

# Banco de Dados II

Professor Msc. Aparecido Vilela Junior  
[aparecido.vilela@unicesumar.edu.br](mailto:aparecido.vilela@unicesumar.edu.br)

# Classificação

- Atribuir um objeto/evento a uma categoria, pertencente a um conjunto finito de categorias)
  - Diagnóstico médico
  - Detecção de fraude em cartões de crédito
  - Detecção de vírus em redes de computadores
  - Filtragem de spam em e-mails
  - Recomendação de produtos em e-commerce
  - Investimentos financeiros
  - Bioinformática (sequências de DNA)
  - Reconhecimento de voz
  - Reconhecimento de caracteres
  - Reconhecimento de imagens

# Resolução de problemas / planejamento / controle

- Executar ações em um ambiente para atingir um determinado objetivo.
  - Resolver problemas de matemática
  - Jogar xadrez, damas ou gamão
  - Dirigir um carro
  - Pilotar um avião, helicóptero ou foguete
  - Controlar um elevador
  - Controlar um personagem em um jogo
  - Controlar um robô móvel

# Medindo o desempenho

- Acurácia da classificação
- Corretude da solução encontrada
- Qualidade da solução (comprimento, eficiência)
- Tempo de execução

## Por que estudar aprendizado de máquina?

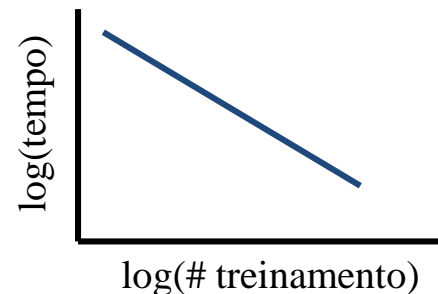
- Desenvolver sistemas que são muito difíceis/caros de construir manualmente porque requerem conhecimentos detalhados de uma determinada tarefa.
- Desenvolver sistemas que possam se adaptar a usuários individualmente.
  - Filtro de notícias ou e-mail personalizado.
  - Sistemas de educação personalizados.
- Extrair conhecimento de grandes bases de dados (**mineração de dados**).
  - Análise de cestas de compras (ex.: fraldas e cerveja)

# Qual é a relação com a mineração de dados?

- A mineração de dados é o processo de extração automática de **conhecimento** a partir de grandes bases de dados.
- Algoritmos de aprendizado automático podem ser vistos como algoritmos que extraem um padrão de comportamento a partir de dados (exemplos).
  - Logo, podem ser utilizados como algoritmos de mineração de dados.
- Porém, algoritmos de aprendizado nem sempre utilizam bases de dados.
  - Podem aprender diretamente a partir da interação com o ambiente ou com um simulador.
- Ambas as áreas “emprestam” muitos métodos da área de estatística.

# Por que estudar aprendizado de máquina?

- Estudos computacionais do aprendizado podem ajudar a entender o aprendizado em humanos e outros organismos.
  - Aprendizado neural “Hebbiano”
    - “Neurônios que disparam juntos se conectam.”
  - A dificuldade relativa de humanos em aprender conceitos disjuntivos em relação a conceitos conjuntivos.
  - Lei da prática



## Por que estudar aprendizado de máquina?

- Grande quantidade de algoritmos efetivos e eficientes.
- Grande quantidade de dados disponíveis (principalmente online).
- Grande quantidade de recursos computacionais disponíveis.



# Definindo a tarefa de aprendizado

Melhorar na tarefa, T, com respeito à métrica de desempenho, D, baseado na experiência, E.

T: Jogar Damas

D: Percentagem de jogos ganhos contra um adversário arbitrário

E: Jogar jogos contra ele próprio

T: Reconhecer palavras escritas a mão

D: Percentagem de palavras corretamente reconhecidas

E: Base de dados de imagens com as palavras correspondentes

T: Dirigir em estradas usando sensores de visão

D: Distância média dirigida antes de um erro

E: Sequência de imagens e comandos de direção gravados de uma pessoa dirigindo

T: Categoriza mensagens de e-mail como spam ou legítimas.

D: Percentagem de mensagens de e-mail corretamente classificadas

E: Base de dados de mensagens, algumas classificados por humanos

## Exemplo de Problema de Aprendizado

- Aprender a jogar damas jogando contra si mesmo.
- Veremos uma abordagem similar a usada no primeiro sistema de aprendizado automático desenvolvido por Arthur Samuels na IBM em 1959.

# Experiência para Treinamento

- **Experiência direta:** Dar alguns exemplos de pares de entrada e saída da função alvo.
  - Tabuleiros de damas identificados com a jogada correta, por exemplo extraídos do jogo de um especialista.
- **Experiência indireta:** Dar feedback que ***não*** é um par de E/S para a função alvo.
  - Sequencias potencialmente arbitrárias de jogadas e o resultado final do jogo.
- **Problema da atribuição de crédito:** Como atribuir crédito a jogadas individuais quando a experiência é indireta?

- Exemplos aleatórios.
  - Exemplos negativos também ou só positivos?
- Exemplos de treinamento escolhidos por um professor benevolente.
  - Exemplos “didáticos”.
- Algoritmo pode pedir o rótulo de um exemplo qualquer no ambiente.
- Algoritmo pode construir um exemplo e pedir o rótulo desse exemplo.
- Algoritmo pode conduzir experimentos no ambiente sem ajuda humana.

