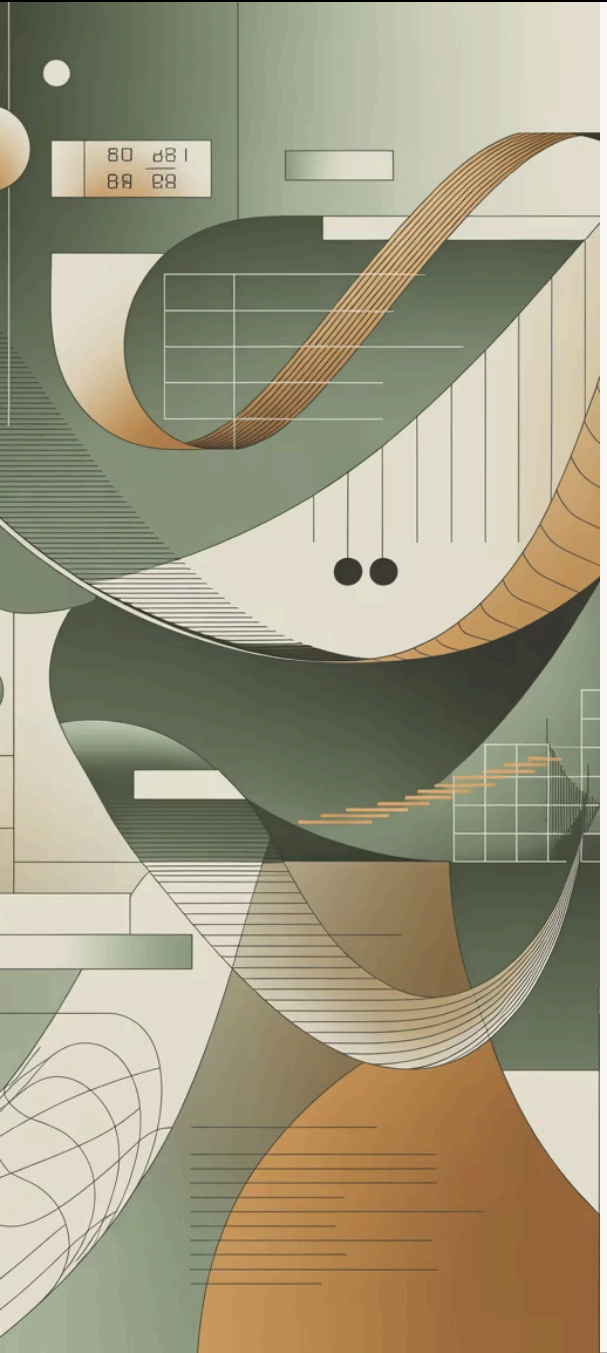
2

## Parâmetros Personalizados

- DAL Toolbox permite média e desvio padrão personalizados.
- Flexibilidade para requisitos específicos de domínio.
- Mantém a reversibilidade para mapeamento a qualquer distribuição alvo.

**Acesse exemplos completos:** Documentação de normalização Z-Score

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools e Techniques (4th Ed.)



# Discretização e Suavização de Dados

Converter variáveis contínuas em intervalos discretos para melhor análise.

# Discretização Baseada em Intervalos

Divide o intervalo de dados em compartimentos de largura igual. Abordagem simples, interpretável e fácil de implementar.

## Características

- Simples e intuitiva
- Cria compartimentos de largura igual
- Pode ignorar a densidade dos dados
- Sensível a valores discrepantes

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Implementação Baseada em Intervalos

01

### Inicializar Suavizador

Criar objeto e definir número de bins.

02

### Ajustar aos Dados

Treinar discretizador na variável.

03

### Transformar Valores

Aplicar intervalos para discretizar dados.

```
obj <- smoothing_inter(n=2)
obj <- fit(obj, datasets::iris$Sepal.Length)
bins <- transform(obj, datasets::iris$Sepal.Length)
```

Código completo em: github.com/cefet-rj-dal/daltoolbox

# Discretização Baseada em Frequência

### Frequências Iguais

Número de observações equilibrado por compartimento.

### Adaptação Flexível

Adapta-se à distribuição dos dados.

### Evita Escassez

Previne compartimentos vazios.

### Lida com Assimetria

Eficaz para distribuições assimétricas.

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Implementação Baseada em Frequência

## Fluxo de Trabalho do DAL Toolbox

- A abordagem de frequência usa o padrão fit-transform.
- Cria compartimentos com contagens de observações iguais.
- Ideal para dados do mundo real e distribuições assimétricas.

```
obj <- smoothing_freq(n=2)
obj <- fit(obj, datasets::iris$Sepal.Length)
bins <- transform(obj, datasets::iris$Sepal.Length)
```

Código completo disponível em: **github.com/cefet-rj-dal/daltoolbox**

# Discretização Baseada em Agrupamento

## Estrutura Natural

Identifica agrupamentos naturais nos dados.

## Altamente Adaptável

Ajusta limites dos bins à distribuição dos dados.

## Complexidade Adicional

Exige mais recursos computacionais e ajuste de parâmetros.

