

SUGESTÕES DE BIBLIOTECAS PYTHON

Seguem sugestões de bibliotecas Python para construção de sistemas de recomendação via Apriori:

APYORI

<https://github.com/ymoch/apyori>

Exemplo de uso:

```
from apyori import apriori

transactions = [
    ['beer', 'nuts'],
    ['beer', 'cheese'],
]

results = list(apriori(transactions))
```

EFFICIENT APRIORI

<https://pypi.org/project/efficient-apriori/>

Exemplo de uso:

```
from efficient_apriori import apriori

transactions = [('eggs', 'bacon', 'soup'),
                ('eggs', 'bacon', 'apple'),
                ('soup', 'bacon', 'banana')]
```

IMPACTA

```
itemsets, rules = apriori(transactions, min_support=0.5,  
min_confidence=1)  
  
print(rules)
```

MLXTEND

<http://rasbt.github.io/mlxtend/>

A biblioteca **mlxtend** possui uma implementação do Apriori.

Exemplo de uso:

```
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],  
['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'], ['Milk',  
'Apple', 'Kidney Beans', 'Eggs'], ['Milk', 'Unicorn', 'Corn', 'Kidney Beans',  
'Yogurt'], ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]  
  
import pandas as pd  
  
from mlxtend.preprocessing import TransactionEncoder  
  
# Transformando em uma base transacional  
  
te = TransactionEncoder()  
te_ary = te.fit(dataset).transform(dataset)  
df = pd.DataFrame(te_ary, columns=te.columns_)  
  
# Usando o apriori do mlxtend  
  
from mlxtend.frequent_patterns import apriori  
apriori(df, min_support=0.6)
```

Você pode ver mais detalhes deste exemplo em

http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/.