

Aula 8 - Análise de associação

8.1 O pacote apyori

Usaremos o pacote `apyori` para realizar as análises. Par ainstalar:

```
pip install apyori
```

```
In [1]: from apyori import apriori
```

O algoritmo apriori do pacote precisa que os dados estejam no formato de lista de listas. Considere o exemplo abaixo como template:

8.2 Um exemplo simples

```
In [2]: records = [[["leite","pao","manteiga","açucar"],
                  ["leite","pao","manteiga"],
                  ["leite","pao","coca","açucar"]]
association_rules = apriori(records, min_support=0.0001, min_confidence=0.0, min_lift=0, min_length=1)
association_results = list(association_rules)
print(association_results[0])

RelationRecord(items=frozenset({'açucar'}), support=0.6666666666666666, ordered_statistics=[OrderedStatistic(it
ems_base=frozenset(), items_add=frozenset({'açucar'}), confidence=0.6666666666666666, lift=1.0)])
```

8.3 Importando um banco de dados

Usando o banco de dados `store_data.csv`, em que cada linha contém os itens comprados em uma loja na França:

```
In [3]: import pandas as pd

dt = pd.read_csv(r"G:\Meu Drive\Arquivos\UFPR\Disciplinas\2 - Intro Mineração de Dados\Python\Datasets\store_d
dt.head()
```

0	shrimp	almonds	avocado	vegetables mix	grapes	weat flour	yams	cheese	drink	juice	fat yogurt	tea	honey	salad	water	salmon	juice
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

8.4 Aplicando o algoritmo

Temos que criar uma lista de listas com os itens.

```
In [4]: vv_itens = []
for linha in range(0, dt.shape[0]):
    vv_itens.append(dt.iloc[linha].dropna().values)
```

Aplicando o algoritmo, com os parâmetros:

`min_support` : suporte mínimo para as regras.

`min_confidence` : confiança mínima para as regras.

`min_lift` : lift mínimo para as regras.

`min_length` : número mínimo de itens que as regras devem ter

```
In [5]: association_rules = apriori(vv_itens, min_support = 0.0045, min_confidence = 0.2, min_lift=3, min_length = 2)

# Transformando os resultados em uma lista, para facilitar a visualização
resultados = list(association_rules)
```

O resultado é uma lista, sendo que cada elemento possui informações a respoeito de uma regra. Os dados da lista são também listas. O primeiro elemento é um `frozenset` com os produtos da regra (antecedente e consequente), podemos criar uma lista para desempacotar os valores:

O segundo elemento da lista interna guarda o valor de suporte da regra:

```
In [6]: regra = 0
print("Suporte :", resultados[regra][1])

Suporte : 0.004532728969470737
```

O terceiro item da lista interna contém mais uma lista com um elemento do tipo `OrderedStatistic`, sendo que este também é tratado como uma lista.o primeiro e segundo elementos dessa lista são cada um as listas dos elementos precedentes e antededenentes da regra.

Podemos extrai-los da seguinte forma:

```
In [7]: regra = 0
ant = [x for x in resultados[regra][2][0][0]]
cons = [x for x in resultados[regra][2][0][1]]
print(ant, "->", cons)

['light cream'] -> ['chicken']
```

O terceiro e quarto elementos desta lista armazenam a confiança e o lift:

```
In [8]: print("Confiança :", resultados[regra][2][0][2])
print("Lift :", resultados[regra][2][0][3])

Confiança : 0.29059829059829057
Lift : 4.84395061728395
```

