



UNIVERSIDADE  
FEDERAL DO CEARÁ



LogIA

# Aprendizagem de Máquina

César Lincoln Cavalcante Mattos

2025

# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências

# Modelos e parâmetros

- O que são parâmetros?

# Modelos e parâmetros

- O que são parâmetros?
  - “Concentram” o que foi aprendido a partir dos dados.

# Modelos e parâmetros

- O que são parâmetros?
  - “Concentram” o que foi aprendido a partir dos dados.
- O que são hiperparâmetros?

# Modelos e parâmetros

- O que são parâmetros?
  - “Concentram” o que foi aprendido a partir dos dados.
- O que são hiperparâmetros?
  - Definem o “comportamento geral” do modelo (ou do algoritmo de aprendizagem).
- **Ideia:** Um modelo pode “não ter” parâmetros?

# Modelos e parâmetros

## Modelos paramétricos

- Sumarizam o que foi aprendido a partir dos dados em um **conjunto finito de parâmetros** em quantidade que não depende do número de padrões de treinamento.
- A complexidade/flexibilidade **não cresce** com mais dados.

# Modelos e parâmetros

## Modelos paramétricos

- Sumarizam o que foi aprendido a partir dos dados em um **conjunto finito de parâmetros** em quantidade que não depende do número de padrões de treinamento.
- A complexidade/flexibilidade **não cresce** com mais dados.

## Modelos não-paramétricos

- Admitem um **conjunto ilimitado de parâmetros**, ou seja, sua quantidade depende do número de padrões de treinamento.
- A complexidade/flexibilidade **cresce** com mais dados.



# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências

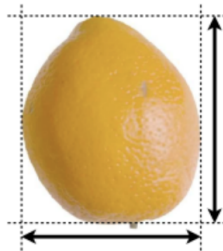
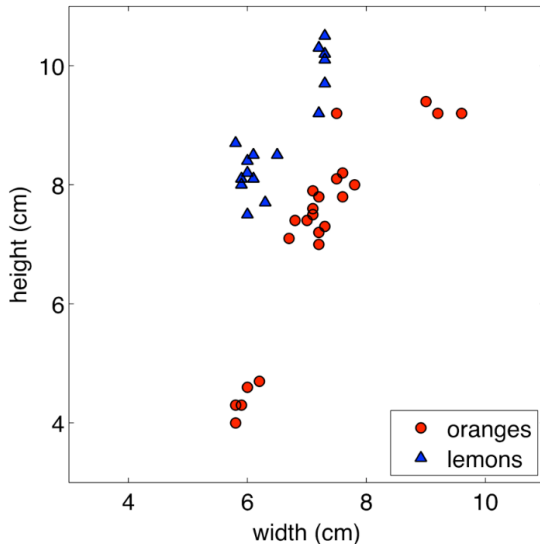
# Aprendizado baseado em instâncias

- **Problema:** Como diferenciar laranjas de limões?



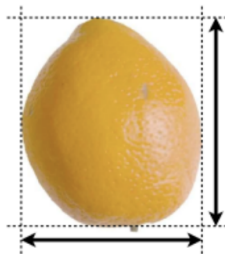
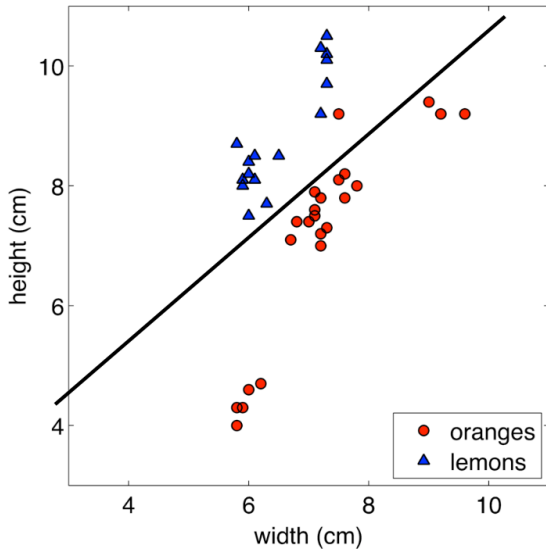
# Aprendizado baseado em instâncias

- **Ideia:** Mapeamos largura (*width*) e altura (*height*) das frutas.



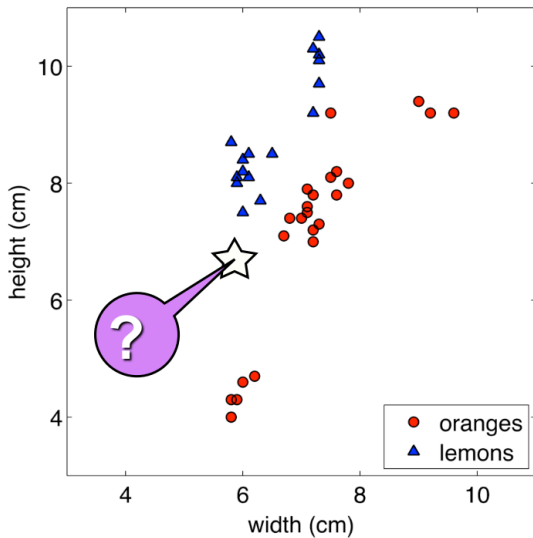
# Aprendizado baseado em instâncias

- **Ideia:** Usamos um classificador linear para separar as frutas.



# Aprendizado baseado em instâncias

- **Ideia:** Classificar um novo padrão a partir dos mais próximos?



# Aprendizado baseado em instâncias

## Modelos baseados em instâncias

- Modelos não-paramétricos.
- Podem não possuir uma etapa separada de treinamento.
- Predições são baseadas nas instâncias de treinamento mais próximas do padrão de teste.
- Precisam armazenar os dados de treinamento para realizar predições.

# Aprendizado baseado em instâncias

## Modelos baseados em instâncias

- Modelos não-paramétricos.
  - Podem não possuir uma etapa separada de treinamento.
  - Predições são baseadas nas instâncias de treinamento mais próximas do padrão de teste.
  - Precisam armazenar os dados de treinamento para realizar predições.
- 
- **Problema:** Como saber se um padrão está próximo de outro?

# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências



# Nearest Neighbors - Vizinhos mais próximos

- **Ideia:** Distância Euclidiana entre os padrões  $\mathbf{x}_i$  e  $\mathbf{x}_j$ :

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}$$

# Nearest Neighbors - Vizinhos mais próximos

- **Ideia:** Distância Euclidiana entre os padrões  $\mathbf{x}_i$  e  $\mathbf{x}_j$ :

$$d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}$$

## Nearest Neighbor (NN) para classificação

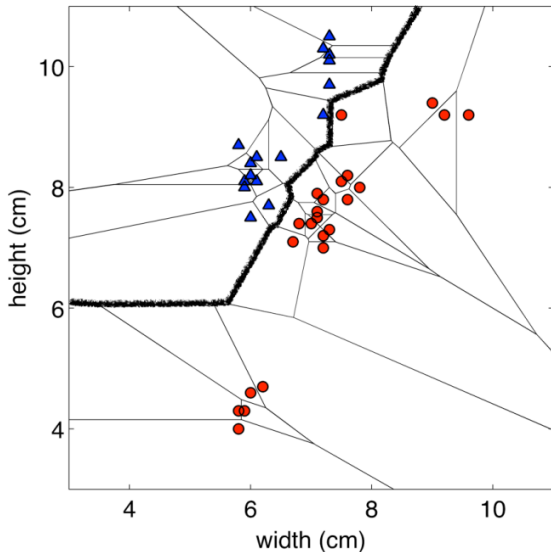
- 1 Encontre o padrão  $\mathbf{x}_{NN}$  de treinamento mais próximo do padrão de teste  $\mathbf{x}_*$ :

$$\mathbf{x}_{NN} = \arg \min_{\mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_N\}} d(\mathbf{x}_i, \mathbf{x}_*).$$

- 2 Retorne a classe  $y_{NN}$  associada a  $\mathbf{x}_{NN}$ .

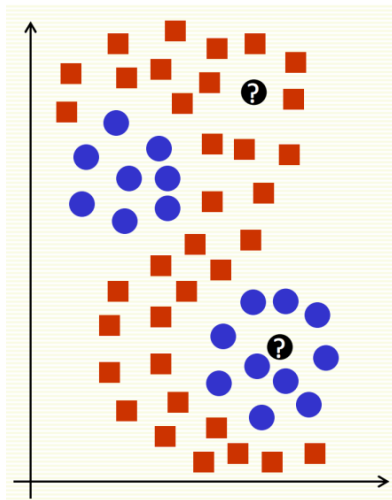
# Nearest Neighbors - Vizinhos mais próximos

- NN cria um **diagrama de Voronoi** e **fronteiras não-lineares**.



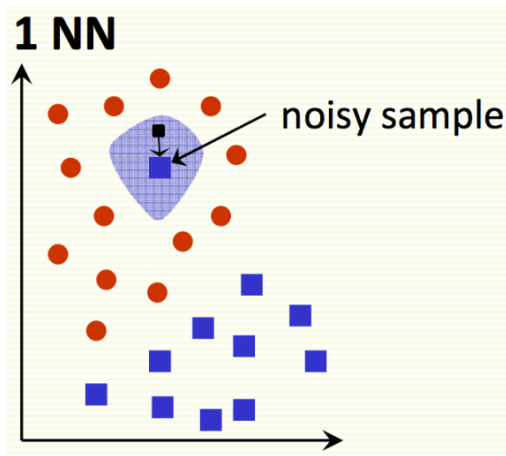
# Nearest Neighbors - Vizinhos mais próximos

- NN permite a classificação de **dados multi-modais**.



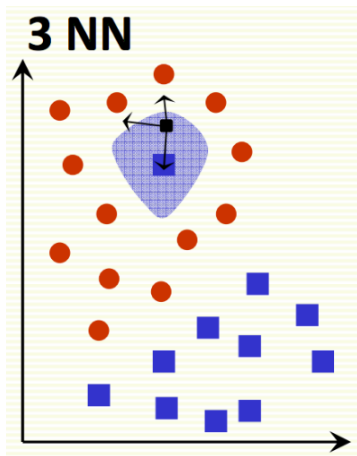
# Nearest Neighbors - Vizinhos mais próximos

- **Problema:** NN é muito sensível a **padrões anômalos**.



# Nearest Neighbors - Vizinhos mais próximos

- **Ideia:** Considerar mais de um vizinho mais próximo.



# Nearest Neighbors - Vizinhos mais próximos

## K Nearest Neighbors (KNN) para classificação

- 1 Encontre os  $K$  padrões  $\mathbf{x}_k, k \in \{1, \dots, K\}$  mais próximo do padrão de teste  $\mathbf{x}_*$ :

$$\mathbf{x}_{\text{KNN}} = \arg \min_{\mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_N\}} d(\mathbf{x}_i, \mathbf{x}_*).$$

- 2 Retorne a classe mais comum entre os  $K$  padrões encontrados.

# KNN - Observações importantes

- **Problema:** Como escolher o valor de  $K$ ?
  - Valores **muito altos** de  $K$  podem incluir informação de dados muito distantes e **simplificam** a região de decisão.
  - Valores **muito baixos** de  $K$  podem ser sensíveis a ruído e tornam a região de decisão **mais complexa**.



# KNN - Observações importantes

- **Problema:** Como escolher o valor de  $K$ ?
  - Valores **muito altos** de  $K$  podem incluir informação de dados muito distantes e **simplificam** a região de decisão.
  - Valores **muito baixos** de  $K$  podem ser sensíveis a ruído e tornam a região de decisão **mais complexa**.
- **Ideia:** Avaliar valores para o hiperparâmetro  $K$  via grid-search.

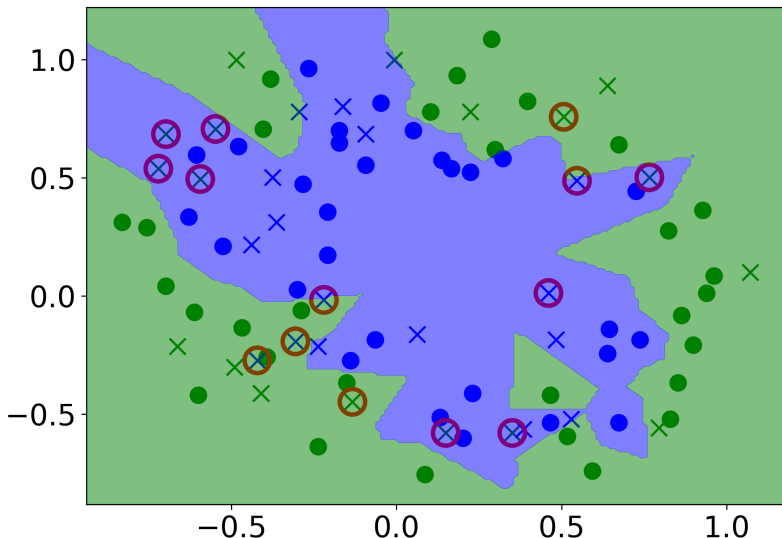
# KNN - Observações importantes

## Grid search para valor de K no modelo KNN

- 1 Separe o conjunto dados em treino, validação e teste;
- 2 Escolha um valor candidato para K;
- 3 Calcule o erro do KNN no conjunto de validação usando os dados de treino;
- 4 Repita os dois passos anteriores para todos os candidatos para K;
- 5 Escolha o valor de K com menor erro na validação;
- 6 Forme um novo conjunto de treino a partir do treino e validação anteriores;
- 7 Calcule o erro nos dados de teste usando o novo conjunto de treinamento e o melhor valor de K encontrado.

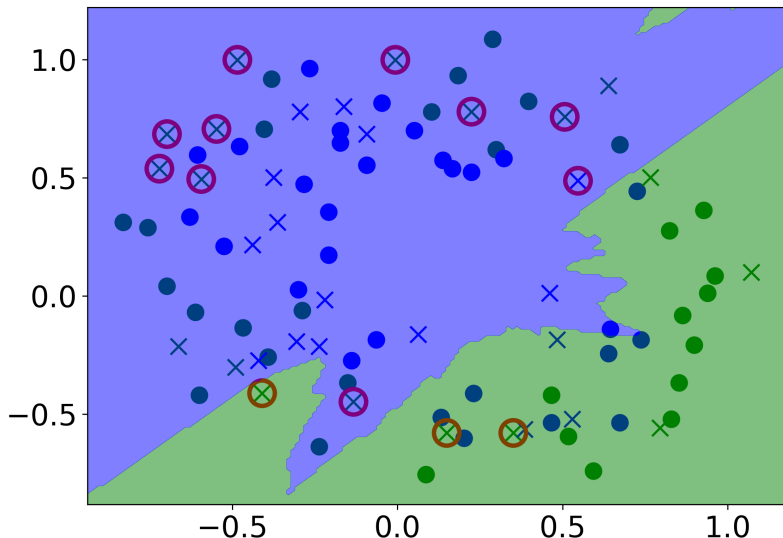
# KNN - Valor pequeno para K

1-NN - error = 41.18%



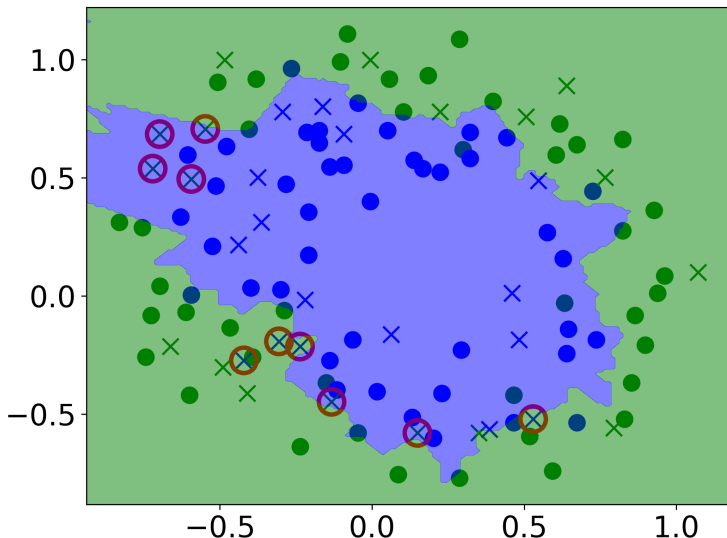
# KNN - Valor alto para K

25-NN - error = 38.24%



# KNN - K escolhido via grid search

5-NN distância 'euclidean' - error = 29.41%



# KNN como um modelo generativo

- Podemos interpretar o KNN para classificação como um modelo generativo.

# KNN como um modelo generativo

- Podemos interpretar o KNN para classificação como um modelo generativo.
- Considere uma esfera de volume  $V(\mathbf{x})$  em torno do ponto  $\mathbf{x}$  crescendo até conter  $K$  observações.

# KNN como um modelo generativo

- Podemos interpretar o KNN para classificação como um modelo generativo.
- Considere uma esfera de volume  $V(\mathbf{x})$  em torno do ponto  $\mathbf{x}$  crescendo até conter  $K$  observações.
- Sendo  $N_c(\mathbf{x})$  o número de exemplos da classe  $c$  na esfera e  $N_c$  o número total de exemplos da classe  $c$ , temos:

$$p(\mathbf{x}|y = c, \mathcal{D}) = \frac{N_c(\mathbf{x})}{N_c V(\mathbf{x})}.$$



# KNN como um modelo generativo

- Podemos interpretar o KNN para classificação como um modelo generativo.
- Considere uma esfera de volume  $V(\mathbf{x})$  em torno do ponto  $\mathbf{x}$  crescendo até conter  $K$  observações.
- Sendo  $N_c(\mathbf{x})$  o número de exemplos da classe  $c$  na esfera e  $N_c$  o número total de exemplos da classe  $c$ , temos:

$$p(\mathbf{x}|y = c, \mathcal{D}) = \frac{N_c(\mathbf{x})}{N_c V(\mathbf{x})}.$$

- Caso a priori de uma classe seja estimada por  $p(y = c|\mathcal{D}) = \frac{N_c}{N}$ , a posteriori pode ser obtida via regra de Bayes:

$$p(y = c|\mathbf{x}, \mathcal{D}) = \frac{\frac{N_c(\mathbf{x})}{N_c V(\mathbf{x})} \frac{N_c}{N}}{\sum_{c'} \frac{N_{c'}(\mathbf{x})}{N_{c'} V(\mathbf{x})} \frac{N_{c'}}{N}} = \frac{N_c(\mathbf{x})}{\sum_{c'} N_{c'}(\mathbf{x})} = \frac{N_c(\mathbf{x})}{K}.$$

# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências

# Nearest Neighbors - Vizinhos mais próximos

## K Nearest Neighbors (KNN) para regressão

- 1 Encontre os  $K$  padrões  $\mathbf{x}_k, k \in \{1, \dots, K\}$  mais próximo do padrão de teste  $\mathbf{x}_*$ :

$$\mathbf{x}_{\text{KNN}} = \arg \min_{\mathbf{x}_i \in \{\mathbf{x}_1, \dots, \mathbf{x}_N\}} d(\mathbf{x}_i, \mathbf{x}_*).$$

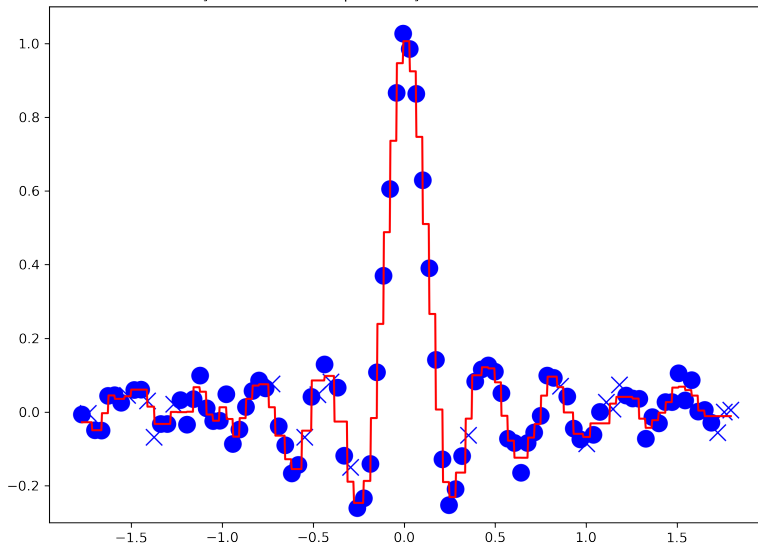
- 2 Retorne uma ponderação das saídas dos  $K$  padrões encontrados:

$$y_* = \frac{\sum_{k=1}^K \gamma_k y_k}{\sum_{k=1}^K \gamma_k}.$$

- Abordagens comuns para a ponderação das saídas:
  - Uniforme:  $\gamma_k = 1, \forall k$ .
  - Inversamente proporcional às distâncias:  $\gamma_k = \frac{1}{d(\mathbf{x}_k, \mathbf{x}_*)}, \forall k$ .

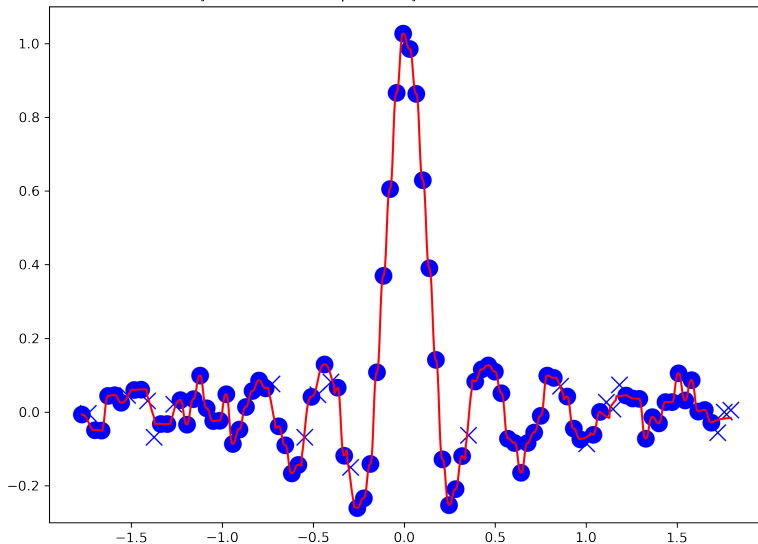
# KNN - Exemplo para regressão

2-NN com função 'euclidean' e ponderação 'uniform'. MSE no teste: 1.36e-03.



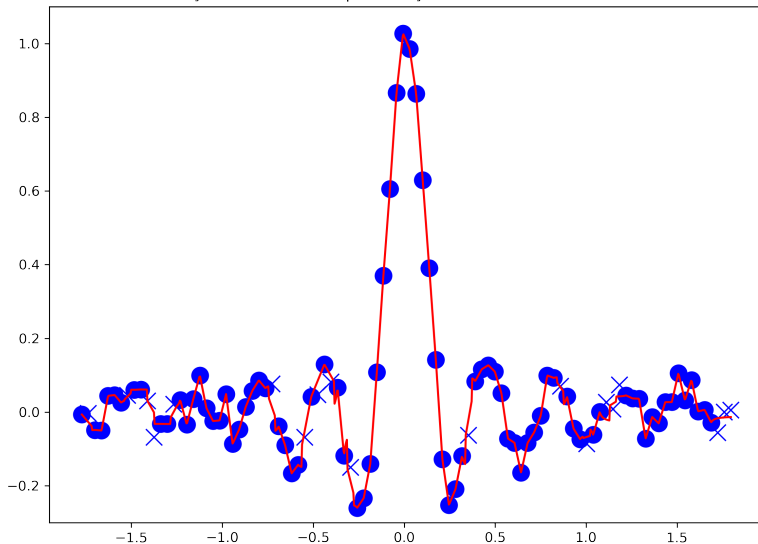
# KNN - Exemplo para regressão

2-NN com função 'euclidean' e ponderação 'distance'. MSE no teste: 1.03e-03.



# KNN - Exemplo para regressão

2-NN com função 'manhattan' e ponderação 'distance'. MSE no teste: 9.37e-04.



## KNN - Observações importantes

- A função de **distância (ou métrica)** é relevante para o KNN.

## KNN - Observações importantes

- A função de **distância (ou métrica)** é relevante para o KNN.
  - Distância **Euclidiana**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}.$$



# KNN - Observações importantes

- A função de **distância (ou métrica)** é relevante para o KNN.
  - Distância **Euclidiana**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}.$$

- Distância de **Manhattan**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{d=1}^D |x_{id} - x_{jd}|.$$

## KNN - Observações importantes

- A função de **distância (ou métrica)** é relevante para o KNN.
  - Distância **Euclidiana**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}.$$

- Distância de **Manhattan**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{d=1}^D |x_{id} - x_{jd}|.$$

- Distância de **Minkowski**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_p = \left( \sum_{d=1}^D |x_{id} - x_{jd}|^p \right)^{1/p}, p \geq 1.$$

## KNN - Observações importantes

- A função de **distância (ou métrica)** é relevante para o KNN.
  - Distância **Euclidiana**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_2 = \sqrt{\sum_{d=1}^D (x_{id} - x_{jd})^2}.$$

- Distância de **Manhattan**:

$$\|\mathbf{x}_i - \mathbf{x}_j\|_1 = \sum_{d=1}^D |x_{id} - x_{jd}|.$$

- Distância de **Minkowski**:

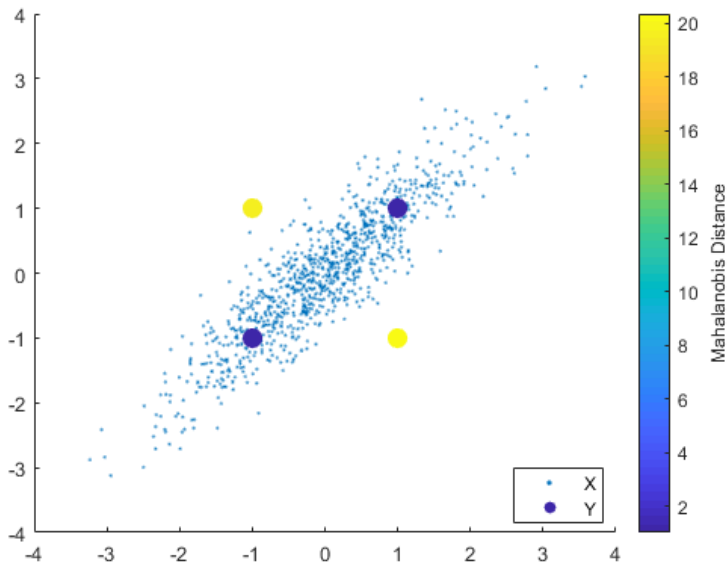
$$\|\mathbf{x}_i - \mathbf{x}_j\|_p = \left( \sum_{d=1}^D |x_{id} - x_{jd}|^p \right)^{1/p}, p \geq 1.$$

- Distância de **Mahalanobis**:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)},$$

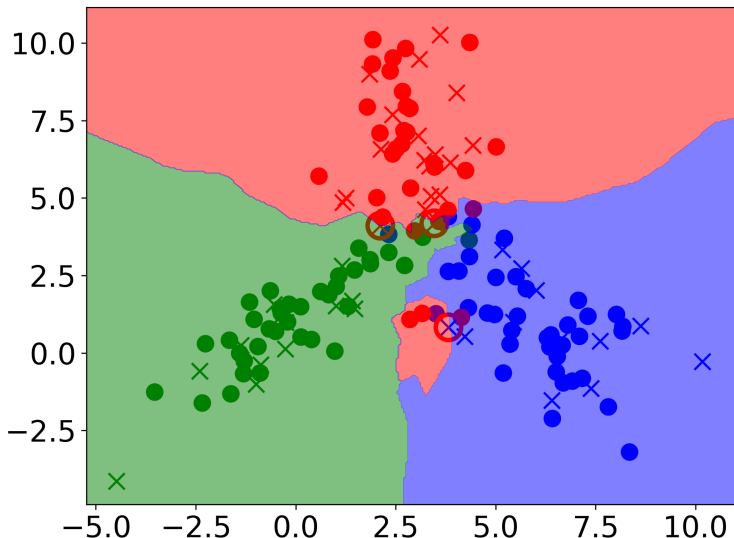
em que  $\Sigma$  é matriz de covariância dos dados de treinamento.

# Ilustração da distância de Mahalanobis

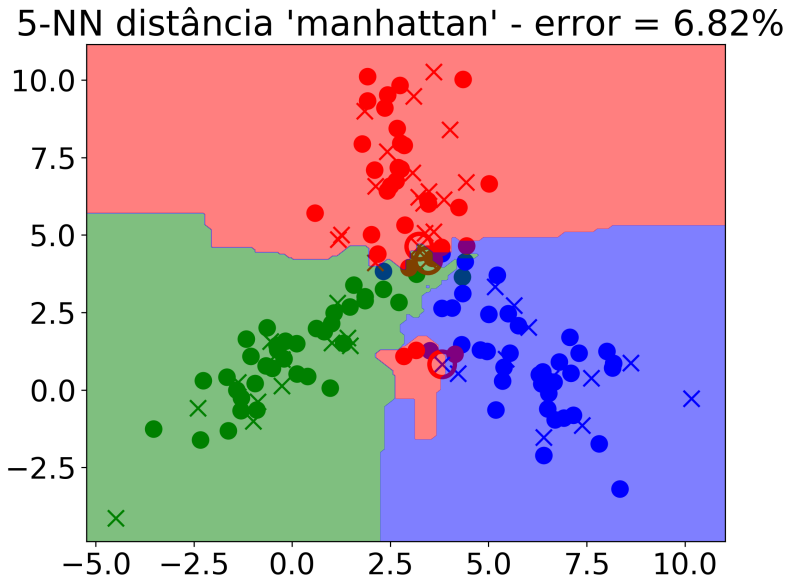


# KNN - Exemplos

5-NN distância 'euclidean' - error = 6.82%

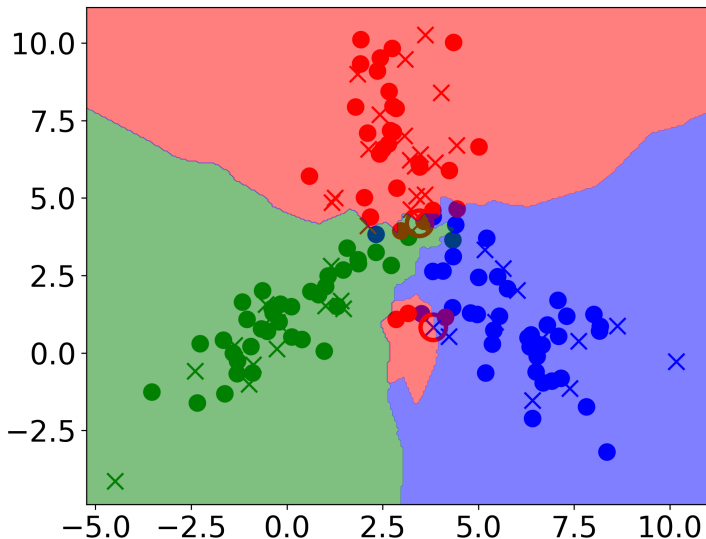


# KNN - Exemplos



# KNN - Exemplos

5-NN distância 'mahalanobis' - error = 4.55%



## KNN - Observações importantes

- **Problema:** Se alguns atributos tiverem magnitude muito maior que outros, eles serão tratados como mais importantes.

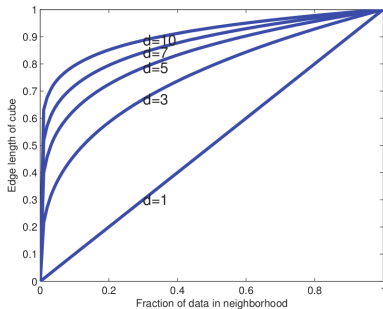
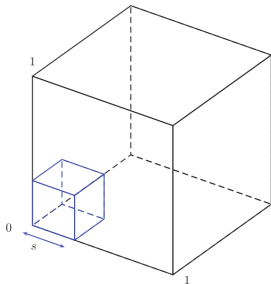


## KNN - Observações importantes

- **Problema:** Se alguns atributos tiverem magnitude muito maior que outros, eles serão tratados como mais importantes.
  - Normalize os dados.

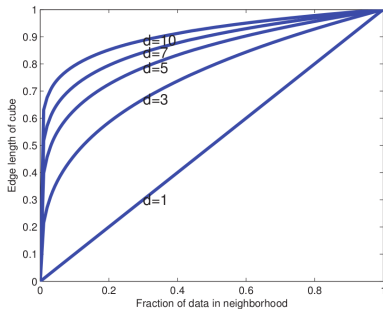
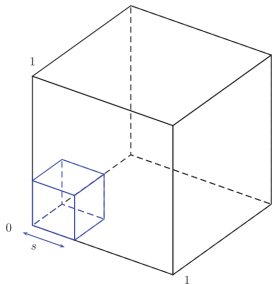
# KNN - Observações importantes

- **Problema:** Se alguns atributos tiverem magnitude muito maior que outros, eles serão tratados como mais importantes.  
→ Normalize os dados.
- **Problema:** Maldição da dimensionalidade?



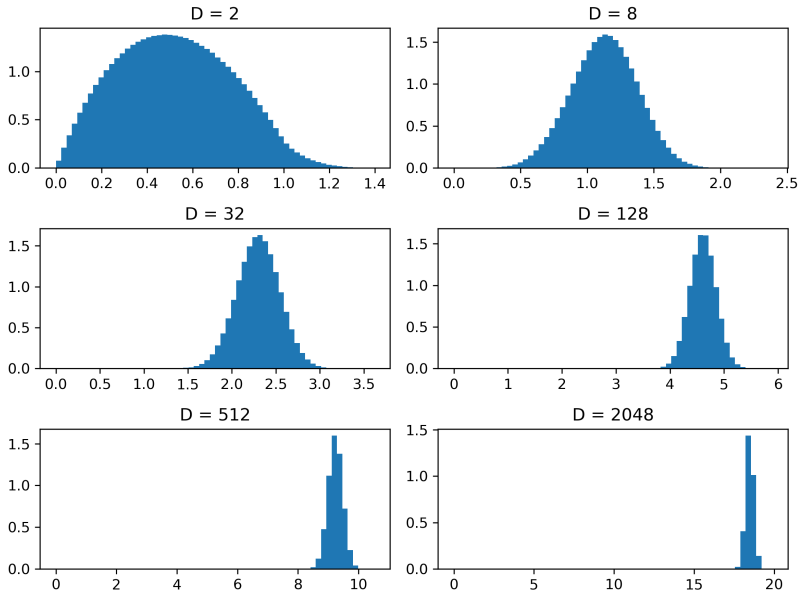
# KNN - Observações importantes

- **Problema:** Se alguns atributos tiverem magnitude muito maior que outros, eles serão tratados como mais importantes.
  - Normalize os dados.
- **Problema:** Maldição da dimensionalidade?

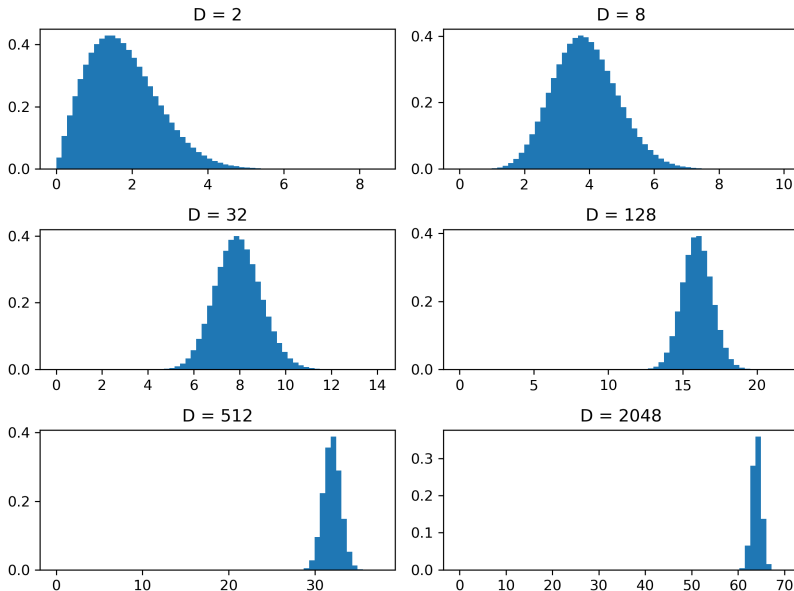


- A complexidade do espaço aumenta com mais atributos.
- A quantidade de dados necessários aumenta exponencialmente.
- Custo computacional aumenta com a dimensionalidade.
- Selecione/combine os atributos mais relevantes.

# Distância Euclidiana - dados uniformes



# Distância Euclidiana - dados Gaussianos



# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências

# Tópicos adicionais

- Técnicas para cálculo rápido de distâncias.
  - “Truques” matriciais.
    - Exemplo para distância Euclidiana ao quadrado entre o vetor  $\mathbf{x}$  e a matriz  $\mathbf{X}$  (em Python/Numpy):  
$$\text{dist2} = -2 * \mathbf{x} @ \mathbf{X.T} + (\mathbf{x}^{**2}).\text{sum}(\text{axis}=1) + (\mathbf{X}^{**2}).\text{sum}(\text{axis}=1)$$
  - Uso de implementações otimizadas, como o framework livre FAISS: <https://github.com/facebookresearch/faiss>
  - Uso de dados estruturados em árvores.
  - Métodos aproximados, como locality-sensitive hashing (LSH).
- Kernel KNN.
- Metric learning
  - Neighborhood components analysis (NCA).
  - Large margin nearest neighbor (LMNN).

# Agenda

- ① Modelos e parâmetros
- ② Aprendizado baseado em instâncias
- ③ Vizinhos mais próximos para classificação
- ④ Vizinhos mais próximos para regressão
- ⑤ Tópicos adicionais
- ⑥ Referências



# Referências bibliográficas

- **Cap. 1** - MURPHY, Kevin P. **Machine learning: a probabilistic perspective**, 2012.
- **Cap. 16** - MURPHY, Kevin P. **Probabilistic Machine Learning: An Introduction**, 2021.
- **Cap. 2** - BISHOP, C. **Pattern recognition and machine learning**, 2006.