Aula 13

k-Vizinhos Mais Próximos

Rafael Geraldeli Rossi

Conteúdo

Introdução

- **O algoritmo k-Nearest Neighbors (k-NN) pertence ao paradigma de aprendizado baseado em instâncias**

- Paradigma baseado em instâncias remete ao fato de não se induzir regras, hiperplanos de separação, ou probabilidades $\rightarrow$ o aprendizado e a classificação dos exemplos **utilizam as próprias instâncias do conjunto de dados de treinamento** ou criam protótipos (objetos representantes das classes) com base nos exemplos de treinamento

- **Algoritmo amplamente utilizado na área de reconhecimento de padrões**

Introdução

- O algoritmo *k-NN* é conhecido por aprendizado do tipo "*lazy*" (preguiçoso)
  - O algoritmo de aprendizado aguarda até o último instante para construir um modelo e classificar um exemplo
  - Dado os exemplos de treinamento, o aprendizado *lazy* apenas armazena os exemplos e espera até que seja dado um exemplo de teste para realizar algum tipo de processamento
  - Classifica um exemplo baseado na semelhança (proximidade) com os exemplos de treinamento

Algoritmo: k Vizinhos Mais Próximos

- A classificação utilizando os vizinhos mais próximos, como o próprio nome diz, faz uso dos rótulos dos vizinhos para descobrir a classe de um objeto não rotulado
- No caso do k-NN são utilizados os rótulos dos k vizinhos mais próximos
- Normalmente é atribuído o rótulo da classe mais frequente dos k vizinhos mais próximos ao exemplo de teste

