Assignment 4

Title: Association Rules

Name: Pranjal Rane

NUID: 002756852

In [7]:

```python
import matplotlib.pyplot as plt
import pandas as pd
import csv
import numpy as np
from IPython.display import clear_output
import pyfpgrowth
```

Helper Functions for implementing the Apriori

In [2]:

```python
# (c) 2016 Everaldo Aguiar & Reid Johnson
#
# Modified from:
# Marcel Caraciolo (https://gist.github.com/marcelcaraciolo/1423287)
#
# Functions to compute and extract association rules from a given frequent itemset
# generated by the Apriori algorithm.
#
# The Apriori algorithm is defined by Agrawal and Srikant in:
# Fast algorithms for mining association rules
# Proc. 20th int. conf. very large data bases, VLDB. Vol. 1215. 1994

def load_dataset(filename):
    '''Loads an example of market basket transactions from a provided csv file.

    Returns: A list (database) of lists (transactions). Each element of a transaction is
    an item.
    '''

    with open(filename,'r') as dest_f:
        data_iter = csv.reader(dest_f, delimiter = ',', quotechar = '"')
        data = [data for data in data_iter]
        data_array = np.asarray(data, dtype=object)

    return data_array

def apriori(dataset, min_support=0.5, verbose=False):
    """Implements the Apriori algorithm.

    The Apriori algorithm will iteratively generate new candidate
    k-itemsets using the frequent (k-1)-itemsets found in the previous
    iteration.

    Parameters
    ----------
    dataset : list
        The dataset (a list of transactions) from which to generate
        candidate itemsets.

    min_support : float
        The minimum support threshold. Defaults to 0.5.

    Returns
```

```python
    -------
    F : list
        The list of frequent itemsets.

    support_data : dict
        The support data for all candidate itemsets.

    References
    ----------
    .. [1] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association
           Rules", 1994.

    """
    C1 = create_candidates(dataset)
    D = list(map(set, dataset))
    F1, support_data = support_prune(D, C1, min_support, verbose=False) # prune candidat
e 1-itemsets
    F = [F1] # list of frequent itemsets; initialized to frequent 1-itemsets
    k = 2 # the itemset cardinality
    while (len(F[k - 2]) > 0):
        Ck = apriori_gen(F[k-2], k) # generate candidate itemsets
        Fk, supK = support_prune(D, Ck, min_support) # prune candidate itemsets
        support_data.update(supK) # update the support counts to reflect pruning
        F.append(Fk) # add the pruned candidate itemsets to the list of frequent itemsets
        k += 1

    if verbose:
        # Print a list of all the frequent itemsets.
        for kset in F:
            for item in kset:
                print("" \
                    + "{" \
                    + "".join(str(i) + ", " for i in iter(item)).rstrip(', ') \
                    + "}" \
                    + ":  sup = " + str(round(support_data[item], 3)))

    return F, support_data

def create_candidates(dataset, verbose=False):
    """Creates a list of candidate 1-itemsets from a list of transactions.

    Parameters
    ----------
    dataset : list
        The dataset (a list of transactions) from which to generate candidate
        itemsets.

    Returns
    -------
    The list of candidate itemsets (c1) passed as a frozenset (a set that is
    immutable and hashable).
    """
    c1 = [] # list of all items in the database of transactions
    for transaction in dataset:
        for item in transaction:
            if not [item] in c1:
                c1.append([item])
    c1.sort()

    if verbose:
        # Print a list of all the candidate items.
        print("" \
            + "{" \
            + "".join(str(i[0]) + ", " for i in iter(c1)).rstrip(', ') \
            + "}")

    # Map c1 to a frozenset because it will be the key of a dictionary.
    return list(map(frozenset, c1))

def support_prune(dataset, candidates, min_support, verbose=False):
    """Returns all candidate itemsets that meet a minimum support threshold.
```

```python
    By the apriori principle, if an itemset is frequent, then all of its
    subsets must also be frequent. As a result, we can perform support-based
    pruning to systematically control the exponential growth of candidate
    itemsets. Thus, itemsets that do not meet the minimum support level are
    pruned from the input list of itemsets (dataset).

    Parameters
    ----------
    dataset : list
        The dataset (a list of transactions) from which to generate candidate
        itemsets.

    candidates : frozenset
        The list of candidate itemsets.

    min_support : float
        The minimum support threshold.

    Returns
    -------
    retlist : list
        The list of frequent itemsets.

    support_data : dict
        The support data for all candidate itemsets.
    """
    sscnt = {} # set for support counts
    for tid in dataset:
        for can in candidates:
            if can.issubset(tid):
                sscnt.setdefault(can, 0)
                sscnt[can] += 1

    num_items = float(len(dataset)) # total number of transactions in the dataset
    retlist = [] # array for unpruned itemsets
    support_data = {} # set for support data for corresponding itemsets
    for key in sscnt:
        # Calculate the support of itemset key.
        support = sscnt[key] / num_items
        if support >= min_support:
            retlist.insert(0, key)
        support_data[key] = support

    # Print a list of the pruned itemsets.
    if verbose:
        for kset in retlist:
            for item in kset:
                print("{" + str(item) + "}")
        print("")
        for key in sscnt:
            print("" \
                + "{" \
                + "".join([str(i) + ", " for i in iter(key)]).rstrip(', ') \
                + "}" \
                + ":  sup = " + str(support_data[key]))

    return retlist, support_data

def apriori_gen(freq_sets, k):
    """Generates candidate itemsets (via the F_k-1 x F_k-1 method).

    This operation generates new candidate k-itemsets based on the frequent
    (k-1)-itemsets found in the previous iteration. The candidate generation
    procedure merges a pair of frequent (k-1)-itemsets only if their first k-2
    items are identical.

    Parameters
    ----------
    freq_sets : list
        The list of frequent (k-1)-itemsets.

    k : integer
```

```python
            The cardinality of the current itemsets being evaluated.

    Returns
    -------
    retlist : list
        The list of merged frequent itemsets.
    """
    retList = [] # list of merged frequent itemsets
    lenLk = len(freq_sets) # number of frequent itemsets
    for i in range(lenLk):
        for j in range(i+1, lenLk):
            a=list(freq_sets[i])
            b=list(freq_sets[j])
            a.sort()
            b.sort()
            F1 = a[:k-2] # first k-2 items of freq_sets[i]
            F2 = b[:k-2] # first k-2 items of freq_sets[j]

            if F1 == F2: # if the first k-2 items are identical
                # Merge the frequent itemsets.
                retList.append(freq_sets[i] | freq_sets[j])

    return retList

def rules_from_conseq(freq_set, H, support_data, rules, min_confidence=0.5, verbose=False):
    """Generates a set of candidate rules.

    Parameters
    ----------
    freq_set : frozenset
        The complete list of frequent itemsets.

    H : list
        A list of frequent itemsets (of a particular length).

    support_data : dict
        The support data for all candidate itemsets.

    rules : list
        A potentially incomplete set of candidate rules above the minimum
        confidence threshold.

    min_confidence : float
        The minimum confidence threshold. Defaults to 0.5.
    """
    m = len(H[0])
    if m == 1:
        Hmp1 = calc_confidence(freq_set, H, support_data, rules, min_confidence, verbose)
    if (len(freq_set) > (m+1)):
        Hmp1 = apriori_gen(H, m+1) # generate candidate itemsets
        Hmp1 = calc_confidence(freq_set, Hmp1, support_data, rules, min_confidence, verbose)
        if len(Hmp1) > 1:
            # If there are candidate rules above the minimum confidence
            # threshold, recurse on the list of these candidate rules.
            rules_from_conseq(freq_set, Hmp1, support_data, rules, min_confidence, verbose)

def calc_confidence(freq_set, H, support_data, rules, min_confidence=0.5, verbose=False):
    """Evaluates the generated rules.

    One measurement for quantifying the goodness of association rules is
    confidence. The confidence for a rule 'P implies H' (P -> H) is defined as
    the support for P and H divided by the support for P
    (support (P|H) / support(P)), where the | symbol denotes the set union
    (thus P|H means all the items in set P or in set H).

    To calculate the confidence, we iterate through the frequent itemsets and
    associated support data. For each frequent itemset, we divide the support
```

```
        of the itemset by the support of the antecedent (left-hand-side of the
        rule).

        Parameters
        ----------
        freq_set : frozenset
            The complete list of frequent itemsets.

        H : list
            A list of frequent itemsets (of a particular length).

        min_support : float
            The minimum support threshold.

        rules : list
            A potentially incomplete set of candidate rules above the minimum
            confidence threshold.

        min_confidence : float
            The minimum confidence threshold. Defaults to 0.5.

        Returns
        -------
        pruned_H : list
            The list of candidate rules above the minimum confidence threshold.
        """
        pruned_H = [] # list of candidate rules above the minimum confidence threshold
        for conseq in H: # iterate over the frequent itemsets
            conf = support_data[freq_set] / support_data[freq_set - conseq]
            if conf >= min_confidence:
                rules.append((freq_set - conseq, conseq, conf))
                pruned_H.append(conseq)

                if verbose:
                    print("" \
                        + "{" \
                        + "".join([str(i) + ", " for i in iter(freq_set-conseq)]).rstrip(',
') \
                        + "}" \
                        + " ---> " \
                        + "{" \
                        + "".join([str(i) + ", " for i in iter(conseq)]).rstrip(', ') \
                        + "}" \
                        + ":  conf = " + str(round(conf, 3)) \
                        + ", sup = " + str(round(support_data[freq_set], 3)))

        return pruned_H

def generate_rules(F, support_data, min_confidence=0.5, verbose=True):
    """Generates a set of candidate rules from a list of frequent itemsets.

    For each frequent itemset, we calculate the confidence of using a
    particular item as the rule consequent (right-hand-side of the rule). By
    testing and merging the remaining rules, we recursively create a list of
    pruned rules.

    Parameters
    ----------
    F : list
        A list of frequent itemsets.

    support_data : dict
        The corresponding support data for the frequent itemsets (L).

    min_confidence : float
        The minimum confidence threshold. Defaults to 0.5.

    Returns
    -------
    rules : list
        The list of candidate rules above the minimum confidence threshold.
    """
```

```
        rules = []
        for i in range(1, len(F)):
            for freq_set in F[i]:
                H1 = [frozenset([itemset]) for itemset in freq_set]
                if (i > 1):
                    rules_from_conseq(freq_set, H1, support_data, rules, min_confidence, ver
bose)
                else:
                    calc_confidence(freq_set, H1, support_data, rules, min_confidence, verbo
se)

        return rules
```

Part 1

Apriori

In [3]:

```
dataset = load_dataset('grocery.csv')
data_set_list = list(map(set, dataset))
```

Task 1

Make use of the provided functions to generate candidate itemsets, select those that are frequent using Apriori, and subsequently list association rules derived from these.

In [4]:

```
min_support_parameters = [0.005*i for i in range(1,30,3)]
min_confidence = 0.4

for min_support in min_support_parameters:
    frequent_itemsets, support_data = apriori(dataset, min_support)
    generated_rules = generate_rules(frequent_itemsets, support_data, min_confidence)
    print(f"\n --------------- Minimum Support --------------- {min_support} -----------
------ \n")
    print(generated_rules)
```

```
{pot plants} ---> {whole milk}:  conf = 0.4, sup = 0.007
{chicken} ---> {whole milk}:  conf = 0.41, sup = 0.018
{pasta} ---> {whole milk}:  conf = 0.405, sup = 0.006
{onions} ---> {other vegetables}:  conf = 0.459, sup = 0.014
{semi-finished bread} ---> {whole milk}:  conf = 0.402, sup = 0.007
{herbs} ---> {whole milk}:  conf = 0.475, sup = 0.008
{herbs} ---> {other vegetables}:  conf = 0.475, sup = 0.008
{herbs} ---> {root vegetables}:  conf = 0.431, sup = 0.007
{mustard} ---> {whole milk}:  conf = 0.432, sup = 0.005
{sliced cheese} ---> {whole milk}:  conf = 0.44, sup = 0.011
{white bread} ---> {whole milk}:  conf = 0.406, sup = 0.017
{cake bar} ---> {whole milk}:  conf = 0.423, sup = 0.006
{baking powder} ---> {whole milk}:  conf = 0.523, sup = 0.009
{soft cheese} ---> {whole milk}:  conf = 0.44, sup = 0.008
{soft cheese} ---> {other vegetables}:  conf = 0.417, sup = 0.007
{hard cheese} ---> {whole milk}:  conf = 0.411, sup = 0.01
{cream cheese} ---> {whole milk}:  conf = 0.415, sup = 0.016
{hamburger meat} ---> {whole milk}:  conf = 0.443, sup = 0.015
{chicken} ---> {other vegetables}:  conf = 0.417, sup = 0.018
{butter milk} ---> {whole milk}:  conf = 0.415, sup = 0.012
{oil} ---> {whole milk}:  conf = 0.402, sup = 0.011
{margarine} ---> {whole milk}:  conf = 0.413, sup = 0.024
{beef} ---> {whole milk}:  conf = 0.405, sup = 0.021
{detergent} ---> {whole milk}:  conf = 0.466, sup = 0.009
{frozen vegetables} ---> {whole milk}:  conf = 0.425, sup = 0.02
{processed cheese} ---> {whole milk}:  conf = 0.423, sup = 0.007
{baking powder} ---> {other vegetables}:  conf = 0.414, sup = 0.007
{grapes} ---> {other vegetables}:  conf = 0.405, sup = 0.009
{ham} ---> {whole milk}:  conf = 0.441, sup = 0.011
{domestic eggs} ---> {whole milk}:  conf = 0.473, sup = 0.03
```

```
{whipped/sour cream} ---> {whole milk}:  conf = 0.45, sup = 0.032
{whipped/sour cream} ---> {other vegetables}:  conf = 0.403, sup = 0.029
{sugar} ---> {whole milk}:  conf = 0.444, sup = 0.015
{root vegetables} ---> {whole milk}:  conf = 0.449, sup = 0.049
{hamburger meat} ---> {other vegetables}:  conf = 0.416, sup = 0.014
{root vegetables} ---> {other vegetables}:  conf = 0.435, sup = 0.047
{tropical fruit} ---> {whole milk}:  conf = 0.403, sup = 0.042
{curd} ---> {whole milk}:  conf = 0.49, sup = 0.026
{flour} ---> {whole milk}:  conf = 0.485, sup = 0.008
{yogurt} ---> {whole milk}:  conf = 0.402, sup = 0.056
{butter} ---> {whole milk}:  conf = 0.497, sup = 0.028
{root vegetables, newspapers} ---> {other vegetables}:  conf = 0.522, sup = 0.006
{root vegetables, shopping bags} ---> {whole milk}:  conf = 0.413, sup = 0.005
{root vegetables, newspapers} ---> {whole milk}:  conf = 0.504, sup = 0.006
{yogurt, beef} ---> {whole milk}:  conf = 0.522, sup = 0.006
{pastry, pip fruit} ---> {whole milk}:  conf = 0.476, sup = 0.005
{root vegetables, soda} ---> {other vegetables}:  conf = 0.443, sup = 0.008
{root vegetables, beef} ---> {other vegetables}:  conf = 0.456, sup = 0.008
{other vegetables, beef} ---> {root vegetables}:  conf = 0.402, sup = 0.008
{other vegetables, onions} ---> {root vegetables}:  conf = 0.4, sup = 0.006
{root vegetables, onions} ---> {other vegetables}:  conf = 0.602, sup = 0.006
{other vegetables, coffee} ---> {whole milk}:  conf = 0.477, sup = 0.006
{root vegetables, sausage} ---> {whole milk}:  conf = 0.517, sup = 0.008
{yogurt, beef} ---> {other vegetables}:  conf = 0.443, sup = 0.005
{whole milk, oil} ---> {other vegetables}:  conf = 0.45, sup = 0.005
{other vegetables, oil} ---> {whole milk}:  conf = 0.51, sup = 0.005
{brown bread, rolls/buns} ---> {whole milk}:  conf = 0.419, sup = 0.005
{chicken, root vegetables} ---> {whole milk}:  conf = 0.551, sup = 0.006
{chicken, whole milk} ---> {other vegetables}:  conf = 0.48, sup = 0.008
{chicken, other vegetables} ---> {whole milk}:  conf = 0.472, sup = 0.008
{chicken, rolls/buns} ---> {whole milk}:  conf = 0.547, sup = 0.005
{bottled water, fruit/vegetable juice} ---> {whole milk}:  conf = 0.407, sup = 0.006
{other vegetables, pastry} ---> {whole milk}:  conf = 0.468, sup = 0.011
{yogurt, margarine} ---> {other vegetables}:  conf = 0.4, sup = 0.006
{yogurt, margarine} ---> {whole milk}:  conf = 0.493, sup = 0.007
{other vegetables, margarine} ---> {whole milk}:  conf = 0.469, sup = 0.009
{root vegetables, margarine} ---> {other vegetables}:  conf = 0.532, sup = 0.006
{citrus fruit, pip fruit} ---> {other vegetables}:  conf = 0.426, sup = 0.006
{root vegetables, citrus fruit} ---> {whole milk}:  conf = 0.517, sup = 0.009
{hygiene articles, whole milk} ---> {other vegetables}:  conf = 0.405, sup = 0.005
{other vegetables, hygiene articles} ---> {whole milk}:  conf = 0.543, sup = 0.005
{onions, whole milk} ---> {other vegetables}:  conf = 0.546, sup = 0.007
{other vegetables, onions} ---> {whole milk}:  conf = 0.464, sup = 0.007
{root vegetables, pastry} ---> {other vegetables}:  conf = 0.537, sup = 0.006
{tropical fruit, newspapers} ---> {whole milk}:  conf = 0.431, sup = 0.005
{yogurt, newspapers} ---> {whole milk}:  conf = 0.43, sup = 0.007
{tropical fruit, sausage} ---> {other vegetables}:  conf = 0.431, sup = 0.006
{bottled water, sausage} ---> {other vegetables}:  conf = 0.424, sup = 0.005
{other vegetables, newspapers} ---> {whole milk}:  conf = 0.432, sup = 0.008
{rolls/buns, pork} ---> {other vegetables}:  conf = 0.495, sup = 0.006
{root vegetables, bottled water} ---> {whole milk}:  conf = 0.468, sup = 0.007
{other vegetables, bottled water} ---> {whole milk}:  conf = 0.434, sup = 0.011
{napkins, other vegetables} ---> {whole milk}:  conf = 0.472, sup = 0.007
{curd, rolls/buns} ---> {whole milk}:  conf = 0.586, sup = 0.006
{yogurt, rolls/buns} ---> {whole milk}:  conf = 0.453, sup = 0.016
{napkins, rolls/buns} ---> {whole milk}:  conf = 0.452, sup = 0.005
{root vegetables, shopping bags} ---> {other vegetables}:  conf = 0.516, sup = 0.007
{citrus fruit, whipped/sour cream} ---> {whole milk}:  conf = 0.579, sup = 0.006
{citrus fruit, whole milk} ---> {other vegetables}:  conf = 0.427, sup = 0.013
{other vegetables, citrus fruit} ---> {whole milk}:  conf = 0.451, sup = 0.013
{citrus fruit, whipped/sour cream} ---> {other vegetables}:  conf = 0.523, sup = 0.006
{citrus fruit, rolls/buns} ---> {whole milk}:  conf = 0.43, sup = 0.007
{whipped/sour cream, sausage} ---> {whole milk}:  conf = 0.562, sup = 0.005
{beef, rolls/buns} ---> {other vegetables}:  conf = 0.425, sup = 0.006
{beef, whole milk} ---> {other vegetables}:  conf = 0.435, sup = 0.009
{other vegetables, beef} ---> {whole milk}:  conf = 0.469, sup = 0.009
{beef, rolls/buns} ---> {whole milk}:  conf = 0.5, sup = 0.007
{yogurt, sausage} ---> {other vegetables}:  conf = 0.415, sup = 0.008
{root vegetables, pork} ---> {other vegetables}:  conf = 0.515, sup = 0.007
{root vegetables, pork} ---> {whole milk}:  conf = 0.5, sup = 0.007
{root vegetables, beef} ---> {whole milk}:  conf = 0.462, sup = 0.008
{tropical fruit, whipped/sour cream} ---> {other vegetables}:  conf = 0.566, sup = 0.008
```

```
{root vegetables, whipped/sour cream} ---> {whole milk}:  conf = 0.554, sup = 0.009
{whipped/sour cream, curd} ---> {whole milk}:  conf = 0.563, sup = 0.006
{butter, whipped/sour cream} ---> {whole milk}:  conf = 0.66, sup = 0.007
{root vegetables, curd} ---> {whole milk}:  conf = 0.57, sup = 0.006
{butter, root vegetables} ---> {whole milk}:  conf = 0.638, sup = 0.008
{other vegetables, curd} ---> {whole milk}:  conf = 0.574, sup = 0.01
{butter, whole milk} ---> {other vegetables}:  conf = 0.417, sup = 0.011
{butter, other vegetables} ---> {whole milk}:  conf = 0.574, sup = 0.011
{butter, yogurt} ---> {other vegetables}:  conf = 0.438, sup = 0.006
{root vegetables, domestic eggs} ---> {other vegetables}:  conf = 0.511, sup = 0.007
{yogurt, domestic eggs} ---> {other vegetables}:  conf = 0.404, sup = 0.006
{yogurt, pip fruit} ---> {whole milk}:  conf = 0.531, sup = 0.01
{root vegetables, frankfurter} ---> {whole milk}:  conf = 0.5, sup = 0.005
{cream cheese , whole milk} ---> {yogurt}:  conf = 0.401, sup = 0.007
{yogurt, cream cheese} ---> {whole milk}:  conf = 0.533, sup = 0.007
{tropical fruit, pip fruit} ---> {other vegetables}:  conf = 0.463, sup = 0.009
{root vegetables, pip fruit} ---> {other vegetables}:  conf = 0.523, sup = 0.008
{whipped/sour cream, pip fruit} ---> {other vegetables}:  conf = 0.604, sup = 0.006
{yogurt, tropical fruit} ---> {other vegetables}:  conf = 0.42, sup = 0.012
{yogurt, whipped/sour cream} ---> {other vegetables}:  conf = 0.49, sup = 0.01
{yogurt, pip fruit} ---> {other vegetables}:  conf = 0.452, sup = 0.008
{whipped/sour cream, domestic eggs} ---> {whole milk}:  conf = 0.571, sup = 0.006
{whipped/sour cream, domestic eggs} ---> {other vegetables}:  conf = 0.51, sup = 0.005
{butter, domestic eggs} ---> {whole milk}:  conf = 0.621, sup = 0.006
{whipped/sour cream, pip fruit} ---> {whole milk}:  conf = 0.648, sup = 0.006
{yogurt, whipped/sour cream} ---> {whole milk}:  conf = 0.525, sup = 0.011
{tropical fruit, whipped/sour cream} ---> {yogurt}:  conf = 0.449, sup = 0.006
{tropical fruit, whipped/sour cream} ---> {whole milk}:  conf = 0.574, sup = 0.008
{root vegetables, pip fruit} ---> {whole milk}:  conf = 0.575, sup = 0.009
{yogurt, frankfurter} ---> {whole milk}:  conf = 0.555, sup = 0.006
{yogurt, cream cheese} ---> {other vegetables}:  conf = 0.426, sup = 0.005
{cream cheese , whole milk} ---> {other vegetables}:  conf = 0.407, sup = 0.007
{other vegetables, cream cheese} ---> {whole milk}:  conf = 0.489, sup = 0.007
{other vegetables, white bread} ---> {whole milk}:  conf = 0.43, sup = 0.006
{other vegetables, brown bread} ---> {whole milk}:  conf = 0.5, sup = 0.009
{butter, rolls/buns} ---> {whole milk}:  conf = 0.492, sup = 0.007
{tropical fruit, whole milk} ---> {other vegetables}:  conf = 0.404, sup = 0.017
{other vegetables, tropical fruit} ---> {whole milk}:  conf = 0.476, sup = 0.017
{root vegetables, frozen vegetables} ---> {other vegetables}:  conf = 0.526, sup = 0.006
{root vegetables, fruit/vegetable juice} ---> {other vegetables}:  conf = 0.551, sup = 0.
007
{other vegetables, fruit/vegetable juice} ---> {whole milk}:  conf = 0.498, sup = 0.01
{yogurt, fruit/vegetable juice} ---> {other vegetables}:  conf = 0.44, sup = 0.008
{other vegetables, bottled beer} ---> {whole milk}:  conf = 0.472, sup = 0.008
{yogurt, other vegetables} ---> {whole milk}:  conf = 0.513, sup = 0.022
{yogurt, root vegetables} ---> {other vegetables}:  conf = 0.5, sup = 0.013
{yogurt, frozen vegetables} ---> {other vegetables}:  conf = 0.426, sup = 0.005
{yogurt, frozen vegetables} ---> {whole milk}:  conf = 0.492, sup = 0.006
{other vegetables, chocolate} ---> {whole milk}:  conf = 0.432, sup = 0.005
{other vegetables, frankfurter} ---> {whole milk}:  conf = 0.463, sup = 0.008
{whole milk, pip fruit} ---> {other vegetables}:  conf = 0.449, sup = 0.014
{other vegetables, pip fruit} ---> {whole milk}:  conf = 0.518, sup = 0.014
{tropical fruit, frankfurter} ---> {whole milk}:  conf = 0.548, sup = 0.005
{tropical fruit, fruit/vegetable juice} ---> {whole milk}:  conf = 0.437, sup = 0.006
{pip fruit, domestic eggs} ---> {whole milk}:  conf = 0.624, sup = 0.005
{sausage, pip fruit} ---> {whole milk}:  conf = 0.519, sup = 0.006
{tropical fruit, sausage} ---> {whole milk}:  conf = 0.518, sup = 0.007
{citrus fruit, pip fruit} ---> {tropical fruit}:  conf = 0.404, sup = 0.006
{soda, domestic eggs} ---> {other vegetables}:  conf = 0.41, sup = 0.005
{whole milk, domestic eggs} ---> {other vegetables}:  conf = 0.41, sup = 0.012
{other vegetables, domestic eggs} ---> {whole milk}:  conf = 0.553, sup = 0.012
{hamburger meat, other vegetables} ---> {whole milk}:  conf = 0.456, sup = 0.006
{hamburger meat, whole milk} ---> {other vegetables}:  conf = 0.428, sup = 0.006
{root vegetables, sausage} ---> {other vegetables}:  conf = 0.456, sup = 0.007
{whipped/sour cream, rolls/buns} ---> {whole milk}:  conf = 0.535, sup = 0.008
{yogurt, sausage} ---> {whole milk}:  conf = 0.446, sup = 0.009
{napkins, yogurt} ---> {whole milk}:  conf = 0.496, sup = 0.006
{yogurt, fruit/vegetable juice} ---> {whole milk}:  conf = 0.505, sup = 0.009
{yogurt, bottled beer} ---> {whole milk}:  conf = 0.56, sup = 0.005
{pastry, rolls/buns} ---> {whole milk}:  conf = 0.408, sup = 0.009
{sausage, pastry} ---> {whole milk}:  conf = 0.455, sup = 0.006
{rolls/buns, pork} ---> {whole milk}:  conf = 0.55, sup = 0.006
```

```
{butter, tropical fruit} ---> {other vegetables}:  conf = 0.551, sup = 0.005
{chicken, root vegetables} ---> {other vegetables}:  conf = 0.523, sup = 0.006
{tropical fruit, citrus fruit} ---> {other vegetables}:  conf = 0.454, sup = 0.009
{citrus fruit, root vegetables} ---> {other vegetables}:  conf = 0.586, sup = 0.01
{margarine, domestic eggs} ---> {whole milk}:  conf = 0.622, sup = 0.005
{root vegetables, rolls/buns} ---> {whole milk}:  conf = 0.523, sup = 0.013
{root vegetables, frozen vegetables} ---> {whole milk}:  conf = 0.535, sup = 0.006
{frozen vegetables, rolls/buns} ---> {whole milk}:  conf = 0.5, sup = 0.005
{rolls/buns, domestic eggs} ---> {whole milk}:  conf = 0.422, sup = 0.007
{root vegetables, fruit/vegetable juice} ---> {whole milk}:  conf = 0.542, sup = 0.007
{rolls/buns, margarine} ---> {whole milk}:  conf = 0.538, sup = 0.008
{other vegetables, rolls/buns} ---> {whole milk}:  conf = 0.42, sup = 0.018
{citrus fruit, domestic eggs} ---> {whole milk}:  conf = 0.549, sup = 0.006
{tropical fruit, rolls/buns} ---> {whole milk}:  conf = 0.446, sup = 0.011
{pip fruit, rolls/buns} ---> {whole milk}:  conf = 0.445, sup = 0.006
{tropical fruit, pip fruit} ---> {whole milk}:  conf = 0.413, sup = 0.008
{frozen vegetables, whole milk} ---> {other vegetables}:  conf = 0.473, sup = 0.01
{other vegetables, frozen vegetables} ---> {whole milk}:  conf = 0.543, sup = 0.01
{tropical fruit, curd} ---> {other vegetables}:  conf = 0.515, sup = 0.005
{tropical fruit, fruit/vegetable juice} ---> {other vegetables}:  conf = 0.481, sup = 0.0
07
{whipped/sour cream, rolls/buns} ---> {other vegetables}:  conf = 0.458, sup = 0.007
{root vegetables, whipped/sour cream} ---> {other vegetables}:  conf = 0.5, sup = 0.009
{root vegetables, bottled water} ---> {other vegetables}:  conf = 0.448, sup = 0.007
{root vegetables, curd} ---> {other vegetables}:  conf = 0.505, sup = 0.005
{butter, whipped/sour cream} ---> {other vegetables}:  conf = 0.57, sup = 0.006
{butter, root vegetables} ---> {other vegetables}:  conf = 0.512, sup = 0.007
{butter, rolls/buns} ---> {other vegetables}:  conf = 0.424, sup = 0.006
{tropical fruit, root vegetables} ---> {whole milk}:  conf = 0.57, sup = 0.012
{root vegetables, domestic eggs} ---> {whole milk}:  conf = 0.596, sup = 0.009
{soda, domestic eggs} ---> {whole milk}:  conf = 0.418, sup = 0.005
{yogurt, domestic eggs} ---> {whole milk}:  conf = 0.539, sup = 0.008
{tropical fruit, domestic eggs} ---> {whole milk}:  conf = 0.607, sup = 0.007
{brown bread, soda} ---> {whole milk}:  conf = 0.403, sup = 0.005
{root vegetables, brown bread} ---> {whole milk}:  conf = 0.56, sup = 0.006
{yogurt, brown bread} ---> {whole milk}:  conf = 0.49, sup = 0.007
{tropical fruit, brown bread} ---> {whole milk}:  conf = 0.533, sup = 0.006
{root vegetables, soda} ---> {whole milk}:  conf = 0.437, sup = 0.008
{yogurt, root vegetables} ---> {whole milk}:  conf = 0.563, sup = 0.015
{root vegetables, pastry} ---> {whole milk}:  conf = 0.519, sup = 0.006
{yogurt, pastry} ---> {whole milk}:  conf = 0.517, sup = 0.009
{tropical fruit, pastry} ---> {whole milk}:  conf = 0.508, sup = 0.007
{yogurt, coffee} ---> {whole milk}:  conf = 0.521, sup = 0.005
{whipped/sour cream, whole milk} ---> {other vegetables}:  conf = 0.454, sup = 0.015
{other vegetables, whipped/sour cream} ---> {whole milk}:  conf = 0.507, sup = 0.015
{whole milk, pork} ---> {other vegetables}:  conf = 0.459, sup = 0.01
{other vegetables, pork} ---> {whole milk}:  conf = 0.469, sup = 0.01
{other vegetables, soda} ---> {whole milk}:  conf = 0.425, sup = 0.014
{whipped/sour cream, soda} ---> {whole milk}:  conf = 0.474, sup = 0.005
{root vegetables, whole milk} ---> {other vegetables}:  conf = 0.474, sup = 0.023
{other vegetables, root vegetables} ---> {whole milk}:  conf = 0.489, sup = 0.023
{sugar, other vegetables} ---> {whole milk}:  conf = 0.585, sup = 0.006
{sugar, whole milk} ---> {other vegetables}:  conf = 0.419, sup = 0.006
{root vegetables, rolls/buns} ---> {other vegetables}:  conf = 0.502, sup = 0.012
{tropical fruit, root vegetables} ---> {other vegetables}:  conf = 0.585, sup = 0.012
{yogurt, curd} ---> {whole milk}:  conf = 0.582, sup = 0.01
{tropical fruit, curd} ---> {whole milk}:  conf = 0.634, sup = 0.007
{tropical fruit, curd} ---> {yogurt}:  conf = 0.515, sup = 0.005
{tropical fruit, bottled water} ---> {whole milk}:  conf = 0.434, sup = 0.008
{butter, bottled water} ---> {whole milk}:  conf = 0.602, sup = 0.005
{yogurt, bottled water} ---> {whole milk}:  conf = 0.42, sup = 0.01
{citrus fruit, bottled water} ---> {whole milk}:  conf = 0.436, sup = 0.006
{butter, tropical fruit} ---> {whole milk}:  conf = 0.622, sup = 0.006
{butter, citrus fruit} ---> {whole milk}:  conf = 0.556, sup = 0.005
{yogurt, tropical fruit} ---> {whole milk}:  conf = 0.517, sup = 0.015
{yogurt, citrus fruit} ---> {whole milk}:  conf = 0.474, sup = 0.01
{tropical fruit, citrus fruit} ---> {whole milk}:  conf = 0.454, sup = 0.009
{butter, yogurt} ---> {whole milk}:  conf = 0.639, sup = 0.009
{long life bakery product, whole milk} ---> {other vegetables}:  conf = 0.421, sup = 0.00
6
{other vegetables, long life bakery product} ---> {whole milk}:  conf = 0.533, sup = 0.00
6
```

