O que George Boole tem a ver com Ciência de Dados?

—

Pesquisando relações entre lógica e ciência de dados

Cássia Sampaio

Março 2019

Já ouviu falar em booleano?

Tipo de variável

```
>>> bool(1)
True
>>> bool(0)
False
>>> bool([])
False
>>> bool( (1,) )
True
```

PEP 285

Operadores lógicos

```
1 A = True
2 B = False
3 print ("Valor de A and B: ", A and B)
4
```

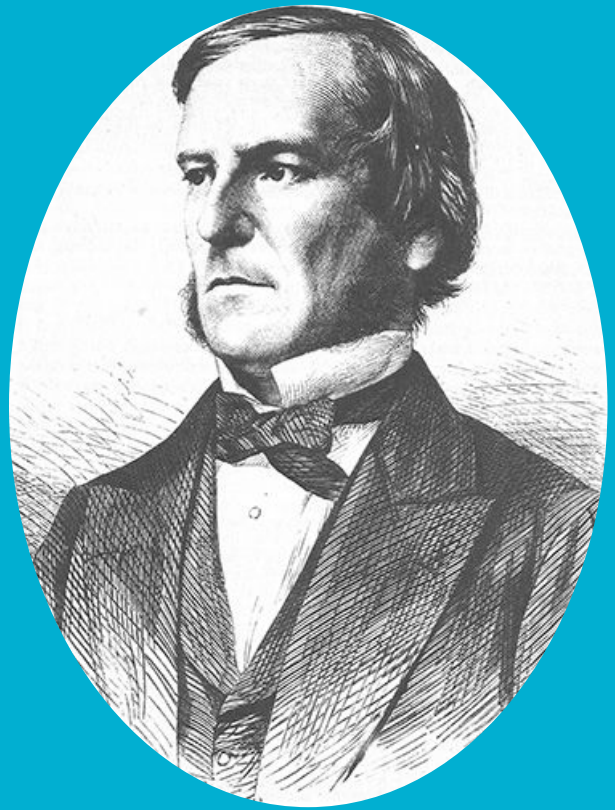
```
['Valor de A and B: ', False]
```

Expressões

```
1 a = 3
2 b = 4
3 c = a < b   # c recebe o valor da comparação a < b
4 d = a > b   # d recebe o valor da comparação a > b
5 e = a == b  # e recebe o valor da comparação a == b
6
7 print("Valor de c: ", c)
8 print("Valor de d: ", d)
9 print("Valor de e: ", e)
```

```
['Valor de c: ', True]
['Valor de d: ', False]
['Valor de e: ', False]
```

Georginho



Georginho e Data Science? Será?

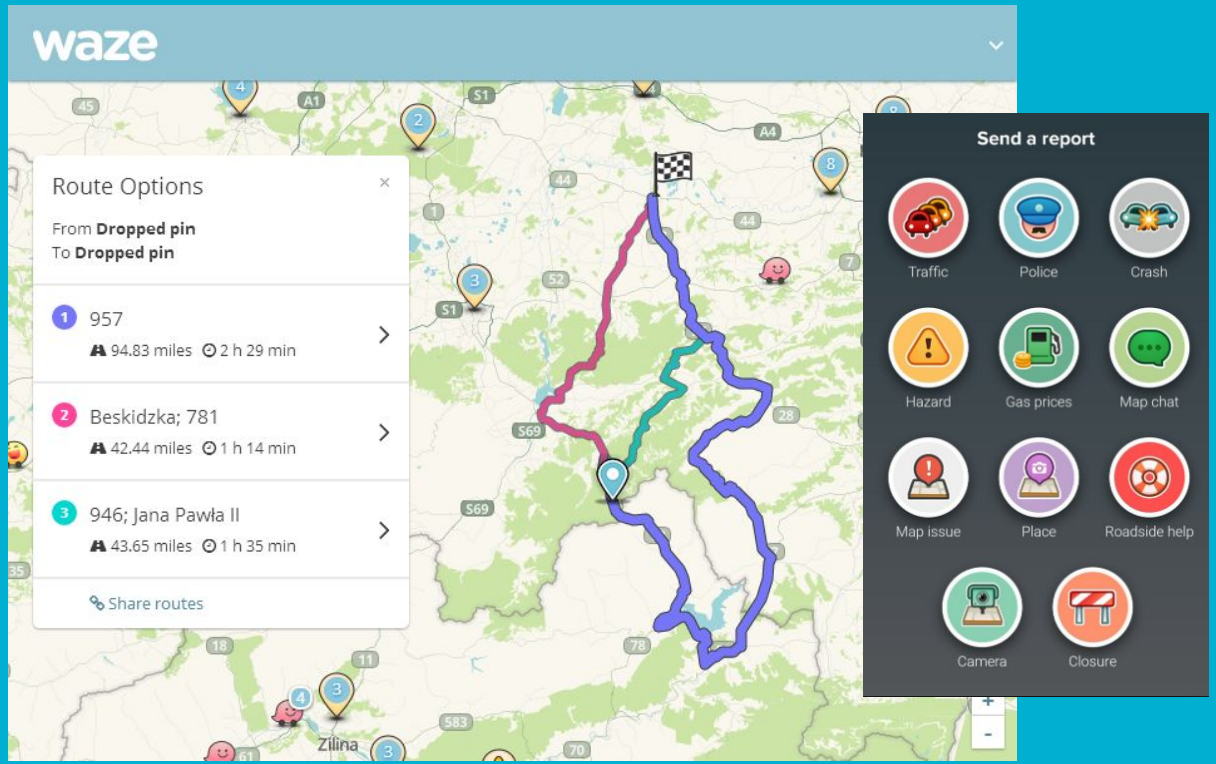
Me propus a entender.

Tem ligação?

Não tem ligação?

Pensa no Waze

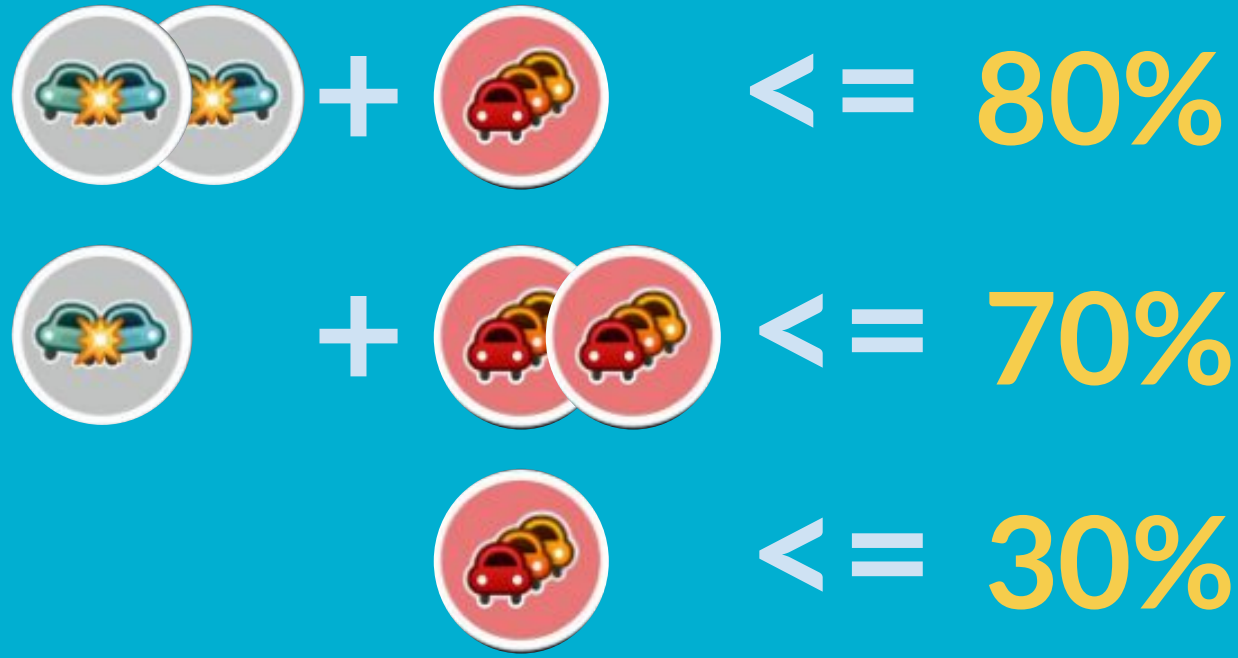
Estimativa por crowdsourcing



Estimativa por crowdsourcing

Estimativa por crowdsourcing

Estimativa por crowdsourcing



Estimativa por crowdsourcing

$$2a + 1t <= 80\%$$

$$1a + 2t <= 70\%$$

$$1t <= 30\%$$

Estimativa por crowdsourcing

$$2 + 1t <= 80\%$$

$$1 + 2t <= 70\%$$

$$1t <= 30\%$$

Estimativa por crowdsourcing

2 + 1 <= 80%

1 + 2 <= 70%

1 <= 30%

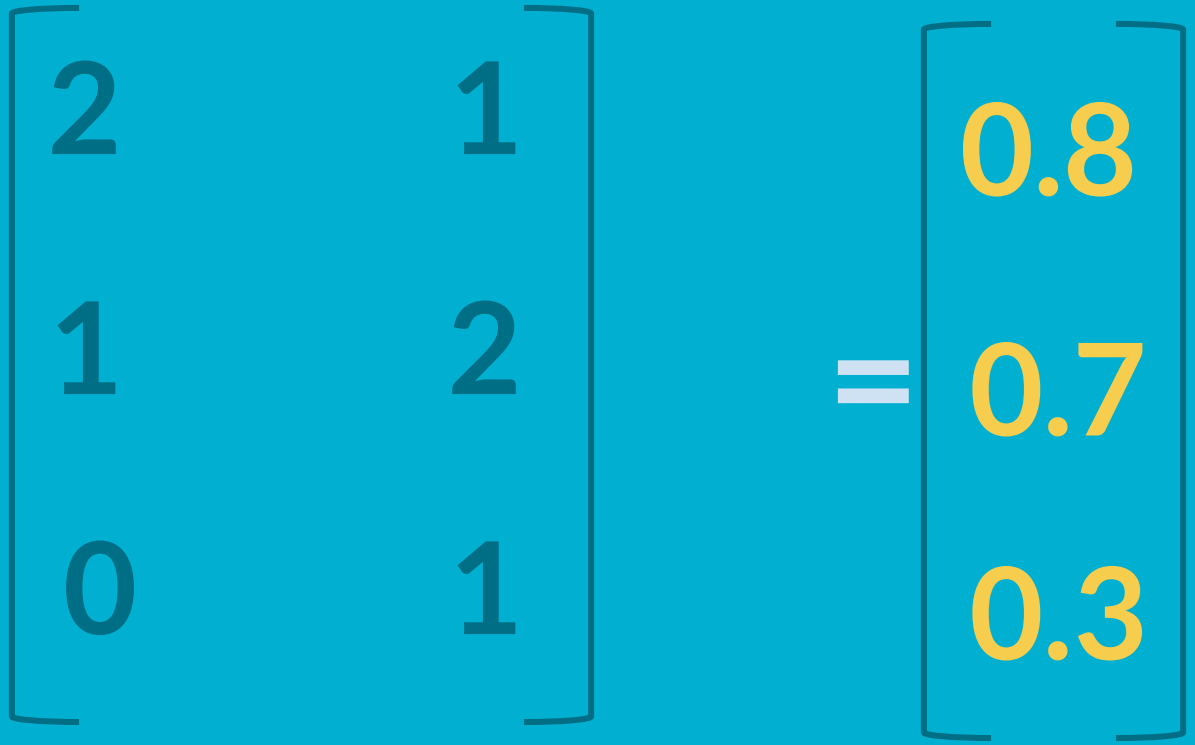
Estimativa por crowdsourcing

$$2 + 1 <= 0.8$$

$$1 + 2 <= 0.7$$

$$1 <= 0.3$$

Estimativa por crowdsourcing



Estimativa por crowdsourcing

Estimativa por crowdsourcing

1a + 1t = ?

Estimativa por crowdsourcing

Função objetiva

$$\begin{bmatrix} 1 & 1 \end{bmatrix} = ?$$

Scipy linprog

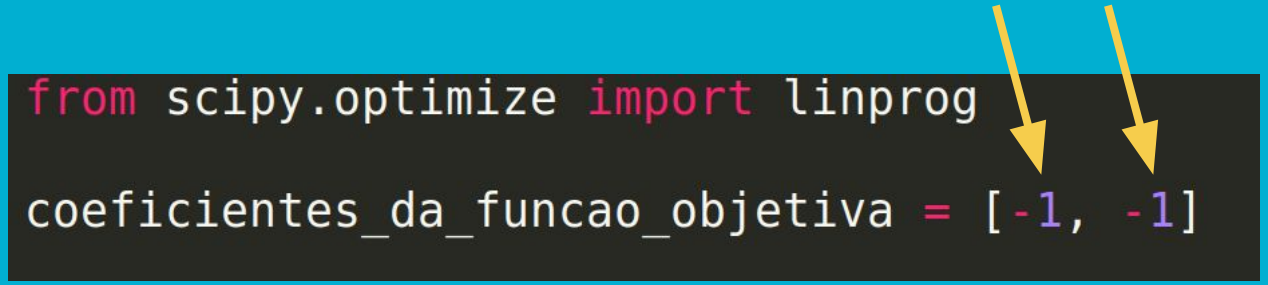
```
from scipy.optimize import linprog
```

Scipy linprog

```
from scipy.optimize import linprog

coeficientes_da_funcao_objetiva = [-1, -1]
```

Scipy linprog



Scipy linprog

```
from scipy.optimize import linprog

coeficientes_da_funcao_objetiva = [-1, -1]
```

```
coeficientes_das_inequacoes = [[2, 1],
                               [1, 2],
                               [0, 1]]
```

Scipy linprog

```
from scipy.optimize import linprog

coeficientes_da_funcao_objetiva = [-1, -1]
```

```
coeficientes_das_inequacoes = [[2, 1],
                               [1, 2],
                               [0, 1]]
```

```
valores_maximos = [0.8,
                   0.7,
                   0.3]
```

Scipy linprog

```
res = linprog(coeficientes_da_funcao_objetiva,
              coeficientes_das_inequacoes,
              valores_maximos,
              bounds=(valor_min_e_max_de_a, valor_min_e_max_de_t),
              method='simplex')
print(res)
```

Scipy linprog

```
from scipy.optimize import linprog

coeficientes_da_funcao_objetiva = [-1, -1]

coeficientes_das_inequacoes = [[2, 1],
                               [1, 2],
                               [0, 1]]

valores_maximos = [0.8,
                   0.7,
                   0.3]

valor_min_e_max_de_a = (0, None)
valor_min_e_max_de_t = (0, None)

res = linprog(coeficientes_da_funcao_objetiva,
              coeficientes_das_inequacoes,
              valores_maximos,
              bounds=(valor_min_e_max_de_a, valor_min_e_max_de_t),
              method='simplex')
print(res)
```

Scipy linprog

```
(lp) cassia@galadriel:~/lp$ python simplex.py
     con: array([], dtype=float64)
     fun: -0.5
 message: 'Optimization terminated successfully.'
     nit: 3
   slack: array([0. , 0. , 0.1])
  status: 0
 success: True
       x: array([0.3, 0.2])
```

a = 0.3 t =0.2

Scipy linprog

a = 0.3 t =0.2

 +  = 50%

Matrizes

```
import numpy as np

coeficientes_das_inequacoes = np.matrix('0 2 1 ; 0 1 2 ; 0.1 0 1')

resultados = np.matrix('1; 0.3; 0.2')

print(coeficientes_das_inequacoes*resultados)
```

Matrizes

```
(lp) cassia@galadriel:~/lp$ python matrix.py
[[0.8]
 [0.7]
 [0.3]]
```

Matrizes

```
import numpy as np

coeficientes_das_inequacoes = np.matrix('0 2 1 ; 0 1 2 ; 0.1 0 1')

resultados = np.matrix('1; 0.3; 0.2')

print(coeficientes_das_inequacoes*resultados)
```

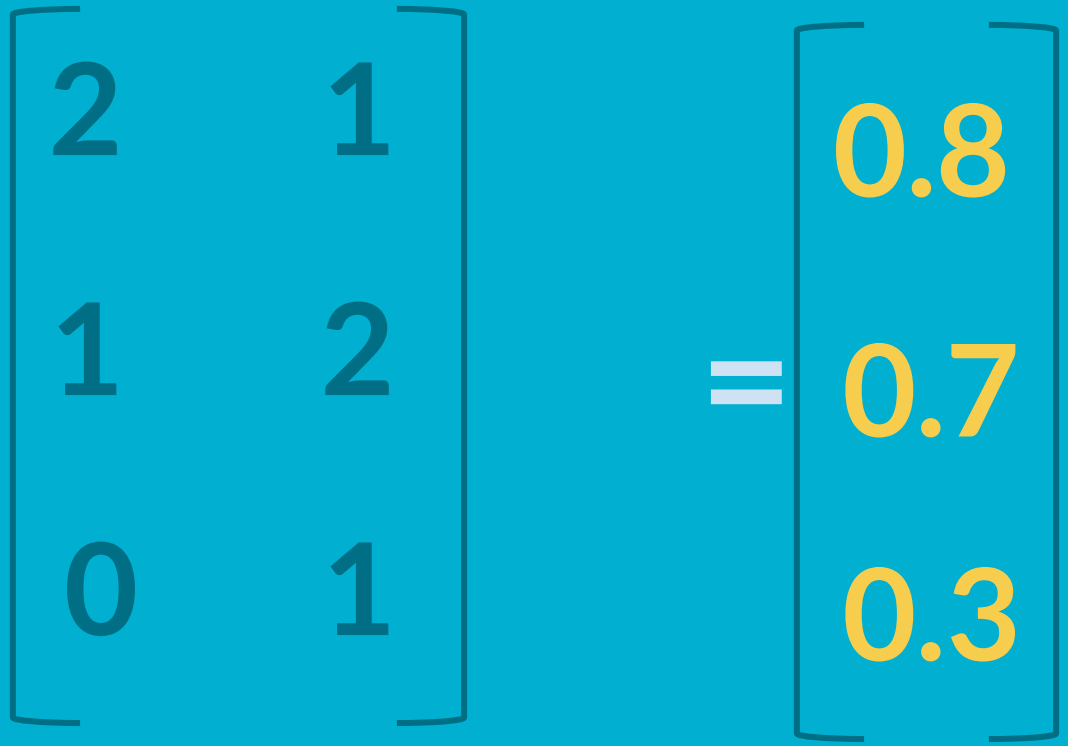
Scipy linprog

```
(lp) cassia@galadriel:~/lp$ python simplex.py
     con: array([], dtype=float64)
     fun: -0.5
 message: 'Optimization terminated successfully.'
     nit: 3
   slack: array([0. , 0. , 0.1])
  status: 0
 success: True
       x: array([0.3, 0.2])
```

Scipy linprog

```
res = linprog(coeficientes_da_funcao_objetiva,
              coeficientes_das_inequacoes,
              valores_maximos,
              bounds=(valor_min_e_max_de_a, valor_min_e_max_de_t),
              method='simplex')
print(res)
```

Estimativa por crowdsourcing



Estimativa por crowdsourcing

$$\begin{bmatrix} 0 \\ 0 \\ 0.1 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

slack a t

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

slack	a	t
0	2	1
0	1	2
0.1	0	1

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

slack	a	t			
0	1+1	1		1	0.8
0	1	2	*	0.3	= 0.7
0.1	0	1		0.2	0.3

Estimativa por crowdsourcing

—

slack a t

$$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

—

slack a t

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0.1 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Estimativa por crowdsourcing

slack P(a) P(t)

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0.1 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

