

Regras de Associação e itens frequentes

Tópicos de Inteligência Artificial

Prof. Dr. [Dieval Guizelini](#)

Fevereiro/2023

Algumas referências históricas

Associação entre as vendas de fraldas e cervejas

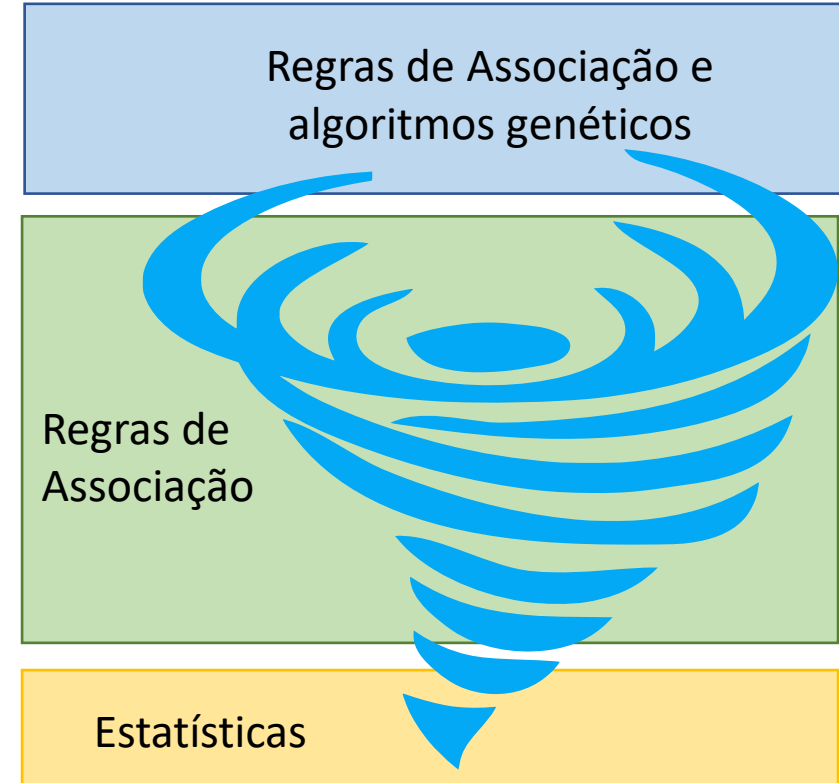


Caso da rede varejista Target



- 87% acerto
- Valor da empresa cresceu de 40 para 64 bi.

Indicação de produtos



Indicação de produtos

Retornar aos resultados



O Algoritmo Mestre: Como a Busca Pelo Algoritmo de Machine Learning Definitivo Recriará Nosso Mundo (Português) Capa comum – 24 janeiro 2017

por [Pedro Domingos](#) (Autor)
★★★★★ 56 classificações

[Ver todos os formatos e edições](#)

Kindle
R\$57,19

Leia com nossos [apps gratuitos](#)

Capa Comum
R\$60,20

1 Usado a partir de R\$62,30
4 Novo a partir de R\$45,00



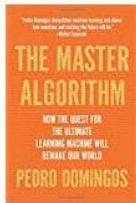
Clientes que visualizaram este item também visualizaram



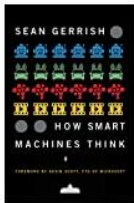
A Revolução do Algoritmo Mestre Como a aprendizagem automática está a mudar o mundo
Pedro Domingos
Capa comum



Aprendizado profundo para leigos
John Paul Mueller
★★★★★ 7
Capa comum
R\$ 45,90



The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake...
Pedro Domingos
★★★★★ 592
Capa comum



How Smart Machines Think
Sean Gerrish
★★★★★ 39
Capa dura
R\$ 217,34



Algoritmos de Destruição em Massa
Cathy O'Neil
★★★★★ 58
Capa comum
R\$ 46,75



Vida 3.0: O ser humano na era da inteligência artificial
Max Tegmark
★★★★★ 37
Capa comum
R\$ 37,70



Big Data: O futuro dos dados e aplicações
Felipe Nery Rodrigues
★★★★★ 17
Capa comum
R\$ 51,89

Frequentemente comprados juntos



+



+



Preço total
[Adicionar](#)

Indicação de produtos



Compre junto



Trocar Produto



Trocar Produto



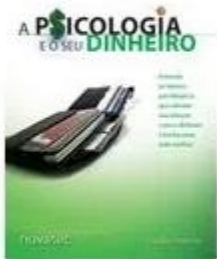
A HORA DA ESSÊNCIA



Quebra-Cabeça Amsterdam 500
peças - Pais&Filhos



Produtos visitados por quem procura este item



Psicologia E O Seu
Dinheiro, A - Novatec



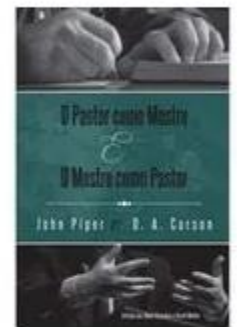
Mestre, O



Mestre da Guerra



MESTRE E O APRENDIZ



O PASTOR COMO MESTRE
E O MESTRE COMO



Indicação de produtos

americanas · mais



aproveite e veja também



Fritadeira Elétrica Air Fryer 4L
1500W Multilaser Vermelha ...



R\$ 349,00
12x de 29,08 s/juros

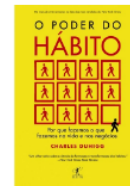


↓ baixou

Colchonete Massageador 10
Motores, Magnético,...



~~R\$ 659,00~~
R\$ 632,64
em \$1x no cartão de crédito



Livro - O poder do hábito



R\$ 31,90
3x de 10,63 s/juros



Livro - Proteção de Dados
Pessoais - A Função e os...

R\$ 109,00
11x de 9,90 s/juros



Livro - Os Inovadores: Uma
Biografia da Revolução Digital



R\$ 19,90
2x de 9,95 s/juros

quem viu os mesmos produtos que você, também viu:



↓ 10%

Fritadeira Air Fry Britânia
BFR10VI Inox Vermelha 3,5L



~~R\$ 629,90~~
R\$ 566,91
em \$1x no cartão de crédito



Livro - CRIE SEU MERCADO
NO MUNDO DIGITAL...



R\$ 32,99
3x de 10,99 s/juros



Kit Oster - Fritadeira Elétrica
Sem Óleo Air Fryer +...



R\$ 799,99
12x de 66,66 s/juros



↓ 10%

Fritadeira Elétrica sem Óleo
Air Fry Multilaser Gourmet 4...



~~R\$ 359,90~~
R\$ 323,37 **à-mais**
em \$1x no cartão de crédito



Livro - Padrões JavaScript:
Construa Aplicações mais...



R\$ 47,99
4x de 11,99 s/juros

Regras de associação

Exemplos:

Quem compra cerveja, também compra fraldas.

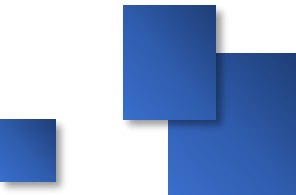
Quem compra pão, também compra leite.

Quem compra queijo, também compra presunto.

Quem compra presunto, também compra queijo.

Quem compra Coca-Cola, também compra sonho de valsa.

Quem votou a favor de religião nas escolas, votou a favor do "país X".



Termologia

- Uma **regra de associação** é uma implicação da forma $X \rightarrow Y$, onde X e Y são conjuntos de itens tais que $X \cap Y = \emptyset$
- **Transação** é denominado o elemento de ligação existente em cada ocorrência de um subconjunto de itens no banco de dados.
- Uma associação é considerada **frequente** se o número de vezes em que a união de conjuntos de itens ($X \cup Y$) ocorrer em relação ao número total de transações do banco de dados for superior a uma frequência mínima (denominada **suporte mínimo**).
- Uma associação é considerada *válida* se o número de vezes em que $X \cup Y$ ocorrer em relação da ocorrência de X for superior a um valor denominado **confiança mínima** (qualidade da regra).
- Denomina-se **k-itemset** os subconjuntos de itens formados exatamente por k elementos originalmente pertencentes ao conjunto de origem.

Como avaliar uma regra de associação?

- Regra:
a compra de um computador implica na compra de programas de antivírus.

Computador => antivírus (suporte=2%, confiança: 60%)

- O “suporte” e a “confiança” são métricas de “interesse” (interestingness) que representam, respectivamente, a utilidade e a confiabilidade da regra descoberta.
- Um suporte de 2% representa que 2% de todas as “transações de compras” analisadas indicam que computadores e antivírus foram comprados juntos.
- A confiança indica que em 60% das compras onde os clientes compraram computadores, também compraram o antivírus.

