DATA MINING MIDTERM PROJECT

Full Name: Karthik Chandrashekar

NJIT UCID: KC69

Email: kc69@njit.edu

Programming language: Python

Amazon.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",,,U

BESTBUY.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,,,,

KMART.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[SEP]","Bed Skirts[SEP]",Bedspreads,

Bedding Collections,Bedspreads,"Bed Skirts[SEP]","Sheets[SEP]","Shams[SEP]","Kids Bedding

",

Decorative Pillows,Quilts,,,,,

"Decorative Pillows,",Quilts,Embroidered Bedspread,,,,

Bedspreads,"Bed Skirts[SEP]","Shams[SEP]","Kids Bedding

","Sheets[SEP]",,

Quilts,Embroidered Bedspread,Bedding Collections,,,,

Bedding Collections,Bedspreads,"Bed Skirts[SEP]","Kids Bedding

","Shams[SEP]",Sheets,

Decorative Pillows,Quilts,,,,,

Embroidered Bedspread,"Shams[SEP]",,,,,

Sheets,"Shams[SEP]","Bed Skirts[SEP]","Kids Bedding

",,,

Decorative Pillows,Quilts,,,,,

Decorative Pillows,"Kids Bedding

","Bed Skirts[SEP]","Shams[SEP]",,,

Decorative Pillows,"Shams[SEP]","Bed Skirts[SEP]",,,,

Quilts,Sheets,"Kids Bedding

",,,,

Shams,"Bed Skirts[SEP]","Kids Bedding

",Sheets,,,

Decorative Pillows,Bedspreads,"Shams[SEP]",Sheets,"Bed Skirts[SEP]","Kids Bedding