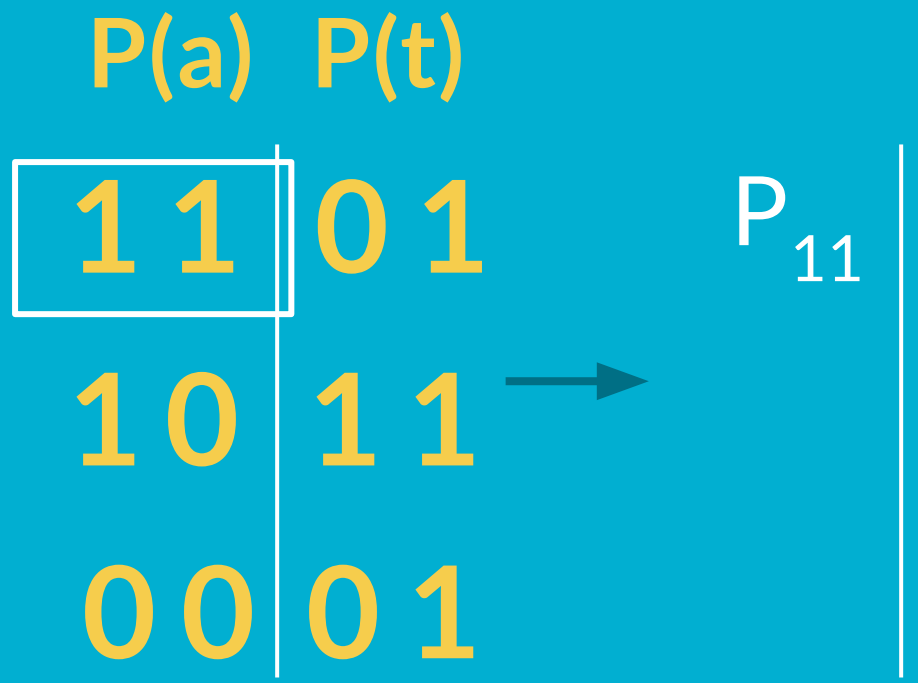
Probabilidades

P(a)	P(t)
1 1	0 1
1 0	1 1
0 0	0 1

Álgebra

P(a)	P(t)
1 1	0 1
1 0	1 1
0 0	0 1

Álgebra



Álgebra

P(a)	P(t)
1 1	0 1
1 0	1 1
0 0	0 1

P_{11}

Álgebra

P(a)	P(t)
1 1	0 1
1 0	1 1
0 0	0 1

P_{11}	P_{01}
P_{10}	P_{11}
P_{00}	P_{01}

Álgebra

P_{11}	P_{01}
P_{10}	P_{11}
P_{00}	P_{01}

Tabela verdade

a	t
0	0
1	1
0	1
1	0

Álgebra

P_{11}	P_{01}
P_{10}	P_{11}
P_{00}	P_{01}

a	t	
0	0	
1	1	P_{11}
0	1	
1	0	

Álgebra

P_{11}	P_{01}
P_{10}	P_{11}
P_{00}	P_{01}

a	t	
0	0	P_{00}
1	1	P_{11}
0	1	P_{01}
1	0	P_{10}

Álgebra

a	t	
0	0	P_{00}
1	1	P_{11}
0	1	P_{01}
1	0	P_{10}

Acidente e trânsito ou só acidente
Trânsito e acidente ou só trânsito
Trânsito
Acidente e trânsito = ?

Álgebra

a	t
0	0
1	1
0	1
1	0

P_{00}
P_{11}
P_{01}
P_{10}

A e t ou a
T e a ou t
T
A e t = ?

Álgebra

A e t ou a
T e a ou t
T
A e t = ?

Álgebra

A ^ t ou a
T ^ a ou t
T
A ^ t = ?

Álgebra

A ^ t v a
T ^ a v t
T
A ^ t = ?

Álgebra

(A ∧ t) ∨ a

(T ∧ a) ∨ t

T

A ∧ t = ?

Álgebra

$(a \wedge t) \vee a$

$(b \wedge a) \vee t$

t

$a \wedge t = ?$

Álgebra

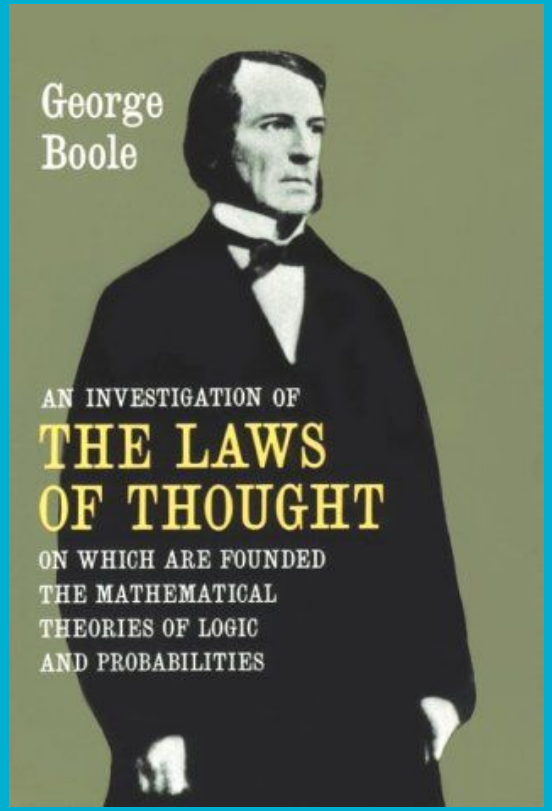
((a ∧ t) ∨ a)
((b ∧ a) ∨ t)
(t)
(a ∧ t)= ?

Lógica booleana probabilística

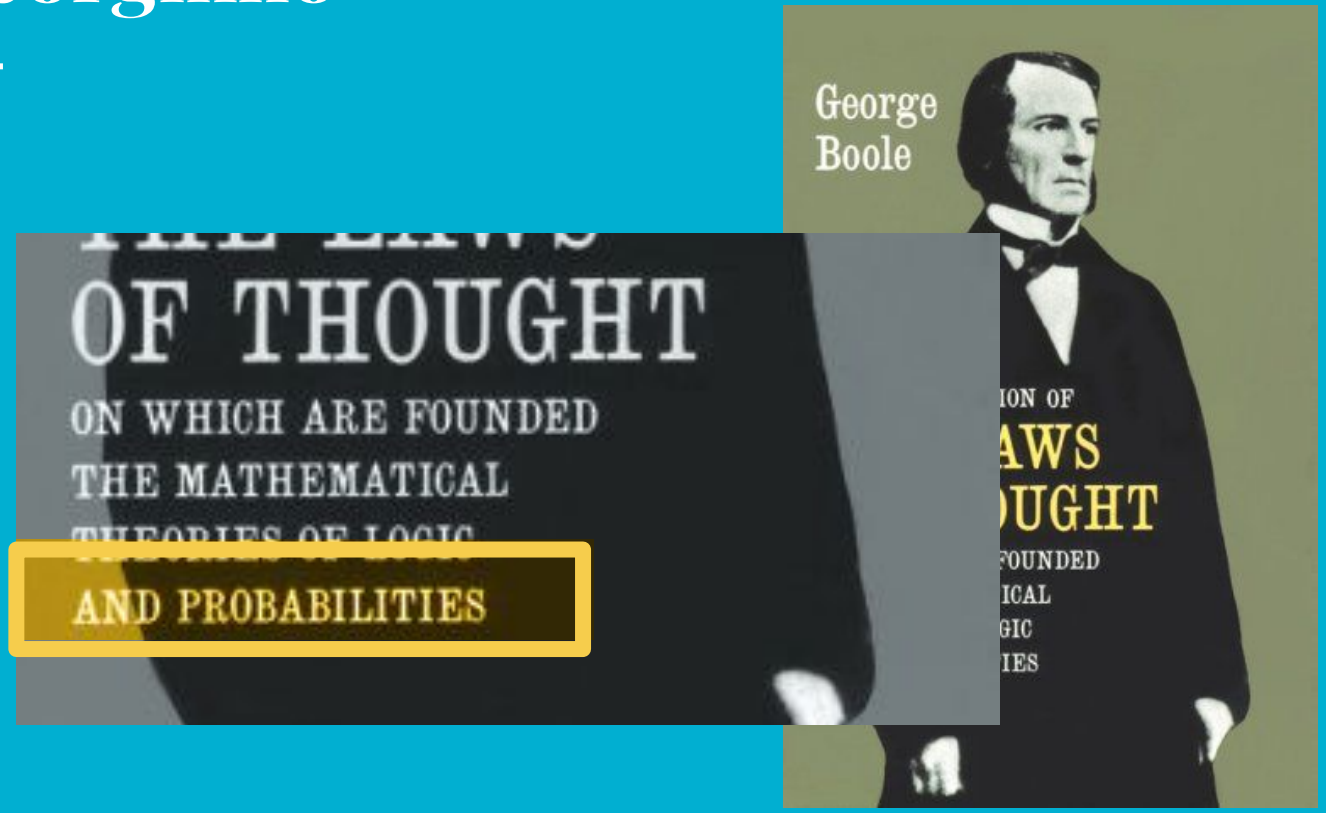
$P((a \wedge t) \vee a)$
$P((b \wedge a) \vee t)$
$P(t)$
$P(a \wedge t) = ?$

