Association Rules

December 10, 2017

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

```
/usr/local/lib/python2.7/dist-packages/pandas/core/computation/__init__.py:18: UserWarning: The
The minimum supported version is 2.4.6

  ver=ver, min_ver=_MIN_NUMEXPR_VERSION), UserWarning)
```

```
In [18]: df = pd.DataFrame()

          with open('groceries.csv', 'r') as f:
              for line in f:
                  df = pd.concat( [df, pd.DataFrame([tuple(line.strip().split(','))])], ignore_in
```

```
In [8]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9835 entries, 0 to 9834
Data columns (total 32 columns):
0     9835 non-null object
1     7676 non-null object
2     6033 non-null object
3     4734 non-null object
4     3729 non-null object
5     2874 non-null object
6     2229 non-null object
7     1684 non-null object
8     1246 non-null object
9     896 non-null object
10    650 non-null object
11    468 non-null object
12    351 non-null object
13    273 non-null object
14    196 non-null object
```

```
15    141 non-null object
16    95 non-null object
17    66 non-null object
18    52 non-null object
19    38 non-null object
20    29 non-null object
21    18 non-null object
22    14 non-null object
23    8 non-null object
24    7 non-null object
25    7 non-null object
26    6 non-null object
27    5 non-null object
28    4 non-null object
29    1 non-null object
30    1 non-null object
31    1 non-null object
dtypes: object(32)
memory usage: 2.4+ MB
```

```
In [9]: df.head()
```

```
Out[9]:                   0                    1                2  \
        0      citrus fruit  semi-finished bread        margarine
        1    tropical fruit               yogurt           coffee
        2        whole milk                  NaN              NaN
        3         pip fruit               yogurt   cream cheese
        4  other vegetables           whole milk  condensed milk

                                3    4    5    6    7    8    9  ...    22   23   24  \
        0             ready soups  NaN  NaN  NaN  NaN  NaN  NaN  ...   NaN  NaN  NaN
        1                     NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   NaN  NaN  NaN
        2                     NaN  NaN  NaN  NaN  NaN  NaN  NaN  ...   NaN  NaN  NaN
        3            meat spreads  NaN  NaN  NaN  NaN  NaN  NaN  ...   NaN  NaN  NaN
        4  long life bakery product  NaN  NaN  NaN  NaN  NaN  NaN  ...   NaN  NaN  NaN

            25   26   27   28   29   30   31
        0  NaN  NaN  NaN  NaN  NaN  NaN  NaN
        1  NaN  NaN  NaN  NaN  NaN  NaN  NaN
        2  NaN  NaN  NaN  NaN  NaN  NaN  NaN
        3  NaN  NaN  NaN  NaN  NaN  NaN  NaN
        4  NaN  NaN  NaN  NaN  NaN  NaN  NaN

        [5 rows x 32 columns]
```

```
In [19]: numpyMatrix = df.as_matrix()
```

```
In [21]: numpyMatrix
```

Out[21]: array([['citrus fruit', 'semi-finished bread', 'margarine', ..., nan, nan,
         nan],
        ['tropical fruit', 'yogurt', 'coffee', ..., nan, nan, nan],
        ['whole milk', nan, nan, ..., nan, nan, nan],
        ...,
        ['chicken', 'citrus fruit', 'other vegetables', ..., nan, nan, nan],
        ['semi-finished bread', 'bottled water', 'soda', ..., nan, nan, nan],
        ['chicken', 'tropical fruit', 'other vegetables', ..., nan, nan, nan]], dtype=ob

```
In [22]: from mlxtend.preprocessing import OnehotTransactions
```

```
In [23]: oht = OnehotTransactions()
         oht_ary = oht.fit(numpyMatrix).transform(numpyMatrix)
         dataframe= pd.DataFrame(oht_ary, columns=oht.columns_)
```

```
In [35]: dataframe.drop(dataframe.columns[0], axis=1)
```

Out[35]:

	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener \
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	1	0
6	0	0	0	0
7	0	1	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	0	0	0	0
16	0	0	0	0
17	0	0	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	0	0	0	0
22	0	0	0	0
23	0	0	0	0
24	0	0	0	0
25	0	0	0	0
26	0	0	0	0
27	0	0	0	0
28	0	0	0	0
29	0	0	0	0
...	...	...	...	...