",

NIKE.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

How to run the file: python project.py datasetname.csv minimum_support(in decimal) minimum_confidence(in decimal)

```python
from csv import reader
import pandas as pd

def load_data_set():
    """
    Load a sample data set
    Returns:
        A data set: A list of transactions. Each transaction contains several items.
    """
    print("Hello, Please enter which data set you need \n 1) Press 1 for Amazon \n 2) Press 2 for BestBuy \n 3) Press 3 for Nike \n 4) Press 4 for KMart")
    while True:
        choice_of_data = input()
        if (choice_of_data == '1'):
            data = 'amazon.csv'
            print('User chose amazon dataset')
            break
        elif(choice_of_data == '2'):
            data = 'bestbuy.csv'
            print('User chose bestbuy dataset')
            break
        elif (choice_of_data == '3'):
            data = 'Nike.csv'
            print('User chose Nike dataset')
            break
        elif (choice_of_data == '4'):
            data = 'kmart.csv'
            print('User chose KMart dataset')
            break
        else:
            print("Please enter the right one")
    with open(data, 'r') as read_obj:
        # pass the file object to reader() to get the reader object
        csv_reader = reader(read_obj)

        # Pass reader object to list() to get a list of lists
        data_set = list(csv_reader)
        for items in data_set:
            for j in range(0, len(items)):
                for items1 in items:

                    if items1 == "":
                        items.remove("")

    return data_set
```

```python
def create_C1(data_set):
    """
    Create frequent candidate 1-itemset C1 by scaning data set.
    Args:
        data_set: A list of transactions. Each transaction contains several items.
    Returns:
        C1: A set which contains all frequent candidate 1-itemsets
    """
    C1 = set()
    for t in data_set:
        for item in t:
            item_set = frozenset([item])
            C1.add(item_set)
    return C1


def is_apriori(Ck_item, Lksub1):
    """
    Judge whether a frequent candidate k-itemset satisfy Apriori property.
    Args:
        Ck_item: a frequent candidate k-itemset in Ck which contains all frequent
                 candidate k-itemsets.
        Lksub1: Lk-1, a set which contains all frequent candidate (k-1)-itemsets.
    Returns:
        True: satisfying Apriori property.
        False: Not satisfying Apriori property.
    """
    for item in Ck_item:
        sub_Ck = Ck_item - frozenset([item])
        if sub_Ck not in Lksub1:
            return False
    return True


def create_Ck(Lksub1, k):
    """
    Create Ck, a set which contains all all frequent candidate k-itemsets
    by Lk-1's own connection operation.
    Args:
        Lksub1: Lk-1, a set which contains all frequent candidate (k-1)-itemsets.
        k: the item number of a frequent itemset.
    Return:
        Ck: a set which contains all all frequent candidate k-itemsets.
```

```
    Ck = set()
    len_Lksub1 = len(Lksub1)
    list_Lksub1 = list(Lksub1)
    for i in range(len_Lksub1):
        for j in range(1, len_Lksub1):
            l1 = list(list_Lksub1[i])
            l2 = list(list_Lksub1[j])
            l1.sort()
            l2.sort()
            if l1[0:k-2] == l2[0:k-2]:
                Ck_item = list_Lksub1[i] | list_Lksub1[j]
                # pruning
                if is_apriori(Ck_item, Lksub1):
                    Ck.add(Ck_item)
    return Ck


def generate_Lk_by_Ck(data_set, Ck, min_support, support_data):
    """
    Generate Lk by executing a delete policy from Ck.
    Args:
        data_set: A list of transactions. Each transaction contains several items.
        Ck: A set which contains all all frequent candidate k-itemsets.
        min_support: The minimum support.
        support_data: A dictionary. The key is frequent itemset and the value is support.
    Returns:
        Lk: A set which contains all all frequent k-itemsets.
    """
    Lk = set()
    item_count = {}
    for t in data_set:
        for item in Ck:
            if item.issubset(t):
                if item not in item_count:
                    item_count[item] = 1
                else:
                    item_count[item] += 1
    t_num = float(len(data_set))
    for item in item_count:
        if (item_count[item] / t_num) >= min_support:
            Lk.add(item)
            support_data[item] = item_count[item] / t_num
    return Lk
```

```python
def generate_L(data_set, k, min_support):
    """
    Generate all frequent itemsets.
    Args:
        data_set: A list of transactions. Each transaction contains several items.
        k: Maximum number of items for all frequent itemsets.
        min_support: The minimum support.
    Returns:
        L: The list of Lk.
        support_data: A dictionary. The key is frequent itemset and the value is support.
    """
    support_data = {}
    C1 = create_C1(data_set)
    L1 = generate_Lk_by_Ck(data_set, C1, min_support, support_data)
    Lksub1 = L1.copy()
    L = []
    L.append(Lksub1)
    for i in range(2, k+1):
        Ci = create_Ck(Lksub1, i)
        Li = generate_Lk_by_Ck(data_set, Ci, min_support, support_data)
        Lksub1 = Li.copy()
        L.append(Lksub1)
    return L, support_data


def generate_big_rules(L, support_data, min_conf):
    """
    Generate big rules from frequent itemsets.
    Args:
        L: The list of Lk.
        support_data: A dictionary. The key is frequent itemset and the value is support.
        min_conf: Minimal confidence.
    Returns:
        big_rule_list: A list which contains all big rules. Each big rule is represented
                       as a 3-tuple.
    """
    big_rule_list = []
    sub_set_list = []
    for i in range(0, len(L)):
        for freq_set in L[i]:
            for sub_set in sub_set_list:
                if sub_set.issubset(freq_set):
                    conf = support_data[freq_set] / support_data[freq_set - sub_set]
                    big_rule = (freq_set - sub_set, sub_set, conf)
```

