**CENTRO UNIVERSITÁRIO UNINORTE**
**CURSO DE PÓS-GRADUAÇÃO EM:** Pós Graduação em Gerência de Banco de Dados.
**DISCIPLINA: Mineração de Dados**

# Classificação

Prof.º: Manoel Limeira
juniorlimeiras@gmail.com

# Agenda

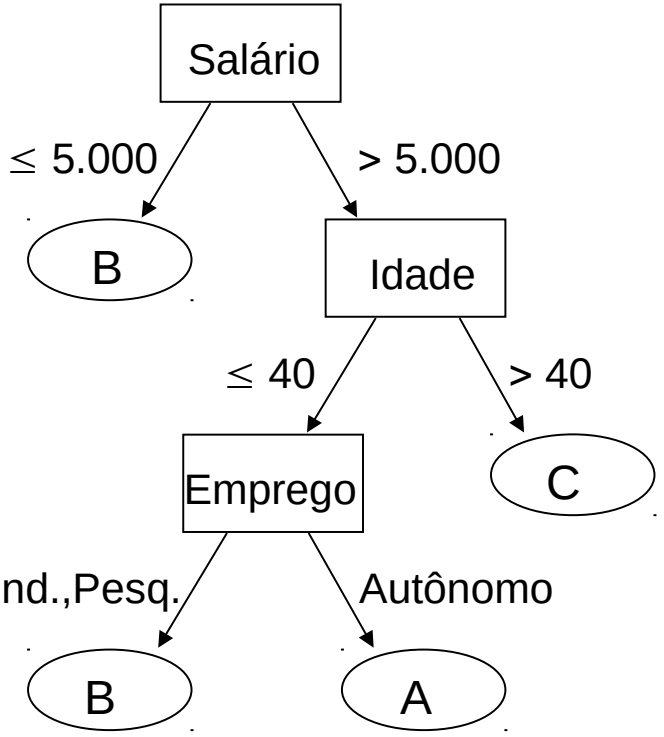
- Algoritmos
  - Árvore de Decisão
  - Naive Bayes
  - KNN
- Avaliação de Classificadores

# Modelo de classificação com Árvore de Decisão

Atributos Independentes — Atributo Dependente

| ID | Salário | Idade | Emprego | Classe |
|---|---|---|---|---|
| 1 | 3.000 | 30 | Autônomo | B |
| 2 | 4.000 | 35 | Indústria | B |
| 3 | 7.000 | 50 | Pesquisa | C |
| 4 | 6.000 | 45 | Autônomo | C |
| 5 | 7.000 | 30 | Pesquisa | B |
| 6 | 6.000 | 35 | Indústria | B |
| 7 | 6.000 | 35 | Autônomo | A |
| 8 | 7.000 | 30 | Autônomo | A |
| 9 | 4.000 | 45 | Indústria | B |

**Árvore de Decisão ou Árvore de Classificação**



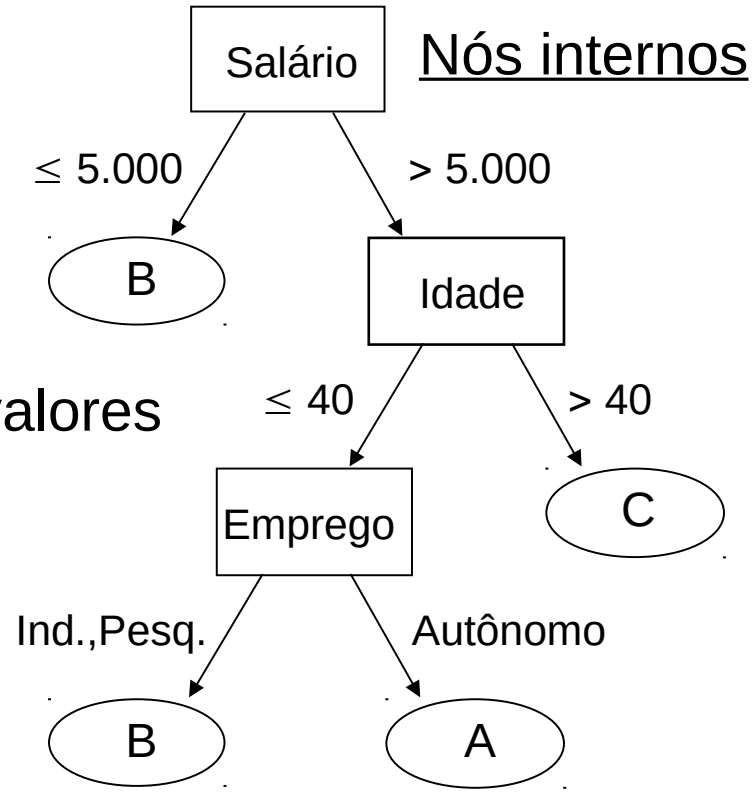

A partir de uma base de treinamento, extrai-se o modelo de classificação (árvore de decisão)

# Árvores de Decisão

- A sua representação é **intuitiva** e torna o modelo de classificação resultante **fácil** de se utilizar e de ser **entendido**
- A precisão de suas previsões, em geral, possui **taxas de acertos competitivas** em relação a de outros modelos
- Algoritmos **rápidos** e escaláveis podem ser implementados para a construção de árvores de decisão, considerando-se grandes bases de treinamento

# Árvores de Decisão

Primeiro Nó: raiz da árvore




Nós internos: Atributos/Teste

Arestas: predicados/valores

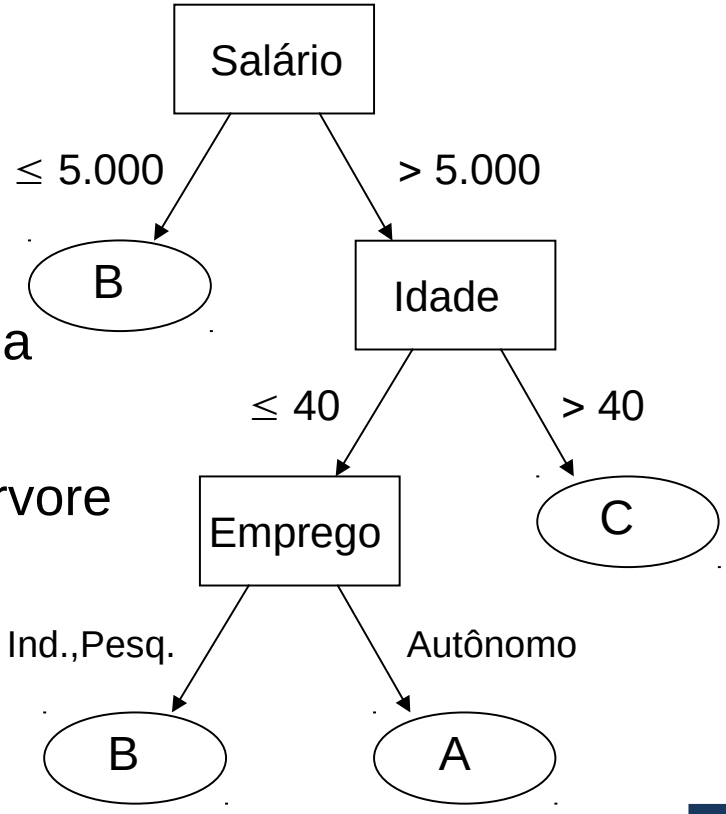
Folhas: valores de classes

Nó interno + Aresta = condição

# Árvores de Decisão ⇒ Regras de Classificação

- Cada caminho da raiz até a folha representa uma regra, definida como a conjunção das condições percorridas, implicando no valor da classe encontrada na folha
- A árvore deve ser definida de forma que, para um mesmo registro, haja um e apenas um caminho da raiz até a folha

