

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: 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', 'Kidney Beans', 'Ice cream', 'Eggs']]
```

```
In [3]: !pip install mlxtend
```

```
Collecting mlxtend
```

```
  Obtaining dependency information for mlxtend from https://files.pythonhosted.org/packages/1c/07/512f6a780239ad6ce06ce2aa7b4067583f5ddcfcc7703a964a082c706a070/mlxtend-0.23.1-py3-none-any.whl.metadata
```

```
    Downloading mlxtend-0.23.1-py3-none-any.whl.metadata (7.3 kB)
```

```
Requirement already satisfied: scipy>=1.2.1 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (1.11.1)
```

```
Requirement already satisfied: numpy>=1.16.2 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (1.24.3)
```

```
Requirement already satisfied: pandas>=0.24.2 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (2.0.3)
```

```
Requirement already satisfied: scikit-learn>=1.0.2 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (1.3.0)
```

```
Requirement already satisfied: matplotlib>=3.0.0 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (3.7.2)
```

```
Requirement already satisfied: joblib>=0.13.2 in c:\users\admin\anaconda3\lib\site-packages (from mlxtend) (1.2.0)
```

```
Requirement already satisfied: contourpy>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.0.5)
```

```
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (0.11.0)
```

```
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (4.25.0)
```

```
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (1.4.4)
```

```
Requirement already satisfied: packaging>=20.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (23.1)
```

```
Requirement already satisfied: pillow>=6.2.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (9.4.0)
```

```
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (3.0.9)
```

```
Requirement already satisfied: python-dateutil>=2.7 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
```

```
Requirement already satisfied: pytz>=2020.1 in c:\users\admin\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2023.3.post1)
```

```
Requirement already satisfied: tzdata>=2022.1 in c:\users\admin\anaconda3\lib\site-packages (from pandas>=0.24.2->mlxtend) (2023.3)
```

```
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\admin\anaconda3\lib\site-packages (from scikit-learn>=1.0.2->mlxtend) (2.2.0)
```

```
Requirement already satisfied: six>=1.5 in c:\users\admin\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
```

```
Downloading mlxtend-0.23.1-py3-none-any.whl (1.4 MB)
```

```
----- 0.0/1.4 MB ? eta -::---  
----- 0.0/1.4 MB ? eta -::---  
----- 0.0/1.4 MB 320.0 kB/s eta 0:00:05  
----- 0.0/1.4 MB 326.8 kB/s eta 0:00:05  
----- 0.0/1.4 MB 326.8 kB/s eta 0:00:05  
----- 0.1/1.4 MB 272.3 kB/s eta 0:00:06  
----- 0.1/1.4 MB 327.3 kB/s eta 0:00:05  
----- 0.1/1.4 MB 327.3 kB/s eta 0:00:05  
----- 0.1/1.4 MB 249.0 kB/s eta 0:00:06  
----- 0.1/1.4 MB 301.9 kB/s eta 0:00:05  
----- 0.2/1.4 MB 374.1 kB/s eta 0:00:04  
----- 0.2/1.4 MB 393.0 kB/s eta 0:00:04  
----- 0.2/1.4 MB 389.1 kB/s eta 0:00:04  
----- 0.2/1.4 MB 389.1 kB/s eta 0:00:04  
----- 0.3/1.4 MB 431.3 kB/s eta 0:00:03  
----- 0.3/1.4 MB 452.0 kB/s eta 0:00:03  
----- 0.3/1.4 MB 455.7 kB/s eta 0:00:03  
----- 0.4/1.4 MB 464.2 kB/s eta 0:00:03  
----- 0.4/1.4 MB 464.2 kB/s eta 0:00:03  
----- 0.4/1.4 MB 464.2 kB/s eta 0:00:03  
----- 0.4/1.4 MB 399.8 kB/s eta 0:00:03  
----- 0.4/1.4 MB 433.1 kB/s eta 0:00:03  
----- 0.4/1.4 MB 430.1 kB/s eta 0:00:03  
----- 0.5/1.4 MB 426.7 kB/s eta 0:00:03  
----- 0.5/1.4 MB 442.8 kB/s eta 0:00:03  
----- 0.6/1.4 MB 489.3 kB/s eta 0:00:02  
----- 0.6/1.4 MB 501.8 kB/s eta 0:00:02  
----- 0.7/1.4 MB 516.1 kB/s eta 0:00:02  
----- 0.7/1.4 MB 516.1 kB/s eta 0:00:02
```

```
----- 0.7/1.4 MB 508.5 kB/s eta 0:00:02
----- 0.7/1.4 MB 508.5 kB/s eta 0:00:02
----- 0.7/1.4 MB 505.6 kB/s eta 0:00:02
----- 0.7/1.4 MB 505.6 kB/s eta 0:00:02
----- 0.7/1.4 MB 481.3 kB/s eta 0:00:02
----- 0.8/1.4 MB 491.4 kB/s eta 0:00:02
----- 0.8/1.4 MB 499.2 kB/s eta 0:00:02
----- 0.9/1.4 MB 508.3 kB/s eta 0:00:02
----- 0.9/1.4 MB 531.2 kB/s eta 0:00:01
----- 1.0/1.4 MB 545.9 kB/s eta 0:00:01
----- 1.0/1.4 MB 542.2 kB/s eta 0:00:01
----- 1.1/1.4 MB 562.5 kB/s eta 0:00:01
----- 1.1/1.4 MB 570.1 kB/s eta 0:00:01
----- 1.1/1.4 MB 570.1 kB/s eta 0:00:01
----- 1.2/1.4 MB 568.9 kB/s eta 0:00:01
----- 1.2/1.4 MB 568.9 kB/s eta 0:00:01
----- 1.2/1.4 MB 577.5 kB/s eta 0:00:01
----- 1.3/1.4 MB 579.2 kB/s eta 0:00:01
----- 1.3/1.4 MB 586.1 kB/s eta 0:00:01
----- 1.3/1.4 MB 574.4 kB/s eta 0:00:01
----- 1.4/1.4 MB 588.8 kB/s eta 0:00:01
----- 1.4/1.4 MB 603.6 kB/s eta 0:00:01
----- 1.4/1.4 MB 603.6 kB/s eta 0:00:01
----- 1.4/1.4 MB 583.0 kB/s eta 0:00:00
```

Installing collected packages: mlxtend  
Successfully installed mlxtend-0.23.1

In [5]: `from mlxtend.preprocessing import TransactionEncoder`

In [6]: `te=TransactionEncoder()  
te_try=te.fit(dataset).transform(dataset)  
te_try`

Out[6]: `array([[False, False, False, True, False, True, True, True, True,  
 False, True],  
 [False, False, True, True, False, True, False, True, True,  
 False, True],  
 [True, False, False, True, False, True, True, False, False,  
 False, False],  
 [False, True, False, False, False, True, True, False, False,  
 True, True],  
 [False, True, False, True, True, False, False, False, True,  
 False, False]])`

In [7]: `te.columns_`

Out[7]: `['Apple',  
'Corn',  
'Dill',  
'Eggs',  
'Ice cream',  
'Kidney Beans',  
'Milk',  
'Nutmeg',  
'Onion',  
'Unicorn',  
'Yogurt']`

In [8]: `df=pd.DataFrame(te_try, columns=te.columns_)`  
`df`

Out[8]:	Apple	Corn	Dill	Eggs	Ice cream	Kidney Beans	Milk	Nutmeg	Onion	Unicorn	Yogurt	
0	False	False	False	True	False		True	True	True	True	False	True
1	False	False	True	True	False		True	False	True	True	False	True
2	True	False	False	True	False		True	True	False	False	False	False
3	False	True	False	False	False		True	True	False	False	True	True
4	False	True	False	True		True		True	False	False	True	False

```
In [9]: from mlxtend.frequent_patterns import apriori
apriori(df,min_support=0.6)
```

Out[9]: support itemsets

0	0.8	(3)
1	1.0	(5)
2	0.6	(6)
3	0.6	(8)
4	0.6	(10)
5	0.8	(3, 5)
6	0.6	(8, 3)
7	0.6	(5, 6)
8	0.6	(8, 5)
9	0.6	(10, 5)
10	0.6	(8, 3, 5)

```
In [10]: apriori(df,min_support=0.6,use_colnames=True)
```

Out[10]: support itemsets

	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)
5	0.8	(Eggs, Kidney Beans)
6	0.6	(Onion, Eggs)
7	0.6	(Kidney Beans, Milk)
8	0.6	(Onion, Kidney Beans)
9	0.6	(Kidney Beans, Yogurt)
10	0.6	(Onion, Eggs, Kidney Beans)

```
In [14]: frequent_itemsets=apriori(df,min_support=0.6, use_colnames=True)
frequent_itemsets["length"]=frequent_items[ "itemsets"].apply(lambda x:len(x))
frequent_itemsets
```

	<b>support</b>	<b>itemsets</b>	<b>length</b>
<b>0</b>	0.8	(Eggs)	1
<b>1</b>	1.0	(Kidney Beans)	1
<b>2</b>	0.6	(Milk)	1
<b>3</b>	0.6	(Onion)	1
<b>4</b>	0.6	(Yogurt)	1
<b>5</b>	0.8	(Eggs, Kidney Beans)	2
<b>6</b>	0.6	(Onion, Eggs)	2
<b>7</b>	0.6	(Kidney Beans, Milk)	2
<b>8</b>	0.6	(Onion, Kidney Beans)	2
<b>9</b>	0.6	(Kidney Beans, Yogurt)	2
<b>10</b>	0.6	(Onion, Eggs, Kidney Beans)	3