## Sensível à Inicialização

Resultados podem variar com centros iniciais.

**Referência:** Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools and Techniques (4th Ed.)

# Implementação Baseada em Clusterização

## Discretização Adaptativa na Prática

Descobre agrupamentos naturais nos dados contínuos, criando "bins" que refletem a estrutura subjacente.

```
obj <- smoothing_cluster(n=2)
obj <- fit(obj, datasets::iris$Sepal.Length)
bins <- transform(obj, datasets::iris$Sepal.Length)
```

Código completo disponível em: github.com/cefet-rj-dal/daltoolbox

# Fundamentos da PCA: Teoria e Conceitos

## Como a PCA Funciona

- Projeta dados de alta dimensão em componentes principais ortogonais.
- Componentes ordenados pela variância capturada (do maior para o menor).
- Remove redundância, eliminando correlações entre características.
- Cria uma representação compacta com as informações essenciais.

### Projeção Ortogonal

Dados em eixos não correlacionados

### Variância Máxima

Componentes por informação capturada

### Remoção de Redundância

Consolida características correlacionadas

### Ganhos de Eficiência

Modelos mais rápidos e leves

**Referências:** Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)

# Implementando PCA com a DAL Toolbox

Implemente PCA facilmente com a DAL Toolbox em três passos: criar, ajustar e transformar. Veja o exemplo com o conjunto de dados Iris:

```
# Create PCA transformation object
mypca <- dt_pca("Species")

# Fit PCA model to training data
mypca <- fit(mypca, datasets::iris)

# Transform dataset to principal components
iris_pca <- transform(mypca, datasets::iris)

# View transformed data
head(iris_pca)
```

A sintaxe cuida da padronização, decomposição e transformação, resultando em um conjunto de dados de dimensão reduzida pronto para análise.

01

### Inicializar

Cria objeto PCA

02

### Ajustar

Ajusta o modelo aos dados

03

### Transformar

Transforma dados

**Exemplo funcional completo:** https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/dal_pca.md

# O Desafio da Seleção de Componentes

Quantos componentes deve-se manter?

### Componentes Demais

Mantém complexidade e ruído desnecessários.

### Componentes de Menos

Perde informações críticas e padrões.

### A Zona de Equilíbrio

Análise de curvatura identifica o equilíbrio ideal.

Regras fixas (como "95% da variância") ignoram a estrutura dos dados. Métodos de curvatura se adaptam, encontrando o ponto ideal onde componentes adicionais agregam valor mínimo.

**Referências:** Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)

# Análise de Curvatura: Encontrando o Balanço

## Abordagem Matemática para Seleção de Parâmetros

A análise de curvatura identifica o balanço, onde a taxa de mudança deixa de ser significativa. Isso oferece um critério objetivo e adaptável para a seleção de parâmetros.

**Dois métodos de curvatura:**

- **Mínimo de curvatura:** Para curvas crescentes (ex: variância acumulada), encontra onde os ganhos se estabilizam.
- **Máximo de curvatura:** Para curvas decrescentes (ex: erro do modelo), localiza onde as melhorias diminuem.

Automatiza a seleção de parâmetros, eliminando subjetividade.

### Detecção Objetiva

Remove julgamento subjetivo na seleção de parâmetros.

### Método Adaptativo

Se ajusta a dados variados sem limiares predefinidos.

### Resultados Reprodutíveis

Os mesmos dados geram recomendações idênticas.

**Referências:** Han, J., Kamber, M., & Pei, J. – *Data Mining: Concepts and Techniques* (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – *Data Mining: Practical Machine Learning Tools and Techniques* (4th Ed.)

# Prática: Mínimo de Curvatura

O mínimo de curvatura identifica o "cotovelo" em curvas *crescentes*. Em PCA, é aplicado à variância explicada cumulativa para encontrar o ponto de retornos decrescentes dos componentes. Abaixo, um exemplo com o conjunto de dados Iris:

```
# Perform PCA on Iris features
pca <- prcomp(datasets::iris[,1:4], center=TRUE, scale.=TRUE)

# Calculate cumulative proportion of variance
y <- cumsum(pca$sdev^2/sum(pca$sdev^2))

# Fit curvature minimum detector
km <- fit_curvature_min()

# Find optimal number of components
res <- transform(km, y)
res$x  # Returns recommended component count
```

O algoritmo analisa a segunda derivada da curva de variância para identificar onde o ganho de variância diminui. Ele equilibra a retenção de informações com a redução de dimensionalidade.

**Explore a implementação completa: https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/curvature_minimum.md**

# Prática: Curvatura Máxima

A curvatura máxima detecta pontos de inflexão em curvas decrescentes. É ideal para cenários como redução de erro de modelo ou funções de perda, indicando onde as melhorias começam a se estabilizar.

```
# Create example decreasing curve
x <- seq(1, 10, by=.5)
v <- -log(x)

# Fit curvature maximum detector
km <- fit_curvature_max()

# Identify elbow point
res <- transform(km, v)
res$x  # Returns x-value at maximum curvature
```

## Uso Comum

- Curvas de erro de treinamento
- Perda por validação cruzada
- Decaimento de informações
- Análise de convergência

**Código completo:** **https://github.com/cefet-rj-dal/daltoolbox/blob/main/examples/transf/curvature_maximum.md**

# Estratégias de Amostragem de Dados para Machine Learning

A avaliação eficaz do modelo depende da divisão adequada dos dados.

