Course: CS634
Subject:: Midterm Project Implementation
Topic: Apriori Algorithm Implementation
Name: Richard Nardone
Email: rmn9@njit.edu
Github: https://github.com/Rich-Nardone/Apriori_Implementation.git

Datasets

1. amazon_transactions.csv

```
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition
Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition,Beginning Programming with Java,,
Android Programming: The Big Nerd Ranch,Beginning Programming with Java,Java 8 Pocket Guide,,
A Beginner’s Guide,Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition,,
A Beginner’s Guide,Head First Java 2nd Edition,Beginning Programming with Java,,
Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,,
Java For Dummies,Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition,Beginning Programming with Java,
Beginning Programming with Java,Java 8 Pocket Guide,C++ Programming in Easy Steps,,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,HTML and CSS: Design and Build Websites,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Java 8 Pocket Guide,HTML and CSS: Design and Build Websites
Java For Dummies,Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition,,
Java For Dummies,Android Programming: The Big Nerd Ranch,,,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,Android Programming: The Big Nerd Ranch,
Head First Java 2nd Edition,Beginning Programming with Java,Java 8 Pocket Guide,,
Android Programming: The Big Nerd Ranch,Head First Java 2nd Edition,,,
A Beginner’s Guide,Java: The Complete Reference,Java For Dummies,,
```

2. bestbuy_transactions.csv

```
Desk Top,Printer,Flash Drive,Microsoft Office,Speakers,Anti-Virus,,,,
Lab Top,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus,,,,,
Lab Top,Printer,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,External Hard-Drive,,,
Lab Top,Printer,Flash Drive,Anti-Virus,External Hard-Drive,Lab Top Case,,,,
Lab Top,Flash Drive,Lab Top Case,Anti-Virus,,,,,
Lab Top,Printer,Flash Drive,Microsoft Office,,,,,,
Desk Top,Printer,Flash Drive,Microsoft Office,,,,,,
Lab Top,External Hard-Drive,Anti-Virus,,,,,,,
Desk Top,Printer,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus,Speakers,External Hard-Drive,,
Digital Camera,Lab Top,Desk Top,Printer,Flash Drive,Microsoft Office,Lab Top Case,Anti-Virus,External Hard-Drive,Speakers
Lab Top,Desk Top,Lab Top Case,External Hard-Drive,Speakers,Anti-Virus,,,,
Digital Camera,Lab Top,Lab Top Case,External Hard-Drive,Anti-Virus,Speakers,,,,
Digital Camera,Speakers,,,,,,,,
Digital Camera,Desk Top,Printer,Flash Drive,Microsoft Office,,,,,
Printer,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,Speakers,External Hard-Drive,,,
Digital Camera,Flash Drive,Microsoft Office,Anti-Virus,Lab Top Case,External Hard-Drive,Speakers,,,
Digital Camera,Lab Top,Lab Top Case,,,,,,
Digital Camera,Lab Top Case,Speakers,,,,,,,
Digital Camera,Lab Top,Printer,Flash Drive,Microsoft Office,Speakers,Lab Top Case,Anti-Virus,,
Digital Camera,Lab Top,Speakers,Anti-Virus,Lab Top Case,,,,,
```

3. kmart_transactions.csv

```
Decorative Pillows,Quilts,Embroidered Bedspread,,,,
Embroidered Bedspread,Shams,Kids Bedding,Bedding Collections,Bed Skirts,Bedspreads,Sheets
Decorative Pillows,Quilts,Embroidered Bedspread,Shams,Kids Bedding,Bedding Collections,
Kids Bedding,Bedding Collections,Sheets,Bedspreads,Bed Skirts,,
Decorative Pillows,Kids Bedding,Bedding Collections,Sheets,Bed Skirts,Bedspreads,
Bedding Collections,Bedspreads,Bed Skirts,Sheets,Shams,Kids Bedding,
Decorative Pillows,Quilts,,,,,
Decorative Pillows,Quilts,Embroidered Bedspread,,,,
Bedspreads,Bed Skirts,Shams,Kids Bedding,Sheets,,
Quilts,Embroidered Bedspread,Bedding Collections,,,,
Bedding Collections,Bedspreads,Bed Skirts,Kids Bedding,Shams,Sheets,
Decorative Pillows,Quilts,,,,,
Embroidered Bedspread,Shams,,,,,
Sheets,Shams,Bed Skirts,Kids Bedding,,,
Decorative Pillows,Quilts,,,,,
Decorative Pillows,Kids Bedding,Bed Skirts,Shams,,,
Decorative Pillows,Shams,Bed Skirts,,,,
Quilts,Sheets,Kids Bedding,,,,
Shams,Bed Skirts,Kids Bedding,Sheets,,,
Decorative Pillows,Bedspreads,Shams,Sheets,Bed Skirts,Kids Bedding,
```