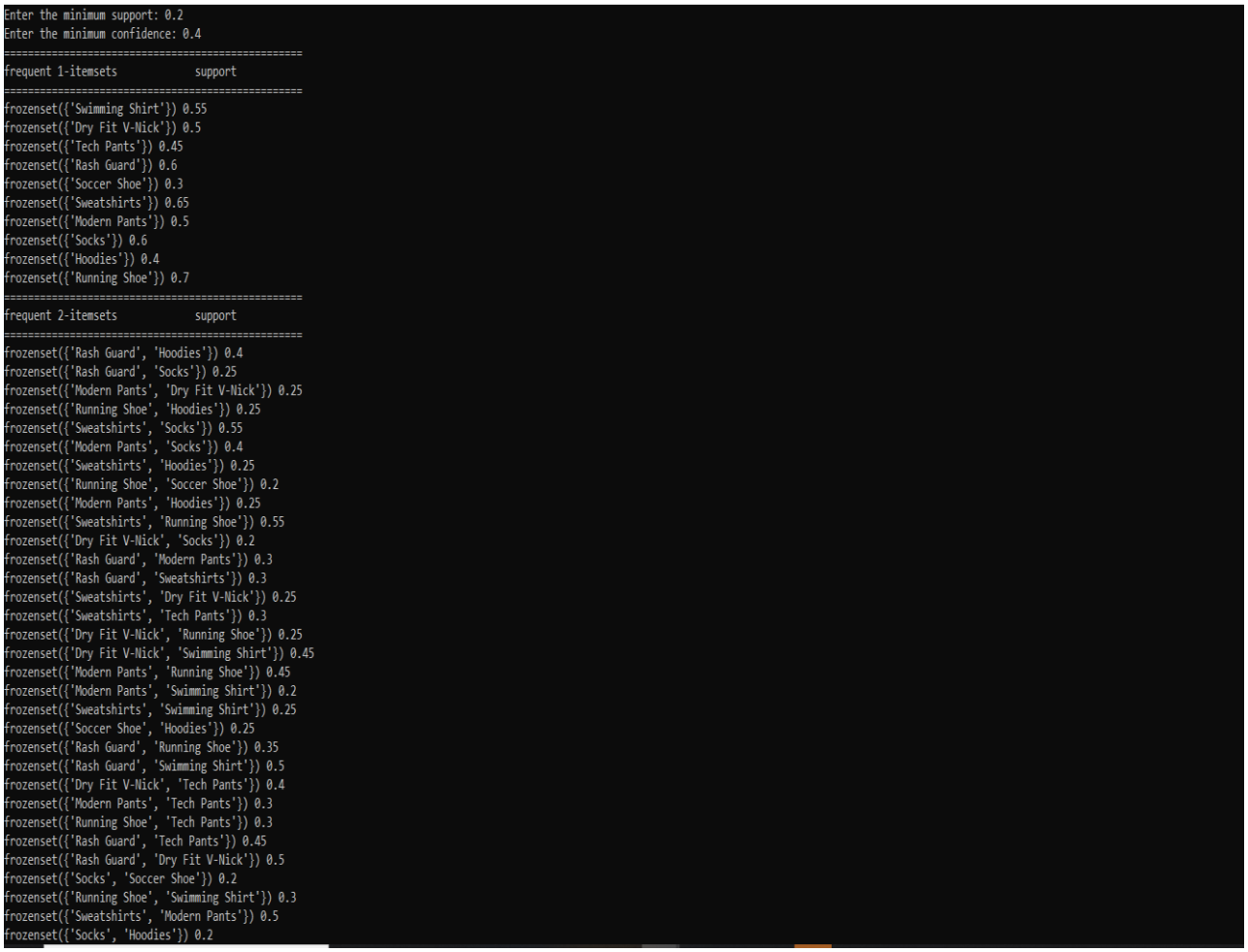
```
                    if conf >= min_conf and big_rule not in big_rule_list:
                        # print freq_set-sub_set, " => ", sub_set, "conf: ", conf
                        big_rule_list.append(big_rule)
            sub_set_list.append(freq_set)
    return big_rule_list


if __name__ == "__main__":
    """
    Test
    """
    data_set = load_data_set()
    L, support_data = generate_L(data_set, k=3, min_support = float(input("Enter the minimum support: ")))
    big_rules_list = generate_big_rules(L, support_data, min_conf= float(input("Enter the minimum confidence: ")))
    for Lk in L:
        print("="*50)
        print("frequent " + str(len(list(Lk)[0])) + "-itemsets\t\tsupport")
        print("="*50)
        for freq_set in Lk:
            print(freq_set, support_data[freq_set])
    print
    print("Big Rules")
    for item in big_rules_list:
        print(item[0], "=>", item[1], "conf: ", item[2])
```

INPUT

```
C:\Users\admin\Desktop\Mid_term_DATA_MINING>python apriori.py
Hello, Please enter which data set you need
 1) Press 1 for Amazon
 2) Press 2 for BestBuy
 3) Press 3 for Nike
 4) Press 4 for KMart
3
User chose Nike dataset
Enter the minimum support: 0.2
Enter the minimum confidence: 0.4
```

OUTPUT



```
Enter the minimum support: 0.2
Enter the minimum confidence: 0.4
================================================
frequent 1-itemsets            support
================================================
frozenset({'Swimming Shirt'}) 0.55
frozenset({'Dry Fit V-Nick'}) 0.5
frozenset({'Tech Pants'}) 0.45
frozenset({'Rash Guard'}) 0.6
frozenset({'Soccer Shoe'}) 0.3
frozenset({'Sweatshirts'}) 0.65
frozenset({'Modern Pants'}) 0.5
frozenset({'Socks'}) 0.6
frozenset({'Hoodies'}) 0.4
frozenset({'Running Shoe'}) 0.7
================================================
frequent 2-itemsets            support
================================================
frozenset({'Rash Guard', 'Hoodies'}) 0.4
frozenset({'Rash Guard', 'Socks'}) 0.25
frozenset({'Modern Pants', 'Dry Fit V-Nick'}) 0.25
frozenset({'Running Shoe', 'Hoodies'}) 0.25
frozenset({'Sweatshirts', 'Socks'}) 0.55
frozenset({'Modern Pants', 'Socks'}) 0.4
frozenset({'Sweatshirts', 'Hoodies'}) 0.25
frozenset({'Running Shoe', 'Soccer Shoe'}) 0.2
frozenset({'Modern Pants', 'Hoodies'}) 0.25
frozenset({'Sweatshirts', 'Running Shoe'}) 0.55
frozenset({'Dry Fit V-Nick', 'Socks'}) 0.2
frozenset({'Rash Guard', 'Modern Pants'}) 0.3
frozenset({'Rash Guard', 'Sweatshirts'}) 0.3
frozenset({'Sweatshirts', 'Dry Fit V-Nick'}) 0.25
frozenset({'Sweatshirts', 'Tech Pants'}) 0.3
frozenset({'Dry Fit V-Nick', 'Running Shoe'}) 0.25
frozenset({'Dry Fit V-Nick', 'Swimming Shirt'}) 0.45
frozenset({'Modern Pants', 'Running Shoe'}) 0.45
frozenset({'Modern Pants', 'Swimming Shirt'}) 0.2
frozenset({'Sweatshirts', 'Swimming Shirt'}) 0.25
frozenset({'Soccer Shoe', 'Hoodies'}) 0.25
frozenset({'Rash Guard', 'Running Shoe'}) 0.35
frozenset({'Rash Guard', 'Swimming Shirt'}) 0.5
frozenset({'Dry Fit V-Nick', 'Tech Pants'}) 0.4
frozenset({'Modern Pants', 'Tech Pants'}) 0.3
frozenset({'Running Shoe', 'Tech Pants'}) 0.3
frozenset({'Rash Guard', 'Tech Pants'}) 0.45
frozenset({'Rash Guard', 'Dry Fit V-Nick'}) 0.5
frozenset({'Socks', 'Soccer Shoe'}) 0.2
frozenset({'Running Shoe', 'Swimming Shirt'}) 0.3
frozenset({'Sweatshirts', 'Modern Pants'}) 0.5
frozenset({'Socks', 'Hoodies'}) 0.2
```

```
================================================
frequent 3-itemsets             support
================================================
frozenset({'Socks', 'Swimming Shirt', 'Tech Pants'}) 0.2
frozenset({'Rash Guard', 'Running Shoe', 'Modern Pants'}) 0.25
frozenset({'Rash Guard', 'Dry Fit V-Nick', 'Hoodies'}) 0.35
frozenset({'Sweatshirts', 'Running Shoe', 'Soccer Shoe'}) 0.2
frozenset({'Modern Pants', 'Soccer Shoe', 'Hoodies'}) 0.2
frozenset({'Rash Guard', 'Soccer Shoe', 'Hoodies'}) 0.25
frozenset({'Sweatshirts', 'Soccer Shoe', 'Tech Pants'}) 0.2
frozenset({'Modern Pants', 'Sweatshirts', 'Tech Pants'}) 0.3
frozenset({'Dry Fit V-Nick', 'Soccer Shoe', 'Tech Pants'}) 0.25
frozenset({'Dry Fit V-Nick', 'Soccer Shoe', 'Hoodies'}) 0.25
frozenset({'Soccer Shoe', 'Rash Guard', 'Modern Pants'}) 0.2
