

# **DATA MINING**

## **Midterm Project**

**Implementing Apriori Algorithm for  
Transactions Dataset**

**By – JAY ASHWINKUMAR SORATHIYA**  
**UCID – js2575**  
**E-mail: [js2575@njit.edu](mailto:js2575@njit.edu)**

## **TABLE OF CONTENTS**

Sr. No.	Headings	Page No.
1.	Source Code	3 – 5
2.	Test Run for Grocery Dataset	6 – 9
3.	Test Run for Dollar Store Dataset	10 – 12
4.	Test Run for Liquor store Dataset	12 – 15
5.	Test Run for Electronics Dataset	16 – 18
6.	Test Run for Fashion Dataset	19 – 21

### **Note:**

- ➔ In this project report, I have implemented the Apriori algorithm to generate Frequent itemsets and their Association rules from the 5 Transaction Databases.
- ➔ This Project report does not include Implementation of the Brute Force method and comparing it with the Apriori algorithm on each of the 5 transactions Databases.
- ➔ It also does not include the Presentation of computation (CPU or Clock) time to demonstrate that the Apriori algorithm is faster than the brute force method on each of the 5 transaction databases.

# Source Code:

**Programming Language used:** Python 3

**How to run the program?**

- ➔ python <apriori\_sc\_js.py> <dataset\_file> <Minimum Support> <Minimum Confidence>
- ➔ Example: python apriori\_sc\_js.py ds\_grocery.txt 20 30

```
import sys
from itertools import permutations
from itertools import combinations
from typing import Dict

#How to run
#python <apriori_sc_js.py> <dataset_file> <Minimum Support> <Minimum Confidence>
#Example python apriori_sc_js.py ds_grocery.txt 20 30

def print_items(freq_items,n):          #Function with 2 args. arg1 is a set and arg2 is a number
    print("oooooooooooooooooooooooooooooooooooooooo")
    print("Selected itemsets after ",n," iteration")
    print("__Itemset__", "__Support__")
    print("oooooooooooooooooooooooooooooooooooooooo")
    print()
    for i in freq_items:
        print(i,round(freq_items[i]*100/total_txns,2),)  #This is the formula to calculate the support/confidence
    print()

def find_freq_items(nlist, few_items, set_items, n):    #This is a function with 4 args. 1-2-3 args are lists, arg 4 is a
number.
    comb = combinations(nlist, n)                      #Getting combinations through arg1 and arg4
    item_support = {}                                  #Dictionary of count defined here
    for i in comb:
        set_i = set(i)                                #Making a set of unique items
        i=tuple(sorted(i))
        for j in set_items:                            #defined in main program   #'for' loop for arg3
            if set_i.issubset(j):
                if few_items:
                    count = 0
                    for k in few_items:                #'for' loop for arg2
                        if k.issubset(set_i):
                            count = 1
                            break
                    if not count:                        #item_support_count was defined in this fn itself
                        if i in item_support:
                            item_support[i] += 1
                        else:
                            item_support[i] = 1
        else:
            if i in item_support:
                item_support[i] += 1
            else:
```

```

        item_support[i] = 1
freq_set_dict = {}                #Defining this dict for loops in this fn itself
few_item_list = []                #Defining this list to use it in this fn itself
#print(item_support)
if item_support:
    print("oooooooooooooooooooooooooooo")
    print("Itemsets for ", n, "Iteration")
    print("oooooooooooooooooooooooooooo")
    print()
    for i in item_support:          #'for' loop for calculating support
        print(i,round(item_support[i]*100/total_txns,2))
        if (item_support[i]/total_txns)*100 >= min_supp:  #'if' condition for minimum support
            freq_set_dict[i] = item_support[i]
        else:
            few_item_list.append(set(list(i)))            #'else' store it in few list set
    print()
    if freq_set_dict:                #Using that dict here again
        print_items(freq_set_dict,n)
        support_all_items.update(item_support)          #Updating the dict with item support value
        association_rules(freq_set_dict)
        return freq_set_dict, few_item_list
    return None,None

def association_rules(freq_items):    #arg of this fn is a dictionary
    for item_pair in freq_items.keys():
        print("Association Rule for Pair - ",item_pair)
        print("__Rule__","__Confidence__")
        item_size=len(item_pair)
        item_set=set(item_pair)
        while item_size-1>0:
            comb = combinations(item_pair, item_size-1)
            for i in comb:
                left_item=i
                right_item=tuple(item_set-set(i))
                item_confidence=round(support_all_items[item_pair]*100/support_all_items[left_item],2)
                if item_confidence>=min_conf:
                    print(left_item,"=>",right_item,item_confidence,"This Rule is Acceptable")
                else:
                    print(left_item,"=>",right_item,item_confidence,"This Rule is Rejected")

            item_size -=1
        print()

### Program Start from here ###
file_name = sys.argv[1]
file_object = open(file_name, "r")
lines = file_object.readlines()
all_txns = []
total_txns=0
support_all_items={}
min_supp = int(sys.argv[2])        #Minimum support argument
min_conf = int(sys.argv[3])        #Minimum support argument
c1 = {} # type: Dict[str, int]
set_items = []                     #List created for input in 2nd function
print("oooooooooooooooooooo")
print("Input Transactions")

```

```

print("oooooooooooooooooooo")
print()

for txns in lines:
    txns = txns.replace("\n", "")
    print(txns)
    all_txns.append("".join(txns.split(" ")[1].split(",")))
    seen = set() #Declared a set here
    for i in "".join(txns.split(" ")[1:]).split(","):

        if (i,) in c1:
            c1[(i,)] += 1
        else:
            # print((i,),txns)
            c1[(i,)] = 1
            seen.add(i)
    set_items.append(seen) #Adding it to the Item set list
    total_txns+=1 #Total no. of txns
    # print(seen)
freq_items = {}
# print(c1)
few_set = []
print()
print("oooooooooooooooooooo")
print("Item Sets", 1, "Iteration")
print("oooooooooooooooooooo")
print()
for i in c1:
    print(i,round(c1[i]*100/total_txns))
    if (c1[i]/20)*100 >= min_supp:
        freq_items[i] = c1[i]
    else:
        few_set.append(set(i))
support_all_items.update(c1)
nlist=[item[0] for item in freq_items.keys()] #This list is created for the loop below (using List Comprehension)

print()
print_items(freq_items,1)
item_count = 1
while len(nlist) > item_count: #This while loop will help us generate frequent item pairs
    freq_items1, few_set1 = find_freq_items(nlist, few_set, set_items, item_count + 1)
    if not freq_items1:
        break
    item_list = [items for item_groups in list(freq_items1.keys()) for items in item_groups]
    nlist = list(set(item_list))
    few_set = few_set1
    freq_items=freq_items1
    item_count += 1

```

# Test Run of Grocery Dataset

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\jayso\Desktop\DM Project>python apriori_sc_js.py ds_grocery.txt 20 30
oooooooooooooooooooo
Input Transactions
oooooooooooooooooooo

1 Bread, Butter, Cocoa, Milk
2 Tea, Coffee, Milk, Banana, Apple
3 Apple, Orange, Banana, Milk, Coffee
4 Juice, ProteinPowder, Milk
5 Soda, Bread, Butter, Chicken
6 Bread, Butter, Oats
7 Banana, Milk, Coffee, Oats
8 Orange, Banana, Milk
9 Bread, ProteinPowder, Butter, Cocoa, Milk, Tea, Coffee, Juice
10 Milk, Juice, ProteinPowder, Orange
11 Tea, Coffee, juice, Banana, Oats
12 Apple, Milk, Chicken
13 Butter, Milk, ProteinPowder
14 Cocoa, Juice, Milk, Tea
15 Banana, ProtrinPowder, Oats, Honey, Chicken
16 Honey, Chips, Milk, Cocoa, Butter
17 Soda, Coffee, Bread, Butter
18 Orange, Oats, Banana, Juice, Chicken
19 Butter, Chicken, Banana, Oats, ProteinPowder
20 Soda, Chips, Butter

oooooooooooooooooooo
Item Sets 1 Iteration
oooooooooooooooooooo

('Bread',) 25
('Butter',) 45
('Cocoa',) 20
('Milk',) 60
('Tea',) 20
('Coffee',) 30
('Banana',) 40
('Apple',) 15
('Orange',) 20
('Juice',) 25
('ProteinPowder',) 25
('Soda',) 15
('Chicken',) 25
('Oats',) 30
('juice',) 5
('ProtrinPowder',) 5
('Honey',) 10
('Chips',) 10
```

```
( 'Honey', ) 10
( 'Chips', ) 10

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 1 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

( 'Bread', ) 25.0
( 'Butter', ) 45.0
( 'Cocoa', ) 20.0
( 'Milk', ) 60.0
( 'Tea', ) 20.0
( 'Coffee', ) 30.0
( 'Banana', ) 40.0
( 'Orange', ) 20.0
( 'Juice', ) 25.0
( 'ProteinPowder', ) 25.0
( 'Chicken', ) 25.0
( 'Oats', ) 30.0

oooooooooooooooooooooooooooooooooooooooo
Itemsets for 2 Iteration
oooooooooooooooooooooooooooooooooooooooo

( 'Bread', 'Butter' ) 25.0
( 'Bread', 'Cocoa' ) 10.0
( 'Bread', 'Milk' ) 10.0
( 'Bread', 'Tea' ) 5.0
( 'Bread', 'Coffee' ) 10.0
( 'Bread', 'Juice' ) 5.0
( 'Bread', 'ProteinPowder' ) 5.0
( 'Bread', 'Chicken' ) 5.0
( 'Bread', 'Oats' ) 5.0
( 'Butter', 'Cocoa' ) 15.0
( 'Butter', 'Milk' ) 20.0
( 'Butter', 'Tea' ) 5.0
( 'Butter', 'Coffee' ) 10.0
( 'Banana', 'Butter' ) 5.0
( 'Butter', 'Juice' ) 5.0
( 'Butter', 'ProteinPowder' ) 15.0
( 'Butter', 'Chicken' ) 10.0
( 'Butter', 'Oats' ) 10.0
( 'Cocoa', 'Milk' ) 20.0
( 'Cocoa', 'Tea' ) 10.0
( 'Cocoa', 'Coffee' ) 5.0
( 'Cocoa', 'Juice' ) 10.0
( 'Cocoa', 'ProteinPowder' ) 5.0
( 'Milk', 'Tea' ) 15.0
( 'Coffee', 'Milk' ) 20.0
```

```
( 'Cocoa', 'ProteinPowder') 5.0
( 'Milk', 'Tea') 15.0
( 'Coffee', 'Milk') 20.0
( 'Banana', 'Milk') 20.0
( 'Milk', 'Orange') 15.0
( 'Juice', 'Milk') 20.0
( 'Milk', 'ProteinPowder') 20.0
( 'Chicken', 'Milk') 5.0
( 'Milk', 'Oats') 5.0
( 'Coffee', 'Tea') 15.0
( 'Banana', 'Tea') 10.0
( 'Juice', 'Tea') 10.0
( 'ProteinPowder', 'Tea') 5.0
( 'Oats', 'Tea') 5.0
( 'Banana', 'Coffee') 20.0
( 'Coffee', 'Orange') 5.0
( 'Coffee', 'Juice') 5.0
( 'Coffee', 'ProteinPowder') 5.0
( 'Coffee', 'Oats') 10.0
( 'Banana', 'Orange') 15.0
( 'Banana', 'Juice') 5.0
( 'Banana', 'ProteinPowder') 5.0
( 'Banana', 'Chicken') 15.0
( 'Banana', 'Oats') 25.0
( 'Juice', 'Orange') 10.0
( 'Orange', 'ProteinPowder') 5.0
( 'Chicken', 'Orange') 5.0
( 'Oats', 'Orange') 5.0
( 'Juice', 'ProteinPowder') 15.0
( 'Chicken', 'Juice') 5.0
( 'Juice', 'Oats') 5.0
( 'Chicken', 'ProteinPowder') 5.0
( 'Oats', 'ProteinPowder') 5.0
( 'Chicken', 'Oats') 15.0
```

oo

Selected itemsets after 2 iteration

\_\_Itemset\_\_ \_\_Support\_\_

oo

```
( 'Bread', 'Butter') 25.0
( 'Butter', 'Milk') 20.0
( 'Cocoa', 'Milk') 20.0
( 'Coffee', 'Milk') 20.0
( 'Banana', 'Milk') 20.0
( 'Juice', 'Milk') 20.0
( 'Milk', 'ProteinPowder') 20.0
( 'Banana', 'Coffee') 20.0
( 'Banana', 'Oats') 25.0
```



Anaconda Prompt (Anaconda3)

```
('Milk', 'ProteinPowder') 20.0
('Banana', 'Coffee') 20.0
('Banana', 'Oats') 25.0

Association Rule for Pair - ('Bread', 'Butter')
Rule__Confidence__
('Bread',) => ('Butter',) 100.0 This Rule is Acceptable
('Butter',) => ('Bread',) 55.56 This Rule is Acceptable

Association Rule for Pair - ('Butter', 'Milk')
Rule__Confidence__
('Butter',) => ('Milk',) 44.44 This Rule is Acceptable
('Milk',) => ('Butter',) 33.33 This Rule is Acceptable

Association Rule for Pair - ('Cocoa', 'Milk')
Rule__Confidence__
('Cocoa',) => ('Milk',) 100.0 This Rule is Acceptable
('Milk',) => ('Cocoa',) 33.33 This Rule is Acceptable

Association Rule for Pair - ('Coffee', 'Milk')
Rule__Confidence__
('Coffee',) => ('Milk',) 66.67 This Rule is Acceptable
('Milk',) => ('Coffee',) 33.33 This Rule is Acceptable

Association Rule for Pair - ('Banana', 'Milk')
Rule__Confidence__
('Banana',) => ('Milk',) 50.0 This Rule is Acceptable
('Milk',) => ('Banana',) 33.33 This Rule is Acceptable

Association Rule for Pair - ('Juice', 'Milk')
Rule__Confidence__
('Juice',) => ('Milk',) 80.0 This Rule is Acceptable
('Milk',) => ('Juice',) 33.33 This Rule is Acceptable

Association Rule for Pair - ('Milk', 'ProteinPowder')
Rule__Confidence__
('Milk',) => ('ProteinPowder',) 33.33 This Rule is Acceptable
('ProteinPowder',) => ('Milk',) 80.0 This Rule is Acceptable

Association Rule for Pair - ('Banana', 'Coffee')
Rule__Confidence__
('Banana',) => ('Coffee',) 50.0 This Rule is Acceptable
('Coffee',) => ('Banana',) 66.67 This Rule is Acceptable

Association Rule for Pair - ('Banana', 'Oats')
Rule__Confidence__
('Banana',) => ('Oats',) 62.5 This Rule is Acceptable
('Oats',) => ('Banana',) 83.33 This Rule is Acceptable

oooooooooooooooooooooooooooooooooooo
```

Anaconda Prompt (Anaconda3)

```
Association Rule for Pair - ('Banana', 'Oats')
Rule__Confidence__
('Banana',) => ('Oats',) 62.5 This Rule is Acceptable
('Oats',) => ('Banana',) 83.33 This Rule is Acceptable

oooooooooooooooooooooooooooooooooooo
Itemsets for 3 Iteration
oooooooooooooooooooooooooooooooooooo

('Banana', 'Coffee', 'Milk') 15.0

(base) C:\Users\jayso\Desktop\DM Project>
```

# Test Run of Dollar Store Dataset

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\jayso\Desktop\DM Project>python apriori_sc_js.py ds_dollarstore.txt 20 40
```

```
oooooooooooooooooooo
```

```
Input Transactions
```

```
oooooooooooooooooooo
```

```
1 Chips, Deodrant, Spoons, Glares, Napkins, Plates
2 Bowl, Spoons, Soda, Foils, Jar
3 Plates, Soda, Glass, PaperDish
4 Lamp, Glass, Jar, Deodrant
5 Knife, Jar, Curtains
6 Bag, Curtains, Bowl, Chips, Napkins
7 GreetingCard, Sanitizer, Plates, Deodrant
8 Glares, Baskets, Lamp
9 Foils, Trash_bags, Sanitizer
10 PaperDish, Napkins, Baskets, Curtains
11 Deodrant, Chips, Trash_bags, Sanitizer
12 Spoons, Bowl, Napkins
13 Soda, Plates, Glares, Bag
14 Glass, Lamp, Foils, Baskets, Chips
15 Jar, Knife, PaperDish
16 Curtains, Bag, Deodrant
17 Sanitizer, GreetingCard, Chips, Knife
18 Baskets, Glares, Bowl, Bag
19 Trash_bags, Foils, Plates, GreetingCard
20 Napkins, PaperDish, Lamp, Glares
```

```
oooooooooooooooooooo
```

```
Item Sets 1 Iteration
```

```
oooooooooooooooooooo
```

```
('Chips',) 25
('Deodrant',) 25
('Spoons',) 15
('Glares',) 25
('Napkins',) 25
('Plates',) 25
('Bowl',) 20
('Soda',) 15
('Foils',) 20
('Jar',) 20
('Glass',) 10
('PaperDish',) 20
('Lamp',) 20
('\tGlass',) 5
('Knife',) 15
('Curtains',) 20
('Bag',) 20
('GreetingCard',) 15
('Sanitizer',) 20
('Baskets',) 20
```

Anaconda Prompt (Anaconda3)

```
('Baskets',) 20
('Trash_bags',) 15

oooooooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 1 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooooooo

('Chips',) 25.0
('Deodrant',) 25.0
('Glares',) 25.0
('Napkins',) 25.0
('Plates',) 25.0
('Bowl',) 20.0
('Foils',) 20.0
('Jar',) 20.0
('PaperDish',) 20.0
('Lamp',) 20.0
('Curtains',) 20.0
('Bag',) 20.0
('Sanitizer',) 20.0
('Baskets',) 20.0

oooooooooooooooooooooooooooooooooooooooooooo
Itemsets for 2 Iteration
oooooooooooooooooooooooooooooooooooooooooooo

('Chips', 'Deodrant') 10.0
('Chips', 'Glares') 5.0
('Chips', 'Napkins') 10.0
('Chips', 'Plates') 5.0
('Bowl', 'Chips') 5.0
('Chips', 'Foils') 5.0
('Chips', 'Lamp') 5.0
('Chips', 'Curtains') 5.0
('Bag', 'Chips') 5.0
('Chips', 'Sanitizer') 10.0
('Baskets', 'Chips') 5.0
('Deodrant', 'Glares') 5.0
('Deodrant', 'Napkins') 5.0
('Deodrant', 'Plates') 10.0
('Deodrant', 'Jar') 5.0
('Deodrant', 'Lamp') 5.0
('Curtains', 'Deodrant') 5.0
('Bag', 'Deodrant') 5.0
('Deodrant', 'Sanitizer') 10.0
('Glares', 'Napkins') 10.0
('Glares', 'Plates') 10.0
('Bowl', 'Glares') 5.0
('Glares', 'PaperDish') 5.0
```

Anaconda Prompt (Anaconda3)

```
('Glares', 'Napkins') 10.0
('Glares', 'Plates') 10.0
('Bowl', 'Glares') 5.0
('Glares', 'PaperDish') 5.0
('Glares', 'Lamp') 10.0
('Bag', 'Glares') 10.0
('Baskets', 'Glares') 10.0
('Napkins', 'Plates') 5.0
('Bowl', 'Napkins') 10.0
('Napkins', 'PaperDish') 10.0
('Lamp', 'Napkins') 5.0
('Curtains', 'Napkins') 10.0
('Bag', 'Napkins') 5.0
('Baskets', 'Napkins') 5.0
('Foils', 'Plates') 5.0
('PaperDish', 'Plates') 5.0
('Bag', 'Plates') 5.0
('Plates', 'Sanitizer') 5.0
('Bowl', 'Foils') 5.0
('Bowl', 'Jar') 5.0
('Bowl', 'Curtains') 5.0
('Bag', 'Bowl') 10.0
('Baskets', 'Bowl') 5.0
('Foils', 'Jar') 5.0
('Foils', 'Lamp') 5.0
('Foils', 'Sanitizer') 5.0
('Baskets', 'Foils') 5.0
('Jar', 'PaperDish') 5.0
('Jar', 'Lamp') 5.0
('Curtains', 'Jar') 5.0
('Lamp', 'PaperDish') 5.0
('Curtains', 'PaperDish') 5.0
('Baskets', 'PaperDish') 5.0
('Baskets', 'Lamp') 10.0
('Bag', 'Curtains') 10.0
('Baskets', 'Curtains') 5.0
('Bag', 'Baskets') 5.0
```

(base) C:\Users\jayso\Desktop\DM Project>

# Test Run of Liquor Store Dataset

■ Select Anaconda Prompt (Anaconda3)

```
(base) C:\Users\jayso\Desktop\DM Project>python apriori_sc_js.py ds_liquor.txt 20 20
```

```
oooooooooooooooooooo
```

```
Input Transactions
```

```
oooooooooooooooooooo
```

```
1 Cabernet, Beer, Tequila, Popcorns, Cigar
2 Merlot, Soda, Beer, OrabgeJuice
3 Sauvignon, Rum, Chips, Stout, Peanuts
4 Rose, Lime, Chips
5 Red, Cognac, Redbull
6 OrangeJuice, Tequila, IPA, Sangria, Chocolates, Cigar
7 Bourbon, Soda, Redbull
8 Scotch, Soda, Peanuts, Lime
9 Gin, Lime, Peanuts
10 Vodka, OrangeJuice, Popcorns
11 Lime, Vodka, OrangeJuice, Sauvignon, Cigar
12 Chips, OrangeJuice, Cognac, IPA
13 Chocolates, OrangeJuice, Beer
14 LagerBeer, Soda, Scotch, Cigar
15 Stout, Chips
16 Water, Tequila, OrangeJuice, Cigar
17 IPA, Chips, Peanuts
18 Beer, Merlot, Cigar
19 Cognac, OrangeJuice, Sauvignon, Redbull, Peanuts, Cigar
20 Sangria, Sauvignon, Redbull
```

```
oooooooooooooooooooo
```

```
Item Sets 1 Iteration
```

```
oooooooooooooooooooo
```

```
('Cabernet',) 5
('Beer',) 15
('Tequila',) 15
('Popcorns',) 10
('Cigar',) 35
('Merlot',) 10
('Soda',) 20
('Beer',) 5
('OrabgeJuice',) 5
('Sauvignon',) 20
('Rum',) 5
('Chips',) 25
('Stout',) 10
('Peanuts',) 5
('Rose',) 5
('Lime',) 20
('Red',) 5
('Cognac',) 15
('Redbull',) 20
('OrangeJuice',) 35
```

```

('Cognac',) 15
('Redbull',) 20
('OrangeJuice',) 35
('IPA',) 15
('Sangria',) 10
('Chocolates',) 10
('Bourbon',) 5
('Scotch',) 10
('Peanuts',) 20
('Gin',) 5
('Vodka',) 10
('LagerBeer',) 5
('Water',) 5

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 1 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

('Cigar',) 35.0
('Soda',) 20.0
('Sauvignon',) 20.0
('Chips',) 25.0
('Lime',) 20.0
('Redbull',) 20.0
('OrangeJuice',) 35.0
('Peanuts',) 20.0

oooooooooooooooooooooooooooooooooooooooo
Itemsets for 2 Iteration
oooooooooooooooooooooooooooooooooooooooo

('Cigar', 'Soda') 5.0
('Cigar', 'Sauvignon') 10.0
('Cigar', 'Lime') 5.0
('Cigar', 'Redbull') 5.0
('Cigar', 'OrangeJuice') 20.0
('Cigar', 'Peanuts') 5.0
('Lime', 'Soda') 5.0
('Redbull', 'Soda') 5.0
('Peanuts', 'Soda') 5.0
('Chips', 'Sauvignon') 5.0
('Lime', 'Sauvignon') 5.0
('Redbull', 'Sauvignon') 10.0
('OrangeJuice', 'Sauvignon') 10.0
('Peanuts', 'Sauvignon') 5.0
('Chips', 'Lime') 5.0
('Chips', 'OrangeJuice') 5.0
('Chips', 'Peanuts') 5.0
('Lime', 'OrangeJuice') 5.0

```

Select Anaconda Prompt (Anaconda3)

oooooooooooooooooooooooooooooooooooo

Itemsets for 2 Iteration

oooooooooooooooooooooooooooooooooooo

```
('Cigar', 'Soda') 5.0
('Cigar', 'Sauvignon') 10.0
('Cigar', 'Lime') 5.0
('Cigar', 'Redbull') 5.0
('Cigar', 'OrangeJuice') 20.0
('Cigar', 'Peanuts') 5.0
('Lime', 'Soda') 5.0
('Redbull', 'Soda') 5.0
('Peanuts', 'Soda') 5.0
('Chips', 'Sauvignon') 5.0
('Lime', 'Sauvignon') 5.0
('Redbull', 'Sauvignon') 10.0
('OrangeJuice', 'Sauvignon') 10.0
('Peanuts', 'Sauvignon') 5.0
('Chips', 'Lime') 5.0
('Chips', 'OrangeJuice') 5.0
('Chips', 'Peanuts') 5.0
('Lime', 'OrangeJuice') 5.0
('Lime', 'Peanuts') 10.0
('OrangeJuice', 'Redbull') 5.0
('Peanuts', 'Redbull') 5.0
('OrangeJuice', 'Peanuts') 5.0
```

oooooooooooooooooooooooooooooooooooo

Selected itemsets after 2 iteration

\_\_Itemset\_\_ \_\_Support\_\_

oooooooooooooooooooooooooooooooooooo

```
('Cigar', 'OrangeJuice') 20.0
```

Association Rule for Pair - ('Cigar', 'OrangeJuice')

\_\_Rule\_\_ \_\_Confidence\_\_

```
('Cigar',) => ('OrangeJuice',) 57.14 This Rule is Acceptable
```

```
('OrangeJuice',) => ('Cigar',) 57.14 This Rule is Acceptable
```

(base) C:\Users\jayso\Desktop\DM Project>

# Test Run of Electronics Dataset

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\jayso\Desktop\DM Project>python apriori_sc_js.py ds_electronics.txt 20 20
oooooooooooooooooooo
Input Transactions
oooooooooooooooooooo

1 DataCable, iWatch, iPhone, AirPods, Fitness+
2 iWatch, AirPods, Fitness+, LAFitness
3 AirPods, Mobile, ExtensionChord
4 iWatch, iPhone
5 ExtensionChord, AirPods
6 Fitness+, iPhone, ExtensionChord, DataCable, iWatch
7 Bluetooth, AirPods, iWatch
8 Fitness+, Bluetooth, AirPods, iWatch
9 AirPods, DataCable, LAFitness
10 Fitness+, ExtensionChord, Mobile, Bluetooth
11 LAFitness, iWatch
12 Fitness+, Bluetooth, iWatch, Mobile
13 ExtensionChord, AirPods
14 DataCable, Mobile, Bluetooth
15 LAFitness, AirPods, ExtensionChord
16 iWatch, Fitness+
17 Bluetooth, AirPods
18 ExtensionChord, Bluetooth, AirPods
19 LAFitness
20 iWatch, AirPods, ExtensionChord, Fitness+

oooooooooooooooooooo
Item Sets 1 Iteration
oooooooooooooooooooo

('DataCable',) 20
('iWatch',) 50
('iPhone',) 15
('AirPods',) 60
('Fitness+',) 40
('LAFitness',) 25
('Mobile',) 20
('ExtensionChord',) 40
('Bluetooth',) 35

oooooooooooooooooooo
Selected itemsets after 1 iteration
__Itemset__ __Support__
oooooooooooooooooooo

('DataCable',) 20.0
('iWatch',) 50.0
('AirPods',) 60.0
('Fitness+',) 40.0
('LAFitness',) 25.0
```



Selected itemsets after 1 iteration

Itemset	Support
---------	---------

('DataCable',)	20.0
('iWatch',)	50.0
('Airpods',)	60.0
('Fitness+',)	40.0
('LAFitness',)	25.0
('Mobile',)	20.0
('ExtensionChord',)	40.0
('Bluetooth',)	35.0

Itemsets for 2 Iteration

('DataCable', 'iWatch')	10.0
('Airpods', 'DataCable')	10.0
('DataCable', 'Fitness+')	10.0
('DataCable', 'LAFitness')	5.0
('DataCable', 'Mobile')	5.0
('DataCable', 'ExtensionChord')	5.0
('Bluetooth', 'DataCable')	5.0
('Airpods', 'iWatch')	25.0
('Fitness+', 'iWatch')	35.0
('LAFitness', 'iWatch')	10.0
('Mobile', 'iWatch')	5.0
('ExtensionChord', 'iWatch')	10.0
('Bluetooth', 'iWatch')	15.0
('Airpods', 'Fitness+')	20.0
('Airpods', 'LAFitness')	15.0
('Airpods', 'Mobile')	5.0
('Airpods', 'ExtensionChord')	30.0
('Airpods', 'Bluetooth')	20.0
('Fitness+', 'LAFitness')	5.0
('Fitness+', 'Mobile')	10.0
('ExtensionChord', 'Fitness+')	15.0
('Bluetooth', 'Fitness+')	15.0
('ExtensionChord', 'LAFitness')	5.0
('ExtensionChord', 'Mobile')	10.0
('Bluetooth', 'Mobile')	15.0
('Bluetooth', 'ExtensionChord')	10.0

Selected itemsets after 2 iteration

Itemset	Support
---------	---------

--	--

Anaconda Prompt (Anaconda3)

```
('Bluetooth', 'ExtensionChord') 10.0

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 2 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

('Airpods', 'iWatch') 25.0
('Fitness+', 'iWatch') 35.0
('Airpods', 'Fitness+') 20.0
('Airpods', 'ExtensionChord') 30.0
('Airpods', 'Bluetooth') 20.0

Association Rule for Pair - ('Airpods', 'iWatch')
__Rule__ __Confidence__
('Airpods',) => ('iWatch',) 41.67 This Rule is Acceptable
('iWatch',) => ('Airpods',) 50.0 This Rule is Acceptable

Association Rule for Pair - ('Fitness+', 'iWatch')
__Rule__ __Confidence__
('Fitness+',) => ('iWatch',) 87.5 This Rule is Acceptable
('iWatch',) => ('Fitness+',) 70.0 This Rule is Acceptable

Association Rule for Pair - ('Airpods', 'Fitness+')
__Rule__ __Confidence__
('Airpods',) => ('Fitness+',) 33.33 This Rule is Acceptable
('Fitness+',) => ('Airpods',) 50.0 This Rule is Acceptable

Association Rule for Pair - ('Airpods', 'ExtensionChord')
__Rule__ __Confidence__
('Airpods',) => ('ExtensionChord',) 50.0 This Rule is Acceptable
('ExtensionChord',) => ('Airpods',) 75.0 This Rule is Acceptable

Association Rule for Pair - ('Airpods', 'Bluetooth')
__Rule__ __Confidence__
('Airpods',) => ('Bluetooth',) 33.33 This Rule is Acceptable
('Bluetooth',) => ('Airpods',) 57.14 This Rule is Acceptable

oooooooooooooooooooooooooooooooooooooooo
Itemsets for 3 Iteration
oooooooooooooooooooooooooooooooooooooooo

('Airpods', 'Fitness+', 'iWatch') 20.0

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 3 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

('Airpods', 'Fitness+', 'iWatch') 20.0
```

```
('Airpods', 'Fitness+', 'iWatch') 20.0

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 3 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

('Airpods', 'Fitness+', 'iWatch') 20.0

Association Rule for Pair - ('Airpods', 'Fitness+', 'iWatch')
__Rule__ __Confidence__
('Airpods', 'Fitness+') => ('iWatch',) 100.0 This Rule is Acceptable
('Airpods', 'iWatch') => ('Fitness+',) 80.0 This Rule is Acceptable
('Fitness+', 'iWatch') => ('Airpods',) 57.14 This Rule is Acceptable
('Airpods',) => ('Fitness+', 'iWatch') 33.33 This Rule is Acceptable
('Fitness+',) => ('Airpods', 'iWatch') 50.0 This Rule is Acceptable
('iWatch',) => ('Fitness+', 'Airpods') 40.0 This Rule is Acceptable
```

(base) C:\Users\jayso\Desktop\DM Project>\_

# Test Run of Fashion Dataset

Anaconda Prompt (Anaconda3)

```
(base) C:\Users\jayso\Desktop\DM Project>python apriori_sc_js.py ds_fashion.txt 20 20
```