# Base

Nº	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
1	Não	Sim	Não	Sim	Sim	Não	Não
2	Sim	Não	Sim	Sim	Sim	Não	Não
3	Não	Sim	Não	Sim	Sim	Não	Não
4	Sim	Sim	Não	Sim	Sim	Não	Não
5	Não	Não	Sim	Não	Não	Não	Não
6	Não	Não	Não	Não	Sim	Não	Não
7	Não	Não	Não	Sim	Não	Não	Não
8	Não	Não	Não	Não	Não	Não	Sim
9	Não	Não	Não	Não	Não	Sim	Sim
10	Não	Não	Não	Não	Não	Sim	Não

Suporte = Número de registros com X e Y / Número total de registros

# Suporte

- Número de vezes que o produto aparece como sim (vendido), dividido pelo número total de transações.
- Por exemplo o produto Leite, só aparece em 2 transações:
- $2/10 = 0,20$  é o suporte desse item relacionado a todas as vendas.

# Suporte

Produto	Suporte
Leite	0,20
Café	0,30
Cerveja	0,2
Pão	0,5
Manteiga	0,5
Arroz	0,2
Feijão	0,2

# Suporte 2 itens

Produtos	Suporte
Café, pão	0,3
Café, Manteiga	0,3
Pão, manteiga	0,4



# Arquivo carregado

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter  
Choose: None Apply

Current relation  
Relations: weather  
Instances: 14

Attributes: 5  
Sum of weights: 14

Selected attribute  
Name: outlook  
Missing: 0 (0%)  
Distinct: 3  
Type: Nominal  
Unique: 0 (0%)

No.	Label	Count	Weight
1	sunny	5	5.0
2	overcast	4	4.0
3	rainy	5	5.0

Attributes  
All None Invert Pattern

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: play (Nom) Visualize All

Status  
OK Log x 0

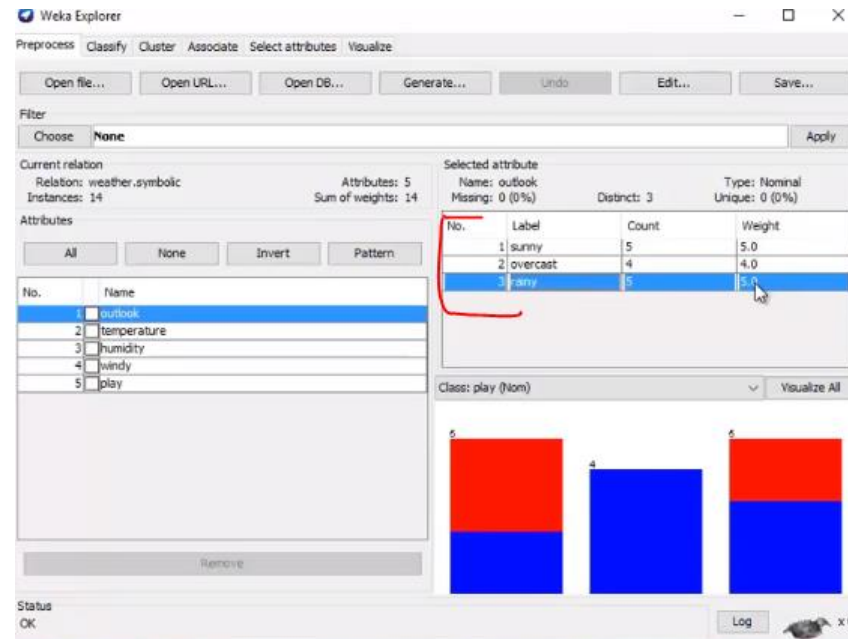
# Arquivo possui

- Nome
- Atributos
- Dados

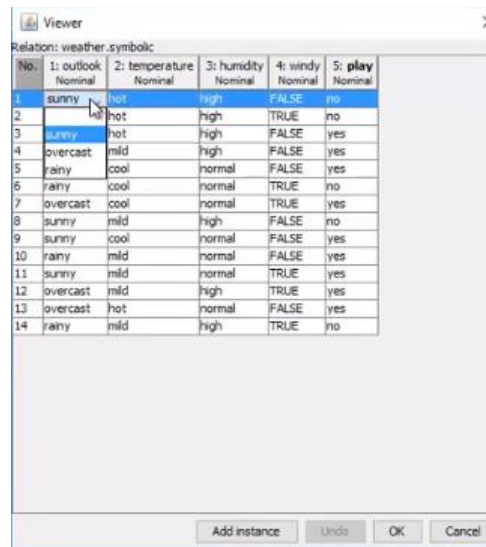
# Exemplo

```
weather.nominal.off 1
2
3 @relation weather.symbolic [
4   @attribute outlook {sunny, overcast, rainy}
5   @attribute temperature {hot, mild, cool}
6   @attribute humidity {high, normal}
7   @attribute windy {TRUE, FALSE}
8   @attribute play {yes, no}
9
10  @data
11  sunny,hot,high,FALSE,no
12  sunny,hot,high,TRUE,no
13  overcast,hot,high,FALSE,yes
14  rainy,mild,high,FALSE,yes
15  rainy,cool,normal,FALSE,yes
16  rainy,cool,normal,TRUE,no
17  overcast,cool,normal,TRUE,yes
18  sunny,mild,high,FALSE,no
19  sunny,cool,normal,FALSE,yes
20  rainy,mild,normal,FALSE,yes
21  sunny,mild,normal,TRUE,yes
22  overcast,mild,high,TRUE,yes
23  overcast,hot,normal,FALSE,yes
24  rainy,mild,high,TRUE,no
```

# Qtde de Valores nos atributos



# Edição de Registros



Relation: weather.symbolic

No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: play Nominal
1	sunny	hot	high	FALSE	no
2		hot	high	TRUE	no
3	sunny	hot	high	FALSE	yes
4	overcast	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Add instance Undo OK Cancel

# Unique

Viewer

Relation: weather.symbolic

No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: play Nominal
1	sunny	hot	high	TRUE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	TRUE	yes
4	rainy	mild	high	TRUE	yes
5	rainy	cool	normal	TRUE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	TRUE	no
9	sunny	cool	normal	TRUE	yes
10	rainy	mild	normal	TRUE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

tribute outlook (0%) Distinct: 3 Type: Nominal Unique: 0 (0%)

Label	Count	Weight
sunny	5	5.0
overcast	4	4.0
rainy	5	5.0

Nom) Visualize All

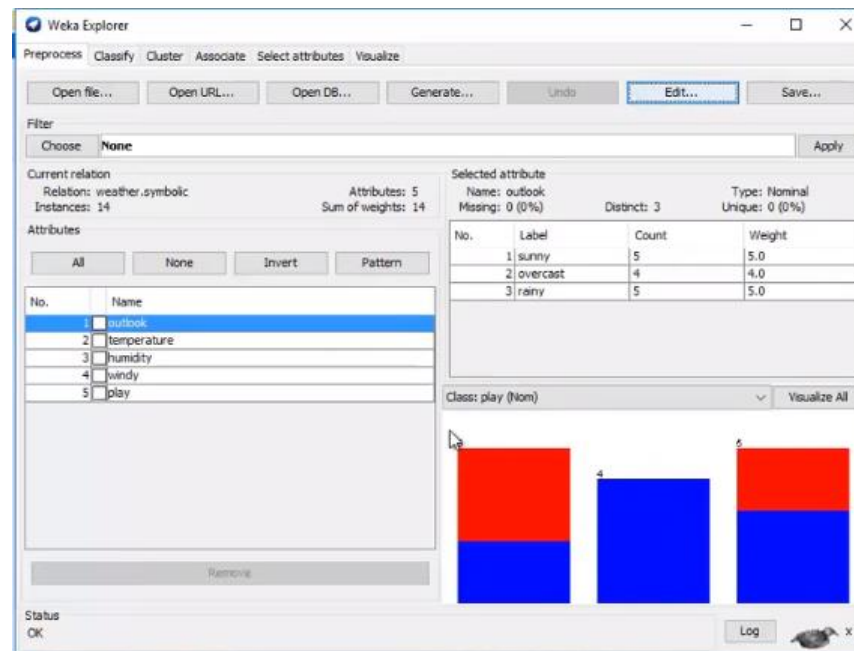
Add instance Undo OK Cancel

Remove

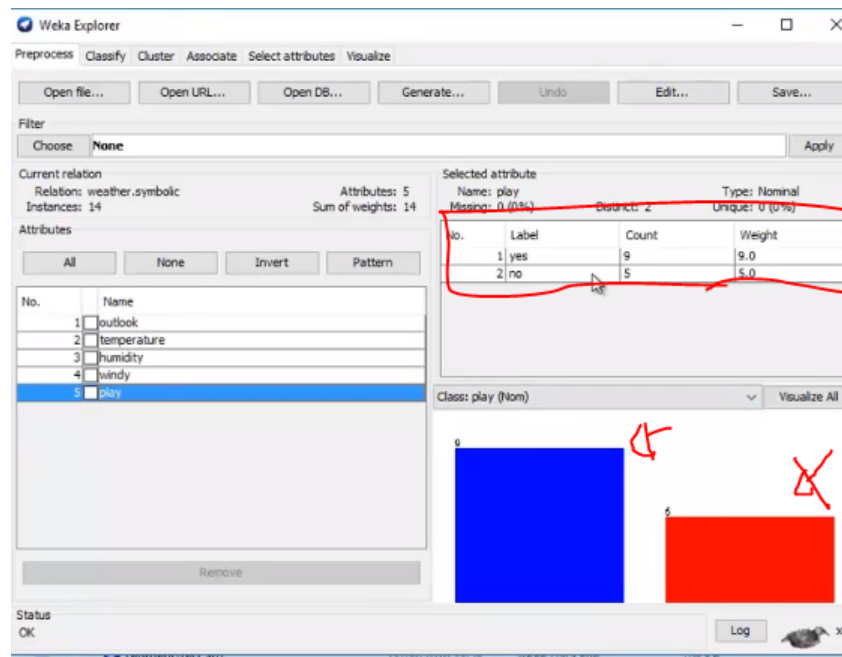
Status OK

Log x 0

# Gráficos

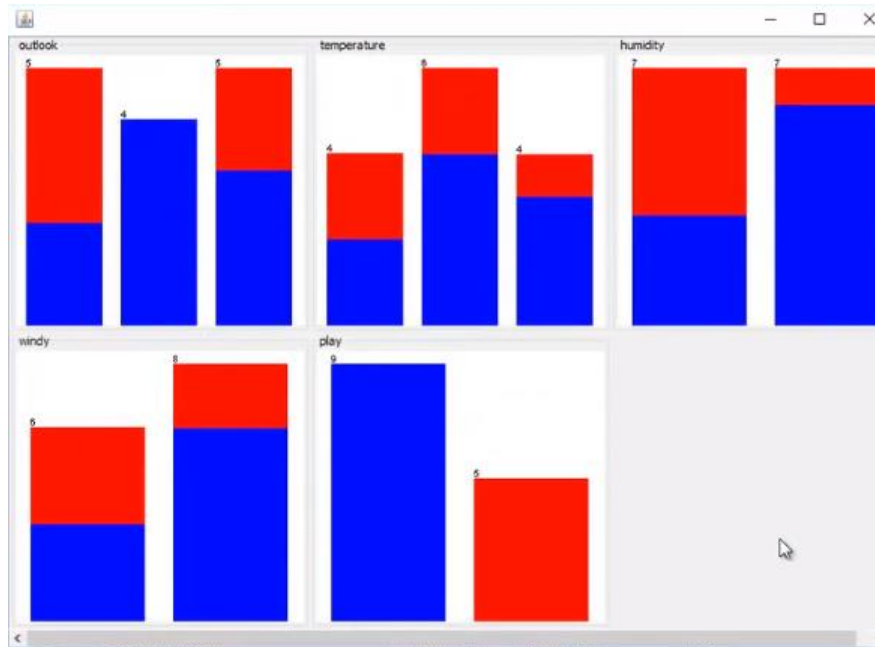


# Gráficos

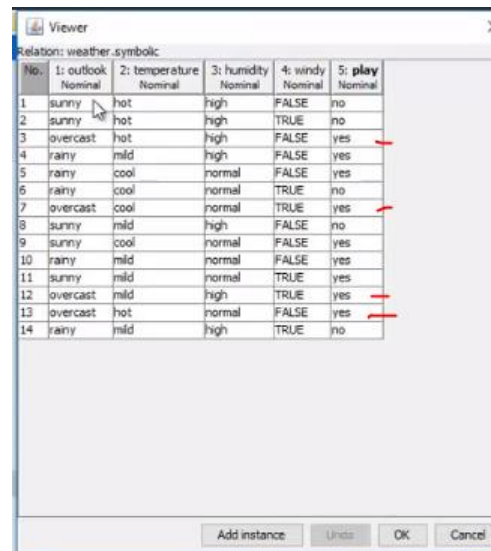




# Visualizando todos os gráficos



# Todos que são overcast -> Jogam



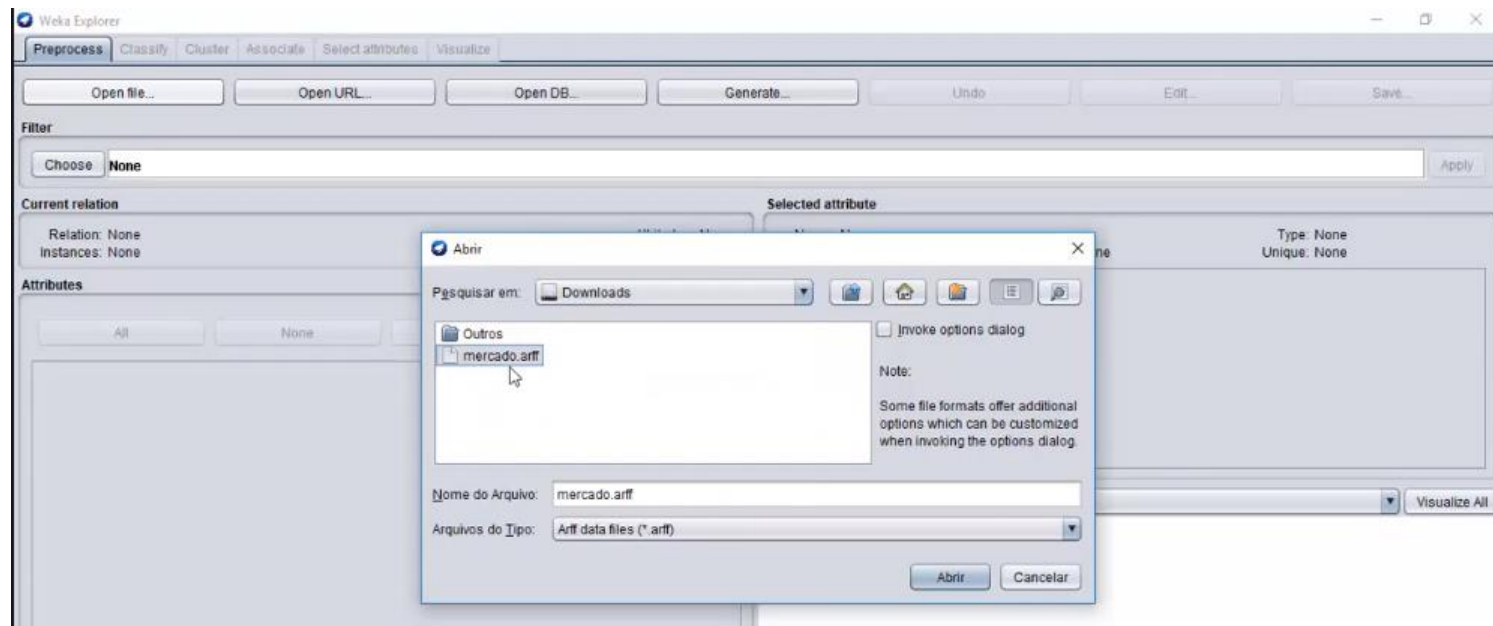
Relation: weather.symbolic

No.	1: outlook Nominal	2: temperature Nominal	3: humidity Nominal	4: windy Nominal	5: play Nominal
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Add instance   Undo   OK   Cancel

# Arquivo mercado.arrf

```
mercado.arrf
1 @relation mercado
2
3 @attribute leite {sim, nao}
4 @attribute cafe {sim, nao}
5 @attribute cerveja {sim, nao}
6 @attribute pao {sim, nao}
7 @attribute manteiga {sim, nao}
8 @attribute arroz {sim, nao}
9 @attribute feijao {sim, nao}
10
11 @data
12 nao, sim, nao, sim, sim, nao, nao
13 sim, nao, sim, sim, sim, nao, nao
14 nao, sim, nao, sim, sim, nao, nao
15 sim, sim, nao, sim, sim, nao, nao
16 nao, nao, sim, nao, nao, nao, nao
17 nao, nao, nao, nao, sim, nao, nao
18 nao, nao, nao, sim, nao, nao, nao
19 nao, nao, nao, nao, nao, nao, sim
20 nao, nao, nao, nao, nao, sim, sim
21 nao, nao, nao, nao, nao, sim, nao
```



# Aba Associate

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Open file... Open URL... Open DB... Generate... Undo Edit... Save...

