

FP - Growth

Ana Flávia Dias
Camilla Damasceno
Jorge Luiz Mendes
Luíza Ávila
Stefany Gaspar

Introdução

- Muito utilizado para mineração de dados.
- Busca por padrões frequentes.
- Descobrir relações entre itens/eventos em uma base de dados, possibilitando o encontro de irregularidades e padrões interessantes a serem explorados.
- O algoritmo utiliza uma estrutura de árvore para representar o banco de dados.
- **Aplicações:**
 - análise de transações no ramo de vendas.
- Semelhante ao algoritmo **Apriori**
 - Regra
 - Suporte
 - Confiança
 - Lift

Algoritmo

- Utiliza estrutura de árvore para representar o banco de dados, essa representação é chamada de **frequent-pattern tree** ou **FP tree**.
- Supera a performance de vários outros algoritmos.
- **Passo 1:**
 - Geração de conjuntos frequentes: procura-se apenas os conjuntos de itens significativos, geralmente são conjuntos que atingem um suporte mínimo.
- **Passo 2:**
 - Geração de regras para os conjuntos: para cada conjunto da etapa anterior são identificadas regras que estão acima de um limite mínimo de confiança.

Transação	Leite	Café	Cerveja	Pão	Manteiga	Arroz	Feijão
T1	Sim	Sim	Não	Sim	Sim	Não	Não
T2	Sim	Não	Sim	Sim	Sim	Não	Não
T3	Não	Sim	Não	Sim	Sim	Não	Não
T4	Sim	Sim	Não	Sim	Sim	Não	Não
T5	Não	Não	Sim	Não	Não	Não	Não
T6	Sim	Sim	Não	Não	Sim	Não	Não
T7	Não	Não	Não	Sim	Não	Não	Não
T8	Sim	Não	Não	Não	Não	Não	Sim
T9	Não	Não	Não	Não	Não	Sim	Sim
T10	Sim	Sim	Não	Sim	Não	Sim	Não

Execução do algoritmo

- **Passo 1:** Igual ao Apriori.
 - Considerando Suporte = 0.3 e confiança = 0.8.
 - Frequência para conjuntos com 1 item:

Itens	Número de transações	Suporte
Leite	2	$6 / 10 = 0.6$
Café	5	$5 / 10 = 0.5$
Cerveja	2	$2 / 10 = 0.2$
Pão	6	$6 / 10 = 0.6$
Manteiga	5	$5 / 10 = 0.5$
Arroz	2	$2 / 10 = 0.2$
Feijão	2	$2 / 10 = 0.2$

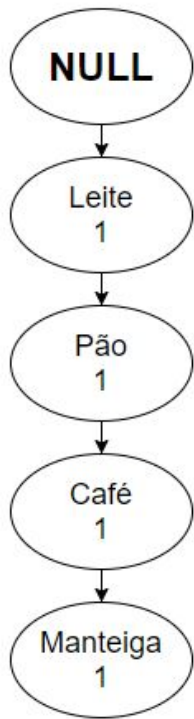
Execução do algoritmo

- Os itens que satisfazem o valor do suporte mínimo são: **Leite, Pão, Café e Manteiga**, portanto os demais não permanecem na próxima iteração.

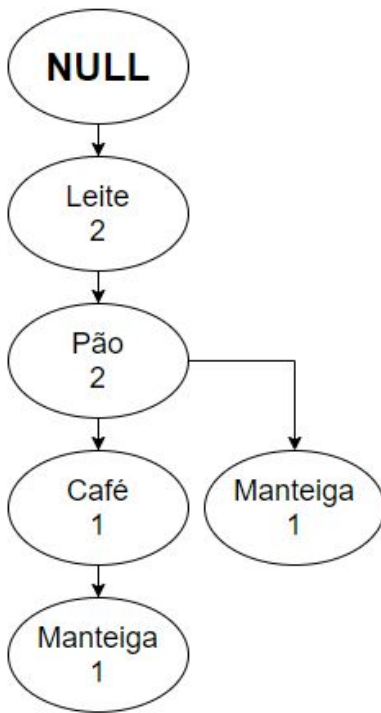
Itens	Número de transações
Leite	6
Pão	6
Café	5
Manteiga	5