Juntando tudo, podemos imprimir as informações para todas as regras retornadas pelo algoritmo:

```
In [9]: for i in range(0, len(resultados)):
    ant = [x for x in resultados[i][2][0][0]]
    cons = [x for x in resultados[i][2][0][1]]
    print(ant, "->", cons)
    print("Suporte :", resultados[i][1])
    print("Confiança :", resultados[i][2][0][2])
    print("Lift :", resultados[i][2][0][3])
    print("=====")

['light cream'] -> ['chicken']
Suporte : 0.004532728969470737
Confiança : 0.29059829059829057
Lift : 4.84395061728395
=====
['mushroom cream sauce'] -> ['escalope']
Suporte : 0.005732568990801226
Confiança : 0.3006993006993007
Lift : 3.790832696715049
=====
['pasta'] -> ['escalope']
Suporte : 0.005865884548726837
Confiança : 0.3728813559322034
Lift : 4.700811850163794
=====
['herb & pepper'] -> ['ground beef']
Suporte : 0.015997866951073192
Confiança : 0.3234501347708895
Lift : 3.2919938411349285
=====
['tomato sauce'] -> ['ground beef']
Suporte : 0.005332622317024397
Confiança : 0.3773584905660377
Lift : 3.840659481324083
=====
['whole wheat pasta'] -> ['olive oil']
Suporte : 0.007998933475536596
Confiança : 0.2714932126696833
Lift : 4.122410097642296
=====
['pasta'] -> ['shrimp']
Suporte : 0.005065991201173177
Confiança : 0.3220338983050847
Lift : 4.506672147735896
=====
['chocolate', 'frozen vegetables'] -> ['shrimp']
Suporte : 0.005332622317024397
Confiança : 0.23255813953488375
Lift : 3.2545123221103784
=====
['ground beef', 'cooking oil'] -> ['spaghetti']
Suporte : 0.004799360085321957
Confiança : 0.5714285714285714
Lift : 3.2819951870487856
=====
['spaghetti', 'frozen vegetables'] -> ['ground beef']
Suporte : 0.008665511265164644
Confiança : 0.31100478468899523
Lift : 3.165328208890303
=====
['milk', 'frozen vegetables'] -> ['olive oil']
Suporte : 0.004799360085321957
Confiança : 0.20338983050847456
Lift : 3.088314005352364
=====
['mineral water', 'shrimp'] -> ['frozen vegetables']
Suporte : 0.007199040127982935
Confiança : 0.30508474576271183
Lift : 3.200616332819722
=====
['spaghetti', 'frozen vegetables'] -> ['olive oil']
Suporte : 0.005732568990801226
Confiança : 0.20574162679425836
Lift : 3.1240241752707125
=====
['spaghetti', 'frozen vegetables'] -> ['shrimp']
Suporte : 0.005999200106652446
Confiança : 0.21531100478468898
Lift : 3.0131489680782684
=====
['spaghetti', 'frozen vegetables'] -> ['tomatoes']
Suporte : 0.006665777896280496
Confiança : 0.23923444976076558
Lift : 3.4980460188216425
=====
['spaghetti', 'grated cheese'] -> ['ground beef']
Suporte : 0.005332622317024397
Confiança : 0.3225806451612903
Lift : 3.283144395325426
=====
['mineral water', 'herb & pepper'] -> ['ground beef']
Suporte : 0.006665777896280496
Confiança : 0.39062500000000006
Lift : 3.975682666214383
=====
['spaghetti', 'herb & pepper'] -> ['ground beef']
Suporte : 0.006399146780429276
Confiança : 0.3934426229508197
Lift : 4.004359721511667
=====
['ground beef', 'milk'] -> ['olive oil']
Suporte : 0.004932675643247567
Confiança : 0.22424242424242427
Lift : 3.40494417862839
=====
['ground beef', 'shrimp'] -> ['spaghetti']
Suporte : 0.005999200106652446
Confiança : 0.5232558139534884
Lift : 3.005315360233627
=====
['spaghetti', 'milk'] -> ['olive oil']
Suporte : 0.007199040127982935
Confiança : 0.20300751879699247
Lift : 3.0825089038385434
=====
['mineral water', 'soup'] -> ['olive oil']
Suporte : 0.005199306759098787
Confiança : 0.22543352601156072
Lift : 3.4230301186492245
=====
['spaghetti', 'pancakes'] -> ['olive oil']
Suporte : 0.005065991201173177
Confiança : 0.20105820105820105
Lift : 3.0529100529100526
=====
['mineral water', 'spaghetti', 'milk'] -> ['frozen vegetables']
Suporte : 0.004532728969470737
Confiança : 0.28813559322033894
Lift : 3.0228043143297376
=====
```

Exercícios

- Considere o conjunto de dados `MateriaisConstrucao.xlsx`, que mantém um registro da quantidade de itens que foram comprados em uma loja de materiais de construção. Será que existem alguns itens que estão relacionados com outros, quando da compra pelos clientes? Use regras de associação para responder.