

```
In [19]: #!/usr/bin/env python
"""
Algoritmo Apriori - base https://rasbt.github.io/mlxtend/
"""

import numpy as np
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import association_rules
from mlxtend.frequent_patterns import apriori, association_rules
```

```
In [2]: #https://www.kaggle.com/irfanasrullah/groceries
dados = pd.read_csv('groceries - groceries.csv', usecols = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32])
```

Trocar os valores não ocupados por 0

```
In [3]: dados.replace(np.nan, 0, inplace = True)
dados.head()
```

```
Out[3]:
```

| | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | Item 6 | Item 7 | Item 8 | Item 9 | Item 10 | ... | Item 23 | Item 24 | Item 25 |
|---|------------------|---------------------|----------------|--------------------------|--------|--------|--------|--------|--------|---------|-----|---------|---------|---------|
| 0 | citrus fruit | semi-finished bread | margarine | ready soups | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| 1 | tropical fruit | yogurt | coffee | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| 2 | whole milk | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| 3 | pip fruit | yogurt | cream cheese | meat spreads | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |
| 4 | other vegetables | whole milk | condensed milk | long life bakery product | 0 | 0 | 0 | 0 | 0 | 0 | ... | 0 | 0 | 0 |

5 rows × 32 columns

Remover esses 0 dos conjuntos

```
In [4]: def remove0correnciasLista(_list, value):
    #filtrar os dados que iram para a lista
    return list(filter(lambda x: x != value, _list))

listar_transacoes = []

for i, row in dados.iterrows():
    transacao = row.values.tolist()
    transacao = remove0correnciasLista(transacao, 0)
    listar_transacoes.append(transacao)
```

Quantidade de transações da base de dados

```
In [5]: len(listar_transacoes)
```

```
Out[5]: 9835
```

```
In [6]: te = TransactionEncoder()
te_ary = te.fit(listar_transacoes).transform(listar_transacoes)
dados = pd.DataFrame(te_ary, columns=te.columns_)
```

O algoritmo apriori é usado para analisar a frequência de itens no conjunto, neste exemplo usaremos um suporte mínimo de 0.01 (1%)

```
In [7]: frequencia_itens_conjunto = apriori(dados, min_support = 0.01, use_colnames=True)
```

Para verificar qual conjunto possui maior e menor suporte só precisamos ordenar por suporte

```
In [8]: frequencia_itens_conjunto.sort_values(by=['support'], ascending = False).head(1)
```

```
Out[8]:
```

| | support | itemsets |
|----|----------|--------------|
| 86 | 0.255516 | (whole milk) |

```
In [9]: frequencia_itens_conjunto.sort_values(by=['support'], ascending = True).head(1)
```

```
Out[9]:
```

| | support | itemsets |
|-----|----------|---------------------------|
| 199 | 0.010066 | (hard cheese, whole milk) |

Isso significa que o conjunto mais popular é o whole milk (25%) sem outro item, e que a compra de whole milk com hard cheese é menos popular (1%)

Para uma confiança mínima de 0.3, usando o mesmo suporte mínimo de 0.1 obtemos a seguinte quantidade de associações:

```
In [10]: regras = association_rules(frequencia_itens_conjunto, metric="confidence",
len(regras.index))
```

```
Out[10]: 125
```

Para saber quais associações são as mais confiáveis além de ordenar por confiabilidade precisamos verificar o lift, para verificar a relevância dos dados.

Dessas associações a 5 mais confiáveis são

```
In [11]: regras.sort_values(by=['lift'], ascending = False).drop(['antecedent support', 'consequent support', 'lift'])
```

```
Out[11]:
```

| | antecedents | consequents | support | confidence | lift |
|----|------------------------------------|-------------------|----------|------------|----------|
| 73 | (other vegetables, citrus fruit) | (root vegetables) | 0.010371 | 0.359155 | 3.295045 |
| 95 | (other vegetables, tropical fruit) | (root vegetables) | 0.012303 | 0.342776 | 3.144780 |

| | antecedents | consequents | support | confidence | lift |
|----|---------------------------------|--------------------|----------|------------|----------|
| 1 | (beef) | (root vegetables) | 0.017387 | 0.331395 | 3.040367 |
| 74 | (citrus fruit, root vegetables) | (other vegetables) | 0.010371 | 0.586207 | 3.029608 |

As 5 menos confiáveis:

```
In [12]: regras.sort_values(by=['lift'], ascending = True).drop(['antecedent support'])
```

| | antecedents | consequents | support | confidence | lift |
|----|-----------------|--------------|----------|------------|----------|
| 59 | (rolls/buns) | (whole milk) | 0.056634 | 0.307905 | 1.205032 |
| 6 | (bottled water) | (whole milk) | 0.034367 | 0.310948 | 1.216940 |
| 61 | (sausage) | (whole milk) | 0.029893 | 0.318182 | 1.245252 |
| 17 | (coffee) | (whole milk) | 0.018709 | 0.322242 | 1.261141 |
| 65 | (waffles) | (whole milk) | 0.012710 | 0.330688 | 1.294196 |

As confiança baseada no lift vai nos dá os produtos mais relevantes das compras, ou seja, quem comprou determinado item a chance dele comprar esse outro é maior/menor.

Quanto maior o valor do lift maior a associação.

Para uma confiança mínima de 0.5 temos a seguinte regra e quantidade de associações

```
In [13]: regras = association_rules(frequencia_itens_conjunto, metric="confidence",
len(regras.index))
```

Out[13]: 15

As 5 mais confiáveis com confiança mínima de 0.5 são:

```
In [14]: regras.sort_values(by=['lift'], ascending = False).drop(['antecedent support'])
```

| | antecedents | consequents | support | confidence | lift |
|---|-----------------------------------|--------------------|----------|------------|----------|
| 1 | (citrus fruit, root vegetables) | (other vegetables) | 0.010371 | 0.586207 | 3.029608 |
| 6 | (tropical fruit, root vegetables) | (other vegetables) | 0.012303 | 0.584541 | 3.020999 |
| 5 | (root vegetables, rolls/buns) | (other vegetables) | 0.012201 | 0.502092 | 2.594890 |
| 7 | (yogurt, root vegetables) | (other vegetables) | 0.012913 | 0.500000 | 2.584078 |
| 2 | (yogurt, curd) | (whole milk) | 0.010066 | 0.582353 | 2.279125 |

As 5 mais fracas (confiança mínima de 0.5)

```
In [15]: regras.sort_values(by=['lift'], ascending = True).drop(['antecedent support'])
```

| | antecedents | consequents | support | confidence | lift |
|----|--|--------------|----------|------------|----------|
| 8 | (other vegetables, whipped/sour cream) | (whole milk) | 0.014642 | 0.507042 | 1.984385 |
| 9 | (other vegetables, yogurt) | (whole milk) | 0.022267 | 0.512881 | 2.007235 |
| 13 | (yogurt, tropical fruit) | (whole milk) | 0.015150 | 0.517361 | 2.024770 |
| 4 | (other vegetables, pip fruit) | (whole milk) | 0.013523 | 0.517510 | 2.025351 |

| antecedents | consequents | support | confidence | lift |
|-------------|-------------|---------|------------|------|
|-------------|-------------|---------|------------|------|

```
In [16]: regras = association_rules(frequencia_itens_conjunto, metric="confidence",
len(regras.index)
```

```
Out[16]: 0
```

```
In [17]: regras.sort_values(by=['lift'], ascending = False).drop(['antecedent support
```

```
Out[17]:
```

| antecedents | consequents | support | confidence | lift |
|-------------|-------------|---------|------------|------|
|-------------|-------------|---------|------------|------|

```
In [18]: regras.sort_values(by=['lift'], ascending = True).drop(['antecedent support
```

```
Out[18]:
```

| antecedents | consequents | support | confidence | lift |
|-------------|-------------|---------|------------|------|
|-------------|-------------|---------|------------|------|

Associações mais fortes de acordo com o lift? (Lift > 1)

Para verificar as associações mais forte com lift > 1 precisamos de uma certa porcentagem de confiança no resultado, usando uma confiança mínima de 50% e verificar os que mais saem

```
In [19]: regras = association_rules(frequencia_itens_conjunto, metric="lift", min_t

regras = association_rules(frequencia_itens_conjunto, metric="confidence",
regras.sort_values(by=['lift'], ascending = False).drop(['antecedent support
```

```
Out[19]:
```

| | antecedents | consequents | support | confidence | lift |
|----|--|--------------------|----------|------------|----------|
| 1 | (root vegetables, citrus fruit) | (other vegetables) | 0.010371 | 0.586207 | 3.029608 |
| 6 | (root vegetables, tropical fruit) | (other vegetables) | 0.012303 | 0.584541 | 3.020999 |
| 5 | (root vegetables, rolls/buns) | (other vegetables) | 0.012201 | 0.502092 | 2.594890 |
| 7 | (root vegetables, yogurt) | (other vegetables) | 0.012913 | 0.500000 | 2.584078 |
| 2 | (curd, yogurt) | (whole milk) | 0.010066 | 0.582353 | 2.279125 |
| 0 | (butter, other vegetables) | (whole milk) | 0.011490 | 0.573604 | 2.244885 |
| 11 | (root vegetables, tropical fruit) | (whole milk) | 0.011998 | 0.570048 | 2.230969 |
| 12 | (root vegetables, yogurt) | (whole milk) | 0.014540 | 0.562992 | 2.203354 |
| 3 | (domestic eggs, other vegetables) | (whole milk) | 0.012303 | 0.552511 | 2.162336 |
| 14 | (whipped/sour cream, yogurt) | (whole milk) | 0.010880 | 0.524510 | 2.052747 |
| 10 | (root vegetables, rolls/buns) | (whole milk) | 0.012710 | 0.523013 | 2.046888 |
| 4 | (pip fruit, other vegetables) | (whole milk) | 0.013523 | 0.517510 | 2.025351 |
| 13 | (tropical fruit, yogurt) | (whole milk) | 0.015150 | 0.517361 | 2.024770 |
| 9 | (yogurt, other vegetables) | (whole milk) | 0.022267 | 0.512881 | 2.007235 |
| 8 | (whipped/sour cream, other vegetables) | (whole milk) | 0.014642 | 0.507042 | 1.984385 |

Defina um limiar para o que seria uma alta confiança. Quais as associações que possuem alta confiança, porém lift < 1?

```
In [20]: regras[(regras['lift'] < 1) ]
```

```
Out[20]:
```

| antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | convic |
|-------------|-------------|-----------------------|-----------------------|---------|------------|------|----------|--------|
|-------------|-------------|-----------------------|-----------------------|---------|------------|------|----------|--------|

Uma alta confiança seria um conjunto onde tanto a confiança quanto o lift sejam os maiores de um conjunto, onde a confiança seja $\geq 50\%$, por exemplo, e lift maior que 1

Nesta base de dados nenhum lift é menor que 1.

```
In [ ]:
```