Interesse?

Os sistemas de busca de regras de associação tendem a produzir muitas regras e a maioria não será significativa ou relevante.

Por isso, normalmente, as regras de associação são consideradas de interesse, quando elas satisfazem os parâmetros de “suporte mínimo” e “confiança mínima”.

Definição 1

Seja $I = \{I_1, I_2, \dots, I_m\}$ um conjunto de m itens.

Seja D um conjunto de dados transacionais onde cada transação T é um conjunto de itens, tal que $T \subseteq I$. Cada transação possui um identificador Tid . Seja A um subconjunto de itens é dito que contém A , se e somente se, $A \subseteq T$.

Uma regra de associação é uma implicação da forma $A \Rightarrow B$, $A \subset I$, $B \subset I$ e $A \cap B = \emptyset$

A regra $A \Rightarrow B$ vale no conjunto de transações D com **suporte** s , onde s é a porcentagem de transações em D que contém $A \cup B$.

A regra $A \Rightarrow B$ tem **confiança** c no conjunto de transações D , onde c é a porcentagem de transações em D contendo A que também contém B .

Quando uma regra satisfaz o suporte mínimo e a confiança mínima é chamada de **forte**.

Definição 1

Um conjunto de itens é chamado de **itemset**. Um itemset que contém k itens é um conjunto de k -itemset.

No exemplo anterior o conjunto {computador, antivírus} é um conjunto de 2-itemset.

A frequência de ocorrência de um itemset é o número de transações que contém o itemset. Isto também é conhecido como **frequência**, **suporte** ou **contagem** de um itemset.

Se um itemset I satisfaz um suporte mínimo ele é dito um “itemset frequente”. Um conjunto de k -itemsets frequente é chamado de L_k .

Matematicamente

$$\textit{suporte}(A) = \frac{\textit{ocorr\^encia}(A)}{\textit{N\^umero de registros/casos}}$$

$$\textit{suporte}(A \cup B) = \frac{\textit{ocorr\^encia}(A \cup B)}{\textit{N\^umero de registros/casos}}$$

$$\textit{confian\c{a}} = \frac{\textit{suporte}(A \cup B)}{\textit{suporte}(A)}$$

Uma simplificação inicial

Um subconjunto de $k+1$ -itemset será frequente, se e somente se, todos os subconjuntos de k -itemset forem frequentes.



Itemset fechado

O principal desafio de mineração de itemsets frequentes em grandes bases de dados é que, frequentemente, esse processo gera um numero muito grande de itemsets frequentes. Isto acontece porque, se um itemset é frequente, cada um de seus subconjuntos também o é. Para superar esta dificuldade, introduz-se o conceito de itemsets frequentes fechados ou maximal itemset frequente.

Definição:

- Um itemset X é fechado em um conjunto de dados S , se não existir nenhum super-itemset próprio Y ($X \subset Y$) tal que Y tenha o mesmo suporte que X .
- Um itemset X é um itemset frequente fechado no conjunto de dados S se X é tanto fechado quanto frequente em S .
- Um itemset X é um maximal itemset frequente (ou max-itemset) no conjunto de dados S se X é frequente, e não existe um super-itemset Y tal que $X \subset Y$ e Y é frequente em S .

Algoritmo *apriori*

Proposto por R.Agrawal e R. Srikant, em 1994.

O algoritmo pode ser descrito em dois passos:

- O conjunto de 1-itemsets frequentes é encontrado por meio da varredura do banco de dados para contagem de cada item e da descoberta daqueles itens que satisfazem um suporte mínimo. O resultado é chamado de L_1 .
- L_1 é usado para encontrar L_2 , o conjunto de 2-itemsets frequentes, o qual é usado para encontrar L_3 e assim por diante, até que nenhum k -itemset frequente possa ser encontrado. Encontrar L_k , requer uma leitura completa do banco de dados.
- Propriedade do apriori: Todos os subconjuntos não vazios de um itemset frequente deve também ser frequente.

Primeiro exemplo

- Considere um conjunto com 9 transações: $|D|=9$
Vamos aplicar o apriori para encontrar os itemset frequentes em D que atenda o suporte mínimo de 2 ($\text{min_sup}=2$).