Filter  
Choose None Apply

Current relation  
Relation: mercado  
Instances: 19  
Attributes: 7  
Sum of weights: 10

Attributes  
All None Invert Pattern

No.	Name
1	leite
2	cafe
3	cerveja
4	pao
5	manteiga
6	arroz
7	feijao

Remove

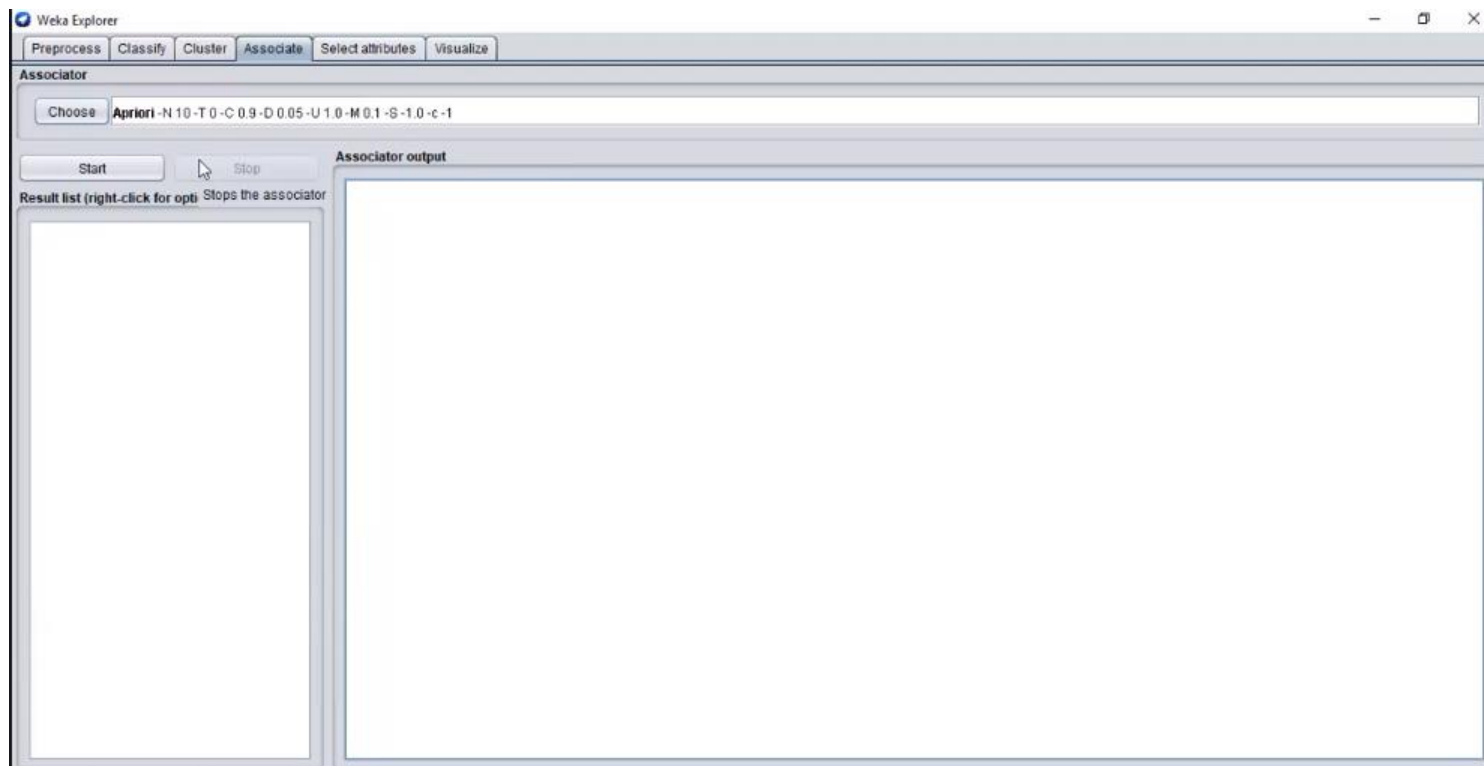
Selected attribute  
Name: feijao  
Missing: 0 (0%)  
Distinct: 2  
Type: Nominal  
Unique: 0 (0%)