```
oooooooooooooooooooo
```

```
Input Transactions
```

```
oooooooooooooooooooo
```

```
1 Joggers, Jacket, Tshirt, Shorts, Sneakers
2 Tshirt, WristWatch, Belt
3 Shirt, Jeans, Blazer, Tie, Perfume
4 Jeans, Shirt, Belt
5 Jacket, Tshirt, WristWatch
6 Hoodie, Tshirt, Perfume, Shorts, Sneakers, Belt
7 Shorts, SlipOns, Perfume
8 Belt, Blazer, Tie, WristWatch
9 Tie, Trouser, Shirt, Belt
10 WristWatch, Tshirt, Shoes
11 Sneakers, Trouser, Tshirt
12 SlipOns, WristWatch, Tshirt, Perfume
13 Shoes, Tie, Jacket
14 Wallet, Blazer, Tie, WristWatch
15 Perfume, WristWatch, Belt, Tshirt
16 Blazer, Tie, SlipOns
17 Belt, Blazer, Tie, WristWatch, Perfume
18 Trouser, Sneakers, Tshirt
19 Jacket, Tshirt, Shorts, Sneakers
20 Tshirt, Perfume, Blazer, Tie
```

```
oooooooooooooooooooo
```

```
Item Sets 1 Iteration
```

```
oooooooooooooooooooo
```

```
('Joggers',) 5
('Jacket',) 20
('Tshirt',) 55
('Shorts',) 20
('Sneakers',) 25
('WristWatch',) 40
('Belt',) 35
('Shirt',) 15
('Jeans',) 10
('Blazer',) 30
('Tie',) 40
('Perfume',) 35
('Hoodie',) 5
('SlipOns',) 15
('Trouser',) 15
('Shoes',) 10
('Wallet',) 5
```

```
oooooooooooooooooooo
```

```
Selected itemsets after 1 iteration
```

Selected itemsets after 1 iteration

Itemset Support

(Jacket,) 20.0  
 (Tshirt,) 55.0  
 (Shorts,) 20.0  
 (Sneakers,) 25.0  
 (WristWatch,) 40.0  
 (Belt,) 35.0  
 (Blazer,) 30.0  
 (Tie,) 40.0  
 (Perfume,) 35.0

Itemsets for 2 Iteration

(Jacket, Tshirt) 15.0  
 (Jacket, Shorts) 10.0  
 (Jacket, Sneakers) 10.0  
 (Jacket, WristWatch) 5.0  
 (Jacket, Tie) 5.0  
 (Shorts, Tshirt) 15.0  
 (Sneakers, Tshirt) 25.0  
 (Tshirt, WristWatch) 25.0  
 (Belt, Tshirt) 15.0  
 (Blazer, Tshirt) 5.0  
 (Tie, Tshirt) 5.0  
 (Perfume, Tshirt) 20.0  
 (Shorts, Sneakers) 15.0  
 (Belt, Shorts) 5.0  
 (Perfume, Shorts) 10.0  
 (Belt, Sneakers) 5.0  
 (Perfume, Sneakers) 5.0  
 (Belt, WristWatch) 20.0  
 (Blazer, WristWatch) 15.0  
 (Tie, WristWatch) 15.0  
 (Perfume, WristWatch) 15.0  
 (Belt, Blazer) 10.0  
 (Belt, Tie) 15.0  
 (Belt, Perfume) 15.0  
 (Blazer, Tie) 30.0  
 (Blazer, Perfume) 15.0  
 (Perfume, Tie) 15.0

Selected itemsets after 2 iteration

Itemset Support

Anaconda Prompt (Anaconda3)

```
('Perfume', 'Tie') 15.0

oooooooooooooooooooooooooooooooooooooooo
Selected itemsets after 2 iteration
__Itemset__ __Support__
oooooooooooooooooooooooooooooooooooooooo

('Sneakers', 'Tshirt') 25.0
('Tshirt', 'WristWatch') 25.0
('Perfume', 'Tshirt') 20.0
('Belt', 'WristWatch') 20.0
('Blazer', 'Tie') 30.0

Association Rule for Pair - ('Sneakers', 'Tshirt')
__Rule__ __Confidence__
('Sneakers',) => ('Tshirt',) 100.0 This Rule is Acceptable
('Tshirt',) => ('Sneakers',) 45.45 This Rule is Acceptable

Association Rule for Pair - ('Tshirt', 'WristWatch')
__Rule__ __Confidence__
('Tshirt',) => ('WristWatch',) 45.45 This Rule is Acceptable
('WristWatch',) => ('Tshirt',) 62.5 This Rule is Acceptable

Association Rule for Pair - ('Perfume', 'Tshirt')
__Rule__ __Confidence__
('Perfume',) => ('Tshirt',) 57.14 This Rule is Acceptable
('Tshirt',) => ('Perfume',) 36.36 This Rule is Acceptable

Association Rule for Pair - ('Belt', 'WristWatch')
__Rule__ __Confidence__
('Belt',) => ('WristWatch',) 57.14 This Rule is Acceptable
('WristWatch',) => ('Belt',) 50.0 This Rule is Acceptable

Association Rule for Pair - ('Blazer', 'Tie')
__Rule__ __Confidence__
('Blazer',) => ('Tie',) 100.0 This Rule is Acceptable
('Tie',) => ('Blazer',) 75.0 This Rule is Acceptable

(base) C:\Users\jayso\Desktop\DM Project>
```