{root vegetables, citrus fruit, whole milk} ---> {other vegetables}: conf = 0.633, sup = 0.006
{other vegetables, citrus fruit, whole milk} ---> {root vegetables}: conf = 0.445, sup = 0.006
{root vegetables, other vegetables, citrus fruit} ---> {whole milk}: conf = 0.559, sup = 0.006
{yogurt, other vegetables, rolls/buns} ---> {whole milk}: conf = 0.522, sup = 0.006
{root vegetables, whole milk, rolls/buns} ---> {other vegetables}: conf = 0.488, sup = 0.006
{other vegetables, root vegetables, rolls/buns} ---> {whole milk}: conf = 0.508, sup = 0.006
{tropical fruit, root vegetables, whole milk} ---> {other vegetables}: conf = 0.585, sup = 0.007
{other vegetables, tropical fruit, whole milk} ---> {root vegetables}: conf = 0.411, sup = 0.007
{root vegetables, other vegetables, tropical fruit} ---> {whole milk}: conf = 0.57, sup = 0.007
{root vegetables, whipped/sour cream, whole milk} ---> {other vegetables}: conf = 0.548, sup = 0.005
{other vegetables, root vegetables, whipped/sour cream} ---> {whole milk}: conf = 0.607, sup = 0.005
{yogurt, whole milk, pip fruit} ---> {other vegetables}: conf = 0.532, sup = 0.005
{yogurt, other vegetables, pip fruit} ---> {whole milk}: conf = 0.625, sup = 0.005
{yogurt, whipped/sour cream, whole milk} ---> {other vegetables}: conf = 0.514, sup = 0.006
{yogurt, other vegetables, whipped/sour cream} ---> {whole milk}: conf = 0.55, sup = 0.006
{other vegetables, tropical fruit, whole milk} ---> {yogurt}: conf = 0.446, sup = 0.008
{yogurt, tropical fruit, whole milk} ---> {other vegetables}: conf = 0.503, sup = 0.008
{yogurt, other vegetables, tropical fruit} ---> {whole milk}: conf = 0.62, sup = 0.008
{root vegetables, whole milk, pip fruit} ---> {other vegetables}: conf = 0.614, sup = 0.005
{other vegetables, whole milk, pip fruit} ---> {root vegetables}: conf = 0.406, sup = 0.005
{other vegetables, root vegetables, pip fruit} ---> {whole milk}: conf = 0.675, sup = 0.005
{yogurt, root vegetables, whole milk} ---> {other vegetables}: conf = 0.538, sup = 0.008
{yogurt, other vegetables, root vegetables} ---> {whole milk}: conf = 0.606, sup = 0.008
{other vegetables, whole milk, fruit/vegetable juice} ---> {yogurt}: conf = 0.485, sup = 0.005
{yogurt, whole milk, fruit/vegetable juice} ---> {other vegetables}: conf = 0.538, sup = 0.005
{yogurt, other vegetables, fruit/vegetable juice} ---> {whole milk}: conf = 0.617, sup = 0.005
{tropical fruit, root vegetables, whole milk} ---> {yogurt}: conf = 0.475, sup = 0.006
{yogurt, tropical fruit, root vegetables} ---> {whole milk}: conf = 0.7, sup = 0.006

 --------------- Minimum Support --------------- 0.005 -----------------

[(frozenset({'pot plants'}), frozenset({'whole milk'}), 0.4), (frozenset({'chicken'}), frozenset({'whole milk'}), 0.4099526066350711), (frozenset({'pasta'}), frozenset({'whole milk'}), 0.40540540540540543), (frozenset({'onions'}), frozenset({'other vegetables'}), 0.459016393442623
96), (frozenset({'semi-finished bread'}), frozenset({'whole milk'}), 0.4022988505747126), (frozenset({'herbs'}), frozenset({'whole milk'}), 0.47500000000000003), (frozenset({'herbs'}), frozenset({'other vegetables'}), 0.47500000000000003), (frozenset({'herbs'}), frozenset({'root vegetables'}), 0.43124999999999997), (frozenset({'mustard'}), frozenset({'whole milk'}), 0.4322033898305085), (frozenset({'sliced cheese'}), frozenset({'whole milk'}), 0.4398340248962655), (frozenset({'white bread'}), frozenset({'whole milk'}), 0.4057971014492754), (frozenset({'cake bar'}), frozenset({'whole milk'}), 0.4230769230769231), (frozenset({'baking powder'}), frozenset({'whole milk'}), 0.5229885057471264), (frozenset({'soft cheese'}), frozenset({'whole milk'}), 0.44047619047619047), (frozenset({'soft cheese'}), frozenset({'other vegetables'}), 0.41666666666666663), (frozenset({'hard cheese'}), frozenset({'whole milk'}), 0.4107883817427386), (frozenset({'cream cheese '}), frozenset({'whole milk'}), 0.41538461538461546), (frozenset({'hamburger meat'}), frozenset({'whole milk'}), 0.4434250764525994), (frozenset({'chicken'}), frozenset({'other vegetables'}), 0.4170616113744075), (frozenset({'butter milk'}), frozenset({'whole milk'}), 0.41454545454545455), (frozenset({'oil'}), frozenset({'whole milk'}), 0.40217391304347827), (frozenset({'margarine'}), frozenset({'whole milk'}), 0.4131944444444444), (frozenset({'beef'}), frozenset({'whole milk'}), 0.4050387596899225), (frozenset({'detergent'}), frozenset({'whole milk'}), 0.4656084656084656), (frozenset({'frozen vegetables'}), frozenset({'whole milk'}), 0.4249471458773784), (frozenset({'processed cheese'}), frozenset({'whole milk'}), 0.4233128834355829), (frozenset({'baking powder'}), frozenset({'other vegetabl

es'}), 0.41379310344827586), (frozenset({'grapes'}), frozenset({'other vegetables'}), 0.4
04545454545454), (frozenset({'ham'}), frozenset({'whole milk'}), 0.44140625), (frozense
t({'domestic eggs'}), frozenset({'whole milk'}), 0.47275641025641024), (frozenset({'whipp
ed/sour cream'}), frozenset({'whole milk'}), 0.449645390070922), (frozenset({'whipped/sou
r cream'}), frozenset({'other vegetables'}), 0.40283687943262414), (frozenset({'sugar'}),
frozenset({'whole milk'}), 0.4444444444444445), (frozenset({'root vegetables'}), frozense
t({'whole milk'}), 0.44869402985074625), (frozenset({'hamburger meat'}), frozenset({'othe
r vegetables'}), 0.41590214067278286), (frozenset({'root vegetables'}), frozenset({'other
vegetables'}), 0.43470149253731344), (frozenset({'tropical fruit'}), frozenset({'whole mi
lk'}), 0.40310077519379844), (frozenset({'curd'}), frozenset({'whole milk'}), 0.490458015
2671756), (frozenset({'flour'}), frozenset({'whole milk'}), 0.48538011695906436), (frozen
set({'yogurt'}), frozenset({'whole milk'}), 0.40160349854227406), (frozenset({'butter'}),
frozenset({'whole milk'}), 0.4972477064220184), (frozenset({'root vegetables', 'newspaper
s'}), frozenset({'other vegetables'}), 0.5221238938053098), (frozenset({'root vegetables'
, 'shopping bags'}), frozenset({'whole milk'}), 0.4126984126984127), (frozenset({'root ve
getables', 'newspapers'}), frozenset({'whole milk'}), 0.504424778761062), (frozenset({'yo
gurt', 'beef'}), frozenset({'whole milk'}), 0.5217391304347826), (frozenset({'pastry', 'p
ip fruit'}), frozenset({'whole milk'}), 0.47619047619047616), (frozenset({'root vegetable
s', 'soda'}), frozenset({'other vegetables'}), 0.4426229508196722), (frozenset({'root veg
etables', 'beef'}), frozenset({'other vegetables'}), 0.456140350877193), (frozenset({'oth
er vegetables', 'beef'}), frozenset({'root vegetables'}), 0.40206185567010316), (frozense
t({'other vegetables', 'onions'}), frozenset({'root vegetables'}), 0.4), (frozenset({'roo
t vegetables', 'onions'}), frozenset({'other vegetables'}), 0.6021505376344086), (frozens
et({'other vegetables', 'coffee'}), frozenset({'whole milk'}), 0.4772727272727273), (froz
enset({'root vegetables', 'sausage'}), frozenset({'whole milk'}), 0.5170068027210885), (f
rozenset({'yogurt', 'beef'}), frozenset({'other vegetables'}), 0.4434782608695652), (froz
enset({'whole milk', 'oil'}), frozenset({'other vegetables'}), 0.45045045045045045), (froz
enset({'other vegetables', 'oil'}), frozenset({'whole milk'}), 0.5102040816326531), (froz
enset({'brown bread', 'rolls/buns'}), frozenset({'whole milk'}), 0.4193548387096774), (fr
ozenset({'chicken', 'root vegetables'}), frozenset({'whole milk'}), 0.5514018691588786),
(frozenset({'chicken', 'whole milk'}), frozenset({'other vegetables'}), 0.479768786127167
6), (frozenset({'chicken', 'other vegetables'}), frozenset({'whole milk'}), 0.47159090909
09091), (frozenset({'chicken', 'rolls/buns'}), frozenset({'whole milk'}), 0.5473684210526
316), (frozenset({'bottled water', 'fruit/vegetable juice'}), frozenset({'whole milk'}),
0.4071428571428572), (frozenset({'other vegetables', 'pastry'}), frozenset({'whole milk'}
), 0.46846846846846846), (frozenset({'yogurt', 'margarine'}), frozenset({'other vegetable
s'}), 0.4), (frozenset({'yogurt', 'margarine'}), frozenset({'whole milk'}), 0.49285714285
71429), (frozenset({'other vegetables', 'margarine'}), frozenset({'whole milk'}), 0.46907
21649484536), (frozenset({'root vegetables', 'margarine'}), frozenset({'other vegetables'
}), 0.5321100917431192), (frozenset({'citrus fruit', 'pip fruit'}), frozenset({'other veg
etables'}), 0.4264705882352941), (frozenset({'root vegetables', 'citrus fruit'}), frozens
et({'whole milk'}), 0.5172413793103449), (frozenset({'hygiene articles', 'whole milk'}),
frozenset({'other vegetables'}), 0.40476190476190477), (frozenset({'other vegetables', 'h
ygiene articles'}), frozenset({'whole milk'}), 0.5425531914893618), (frozenset({'onions',
'whole milk'}), frozenset({'other vegetables'}), 0.546218487394958), (frozenset({'other v
egetables', 'onions'}), frozenset({'whole milk'}), 0.4642857142857143), (frozenset({'root
vegetables', 'pastry'}), frozenset({'other vegetables'}), 0.537037037037037), (frozenset(
{'tropical fruit', 'newspapers'}), frozenset({'whole milk'}), 0.43103448275862066), (froz
enset({'yogurt', 'newspapers'}), frozenset({'whole milk'}), 0.43046357615894043), (frozen
set({'tropical fruit', 'sausage'}), frozenset({'other vegetables'}), 0.43065693430656937)
, (frozenset({'bottled water', 'sausage'}), frozenset({'other vegetables'}), 0.4237288135
5932196), (frozenset({'other vegetables', 'newspapers'}), frozenset({'whole milk'}), 0.43
1578947368421), (frozenset({'rolls/buns', 'pork'}), frozenset({'other vegetables'}), 0.49
54954954954955), (frozenset({'root vegetables', 'bottled water'}), frozenset({'whole milk
'}), 0.4675324675324675), (frozenset({'other vegetables', 'bottled water'}), frozenset({'
whole milk'}), 0.4344262295081967), (frozenset({'napkins', 'other vegetables'}), frozense
t({'whole milk'}), 0.47183098591549294), (frozenset({'curd', 'rolls/buns'}), frozenset({'
whole milk'}), 0.5858585858585859), (frozenset({'yogurt', 'rolls/buns'}), frozenset({'who
le milk'}), 0.45266272189349116), (frozenset({'napkins', 'rolls/buns'}), frozenset({'whol
e milk'}), 0.4521739130434782), (frozenset({'root vegetables', 'shopping bags'}), frozens
et({'other vegetables'}), 0.5158730158730159), (frozenset({'citrus fruit', 'whipped/sour
cream'}), frozenset({'whole milk'}), 0.5794392523364487), (frozenset({'citrus fruit', 'wh
ole milk'}), frozenset({'other vegetables'}), 0.42666666666666664), (frozenset({'other ve
getables', 'citrus fruit'}), frozenset({'whole milk'}), 0.45070422535211263), (frozenset(
{'citrus fruit', 'whipped/sour cream'}), frozenset({'other vegetables'}), 0.5233644859813
085), (frozenset({'citrus fruit', 'rolls/buns'}), frozenset({'whole milk'}), 0.4303030303
0303035), (frozenset({'whipped/sour cream', 'sausage'}), frozenset({'whole milk'}), 0.561
7977528089887), (frozenset({'beef', 'rolls/buns'}), frozenset({'other vegetables'}), 0.42
53731343283582), (frozenset({'beef', 'whole milk'}), frozenset({'other vegetables'}), 0.4
3540669856459324), (frozenset({'other vegetables', 'beef'}), frozenset({'whole milk'}), 0
.4690721649484536), (frozenset({'beef', 'rolls/buns'}), frozenset({'whole milk'}), 0.5),
(frozenset({'yogurt', 'sausage'}), frozenset({'other vegetables'}), 0.41450777202072536),

(frozenset({'root vegetables', 'pork'}), frozenset({'other vegetables'}), 0.5149253731343
284), (frozenset({'root vegetables', 'pork'}), frozenset({'whole milk'}), 0.5), (frozense
t({'root vegetables', 'beef'}), frozenset({'whole milk'}), 0.4619883040935672), (frozense
t({'tropical fruit', 'whipped/sour cream'}), frozenset({'other vegetables'}), 0.566176470
5882353), (frozenset({'root vegetables', 'whipped/sour cream'}), frozenset({'whole milk'}
), 0.5535714285714286), (frozenset({'whipped/sour cream', 'curd'}), frozenset({'whole mil
k'}), 0.5631067961165048), (frozenset({'butter', 'whipped/sour cream'}), frozenset({'whol
e milk'}), 0.66), (frozenset({'root vegetables', 'curd'}), frozenset({'whole milk'}), 0.5
700934579439252), (frozenset({'butter', 'root vegetables'}), frozenset({'whole milk'}), 0
.6377952755905512), (frozenset({'other vegetables', 'curd'}), frozenset({'whole milk'}),
0.5739644970414202), (frozenset({'butter', 'whole milk'}), frozenset({'other vegetables'}
), 0.4169741697416974), (frozenset({'butter', 'other vegetables'}), frozenset({'whole mil
k'}), 0.5736040609137055), (frozenset({'butter', 'yogurt'}), frozenset({'other vegetables
'}), 0.4375), (frozenset({'root vegetables', 'domestic eggs'}), frozenset({'other vegetab
les'}), 0.5106382978723404), (frozenset({'yogurt', 'domestic eggs'}), frozenset({'other v
egetables'}), 0.4042553191489362), (frozenset({'yogurt', 'pip fruit'}), frozenset({'whole
milk'}), 0.5310734463276836), (frozenset({'root vegetables', 'frankfurter'}), frozenset({
'whole milk'}), 0.5), (frozenset({'cream cheese ', 'whole milk'}), frozenset({'yogurt'}),
0.4012345679012346), (frozenset({'yogurt', 'cream cheese '}), frozenset({'whole milk'}),
0.5327868852459017), (frozenset({'tropical fruit', 'pip fruit'}), frozenset({'other veget
ables'}), 0.46268656716417916), (frozenset({'root vegetables', 'pip fruit'}), frozenset({
'other vegetables'}), 0.5228758169934641), (frozenset({'whipped/sour cream', 'pip fruit'}
), frozenset({'other vegetables'}), 0.6043956043956045), (frozenset({'yogurt', 'tropical
fruit'}), frozenset({'other vegetables'}), 0.4201388888888889), (frozenset({'yogurt', 'wh
ipped/sour cream'}), frozenset({'other vegetables'}), 0.4901960784313725), (frozenset({'y
ogurt', 'pip fruit'}), frozenset({'other vegetables'}), 0.4519774011299435), (frozenset({
'whipped/sour cream', 'domestic eggs'}), frozenset({'whole milk'}), 0.5714285714285715),
(frozenset({'whipped/sour cream', 'domestic eggs'}), frozenset({'other vegetables'}), 0.5
102040816326531), (frozenset({'butter', 'domestic eggs'}), frozenset({'whole milk'}), 0.6
210526315789474), (frozenset({'whipped/sour cream', 'pip fruit'}), frozenset({'whole milk
'}), 0.6483516483516485), (frozenset({'yogurt', 'whipped/sour cream'}), frozenset({'whole
milk'}), 0.5245098039215685), (frozenset({'tropical fruit', 'whipped/sour cream'}), froze
nset({'yogurt'}), 0.44852941176470584), (frozenset({'tropical fruit', 'whipped/sour cream
'}), frozenset({'whole milk'}), 0.573529411764706), (frozenset({'root vegetables', 'pip f
ruit'}), frozenset({'whole milk'}), 0.5751633986928104), (frozenset({'yogurt', 'frankfurt
er'}), frozenset({'whole milk'}), 0.5545454545454546), (frozenset({'yogurt', 'cream chees
e '}), frozenset({'other vegetables'}), 0.42622950819672134), (frozenset({'cream cheese '
, 'whole milk'}), frozenset({'other vegetables'}), 0.4074074074074074), (frozenset({'othe
r vegetables', 'cream cheese '}), frozenset({'whole milk'}), 0.4888888888888889), (frozen
set({'other vegetables', 'white bread'}), frozenset({'whole milk'}), 0.42962962962962964)
, (frozenset({'other vegetables', 'brown bread'}), frozenset({'whole milk'}), 0.5), (froz
enset({'butter', 'rolls/buns'}), frozenset({'whole milk'}), 0.49242424242424243), (frozen
set({'tropical fruit', 'whole milk'}), frozenset({'other vegetables'}), 0.403846153846153
9), (frozenset({'other vegetables', 'tropical fruit'}), frozenset({'whole milk'}), 0.4759
2067988668557), (frozenset({'root vegetables', 'frozen vegetables'}), frozenset({'other v
egetables'}), 0.5263157894736842), (frozenset({'root vegetables', 'fruit/vegetable juice'
}), frozenset({'other vegetables'}), 0.5508474576271186), (frozenset({'other vegetables',
'fruit/vegetable juice'}), frozenset({'whole milk'}), 0.4975845410628019), (frozenset({'y
ogurt', 'fruit/vegetable juice'}), frozenset({'other vegetables'}), 0.44021739130434784),
(frozenset({'other vegetables', 'bottled beer'}), frozenset({'whole milk'}), 0.4716981132
075472), (frozenset({'yogurt', 'other vegetables'}), frozenset({'whole milk'}), 0.5128805
620608898), (frozenset({'yogurt', 'root vegetables'}), frozenset({'other vegetables'}), 0
.5), (frozenset({'yogurt', 'frozen vegetables'}), frozenset({'other vegetables'}), 0.4262
2950819672134), (frozenset({'yogurt', 'frozen vegetables'}), frozenset({'whole milk'}), 0
.49180327868852464), (frozenset({'other vegetables', 'chocolate'}), frozenset({'whole mil
k'}), 0.43200000000000005), (frozenset({'other vegetables', 'frankfurter'}), frozenset({'
whole milk'}), 0.4629629629629629), (frozenset({'whole milk', 'pip fruit'}), frozenset({'
other vegetables'}), 0.44932432432432434), (frozenset({'other vegetables', 'pip fruit'}),
frozenset({'whole milk'}), 0.5175097276264592), (frozenset({'tropical fruit', 'frankfurte
r'}), frozenset({'whole milk'}), 0.5483870967741935), (frozenset({'tropical fruit', 'frui
t/vegetable juice'}), frozenset({'whole milk'}), 0.43703703703703706), (frozenset({'pip f
ruit', 'domestic eggs'}), frozenset({'whole milk'}), 0.6235294117647059), (frozenset({'sa
usage', 'pip fruit'}), frozenset({'whole milk'}), 0.5188679245283019), (frozenset({'tropi
cal fruit', 'sausage'}), frozenset({'whole milk'}), 0.5182481751824818), (frozenset({'cit
rus fruit', 'pip fruit'}), frozenset({'tropical fruit'}), 0.4044117647058823), (frozenset
({'soda', 'domestic eggs'}), frozenset({'other vegetables'}), 0.4098360655737705), (froze
nset({'whole milk', 'domestic eggs'}), frozenset({'other vegetables'}), 0.410169491525423
76), (frozenset({'other vegetables', 'domestic eggs'}), frozenset({'whole milk'}), 0.5525
114155251142), (frozenset({'hamburger meat', 'other vegetables'}), frozenset({'whole milk
'}), 0.45588235294117646), (frozenset({'hamburger meat', 'whole milk'}), frozenset({'othe
r vegetables'}), 0.42758620689655175), (frozenset({'root vegetables', 'sausage'}), frozen
set({'other vegetables'}), 0.4557823129251701), (frozenset({'whipped/sour cream', 'rolls/

