

7vxhxofkb

April 29, 2024

```
[ ]: import pandas as pd
import numpy as np
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import association_rules
from mlxtend.frequent_patterns import fpgrowth
```

```
[ ]: data = pd.read_csv('/content/Market_Basket_Optimisation (2).csv')
data.shape
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during the transform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)

```
[ ]: (7500, 20)
```

```
[ ]: transaction = []
for i in range(0, data.shape[0]):
    for j in range(0, data.shape[1]):
        transaction.append(data.values[i,j])

transaction = np.array(transaction)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during the transform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)

```
[ ]: print(transaction)
```

```
['burgers' 'meatballs' 'eggs' ... 'nan' 'nan' 'nan']
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`

argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)

```
[ ]: data.head()
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)
```

```
[ ]:      shrimp      almonds      avocado      vegetables mix green grapes \
0      burgers meatballs      eggs      NaN      NaN
1      chutney      NaN      NaN      NaN      NaN
2      turkey      avocado      NaN      NaN      NaN
3  mineral water      milk  energy bar  whole wheat rice      green tea
4  low fat yogurt      NaN      NaN      NaN      NaN

      whole weat flour yams cottage cheese energy drink tomato juice \
0      NaN  NaN      NaN      NaN      NaN      NaN
1      NaN  NaN      NaN      NaN      NaN      NaN
2      NaN  NaN      NaN      NaN      NaN      NaN
3      NaN  NaN      NaN      NaN      NaN      NaN
4      NaN  NaN      NaN      NaN      NaN      NaN

      low fat yogurt green tea honey salad mineral water salmon antioxydant juice \
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

      frozen smoothie spinach  olive oil
0      NaN      NaN      NaN
1      NaN      NaN      NaN
2      NaN      NaN      NaN
3      NaN      NaN      NaN
4      NaN      NaN      NaN
```

```
[ ]: df = pd.DataFrame(transaction, columns=["items"])

df["incident_count"] = 1

indexNames = df[df['items'] == "nan" ].index
df.drop(indexNames , inplace=True)
```

```
df_table = df.groupby("items").sum().sort_values("incident_count",
↪ascending=False).reset_index()
df_table.head(5)
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
 DeprecationWarning: `should_run_async` will not call `transform_cell`
 automatically in the future. Please pass the result to `transformed_cell`
 argument and any exception that happen during the transform in
 `preprocessing_exc_tuple` in IPython 7.17 and above.
 and should_run_async(code)

```
[ ]:      items  incident_count
0  mineral water          1787
1         eggs           1348
2    spaghetti           1306
3  french fries           1282
4     chocolate           1230
```

```
[ ]: transaction = []
for i in range(data.shape[0]):
    transaction.append([str(data.values[i,j]) for j in range(data.shape[1])])
transaction = np.array(transaction)
te = TransactionEncoder()
te_array = te.fit(transaction).transform(transaction)
df = pd.DataFrame(te_array, columns=te.columns_)
df.head()
```

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
 DeprecationWarning: `should_run_async` will not call `transform_cell`
 automatically in the future. Please pass the result to `transformed_cell`
 argument and any exception that happen during the transform in
 `preprocessing_exc_tuple` in IPython 7.17 and above.
 and should_run_async(code)

```
[ ]:      asparagus  almonds  antioxydant juice  asparagus  avocado  babies food \
0      False      False                False      False      False      False
1      False      False                False      False      False      False
2      False      False                False      False      True       False
3      False      False                False      False      False      False
4      False      False                False      False      False      False

      bacon  barbecue sauce  black tea  blueberries  ...  turkey  vegetables mix \
0  False                False      False      False  ...  False                False
1  False                False      False      False  ...  False                False
2  False                False      False      False  ...  True                 False
3  False                False      False      False  ...  False                False
4  False                False      False      False  ...  False                False
```

	water spray	white wine	whole weat flour	whole wheat pasta	\
0	False	False	False	False	
1	False	False	False	False	
2	False	False	False	False	
3	False	False	False	False	
4	False	False	False	False	

	whole wheat rice	yams	yogurt cake	zucchini
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	True	False	False	False
4	False	False	False	False