```
In [15]: frequent_itemsets[ (frequent_itemsets['length'] == 2) & (frequent_itemsets['support'] >= 0.6) ]
```

	<b>support</b>	<b>itemsets</b>	<b>length</b>
<b>5</b>	0.8	(Eggs, Kidney Beans)	2
<b>6</b>	0.6	(Onion, Eggs)	2
<b>7</b>	0.6	(Kidney Beans, Milk)	2
<b>8</b>	0.6	(Onion, Kidney Beans)	2
<b>9</b>	0.6	(Kidney Beans, Yogurt)	2

```
In [16]: frequent_itemsets[ frequent_itemsets['itemsets'] == {'Onion', 'Kidney Beans'} ]
```

	<b>support</b>	<b>itemsets</b>	<b>length</b>
<b>8</b>	0.6	(Onion, Kidney Beans)	2

```
In [17]: oht_ary= te.fit(dataset).transform(dataset,sparse=True)
sparse_df=pd.DataFrame(te_ary,columns=te.columns_)
sparse_df
```

	<b>Apple</b>	<b>Corn</b>	<b>Dill</b>	<b>Eggs</b>	<b>Ice cream</b>	<b>Kidney Beans</b>	<b>Milk</b>	<b>Nutmeg</b>	<b>Onion</b>	<b>Unicorn</b>	<b>Yogurt</b>	
<b>0</b>	False	False	False	True	False		True	True	True	True	False	True
<b>1</b>	False	False	True	True	False		True	False	True	True	False	True
<b>2</b>	True	False	False	True	False		True	True	False	False	False	False
<b>3</b>	False	True	False	False	False		True	True	False	False	True	True
<b>4</b>	False	True	False	True		True		True	False	False	True	False

```
In [18]: apriori(sparse_df,min_support=0.6,use_colnames=True, verbose=1)
```

Processing 21 combinations | Sampling itemset size 3

	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)
5	0.8	(Eggs, Kidney Beans)
6	0.6	(Onion, Eggs)
7	0.6	(Kidney Beans, Milk)
8	0.6	(Onion, Kidney Beans)
9	0.6	(Kidney Beans, Yogurt)
10	0.6	(Onion, Eggs, Kidney Beans)

```
In [19]: apriori(sparse_df, min_support=0.3, use_colnames=True, verbose=1, max_len=4)
```

Processing 32 combinations | Sampling itemset size 4

Out[19]:	support	itemsets
<b>0</b>	0.4	(Corn)
<b>1</b>	0.8	(Eggs)
<b>2</b>	1.0	(Kidney Beans)
<b>3</b>	0.6	(Milk)
<b>4</b>	0.4	(Nutmeg)
<b>5</b>	0.6	(Onion)
<b>6</b>	0.6	(Yogurt)
<b>7</b>	0.4	(Kidney Beans, Corn)
<b>8</b>	0.8	(Eggs, Kidney Beans)
<b>9</b>	0.4	(Eggs, Milk)
<b>10</b>	0.4	(Eggs, Nutmeg)
<b>11</b>	0.6	(Onion, Eggs)
<b>12</b>	0.4	(Eggs, Yogurt)
<b>13</b>	0.6	(Kidney Beans, Milk)
<b>14</b>	0.4	(Kidney Beans, Nutmeg)
<b>15</b>	0.6	(Onion, Kidney Beans)
<b>16</b>	0.6	(Kidney Beans, Yogurt)
<b>17</b>	0.4	(Milk, Yogurt)
<b>18</b>	0.4	(Onion, Nutmeg)
<b>19</b>	0.4	(Nutmeg, Yogurt)
<b>20</b>	0.4	(Onion, Yogurt)
<b>21</b>	0.4	(Eggs, Kidney Beans, Milk)
<b>22</b>	0.4	(Eggs, Nutmeg, Kidney Beans)
<b>23</b>	0.6	(Onion, Eggs, Kidney Beans)
<b>24</b>	0.4	(Eggs, Kidney Beans, Yogurt)
<b>25</b>	0.4	(Onion, Eggs, Nutmeg)
<b>26</b>	0.4	(Eggs, Nutmeg, Yogurt)
<b>27</b>	0.4	(Onion, Eggs, Yogurt)
<b>28</b>	0.4	(Kidney Beans, Milk, Yogurt)
<b>29</b>	0.4	(Onion, Nutmeg, Kidney Beans)
<b>30</b>	0.4	(Kidney Beans, Nutmeg, Yogurt)
<b>31</b>	0.4	(Onion, Kidney Beans, Yogurt)
<b>32</b>	0.4	(Onion, Nutmeg, Yogurt)
<b>33</b>	0.4	(Onion, Eggs, Nutmeg, Kidney Beans)
<b>34</b>	0.4	(Eggs, Nutmeg, Kidney Beans, Yogurt)
<b>35</b>	0.4	(Onion, Eggs, Kidney Beans, Yogurt)
<b>36</b>	0.4	(Onion, Eggs, Nutmeg, Yogurt)
<b>37</b>	0.4	(Onion, Nutmeg, Kidney Beans, Yogurt)

In [ ]: