

```
import numpy as np # linear algebra
import pandas as pd # data processing

data = pd.read_csv("/content/Groceries_dataset.csv")
data.head()
```

| | Member_number | Date | itemDescription |
|---|---------------|------------|------------------|
| 0 | 1808 | 21-07-2015 | tropical fruit |
| 1 | 2552 | 05-01-2015 | whole milk |
| 2 | 2300 | 19-09-2015 | pip fruit |
| 3 | 1187 | 12-12-2015 | other vegetables |
| 4 | 3037 | 01-02-2015 | whole milk |

```
%pip install pyfpgrowth
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/
Collecting pyfpgrowth
  Downloading pyfpgrowth-1.0.tar.gz (1.6 MB)
    1.6/1.6 MB 18.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pyfpgrowth
  Building wheel for pyfpgrowth (setup.py) ... done
  Created wheel for pyfpgrowth: filename=pyfpgrowth-1.0-py2.py3-none-any.whl size=5504 sha2
  Stored in directory: /root/.cache/pip/wheels/09/fc/dc/afff211038bfc745722d8d7e846e854e579
Successfully built pyfpgrowth
Installing collected packages: pyfpgrowth
Successfully installed pyfpgrowth-1.0
```

```
import pyfpgrowth
```

```
#encoding
products = data['itemDescription'].unique()
dummy = pd.get_dummies(data['itemDescription'])
data.drop(['itemDescription'], inplace =True, axis=1)

data = data.join(dummy)

data.head()
```

| | Member_number | Date | Instant food products | UHT- milk | abrasive cleaner | artif. sweetener | baby cosmetics | bags | bakery products |
|---|---------------|------------|-----------------------------|--------------|---------------------|---------------------|-------------------|------|--------------------|
| 0 | 1808 | 21-07-2015 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 2552 | 05-01-2015 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | 2300 | 19-09-2015 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 1187 | 12-12-2015 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 3037 | 01-02-2015 | 0 | 0 | 0 | 0 | 0 | 0 | |

5 rows x 169 columns

```
data1 = data.groupby(['Member_number', 'Date'])[products[:]].sum()
data1 = data1.reset_index()[products]

print("New Dimension", data1.shape)
data1.head()
```

New Dimension (14963, 167)

| | tropical fruit | whole milk | pip fruit | other vegetables | rolls/buns | pot plants | citrus fruit | beef | frankfur |
|---|-------------------|---------------|--------------|---------------------|------------|---------------|-----------------|------|----------|
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

5 rows x 167 columns

```
def productnames(x):
    for product in products:
        if x[product] >0:
            x[product] = product
    return x

data1 = data1.apply(productnames, axis=1)
data1.head()
```

| | tropical fruit | whole milk | pip fruit | other vegetables | rolls/buns | pot plants | citrus fruit | beef | frankfu |
|---|-------------------|---------------|--------------|---------------------|------------|---------------|-----------------|------|---------|
| 0 | 0 | whole milk | 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | whole milk | 0 | 0 | 0 | 0 | 0 | 0 | |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