[5 rows x 121 columns]

```
[ ]: frequent_itemsets = fpgrowth(df, min_support=0.1, use_colnames=True)
frequent_itemsets
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during thetransform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)
```

```
[ ]:      support      itemsets
0    1.000000      (nan)
1    0.179733      (eggs)
2    0.238267 (mineral water)
3    0.132000 (green tea)
4    0.129600      (milk)
5    0.170933 (french fries)
6    0.174133 (spaghetti)
7    0.163867 (chocolate)
8    0.179733 (eggs, nan)
9    0.238267 (nan, mineral water)
10   0.132000 (green tea, nan)
11   0.129600 (milk, nan)
12   0.170933 (nan, french fries)
13   0.174133 (spaghetti, nan)
14   0.163867 (nan, chocolate)
```

```
[ ]: frequent_itemsets=association_rules(frequent_itemsets, metric="lift",
↳min_threshold=1)
frequent_itemsets
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during the transform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)
```

```
[ ]: antecedents consequents antecedent support consequent support \
0 (eggs) (nan) 0.179733 1.000000
1 (nan) (eggs) 1.000000 0.179733
2 (nan) (mineral water) 1.000000 0.238267
3 (mineral water) (nan) 0.238267 1.000000
4 (green tea) (nan) 0.132000 1.000000
5 (nan) (green tea) 1.000000 0.132000
6 (milk) (nan) 0.129600 1.000000
7 (nan) (milk) 1.000000 0.129600
8 (nan) (french fries) 1.000000 0.170933
9 (french fries) (nan) 0.170933 1.000000
10 (spaghetti) (nan) 0.174133 1.000000
11 (nan) (spaghetti) 1.000000 0.174133
12 (nan) (chocolate) 1.000000 0.163867
13 (chocolate) (nan) 0.163867 1.000000
```

	support	confidence	lift	leverage	conviction	zhangs_metric
0	0.179733	1.000000	1.0	0.0	inf	0.0
1	0.179733	0.179733	1.0	0.0	1.0	0.0
2	0.238267	0.238267	1.0	0.0	1.0	0.0
3	0.238267	1.000000	1.0	0.0	inf	0.0
4	0.132000	1.000000	1.0	0.0	inf	0.0
5	0.132000	0.132000	1.0	0.0	1.0	0.0
6	0.129600	1.000000	1.0	0.0	inf	0.0
7	0.129600	0.129600	1.0	0.0	1.0	0.0
8	0.170933	0.170933	1.0	0.0	1.0	0.0
9	0.170933	1.000000	1.0	0.0	inf	0.0
10	0.174133	1.000000	1.0	0.0	inf	0.0
11	0.174133	0.174133	1.0	0.0	1.0	0.0
12	0.163867	0.163867	1.0	0.0	1.0	0.0
13	0.163867	1.000000	1.0	0.0	inf	0.0

```
[ ]: frequent_itemsets.sort_values("confidence",ascending=False)
```

```
/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel.py:283:
DeprecationWarning: `should_run_async` will not call `transform_cell`
automatically in the future. Please pass the result to `transformed_cell`
argument and any exception that happen during the transform in
`preprocessing_exc_tuple` in IPython 7.17 and above.
and should_run_async(code)
```

```
[ ]:      antecedents      consequents antecedent support consequent support \
0          (eggs)          (nan)          0.179733          1.000000
3  (mineral water)          (nan)          0.238267          1.000000
4          (green tea)          (nan)          0.132000          1.000000
6          (milk)          (nan)          0.129600          1.000000
9  (french fries)          (nan)          0.170933          1.000000
10         (spaghetti)          (nan)          0.174133          1.000000
13        (chocolate)          (nan)          0.163867          1.000000
2          (nan) (mineral water)          1.000000          0.238267
1          (nan)          (eggs)          1.000000          0.179733
11         (nan)        (spaghetti)          1.000000          0.174133
8          (nan)        (french fries)          1.000000          0.170933
12         (nan)        (chocolate)          1.000000          0.163867
5          (nan)        (green tea)          1.000000          0.132000
7          (nan)          (milk)          1.000000          0.129600
```

```
      support confidence lift leverage conviction zhangs_metric
0  0.179733    1.000000    1.0      0.0          inf          0.0
3  0.238267    1.000000    1.0      0.0          inf          0.0
4  0.132000    1.000000    1.0      0.0          inf          0.0
6  0.129600    1.000000    1.0      0.0          inf          0.0
9  0.170933    1.000000    1.0      0.0          inf          0.0
10 0.174133    1.000000    1.0      0.0          inf          0.0
13 0.163867    1.000000    1.0      0.0          inf          0.0
2  0.238267    0.238267    1.0      0.0          1.0          0.0
1  0.179733    0.179733    1.0      0.0          1.0          0.0
11 0.174133    0.174133    1.0      0.0          1.0          0.0
8  0.170933    0.170933    1.0      0.0          1.0          0.0
12 0.163867    0.163867    1.0      0.0          1.0          0.0
5  0.132000    0.132000    1.0      0.0          1.0          0.0
7  0.129600    0.129600    1.0      0.0          1.0          0.0
```

```
[ ]:
```