Algoritmo: k Vizinhos Mais Próximos

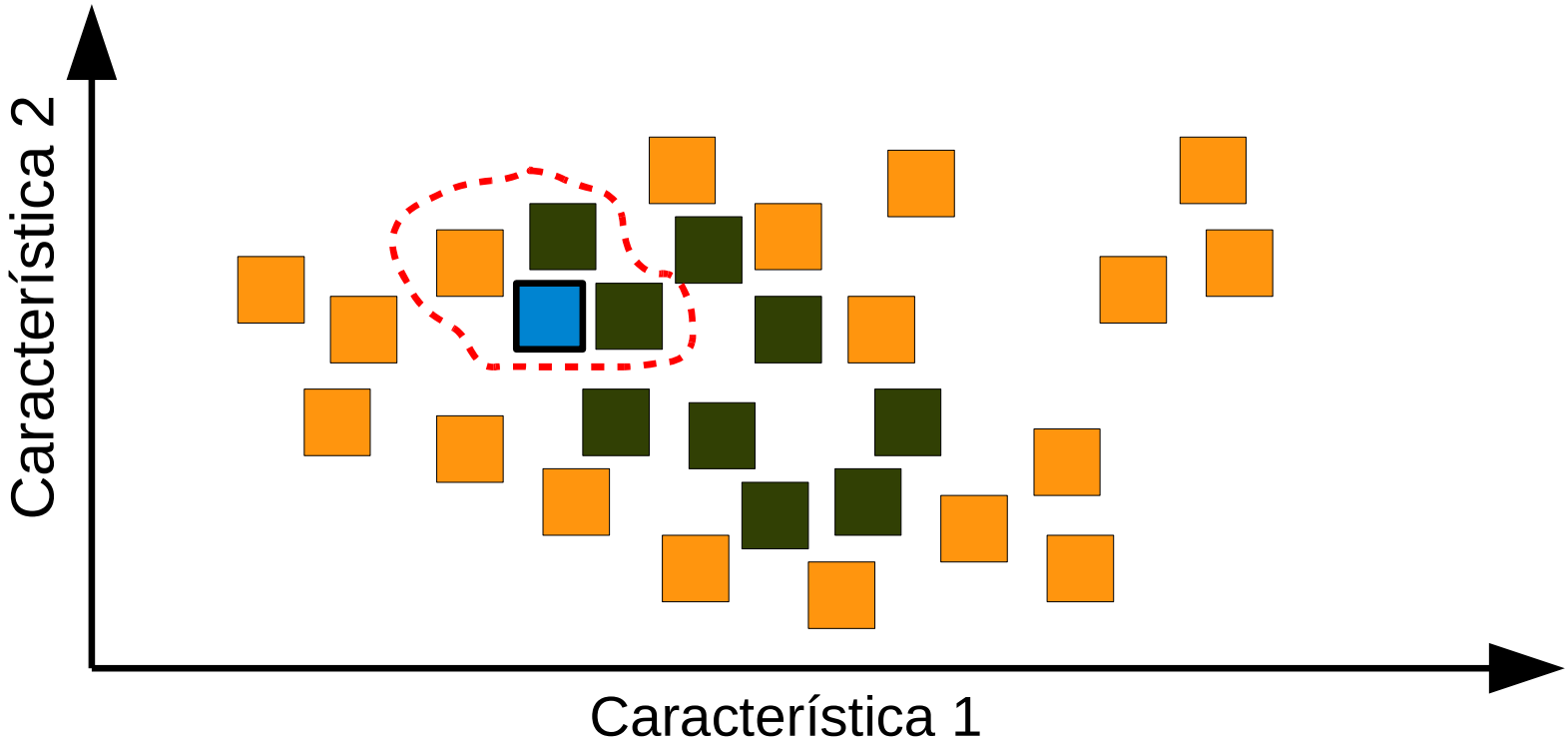


Figura: Exemplo de classificação utilizando os 3 vizinhos mais próximos

Algoritmo: k Vizinhos Mais Próximos

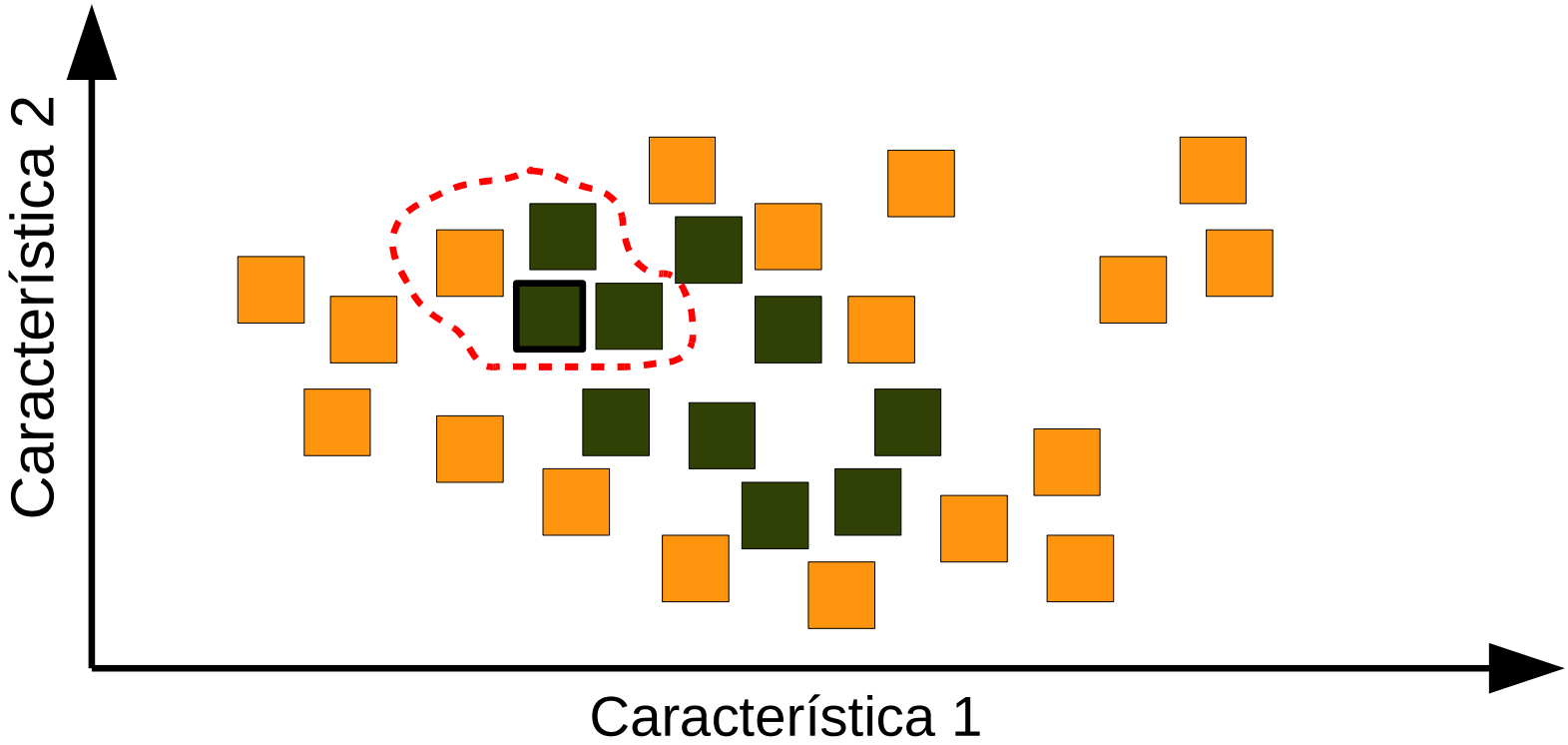


Figura: Exemplo de classificação utilizando os 3 vizinhos mais próximos

Algoritmo: k Vizinhos Mais Próximos

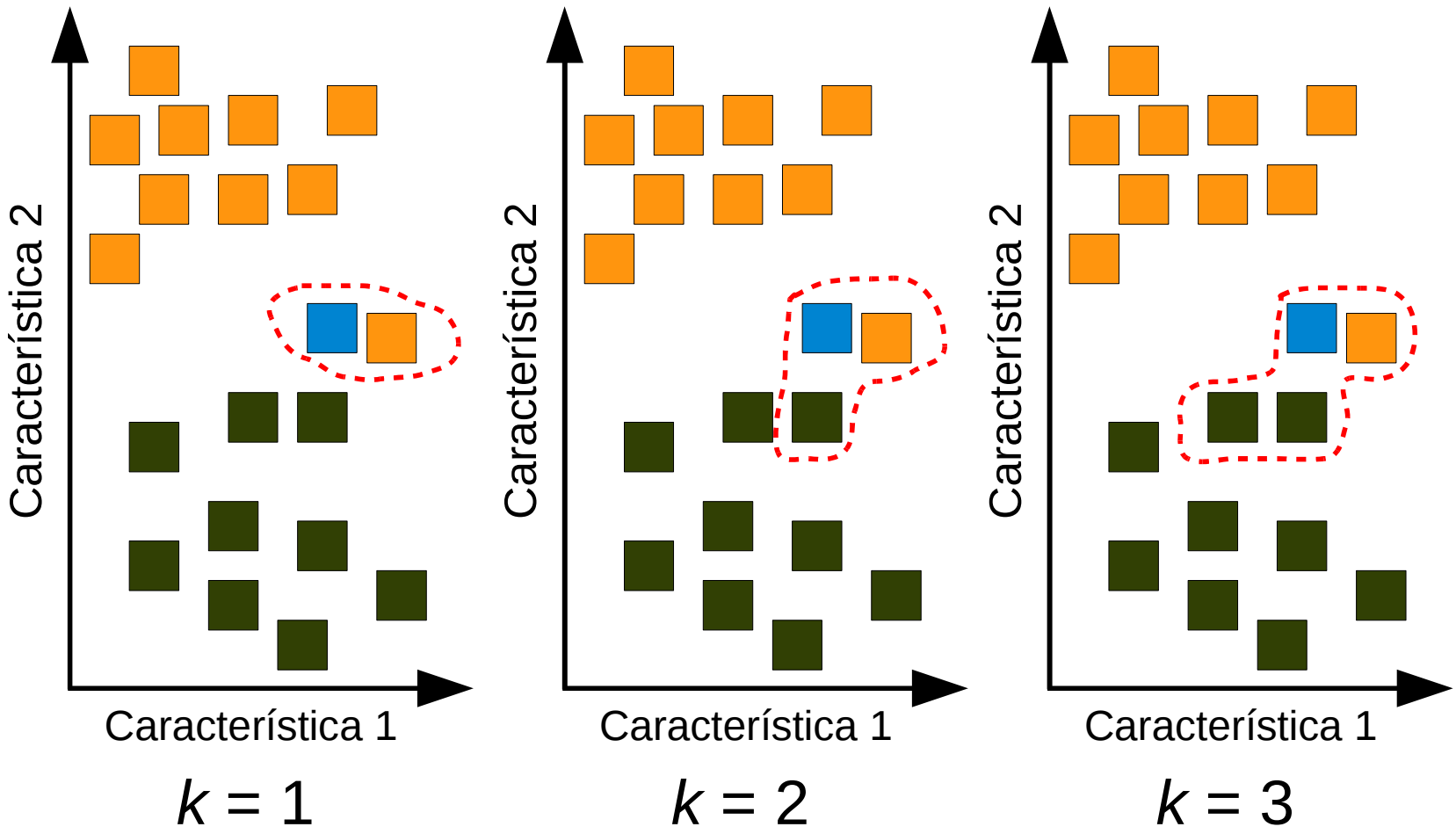


Figura: Efeito do valor de k

Algoritmo: k Vizinhos Mais Próximos

Exemplo

Tabela: Parte do conjunto de dados *Iris*

ID	Sepal Length	Sepal Width	Petal Length	Petal Width	Class
1	5,1	3,5	1,4	0,2	Iris-setosa
2	4,9	3,0	1,4	0,2	Iris-setosa
3	7,0	3,2	4,7	1,4	Iris-versicolor
4	6,4	3,2	4,5	1,5	Iris-versicolor
5	6,3	3,3	6,0	2,5	Iris-virginica
6	5,8	2,7	5,1	1,9	Iris-virginica

Sepal Length	Sepal Width	Petal Length	Petal Width	Class
5,4	3,1	2,5	1,0	**???**

Algoritmo: k Vizinhos Mais Próximos

Exemplo

$$d(t,1) = \sqrt{(5,4-5,1)^2 + (3,1-3,5)^2 + (2,5-1,4)^2 + (1,0-0,2)^2}$$
$$d(t,1) = \sqrt{(0,09 + 0,16 + 1,21 + 0,64)} = \sqrt{2,1} = 1,44$$

$$d(t,2) = \sqrt{(5,4-4,9)^2 + (3,1-3,0)^2 + (2,5-1,4)^2 + (1,0-0,2)^2}$$
$$d(t,2) = \sqrt{(0,25 + 0,01 + 1,21 + 0,64)} = \sqrt{2,11} = 1,45$$

$$d(t,3) = \sqrt{(5,4-7,0)^2 + (3,1-3,2)^2 + (2,5-4,7)^2 + (1,0-1,4)^2}$$
$$d(t,3) = \sqrt{2,56 + 0,01 + 4,84 + 0,16} = \sqrt{7,21} = 2,68$$

$$d(t,4) = \sqrt{(5,4-6,4)^2 + (3,1-3,2)^2 + (2,5-4,5)^2 + (1-1,5)^2}$$
$$d(t,4) = \sqrt{1,0 + 0,01 + 4,0 + 0,25} = \sqrt{5,26} = 2,29$$

$$d(t,5) = \sqrt{(5,4-6,3)^2 + (3,1-3,3)^2 + (2,5-6,0)^2 + (1,0-2,5)^2}$$
$$d(t,5) = \sqrt{0,81 + 0,04 + 12,25 + 2,25} = \sqrt{15,35} = 3,91$$

$$d(t,6) = \sqrt{((5,4-5,8))^2 + (3,1-2,7)^2 + (2,5-5,1)^2 + (1,0-1,9)^2}$$
$$d(t,6) = \sqrt{0,16 + 0,16 + 6,76 + 0,81} = \sqrt{7,89} = 2,80$$

Algoritmo: k Vizinhos Mais Próximos

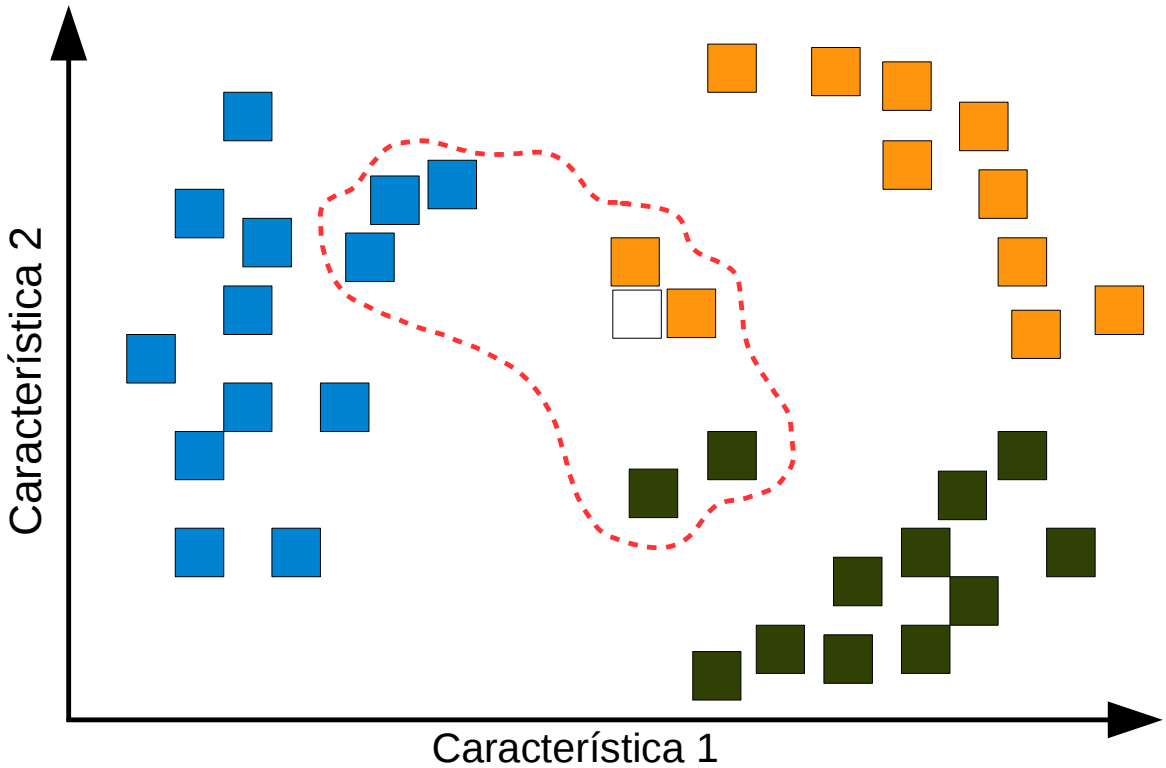
Tabela: *Ranking* dos vizinhos mais próximos

Ranking	ID	Distância	Classe
1º	1	1,44	Iris-setosa
2º	2	1,45	Iris-setosa
3º	4	2,29	Iris-versicolor
4º	3	2,68	Iris-versicolor
5º	6	2,80	Iris-virginica
6º	5	3,91	Iris-virginica

- **Resultados de classificação**
  - **1-NN**: Iris-setosa
  - **2-NN**: Iris-setosa
  - **3-NN**: Iris-setosa
  - **4-NN**: Empate entre Iris-setosa e Iris-versicolor
  - **5-NN**: Empate entre Iris-setosa e Iris-versicolor
  - **6-NN**: Empate entre Iris-setosa, Iris-versicolor e Iris-virginica

Algoritmo: k Vizinhos Mais Próximos

- Vale ressaltar que nesta versão tradicional do algoritmo *k-NN*, exemplos menos próximos tem o mesmo peso no voto de exemplos mais próximos



Algoritmo: k Vizinhos Mais Próximos

- Pode-se dar um peso diferente ao voto de cada vizinho
  - O peso do voto é dado por

$$voto = \frac{1}{dist(x, novo)}$$

    na qual $dist(x, novo)$ é a distância de um objeto x da base de treinamento ao objeto a ser classificado
  - É realizado um somatório com o peso do voto dos objetos de cada classe
  - O objeto é classificado com a classe que obteve o maior somatório de votos (considerando o peso)
  - Reduz a sensibilidade da escolha do valor de k

Algoritmo: k Vizinhos Mais Próximos

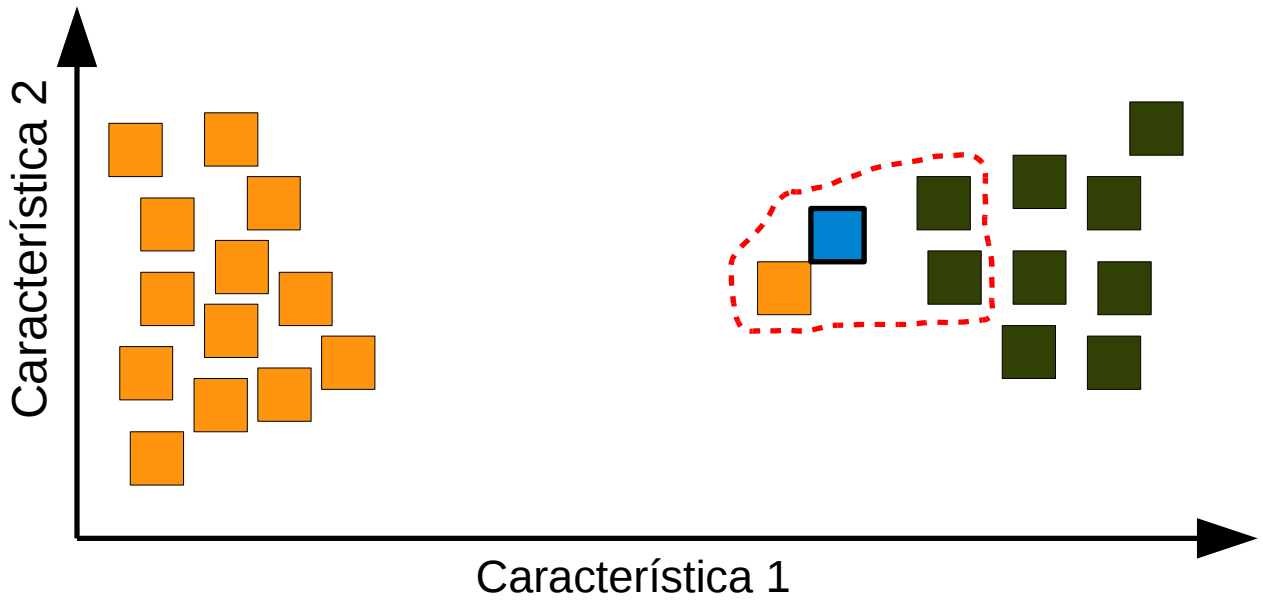
Tabela: *Ranking* dos vizinhos mais próximos

Ranking	ID	Distância	Classe
1º	1	1,44	Iris-setosa
2º	2	1,45	Iris-setosa
3º	4	2,29	Iris-versicolor
4º	3	2,68	Iris-versicolor
5º	6	2,80	Iris-virginica
6º	5	3,91	Iris-virginica

- Resultados da classificação
  - 1-NN: **Iris-setosa** $= 1/1,44 = 0,69$; Iris-versicolor $= 0$; Iris-virginica $= 0$
  - 2-NN: **Iris-setosa** $= 1/1,44 + 1/1,45 = 1,37$; Iris-versicolor $= 0$; Iris-virginica $= 0$
  - 3-NN: **Iris-setosa** $= 1/1,44 + 1/1,45 = 1,37$; Iris-versicolor $= 1/2,29 = 0,43$; Iris-virginica $= 0$
  - 4-NN: **Iris-setosa** $= 1/1,44 + 1/1,45 = 1,37$; Iris-versicolor $= 1/2,29 + 1/2,68 = 0,8$; Iris-virginica $= 0$
  - 5-NN: **Iris-setosa** $= 1/1,44 + 1/1,45 = 1,37$; Iris-versicolor $= 1/2,29 + 1/2,68 = 0,8$; Iris-virginica $= 1/2,80 = 0,35$
  - 6-NN: **Iris-setosa** $= 1/1,44 + 1/1,45 = 1,37$; Iris-versicolor $= 1/2,29 + 1/2,68 = 0,8$; Iris-virginica $= 1/2,80 + 1/3,91 = 0,60$