Transação	Lista de itens
1	Item-1, Item-2, Item-5
2	Item-2,Item-4
3	Item-2,Item-3
4	Item-1, Item-2, Item-4
5	Item-1,Item-3
6	Item-2,Item-3
7	Item-1,Item-3
8	Item-1,Item-2,Item-3,Item-5
9	Item-1,Item-2,Item-3

Etapas para encontrar o k-itemset

Varre o banco de dados (D) e faz a contagem de cada item candidato.



itemset	Ocorrência
{item-1}	6
{item-2}	7
{item-3}	6
{item-4}	2
{item-5}	2

C_1

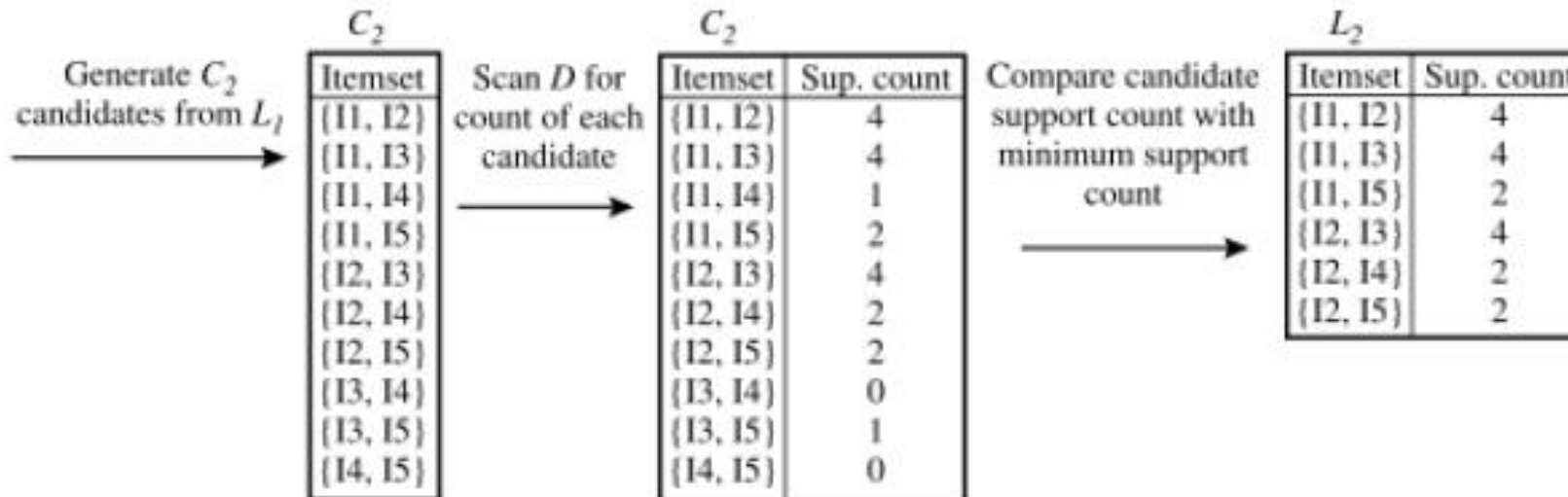
Filtra os itens que atende a frequência mínima (suporte=2).

Sup.min ≥ 2



itemset	Ocorrência
{item-1}	6
{item-2}	7
{item-3}	6
{item-4}	2
{item-5}	2

L_1



Etapas para encontrar o k-itemset

Geramos
candidatos
C2 a partir
de L1.



itemset
{item-1,item-2}
{item-1,item-3}
{item-1,item-4}
{item-1,item-5}
{item-2,item-3}
{item-2,item-4}
{item-2,item-5}
{item-3,item-4}
{item-3,item-5}
{item-4,item-5}

Varre o banco
de dados D
para contar
cada
candidato



itemset	Ocorrência
{item-1,item-2}	4
{item-1,item-3}	4
{item-1,item-4}	1
{item-1,item-5}	2
{item-2,item-3}	4
{item-2,item-4}	2
{item-2,item-5}	2
{item-3,item-4}	0
{item-3,item-5}	1
{item-4,item-5}	0

Filtra os itens
que atende a
frequência
mínima
(suporte=2).