frozenset({'Sweatshirts', 'Soccer Shoe', 'Hoodies'}) 0.2
frozenset({'Modern Pants', 'Sweatshirts', 'Swimming Shirt'}) 0.2
frozenset({'Socks', 'Running Shoe', 'Hoodies'}) 0.2
frozenset({'Swimming Shirt', 'Soccer Shoe', 'Hoodies'}) 0.2
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Soccer Shoe'}) 0.2
frozenset({'Tech Pants', 'Soccer Shoe', 'Hoodies'}) 0.25
frozenset({'Modern Pants', 'Dry Fit V-Nick', 'Rash Guard'}) 0.25
frozenset({'Rash Guard', 'Sweatshirts', 'Soccer Shoe'}) 0.2
frozenset({'Rash Guard', 'Sweatshirts', 'Modern Pants'}) 0.3
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Socks'}) 0.2
frozenset({'Dry Fit V-Nick', 'Running Shoe', 'Tech Pants'}) 0.25
frozenset({'Sweatshirts', 'Socks', 'Hoodies'}) 0.2
frozenset({'Socks', 'Running Shoe', 'Soccer Shoe'}) 0.2
frozenset({'Modern Pants', 'Running Shoe', 'Socks'}) 0.4
frozenset({'Rash Guard', 'Swimming Shirt', 'Tech Pants'}) 0.35
frozenset({'Rash Guard', 'Modern Pants', 'Hoodies'}) 0.25
frozenset({'Modern Pants', 'Socks', 'Rash Guard'}) 0.25
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Rash Guard'}) 0.25
frozenset({'Modern Pants', 'Dry Fit V-Nick', 'Swimming Shirt'}) 0.2
frozenset({'Modern Pants', 'Running Shoe', 'Tech Pants'}) 0.25
frozenset({'Modern Pants', 'Dry Fit V-Nick', 'Socks'}) 0.2
frozenset({'Sweatshirts', 'Tech Pants', 'Hoodies'}) 0.25
frozenset({'Running Shoe', 'Dry Fit V-Nick', 'Socks'}) 0.2
frozenset({'Dry Fit V-Nick', 'Running Shoe', 'Hoodies'}) 0.2
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Tech Pants'}) 0.25
frozenset({'Rash Guard', 'Sweatshirts', 'Swimming Shirt'}) 0.2
frozenset({'Sweatshirts', 'Socks', 'Running Shoe'}) 0.5
frozenset({'Modern Pants', 'Socks', 'Swimming Shirt'}) 0.2
frozenset({'Rash Guard', 'Running Shoe', 'Hoodies'}) 0.25
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Swimming Shirt'}) 0.2
frozenset({'Modern Pants', 'Dry Fit V-Nick', 'Running Shoe'}) 0.2
frozenset({'Modern Pants', 'Tech Pants', 'Hoodies'}) 0.25
frozenset({'Rash Guard', 'Dry Fit V-Nick', 'Soccer Shoe'}) 0.25
frozenset({'Soccer Shoe', 'Swimming Shirt', 'Tech Pants'}) 0.2
frozenset({'Tech Pants', 'Running Shoe', 'Hoodies'}) 0.25
frozenset({'Socks', 'Running Shoe', 'Tech Pants'}) 0.25
```

```
frozenset({'Sweatshirts', 'Dry Fit V-Nick', 'Running Shoe'}) 0.2
frozenset({'Sweatshirts', 'Socks', 'Tech Pants'}) 0.25
Big Rules
frozenset({'Hoodies'}) => frozenset({'Rash Guard'}) conf:  1.0
frozenset({'Rash Guard'}) => frozenset({'Hoodies'}) conf:  0.6666666666666667
frozenset({'Socks'}) => frozenset({'Rash Guard'}) conf:  0.4166666666666667
frozenset({'Rash Guard'}) => frozenset({'Socks'}) conf:  0.4166666666666667
frozenset({'Modern Pants'}) => frozenset({'Dry Fit V-Nick'}) conf:  0.5
frozenset({'Dry Fit V-Nick'}) => frozenset({'Modern Pants'}) conf:  0.5
frozenset({'Hoodies'}) => frozenset({'Running Shoe'}) conf:  0.625
frozenset({'Socks'}) => frozenset({'Sweatshirts'}) conf:  0.9166666666666667