No.	Label	Count	Weight
1	sim	2	2.0
2	nao	8	8.0

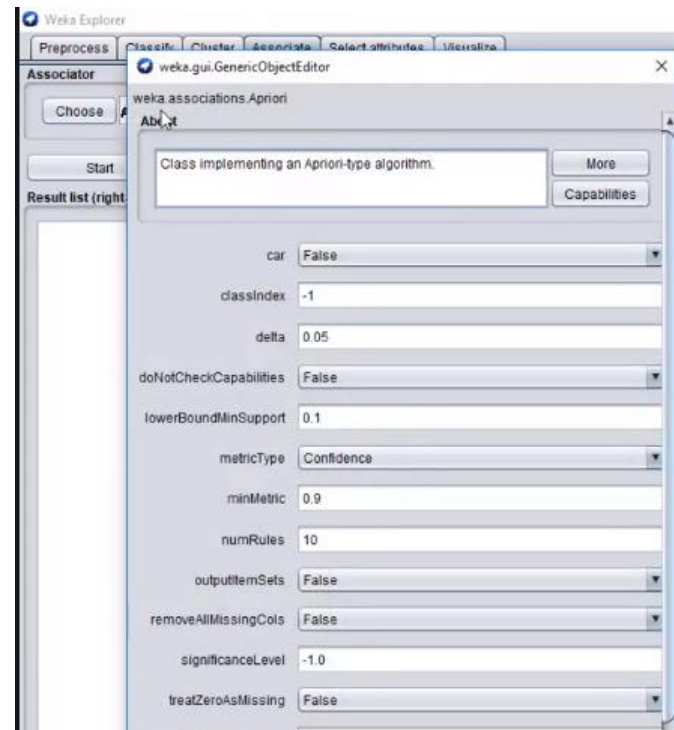
Class: feijao (Nom) Visualize All

Status  
OK Log x 0

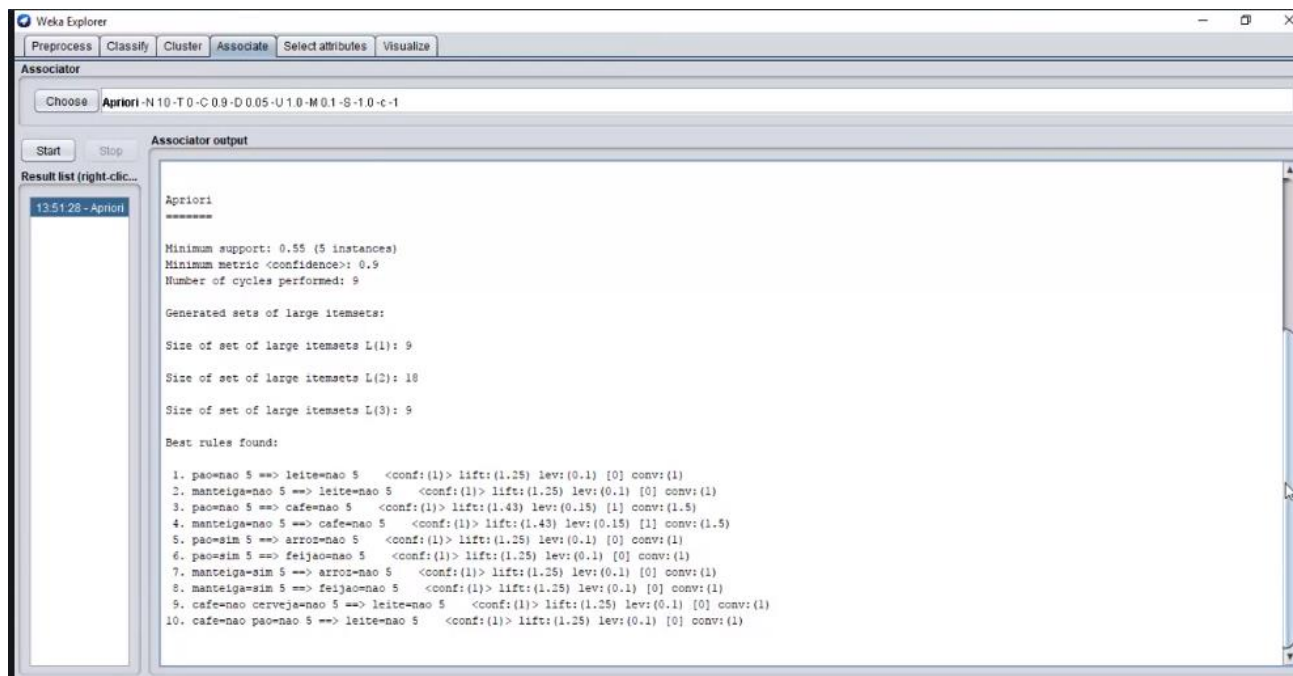
# A Priori



# Parâmetros



# Start



The screenshot shows the Weka Explorer application window. The 'Associate' tab is selected in the top menu. The 'Choose' button is clicked, and the 'Apriori' algorithm is selected from the list. The 'Start' button is also clicked. The 'Result list (right-click...)' panel on the left shows the selected algorithm. The 'Associator output' panel on the right displays the results of the Apriori algorithm.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Choose Apriori-N10-T0-C0.9-D0.05-U1.0-M0.1-S-1.0-c-1

Start Stop

Result list (right-click...)

13:51:28 - Apriori

Associator output

Apriori

\*\*\*\*\*

Minimum support: 0.55 (5 instances)  
Minimum metric (confidence): 0.9  
Number of cycles performed: 9

Generated sets of large itemsets:

Size of set of large itemsets L(1): 9  
Size of set of large itemsets L(2): 18  
Size of set of large itemsets L(3): 9

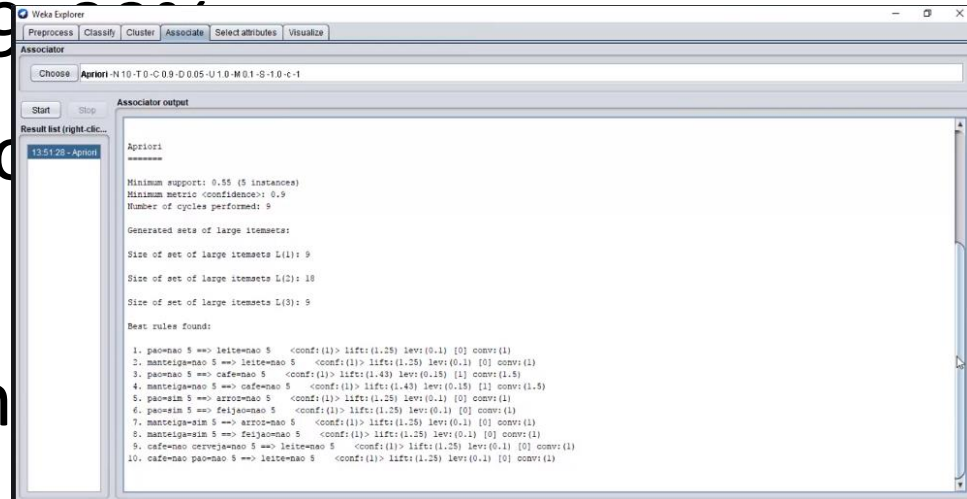
Best rules found:

1. pao=nao 5 ==> leite=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
2. manteiga=nao 5 ==> leite=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
3. pao=nao 5 ==> cafe=nao 5 <conf:(1)> lift:(1.43) lev:(0.15) [1] conv:(1.5)
4. manteiga=nao 5 ==> cafe=nao 5 <conf:(1)> lift:(1.43) lev:(0.15) [1] conv:(1.5)
5. pao=sim 5 ==> arroz=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
6. pao=sim 5 ==> feijao=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
7. manteiga=sim 5 ==> arroz=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
8. manteiga=sim 5 ==> feijao=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
9. cafe=nao cerveja=nao 5 ==> leite=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)
10. cafe=nao pao=nao 5 ==> leite=nao 5 <conf:(1)> lift:(1.25) lev:(0.1) [0] conv:(1)



# Start

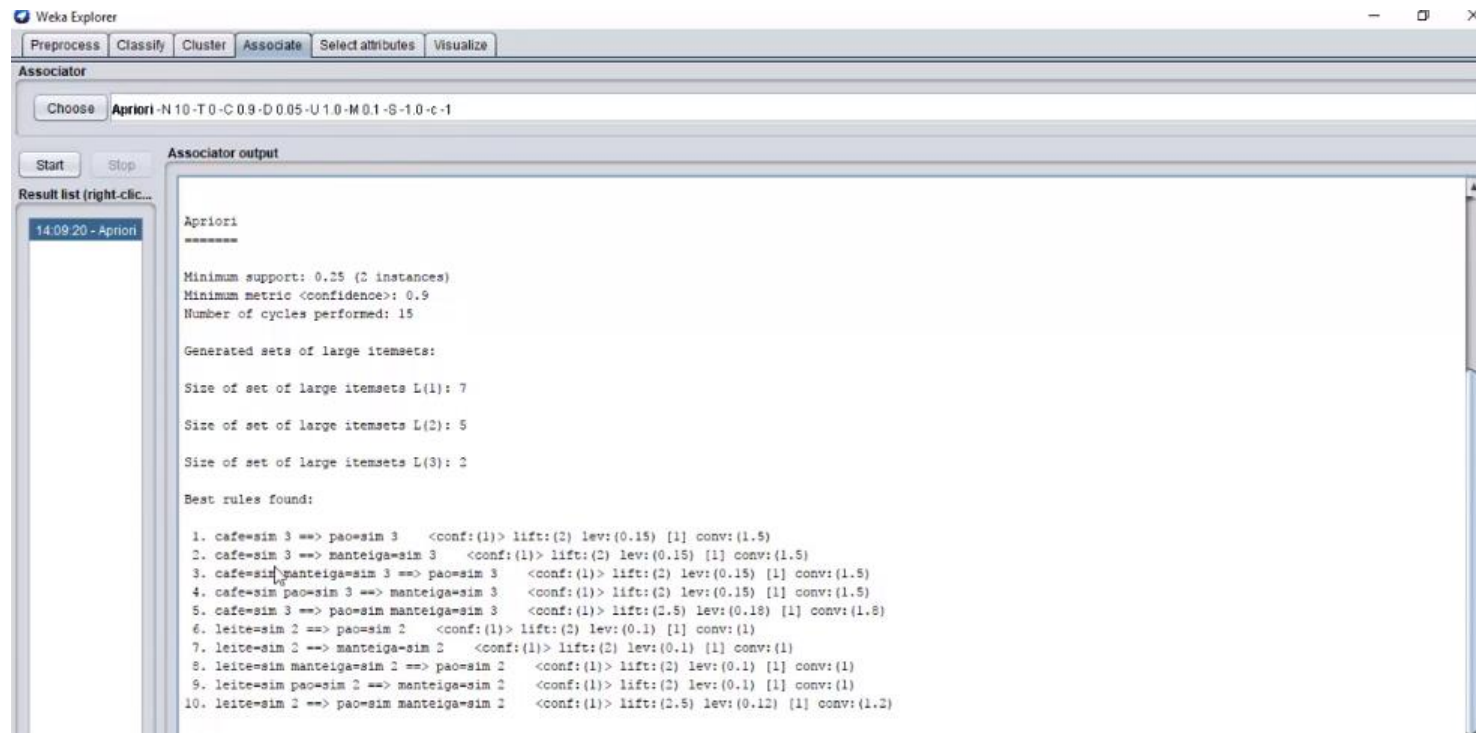
- Suporte
- Confiança  $\rightarrow 0,9$
- Número de ciclos
- Número de itens



# Alterando o Arquivo

```
mercado.aff mercado2.aff
1 @relation mercado
2
3 @attribute leite (sim)
4 @attribute cafe (sim)
5 @attribute cerveja (sim)
6 @attribute pao (sim)
7 @attribute manteiga (sim)
8 @attribute arroz (sim)
9 @attribute feijao (sim)
10
11 @data
12 ?, sim, ?, sim, sim, ?, ?
13 sim, ?, sim, sim, sim, ?, ?
14 ?, sim, ?, sim, sim, ?, ?
15 sim, sim, ?, sim, sim, ?, ?
16 ?, ?, sim, ?, ?, ?, ?
17 ?, ?, ?, ?, sim, ?, ?
18 ?, ?, ?, sim, ?, ?, ?
19 ?, ?, ?, ?, ?, ?, sim
20 ?, ?, ?, ?, ?, sim, sim
21 ?, ?, ?, ?, ?, sim, ?
```

# Alterando a análise



The screenshot shows the Weka Explorer application window. The 'Associate' tab is selected in the top menu. The 'Choose' button is clicked, and the 'Apriori' algorithm is selected from the list. The 'Start' button is also visible. The 'Associator output' pane displays the results of the Apriori algorithm execution.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

Choose Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 1

Start Stop

Associator output

Result list (right-click...)

