

Reglas de Asociación

December 23, 2021

```
[1]: import pandas as pd
      from apyori import apriori
```

1 Reglas de asociación

```
[2]: # Celda 1 #

# Lectura del dataset
cc = pd.read_table('cesta_compra2.csv', header=None)
# Transformación de la lectura para tomar la lista de items en un único campo
cc = cc[0].str.split(',', 2, expand=True)
cc.columns = cc.iloc[0]
cc = cc[1:]
# Información del dataset
cc.info()
cc.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14963 entries, 1 to 14963
Data columns (total 3 columns):
#   Column   Non-Null Count  Dtype
---  -
0   Usuario  14963 non-null  object
1   Fecha    14963 non-null  object
2   Item     14963 non-null  object
dtypes: object(3)
memory usage: 350.8+ KB
```

```
[2]: 0 Usuario      Fecha      Item
1    0440  2020-07-14    leche entera,chocolate
2    0440  2020-07-26      bollos,bollos
3    0440  2021-09-06  bolsas de compra,bolsas de compra
4    0440  2020-01-01    otros vegetales,yogurt
5    0440  2020-01-06  frutas tropicales,zumo de frutas
```

```
[3]: # Celda 2 #

# Creamos una lista de listas con los items de cada compra para poder usar el
↳ algoritmo priori
data = []
for i in range(14963):
    data.append(cc['Item'][i+1].split(','))
```

```
[4]: # Celda 3 #

# Ejecutamos el algoritmo priori
result = apriori(data, min_support=0.001, min_confidence=0.1, min_lift=1.25)
output = list(result)
```

```
[5]: # Celda 4 #

# Función para hacer una tabla con la salida de la reglas de asociación
def inspect(output):
    lhs = [tuple(result[2][0][0]) for result in output]
    rhs = [tuple(result[2][0][1]) 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))
# Transformamos la salida a una tabla y mostamos las asociaciones generadas
output_DataFrame = pd.DataFrame(inspect(output), columns = ['Si se compra...',
↳ 'se tiende a comprar...', 'Soporte', 'Confianza', 'Lift'])
output_DataFrame
```

```
[5]:
```

	Si se compra...	se tiende a comprar...	Soporte	Confianza \
0	(frutas empaquetadas,)	(bollos,)	0.001203	0.141732
1	(productos de temporada,)	(bollos,)	0.001002	0.141509
2	(queso gratinar,)	(bollos,)	0.001470	0.144737
3	(chicle,)	(yogurt,)	0.001403	0.116667
4	(detergente,)	(yogurt,)	0.001069	0.124031
5	(harina,)	(frutas tropicales,)	0.001069	0.109589
6	(queso gratinar,)	(hortalizas,)	0.001069	0.105263
7	(infusiones,)	(yogurt,)	0.001136	0.107595
8	(queso freco,)	(yogurt,)	0.001270	0.126667
9	(bollos, salchichas)	(leche entera,)	0.001136	0.212500
10	(leche entera, salchichas)	(yogurt,)	0.001470	0.164179

	Lift
0	1.288421
1	1.286395
2	1.315734
3	1.358508

4	1.444261
5	1.617141
6	1.513019
7	1.252874
8	1.474952
9	1.345594
10	1.911760

[]: