```
# Step 1: Import necessary libraries
import pandas as pd
!pip install apyori
from apyori import apriori
```

```
Collecting apyori
  Downloading apyori-1.1.2.tar.gz (8.6 kB)
  Preparing metadata (setup.py) ... e=apyori-1.1.2-py3-none-any.whl
size=5953
sha256=abac294162e660f49a27d472b642312cb7fc68b86a42bc601964e2336ea9b4e
c
  Stored in directory:
/root/.cache/pip/wheels/c4/1a/79/20f55c470a50bb3702a8cb7c94d8ada155735
38c7f4baebe2d
Successfully built apyori
Installing collected packages: apyori
Successfully installed apyori-1.1.2
```

```
# Step 2: Load the dataset
data=pd.read_csv(r"/content/Market_Basket_Optimisation -
Copy.csv",header=None)
```

```
data
```

```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 7501,\n  \"fields\":
[\n    {\n      \"column\": 0,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 115,\n
\"samples\": [\n          \"gums\",\n          \"mineral water\",\n
\"pancakes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 1,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 117,\n        \"samples\": [\n
\"tomatoes\",\n          \"french fries\",\n          \"eggplant\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 2,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 115,\n
\"samples\": [\n          \"mint\",\n          \"green tea\",\n
\"mashed potato\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 3,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 114,\n        \"samples\": [\n
\"toothpaste\",\n          \"spaghetti\",\n          \"cottage
cheese\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 4,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 110,\n        \"samples\": [\n
\"magazines\",\n          \"energy bar\",\n          \"mineral
water\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 5,\n
\"properties\": {\n        \"dtype\": \"category\",\n
```

```
\"num_unique_values\": 106,\n        \"samples\": [\n          \"flax
seed\",\n          \"energy drink\",\n          \"green tea\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 6,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 102,\n
\"samples\": [\n          \"energy drink\",\n          \"zucchini\",\n
\"extra dark chocolate\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": 7,\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 98,\n
\"samples\": [\n          \"whole weat flour\",\n          \"pasta\",\
n          \"body spray\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": 8,\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 88,\n
\"samples\": [\n          \"strawberries\",\n          \"energy
drink\",\n          \"mashed potato\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 9,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 80,\n
\"samples\": [\n          \"rice\",\n          \"tomato juice\",\n
\"cake\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 10,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 66,\n        \"samples\": [\n          \"strong
cheese\",\n          \"bramble\",\n          \"low fat yogurt\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 11,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 50,\n
\"samples\": [\n          \"frozen smoothie\",\n          \"green
grapes\",\n          \"brownies\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 12,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 43,\n
\"samples\": [\n          \"magazines\",\n          \"burger sauce\",\
n          \"white wine\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": 13,\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 28,\n
\"samples\": [\n          \"pancakes\",\n          \"yogurt cake\",\n
\"toothpaste\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 14,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 19,\n        \"samples\": [\n
\"mineral water\",\n          \"pancakes\",\n          \"fresh
bread\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 15,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 8,\n        \"samples\": [\n
\"chocolate\",\n          \"sparkling water\",\n          \"salmon\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 16,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"antioxydant juice\",\n          \"frozen
smoothie\",\n          \"french fries\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 17,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"frozen smoothie\",\n          \"protein
bar\",\n          \"spinach\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": 18,\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"spinach\",\n          \"mayonnaise\",\n
\"cereals\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 19,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          \"olive
oil\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"data"}
```

```
# Step 3: Check for missing values
data.isnull().sum()
```

```
0        0
1     1754
2     3112
3     4156
4     4972
5     5637
6     6132
7     6520
8     6847
9     7106
10    7245
11    7347
12    7414
13    7454
14    7476
15    7493
16    7497
17    7497
18    7498
19    7500
dtype: int64
```

```
# Step 4: Fill missing values with 0
data.fillna(0,inplace=True)
```

```
# Step 5: Display the first few rows of the data
data.head()
```

```
{"summary":"{\n  \"name\": \"data\",\n  \"rows\": 7501,\n  \"fields\":
[\n    {\n      \"column\": 0,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 115,\n
\"samples\": [\n          \"gums\",\n          \"mineral water\",\n
\"pancakes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 1,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 118,\n        \"samples\": [\n
\"bacon\",\n          \"shampoo\",\n          \"milk\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 2,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 116,\n
\"samples\": [\n          \"zucchini\",\n          \"shallot\",\n
\"tomato sauce\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 3,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 115,\n        \"samples\": [\n
\"toothpaste\",\n          \"honey\",\n          \"ground beef\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 4,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 111,\n
\"samples\": [\n          \"cider\",\n          \"milk\",\n
\"oil\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 5,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 107,\n        \"samples\": [\n