9805	0	0	0	0
9806	0	0	0	0
9807	0	0	0	0
9808	0	0	0	0
9809	0	0	0	0
9810	1	0	0	0
9811	0	0	0	0
9812	0	0	0	0
9813	0	0	0	0
9814	0	0	0	0
9815	0	0	0	0
9816	0	0	0	0
9817	0	0	0	0
9818	0	0	0	0
9819	0	0	0	0
9820	0	0	0	0
9821	0	1	0	0
9822	0	0	0	0
9823	0	0	0	0
9824	0	0	0	0
9825	0	0	0	0
9826	0	0	0	0
9827	0	0	0	0
9828	0	0	0	0
9829	0	0	0	0
9830	0	0	0	0
9831	0	0	0	0
9832	0	0	0	0
9833	0	0	0	0
9834	0	0	0	0

	baby cosmetics	baby food	bags	baking powder	bathroom cleaner	beef	\
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	0	
3	0	0	0	0	0	0	
4	0	0	0	0	0	0	
5	0	0	0	0	0	0	
6	0	0	0	0	0	0	
7	0	0	0	0	0	0	
8	0	0	0	0	0	0	
9	0	0	0	0	0	0	
10	0	0	0	0	0	0	
11	0	0	0	0	0	0	
12	0	0	0	0	0	1	
13	0	0	0	0	0	0	
14	0	0	0	0	0	0	
15	0	0	0	0	0	0	

16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	1	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0
29	0	0	0	0	0	0
...	...	...	...	...	...	...
9805	0	0	0	0	0	0
9806	0	0	0	0	0	0
9807	0	0	0	0	0	0
9808	0	0	0	0	0	0
9809	0	0	0	0	0	1
9810	0	0	0	0	0	0
9811	0	0	0	0	0	0
9812	0	0	0	0	0	0
9813	0	0	0	0	0	0
9814	0	0	0	0	0	0
9815	0	0	0	0	0	0
9816	0	0	0	0	0	0
9817	0	0	0	0	0	0
9818	0	0	0	0	0	0
9819	0	0	0	0	0	0
9820	0	0	0	1	0	1
9821	0	0	0	0	0	0
9822	0	0	0	0	0	0
9823	0	0	0	0	0	0
9824	0	0	0	0	0	0
9825	0	0	0	0	0	0
9826	0	0	0	0	0	0
9827	0	0	0	0	0	0
9828	0	0	0	0	0	0
9829	0	0	0	0	0	0
9830	0	0	0	0	0	1
9831	0	0	0	0	0	0
9832	0	0	0	0	0	0
9833	0	0	0	0	0	0
9834	0	0	0	0	0	0

	...	turkey	vinegar	waffles	whipped/sour cream	whisky	\
0	...	0	0	0	0	0	

1	...	0	0	0	0	0
2	...	0	0	0	0	0
3	...	0	0	0	0	0
4	...	0	0	0	0	0
5	...	0	0	0	0	0
6	...	0	0	0	0	0
7	...	0	0	0	0	0
8	...	0	0	0	0	0
9	...	0	0	0	0	0
10	...	0	0	0	0	0
11	...	0	0	0	0	0
12	...	0	0	0	0	0
13	...	0	0	0	0	0
14	...	0	0	0	0	0
15	...	0	0	0	0	0
16	...	0	0	0	0	0
17	...	0	0	0	0	0
18	...	0	0	0	0	0
19	...	0	0	0	0	0
20	...	0	0	0	0	0
21	...	0	0	0	0	0
22	...	0	0	0	0	0
23	...	0	0	0	0	0
24	...	0	0	1	0	0
25	...	0	0	0	0	0
26	...	0	0	0	0	0
27	...	0	0	0	0	0
28	...	0	0	0	0	0
29	...	0	0	0	0	0
...	...	...	...	...	...	...
9805	...	0	0	0	0	0
9806	...	0	0	0	0	0
9807	...	0	0	0	0	0
9808	...	0	0	0	0	0
9809	...	0	0	0	0	0
9810	...	0	0	0	0	0
9811	...	0	0	0	0	0
9812	...	0	0	0	0	0
9813	...	0	0	0	0	0
9814	...	0	0	0	0	0
9815	...	0	0	0	0	0
9816	...	0	0	0	0	0
9817	...	0	0	0	1	0
9818	...	0	0	0	0	0
9819	...	0	0	1	1	0
9820	...	0	0	0	0	0
9821	...	0	0	0	0	0
9822	...	0	0	0	0	0