Georginho

Leis do Pensamento
[Boole 1854]



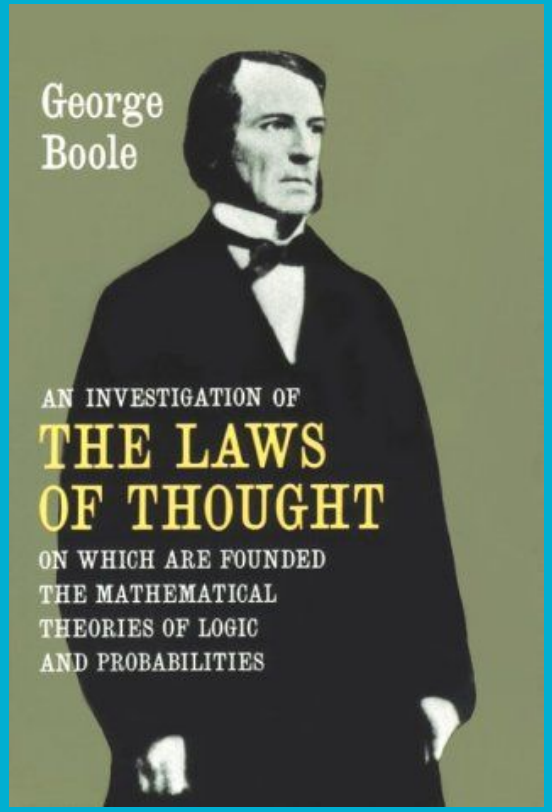
Georginho


George
Boole
OF THOUGHT
ON WHICH ARE FOUNDED
THE MATHEMATICAL
AND PROBABILITIES

Georginho

Satisfatibilidade probabilística (PSAT)

foi proposta em
Leis do Pensamento
[Boole 1854]



Pesquisa

slack	a	t
0	2	1
0	1	2
0.1	0	1

$$\begin{bmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0.1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Pesquisa

slack	a	t
0	2	1
0	1	2
0.1	0	1

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0.3 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.3 \end{bmatrix}$$

Contato

twitter: @cassiasamp

linkedin: /in/cassiasamp/

Fontes

https://panda.ime.usp.br/aulasPython/static/aulasPython/aula05.html

https://public.tepper.cmu.edu/jnh/booleBicentenary.pdf

https://docs.python.org/2.3/whatsnew/section-bool.html

https://www.ime.usp.br/~mfinger/home/psat/Hansen+-Models+AlgoProb+BayesLogic-IJCAI1995.pdf

https://www.youtube.com/watch?v=Q1qPHIGWJDg

Imagens

http://i.imgur.com/B6XlewH.png

http://pluspng.com/img-png/waze-png-waze-icon-png-50-px-1600.png

https://tech4gamers.com/wp-content/uploads/2015/11/boole.jpg

https://i.pinimg.com/736x/a2/99/cc/a299cc484dfc6c3c9c6b891f977e4b3b--family-trips-nerdy.jpg