\"tomato juice\",\n          \"spaghetti\",\n          \"chocolate\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 6,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 103,\n
\"samples\": [\n          \"champagne\",\n          \"chili\",\n
\"protein bar\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 7,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 99,\n        \"samples\": [\n          \"black
tea\",\n          \"tea\",\n          \"body spray\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 8,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 89,\n
\"samples\": [\n          \"sparkling water\",\n
\"asparagus\",\n          \"salmon\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 9,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 81,\n
\"samples\": [\n          \"fromage blanc\",\n          \"tomato
juice\",\n          \"muffins\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 10,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 67,\n
\"samples\": [\n          \"gums\",\n          \"chocolate\",\n
\"pancakes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 11,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 51,\n        \"samples\": [\n          \"strong
cheese\",\n          \"green grapes\",\n          \"cooking oil\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 12,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 44,\n
\"samples\": [\n          \"asparagus\",\n          \"cooking oil\",\n
\"burger sauce\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 13,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 29,\n        \"samples\": [\n
\"cookies\",\n          \"frozen smoothie\",\n          \"muffins\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": 14,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 20,\n
\"samples\": [\n          \"mineral water\",\n          \"yogurt
cake\",\n          \"low fat yogurt\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 15,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 9,\n
\"samples\": [\n          \"protein bar\",\n          0,\n
\"cake\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 16,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          0,\n
\"french fries\",\n          \"antioxydant juice\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 17,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 4,\n
\"samples\": [\n          0,\n          \"spinach\",\n
\"frozen smoothie\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": 18,\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 4,\n        \"samples\": [\n          0,\n
\"cereals\",\n          \"spinach\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": 19,\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          0,\n          \"olive oil\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"data"}

# Step 6: Prepare the transactions list
transactions = []
```

```
for i in range(0, 7501):
  transactions.append([str(data.values[i,j]) for j in range(0, 20)])

# transactions list
transactions[0]
```

```
['shrimp',
 'almonds',
 'avocado',
 'vegetables mix',
 'green grapes',
 'whole weat flour',
 'yams',
 'cottage cheese',
 'energy drink',
 'tomato juice',
 'low fat yogurt',
 'green tea',
 'honey',
 'salad',
 'mineral water',
 'salmon',
 'antioxydant juice',
 'frozen smoothie',
 'spinach',
 'olive oil']
```

```
rule_list = apriori(transactions, min_support = 0.003,  min_lift = 3,
min_length = 2)

rule_list
```

```
<generator object apriori at 0x7d48e44360a0>
```

```
print("Association Rules with Antecedents, Consequents, Lift, and
Support:\n")

max_rules_to_print = 10
rules_printed = 0

for rule in Results:
    for ordered_statistics in rule.ordered_statistics:
        antecedent = list(ordered_statistics.items_base)
        consequent = list(ordered_statistics.items_add)
        support = rule.support
        lift = ordered_statistics.lift

        print(f"Antecedent: {antecedent}, Consequent: {consequent},
Support: {support:.4f}, Lift: {lift:.4f}")

        rules_printed += 1
        if rules_printed >= max_rules_to_print:
            break

    if rules_printed >= max_rules_to_print:
        break
```

```
Association Rules with Antecedents, Consequents, Lift, and Support:

Antecedent: ['brownies'], Consequent: ['cottage cheese'], Support:
0.0035, Lift: 3.2253
Antecedent: ['cottage cheese'], Consequent: ['brownies'], Support:
0.0035, Lift: 3.2253
Antecedent: ['chicken'], Consequent: ['light cream'], Support: 0.0045,
Lift: 4.8440
Antecedent: ['light cream'], Consequent: ['chicken'], Support: 0.0045,
Lift: 4.8440
Antecedent: ['escalope'], Consequent: ['mushroom cream sauce'],
Support: 0.0057, Lift: 3.7908
Antecedent: ['mushroom cream sauce'], Consequent: ['escalope'],
Support: 0.0057, Lift: 3.7908
Antecedent: ['escalope'], Consequent: ['pasta'], Support: 0.0059,
Lift: 4.7008
Antecedent: ['pasta'], Consequent: ['escalope'], Support: 0.0059,
Lift: 4.7008
Antecedent: ['fresh bread'], Consequent: ['tomato juice'], Support:
0.0043, Lift: 3.2594
Antecedent: ['tomato juice'], Consequent: ['fresh bread'], Support:
0.0043, Lift: 3.2594
```

```
Results = list(rule_list)
```

```
print(f"Number of rules: {len(Results)}")
```

```
Number of rules: 188
```

```
results = pd.DataFrame(Results)
results.head()
```

{"summary":"{\n \"name\": \"results\",\n \"rows\": 188,\n
\"fields\": [\n   {\n   \"column\": \"items\",\n
\"properties\": {\n   \"dtype\": \"string\",\n
\"num_unique_values\": 188,\n   \"samples\": [\n
\"frozenset({'mineral water', '0', 'spaghetti', 'pancakes', 'ground
beef'})\",\n   \"frozenset({'0', 'spaghetti', 'frozen
vegetables', 'milk', 'chocolate'})\",\n   \"frozenset({'fromage
blanc', '0', 'honey'})\"\n   ],\n   \"semantic_type\":
\"\",\n   \"description\": \"\"\n   }\n   },\n   {\n
\"column\": \"support\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.001768423204432923,\n   \"min\":
0.0030662578322890282,\n   \"max\": 0.015997866951073192,\n
\"num_unique_values\": 27,\n   \"samples\": [\n

```
0.005332622317024397,\n          0.0030662578322890282,\n
0.003199573390214638\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"ordered_statistics\",\n      \"properties\": {\n        \"dtype\":
\"object\",\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"results"}
```