4. nike_transactions.csv

```
Running Shoe,Socks,Sweatshirts,Modern Pants,,,,,,
Running Shoe,Socks,Sweatshirts,,,,,,,
Running Shoe,Socks,Sweatshirts,Modern Pants,,,,,,
Running Shoe,Sweatshirts,Modern Pants,,,,,,,
Running Shoe,Socks,Sweatshirts,Modern Pants,Soccer Shoe,,,,,
Running Shoe,Socks,Sweatshirts,,,,,,,
Running Shoe,Socks,Sweatshirts,Modern Pants,Tech Pants,Rash Guard,Hoodies,,,
Swimming Shirt,Socks,Sweatshirts,,,,,,,
Swimming Shirt,Rash Guard,Dry Fit V-Nick,Hoodies,Tech Pants,,,,,
Swimming Shirt,Rash Guard,Dry Fit V-Nick,,,,,,,
Swimming Shirt,Rash Guard,Dry Fit V-Nick,,,,,,,
Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Rash Guard,Hoodies,Tech Pants,Dry Fit V-Nick
Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Rash Guard,Tech Pants,Dry Fit V-Nick,Hoodies
Running Shoe,Swimming Shirt,Rash Guard,Tech Pants,Hoodies,Dry Fit V-Nick,,,,
Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Dry Fit V-Nick,Rash Guard,Tech Pants,,
Swimming Shirt,Soccer Shoe,Hoodies,Dry Fit V-Nick,Tech Pants,Rash Guard,,,,
Running Shoe,Socks,,,,,,,,
Socks,Sweatshirts,Modern Pants,Soccer Shoe,Hoodies,Rash Guard,Tech Pants,Dry Fit V-Nick,,
Running Shoe,Swimming Shirt,Rash Guard,,,,,,,
Running Shoe,Swimming Shirt,Socks,Sweatshirts,Modern Pants,Soccer Shoe,Hoodies,Tech Pants,Rash Guard,Dry Fit V-Nick
```

5. espn_transactions.csv

```
NFL,MMA,NCAAF,NHL,,,,,
NFL,MMA,NCAAF,,,,,,
NFL,MMA,NCAAF,NHL,,,,,
NFL,NCAAF,NBA,,,,,,
NFL,MMA,NCAAF,Golf,Soccer,,,,
NFL,MMA,NCAAF,,,,,,
NFL,MMA,NCAAF,NCAAH,NHL,Soccer,Golf,,
MLB,MMA,NCAAF,,,,,,
Golf,Soccer,MLB,NCAAB,NCAAH,,,,
NFL,Soccer,NBA,,,,,,
NBA,Soccer,MLB,,,,,,
NFL,NBA,MLB,NHL,Soccer,NCAAF,NCAAB,NCAAH,MMA,Golf
NFL,NBA,MLB,NHL,Soccer,NCAAF,NCAAB,NCAAH,MMA,Golf
NFL,NCAAF,Soccer,NBA,MMA,Golf,,,
NFL,MLB,MMA,NCAAF,Golf,NCAAH,Soccer,NBA,,
Golf,MMA,NHL,NFL,NBA,Soccer,,,
NFL,MMA,,,,,,,
MLB,NHL,Soccer,NCAAF,NCAAB,NCAAH,MMA,Golf,,
NFL,MLB,Soccer,,,,,,
NFL,NBA,MLB,NHL,Soccer,NCAAF,NCAAB,NCAAH,MMA,Golf
```