Transações	Itens	Itens Ordenados
T1	Leite, Café, Pão, Manteiga	Leite, Pão, Café, Manteiga
T2	Leite, Cerveja, Pão, Manteiga	Leite, Pão, Manteiga
T3	Café, Pão, Manteiga	Pão, Café, Manteiga
T4	Leite, Café, Pão, Manteiga	Leite, Pão, Café, Manteiga
T5	Cerveja	-
T6	Leite, Café, Manteiga	Leite, Café, Manteiga
T7	Pão	Pão
T8	Leite, Feijão	Leite
T9	Arroz, Feijão	-
T10	Leite, Café, Pão, Arroz	Leite, Pão, Café

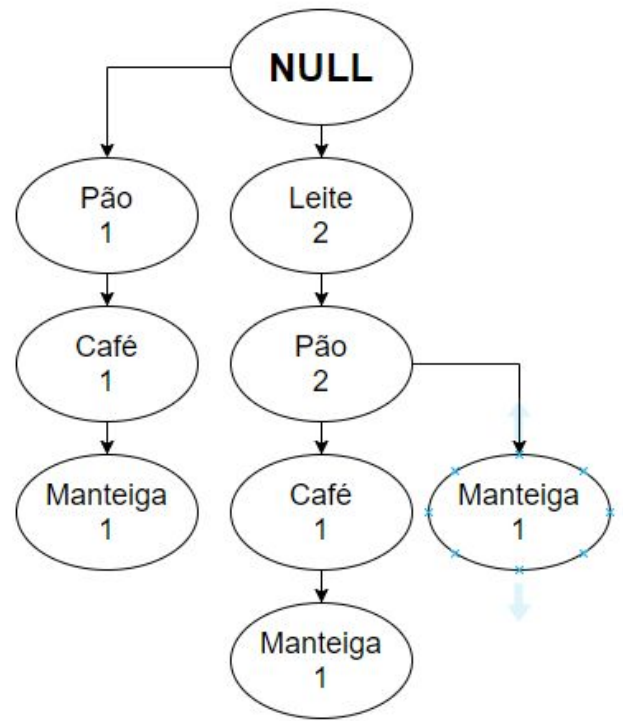
Árvores geradas - 1, 2, 3



T1 > Leite, Pão, Café, Manteiga



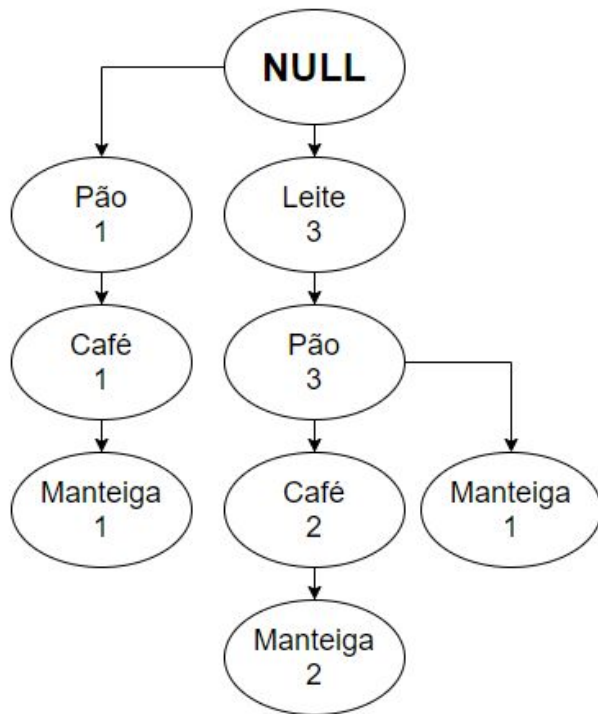
T2 > Leite, Pão, Manteiga



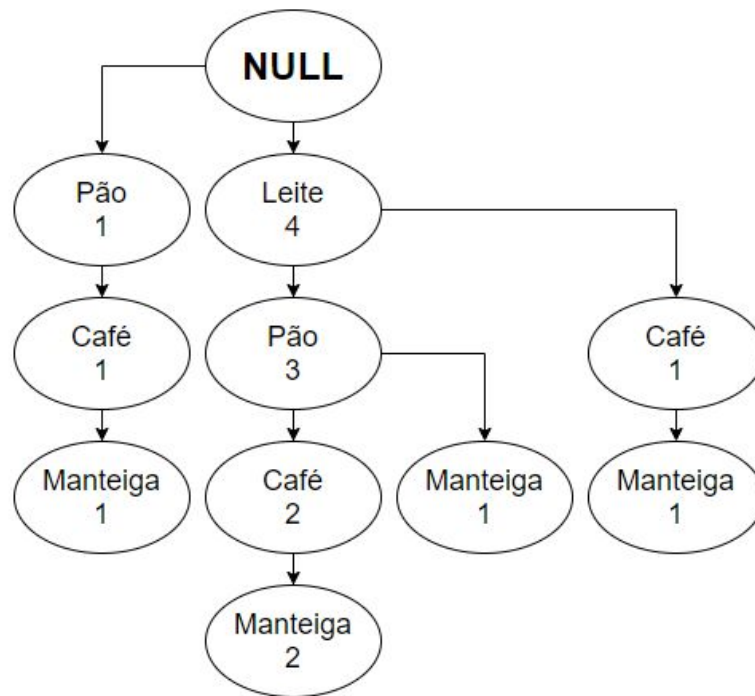
T3 > Pão, Café, Manteiga

Forma de realizar as árvores:
Acrescentar +1 cada vez que passar por um nó, que está presente na transação que está sendo analisada na vez. Caso o “caminho” da transação não exista, acrescentar outro nó

Árvores geradas - 4, 5

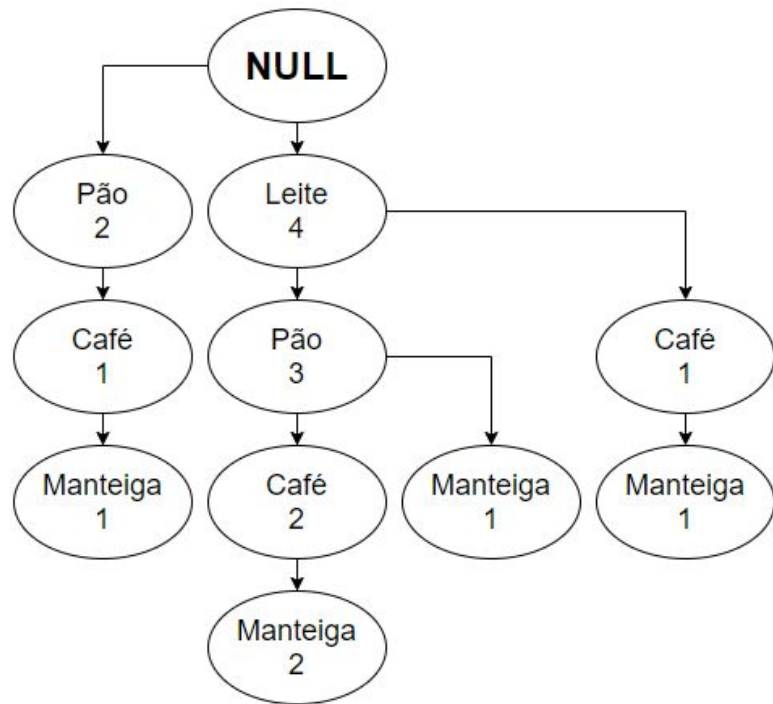


T4 > Leite, Pão, Café, Manteiga

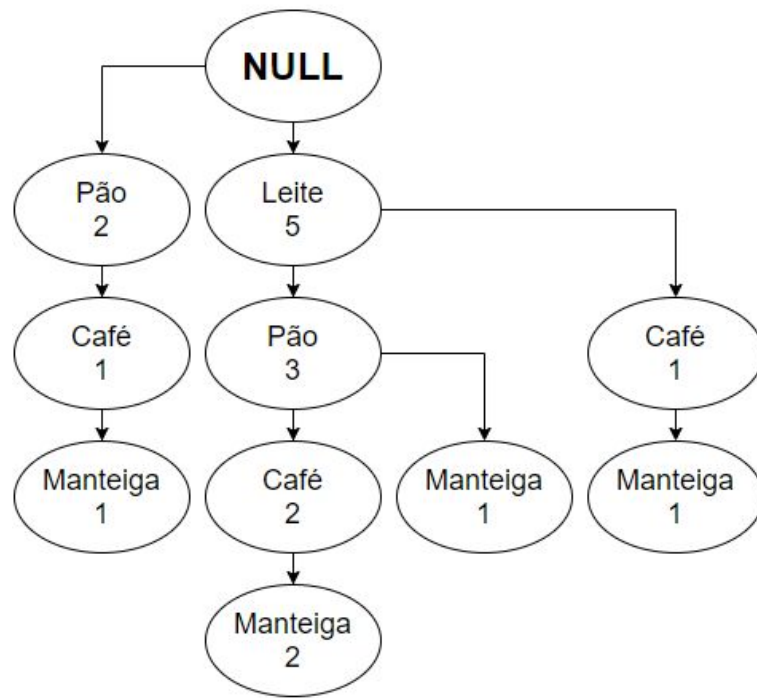


T6 > Leite, Café, Manteiga

Árvores geradas - 6, 7

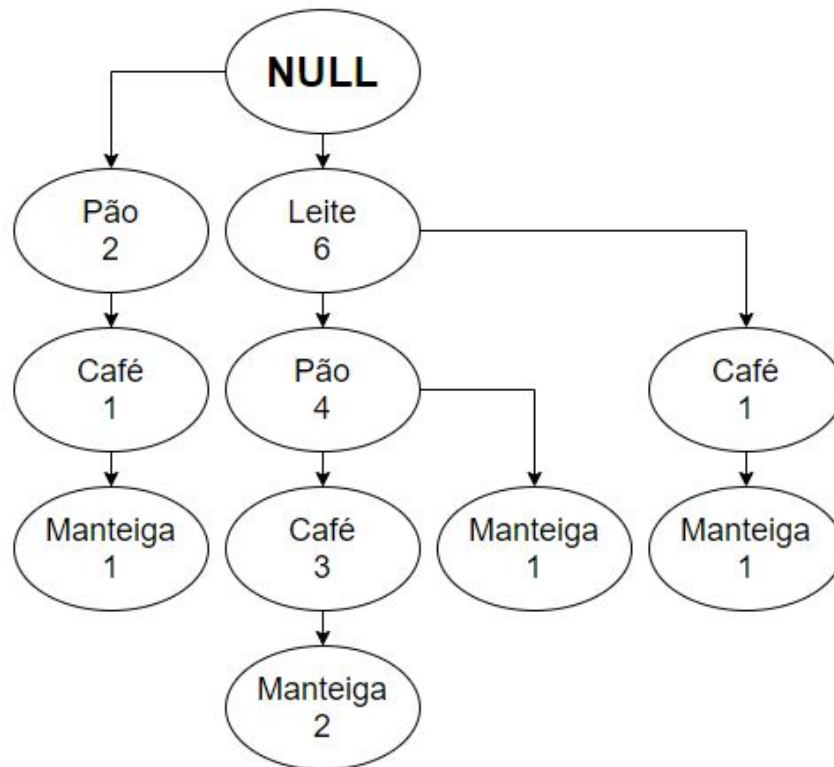


T7 > Pão



T8 > Leite

Árvores geradas - 8



T10 > Leite, Pão, Café

Frequência inversa x Conditional Pattern

- Baseado na árvore obtida, obtemos os conjuntos intermediários conditional pattern base e conditional FP tree, para depois gerar os conjuntos frequentes.

Frequência inversa	Conditional Pattern
Manteiga	{Pão, Café : 1}, {Leite, Pão, Café : 2}, {Leite, Pão : 1}, {Leite, Café : 1}
Café	{Pão : 1}, {Leite, Pão : 3}, {Leite: 1}
Pão	{Leite: 4}
Leite	-

Conditional pattern base x conditional FP tree

Conditional pattern base	Conditional FP tree
{Pão, Café : 1}, {Leite, Pão, Café : 2}, {Leite, Pão : 1}, {Leite, Café : 1}	{Pão, Café, Leite :4}
{Pão : 1}, {Leite, Pão : 3}, {Leite: 1}	{Pão, Leite :4}
{Leite: 4}	{Leite :4}
-	-

Conjuntos frequentes

Frequência inversa	Conditional FP Tree	Conjuntos Frequentes
Manteiga	[Pão, Café, Leite :4]	{Pão, Manteiga}{Café, Manteiga}{Leite, Manteiga}{Pão, Manteiga, Café} {Pão, Manteiga, Leite} {Café, Manteiga, Leite}{Café, Manteiga, Leite, Pão}
Café	[Pão, Leite :4]	{Leite, Café}{Pão, Café}{Leite, Pão, Café}
Pão	[Leite :4]	{Leite, Pão}
Leite	-	-

Confiança

- Pão -> Manteiga	= $4/6 = 0.66$
- Café -> Manteiga	= $4/5 = 0.8$
- Leite -> Manteiga	= $4/6 = 0.66$
- Pão -> Manteiga, Café	= $3/6 = 0.5$
- Pão -> Manteiga, Leite	= $2/6 = 0.33$
- Café -> Manteiga, Leite	= $3/5 = 0.6$
- Café -> Manteiga, Leite, Pão	= $2/5 = 0.4$
- Leite -> Café	= $4/6 = 0.66$
- Pão -> Café	= $4/6 = 0.66$
- Leite -> Pão, Café	= $3/6 = 0.5$
- Leite -> Pão	= $4/6 = 0.66$
- Café -> Pão, Manteiga	= $3/5 = 0.33$
- Leite -> Manteiga, Pão	= $2/6 = 0.33$
- Leite -> Manteiga, Café	= $3/6 = 0.5$
- Pão -> Café, Manteiga, Leite	= $2/6 = 0.33$
- Leite -> Pão, Café, Manteiga	= $2/6 = 0.33$
- Pão, Manteiga -> Café	= $3/4 = 0.75$
- Pão, Manteiga -> Leite	= $2/4 = 0.5$
- Café, Manteiga -> Leite	= $3/4 = 0.75$
- Café, Manteiga -> Leite, Pão	= $2/4 = 0.5$
- Leite, Café -> Manteiga, Pão	= $2/4 = 0.5$

- Manteiga -> Pão	= $4/5 = 0.8$
- Manteiga -> Café	= $4/5 = 0.8$
- Manteiga -> Leite	= $4/5 = 0.8$
- Manteiga -> Café, Pão	= $3/5 = 0.6$
- Manteiga -> Leite, Pão	= $2/5 = 0.4$
- Manteiga -> Leite, Café	= $3/5 = 0.6$
- Manteiga -> Leite, Pão, Café	= $2/5 = 0.4$
- Café -> Leite	= $4/5 = 0.8$
- Café -> Pão	= $4/5 = 0.8$
- Café -> Leite, Pão	= $3/5 = 0.6$
- Pão -> Leite	= $4/6 = 0.66$
- Manteiga, Café -> Pão	= $3/4 = 0.75$
- Manteiga, Leite -> Pão	= $2/4 = 0.5$
- Manteiga, Leite -> Café	= $3/4 = 0.75$
- Manteiga, Leite -> Pão, Café	= $2/4 = 0.5$
- Café, Leite -> Pão	= $3/4 = 0.75$
- Pão, Café -> Manteiga	= $3/4 = 0.75$
- Leite, Pão -> Manteiga	= $2/4 = 0.5$
- Leite, Café -> Manteiga	= $3/4 = 0.75$
- Manteiga, Leite -> Pão, Café	= $2/4 = 0.5$
- Café, Manteiga -> Leite, Pão	= $2/4 = 0.5$
- Pão, Manteiga -> Leite, Café	= $2/4 = 0.5$
- Pão, Leite -> Manteiga, Café	= $2/4 = 0.5$
- Pão, Café -> Manteiga, Leite	= $2/4 = 0.5$

Regras

1. Manteiga \Rightarrow Pão = 0.8
2. Café \Rightarrow Pão = 0.8
3. Manteiga \Rightarrow Leite = 0.8
4. Café \Rightarrow Leite = 0.8
5. Manteiga \Rightarrow Café = 0.8
6. Café \Rightarrow Manteiga = 0.8

Vantagens

- Só escaneia o database 2 vezes, comparado ao Apriori
- O database é armazenado de forma compacta
- É eficiente e escalável para minerar grandes ou pequenos padrões frequentes

Desvantagens

- Mais pesada e de difícil construção que o Apriori
- Pode ter alto custo
- Quando a database é grande, o algoritmo pode não caber na memória compartilhada

Weka FP Growth

The screenshot displays the Weka Explorer interface with the 'Associate' tab selected. The 'Associator' section shows the configuration for the 'FPGrowth' algorithm. The 'Result list' on the left shows a series of 'FPGrowth' operations. The 'Associator output' pane displays the following information:

```
=== Run information ===  
Scheme:      weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.8 -D 0.05 -U 1.0 -M 0.3  
Relation:    PãoeManteiga Sim  
Instances:    10  
Attributes:   7  
             Leite  
             Café  
             Cerveja  
             Pão  
             Manteiga  
             Arroz  
             Feijão  
=== Associator model (full training set) ===  
  
FPGrowth found 6 rules (displaying top 6)  
  
1. [Manteiga=Sim]: 5 ==> [Pão=Sim]: 4 <conf:(0.8)> lift:(1.33) lev:(0.1) conv:(1)  
2. [Café=Sim]: 5 ==> [Pão=Sim]: 4 <conf:(0.8)> lift:(1.33) lev:(0.1) conv:(1)  
3. [Manteiga=Sim]: 5 ==> [Leite=Sim]: 4 <conf:(0.8)> lift:(1.33) lev:(0.1) conv:(1)  
4. [Café=Sim]: 5 ==> [Leite=Sim]: 4 <conf:(0.8)> lift:(1.33) lev:(0.1) conv:(1)  
5. [Manteiga=Sim]: 5 ==> [Café=Sim]: 4 <conf:(0.8)> lift:(1.6) lev:(0.15) conv:(1.25)  
6. [Café=Sim]: 5 ==> [Manteiga=Sim]: 4 <conf:(0.8)> lift:(1.6) lev:(0.15) conv:(1.25)
```

The 'Status' bar at the bottom shows 'OK'.

