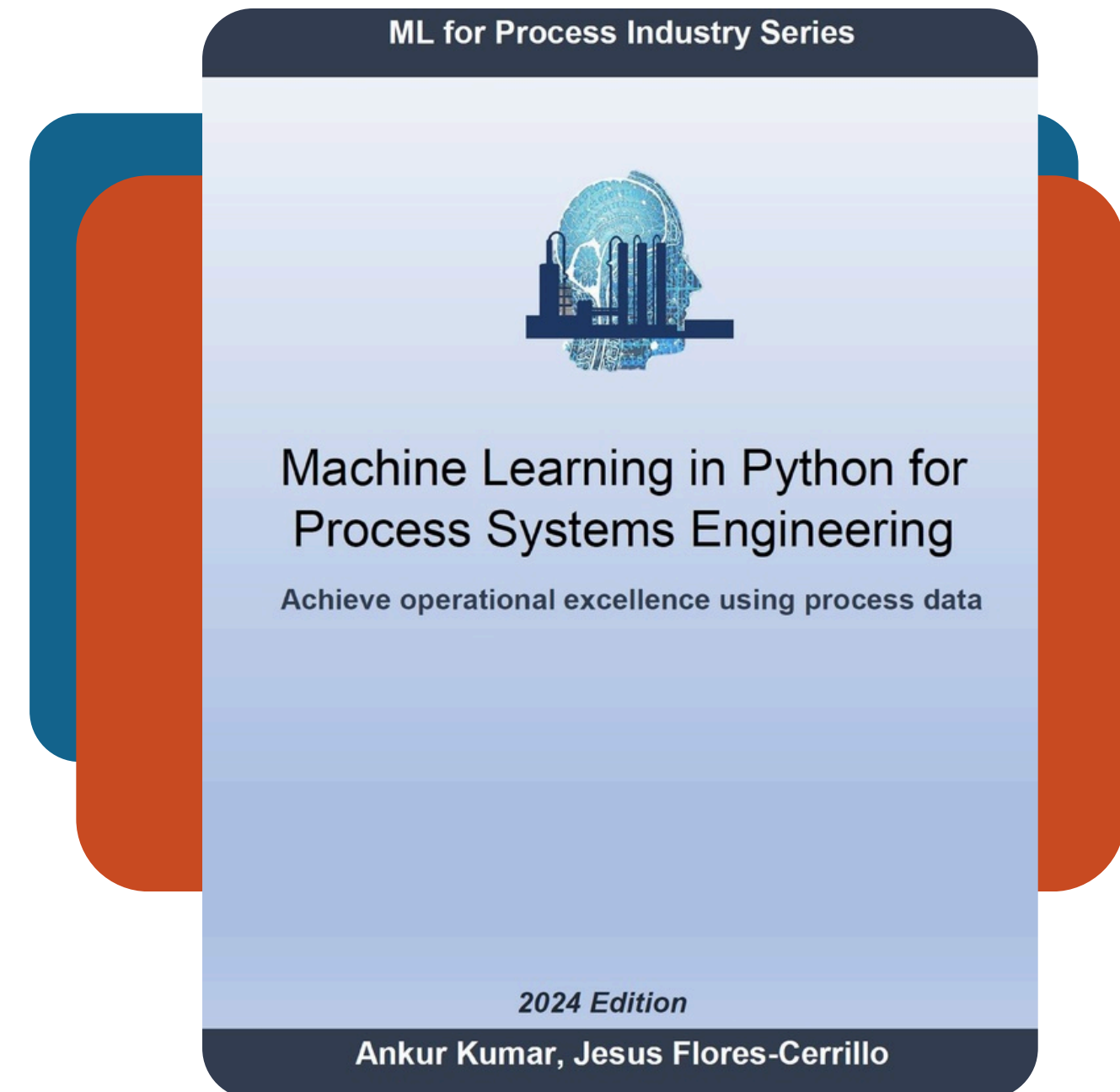
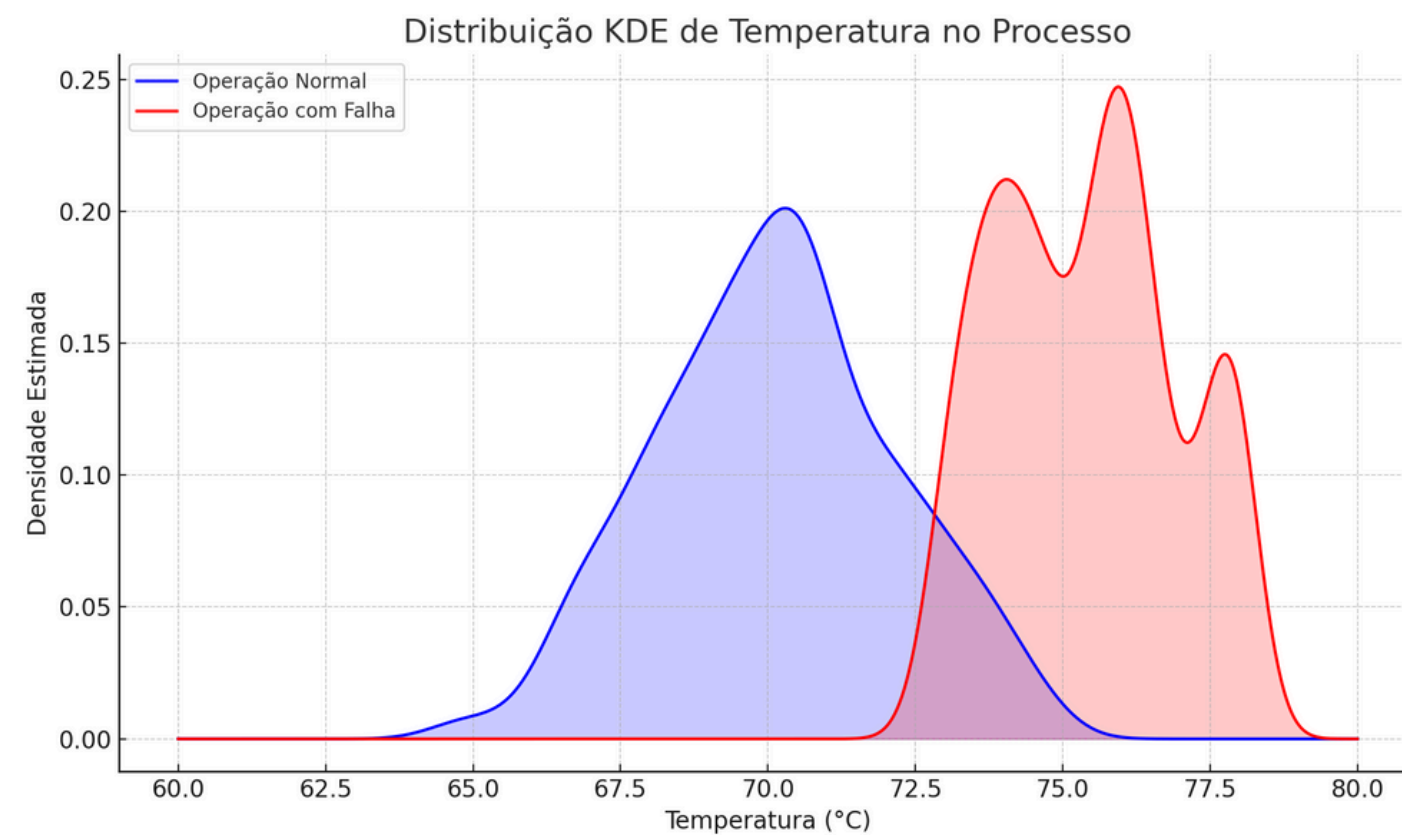


