SICO7A
SISTEMAS INTELIGENTES 1

Aula 04 B

Aprendizado baseado em Instâncias

Prof. Rafael G. **Mantovani**

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Apucarana - PR, Brasil

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação

Roteiro

1. Introdução
2. k-NN
3. DWNN
4. Exercício
5. Referências

Roteiro

1. **Introdução**
2. **k-NN**
3. **DWNN**
4. **Exercício**
5. **Referências**

Introdução

Introdução

- **Aprendizado Baseado em Instâncias**
  - *Instance-Base Learning* (IBL)

Introdução

- **Aprendizado Baseado em Instâncias**
  - *Instance-Base Learning* (IBL)

- Dois únicos itens necessários:
  - Alguma noção de distância (Ex: euclidiana)
  - Hipótese sobre a semelhança entre pontos próximos

Introdução

- **Classificador:**
  - armazena **exemplos** de treinamento
  - **exemplos:** também são denominados **instâncias**
  - não existe um **modelo**

Introdução

- IBL: **generaliza** informações com base nos exemplos de treinamento:
  - Para inferir a classe de novas instâncias
  - Cada vez que uma instância é recebida, computa-se uma função objetivo, com base no conhecimento oferecido pela base de exemplos de treinamento
  - Estima-se a classe da nova instância com base em comportamentos locais

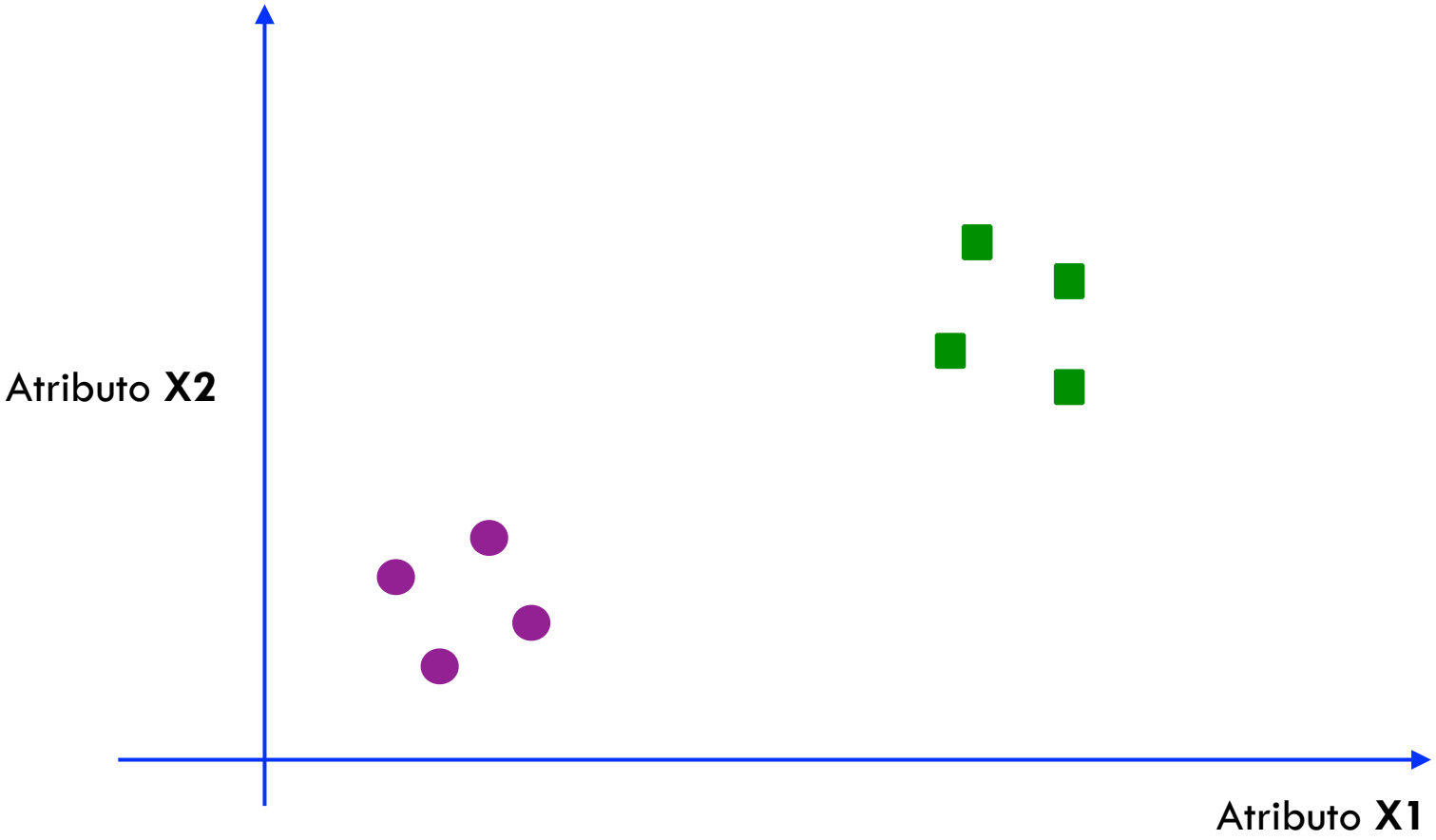
Introdução

- IBL: **generaliza** informações com base nos exemplos de treinamento:
  - Para inferir a classe de novas instâncias
  - Cada vez que uma instância é recebida, computa-se uma função objetivo, com base no conhecimento oferecido pela base de exemplos de treinamento
  - Estima-se a classe da nova instância com base em comportamentos locais

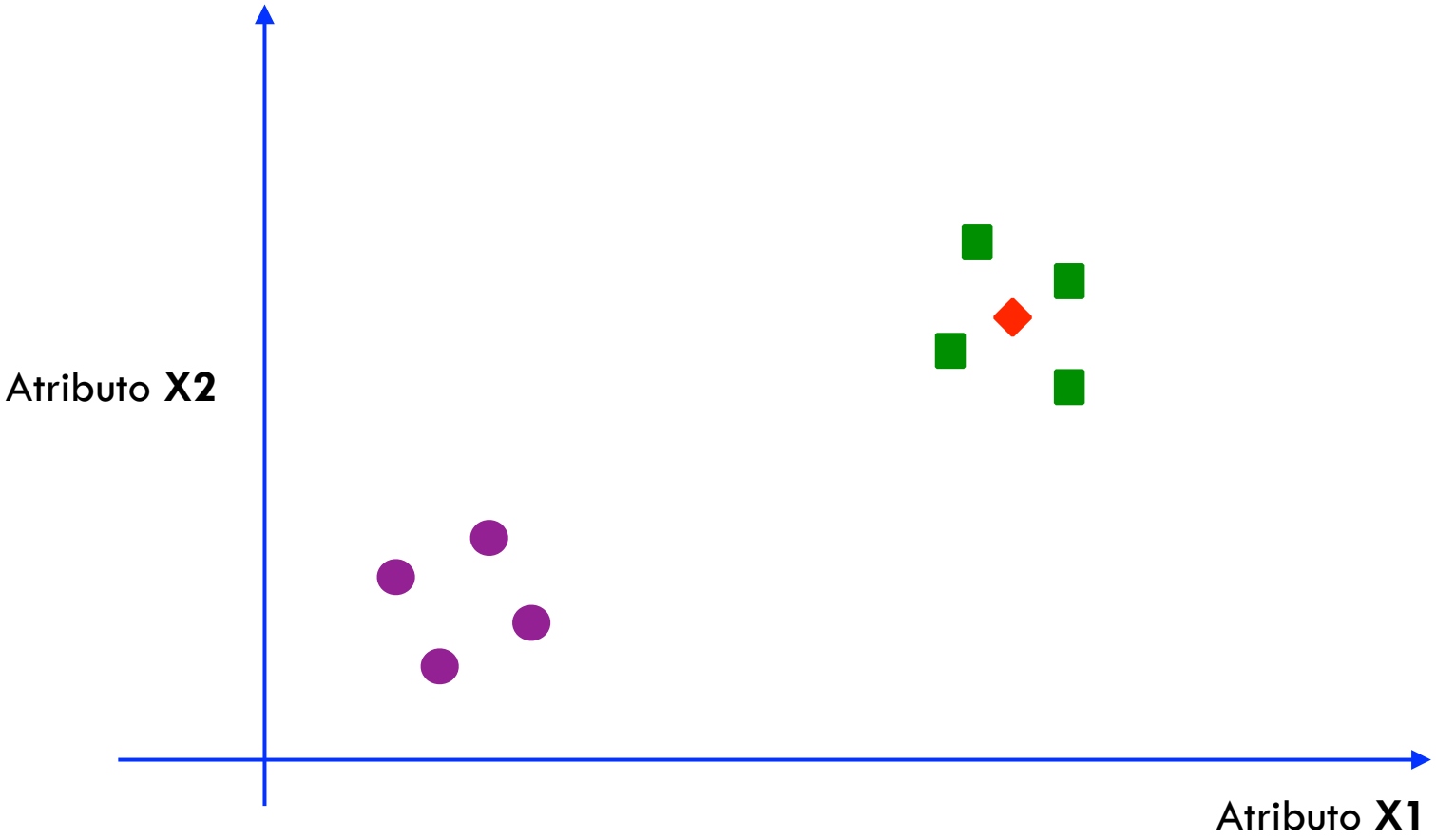
É uma técnica incremental!

Introdução

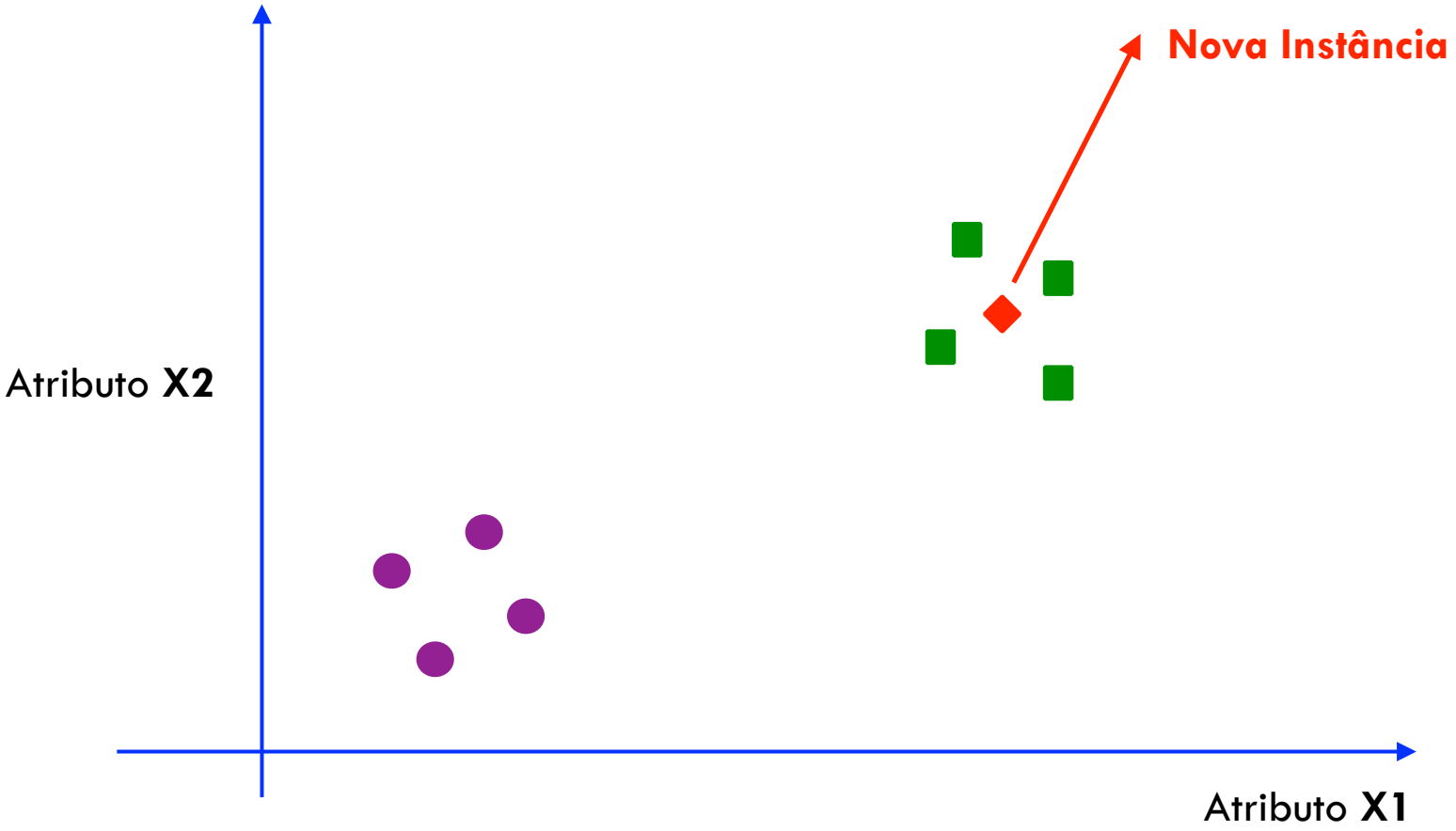
Introdução



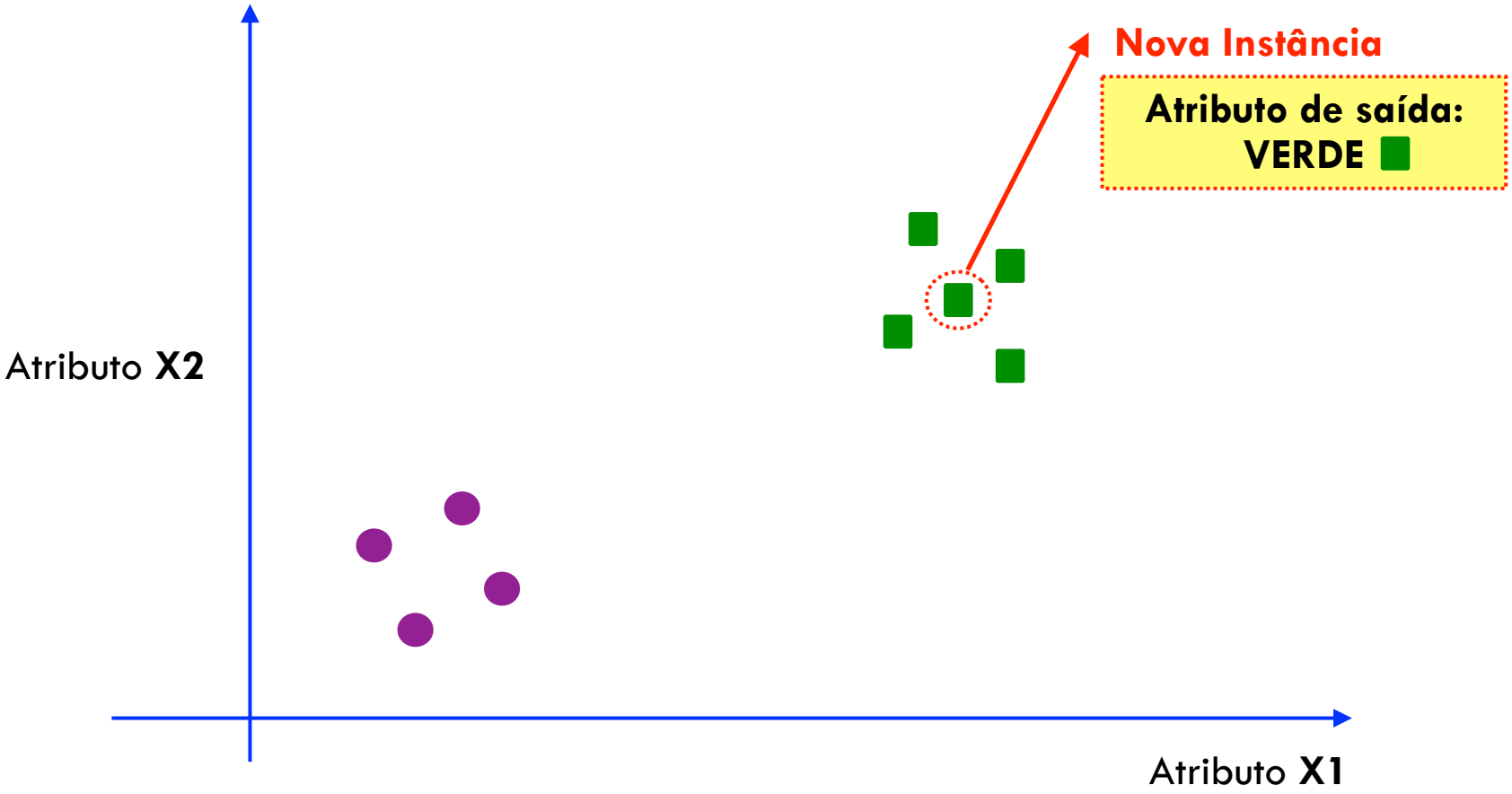
Introdução



Introdução



Introdução


Nova Instância
Atributo de saída:
VERDE
Atributo X2
Atributo X1

Roteiro

1. Introdução
2. k-NN
3. DWNN
4. Exercício
5. Referências

k-Nearest Neighbors

- k-Vizinhos Mais Próximos
  - *k-Nearest Neighbors* (**k-NN**)
  - técnica mais comum de IBL

k-Nearest Neighbors

k-NN

k-Nearest Neighbors

k-NN

1. Calcula-se a distância euclidiana da nova instância em função de cada instância da base de conhecimento

k-Nearest Neighbors

k-NN

1. Calcula-se a distância euclidiana da nova instância em função de cada instância da base de conhecimento

2. Seleciona-se as K instâncias de treinamento mais próximas

k-Nearest Neighbors

k-NN

1. Calcula-se a distância euclidiana da nova instância em função de cada instância da base de conhecimento

2. Seleciona-se as K instâncias de treinamento mais próximas

3. Define-se o atributo de saída:
   - **Discreta**: maior número de votos
   - **Contínua**: ponderação das saídas das K instâncias mais próximas

k-Nearest Neighbors

k-NN

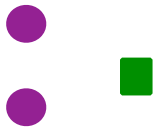
1. Calcula-se a distância euclidiana da nova instância em função de cada instância da base de conhecimento

2. Seleciona-se as K instâncias de treinamento mais próximas

3. Define-se o atributo de saída:
   - **Discreta**: maior número de votos (**moda**)
   - **Contínua**: ponderação das saídas das K instâncias mais próximas (**média**)

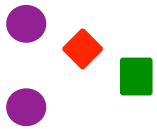
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



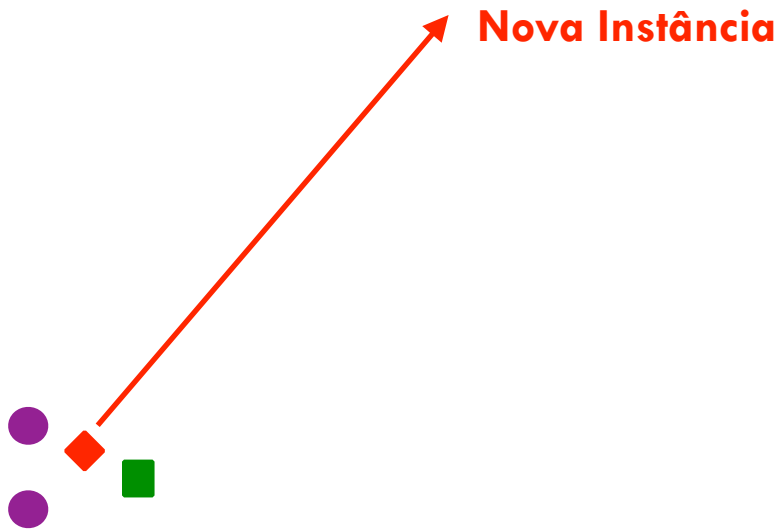
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



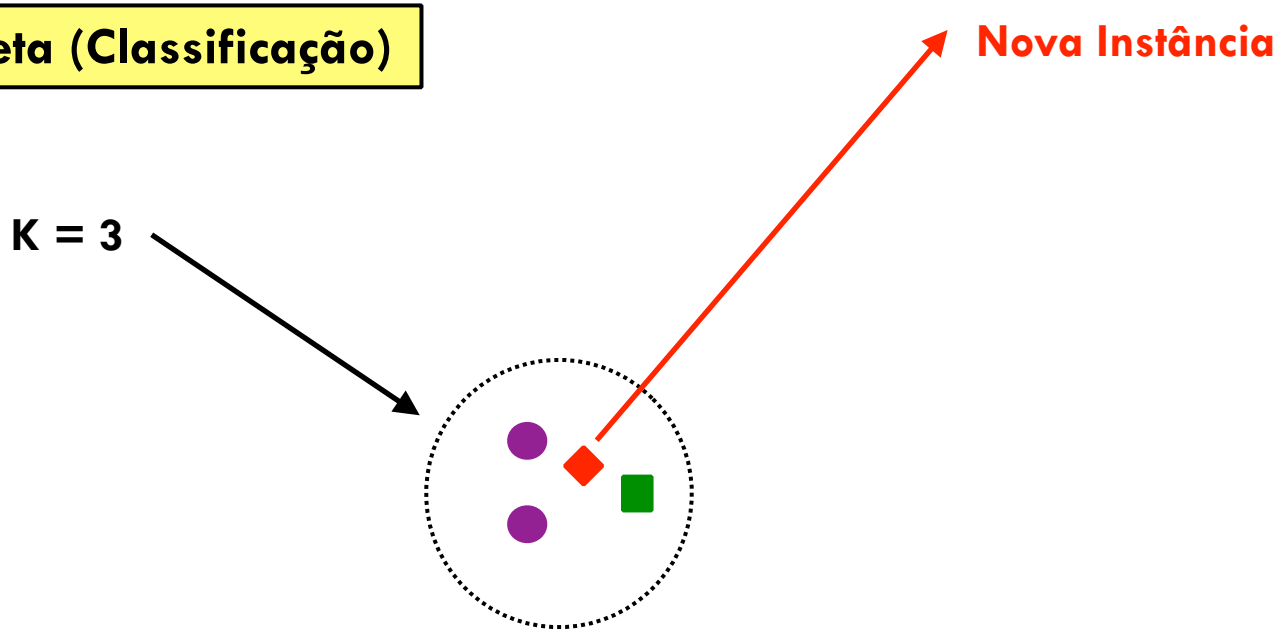
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