Code

Imports:

```
from itertools import combinations,permutations
import pandas as pd
import numpy as np
```

Imported three libraries: pandas, numpy, and itertools

User Interface:

```
print("Welcome to Richards Apriori Algorithm.")
print("\t 1)  Amazon")
print("\t 2)  BestBuy")
print("\t 3)  Kmart")
print("\t 4)  Nike")
print("\t 5)  ESPN")
print("\t 6)  Custom")
dataset = int(input("\nPlease choose your dataset: "))
if(dataset == 1):
    file = 'amazon_transactions.csv'
elif(dataset == 2):
    file = 'bestbuy_transactions.csv'
elif(dataset == 3):
    file = 'kmart_transactions.csv'
elif(dataset == 4):
    file = 'nike_transactions.csv'
elif(dataset == 5):
    file = 'espn_transactions.csv'
elif(dataset == 6):
    file = input('Please enter your CSV files name:')
min_support = input('Please enter the minimum support(0<=minimum support<=1):')
min_confidence = input('Please enter the minimum confidence(0<=minimum confidence<=1):')
result = apriori(file,float(min_support),float(min_confidence))
print()
print('Under the minimum support of',min_support,'this is your frequent items list:')
for i in result[0]:
    print('\t'+str(i))
print()
print('Under the minimum support of',min_support,'and the minimum confidence of',min_confidence,'these are your association rules
for i in result[1].keys():
    print('\t'+i,'Confidence: '+str(result[1][i]))
```

Prompt the user to choose a database from the presets or create their own. If you choose to create your own database you enter the file name. User is prompted for minimum support, then prompted for minimum confidence. The Apriori algorithm is then run and the frequent item sets are returned as well as the association rules.

Helper Functions:

get_support:

```
def get_support(items,trans):
    count = 0
    for tran in trans.keys():
        flag = True
        for i in items:
            if(i not in trans[tran]):
                flag = False
        if(flag):
            count+=1
    return count
```

get_support takes a list of items and the transactions to get the support of the item list. get_support is used primarily for getting the confidence of an association rule.

get_confidence:

```
def get_confidence(items1,items2,trans):
    support1 = get_support(items1,trans)
    support2 = get_support(items2,trans)
    if(support2 == 0):
        return 0
    conf = support1/support2
    return conf
```

get_confidence takes two item lists and the transactions to get the confidence of an association rule. items1 is the union of the association rule and items2 is the first part of the association rule. get_confidence checks for division by zero and returns zero if division by zero arises. The function returns the support of the union of items divided by the support of the first part of the association rule.

apriori:

```
def apriori(csv,min_support,min_confidence):
    df = pd.read_csv(csv,header = None).T
    dic = df.to_dict()
    items=[]
    trans = {}
    for key, value in dic.items():
        trans[key] = []
        for j in value:
            if(not pd.isnull(value[j])):
                trans[key].append(value[j])
                if(value[j] not in items):
                    items.append(value[j])
    min_support = len(trans)*min_support
    k= 1
    result = []
    L = items
    while(L != []):
        L = [','.join(i) for i in combinations(L, k)]
        C = pd.DataFrame(index = trans.keys(),columns=L)
        support_count = support(C,trans)
        support_count = support_count[support_count['Value'] >= min_support]
        L = []
        for l in support_count.index:
            hold = l.split(',')
            result.append(hold)
            for i in hold:
                if(i not in L):
                    L.append(i)
        k+=1
    association_rules = get_association_rules(result,trans,min_support,min_confidence)

    return [result,association_rules]
```

apriori takes 3 parameters: csv (name of the csv file containing transactions lists), min_support (minimum support specified by the user), min_confidence (minimum confidence specified by the user). The csv file is converted to a dataframe then a dictionary. An item list is created by looping through the data frame and adding items not already included in the list not including nan.