5 rows x 167 columns

```
# Extracting the list of items bought per customer
```

```
x = data1.values
x = [sub[~(sub==0)].tolist() for sub in x if sub [sub != 0].tolist()]
transactions = x
transactions[0:10]
```

```
[['whole milk', 'yogurt', 'sausage', 'semi-finished bread'],
 ['whole milk', 'pastry', 'salty snack'],
 ['canned beer', 'misc. beverages'],
 ['sausage', 'hygiene articles'],
 ['soda', 'pickled vegetables'],
 ['frankfurter', 'curd'],
 ['whole milk', 'rolls/buns', 'sausage'],
 ['whole milk', 'soda'],
 ['beef', 'white bread'],
 ['frankfurter', 'soda', 'whipped/sour cream']]
```

```
# Identify frequent itemsets
```

```
patterns = pyfpgrowth.find_frequent_patterns(transactions, 10)
patterns
sorted_patterns = {key: value for key, value in sorted(patterns.items(), key=lambda item: item
sorted_patterns
```

```
{('whole milk',): 2363,
 ('other vegetables',): 1827,
 ('other vegetables', 'whole milk'): 222,
 ('rolls/buns', 'whole milk'): 209,
 ('soda', 'whole milk'): 174,
 ('whole milk', 'yogurt'): 167,
 ('other vegetables', 'rolls/buns'): 158,
 ('other vegetables', 'soda'): 145,
 ('sausage', 'whole milk'): 134,
 ('packaged fruit/vegetables',): 127,
 ('tropical fruit', 'whole milk'): 123,
 ('other vegetables', 'yogurt'): 121,
 ('rolls/buns', 'yogurt'): 117,
 ('root vegetables', 'whole milk'): 113,
 ('bottled beer', 'whole milk'): 107,
 ('citrus fruit', 'whole milk'): 107,
 ('bottled water', 'whole milk'): 107,
 ('pip fruit', 'whole milk'): 99,
 ('pastry', 'whole milk'): 97,
 ('shopping bags', 'whole milk'): 95,
 ('other vegetables', 'tropical fruit'): 94,
 ('frozen dessert',): 92,
 ('rolls/buns', 'tropical fruit'): 91,
```

```
( 'canned beer', 'whole milk'): 90,
( 'other vegetables', 'sausage'): 90,
( 'salt',): 89,
( 'soda', 'yogurt'): 87,
( 'rolls/buns', 'root vegetables'): 86,
( 'pet care',): 85,
( 'newspapers', 'whole milk'): 84,
( 'roll products ',): 82,
( 'canned vegetables',): 82,
( 'bottled water', 'other vegetables'): 82,
( 'soda', 'tropical fruit'): 81,
( 'turkey',): 80,
( 'photo/film',): 79,
( 'domestic eggs', 'whole milk'): 79,
( 'frankfurter', 'whole milk'): 79,
( 'root vegetables', 'soda'): 79,
( 'other vegetables', 'root vegetables'): 79,
( 'tropical fruit', 'yogurt'): 78,
( 'frankfurter', 'other vegetables'): 77,
( 'mayonnaise',): 75,
( 'pork', 'whole milk'): 75,
( 'cling film/bags',): 74,
( 'other vegetables', 'shopping bags'): 74,
( 'pip fruit', 'rolls/buns'): 74,
( 'other vegetables', 'pip fruit'): 74,
( 'dish cleaner',): 73,
( 'frozen potato products',): 72,
( 'specialty cheese',): 72,
( 'citrus fruit', 'other vegetables'): 72,
( 'bottled water', 'soda'): 72,
( 'rolls/buns', 'shopping bags'): 71,
( 'beef', 'whole milk'): 70,
( 'butter', 'whole milk'): 70,
( 'bottled beer', 'other vegetables'): 70,
( 'citrus fruit', 'rolls/buns'): 70,
```

```
rules = pyfpgrowth.generate_association_rules(patterns, 0.02)
rules
```

```
↳ ( 'house keeping products',): (('whole milk',), 0.24444444444444444),
( 'chocolate marshmallow',): (('whole milk',), 0.16666666666666666),
( 'finished products',): (('whole milk',), 0.203125),
( 'candles',): (('whole milk',), 0.16666666666666666),
( 'dog food',): (('whole milk',), 0.14925373134328357),
( 'dish cleaner',): (('whole milk',), 0.1780821917808219),
( 'cling film/bags',): (('whole milk',), 0.13513513513513514),
( 'turkey',): (('whole milk',), 0.1375),
( 'roll products ',): (('whole milk',), 0.12195121951219512),
( 'canned vegetables',): (('whole milk',), 0.17073170731707318),
( 'pet care',): (('whole milk',), 0.12941176470588237),
( 'frozen dessert',): (('whole milk',), 0.15217391304347827),
( 'packaged fruit/vegetables',): (('rolls/buns',), 0.14173228346456693),
( 'other vegetables',): (('whole milk',), 0.12151067323481117),
( 'whole milk',): (('other vegetables',), 0.09394837071519255),
( 'frankfurter', 'other vegetables'): (('whole milk',), 0.15584415584415584),
( 'frankfurter', 'whole milk'): (('other vegetables',), 0.1518987341772152),
( 'other vegetables', 'whole milk'): (('rolls/buns',), 0.08108108108108109),
( 'bottled beer', 'rolls/buns'): (('whole milk',), 0.16666666666666666),
( 'bottled beer', 'whole milk'): (('rolls/buns',), 0.09345794392523364),
( 'rolls/buns', 'whole milk'): (('other vegetables',), 0.0861244019138756),
( 'canned beer', 'rolls/buns'): (('whole milk',), 0.15873015873015872),
( 'canned beer', 'whole milk'): (('rolls/buns',), 0.11111111111111111),
( 'rolls/buns', 'shopping bags'): (('soda',), 0.14084507042253522),
( 'shopping bags', 'soda'): (('rolls/buns',), 0.15151515151515152),
( 'pip fruit', 'rolls/buns'): (('whole milk',), 0.13513513513513514),
( 'pip fruit', 'whole milk'): (('rolls/buns',), 0.10101010101010101),
( 'pastry', 'sausage'): (('whole milk',), 0.22916666666666666),
```

```
( 'other vegetables', 'pastry' ): (('whole milk',), 0.10101010101010102),  
( 'pastry', 'rolls/buns' ): (('whole milk',), 0.1864406779661017),  
( 'pastry', 'soda' ): (('whole milk',), 0.22950819672131148),  
( 'soda', 'whole milk' ): (('other vegetables',), 0.09770114942528736),  
( 'citrus fruit', 'whole milk' ): (('rolls/buns',), 0.09345794392523364),  
( 'citrus fruit', 'yogurt' ): (('whole milk',), 0.14492753623188406),  
( 'whole milk', 'yogurt' ): (('other vegetables',), 0.10179640718562874),  
( 'citrus fruit', 'rolls/buns' ): (('whole milk',), 0.14285714285714285),  
( 'other vegetables', 'rolls/buns' ): (('whole milk',), 0.11392405063291139),  
( 'other vegetables', 'sausage' ): (('whole milk',), 0.15555555555555556),  
( 'soda', 'yogurt' ): (('whole milk',), 0.16091954022988506),  
( 'other vegetables', 'soda' ): (('whole milk',), 0.11724137931034483),  
( 'bottled water', 'soda' ): (('whole milk',), 0.16666666666666666),  
( 'bottled water', 'whole milk' ): (('soda',), 0.11214953271028037),  
( 'tropical fruit', 'whole milk' ): (('other vegetables',),  
0.08943089430894309),  
( 'tropical fruit', 'yogurt' ): (('whole milk',), 0.15384615384615385),  
( 'soda', 'tropical fruit' ): (('whole milk',), 0.12345679012345678),  
( 'rolls/buns', 'tropical fruit' ): (('whole milk',), 0.10989010989010989),  
( 'other vegetables', 'tropical fruit' ): (('whole milk',),  
0.11702127659574468),  
( 'root vegetables', 'whole milk' ): (('other vegetables',),  
0.08849557522123894).
```