14.09.20 - Apriori

Apriori  
\*\*\*\*\*

Minimum support: 0.25 (2 instances)  
Minimum metric <confidence>: 0.9  
Number of cycles performed: 15

Generated sets of large itemsets:

Size of set of large itemsets L(1): 7  
Size of set of large itemsets L(2): 5  
Size of set of large itemsets L(3): 2

Best rules found:

1. cafe=sim 3 ==> pao=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
2. cafe=sim 3 ==> manteiga=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
3. cafe=sim manteiga=sim 3 ==> pao=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
4. cafe=sim pao=sim 3 ==> manteiga=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
5. cafe=sim 3 ==> pao=sim manteiga=sim 3 <conf:(1)> lift:(2.5) lev:(0.18) [1] conv:(1.8)
6. leite=sim 2 ==> pao=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
7. leite=sim 2 ==> manteiga=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
8. leite=sim manteiga=sim 2 ==> pao=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
9. leite=sim pao=sim 2 ==> manteiga=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
10. leite=sim 2 ==> pao=sim manteiga=sim 2 <conf:(1)> lift:(2.5) lev:(0.12) [1] conv:(1.2)

# Verificando o suporte

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

Choose Apriori-N10-T0-C09-D005-U10-M01-S-10-c-1


Start Stop

Associator output


Result list (right-click...)

14.09.20 - Apriori

Apriori  
=====

Minimum support: 0.25 (2 instances) 

Minimum metric <confidence>: 0.9

Number of cycles performed: 15 

Generated sets of large itemsets:

Size of set of large itemsets L(1): 7

Size of set of large itemsets L(2): 5

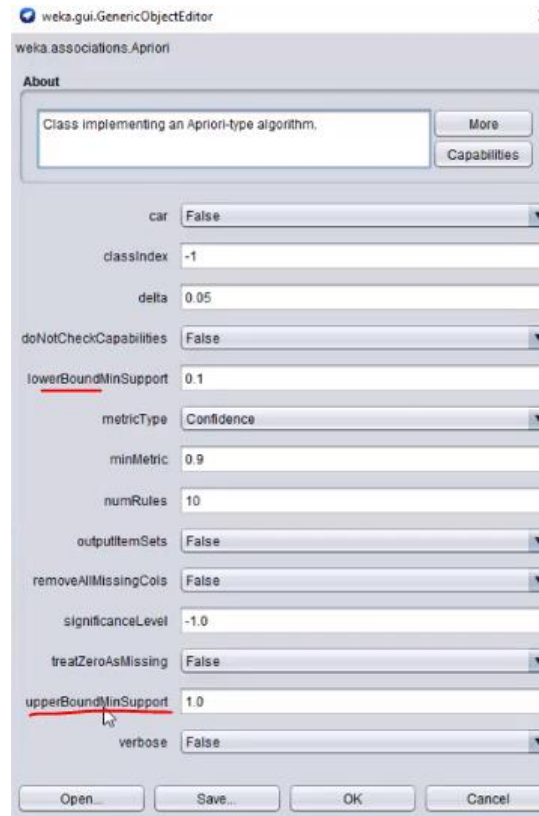
Size of set of large itemsets L(3): 2

Best rules found:

1. cafe=sim 3 ==> pao=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
2. cafe=sim 3 ==> manteiga=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
3. cafe=sim manteiga=sim 3 ==> pao=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
4. cafe=sim pao=sim 3 ==> manteiga=sim 3 <conf:(1)> lift:(2) lev:(0.15) [1] conv:(1.5)
5. cafe=sim 3 ==> pao=sim manteiga=sim 3 <conf:(1)> lift:(2.5) lev:(0.15) [1] conv:(1.8)
6. leite=sim 2 ==> pao=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
7. leite=sim 3 ==> manteiga=sim 3 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
8. leite=sim manteiga=sim 2 ==> pao=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
9. leite=sim pao=sim 2 ==> manteiga=sim 2 <conf:(1)> lift:(2) lev:(0.1) [1] conv:(1)
10. leite=sim 2 ==> pao=sim manteiga=sim 2 <conf:(1)> lift:(2.5) lev:(0.12) [1] conv:(1.2)

# Parâmetros

- Upper -> Geralmente 1 ou seja 100%
- Lower -> Suporte mínimo de 10%
- O delta é o parâmetro de decremento.
- MinMetricType -> Confiança mínima



# Analizando

- Altere o delta para .10, execute e analise o que o que ocorreu.

# Aplicando o suporte

Passo 1: Calcular o suporte de conjuntos com 1 item (itemsets)

Item	Suporte
Leite	0,2
Café	0,3
Cerveja	0,2
Pão	0,5
Manteiga	0,5
Arroz	0,2
Feijão	0,2

# Altere o valor

- Upper -> 0,30
- Lower -> 0,30



# Mercado – Nova Zelândia

weka.gui.GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm. [More](#) [Capabilities](#)

car

classIndex

delta

doNotCheckCapabilities

lowerBoundMinSupport

metricType

minMetric

numRules

outputItemSets

removeAllMissingCols

significanceLevel

treatZeroAsMissing

upperBoundMinSupport