L is initialized to the items in the list. The while loop runs until the L is empty. Each iteration k is incremented by 1. L is assigned values combinations of all items of size k. C is set to the transaction keys index and the support of each combination in L is computed. The values greater than or equal to min support are kept and they are appended to a list of frequent items. The loop returns to the start with the items in L being the ones left over at the end.

The association rules are computed and returned along with the frequent item sets.

get_association_rules

```
def get_association_rules(frequent_item_sets,trans,min_support,min_confidence):
    rules = {}
    for frequent_item_set in frequent_item_sets:
        if(len(frequent_item_set)>1):
            for i in range(1,len(frequent_item_set)):
                for j in [list(k) for k in combinations(frequent_item_set,i)]:
                    other = [z for z in frequent_item_set if z not in j]
                    conf1 = get_confidence(frequent_item_set,j,trans)
                    rule1 = str(j)+' -> '+str(other)
                    conf2 = get_confidence(frequent_item_set,other,trans)
                    rule2 = str(other)+' -> '+str(j)
                    if(rule1 not in rules.keys() and conf1>=min_confidence):
                        rules[rule1] = conf1
                    if(rule2 not in rules.keys() and conf2>=min_confidence):
                        rules[rule2] = conf2
    return rules
```

get_association_rules takes 4 parameters: frequent_item_sets (list of frequent item sets from apriori algorithm), trans (transaction lists), min_support, and min_confidence.

Creates a dictionary of rules by iterating through each frequent item set and creating every combination of every size of that set. The confidence is then computed for the rule as well as the complement of the rule. Before the rule is entered into the dictionary the method checks if it already exists.

support:

```
def support(df,trans):
    for item in df.columns:
        item_split = item.split(',')
        for key,items in trans.items():
            flag =0
            for i in item_split:
                if(i not in items):
                    flag = 1
            if(flag == 0):
                if(pd.isnull(df[item][key])):
                    df[item][key] = 1
                else:
                    df[item][key] += 1
    return pd.DataFrame(df.sum(),columns=['Value'])
```

support takes two parameters, a dataframe and the transaction list. This method is used primarily during the apriori algorithm to create a dataframe of supports.

Example 1:

```
Welcome to Richards Apriori Algorithm.
        1)  Amazon
        2)  BestBuy
        3)  Kmart
        4)  Nike
        5)  ESPN
        6)  Custom

Please choose your dataset: 4
Please enter the minimum support(0<=minimum support<=1):.5
Please enter the minimum confidence(0<=minimum confidence<=1):.5

Under the minimum support of .5 this is your frequent items list:
        ['Running Shoe']
        ['Socks']
        ['Sweatshirts']
        ['Modern Pants']
        ['Rash Guard']
        ['Swimming Shirt']
        ['Dry Fit V-Nick']
        ['Running Shoe', 'Socks']
        ['Running Shoe', 'Sweatshirts']
        ['Socks', 'Sweatshirts']
        ['Sweatshirts', 'Modern Pants']
        ['Rash Guard', 'Swimming Shirt']
        ['Rash Guard', 'Dry Fit V-Nick']
        ['Running Shoe', 'Socks', 'Sweatshirts']

Under the minimum support of .5 and the minimum confidence of .5 these are your association rules:
        ['Running Shoe'] -> ['Socks'] Confidence: 0.7857142857142857
        ['Socks'] -> ['Running Shoe'] Confidence: 0.8461538461538461
        ['Running Shoe'] -> ['Sweatshirts'] Confidence: 0.7857142857142857
        ['Sweatshirts'] -> ['Running Shoe'] Confidence: 0.8461538461538461
        ['Socks'] -> ['Sweatshirts'] Confidence: 0.9230769230769231
        ['Sweatshirts'] -> ['Socks'] Confidence: 0.9230769230769231
        ['Sweatshirts'] -> ['Modern Pants'] Confidence: 0.7692307692307693
        ['Modern Pants'] -> ['Sweatshirts'] Confidence: 1.0
        ['Rash Guard'] -> ['Swimming Shirt'] Confidence: 0.8333333333333334
        ['Swimming Shirt'] -> ['Rash Guard'] Confidence: 0.9090909090909091
        ['Rash Guard'] -> ['Dry Fit V-Nick'] Confidence: 0.8333333333333334
        ['Dry Fit V-Nick'] -> ['Rash Guard'] Confidence: 1.0
        ['Running Shoe'] -> ['Socks', 'Sweatshirts'] Confidence: 0.7142857142857143
        ['Socks', 'Sweatshirts'] -> ['Running Shoe'] Confidence: 0.8333333333333334
        ['Socks'] -> ['Running Shoe', 'Sweatshirts'] Confidence: 0.7692307692307693
        ['Running Shoe', 'Sweatshirts'] -> ['Socks'] Confidence: 0.9090909090909091
        ['Sweatshirts'] -> ['Running Shoe', 'Socks'] Confidence: 0.7692307692307693
        ['Running Shoe', 'Socks'] -> ['Sweatshirts'] Confidence: 0.9090909090909091
```