buns'}), frozenset({'whole milk'}), 0.5347222222222222), (frozenset({'yogurt', 'sausage'}
), frozenset({'whole milk'}), 0.44559585492227977), (frozenset({'napkins', 'yogurt'}), fr
ozenset({'whole milk'}), 0.4958677685950413), (frozenset({'yogurt', 'fruit/vegetable juic
e'}), frozenset({'whole milk'}), 0.5054347826086957), (frozenset({'yogurt', 'bottled beer
'}), frozenset({'whole milk'}), 0.5604395604395604), (frozenset({'pastry', 'rolls/buns'})
, frozenset({'whole milk'}), 0.4077669902912622), (frozenset({'sausage', 'pastry'}), froz
enset({'whole milk'}), 0.4552845528455285), (frozenset({'rolls/buns', 'pork'}), frozenset
({'whole milk'}), 0.5495495495495495), (frozenset({'butter', 'tropical fruit'}), frozense
t({'other vegetables'}), 0.5510204081632654), (frozenset({'chicken', 'root vegetables'}),
frozenset({'other vegetables'}), 0.5233644859813085), (frozenset({'tropical fruit', 'citr
us fruit'}), frozenset({'other vegetables'}), 0.45408163265306123), (frozenset({'citrus f
ruit', 'root vegetables'}), frozenset({'other vegetables'}), 0.5862068965517241), (frozen
set({'margarine', 'domestic eggs'}), frozenset({'whole milk'}), 0.6219512195121952), (fro
zenset({'root vegetables', 'rolls/buns'}), frozenset({'whole milk'}), 0.5230125523012552)
, (frozenset({'root vegetables', 'frozen vegetables'}), frozenset({'whole milk'}), 0.5350
877192982456), (frozenset({'frozen vegetables', 'rolls/buns'}), frozenset({'whole milk'})
, 0.5), (frozenset({'rolls/buns', 'domestic eggs'}), frozenset({'whole milk'}), 0.4220779
220779221), (frozenset({'root vegetables', 'fruit/vegetable juice'}), frozenset({'whole m
ilk'}), 0.5423728813559322), (frozenset({'rolls/buns', 'margarine'}), frozenset({'whole m
ilk'}), 0.5379310344827587), (frozenset({'other vegetables', 'rolls/buns'}), frozenset({'
whole milk'}), 0.4200477326968974), (frozenset({'citrus fruit', 'domestic eggs'}), frozen
set({'whole milk'}), 0.5490196078431373), (frozenset({'tropical fruit', 'rolls/buns'}), f
rozenset({'whole milk'}), 0.4462809917355372), (frozenset({'pip fruit', 'rolls/buns'}), f
rozenset({'whole milk'}), 0.44525547445255476), (frozenset({'tropical fruit', 'pip fruit'
}), frozenset({'whole milk'}), 0.4129353233830846), (frozenset({'frozen vegetables', 'who
le milk'}), frozenset({'other vegetables'}), 0.472636815920398), (frozenset({'other veget
ables', 'frozen vegetables'}), frozenset({'whole milk'}), 0.5428571428571428), (frozenset
({'tropical fruit', 'curd'}), frozenset({'other vegetables'}), 0.5148514851485149), (froz
enset({'tropical fruit', 'fruit/vegetable juice'}), frozenset({'other vegetables'}), 0.48
14814814814815), (frozenset({'whipped/sour cream', 'rolls/buns'}), frozenset({'other vege
tables'}), 0.4583333333333333), (frozenset({'root vegetables', 'whipped/sour cream'}), fr
ozenset({'other vegetables'}), 0.5), (frozenset({'root vegetables', 'bottled water'}), fr
ozenset({'other vegetables'}), 0.44805194805194803), (frozenset({'root vegetables', 'curd
'}), frozenset({'other vegetables'}), 0.5046728971962617), (frozenset({'butter', 'whipped
/sour cream'}), frozenset({'other vegetables'}), 0.5700000000000001), (frozenset({'butter
', 'root vegetables'}), frozenset({'other vegetables'}), 0.5118110236220472), (frozenset(
{'butter', 'rolls/buns'}), frozenset({'other vegetables'}), 0.42424242424242425), (frozen
set({'tropical fruit', 'root vegetables'}), frozenset({'whole milk'}), 0.570048309178744)
, (frozenset({'root vegetables', 'domestic eggs'}), frozenset({'whole milk'}), 0.59574468
08510639), (frozenset({'soda', 'domestic eggs'}), frozenset({'whole milk'}), 0.4180327868
852459), (frozenset({'yogurt', 'domestic eggs'}), frozenset({'whole milk'}), 0.5390070921
985816), (frozenset({'tropical fruit', 'domestic eggs'}), frozenset({'whole milk'}), 0.60
71428571428571), (frozenset({'brown bread', 'soda'}), frozenset({'whole milk'}), 0.403225
8064516129), (frozenset({'root vegetables', 'brown bread'}), frozenset({'whole milk'}), 0
.56), (frozenset({'yogurt', 'brown bread'}), frozenset({'whole milk'}), 0.489510489510489
53), (frozenset({'tropical fruit', 'brown bread'}), frozenset({'whole milk'}), 0.53333333
33333333), (frozenset({'root vegetables', 'soda'}), frozenset({'whole milk'}), 0.43715846
994535523), (frozenset({'yogurt', 'root vegetables'}), frozenset({'whole milk'}), 0.56299
2125984252), (frozenset({'root vegetables', 'pastry'}), frozenset({'whole milk'}), 0.5185
185185185185), (frozenset({'yogurt', 'pastry'}), frozenset({'whole milk'}), 0.51724137931
03449), (frozenset({'tropical fruit', 'pastry'}), frozenset({'whole milk'}), 0.5076923076
923077), (frozenset({'yogurt', 'coffee'}), frozenset({'whole milk'}), 0.5208333333333333)
, (frozenset({'whipped/sour cream', 'whole milk'}), frozenset({'other vegetables'}), 0.45
42586750788643), (frozenset({'other vegetables', 'whipped/sour cream'}), frozenset({'whol
e milk'}), 0.5070422535211268), (frozenset({'whole milk', 'pork'}), frozenset({'other veg
etables'}), 0.45871559633027514), (frozenset({'other vegetables', 'pork'}), frozenset({'w
hole milk'}), 0.4694835680751174), (frozenset({'other vegetables', 'soda'}), frozenset({'
whole milk'}), 0.42546583850931674), (frozenset({'whipped/sour cream', 'soda'}), frozense
t({'whole milk'}), 0.4736842105263158), (frozenset({'root vegetables', 'whole milk'}), fr
ozenset({'other vegetables'}), 0.47401247401247404), (frozenset({'other vegetables', 'roo
t vegetables'}), frozenset({'whole milk'}), 0.4892703862660944), (frozenset({'sugar', 'ot
her vegetables'}), frozenset({'whole milk'}), 0.5849056603773586), (frozenset({'sugar', '
whole milk'}), frozenset({'other vegetables'}), 0.4189189189189189), (frozenset({'root ve
getables', 'rolls/buns'}), frozenset({'other vegetables'}), 0.502092050209205), (frozense
t({'tropical fruit', 'root vegetables'}), frozenset({'other vegetables'}), 0.584541062801
9324), (frozenset({'yogurt', 'curd'}), frozenset({'whole milk'}), 0.5823529411764706), (f
rozenset({'tropical fruit', 'curd'}), frozenset({'whole milk'}), 0.6336633663366337), (fr
ozenset({'tropical fruit', 'curd'}), frozenset({'yogurt'}), 0.5148514851485149), (frozens
et({'tropical fruit', 'bottled water'}), frozenset({'whole milk'}), 0.4340659340659341),
(frozenset({'butter', 'bottled water'}), frozenset({'whole milk'}), 0.6022727272727273),
(frozenset({'yogurt', 'bottled water'}), frozenset({'whole milk'}), 0.42035398230088494),
(frozenset({'citrus fruit', 'bottled water'}), frozenset({'whole milk'}), 0.4360902255639

0975), (frozenset({'butter', 'tropical fruit'}), frozenset({'whole milk'}), 0.62244897959
18368), (frozenset({'butter', 'citrus fruit'}), frozenset({'whole milk'}), 0.555555555555
5555), (frozenset({'yogurt', 'tropical fruit'}), frozenset({'whole milk'}), 0.51736111111
1111), (frozenset({'yogurt', 'citrus fruit'}), frozenset({'whole milk'}), 0.474184037558
6856), (frozenset({'tropical fruit', 'citrus fruit'}), frozenset({'whole milk'}), 0.45408
163265306123), (frozenset({'butter', 'yogurt'}), frozenset({'whole milk'}), 0.63888888888
88888), (frozenset({'long life bakery product', 'whole milk'}), frozenset({'other vegetab
les'}), 0.4210526315789474), (frozenset({'other vegetables', 'long life bakery product'})
, frozenset({'whole milk'}), 0.5333333333333333), (frozenset({'root vegetables', 'citrus
fruit', 'whole milk'}), frozenset({'other vegetables'}), 0.6333333333333333), (frozenset(
{'other vegetables', 'citrus fruit', 'whole milk'}), frozenset({'root vegetables'}), 0.44
53125), (frozenset({'root vegetables', 'other vegetables', 'citrus fruit'}), frozenset({'
whole milk'}), 0.5588235294117647), (frozenset({'yogurt', 'other vegetables', 'rolls/buns
'}), frozenset({'whole milk'}), 0.5221238938053098), (frozenset({'root vegetables', 'whol
e milk', 'rolls/buns'}), frozenset({'other vegetables'}), 0.488), (frozenset({'other vege
tables', 'root vegetables', 'rolls/buns'}), frozenset({'whole milk'}), 0.5083333333333333
), (frozenset({'tropical fruit', 'root vegetables', 'whole milk'}), frozenset({'other veg
etables'}), 0.5847457627118644), (frozenset({'other vegetables', 'tropical fruit', 'whole
milk'}), frozenset({'root vegetables'}), 0.4107142857142857), (frozenset({'root vegetable
s', 'other vegetables', 'tropical fruit'}), frozenset({'whole milk'}), 0.5702479338842975
), (frozenset({'root vegetables', 'whipped/sour cream', 'whole milk'}), frozenset({'other
vegetables'}), 0.5483870967741935), (frozenset({'other vegetables', 'root vegetables', 'w
hipped/sour cream'}), frozenset({'whole milk'}), 0.6071428571428571), (frozenset({'yogurt
', 'whole milk', 'pip fruit'}), frozenset({'other vegetables'}), 0.5319148936170213), (fr
ozenset({'yogurt', 'other vegetables', 'pip fruit'}), frozenset({'whole milk'}), 0.625),
(frozenset({'yogurt', 'whipped/sour cream', 'whole milk'}), frozenset({'other vegetables'
}), 0.514018691588785), (frozenset({'yogurt', 'other vegetables', 'whipped/sour cream'}),
frozenset({'whole milk'}), 0.55), (frozenset({'other vegetables', 'tropical fruit', 'whol
e milk'}), frozenset({'yogurt'}), 0.4464285714285714), (frozenset({'yogurt', 'tropical fr
uit', 'whole milk'}), frozenset({'other vegetables'}), 0.5033557046979866), (frozenset({'
yogurt', 'other vegetables', 'tropical fruit'}), frozenset({'whole milk'}), 0.61983471074
38016), (frozenset({'root vegetables', 'whole milk', 'pip fruit'}), frozenset({'other veg
etables'}), 0.6136363636363636), (frozenset({'other vegetables', 'whole milk', 'pip fruit
'}), frozenset({'root vegetables'}), 0.40601503759398494), (frozenset({'other vegetables'
, 'root vegetables', 'pip fruit'}), frozenset({'whole milk'}), 0.675), (frozenset({'yogur
t', 'root vegetables', 'whole milk'}), frozenset({'other vegetables'}), 0.538461538461538
5), (frozenset({'yogurt', 'other vegetables', 'root vegetables'}), frozenset({'whole milk
'}), 0.6062992125984252), (frozenset({'other vegetables', 'whole milk', 'fruit/vegetable
juice'}), frozenset({'yogurt'}), 0.4854368932038835), (frozenset({'yogurt', 'whole milk',
'fruit/vegetable juice'}), frozenset({'other vegetables'}), 0.5376344086021505), (frozens
et({'yogurt', 'other vegetables', 'fruit/vegetable juice'}), frozenset({'whole milk'}), 0
.6172839506172838), (frozenset({'tropical fruit', 'root vegetables', 'whole milk'}), froz
enset({'yogurt'}), 0.4745762711864407), (frozenset({'yogurt', 'tropical fruit', 'root veg
etables'}), frozenset({'whole milk'}), 0.7000000000000001)]
{margarine} ---> {whole milk}: conf = 0.413, sup = 0.024
{beef} ---> {whole milk}: conf = 0.405, sup = 0.021
{frozen vegetables} ---> {whole milk}: conf = 0.425, sup = 0.02
{domestic eggs} ---> {whole milk}: conf = 0.473, sup = 0.03
{whipped/sour cream} ---> {whole milk}: conf = 0.45, sup = 0.032
{whipped/sour cream} ---> {other vegetables}: conf = 0.403, sup = 0.029
{root vegetables} ---> {whole milk}: conf = 0.449, sup = 0.049
{root vegetables} ---> {other vegetables}: conf = 0.435, sup = 0.047
{tropical fruit} ---> {whole milk}: conf = 0.403, sup = 0.042
{curd} ---> {whole milk}: conf = 0.49, sup = 0.026
{yogurt} ---> {whole milk}: conf = 0.402, sup = 0.056
{butter} ---> {whole milk}: conf = 0.497, sup = 0.028
{yogurt, other vegetables} ---> {whole milk}: conf = 0.513, sup = 0.022
{root vegetables, whole milk} ---> {other vegetables}: conf = 0.474, sup = 0.023
{other vegetables, root vegetables} ---> {whole milk}: conf = 0.489, sup = 0.023

 --------------- Minimum Support --------------- 0.02 ----------------

[(frozenset({'margarine'}), frozenset({'whole milk'}), 0.4131944444444444), (frozenset({'
beef'}), frozenset({'whole milk'}), 0.4050387596899225), (frozenset({'frozen vegetables'}
), frozenset({'whole milk'}), 0.4249471458773784), (frozenset({'domestic eggs'}), frozens
et({'whole milk'}), 0.47275641025641024), (frozenset({'whipped/sour cream'}), frozenset({
'whole milk'}), 0.449645390070922), (frozenset({'whipped/sour cream'}), frozenset({'other
vegetables'}), 0.40283687943262414), (frozenset({'root vegetables'}), frozenset({'whole m
ilk'}), 0.44869402985074625), (frozenset({'root vegetables'}), frozenset({'other vegetabl
es'}), 0.43470149253731344), (frozenset({'tropical fruit'}), frozenset({'whole milk'}), 0
.40310077519379844), (frozenset({'curd'}), frozenset({'whole milk'}), 0.4904580152671756)
, (frozenset({'yogurt'}), frozenset({'whole milk'}), 0.40160349854227406), (frozenset({'b

```
utter'}), frozenset({'whole milk'}), 0.4972477064220184), (frozenset({'yogurt', 'other ve
getables'}), frozenset({'whole milk'}), 0.5128805620608898), (frozenset({'root vegetables
', 'whole milk'}), frozenset({'other vegetables'}), 0.47401247401247404), (frozenset({'ot
her vegetables', 'root vegetables'}), frozenset({'whole milk'}), 0.4892703862660944)]
{root vegetables} ---> {whole milk}:  conf = 0.449, sup = 0.049
{root vegetables} ---> {other vegetables}:  conf = 0.435, sup = 0.047
{tropical fruit} ---> {whole milk}:  conf = 0.403, sup = 0.042
{yogurt} ---> {whole milk}:  conf = 0.402, sup = 0.056

 --------------- Minimum Support --------------- 0.035 ----------------

[(frozenset({'root vegetables'}), frozenset({'whole milk'}), 0.44869402985074625), (froze
nset({'root vegetables'}), frozenset({'other vegetables'}), 0.43470149253731344), (frozen
set({'tropical fruit'}), frozenset({'whole milk'}), 0.40310077519379844), (frozenset({'yo
gurt'}), frozenset({'whole milk'}), 0.40160349854227406)]
{yogurt} ---> {whole milk}:  conf = 0.402, sup = 0.056

 --------------- Minimum Support --------------- 0.05 ----------------

[(frozenset({'yogurt'}), frozenset({'whole milk'}), 0.40160349854227406)]

 --------------- Minimum Support --------------- 0.065 ----------------

[]

 --------------- Minimum Support --------------- 0.08 ----------------

[]

 --------------- Minimum Support --------------- 0.095 ----------------

[]

 --------------- Minimum Support --------------- 0.11 ----------------

[]

 --------------- Minimum Support --------------- 0.125 ----------------

[]

 --------------- Minimum Support --------------- 0.14 ----------------

[]
```

Task 2

We can find a relationship between the confidence level and number of rules found for a certain support value. For this, plot the number of rules found on y-axis and confidence levels on x-axis for different support values. Use 10%, 20%, 30%, 40%, 50% confidence levels for each of 2%, 3%, 4%, 5% support levels in the same figure. Plot a separate line for each support level.