Redes Neurais

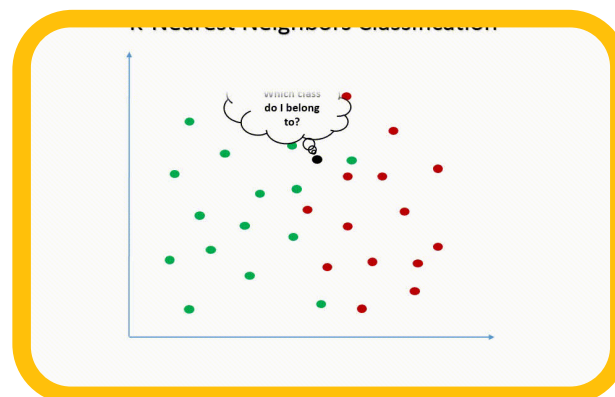
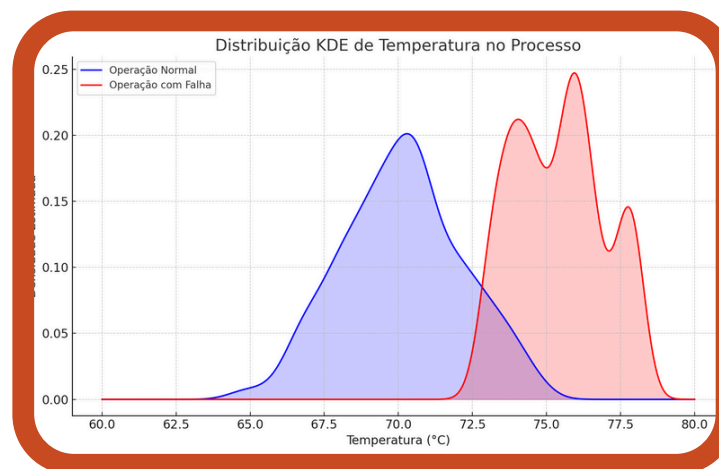
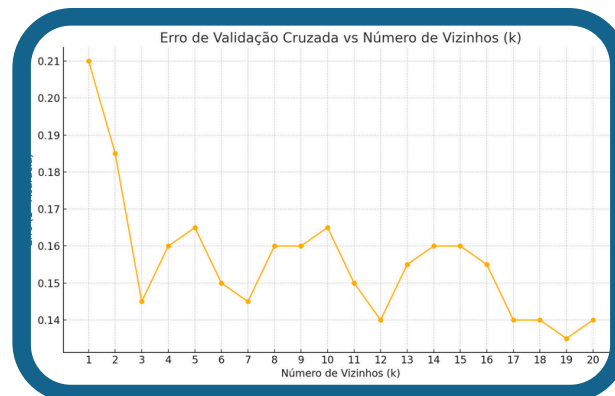
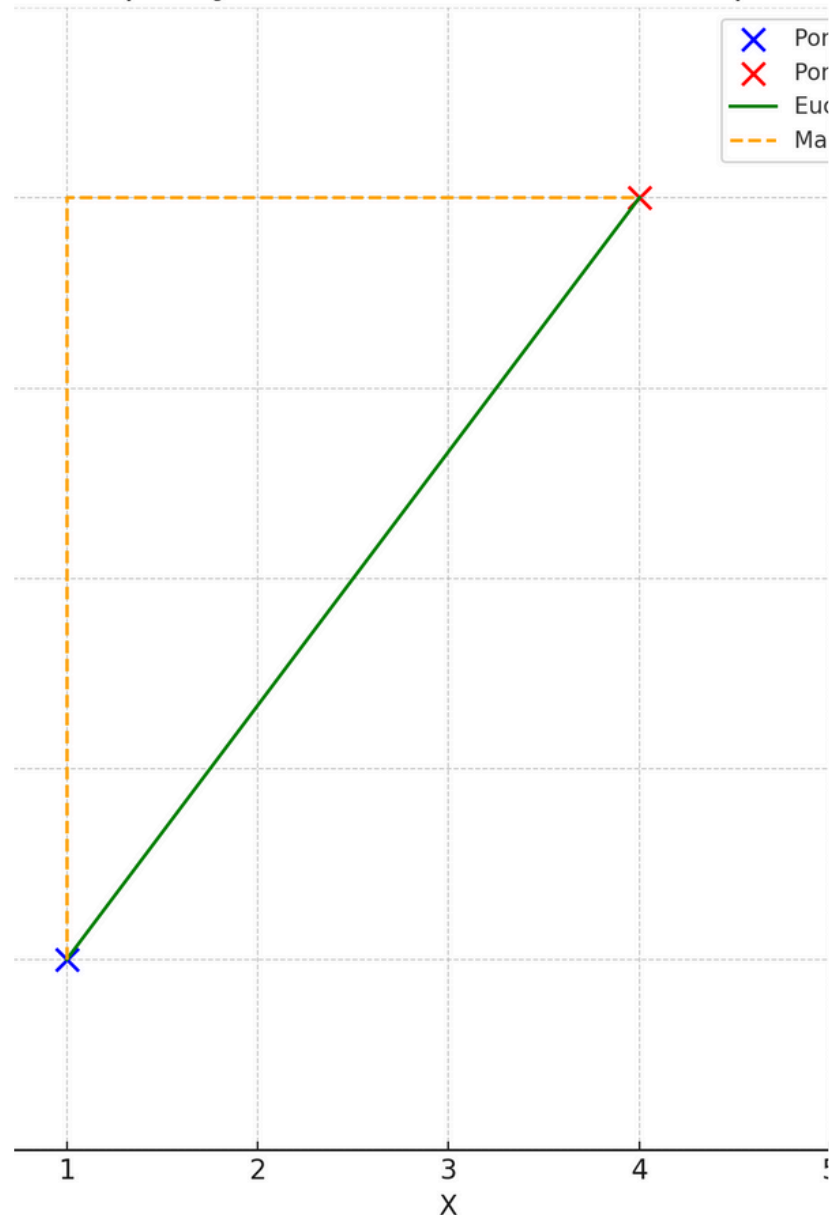
Capítulo 10 – Machine Learning in Python for Process Systems Engineering

[Start Presentation](#)

Kernel Density Estimation (KDE) e k-Nearest Neighbors (kNN).



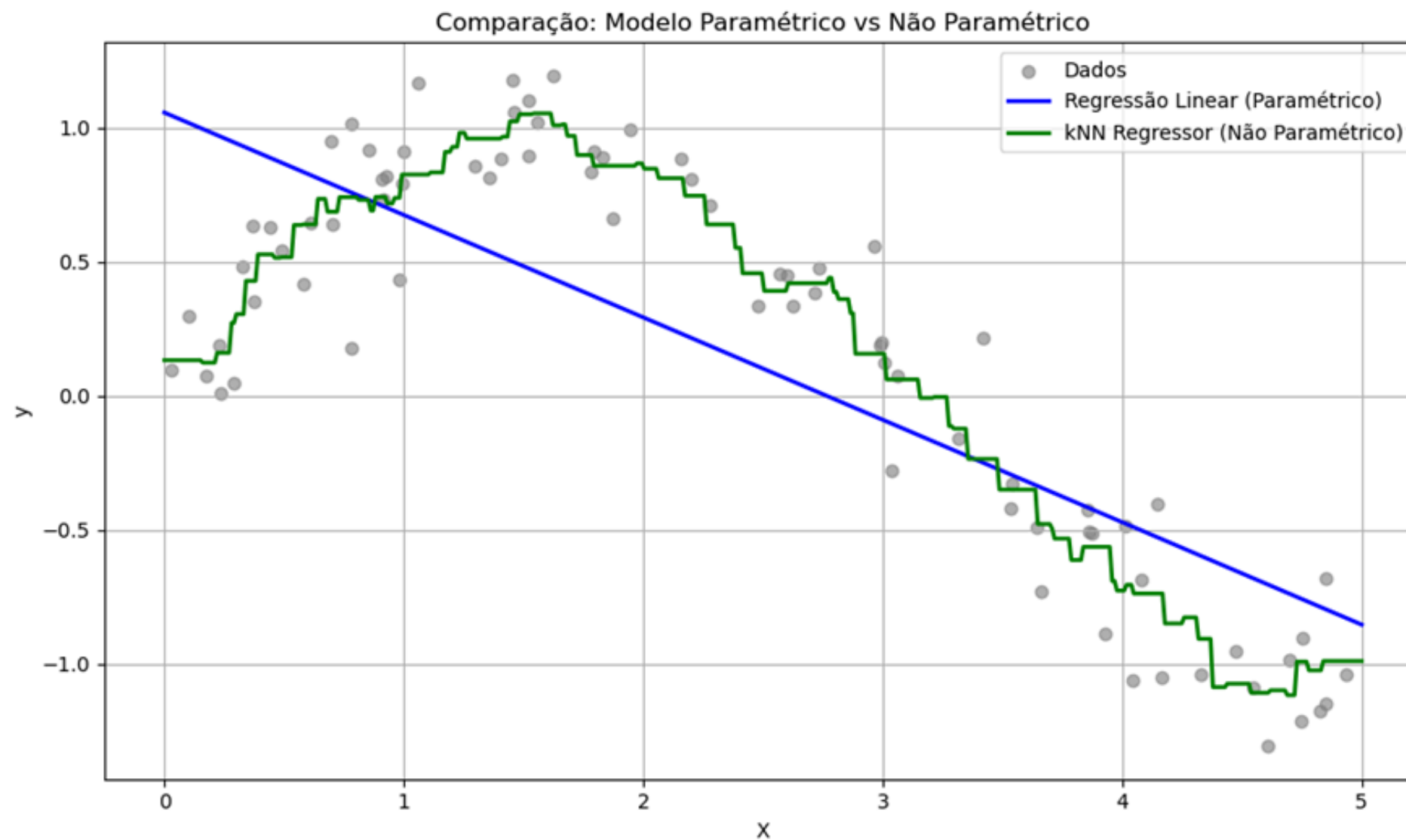
Comparação de Distâncias entre dois pontos



Objetivos

01. Compreender o que são métodos não paramétricos em machine learning.
02. Explorar duas técnicas
03. Visualizar o impacto da escolha de parâmetros (bandwidth em KDE, k em kNN)
04. Aprender a implementar e aplicar essas técnicas com dados reais em Python.

"Este capítulo mostra como essas técnicas simples e flexíveis podem ser aplicadas na engenharia de processos sem exigir modelos paramétricos complexos."



Não fazem suposições sobre a distribuição dos dados.

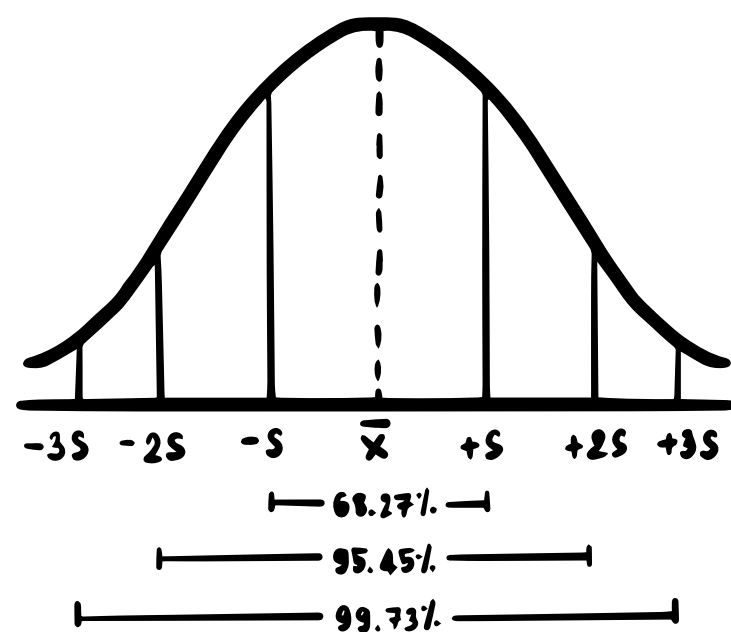


Se adaptam melhor a dados reais e complexos.

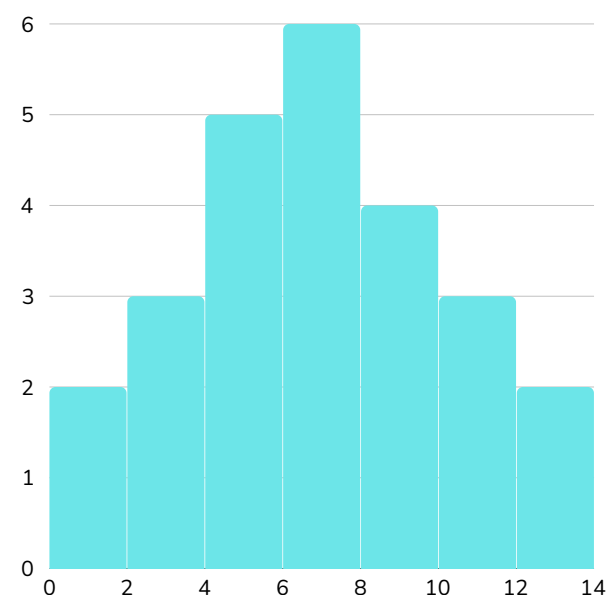


Usam diretamente os dados de treino para fazer previsões.

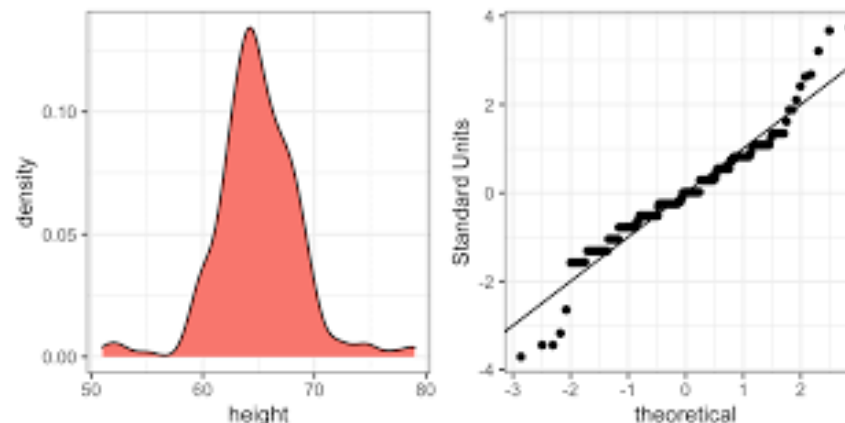
Kernel Density Estimation (KDE) é um método para estimar distribuições de probabilidade sem pressupor forma funcional (como normal, exponencial, etc.).