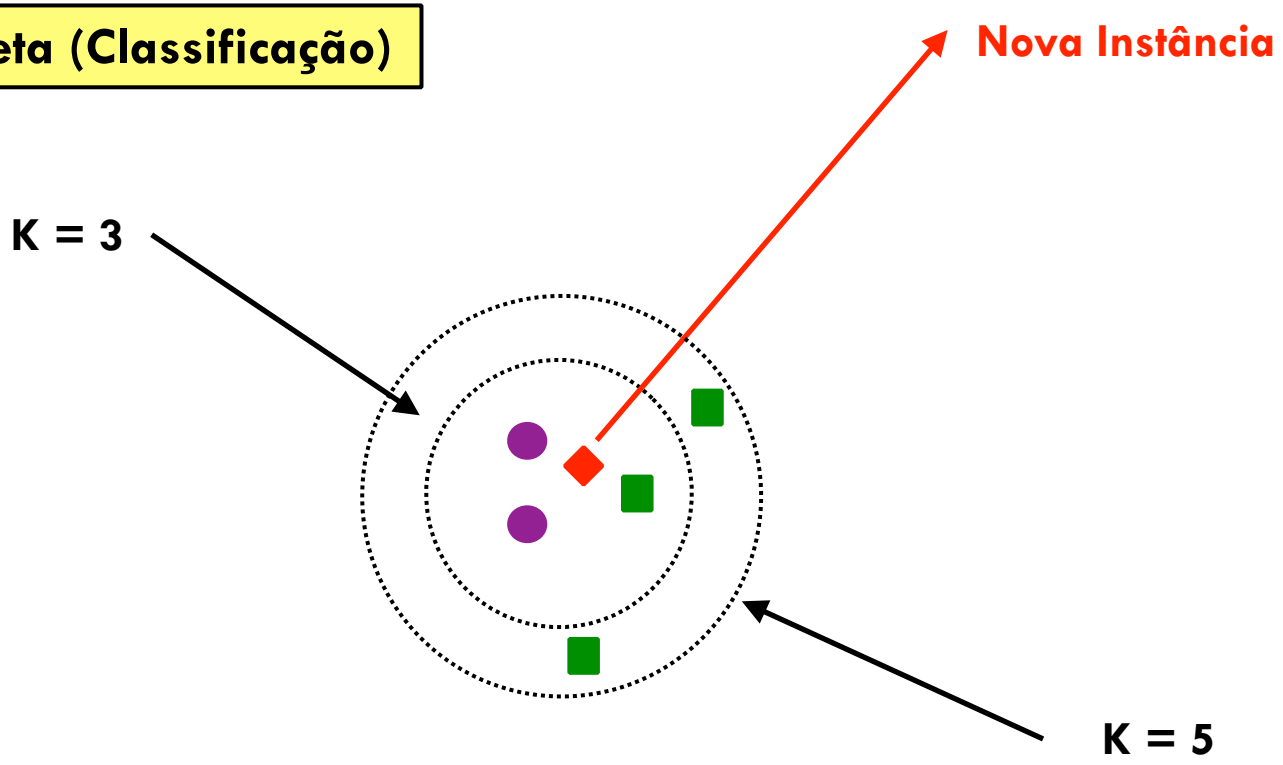
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



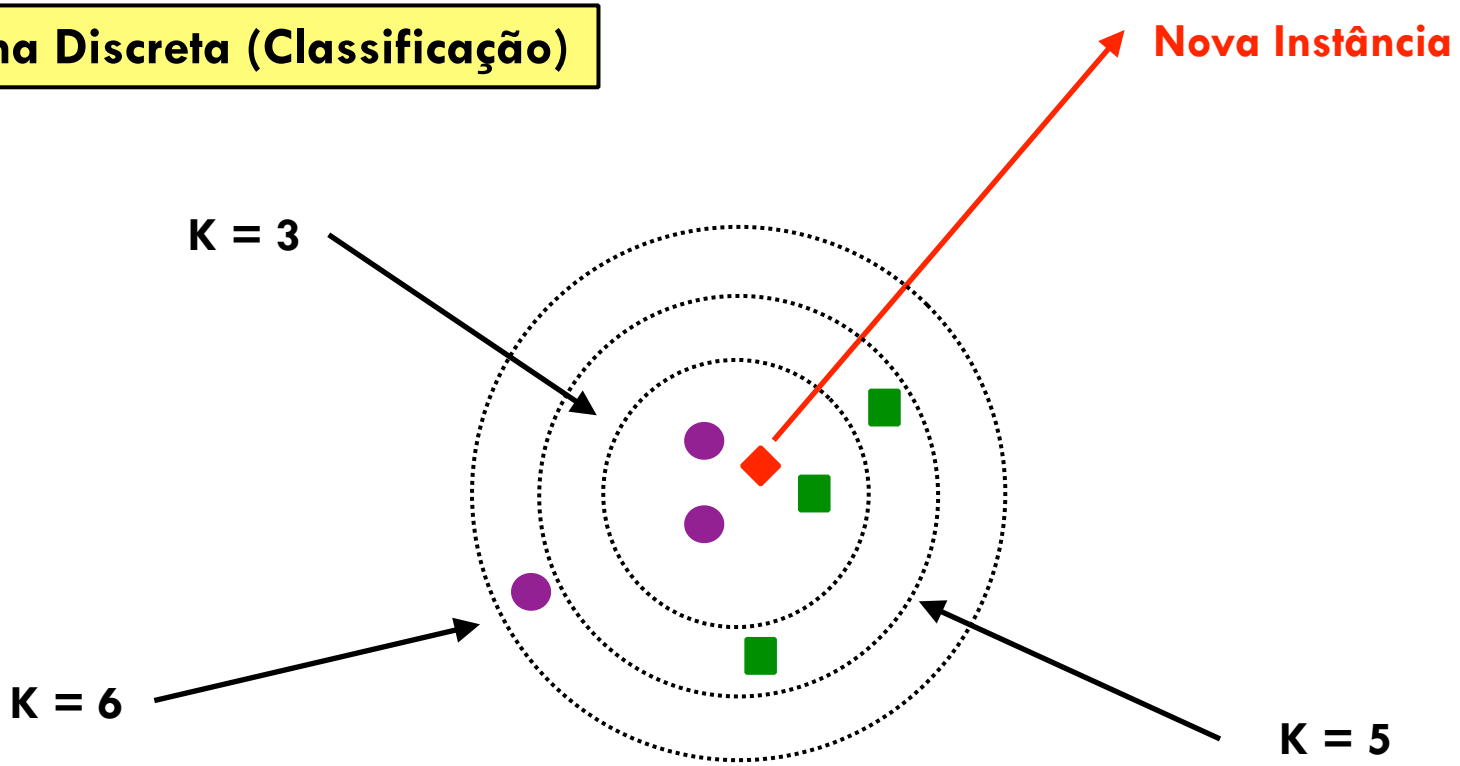
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