Sup.min ≥ 2



itemset	f(x)
{item-1,item-2}	4
{item-1,item-3}	4
{item-1,item-5}	2
{item-2,item-3}	4
{item-2,item-4}	2
{item-2,item-5}	2

L₂

Etapas para encontrar o k-itemset

Geramos candidatos C_3 a partir de L_2 .



itemset
{lt-1,lt-2,lt-3}
{lt-1,lt-2,lt-5}

Varre o banco de dados D para contar cada candidato



itemset	Ocorrência
{lt-1,lt-2,lt-3}	2
{lt-1,lt-2,lt-5}	2

Filtra os itens que atende a frequência mínima (suporte=2).

Sup.min ≥ 2



L_3

itemset	Ocorrência
{lt-1,lt-2,lt-3}	2
{lt-1,lt-2,lt-5}	2

Algoritmo *apriori*

```
1)  $L_1 = \{\text{large 1-itemsets}\};$ 
2) for (  $k = 2; L_{k-1} \neq \emptyset; k++$  ) do begin
3)    $C_k = \text{apriori-gen}(L_{k-1});$  // New candidates
4)   forall transactions  $t \in \mathcal{D}$  do begin
5)      $C_t = \text{subset}(C_k, t);$  // Candidates contained in  $t$ 
6)     forall candidates  $c \in C_t$  do
7)        $c.\text{count}++;$ 
8)   end
9)    $L_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\}$ 
10) end
11)  $\text{Answer} = \bigcup_k L_k;$ 
```

Figure 1: Algorithm Apriori

Um exemplo prático

Considere as transações abaixo em um dado período

Nº Transação	Produto
1	A
1	B
1	C
2	A
2	C
3	A
4	C
4	B
4	D
4	A
5	A

Nº Transação	Produto
5	D
6	B
7	A
8	A
8	C
8	D
9	B
9	D
10	A
10	B
10	D

Obs: representação dos fatos no formato Basket (este modelo é aplicado em situações que o número de itens não possam ser limitados pelo número de atributos de uma tabela ou visão).

Um exemplo prático: 1º passo

A partir dos dados é possível:

- Definir o conjunto $R=\{A,B,C,D\}$ e
- A relação r que terá o modelo dessa tabela.

Nº Transação	A	B	C	D
1	1	1	1	0
2	1	0	1	0
3	1	0	0	0
4	1	1	1	1
5	1	0	0	1
6	0	1	0	0
7	1	0	0	0
8	1	0	1	1
9	0	1	0	1
10	1	1	0	1

Frequência dos subconjuntos unitários (1-itemset)

- Considerando os valores de $\sigma=0.1$ e $\Theta=0.8$.
- σ = suporte mínimo
- Θ = confiança mínima
- Exemplo para os nossos dados:
 - $\sigma(\{A\}) = 0,8$ e $\sigma(\{C\}) = 0,4$
 - $\Theta(\{A\}) = \text{conf}(X \Rightarrow Y) = \sigma(X \cup Y) / \sigma(X) = 0,8/0,8 = 1$. (confiança de 100%).

Conjunto	Ocorrência / Frequência
{A}	8 / 0,8
{B}	5 / 0,5
{C}	4 / 0,4
{D}	5 / 0,5

Freqüência nos subconjuntos binários (2-itemset)

- Considerando os valores de $\sigma=0.1$ e $\Theta=0.8$.
- σ = suporte mínimo
- Θ = confiança mínima
- Calcule os valores de suporte e confiança para cada subconjunto.

Conjunto	Freqüência
{A,B}	3
{A,C}	4
{A,D}	4
{B,C}	2
{B,D}	3
{C,D}	2

Quadro das regras potenciais

Para os parâmetros $\sigma=0.1$ $\Theta=0,8$

Regra	Confiança
$A \rightarrow B$	3/8
$A \rightarrow C$	4/8
$A \rightarrow D$	4/8
$B \rightarrow C$	2/5
$B \rightarrow D$	3/5
$C \rightarrow D$	2/4

Regra	Confiança
$B \rightarrow A$	3/5
$C \rightarrow A$	4/4
$C \rightarrow B$	2/4
$D \rightarrow A$	4/5
$D \rightarrow B$	3/5
$D \rightarrow C$	2/5

Associações consideradas válidas: $C \rightarrow A$ e $D \rightarrow A$

Regra:

Sendo $C \rightarrow A$ com 100% de confiança e $D \rightarrow A$ com confiança de 80%.

Frequência dos subconjuntos de ternários (3-itemset)

Conjunto	Freqüência
{A,B,C}	2
{A,C,D}	2
{A,B,D}	2
{B,C,D}	1

Quadro das regras potenciais

Para os parâmetros $\sigma=0.1$ $\Theta=0,8$

Regra	Confiança
$A, B \rightarrow C$	$2/3$
$A, C \rightarrow B$	$2/4$
$A, B \rightarrow D$	$2/3$
$A, D \rightarrow B$	$2/4$
$A, C \rightarrow D$	$2/4$
$A, D \rightarrow C$	$2/4$

Regra	Confiança
$B, C \rightarrow D$	$1/2$
$B, D \rightarrow C$	$1/3$
$B, C \rightarrow A$	$2/2$
$B, D \rightarrow A$	$2/3$
$C, D \rightarrow A$	$2/2$
$C, D \rightarrow B$	$1/2$

Associações consideradas válidas: $B, C \rightarrow A$ e $C, D \rightarrow A$

Regra: Na ocorrência dos itens B e C, implicam a ocorrência de A

Na ocorrência dos itens C e D, implicam na ocorrência de A

Quadro das regras potenciais

Regra	Confiança
$A, B, C \rightarrow D$	$\frac{1}{2}$
$A, C, D \rightarrow B$	$\frac{1}{2}$
$A, B, D \rightarrow C$	$\frac{1}{2}$
$B, C, D \rightarrow A$	$1/1$

Regra: Com 100% de confiança, se ocorrer os produtos B,C e D, então ocorre produto A.

A correlação entre os itens

Lift é uma medida de correlação simples.

A ocorrência de um itemset A é independente da ocorrência de um itemset B se $P(A \cup B) = P(A)P(B)$; caso contrário, os itemsets A e B são dependentes ou correlatos (ou correlacionados) como eventos.

Logo:

$$lift(A, B) = \frac{conf(A \Rightarrow B)}{suporte(B)} = \frac{suporte(AB)}{suporte(A) \cdot suporte(B)}$$

Se o valor resultante é menor do que 1, então a ocorrência de A é negativamente correlacionada com a ocorrência de B.

Se o valor resultante é maior do que 1, então A e B são positivamente correlacionados, significando que a ocorrência de um implica na ocorrência de outro. Se o resultado é igual a 1, então A e B são independentes e não há correlação entre eles.

Medidas

$$suporte = \frac{|X \cup Y|}{n}$$

$$confiança(X \rightarrow Y) = \frac{suporte(X \cup Y)}{suporte(X)}$$

$$lift(X \Rightarrow Y) = \frac{suporte(X \cup Y)}{suporte(X) \times suporte(Y)}$$

$$convicção(X \Rightarrow Y) = \frac{1 - suporte(Y)}{1 - confiança(X \Rightarrow Y)}$$

Outros algoritmos

- Eclat
 - DFS (depth-first-search)
 - Consome menos memória que o apriori
 - Exemplo: <https://rdrr.io/cran/arules/man/eclat.html>
- fp-grow
 - Utilizado no apache Spark
 - <https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html>
- partition

Em Python

- Apyori
<https://pypi.org/project/apyori/>
- Efficient-Apriori
<https://pypi.org/project/efficient-apriori/>
- Exemplo com Scikit-learn no kaggle
<https://www.kaggle.com/sangwookchn/association-rule-learning-with-scikit-learn>
- Mlxtend
http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/

No Google colab

```
!pip install apyori
```

```
import numpy as np
```

```
import pandas as pd
```

```
# Loading the Data
```

```
Data = pd.read_csv('market_basket.csv', header =  
None)
```

```
Data.head()
```

```
transacts = []
```

```
# populating a list of transactions
```

```
for ireg in range(0, 7501):
```

```
    reg = list(Data.values[ireg])
```

```
    r = []
```

```
    for campo in reg:
```

```
        if str(campo) != 'nan':
```

```
            r.append(campo)
```

```
    transacts.append(r)
```

No Google colab

```
from apyori import apriori
rule = apriori(transactions = transacts, min_support = 0.003, \
               min_confidence = 0.2, min_lift = 3, min_length = 2, max_length = 2)

results = list(rule) # returns a non-tabular output

print(results)

def inspect(output):
    lhs      = [tuple(result[2][0][0])[0] for result in output]
    rhs      = [tuple(result[2][0][1])[0] for result in output]
    support   = [result[1] for result in output]
    confidence = [result[2][0][2] for result in output]
    lift      = [result[2][0][3] for result in output]
    return list(zip(lhs, rhs, support, confidence, lift))

out = pd.DataFrame(inspect(results), columns = ['Left_Hand_Side', \
        'Right_Hand_Side', 'Support', 'Confidence', 'Lift'])
out.nlargest(n = 10, columns = 'Lift')
```

Preprocess Classify Cluster Associate Select attributes Visualize

Associator

Choose **Apriori -N 10000000 -T 0 -C 0.0 -D 0.05 -U 1.0 -M 0.0 -S -1.0 -c -1**

Start Stop

Result list (right-click...)