Algoritmo: k Vizinhos Mais Próximos

- O uso de peso nos votos pode gerar erros devido a presença de *outliers* ou a classificação pode ser baseada em um único vizinho
- Um objeto pode estar tão próximo a um *outlier* (ou vizinho) de modo que o peso os votos de outros objetos não sejam o suficiente para definir a classe de um objeto



Padronização dos Dados

- Normalmente os valores dos atributos são normalizados ou padronizados para que um atributo não interfira excessivamente na (dis)similaridade dos exemplos
- Nos exemplos anteriores, o tamanho da pétala e da sépala de uma íris possuem o mesmo intervalo de valores
- Porém, ao considerar atributos com diferentes intervalos de valores, é necessário ter cuidado na hora de calcular a (dis)similaridade dos exemplos

Ex:

Tabela: Conjunto de dados original

ID	Idade	Salário	Classe
1	34	3000	Sim
2	36	3200	Sim
3	65	2700	Não
4	67	2600	Não

Padronização dos Dados

- Agora vamos supor que queiramos descobrir o exemplo mais próximo do conjunto de treinamento, utilizando a distância Euclidiana, para o segundo exemplo:

Tabela: Nova instância

Idade	**Salário**	**Classe**
35	2800	???

- Resultado do cálculo da distância Euclidiana entre os exemplos:

Tabela: *Ranking* dos vizinhos mais próximos

Ranking	**ID**	**Distância**	**Classe**
1º	3	104,4000	Não
2º	1	200,0024	Sim
3º	4	202,5400	Não
4º	2	400,0012	Sim

Padronização dos Dados

- No exemplo anterior pode-se perceber que a dissimilaridade é praticamente definida apenas pelo atributo salário
- Para fazer com que os atributos possuam os mesmos pesos e influenciem o cálculo da similaridade igualitariamente, deve-se padronizar os valores dos atributos
- Existem diversas formas de padronização dos dados
- Uma das mais utilizadas é a normalização linear ou normalização *min-max* $\rightarrow$ que mapeia o intervalo de dados original para o intervalo $[0, 1]$

Padronização dos Dados

- Na normalização *min-max*, o novo valor (v_{novo}) em um determinado atributo é dado por:

$$v_{novo} = \frac{v_{original} - v_{minimo}}{v_{maximo} - v_{minimo}}$$

na qual

- $v_{original}$ é o valor original do atributo
- v_{minimo} é o menor valor do atributo sendo normalizado
- v_{maximo} é o maior valor do atributo sendo normalizado

Padronização dos Dados

- Utilizando a normalização, o conjunto de dados apresentado anteriormente fica da seguinte forma:

Tabela: Conjunto de dados original

ID	Idade	Salário	Classe
1	34	3000	Sim
2	36	3200	Sim
3	65	2700	Não
4	67	2600	Não

Tabela: Conjunto de dados padronizado

ID	Idade	Salário	Classe
1	0,00	0,66	Sim
2	0,06	1,00	Sim
3	0,93	0,16	Não
4	1,00	0,00	Não

Padronização dos Dados

- Com os dados padronizados, o cálculo da dissimilaridade utilizando a medida Euclidiana fica da seguinte forma:

Tabela: Nova instância

Idade	**Salário**	**Classe**
35	2800	???

Tabela: *Ranking*

Ranking	**ID**	**Distância**	**Classe**
1º	1	0,4177	Sim
2º	2	0,7506	Sim
3º	3	0,9132	Não
4º	4	1,0016	Não

- **OBSERVAÇÃO:** pode-se ponderar a importância do peso de cada atributo no cálculo da distância Euclidiana caso o usuário queira dar mais peso a um determinado atributo

k-NN para Predição Numérica

- **O k-NN também pode ser utilizado para predição numérica, na qual o valor retornado é a média dos valores dos k vizinhos**
  - Calcular a média do atributo alvo utilizando o valor do mesmo atributo dos k-vizinhos mais próximos
- Exemplo: calcular o salário do exemplo $\{Idade = 31; Tempo\ de\ Serviço = 13\}$ utilizando 3 vizinhos mais próximos e o seguinte conjunto de treinamento

Tabela: Conjunto de dados original

ID	Idade	Tempo de Serviço	Salário
1	20	2	2000
2	25	3	2500
3	50	25	8000
4	30	10	5000
5	27	5	3000
6	33	10	2700