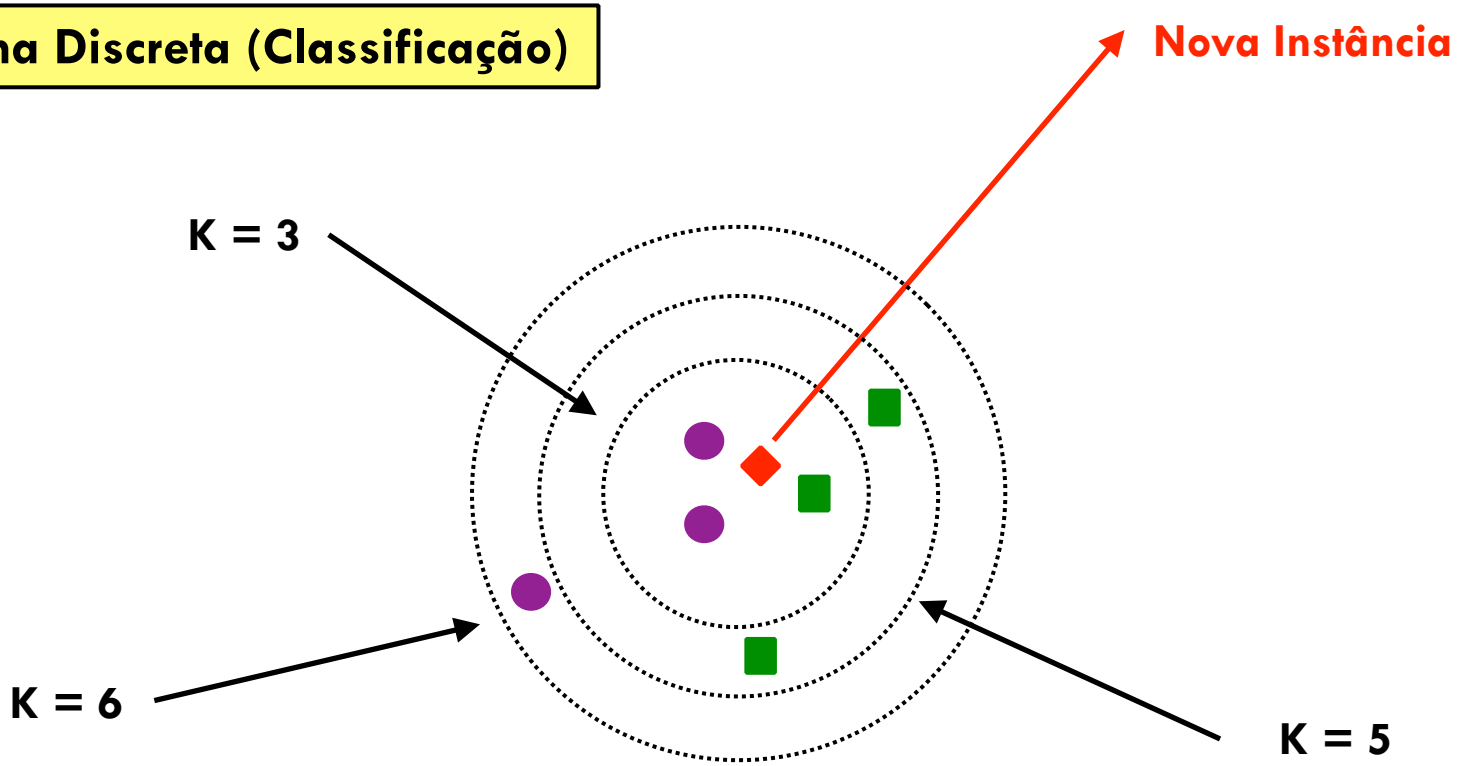
k-Nearest Neighbors

*** Forma Discreta (Classificação)**



k-Nearest Neighbors

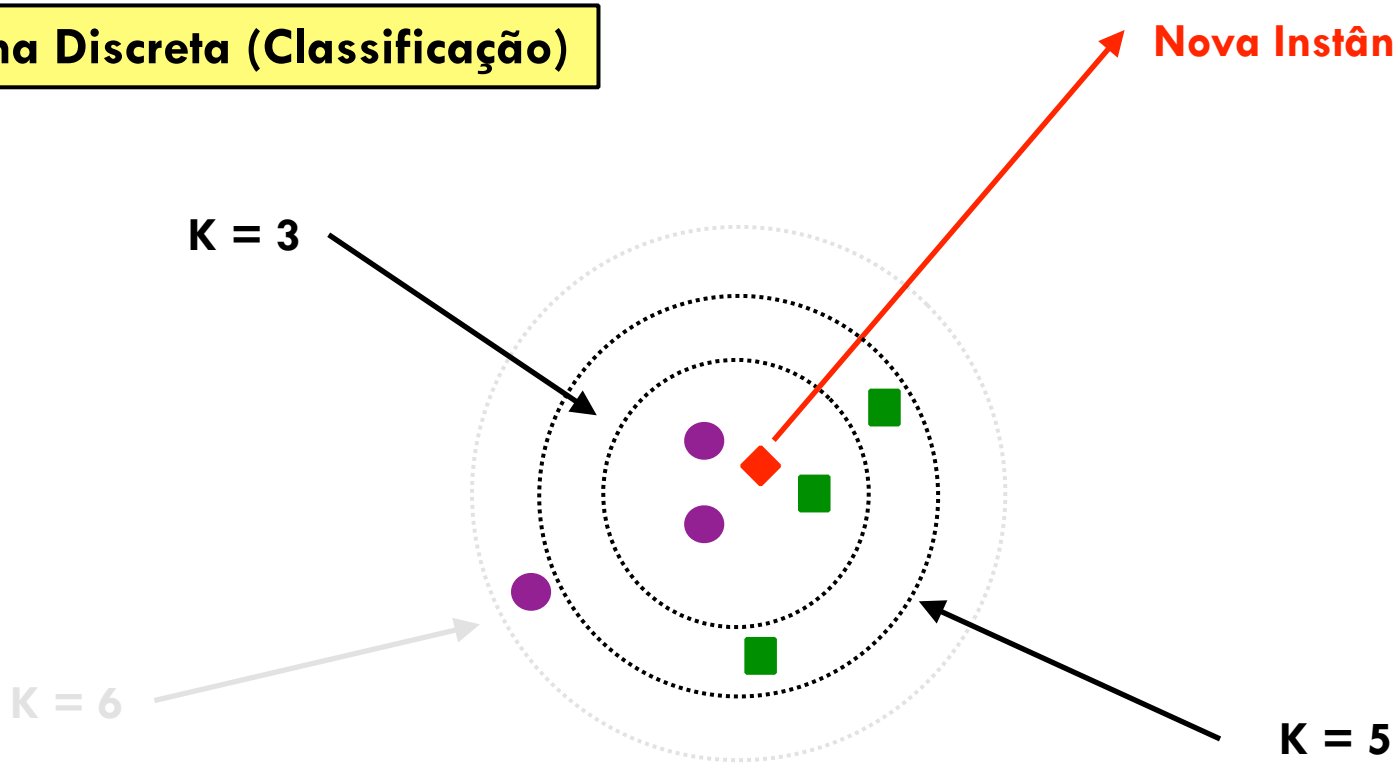
*** Forma Discreta (Classificação)**



K par pode gerar empate em termos de votos!

k-Nearest Neighbors

*** Forma Discreta (Classificação)**



K par pode gerar empate em termos de votos!

k-Nearest Neighbors

* **Forma Contínua (Regressão)**

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

k-Nearest Neighbors

* Forma Contínua (Regressão)

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} f(x_i)}{k}$$

* Considera-se uma média do atributo de saída de seus K vizinhos

Roteiro

1. Introdução
2. k-NN
3. **DWNN**
4. Exercício
5. Referências

Distance-Weighted Nearest Neighbors

Distance-Weighted Nearest Neighbors

- Vizinhos Mais Próximos com Distância Ponderada
  - *Distance-Weighted Nearest Neighbors* (**DWNN**)
  - Refinamento ou variação do k-NN
  - Considera uma **ponderação** da influência de cada vizinho em função de sua distância à nova instância

Distance-Weighted Nearest Neighbors

*** Forma Discreta (Classificação)**

$$\hat{f}(x_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \, \delta(v, f(x_i))$$

$$w_i = \frac{1}{d(x_1, x_i)^2}$$

Distance-Weighted Nearest Neighbors

*** Forma Discreta (Classificação)**

$$\hat{f}(x_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} w_i \, \delta(v, f(x_i))$$

$$w_i = \frac{1}{d(x_1, x_i)^2}$$

OBS: Se a nova instância é exatamente igual a uma instância de treinamento, então d(.) = 0, logo deve-se prever essa situação e o algoritmo deve retornar $f(x_i)$