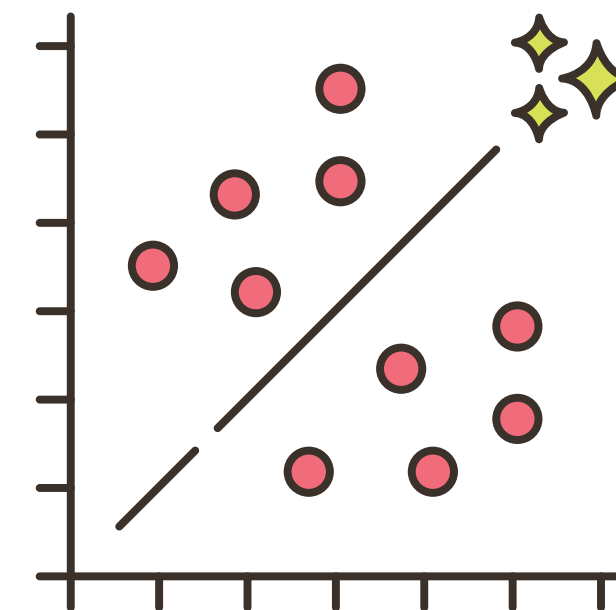
**Estima a função
densidade de
probabilidade**



**Alternativa à
histogramas**



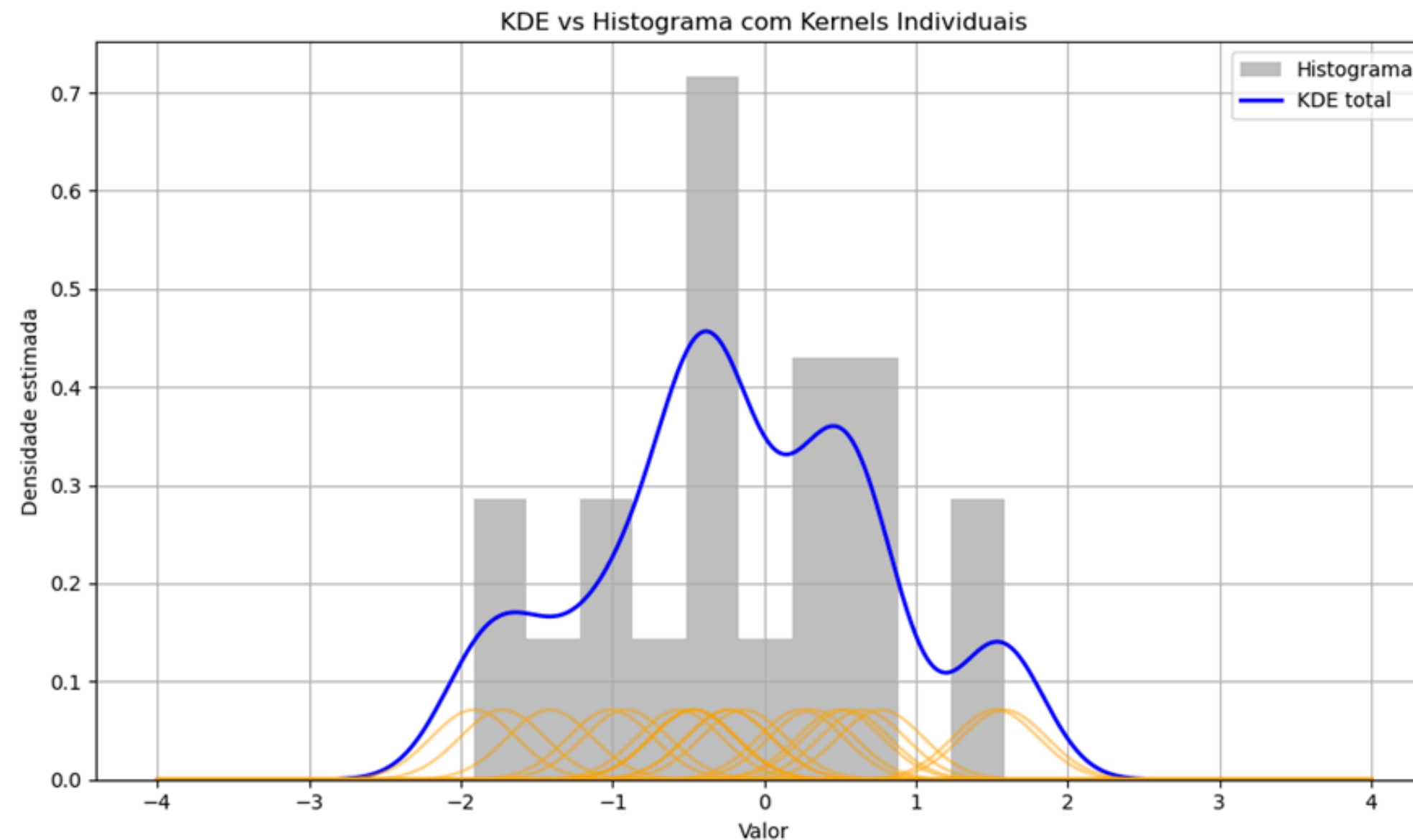
**Visualização de
distribuições reais**



**Deteção de
outliers**

“Imagine que cada ponto de dado é uma vela que emite luz em forma de curva suave. KDE soma todas essas luzes para revelar a forma da paisagem dos dados.”

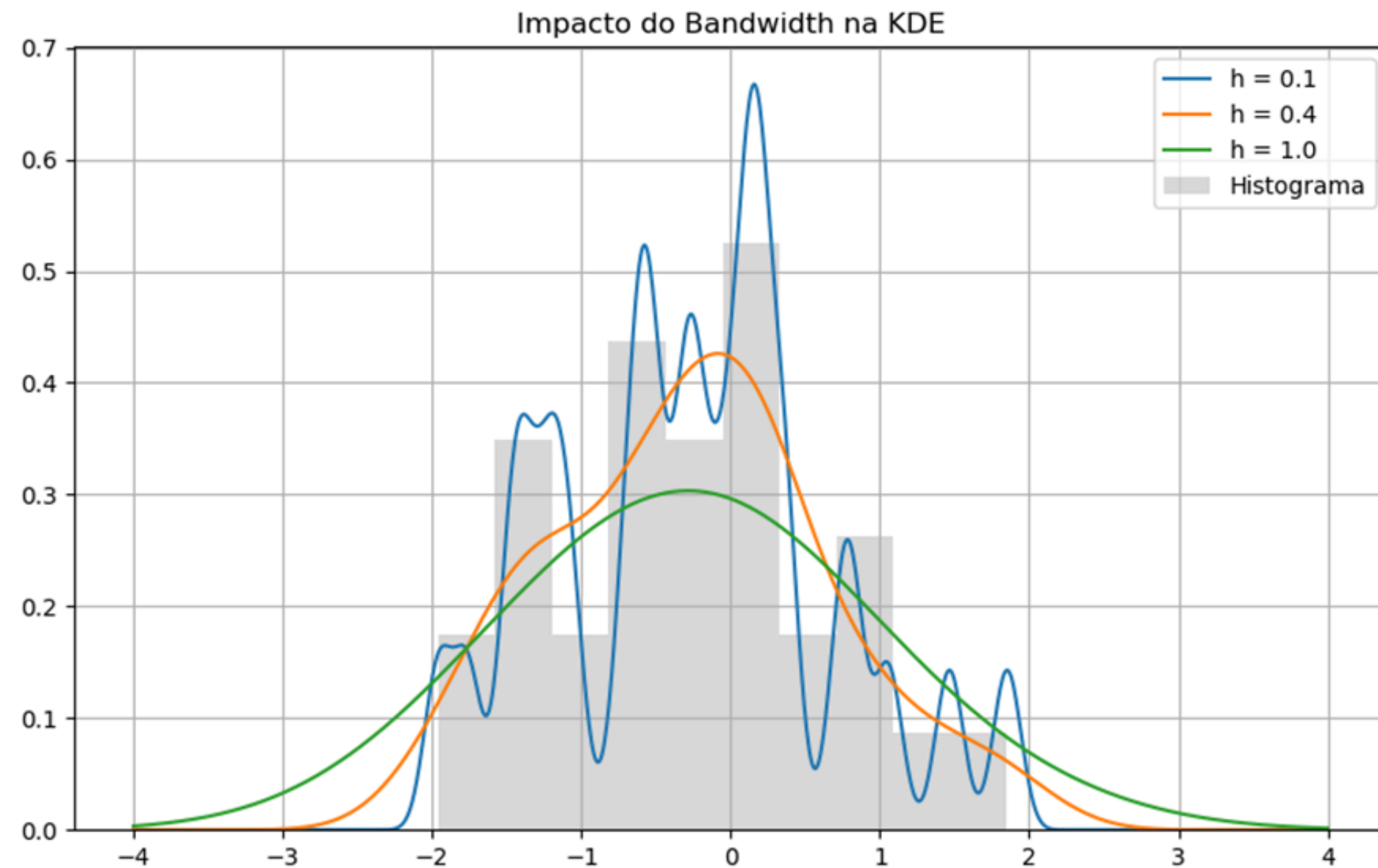
Kernel Density Estimation (KDE) é um método para estimar distribuições de probabilidade sem pressupor forma funcional (como normal, exponencial, etc.).