frozenset({'Sweatshirts'}) => frozenset({'Socks'}) conf:  0.8461538461538461
frozenset({'Socks'}) => frozenset({'Modern Pants'}) conf:  0.6666666666666667
frozenset({'Modern Pants'}) => frozenset({'Socks'}) conf:  0.8
frozenset({'Hoodies'}) => frozenset({'Sweatshirts'}) conf:  0.625
frozenset({'Soccer Shoe'}) => frozenset({'Running Shoe'}) conf:  0.6666666666666667
frozenset({'Hoodies'}) => frozenset({'Modern Pants'}) conf:  0.625
frozenset({'Modern Pants'}) => frozenset({'Hoodies'}) conf:  0.5
frozenset({'Running Shoe'}) => frozenset({'Sweatshirts'}) conf:  0.7857142857142858
frozenset({'Sweatshirts'}) => frozenset({'Running Shoe'}) conf:  0.8461538461538461
frozenset({'Dry Fit V-Nick'}) => frozenset({'Socks'}) conf:  0.4
frozenset({'Modern Pants'}) => frozenset({'Rash Guard'}) conf:  0.6
frozenset({'Rash Guard'}) => frozenset({'Modern Pants'}) conf:  0.5
frozenset({'Sweatshirts'}) => frozenset({'Rash Guard'}) conf:  0.4615384615384615
frozenset({'Rash Guard'}) => frozenset({'Sweatshirts'}) conf:  0.5
frozenset({'Dry Fit V-Nick'}) => frozenset({'Sweatshirts'}) conf:  0.5
frozenset({'Sweatshirts'}) => frozenset({'Tech Pants'}) conf:  0.4615384615384615
frozenset({'Tech Pants'}) => frozenset({'Sweatshirts'}) conf:  0.6666666666666666
frozenset({'Dry Fit V-Nick'}) => frozenset({'Running Shoe'}) conf:  0.5
frozenset({'Dry Fit V-Nick'}) => frozenset({'Swimming Shirt'}) conf:  0.9
frozenset({'Swimming Shirt'}) => frozenset({'Dry Fit V-Nick'}) conf:  0.8181818181818181
frozenset({'Running Shoe'}) => frozenset({'Modern Pants'}) conf:  0.6428571428571429
frozenset({'Modern Pants'}) => frozenset({'Running Shoe'}) conf:  0.9
frozenset({'Modern Pants'}) => frozenset({'Swimming Shirt'}) conf:  0.4
frozenset({'Swimming Shirt'}) => frozenset({'Sweatshirts'}) conf:  0.45454545454545453
frozenset({'Hoodies'}) => frozenset({'Soccer Shoe'}) conf:  0.625
frozenset({'Soccer Shoe'}) => frozenset({'Hoodies'}) conf:  0.8333333333333334
frozenset({'Running Shoe'}) => frozenset({'Rash Guard'}) conf:  0.5
frozenset({'Rash Guard'}) => frozenset({'Running Shoe'}) conf:  0.5833333333333334
frozenset({'Rash Guard'}) => frozenset({'Swimming Shirt'}) conf:  0.8333333333333334
frozenset({'Swimming Shirt'}) => frozenset({'Rash Guard'}) conf:  0.9090909090909091
frozenset({'Tech Pants'}) => frozenset({'Dry Fit V-Nick'}) conf:  0.8888888888888889
frozenset({'Dry Fit V-Nick'}) => frozenset({'Tech Pants'}) conf:  0.8
frozenset({'Modern Pants'}) => frozenset({'Tech Pants'}) conf:  0.6
frozenset({'Tech Pants'}) => frozenset({'Modern Pants'}) conf:  0.6666666666666666
frozenset({'Running Shoe'}) => frozenset({'Tech Pants'}) conf:  0.4285714285714286
frozenset({'Tech Pants'}) => frozenset({'Running Shoe'}) conf:  0.6666666666666666
frozenset({'Rash Guard'}) => frozenset({'Tech Pants'}) conf:  0.75
frozenset({'Tech Pants'}) => frozenset({'Rash Guard'}) conf:  1.0
frozenset({'Rash Guard'}) => frozenset({'Dry Fit V-Nick'}) conf:  0.8333333333333334
```