Distance-Weighted Nearest Neighbors

*** Forma Contínua (Regressão)**

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} w_i \, f(x_i)}{\sum_{i=1}^{k} w_i}$$

$$w_i = \frac{1}{d(x_1, x_i)^2}$$

Distance-Weighted Nearest Neighbors

*** Forma Contínua (Regressão)**

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} w_i \, f(x_i)}{\sum_{i=1}^{k} w_i}$$

→ **Média ponderada**

$$w_i = \frac{1}{d(x_1, x_i)^2}$$

Distance-Weighted Nearest Neighbors

*** Forma Contínua (Regressão)**

$$\hat{f}(x_q) = \frac{\sum_{i=1}^{k} w_i \, f(x_i)}{\sum_{i=1}^{k} w_i}$$

→ **Média ponderada**

$$w_i = \frac{1}{d(x_1, x_i)^2}$$

*** Se a nova instância é exatamente a uma instância de treinamento, então d(.) = 0, logo deve-se prever essa situação e o algoritmo deve retornar $f(x_i)$**

Distance-Weighted Nearest Neighbors

- Essas duas variantes de k-NN consideram os k vizinhos mais próximos
  - No entanto, para DWNN, podemos considerar todas as instâncias de treinamento, pois as mais distantes terão pouca ou nenhum influência na instância de consulta
  - A desvantagem é que o classificador fica mais lento

  - Se somente os k mais próximos forem considerados:
    - Algoritmo é denominado **local**
  - Se todas as instâncias forem consideradas:
    - Algoritmo é denominado **global**

Questões

- As duas versões sempre:
  - computam a saída de função de todos atributos de entrada
  - Logo, se os atributos de entrada não forem bem definidos, o resultado pode ser ruim
  - Uma abordagem para resolver esse problema é dar um peso para cada atributo
  - Outra é remover os atributos menos significativos

Roteiro

1. Introdução
2. k-NN
3. DWNN
4. Exercício
5. Referências

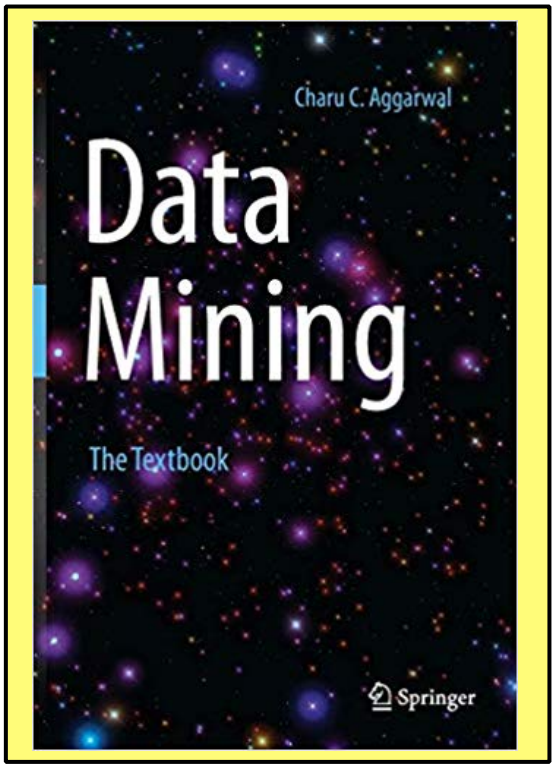
Exercício

- Implemente os algoritmos k-NN e DWNN para problemas de classificação. Use o **dataset Iris** para testes.

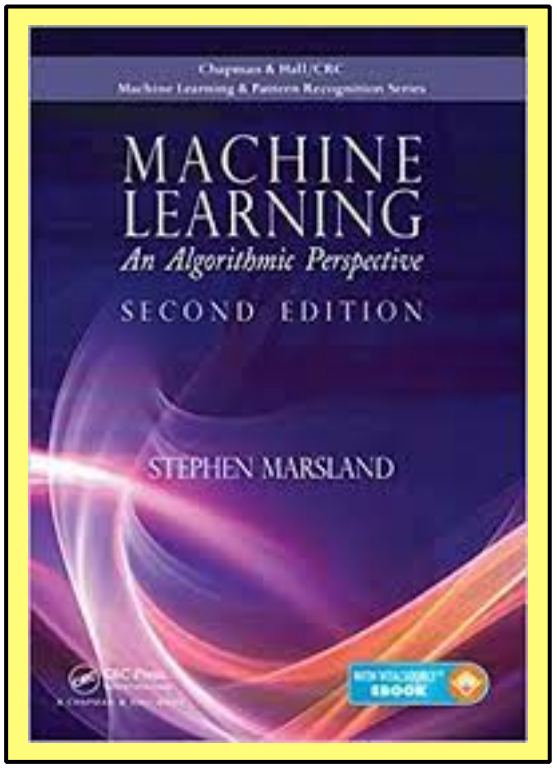
Roteiro

1. Introdução
2. k-NN
3. DWNN
4. Exercício
5. Referências

Referências

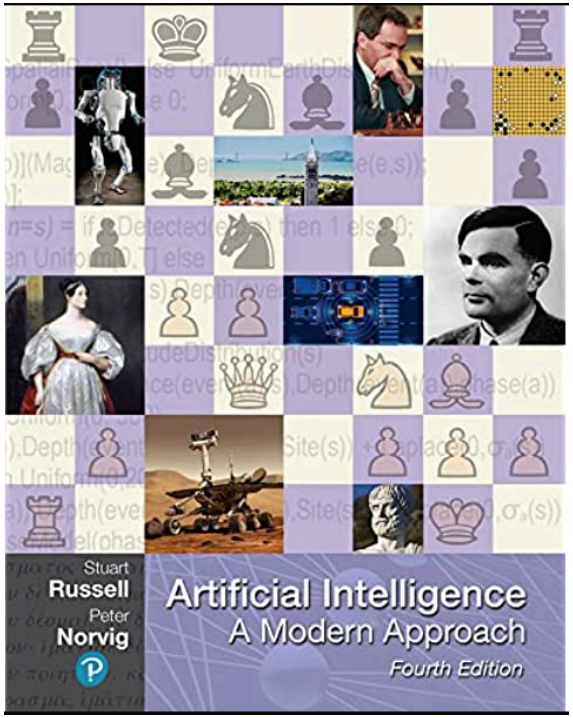


[Aggarwal, 2015]

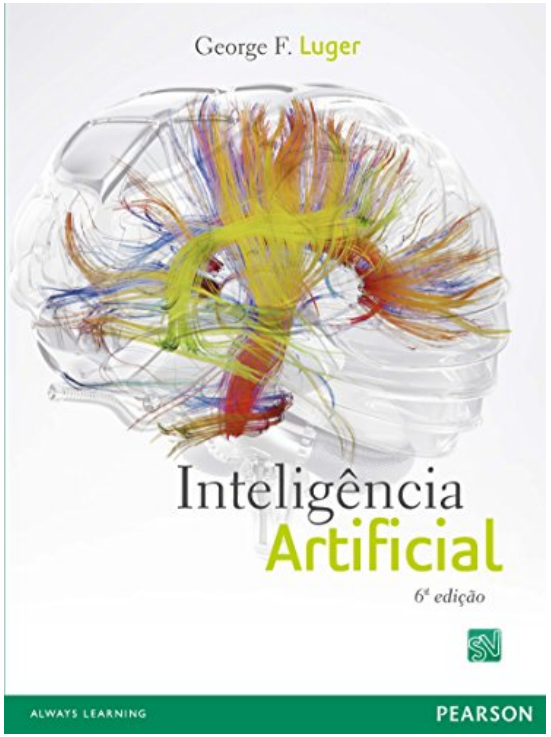


[Marsland, 2014]

Referências



[Russel & Norvig, 2021]



[Luger, 2013]

Perguntas?

Prof. Rafael G. **Mantovani**

rafaelmantovani@utfpr.edu.br