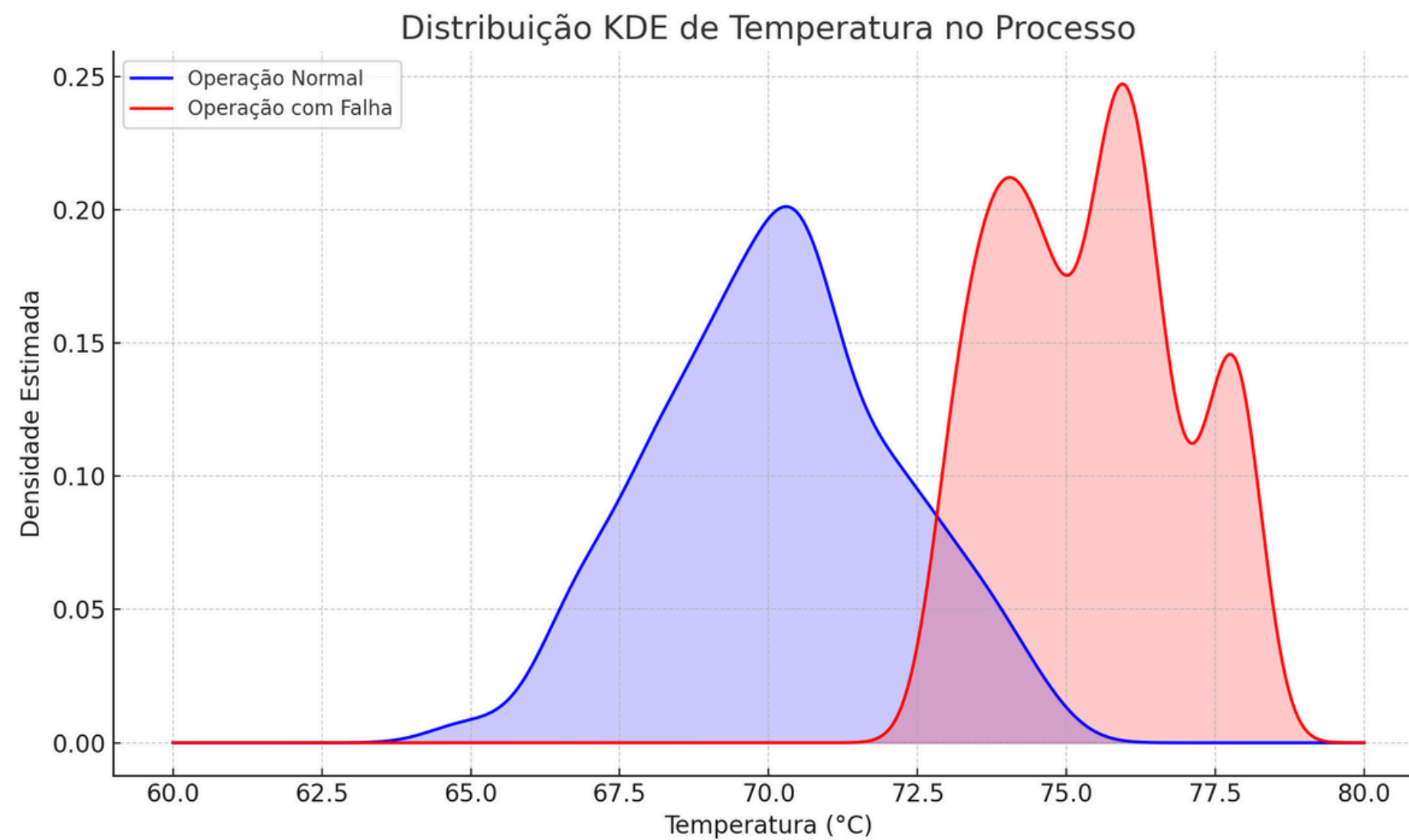
“Imagine que cada ponto de dado é uma vela que emite luz em forma de curva suave. KDE soma todas essas luzes para revelar a forma da paisagem dos dados.”

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

“A fórmula da KDE traduz o princípio de que cada dado ilumina seu entorno – juntos, revelam a densidade oculta.”



“Escolher o bandwidth certo é como ajustar o foco de uma lente: demais e você perde os detalhes, de menos e tudo vira ruído.”



```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde

# Simulando dados de temperatura em processo industrial
np.random.seed(0)
normal_operation = np.random.normal(loc=70, scale=2, size=100)
faulty_operation = np.random.normal(loc=75, scale=1.5, size=40)

# Geração do eixo x
x_grid = np.linspace(60, 80, 1000)

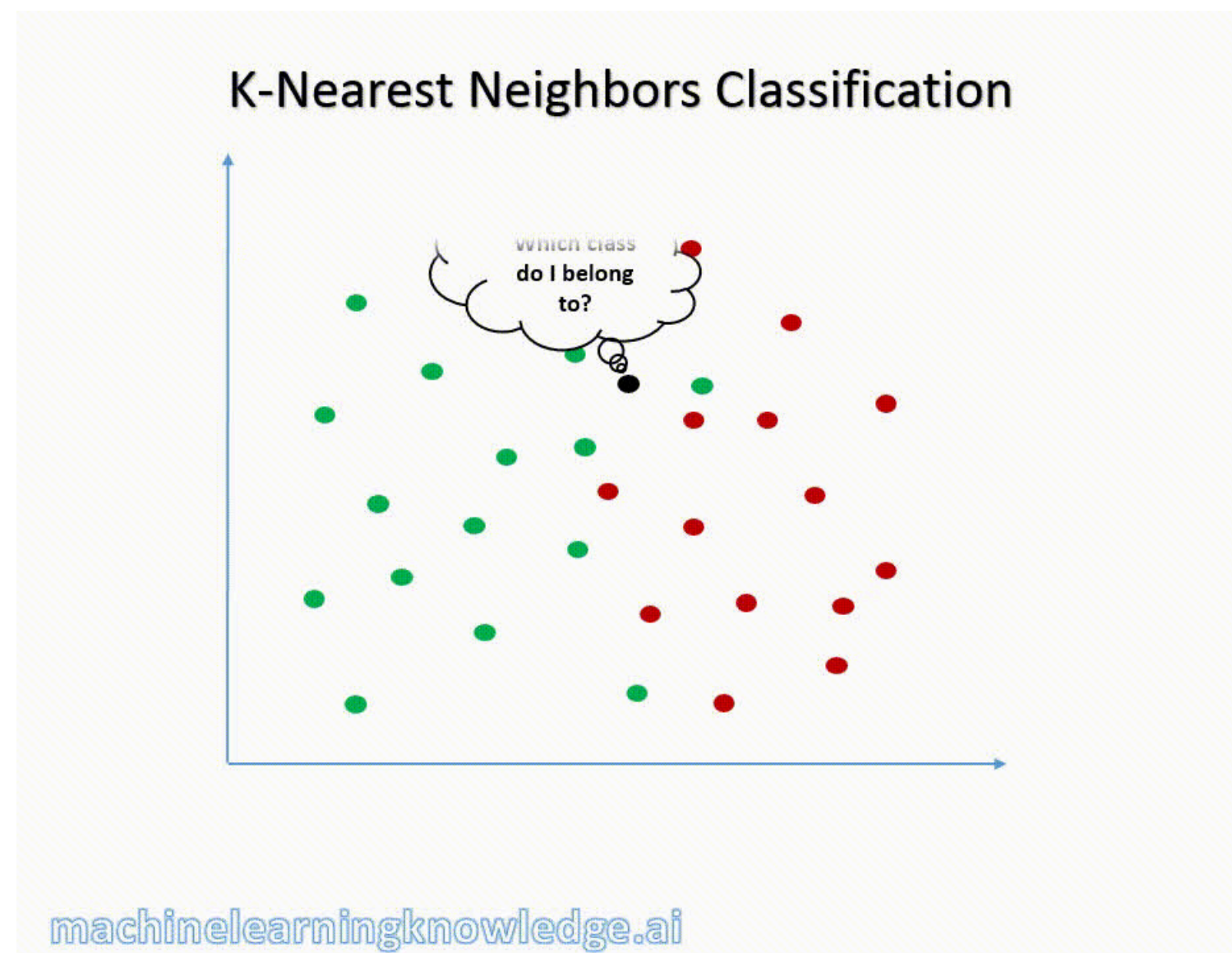
# Estimando densidades
kde_normal = gaussian_kde(normal_operation, bw_method=0.3)
kde_faulty = gaussian_kde(faulty_operation, bw_method=0.3)

# Plotando
plt.figure(figsize=(10, 6))
plt.plot(x_grid, kde_normal.evaluate(x_grid), label='Operação Normal', color='blue')
plt.plot(x_grid, kde_faulty.evaluate(x_grid), label='Operação com Falha', color='red')
plt.fill_between(x_grid, kde_normal.evaluate(x_grid), alpha=0.2, color='blue')
plt.fill_between(x_grid, kde_faulty.evaluate(x_grid), alpha=0.2, color='red')

plt.title("Distribuição KDE de Temperatura no Processo")
plt.xlabel("Temperatura (°C)")
plt.ylabel("Densidade Estimada")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

“KDE transforma dados brutos em insights visuais sobre o comportamento do processo.”

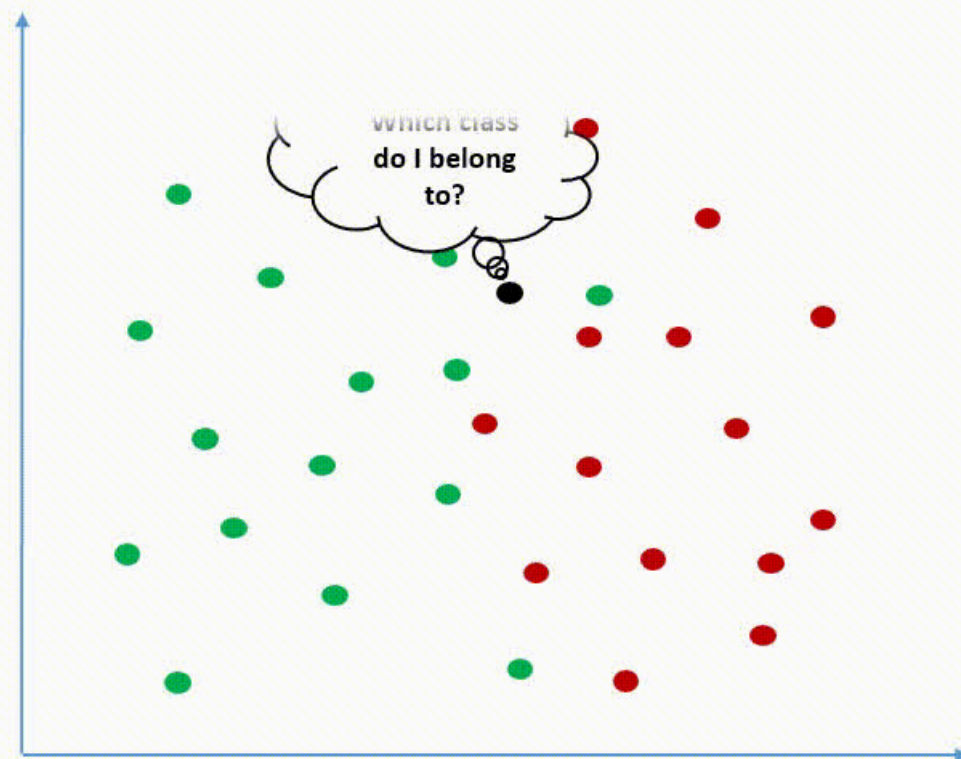
k-Nearest Neighbors (kNN) é um algoritmo de classificação e regressão baseado em similaridade local.



“No kNN, os dados antigos guiam a decisão presente.”

k-Nearest Neighbors (kNN) é um algoritmo de classificação e regressão baseado em similaridade local.

K-Nearest Neighbors Classification



machinelearningknowledge.ai

Etapas do Algoritmo

- Receber um ponto novo de entrada.
- Calcular a distância para todos os pontos do conjunto de treino.
- Selecionar os k pontos mais próximos.
- Classificar por maioria ou calcular a média dos valores.



Como Medir a Distância no kNN?



k-Nearest Neighbors (kNN) é um algoritmo de classificação e regressão baseado em similaridade local.

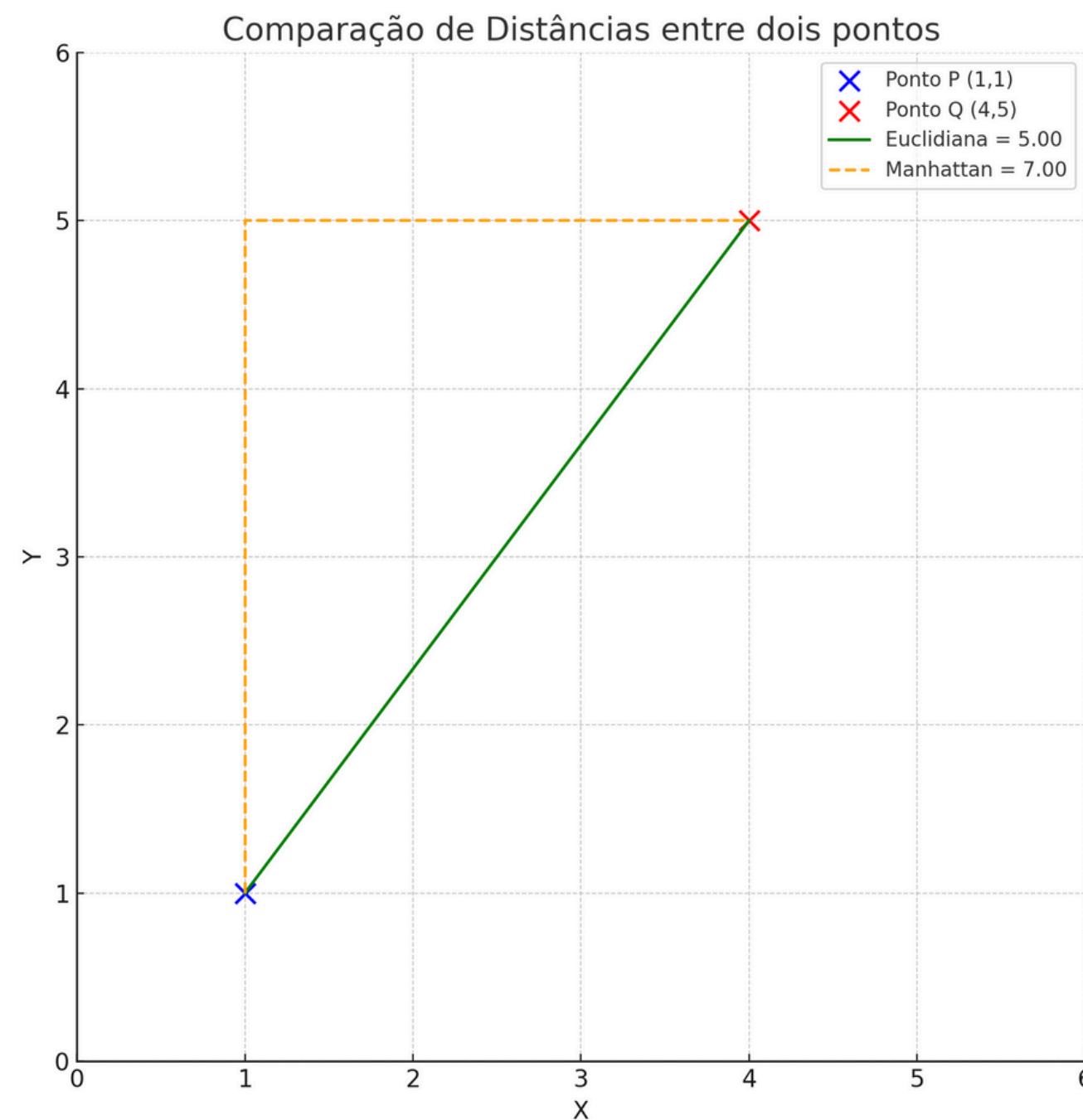
- Euclidiana;
- Manhattan;
- Minkowski;

$$d(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

$$d(p, q) = \left(\sum_{i=1}^n |p_i - q_i|^p \right)^{1/p}$$

k-Nearest Neighbors (kNN) é um algoritmo de classificação e regressão baseado em similaridade local.

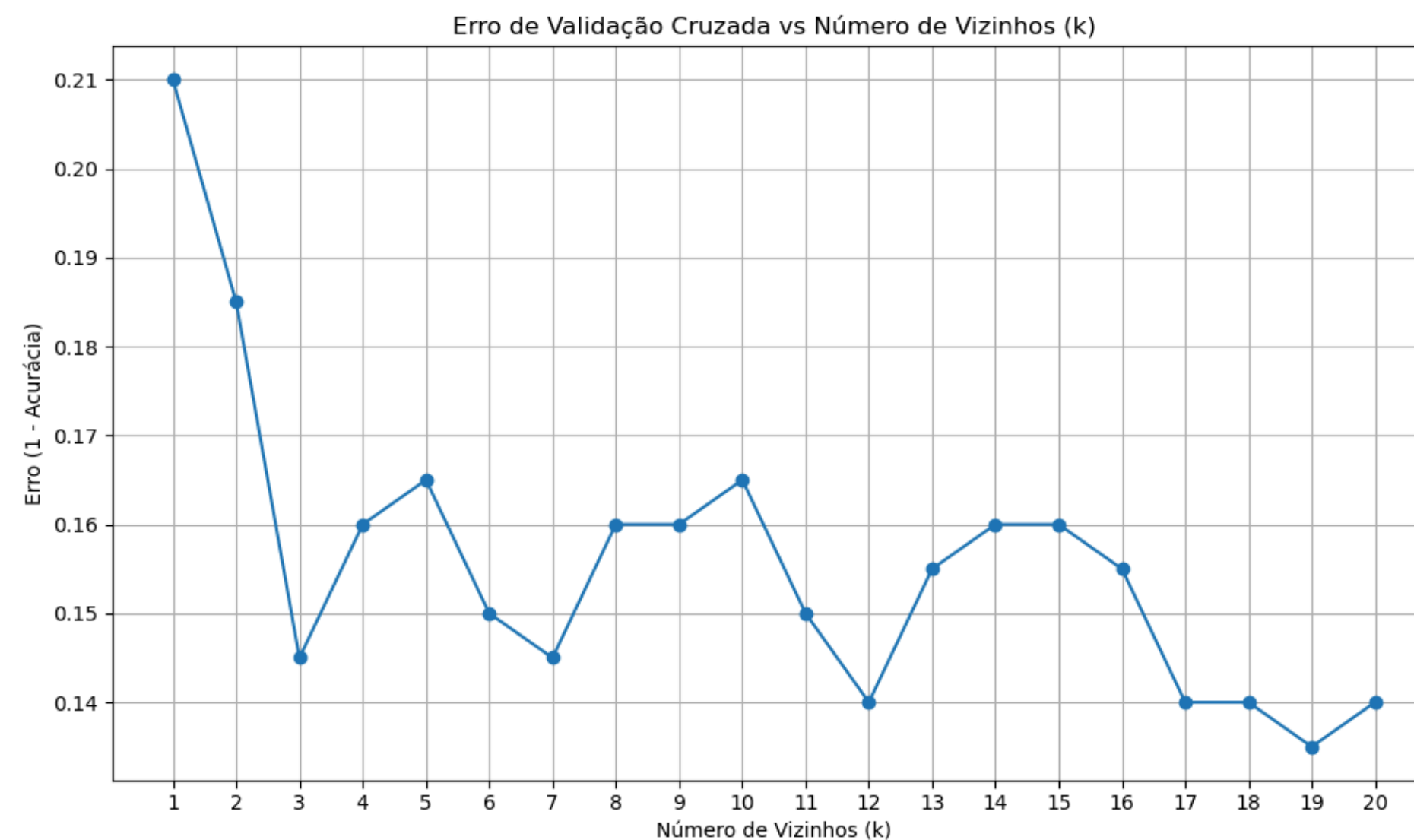
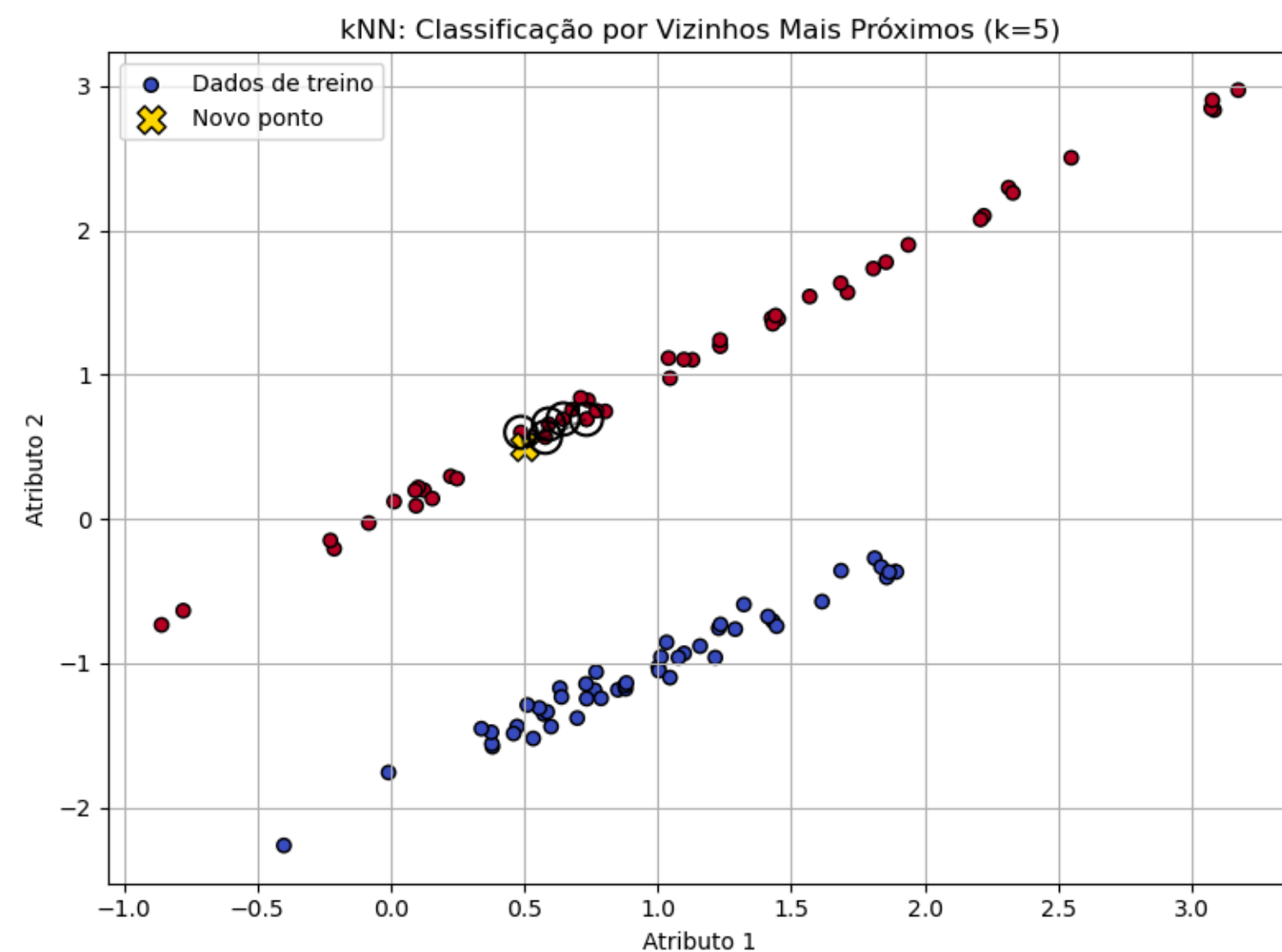




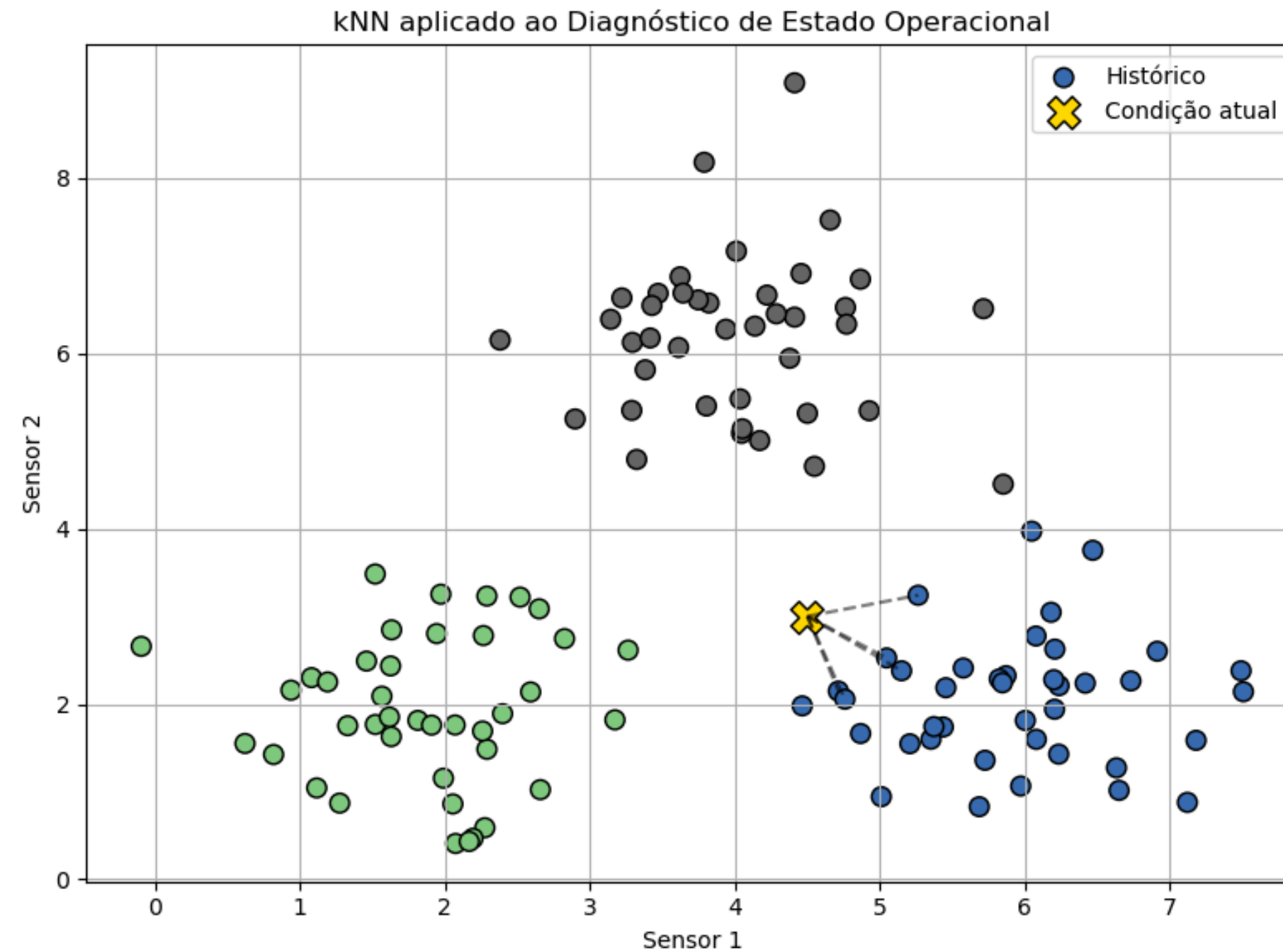
Como Escolher a quantidade de Vizinhos?



k-Nearest Neighbors (kNN) é um algoritmo de classificação e regressão baseado em similaridade local.



“Escolher k é como escolher quantos conselhos ouvir: poucos podem ser enganosos, muitos podem diluir a resposta certa.”



“kNN aprende com o passado e age no presente — sem fórmulas, só proximidade.”

Aspecto	KDE	KNN
Tipo de Tarefa	Estimativa de Densidade	Classificação / Regressão
Natureza dos Dados	Contínuos	Supervisionados
Saída	Função de Densidade	Rótulo ou valor Numérico
Modelo	Não Supervisionado	Supervisionado
Dependência de Dados	Alta	Alta*
Parâmetro Chave	Bandwidth (h)	Número de Vizinhos (k)
Aplicações	Visualização, detecção de outliers	Diagnóstico, previsão, classificação