9823	...	0	0	0	0	0
9824	...	0	0	0	0	0
9825	...	0	0	0	0	0
9826	...	0	0	0	0	0
9827	...	0	0	0	0	0
9828	...	0	0	1	0	0
9829	...	0	0	0	0	0
9830	...	0	0	0	1	0
9831	...	0	0	0	0	0
9832	...	0	0	0	0	0
9833	...	0	0	0	0	0
9834	...	0	1	0	0	0

	white bread	white wine	whole milk	yogurt	zwieback
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	1	0	0
5	0	0	1	1	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	1	0	0
10	1	0	0	0	0
11	0	0	1	1	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	1	0	0
23	0	0	0	0	0
24	0	0	0	0	0
25	0	0	0	0	0
26	0	0	0	1	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
...	...	...	...	...	...
9805	1	0	0	0	0
9806	0	0	1	0	0
9807	0	0	0	0	0

9808	0	0	0	0	0
9809	0	0	0	0	0
9810	0	0	0	0	0
9811	0	0	1	0	0
9812	0	0	0	0	0
9813	0	0	1	0	0
9814	0	0	1	1	0
9815	0	0	0	0	0
9816	0	0	0	0	0
9817	0	0	1	1	0
9818	0	0	0	0	0
9819	0	0	1	1	0
9820	0	0	0	0	0
9821	0	0	1	0	1
9822	0	0	0	1	0
9823	0	0	0	0	0
9824	0	0	0	0	0
9825	0	0	0	0	0
9826	0	0	0	0	0
9827	0	0	1	0	0
9828	0	0	0	0	0
9829	0	0	0	0	1
9830	0	0	1	0	0
9831	0	0	0	0	0
9832	0	0	0	1	0
9833	0	0	0	0	0
9834	0	0	0	0	0

[9835 rows x 171 columns]

In [56]:
```
from mlxtend.frequent_patterns import apriori

frequent_itemsets= apriori(dataframe, min_support=0.05, use_colnames=True)
frequent_itemsets
```

Out[56]:

	support	itemsets
0	0.052466	[beef]
1	0.080529	[bottled beer]
2	0.110524	[bottled water]
3	0.064870	[brown bread]
4	0.055414	[butter]
5	0.077682	[canned beer]
6	0.082766	[citrus fruit]
7	0.058058	[coffee]
8	0.053279	[curd]
9	0.063447	[domestic eggs]
10	0.058973	[frankfurter]
11	0.072293	[fruit/vegetable juice]

```
12  0.058566                        [margarine]
13  0.052364                          [napkins]
14  0.079817                       [newspapers]
15  0.193493                 [other vegetables]
16  0.088968                           [pastry]
17  0.075648                        [pip fruit]
18  0.057651                             [pork]
19  0.183935                       [rolls/buns]
20  0.108998                  [root vegetables]
21  0.093950                          [sausage]
22  0.098526                    [shopping bags]
23  0.174377                             [soda]
24  0.104931                   [tropical fruit]
25  0.071683               [whipped/sour cream]
26  0.255516                       [whole milk]
27  0.139502                           [yogurt]
28  0.074835  [other vegetables, whole milk]
29  0.056634        [rolls/buns, whole milk]
30  0.056024            [whole milk, yogurt]
```

In [57]:
```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules
```

Out[57]:

	antecedants	consequents	support	confidence	lift
0	(whole milk)	(other vegetables)	0.255516	0.292877	1.513634
1	(other vegetables)	(whole milk)	0.193493	0.386758	1.513634
2	(whole milk)	(yogurt)	0.255516	0.219260	1.571735
3	(yogurt)	(whole milk)	0.139502	0.401603	1.571735
4	(whole milk)	(rolls/buns)	0.255516	0.221647	1.205032
5	(rolls/buns)	(whole milk)	0.183935	0.307905	1.205032