Database: nike
Minimum Support: .5
Minimum Confidence: .5

Example 2:

```
Welcome to Richards Apriori Algorithm.
        1)  Amazon
        2)  BestBuy
        3)  Kmart
        4)  Nike
        5)  ESPN
        6)  Custom

Please choose your dataset: 1
Please enter the minimum support(0<=minimum support<=1):.45
Please enter the minimum confidence(0<=minimum confidence<=1):.5

Under the minimum support of .45 this is your frequent items list:
       ['A Beginner's Guide']
       ['Java: The Complete Reference']
       ['Java For Dummies']
       ['Android Programming: The Big Nerd Ranch']
       ['A Beginner's Guide', 'Java: The Complete Reference']
       ['A Beginner's Guide', 'Java For Dummies']
       ['Java: The Complete Reference', 'Java For Dummies']
       ['Java For Dummies', 'Android Programming: The Big Nerd Ranch']
       ['A Beginner's Guide', 'Java: The Complete Reference', 'Java For Dummies']

Under the minimum support of .45 and the minimum confidence of .5 these are your association rules:
       ['A Beginner's Guide'] -> ['Java: The Complete Reference'] Confidence: 0.8181818181818182
       ['Java: The Complete Reference'] -> ['A Beginner's Guide'] Confidence: 0.9
       ['A Beginner's Guide'] -> ['Java For Dummies'] Confidence: 0.8181818181818182
       ['Java For Dummies'] -> ['A Beginner's Guide'] Confidence: 0.6923076923076923
       ['Java: The Complete Reference'] -> ['Java For Dummies'] Confidence: 1.0
       ['Java For Dummies'] -> ['Java: The Complete Reference'] Confidence: 0.7692307692307693
       ['Java For Dummies'] -> ['Android Programming: The Big Nerd Ranch'] Confidence: 0.6923076923076923
       ['Android Programming: The Big Nerd Ranch'] -> ['Java For Dummies'] Confidence: 0.6923076923076923
       ['A Beginner's Guide'] -> ['Java: The Complete Reference', 'Java For Dummies'] Confidence: 0.8181818181818182
       ['Java: The Complete Reference', 'Java For Dummies'] -> ['A Beginner's Guide'] Confidence: 0.9
       ['Java: The Complete Reference'] -> ['A Beginner's Guide', 'Java For Dummies'] Confidence: 0.9
       ['A Beginner's Guide', 'Java For Dummies'] -> ['Java: The Complete Reference'] Confidence: 1.0
       ['Java For Dummies'] -> ['A Beginner's Guide', 'Java: The Complete Reference'] Confidence: 0.6923076923076923
       ['A Beginner's Guide', 'Java: The Complete Reference'] -> ['Java For Dummies'] Confidence: 1.0
```

Database: amazon
Minimum Support: .45
Minimum Confidence: .5

Example 3:

```
Welcome to Richards Apriori Algorithm.
         1)  Amazon
         2)  BestBuy
         3)  Kmart
         4)  Nike
         5)  ESPN
         6)  Custom

Please choose your dataset: 5
Please enter the minimum support(0<=minimum support<=1):.6
Please enter the minimum confidence(0<=minimum confidence<=1):.6

Under the minimum support of .6 this is your frequent items list:
        ['NFL']
        ['MMA']
        ['NCAAF']
        ['Soccer']
        ['NFL', 'MMA']
        ['NFL', 'NCAAF']
        ['MMA', 'NCAAF']

Under the minimum support of .6 and the minimum confidence of .6 these are your association rules:
        ['NFL'] -> ['MMA'] Confidence: 0.8125
        ['MMA'] -> ['NFL'] Confidence: 0.8666666666666667
        ['NFL'] -> ['NCAAF'] Confidence: 0.75
        ['NCAAF'] -> ['NFL'] Confidence: 0.8571428571428571
        ['MMA'] -> ['NCAAF'] Confidence: 0.8666666666666667
        ['NCAAF'] -> ['MMA'] Confidence: 0.9285714285714286
```

Database: espn
Minimum Support: .6
Minimum Confidence: .6

Example 4:

```
Welcome to Richards Apriori Algorithm.
         1)  Amazon
         2)  BestBuy
         3)  Kmart
         4)  Nike
         5)  ESPN
         6)  Custom

Please choose your dataset: 2
Please enter the minimum support(0<=minimum support<=1):.5
Please enter the minimum confidence(0<=minimum confidence<=1):.75

Under the minimum support of .5 this is your frequent items list:
        ['Printer']
        ['Flash Drive']
        ['Microsoft Office']
        ['Speakers']
        ['Anti-Virus']
        ['Lab Top']
        ['Lab Top Case']
        ['Printer', 'Flash Drive']
        ['Flash Drive', 'Microsoft Office']
        ['Flash Drive', 'Anti-Virus']
        ['Anti-Virus', 'Lab Top']
        ['Anti-Virus', 'Lab Top Case']
        ['Lab Top', 'Lab Top Case']

Under the minimum support of .5 and the minimum confidence of .75 these are your association rules:
        ['Printer'] -> ['Flash Drive'] Confidence: 1.0
        ['Flash Drive'] -> ['Printer'] Confidence: 0.7692307692307693
        ['Flash Drive'] -> ['Microsoft Office'] Confidence: 0.8461538461538461
        ['Microsoft Office'] -> ['Flash Drive'] Confidence: 1.0
        ['Flash Drive'] -> ['Anti-Virus'] Confidence: 0.7692307692307693
        ['Lab Top'] -> ['Anti-Virus'] Confidence: 0.8333333333333334
        ['Anti-Virus'] -> ['Lab Top Case'] Confidence: 0.8571428571428571
        ['Lab Top Case'] -> ['Anti-Virus'] Confidence: 0.8571428571428571
        ['Lab Top'] -> ['Lab Top Case'] Confidence: 0.8333333333333334
```

Database: bestbuy
Minimum Support: .5
Minimum Confidence: .75

Example 5:

```
Welcome to Richards Apriori Algorithm.
         1)  Amazon
         2)  BestBuy
         3)  Kmart
         4)  Nike
         5)  ESPN
         6)  Custom

Please choose your dataset: 3
Please enter the minimum support(0<=minimum support<=1):.5
Please enter the minimum confidence(0<=minimum confidence<=1):.6

Under the minimum support of .5 this is your frequent items list:
        ['Decorative Pillows']
        ['Shams']
        ['Kids Bedding']
        ['Bed Skirts']
        ['Sheets']
        ['Kids Bedding', 'Bed Skirts']
        ['Kids Bedding', 'Sheets']

Under the minimum support of .5 and the minimum confidence of .6 these are your association rules:
        ['Kids Bedding'] -> ['Bed Skirts'] Confidence: 0.8333333333333334
        ['Bed Skirts'] -> ['Kids Bedding'] Confidence: 0.9090909090909091
        ['Kids Bedding'] -> ['Sheets'] Confidence: 0.8333333333333334
        ['Sheets'] -> ['Kids Bedding'] Confidence: 1.0
```

Database: kmart
Minimum Support: .5
Minimum Confidence: .6