- 01:12:12 - Apriori
- 23:25:25 - Apriori
- 23:37:53 - Apriori**
- 23:50:17 - Apriori

Associator output

```

Scheme:      weka.associations.Apriori -N 10000000 -T 0 -C 0.0 -D 0.05 -U 1.0 -M 0.0 -S -1.0 -c -1
Relation:    hashtags.symbolic-weka.filters.unsupervised.attribute.StringToNominal-Rlast
Instances:   86
Attributes:  2
              k1
              hash

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

Apriori
=====

Minimum support: 0.01 (1 instances)
Minimum metric <confidence>: 0
Number of cycles performed: 20

Generated sets of large itemsets:

Size of set of large itemsets L(1): 84
Size of set of large itemsets L(2): 81

Best rules found:

1. hash=#flight420 2 ==> k1=smoke 2 <conf:(1)> lift:(1.91) lev:(0.01) [0] conv:(0.95)
2. hash=#inneedofsmokeinmylungs... 2 ==> k1=smoke 2 <conf:(1)> lift:(1.91) lev:(0.01) [0] conv:(0.95)
3. hash=#oomf 2 ==> k1=smoke 2 <conf:(1)> lift:(1.91) lev:(0.01) [0] conv:(0.95)
4. hash=#justsoyouknow 2 ==> k1=smoke 2 <conf:(1)> lift:(1.91) lev:(0.01) [0] conv:(0.95)
5. hash=#rachietweet 2 ==> k1=smoke 2 <conf:(1)> lift:(1.91) lev:(0.01) [0] conv:(0.95)
6. hash=#nykextra 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
7. hash=#listenlive 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
8. hash=#relax 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
9. hash=#americanizedversion 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
10. hash=#icant 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
11. hash=#facetime 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
12. hash=#cannabis. 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
13. hash=#airplane! 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
14. hash=#tmlt 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
15. hash=#seehowthatgoes? 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
16. hash=#addicted 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
17. hash=#doitnow! 1 ==> k1=smoking 1 <conf:(1)> lift:(5.73) lev:(0.01) [0] conv:(0.83)
18. hash=#wallets 1 ==> k1=tobacco 1 <conf:(1)> lift:(28.67) lev:(0.01) [0] conv:(0.97)

```

Status OK

Log x 0

Bases de dados

- Conjunto de dados de doenças cardíacas (derivado de 4 outros bancos)
<http://archive.ics.uci.edu/ml/datasets/heart+Disease>
- Derivado do banco USDA de plantas que ocorrem nos EUA e Canadá
<https://archive.ics.uci.edu/ml/datasets/Plants>
- Classificação de cogumelos em comestíveis ou venenosos
<https://archive.ics.uci.edu/ml/datasets/Mushroom>
- Questionário sobre informações do blogger realizado no Irã
<https://archive.ics.uci.edu/ml/datasets/BLOGGER>
- Conjunto de dados de usuários de Internet desonestos
<https://archive.ics.uci.edu/ml/datasets/Dishonest+Internet+users+Dataset>
- Análise de Cesta de Mercado Instacart
<https://www.kaggle.com/datatheque/association-rules-mining-market-basket-analysis>

Exemplo no R

- <https://www.datacamp.com/community/tutorials/market-basket-analysis-r>



Referências

- Caso Target
<https://www.traycorp.com.br/conteudo/target-e-o-comportamento-do-cliente/>
- How Companies Learn Your Secrets (Caso Target)
<https://www.nytimes.com/2012/02/19/magazine/shopping-habits.html?pagewanted=1&r=1&hp>
- SIQUEIRA, G. M. Algoritmos de Mineração de Dados Eficientes Quanto ao Consumo de Memória, dissertação de mestrado. UFMG, 2004. Disponível em: <https://repositorio.ufmg.br/bitstream/1843/SLBS-643J9H/1/gustavomenezessiqueira.pdf>