- Amostragem aleatória e estratificada são as abordagens principais.
- Cada método oferece vantagens para diferentes cenários.

Esta apresentação, usando o DAL Toolbox em R, irá:

- Explorar a teoria por trás de cada abordagem.
- Demonstrar a implementação com exemplos de código.
- Orientar na escolha da estratégia ideal para seus projetos.

# Amostragem Aleatória: Teoria e Princípios

## Como Funciona

- Divide dados arbitrariamente, sem considerar características ou classes.
- Cada ponto de dado tem igual probabilidade de ser alocado para treinamento ou teste.
- Método rápido e com mínima sobrecarga computacional.

## Desvantagens

- Embora preserve proporções em média, divisões individuais podem ser desequilibradas.
- Risco de viés na avaliação do modelo se uma classe for sub-representada em um conjunto.

### Velocidade

Execução rápida

### Simplicidade

Fácil de implementar

### Risco

Pode gerar desequilíbrio

# Prática: Implementando Amostragem Aleatória

## Amostragem Aleatória com DAL Toolbox

- Use `sample_random()` e `train_test()` do DAL Toolbox.
- Automaticamente divide o conjunto de dados (ex: Iris) em treinamento e teste.

```
tt <- train_test(sample_random(), datasets::iris)
table(tt$train$Species)
table(tt$test$Species)
```

- A função `table()` mostra a distribuição das classes pós-divisão.
- Pode haver variação nas contagens, ilustrando o desequilíbrio potencial da amostragem aleatória.
- Execuções repetidas geram distribuições diferentes, destacando a variabilidade.

- `train_test()` retorna dois data frames: `$train` e `$test`.
- Divisão padrão de 80-20, mas personalizável.

**Ver Código Completo**

# Amostragem Estratificada: Mantendo o Equilíbrio

### Preservação Proporcional

Mantém as proporções das classes do conjunto de dados original.

### Essencial para Classificação

Crítico para conjuntos de dados desequilibrados, assegurando representação de classes minoritárias.

### Avaliação Confiável

Produz resultados consistentes e reproduzíveis do desempenho do modelo.

A amostragem estratificada supera as limitações da amostragem aleatória, mantendo as proporções das classes nos conjuntos de treinamento e teste. É essencial para conjuntos de dados desequilibrados, garantindo a representação de classes minoritárias.

Este método divide o conjunto de dados pela variável alvo e amostra proporcionalmente de cada grupo (estrato). Isso assegura representação justa de todas as classes, resultando em avaliações de modelo mais confiáveis.

Referência: Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (3rd Ed.); Witten, I. H., Frank, E., Hall, M. A., & Pal, C. – Data Mining: Practical Machine Learning Tools e Techniques (4th Ed.)

# Prática: Implementando Amostragem Estratificada

## DAL Toolbox: Amostragem Estratificada

Para implementar a amostragem estratificada, basta especificar a variável de estratificação. Use a função `sample_stratified()` com o nome da coluna alvo, como "Species" neste exemplo.

```
tt <- train_test(
  sample_stratified("Species"),
  datasets::iris
)
table(tt$train$Species)
table(tt$test$Species)
```

As tabelas de saída demonstram proporções idênticas entre os conjuntos de treinamento e teste, garantindo precisão.

Ao contrário da amostragem aleatória, este método produz distribuições de classe consistentes, mesmo com múltiplas execuções. Isso fornece bases mais confiáveis para a avaliação do modelo.

**Documentação Completa**

# Avançado: Validação Cruzada K-Fold Estratificada

A validação cruzada K-fold estratificada estende a amostragem estratificada para múltiplas divisões, fornecendo estimativas de desempenho mais robustas.

- Divide os dados em 'k' subconjuntos (folds).
- Cada fold é usado como teste, enquanto os restantes são para treinamento.
- Garante que cada fold mantenha as proporções de classe originais.

## Implementação no DAL Toolbox

```
folds <- k_fold(
  sample_stratified("Species"),
  datasets::iris,
  4
)
do.call(rbind, lapply(folds, function(f) table(f$Species)))
```

### Entendendo a Saída

- `k_fold()` retorna uma lista de 'k' data frames (folds).
- A saída confirma a estratificação correta, mostrando a distribuição de classes em cada fold.

### Por Que Usar K-Fold?

- Reduz a variância da avaliação.
- Oferece estimativas de desempenho mais confiáveis.
- Ideal para conjuntos de dados menores.

**Mais Exemplos**


CEFET/RJ

# Agradecimentos

FAPERJ

CAPES

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

**Eduardo Ogasawara**

eduardo.ogasawara@cefet-rj.br

https://eic.cefet-rj.br/~eogasawara