In [5]:

```
confidence_levels = [0.1, 0.2, 0.3, 0.4, 0.5]
support_levels = [0.02, 0.03, 0.04, 0.05]
```

In [6]:

```
fig, ax = plt.subplots()

for support in support_levels:
    frequent_itemsets, support_info = apriori(dataset, min_support=support)

    counts_of_rules = []

    for confidence in confidence_levels:
        generated_rules = generate_rules(frequent_itemsets, support_info, min_confidence
=confidence)
        counts_of_rules.append(len(generated_rules))
```
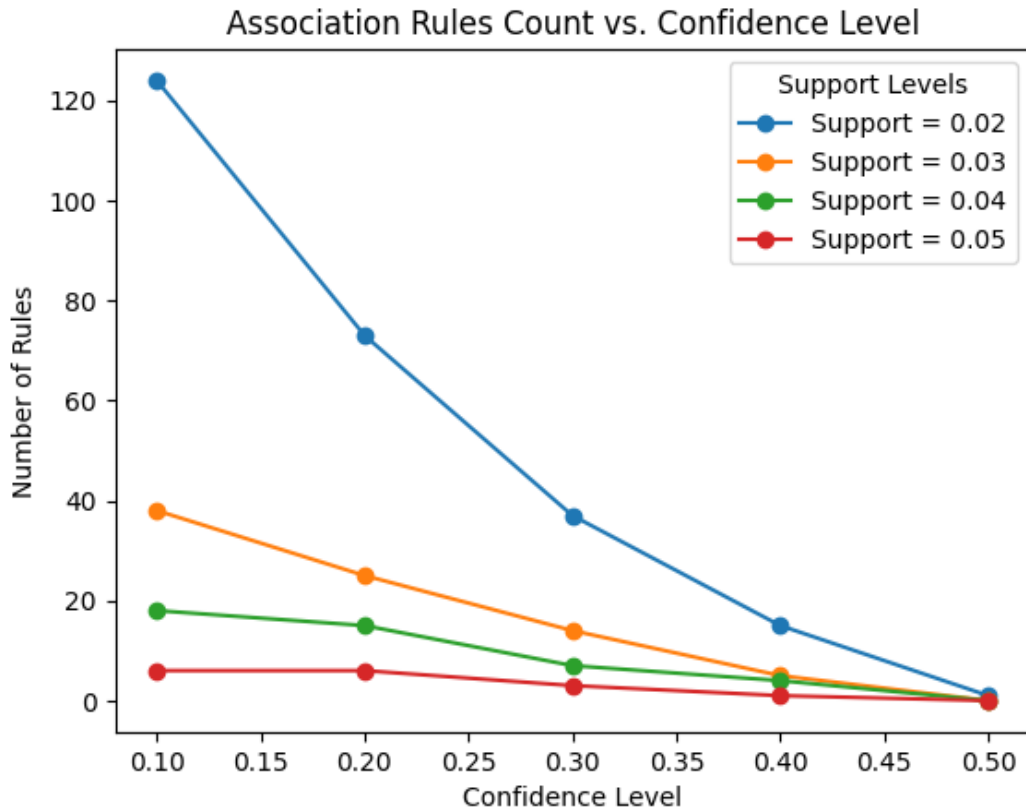
```
    clear_output(wait=True)

    ax.plot(confidence_levels, counts_of_rules, marker='o', label=f'Support = {support}'
)

ax.set_xlabel('Confidence Level')
ax.set_ylabel('Number of Rules')
ax.legend(title='Support Levels')
plt.title('Association Rules Count vs. Confidence Level')
plt.show()
```


Part 2

FPgrowth

I am using the pyfpgrowth library. It contains Python implementation of FPgrowth algorithm implementation by Evan Dempsey

In [10]:

```
fig, plot_ax = plt.subplots()

for support in support_levels:
    abs_support = int(support * len(data_set_list))

    frequent_patterns = pyfpgrowth.find_frequent_patterns(data_set_list, abs_support)

    num_rules_at_confidence = []

    for confidence in confidence_levels:
        generated_rules = pyfpgrowth.generate_association_rules(frequent_patterns, confi
dence)
        num_rules_at_confidence.append(len(generated_rules))

    plot_ax.plot(confidence_levels, num_rules_at_confidence, marker='.', linestyle='-',
label=f'Support {support}')

plot_ax.set_title('Association Rule Count vs. Confidence Level')
plot_ax.set_xlabel('Confidence Level')
plot_ax.set_ylabel('Number of Generated Rules')
```
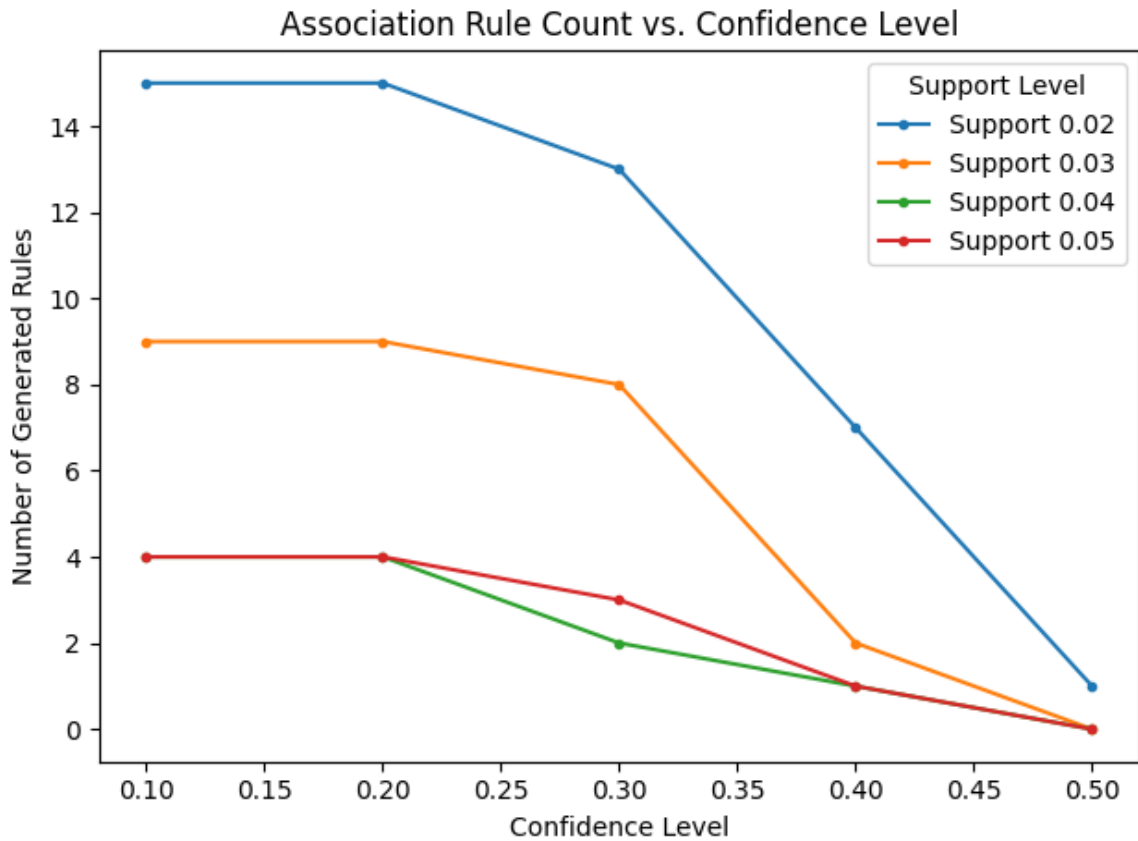
```
plot_ax.legend(title='Support Level', loc='upper right')

plt.tight_layout()
plt.show()
```


Association Rule Count vs. Confidence Level

Part 3

Interest Factor

In [18]:

```
def calc_interest_factor(sup_a, sup_b, sup_ab):
    return sup_ab / (sup_a * sup_b)

def order_rules(all_rules, sup_data, top_n=None):
    by_support = sorted(all_rules, key=lambda r: -sup_data[r[0]])
    by_confidence = sorted(all_rules, key=lambda r: -r[2])
    by_interest = sorted(all_rules, key=lambda r: calc_interest_factor(sup_data[r[0]], s
up_data[r[1]], sup_data[r[0] | r[1]]))

    if top_n:
        return by_support[:top_n], by_confidence[:top_n], by_interest[:top_n]
    else:
        return by_support, by_confidence, by_interest

def display_top_rules(sorted_rules, limit=5):
    for index, (ante, conse, conf) in enumerate(sorted_rules[:limit], 1):
        print(f"Rule {index}: {', '.join(str(item) for item in ante)} --> {', '.join(str
(item) for item in conse)} with confidence {conf}")
    print(" ")
```

In [21]:

```
frequent_sets, supp_data = apriori(dataset, 0.02, False)
derived_rules = generate_rules(frequent_sets, supp_data, 0.3, False)

support_sorted, confidence_sorted, interest_sorted = order_rules(derived_rules, supp_data
, 5)

print("Top 5 by Support:")
display_top_rules(support_sorted)
```

```python
print("Top 5 by Confidence:")
display_top_rules(confidence_sorted)

print("Top 5 by Interest Factor:")
display_top_rules(interest_sorted)

common_support_confidence = set(support_sorted) & set(confidence_sorted)
common_support_interest = set(support_sorted) & set(interest_sorted)
common_confidence_interest = set(confidence_sorted) & set(interest_sorted)

if common_support_confidence:
    print("Shared between Support & Confidence:")
    display_top_rules(list(common_support_confidence), len(common_support_confidence))

print(" ")

if common_support_interest:
    print("Shared between Support & Interest Factor:")
    display_top_rules(list(common_support_interest), len(common_support_interest))

if common_confidence_interest:
    print("Shared between Confidence & Interest Factor:")
    display_top_rules(list(common_confidence_interest), len(common_confidence_interest))
```

```
Top 5 by Support:
Rule 1: other vegetables --> whole milk with confidence 0.38675775091960063
Rule 2: rolls/buns --> whole milk with confidence 0.30790491984521834
Rule 3: yogurt --> other vegetables with confidence 0.3112244897959184
Rule 4: yogurt --> whole milk with confidence 0.40160349854227406
Rule 5: bottled water --> whole milk with confidence 0.31094756209751606

Top 5 by Confidence:
Rule 1: yogurt, other vegetables --> whole milk with confidence 0.5128805620608898
Rule 2: butter --> whole milk with confidence 0.4972477064220184
Rule 3: curd --> whole milk with confidence 0.4904580152671756
Rule 4: other vegetables, root vegetables --> whole milk with confidence 0.48927038626609
44
Rule 5: root vegetables, whole milk --> other vegetables with confidence 0.47401247401247
404

Top 5 by Interest Factor:
Rule 1: rolls/buns --> whole milk with confidence 0.30790491984521834
Rule 2: bottled water --> whole milk with confidence 0.31094756209751606
Rule 3: sausage --> whole milk with confidence 0.3181818181818182
Rule 4: newspapers --> whole milk with confidence 0.34267515923566877
Rule 5: frankfurter --> whole milk with confidence 0.3482758620689655


Shared between Support & Interest Factor:
Rule 1: rolls/buns --> whole milk with confidence 0.30790491984521834
Rule 2: bottled water --> whole milk with confidence 0.31094756209751606
```