Tabela: Conjunto de dados padronizado para o cálculo das distâncias

ID	Idade	Tempo de Serviço	Salário
1	0,00	0,00	2000
2	0,17	0,04	2500
3	1,00	1,00	8000
4	0,33	0,35	5000
5	0,23	0,13	3000
6	0,43	0,35	2700

k-NN para Predição Numérica

- Exemplo de teste padronizado:
  $\{Idade = 0,37; Tempo\ de\ Serviço = 0,48\}$

Tabela: *Ranking* dos vizinhos mais próximos

Ranking	**ID**	**Distância**	**Salário**
1º	4	0,1	5000
2º	6	0,1	2700
3º	2	0,47	2500
4º	1	0,6	2000
5º	3	0,66	8000
6º	5	0,7	3000

- **Salário do exemplo de teste:**

$$Salário = \frac{5000 + 2700 + 2500}{3} = 3400,00$$

k-NN para Predição Numérica

- O mesmo procedimento pode ser utilizado para a imputação de valores ausentes
  - Deve-se desconsiderar o atributo que possui valor ausente no cálculo das distâncias

Tabela: Conjunto de dados original

ID	Idade	Tempo de Serviço	Salário
1	20	2	2000
2	25	–	2500
3	50	25	8000
4	30	10	5000
5	27	5	3000
6	33	10	2700

Tabela: Conjunto de dados padronizados

ID	Idade	Tempo de Serviço	Salário
1	0,00	2	0,00
2	0,17	–	0,08
3	1,00	25	1,00
4	0,33	10	0,50
5	0,23	5	0,17
6	0,43	10	0,12

Algoritmo: k Vizinhos Mais Próximos

Tabela: *Ranking*

Ranking	ID	Distância	Tempo de Serviço
1º	5	0,1	5
2º	1	0,14	2
3º	6	0,24	10
4º	4	0,43	10
5º	3	1,23	25

- Utilizando 2 vizinho mais próximos temos

$$\textit{Tempo de Serviço} = \frac{5+2}{2} = 3,5$$

Características e Consdiderações sobre o *k-NN*

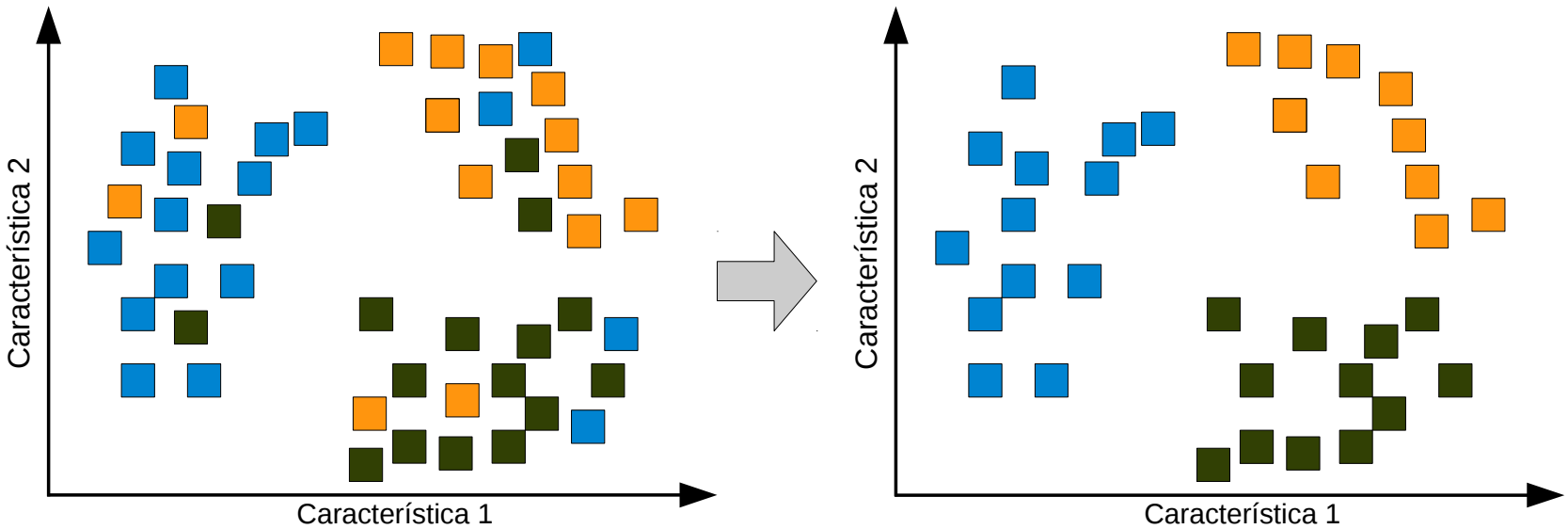
- O valor de k é determinado experimentalmente
- A cada valor de k é realizada uma avaliação em um conjunto de teste
- É escolhido o valor de k com melhor desempenho no conjunto de teste
- Em geral
  - Valor de k pequeno
    - Função de discriminação entre classes é muito flexível
    - Sensível a ruído
  - Valor de k grande
    - Função de discriminação entre classes é menos flexível
    - Tende a incluir objetos de outras classes
    - Menos sensível a ruído

Características e Consdideraçẽos sobre o *k-NN*

- A escolha da métrica de distância é fundamental
- Seja $|D|$ o número de exemplos de treinamento e $|A|$ o número de atributos, a complexidade do k-NN é $O(|D| \times |A|)$
- Técnica para acelerar a classificação
  - Implementações paralelas
  - Calculo da distância baseada em um subconjunto de atributos
  - kD-tree
  - Remover exemplos de treinamento que são inconsistentes com seus próprios vizinhos
  - ...