The 'weka.gui.GenericObjectEditor' window is open, showing the 'About' dialog for 'weka.associations.FPGrowth'. The 'About' text states: 'Class implementing the FP-growth algorithm for finding large item sets without candidate generation.' The 'More' button is highlighted. The 'Capabilities' button is also visible.

The 'weka.gui.GenericObjectEditor' window also displays the following configuration parameters:

- delta: 0.05
- doNotCheckCapabilities: False
- findAllRulesForSupportLevel: False
- lowerBoundMinSupport: 0.3
- maxNumberOfItems: -1
- metricType: Confidence
- minMetric: 0.8
- numRulesToFind: 10
- positiveIndex: 2
- rulesMustContain:
- transactionsMustContain:
- upperBoundMinSupport: 1.0

The 'Open...', 'Save...', 'OK', and 'Cancel' buttons are visible at the bottom of the 'GenericObjectEditor' window.

Weka Apriori

The screenshot shows the Weka Explorer interface with the Apriori algorithm selected. The 'Associator output' pane displays the results of the algorithm, including the minimum support, minimum metric, and the generated sets of large itemsets. The 'Result list (right-click f...)' pane shows a list of results, with the Apriori result selected. The 'Status' pane shows 'OK'.

Associator output

```
==== Associator model (full training set) ====

Apriori
=====

Minimum support: 0.3 (3 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4
Size of set of large itemsets L(2): 6
Size of set of large itemsets L(3): 4

Best rules found:

1. Café=Sim 5 ==> Leite=Sim 4 <conf:(0.8)> lift:(1.33) lev:(0.1) [1] conv:(1)
2. Manteiga=Sim 5 ==> Leite=Sim 4 <conf:(0.8)> lift:(1.33) lev:(0.1) [1] conv:(1)
3. Café=Sim 5 ==> Pão=Sim 4 <conf:(0.8)> lift:(1.33) lev:(0.1) [1] conv:(1)
4. Manteiga=Sim 5 ==> Café=Sim 4 <conf:(0.8)> lift:(1.6) lev:(0.15) [1] conv:(1.25)
5. Café=Sim 5 ==> Manteiga=Sim 4 <conf:(0.8)> lift:(1.6) lev:(0.15) [1] conv:(1.25)
6. Manteiga=Sim 5 ==> Pão=Sim 4 <conf:(0.8)> lift:(1.33) lev:(0.1) [1] conv:(1)
```

GenericObjectEditor

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm. More Capabilities

car False

classIndex -1

delta 0.05

doNotCheckCapabilities False

lowerBoundMinSupport 0.3

metricType Confidence

minMetric 0.8

numRules 10

outputItemSets False

removeAllMissingCols False

significanceLevel -1.0

treatZeroAsMissing False

upperBoundMinSupport 1.0

Open... Save... OK Cancel

Ao rodarmos o Apriori, os mesmos resultados foram encontrados.

Referências

<https://medium.com/@abnersuniga7/encontre-padr%C3%B5es-nos-seus-dados-com-apriori-e-fp-growth-4a581ec1b22>

[https://towardsdatascience.com/understand-and-build-fp-growth-algorithm-in-python-d8b989bab342#:~:text=What%20is%20FP%2D%20Growth,\(AKA%20Association%20Rule%20Mining\).&text=The%20Apriori%20Algorithm%20produces%20frequent,threshold%20%E2%80%9Cminimal%20support%20count%E2%80%9D.](https://towardsdatascience.com/understand-and-build-fp-growth-algorithm-in-python-d8b989bab342#:~:text=What%20is%20FP%2D%20Growth,(AKA%20Association%20Rule%20Mining).&text=The%20Apriori%20Algorithm%20produces%20frequent,threshold%20%E2%80%9Cminimal%20support%20count%E2%80%9D.)

<https://www.youtube.com/user/jacaoschreiber>

<https://www.softwaretestinghelp.com/fp-growth-algorithm-data-mining/>