**Árvore de Decisão ou Árvore de Classificação**




Regras de classificação obtidas a partir da árvore de decisão:

(Sal ≤ 5k) ⇒ Classe = B
(Sal > 5k) ^ (Idade > 40) ⇒ Classe = C
(Sal > 5k) ^ (Idade ≤ 40) ^ (Emprego = Autônomo) ⇒ Classe = A
(Sal > 5k) ^ (Idade ≤ 40) ^ ((Emprego = Indústria) ˅ (Emprego = Pesquisa)) ⇒ Classe = B

# Algoritmo ID3

- Utilizado para construir árvores de decisão
- **Entrada**
  - base de treinamento que contém os registros
  - lista dos atributos independentes
  - definição do atributo dependente (alvo ou objetivo)
- **Saída**
  - árvore de decisão que permite classificar (definir o valor do atributo dependente) um novo registro a partir de seus atributos independentes

# Algoritmo ID3 – Como Escolher o Atributo?

- O algoritmo ID3 utiliza uma medida conhecida como **Ganho de Informação** que se baseia no conceito de entropia
- Trata-se de uma heurística para selecionar o atributo, tentando minimizar o número de testes necessários para classificar os registros das partições resultantes
- Mede quão bem um determinado atributo separa os registros de treino de acordo com o valor da classe
- O atributo com maior Ganho de Informação é escolhido como **atributo teste** para o nó corrente

# Medida de Ganho de Informação

- **Entropia**: medida da quantidade de “desordem” de um conjunto de registros (representa a quantidade de informação necessária para classificar um registro)
- Ganho(Atr): redução da entropia escolhendo-se Atr
- Ganho(Atr) = E(S) - E(S,Atr),
- Onde:
  - E(S): entropia de uma partição S da base
  - E(S,Atr): entropia, considerando-se o particionamento de S de acordo com os valores do atributo Atr

# Entropia

- Dado um conjunto **S**, contendo **s** registros que pertencem a **m** classes, a entropia de **S** é calculada da seguinte maneira:

$$E(S) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- onde:
- $\mathbf{p_i}$ é a proporção de registros de **S** pertencente a  i-ésima classe, i = 1, 2, ..., m

# Entropia E(S, Atr)

- A entropia de **S**, considerando-se o seu particionamento de acordo com os valores do atributo **Atr**, é calculada da seguinte maneira:

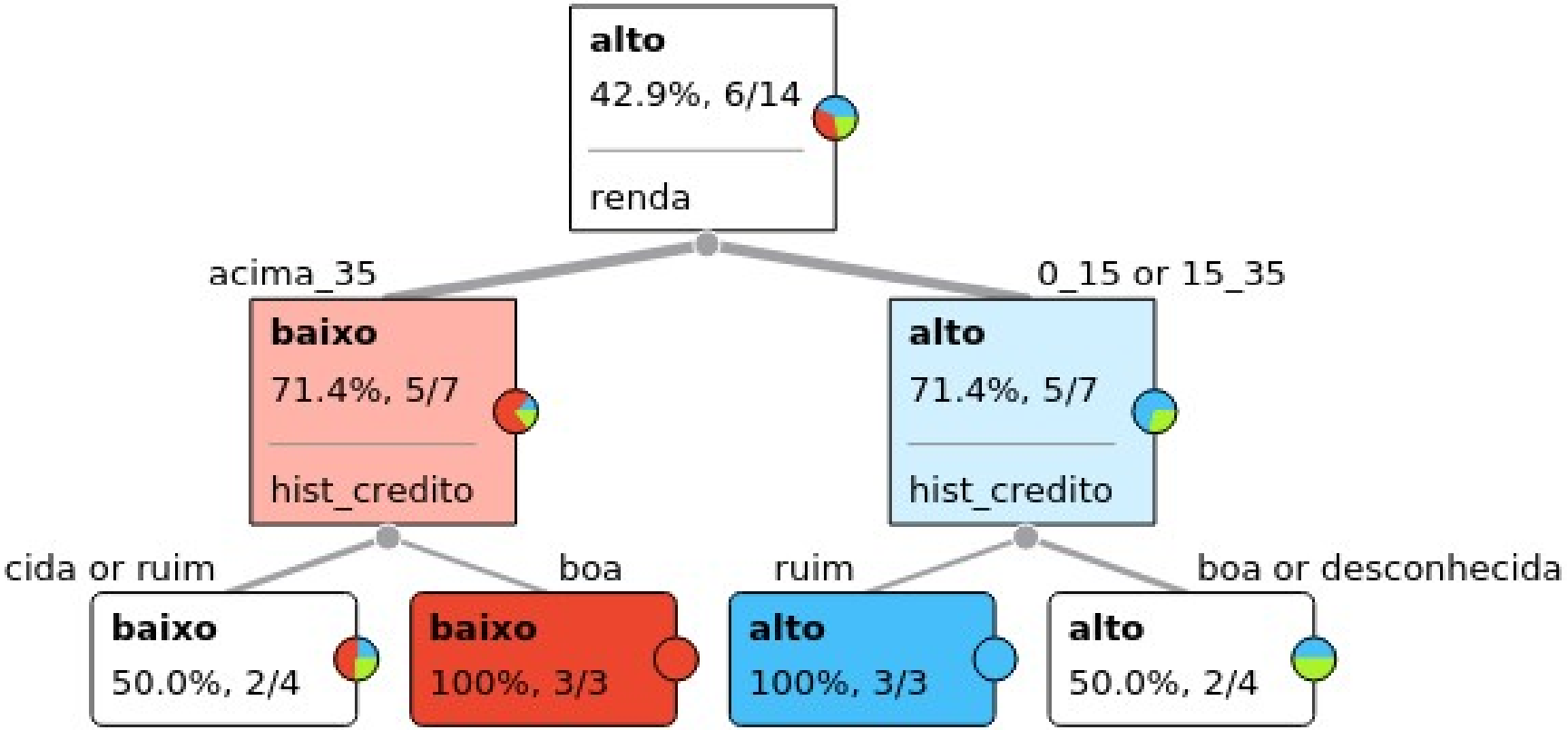
$$E(S, Atr) = \sum_{v \in Domínio(Atr)} \frac{|S_v|}{|S|} E(S_v)$$

- onde:
- $\mathbf{S_v}$ é a partição de **S** que contém o valor **v** em **Atr**

# Exemplos: atributos nominais

| # | História de crédito | Dívida | Garantia | Renda anual | Risco |
|---|---|---|---|---|---|
| 1 | Ruim | Alta | Nenhuma | < 15.000 | Alto |
| 2 | Desconhecida | Alta | Nenhuma | >= 15.000 e <=35.000 | Alto |
| 3 | Desconhecida | Baixa | Nenhuma | >= 15.000 e <=35.000 | Moderado |
| 4 | Desconhecida | Baixa | Nenhuma | > 35.000 | Alto |
| 5 | Desconhecida | Baixa | Nenhuma | > 35.000 | Baixo |
| 6 | Desconhecida | Baixa | Adequada | > 35.000 | Baixo |
| 7 | Ruim | Baixa | Nenhuma | < 15.000 | Alto |
| 8 | Ruim | Baixa | Adequada | > 35.000 | Moderado |
| 9 | Boa | Baixa | Nenhuma | > 35.000 | Baixo |
| 10 | Boa | Alta | Adequada | > 35.000 | Baixo |
| 11 | Boa | Alta | Nenhuma | < 15.000 | Alto |
| 12 | Boa | Alta | Nenhuma | >= 15.000 e <=35.000 | Moderado |
| 13 | Boa | Alta | Nenhuma | > 35.000 | Baixo |
| 14 | Ruim | Alta | Nenhuma | >= 15.000 e <= 35.000 | Alto |

# Árvore de Decisão



# Classificação Bayesiana

- Classificadores Bayesianos são classificadores estatísticos, que se baseiam no Teorema de Bayes
- Trabalham com a ideia de calcular a **probabilidade** de uma instância de entrada pertencer a cada uma das classes
- **Naive Bayes** é o mais popular classificador Bayesiano e apresenta resultados competitivos em termos de acurácia e tempo de processamento
- Classificador Naive Bayes considera a **"independência condicional"** em relação à classe, ou seja, o efeito do valor de um atributo sobre a classe é independente dos valores dos demais atributos

# Classificação Bayesiana

- O classificador Naive Bayes calcula a probabilidade posterior $P(C_i/\mathbf{X})$ – probabilidade de **X** ser da classe $C_i$ considerando os valores dos atributos de **X** – para cada classe $C_i$
- O classificador decide que **X** é da classe $C_i$, se e somente se $P(C_i/\mathbf{X})$ for maior do que $P(C_j/\mathbf{X})$ para qualquer outra classe $C_j$, ou seja, **X** é da classe $C_{i,}$ sse:

$$P(C_i/\mathbf{X}) > P(C_j/\mathbf{X}) \text{ para todo } 1 \leq j \leq m, j \neq i$$

# Teorema de Bayes

- A probabilidade posterior P($C_i$/**X**) será calculada a partir do Teorema de Bayes:

$$P(Ci/X) = \frac{P(X/Ci) \cdot P(Ci)}{P(X)}$$

- Como P(**X**) é constante para todas as classes, basta maximizar/comparar:

$$P(Ci/X) \approx P(X/Ci) \cdot P(Ci)$$

# Classificador Naive Bayes

- Basta então calcular $P(X/C_i)$ e $P(C_i)$
- $P(C_i)$ é a probabilidade anterior da classe $C_i$, estimada por:

$$P(Ci/X) = \frac{|Ci, D|}{|D|}$$

- onde $|C_i, \mathbf{D}|$ é o número de tuplas da classe *i* em **D** e $|\mathbf{D}|$ é o número de tuplas de **D**

# Classificador Naive Bayes

- Considerando a independência condicional, P(**X**/$C_i$) é calculado da seguinte forma:

$$P(X/Ci) = \prod_{k=1}^{n} P(xk/Ci)$$

$$P(X/Ci) = P(x1/Ci) \times P(x2/Ci) \times \ldots \times P(xn/Ci)$$

- onde para cada *k*, P(*xk*/$C_i$) pode ser estimado a partir da base de treinamento como apresentado a seguir

# Classificador Naive Bayes

- Resumindo: para predizer a classe de **X**, o classificador Bayesiano vai calcular $P(\mathbf{X}/C_i).P(C_i)$ para cada classe $C_i$ e vai associar **X** a $C_i$, se e somente se:

$P(\mathbf{X}/C_i).P(C_i) > P(\mathbf{X}/C_j).P(C_j)$, para todo $1 \leq j \leq m, j \neq i$.

# Exemplos: atributos nominais

| # | História de crédito | Dívida | Garantia | Renda anual | Risco |
|---|---|---|---|---|---|
| 1 | Ruim | Alta | Nenhuma | < 15.000 | Alto |
| 2 | Desconhecida | Alta | Nenhuma | >= 15.000 e <=35.000 | Alto |
| 3 | Desconhecida | Baixa | Nenhuma | >= 15.000 e <=35.000 | Moderado |
| 4 | Desconhecida | Baixa | Nenhuma | > 35.000 | Alto |
| 5 | Desconhecida | Baixa | Nenhuma | > 35.000 | Baixo |
| 6 | Desconhecida | Baixa | Adequada | > 35.000 | Baixo |
| 7 | Ruim | Baixa | Nenhuma | < 15.000 | Alto |
| 8 | Ruim | Baixa | Adequada | > 35.000 | Moderado |
| 9 | Boa | Baixa | Nenhuma | > 35.000 | Baixo |
| 10 | Boa | Alta | Adequada | > 35.000 | Baixo |
| 11 | Boa | Alta | Nenhuma | < 15.000 | Alto |
| 12 | Boa | Alta | Nenhuma | >= 15.000 e <=35.000 | Moderado |
| 13 | Boa | Alta | Nenhuma | > 35.000 | Baixo |
| 14 | Ruim | Alta | Nenhuma | >= 15.000 e <= 35.000 | Alto |

# Exemplo de Classificação

| Risco de Crédito | História de Crédito | | | Dívida | | Garantia | | Renda anual | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Boa | Desconhecida | Ruim | Alta | Baixa | Nenhuma | Adequada | ‘<15‘ | ‘15<=x<=35‘ | ‘>35‘ |
| | 5 | 5 | 4 | 7 | 7 | 11 | 3 | 3 | 4 | 7 |
| **Alto** 6/14 | 1/6 | 2/6 | 3/6 | 4/6 | 2/6 | 6/6 | 0 | 3/6 | 2/6 | 1/6 |
| **Moderado** 3/14 | 1/3 | 1/3 | 1/3 | 1/3 | 2/3 | 2/3 | 1/3 | 0 | 2/3 | 1/3 |
| **Baixo** 5/14 | 3/5 | 2/5 | 0 | 2/5 | 3/5 | 3/5 | 2/5 | 0 | 0 | 5/5 |

**Instância de Teste**
História=Boa
Dívida=Alta
Garantia=Nenhuma
Renda=’>35’

P(Alto) = 6/14 * 1/6 * 4/6 * 6/6 * 1/6 = 0,0079
P(Moderado) = 3/14 * 1/3 * 1/3 * 2/3 * 1/3 = 0,0052
P(Baixo) = 5/14 * 3/5 * 2/5 * 3/5 * 5/5 = **0,0514**

P(Alto) = 0,0079 / 0,0645 * 100 = 12,24%
P(Moderado) = 0,0052 / 0,0645 *100 = 8,06%
P(Baixo) = 0,0514 / 0,0645 * 100 = **79,68%**

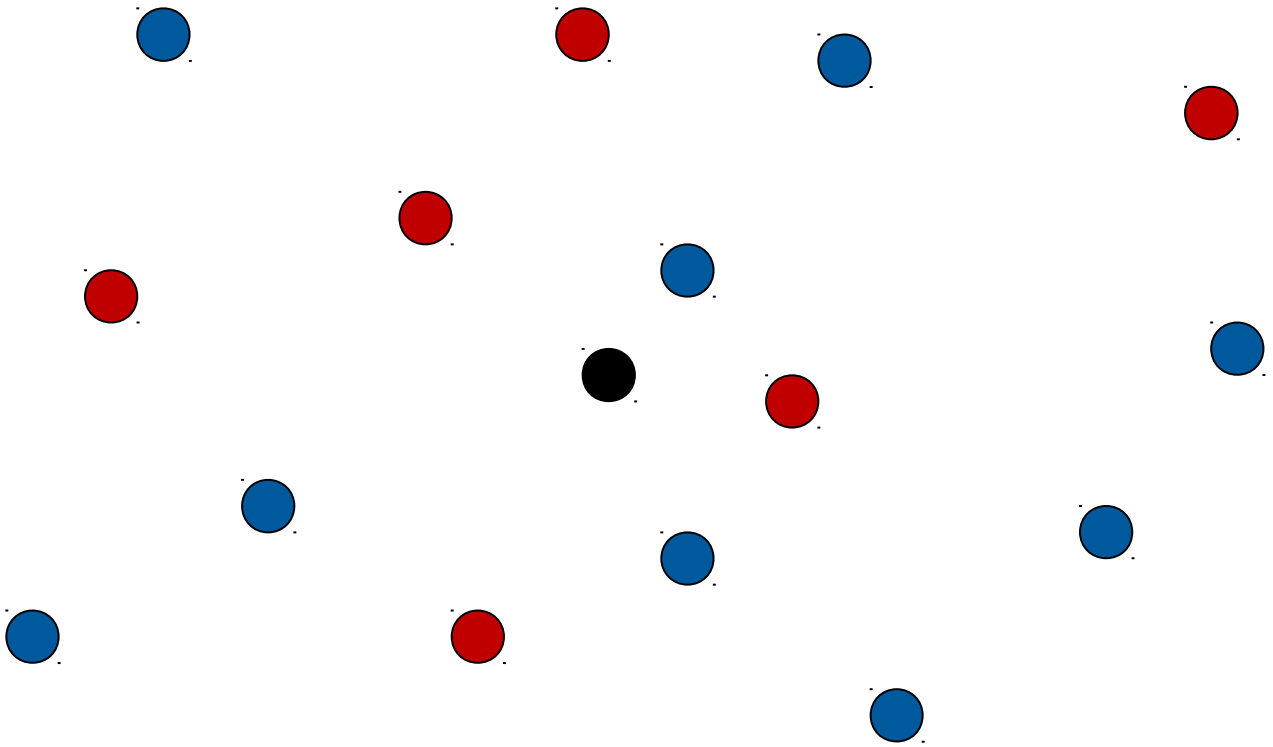
Soma
0,0079
+0,0052
+0,0514
**0,0645**

# Classificadores k-NN

- Classificadores k-NN (k-*Nearest Neighbor*) se baseiam na ideia de **aprendizagem por analogia**, ou seja, a classe de uma tupla de entrada será determinada pelo conhecimento das classes de tuplas similares da base de treinamento
- Cada tupla possui **n** atributos e, portanto, pode ser caracterizada por um ponto em um espaço n-dimensional
- A técnica procura pelas **k** tuplas de treinamento mais próximas à tupla a ser classificada no espaço n-dimensional. Essas tuplas serão os **k** vizinhos mais próximos

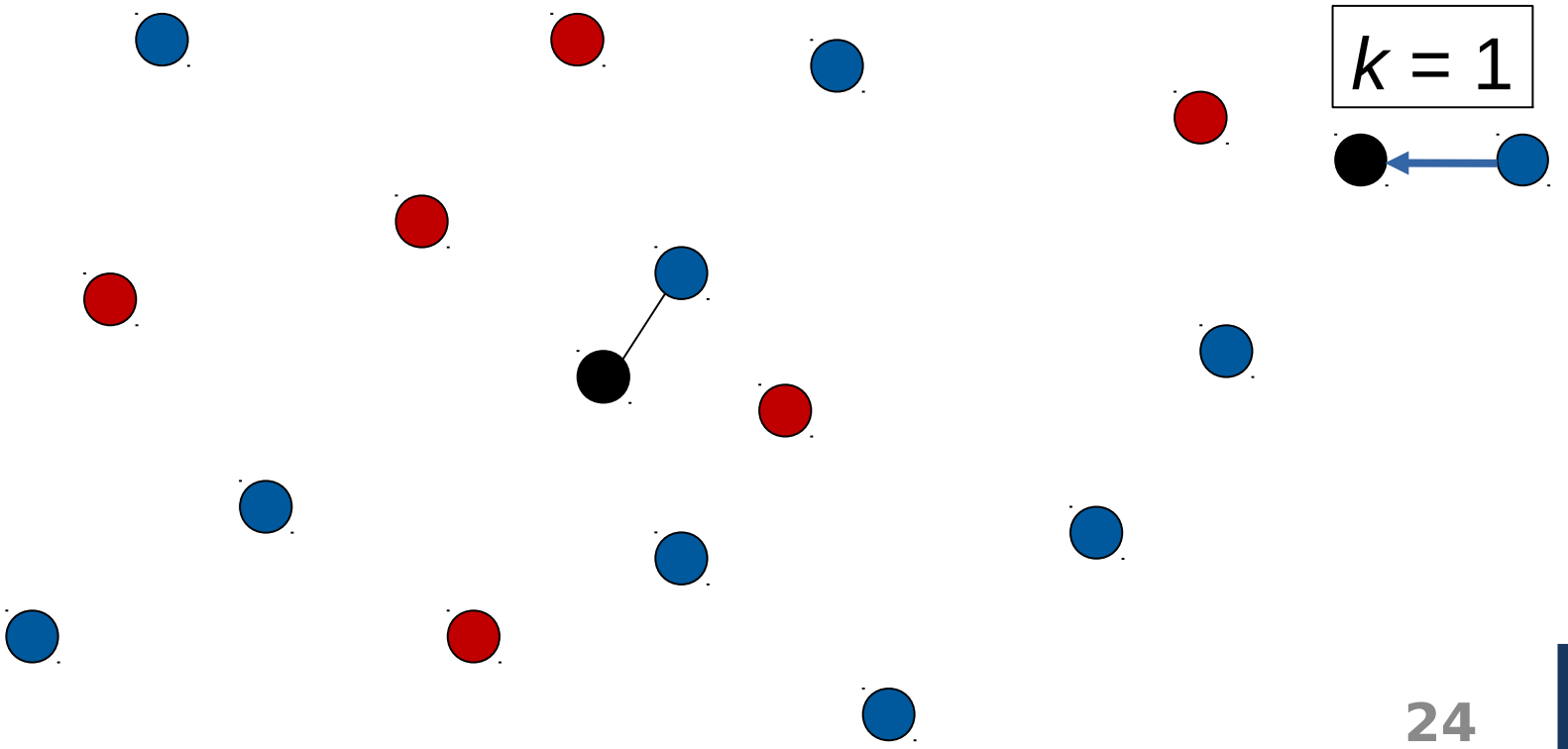
# Classificadores k-NN: o valor de k

- Depois de identificados os **k** vizinhos mais próximos da tupla ***t*** de entrada a ser classificada, o k-NN atribui a t a classe predominante entre esses **k** vizinhos



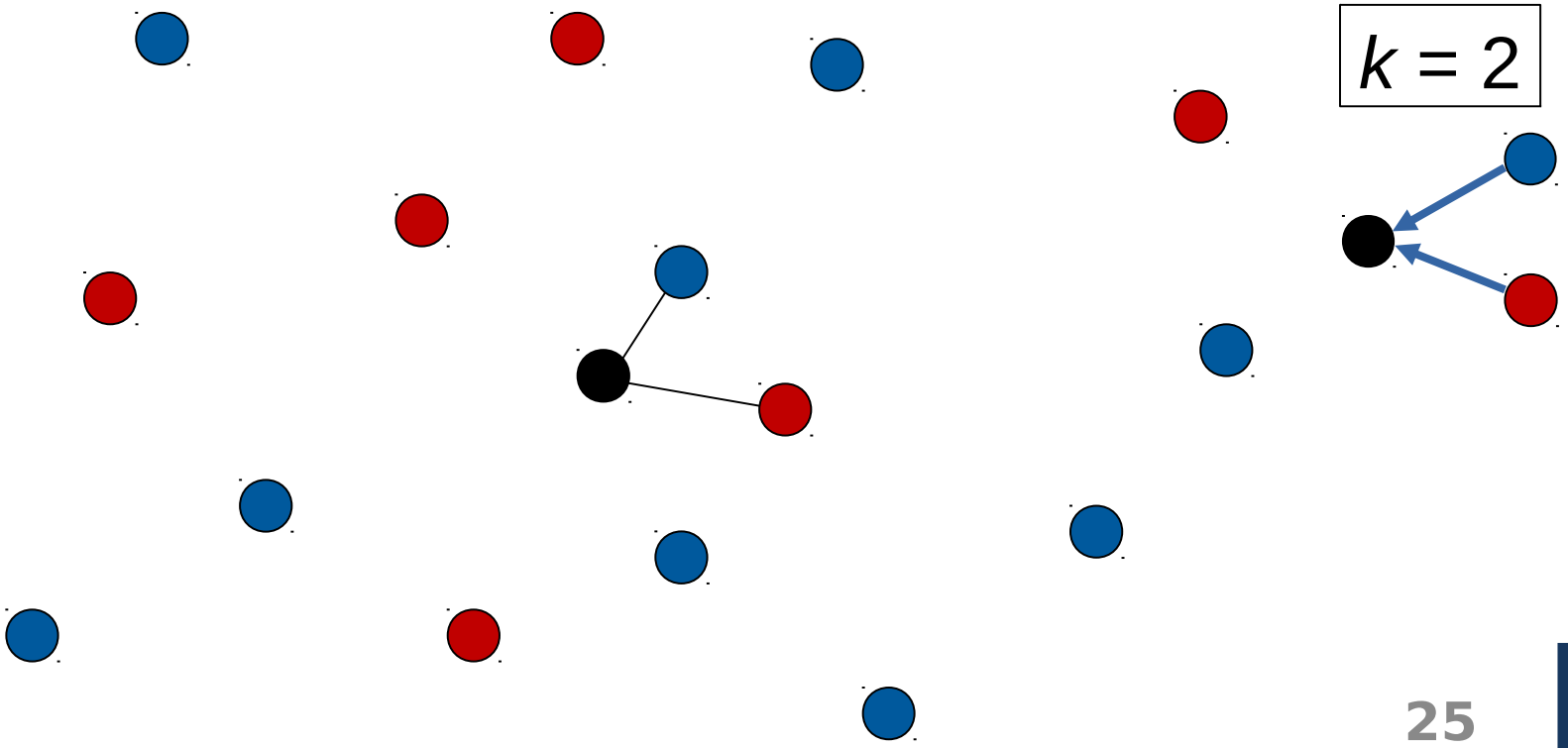
# Classificadores k-NN: o valor de k

- Depois de identificados os **k** vizinhos mais próximos da tupla ***t*** de entrada a ser classificada. Quando **k** = 1, o k-NN atribui à tupla ***t*** a classe do seu elemento mais próximo



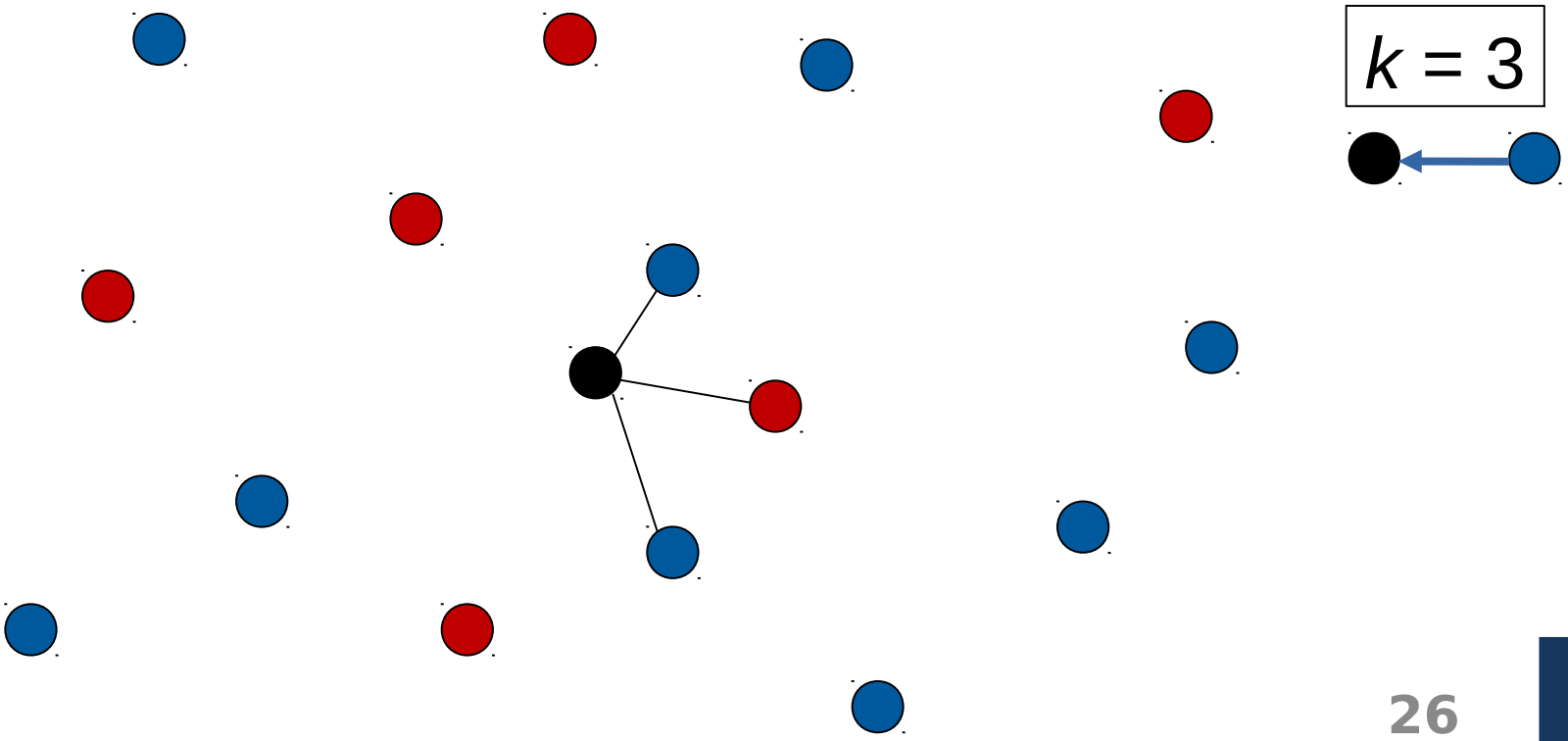
# Classificadores k-NN: o valor de k

- Quando **k** = 2, o k-NN atribui a t a classe predominante entre **k** vizinhos gera um empate e pode ser decidida pela menor distância



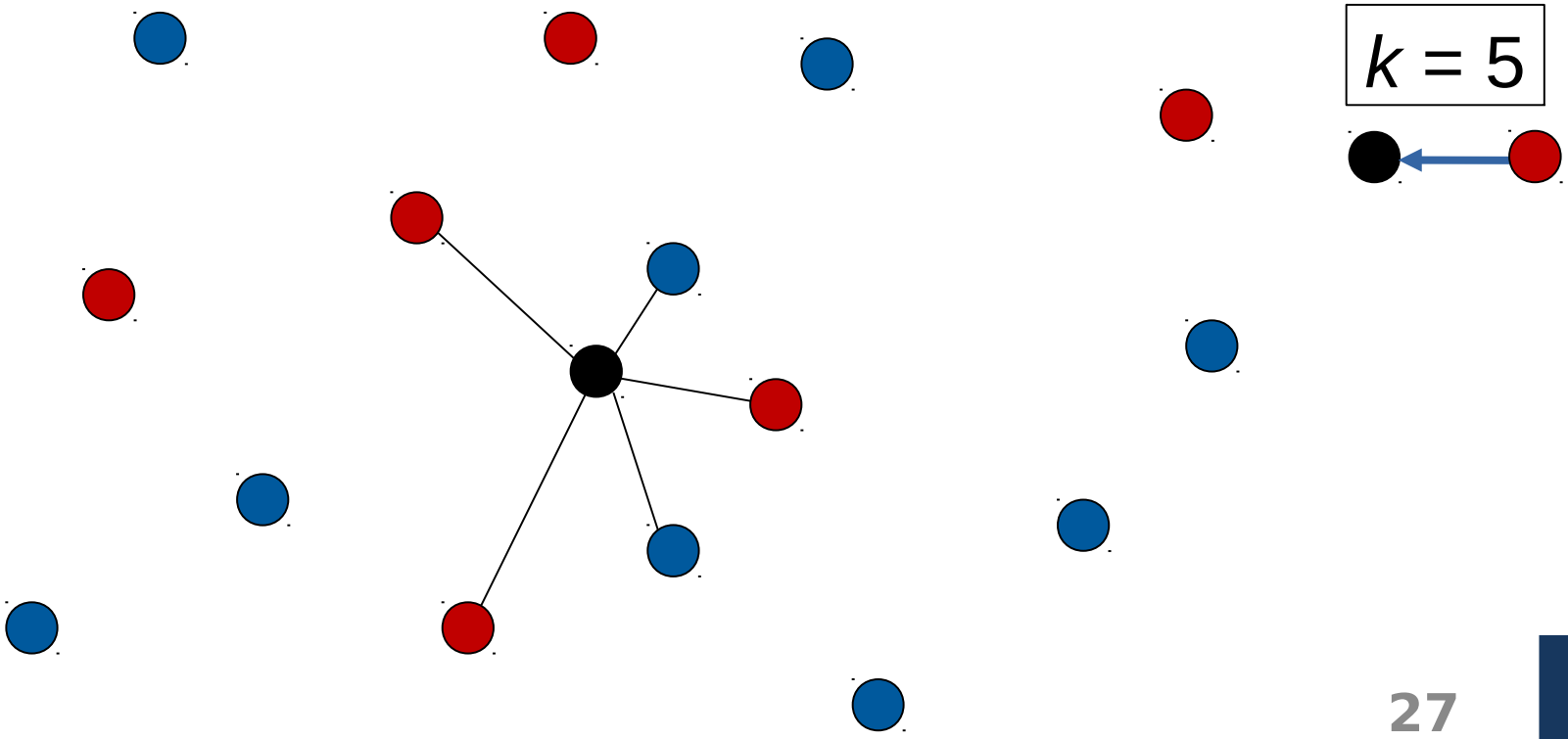
# Classificadores k-NN: o valor de k

- Quando **k** = 3, o k-NN atribui a t a classe predominante entre esses **k** vizinhos



# Classificadores k-NN: o valor de k

- O classe atribuída pode variar de acordo com o valor do parâmetro **k** escolhido
- Um valor adequado pode ser escolhido empiricamente



# Classificadores k-NN: distância

- Proximidade (ou semelhança, similaridade) é definida a partir de uma métrica de distância, como p.e., a distância Euclidiana
- Sejam duas tuplas $X_1 = (x_{11},x_{12},...,x_{1n})$ e $X_2 = (x_{21},x_{22},...,x_{2n})$. A distância Euclidiana entre $X_1$ e $X_2$ é estimada por:

$$dist(X_1,X_2) = \sqrt{\sum_{1}^{n} (x_{1i} - x_{2i})^2}$$

- Essa fórmula exige atributos numéricos
- Quanto menor dist($X_1$,$X_2$), mais próximos, semelhantes, similares são as tuplas $X_1$ e $X_2$

# Classificadores k-NN: transformações nos atributos numéricos

- **Normalização** dos valores dos atributos para evitar que atributos diferentes (idade e salário) contribuam de formas diferentes no cálculo da distância
- Normalização min-max transforma o valor v de um atributo Atr em um valor **v'** no intervalo [0,1]

$$v' = \frac{v - \min_{Atr}}{\max_{Atr} - \min_{Atr}},$$

- onde $\min_{Atr}$ e $\max_{Atr}$ são os valores mínimo e máximo de Atr
- A **padronização** também pode ser utilizada

# Classificadores k-NN: atributos nominais

- Para atributos nominais (cor, meses, história de crédito)
- A diferença entre os valores do atributo (cor) das duas tuplas será zero se os valores forem iguais (cores iguais), e
- A diferença será igual a 1 caso os valores sejam diferentes (cores diferentes)
- Valores intermediários, entre 0 e 1, podem ser adotados para representar diferenças mais (ou menos) “fortes” entre os valores. Por exemplo, o vermelho poderia ser considerado diferente do preto em uma grau maior do que o cinza é do preto

# Classificadores k-NN

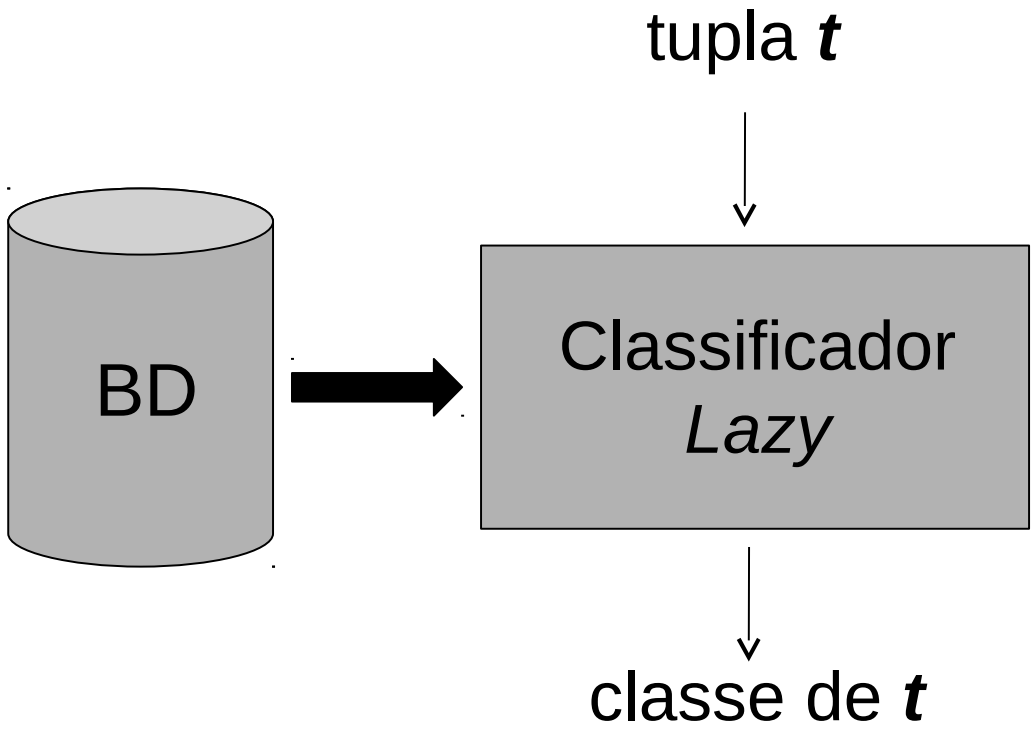
- Classificadores k-NN podem ser utilizados também para problemas de **regressão**, ou seja, para estimar um valor numérico (classe) de uma tupla de entrada
- Nesse caso, o classificador retorna a média dos valores do atributo classe numérico dos **k** vizinhos mais próximos
- Por exemplo, para estimar o valor de venda de um imóvel, calcula-se a média dos valores de venda dos imóveis mais semelhantes ao imóvel de entrada

# Classificadores k-NN: paradigma de aprendizagem

- De uma forma geral, o k-NN apresenta um alto custo computacional para classificar uma nova tupla ***t***, pois tem que calcular a distância de ***t*** para todas as tuplas da base
- Por outro lado, a atualização da base é automaticamente refletida no classificador

# Classificadores k-NN: paradigmas de aprendizagem

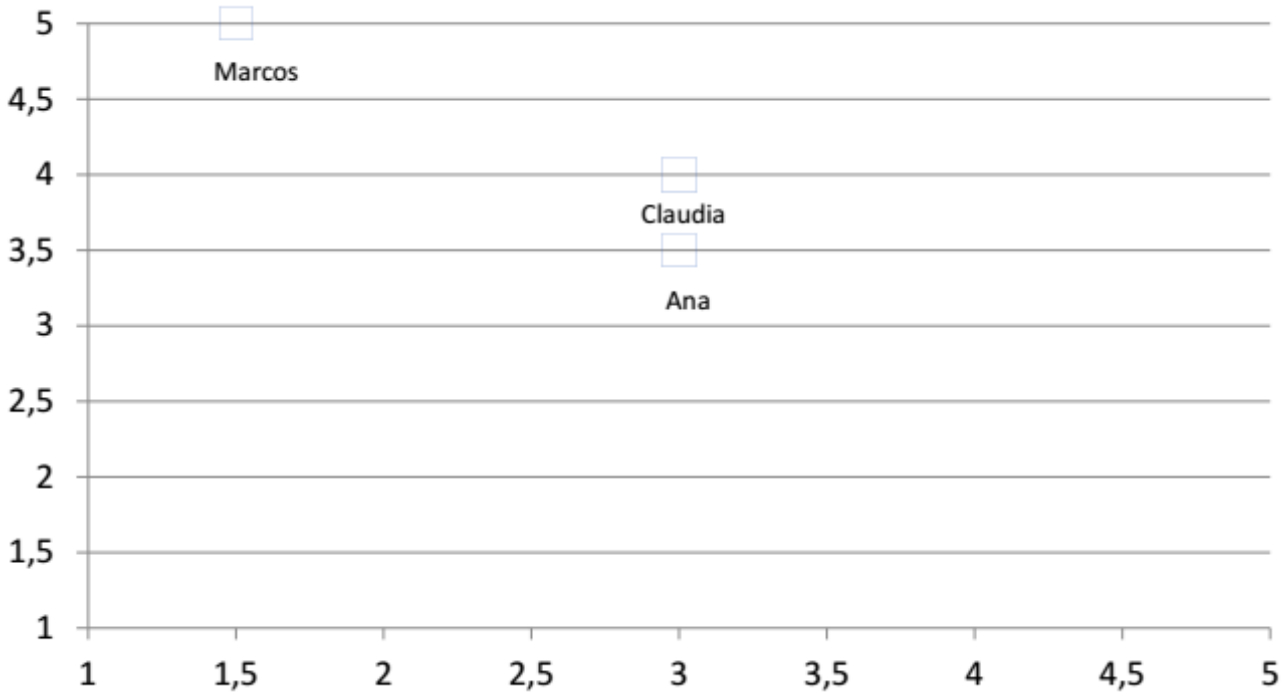
- Não há construção de modelos
- Custo computacional mais caro para classificar a tupla de entrada
- Não necessita retreinar o modelo em caso de atualização da base, por exemplo: k-NN



# Exemplos: notas de recomendação

$$dist(X_1, X_2) = \sqrt{\sum_{1}^{n} (x_{1i} - x_{2i})^2}$$

**Tropa de Elite**




**A culpa é das estrelas**

**Ana x Marcos**
X = 3,0  3,5
Y = 1,5  5,0
$(3,0 – 1,5)^2 = 2,25$
$(3,5 – 5,0)^2 = 2,25$

2,25 + 2,25 = 4,5
Raiz(4,5) = **2,12**

**Ana x Claudia**
X = 3,0  3,5
Y = 3,0  4,0
$(3,0 – 3,0)^2 = 0,00$
$(3,5 – 4,0)^2 = 0,25$

0,00 + 0,25 = 0,25
Raiz(0,25) = **0,50**

# Exemplos: atributos nominais

| Hist. de crédito | Dívida | Garantia | Renda anual | Risco |
|---|---|---|---|---|
| 3 | 1 | 1 | 1 | Alto |
| 2 | 1 | 1 | 2 | Alto |
| 2 | 2 | 1 | 2 | Moderado |
| 2 | 2 | 1 | 3 | Alto |
| 2 | 2 | 1 | 3 | Baixo |
| 3 | 2 | 2 | 3 | Baixo |
| 3 | 2 | 1 | 1 | Alto |
| 1 | 2 | 2 | 3 | Moderado |
| 1 | 2 | 1 | 3 | Baixo |
| 1 | 1 | 2 | 3 | Baixo |
| 1 | 1 | 1 | 1 | Alto |
| 1 | 1 | 1 | 2 | Moderado |
| 1 | 1 | 1 | 3 | Baixo |
| 3 | 1 | 1 | 2 | Alto |

**Nova instância**
História = **Boa** (1)
Dívida = **Alta** (1)
Garantia = **Nenhuma** (1)
Renda = **> 35.000** (3)

**Nova x 3º**
1 1 1 3
2 2 1 2

$1^2 + 1^2 + 0 + 1^2$
$1 + 1 + 0 + 1 = 3$
Raiz(3) = **1,7**

**Nova x 9º**
1 1 1 3
1 2 1 3

$0 + 1^2 + 0 + 0$
$0 + 1 + 0 + 0 = 1$
Raiz(1) = **1**

# Exemplo: problema na escala das variáveis

$$dist(X_1, X_2) = \sqrt{\sum_{1}^{n} (x_{1i} - x_{2i})^2}$$

| # | Idade | Renda anual |
|---|---|---|
| 1 | 60 | 30.000 |
| 2 | 65 | 75.000 |
| 3 | 20 | 29.500 |

**1º x 2º**
60 30.000
65 75.000
$(-5)^2 + (-45.000)^2$
25 + 2.025.000.000 = 2.025.000.025
Raiz(2.025.000.000) = **45.000**

**1º x 3º**
60 30.000
20 29.500
$40^2 + 500^2$
1.600 + 250.000 = 251.600
Raiz(251.600) = **501,59**

# Exemplos: normalização

$$\frac{v - \min_{Atr}}{\max_{Atr} - \min_{Atr}}$$

| | Valores originais | | Valores normalizados | |
|---|---|---|---|---|
| # | **Idade** | **Renda anual** | **Idade** | **Renda anual** |
| 1 | 60 | 30.000 | 0,88 | 0,01 |
| 2 | 65 | 75.000 | 1,00 | 1,00 |
| 4 | 20 | 29.500 | 0,00 | 0,00 |

**1º**
idade' = 60 – 20 / 65 – 20 = **0,88**
renda' = 30000 – 29500 / 75000 – 29500 = **0,01**

**2º**
idade' = 65 – 20 / 65 – 20 = **1**
renda' = 75000 – 29500 / 75000 – 29500 = **1,00**

**3º**
idade' = 20 – 20 / 65 – 20 = **0,00**
renda' = 29500 – 29500 / 75000 – 29500 = **0,00**

**1º x 2º**
0,88 0,01
1,00 1,00
$(-0,12)^2 + (-0,99)^2$
0,014 + 0,980 = 0,994
Raiz(0,994) = **0,996**

**1º x 3º**
0,88 0,01
0,00 0,00
$0,88^2 + 0,01^2$
0,77 + 0,0001 = 0,7701
Raiz(0,7701) = **0,877**

# Exemplos: padronização

$$\frac{v - média(v)}{desviopadrão(v)}$$

| | Valores originais | | Valores padronizados | |
|---|---|---|---|---|
| # | **Idade** | **Renda anual** | **Idade** | **Renda anual** |
| 1 | 60 | 30.000 | 0,48 | -0,57 |
| 2 | 65 | 75.000 | 0,68 | 1,15 |
| 4 | 20 | 29.500 | -1,12 | -0,59 |

**1º**
idade' = 60 – 48 / 25 = **0,48**
renda' = 30000 – 44833 / 26126 = **-0,57**

**2º**
idade' = 65 – 48 / 25 = **0,68**
renda' = 75000 – 44833 / 26126 = **1,15**

**3º**
idade' = 20 – 48 / 25 = **-1,12**
renda' = 29500- – 44833 / 26126 = **-0,59**

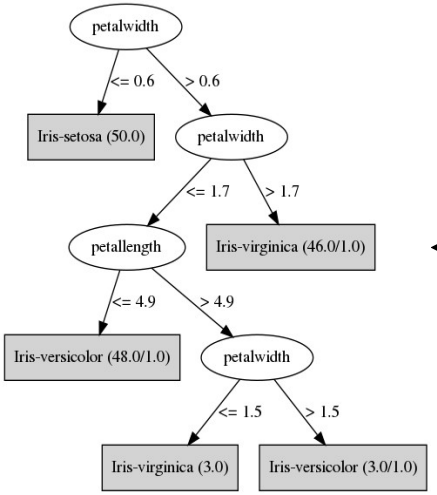
**1º x 2º**
0,48 -0,57
0,68 1,15
$(-0,20)^2 + (-1,72)^2$
0,04 + 2,95 = 2,99
Raiz(2,99) = **1,72**

**1º x 3º**
0,48 0,57
-1,12 -0,59
$1,60^2 + 0,02^2$
2,56 + 0,0004 = 2,5604
Raiz(2,5604) = **1,60**

# Algumas considerações

- Algoritmo de classificação/regressão simples com alto poder preditivo
- Valores pequenos de **k** são prejudicados por ruídos e *outliers*
- Valores grandes de **k** podem gerar *overfitting*
- Preferência por valores de **k** ímpares
- Mais lento para realizar as previsões

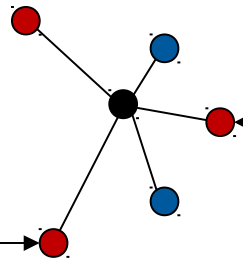
# Paradigmas de Aprendizagem




**Base de Treinamento**

| Risco de Crédito | História de Crédito | | | Dívida | | Garantia | | Renda anual | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Boa | Desconhecida | Ruim | Alta | Baixa | Nenhuma | Adequada | '<15' | '15<=x<=35' | '>35' |
| | 5 | 5 | 4 | 7 | 7 | 11 | 3 | 3 | 4 | 7 |
| Alto 6/14 | 1/6 | 2/6 | 3/6 | 4/6 | 2/6 | 6/6 | 0 | 3/6 | 2/6 | 1/6 |
| Moderado 3/14 | 1/3 | 1/3 | 1/3 | 1/3 | 2/3 | 2/3 | 1/3 | 0 | 2/3 | 1/3 |
| Baixo 5/14 | 3/5 | 2/5 | 0 | 2/5 | 3/5 | 3/5 | 2/5 | 0 | 0 | 5/5 |

**Base de Testes**



**CENTRO UNIVERSITÁRIO UNINORTE**
**CURSO DE PÓS-GRADUAÇÃO EM:** Pós Graduação em Gerência de Banco de Dados.
**DISCIPLINA: Mineração de Dados**

# Classificação

Prof.º: Manoel Limeira
juniorlimeiras@gmail.com