Características e Consdideraçẽos sobre o *k-NN*

- Técnica para acelerar a classificação
  - Remover exemplos de treinamento que são inconsistentes com seus próprios vizinhos $\rightarrow$ seleção de instâncias e deteção de outliers
    - Se os vizinhos de um exemplo de treinamento possuirem a classe diferente de um determinado exemplo de treinamento, esse exemplo deve ser removido



Características e Consdiderações sobre o *k-NN*

- Ao realizar um aprendizado do tipo *lazy*, há um menor esforço na etapa de aprendizado e um maior esforço na etapa de classificação
- Requer técnicas eficientes de armazenamento e recuperação
- Naturalmente suportam aprendizado incremental
- O *k-NN* é não paramêtrico $\rightarrow$ não assume qualquer distribuição a respeito dos dados
- São capazes de modelar espaços de decisões complexos
- Veja: `http://vision.stanford.edu/teaching/cs231n-demos/knn/`

Material Complementar

- K-nearest neighbors algorithm

  `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`

- Aprendizagem de máquina com o KNN

  `https://www.youtube.com/watch?v=_3uA9tGBx0s`

- Feature scaling

  `https://en.wikipedia.org/wiki/Feature_scaling`

Material Complementar

- Develop k-Nearest Neighbors in Python From Scratch

  `https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/`

- Nearest Neighbors Classification

  `https://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py`

Material Complementar

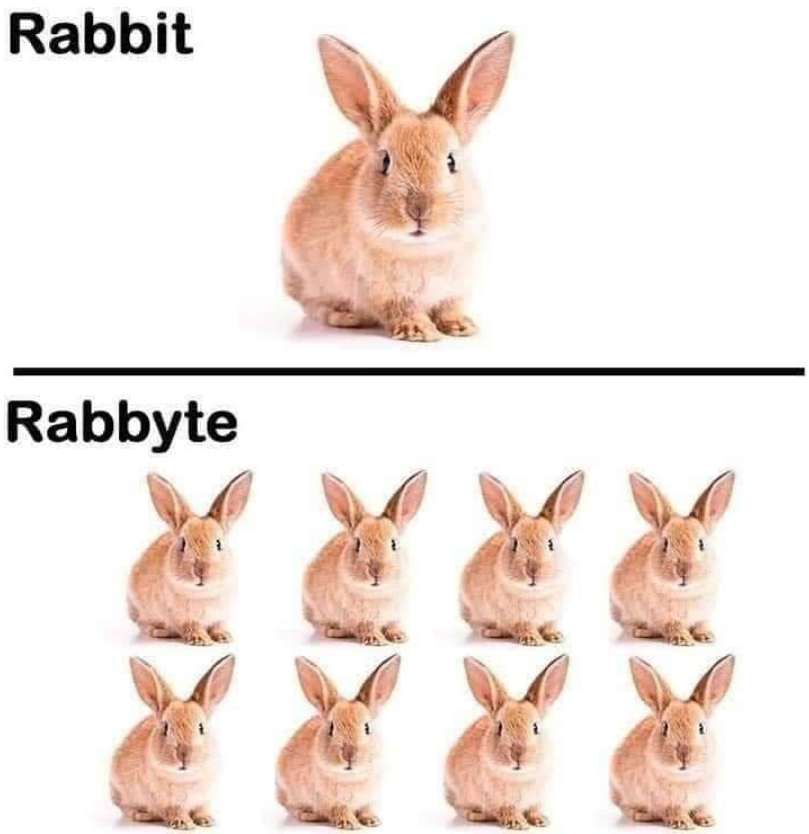
- Preprocessing With Sklearn a complete and comprehensive guide

  `https://towardsdatascience.com/preprocessing-with-sklearn-a-complete-and-comprehensive-guide-670cb98fcfb`

- Scale, Standardize, or Normalize with Scikit-Learn

  `https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02`

Imagem do Dia



Inteligência Artificial
http://lives.ufms.br/moodle/

Rafael Geraldeli Rossi
rafael.g.rossi@ufms.br

Slides baseados em [Han et al., 2011]

Referências Bibliográficas I

Han, J., Kamber, M., and Pei, J. (2011).
Data Mining: Concepts and Techniques.
The Morgan Kaufmann Series in Data Management Systems.
Elsevier.