

Reading File

```
In [368]: import csv
import time
import pandas as pd
import itertools
```

```
In [369]: def load_data(filename):
    full_transaction_list= []
    with open(filename, encoding = 'utf-8-sig') as data:
        transaction_data = csv.reader(data, delimiter = ',')
        for row in transaction_data:
            filtered_rows = [value for value in row if value != '']
            full_transaction_list.append(filtered_rows)
    return full_transaction_list
```

```
In [370]: #Asking the user to input the file
new_list = load_data(input('please enter file name\n'))
```

```
please enter file name
Transactions_2.csv
```

Asking user for min support and confidence

```
In [371]: try:
    user_input_minsupport =int(input('Please enter the minimum support
    user_input_minConfidence = int(input('\nPlease enter the min confi
except:
    if user_input_minsupport == int(user_input_minsupport) or user_inp
        print('please enter a numerical value')
```

```
Please enter the minimum support value in percentage ex: 25 is 25%:
```

```
35
```

```
Please enter the min confidence value n percentage ex: 50 is 50%:
```

```
75
```

Transaction 2

Apriori Algorithm

```
In [372]: start_time = time.time()

#function to find all the unique values with their counts
def Uniquevalues(Transactions):
    unique_items = {}
    for rows in Transactions:
        for items in rows:
            if items not in unique_items:
                unique_items[items] = 1
            else:
                unique_items[items] = unique_items[items] + 1
    uniqueitemlist = []
    for value in unique_items:
        Valuelist = []
        Valuelist.append(value)
        uniqueitemlist.append(Valuelist)
        uniqueitemlist.append(unique_items[value])
    return uniqueitemlist
```

```
In [373]: One_UniqueItems = Uniquevalues(new_list)
print('These are the unique items for Transaction 1:\n\n', One_UniqueI
```

These are the unique items for Transaction 1:

```
 [['DeskTop'], 6, ['Printer'], 10, ['FlashDrive'], 13, ['MicrosoftOff
ice'], 11, ['Speakers'], 11, ['Anti-Virus'], 14, ['LabTop'], 12, ['La
bTopCase'], 14, ['ExternalHard-Drive'], 9, ['DigitalCamera'], 9]
```

```
In [374]: #function used to remove the items that do not meet the threshold
def remove_lessthansupportone(Candidates, transactions):
    Firstcandidate_list= []
    for i in range(len(Candidates)):
        if i%2 != 0:
            if (Candidates[i] / len(new_list))*100 >= user_input_minsu
                Firstcandidate_list.append(Candidates[i-1])
                Firstcandidate_list.append(Candidates[i])
    candidatesforcombo = []
    for i in range(len(Firstcandidate_list)):
        if i%2 == 0:
            candidatesforcombo.append(Firstcandidate_list[i])
    return candidatesforcombo
```

```
In [375]: removed_first = remove_lessthansupportone(One_UniqueItems, new_list)
print('\nThese are the candidates after the first pass\n\n',removed_fi
```

These are the candidates after the first pass

```
 [['Printer'], ['FlashDrive'], ['MicrosoftOffice'], ['Speakers'], ['A
nti-Virus'], ['LabTop'], ['LabTopCase'], ['ExternalHard-Drive'], ['Di
gitalCamera']]
```

```
In [376]: #function used to output all possible combinations (k itemsets)
def Allpossiblecombinations(candidatesforcombo):
    if not candidatesforcombo:
        return [[]]
    first= candidatesforcombo[0]
    Allothers = candidatesforcombo[1:]
    Withoutfirst = Allpossiblecombinations(Allothers)
    Withfirst = [combo + [first] for combo in Withoutfirst]
    Combinedlist=Withoutfirst + Withfirst
    return Combinedlist
```

```
In [377]: All_combos = Allpossiblecombinations(removed_first)
```

```
In [378]: #function used to add the number of counts to the list provided before
def allcombosunique(Combination, dataset):
    from collections import Counter
    Count = Counter()
    for row in Combination:
        for s in dataset:
            if all(item in s for item in sum(row, [])):
                Count[tuple(map(tuple, row))] += 1
    listcount = [[list(subset), count] for subset, count in Count.items()]
    return(listcount)
```

```
In [379]: All_uniquecombos = allcombosunique(All_combos, new_list)
print('\nAll unique possible combinations\n',All_uniquecombos )
('Anti-Virus',), ('Speakers',), ('Printer',)], 2], [[('ExternalHard-Drive',), ('LabTopCase',), ('LabTop',), ('Anti-Virus',), ('Speakers',), ('Printer',)], 1], [[('DigitalCamera',), ('ExternalHard-Drive',), ('LabTopCase',), ('LabTop',), ('Anti-Virus',), ('Speakers',), ('Printer',)], 1], [[('MicrosoftOffice',), ('Printer',)], 9], [[('DigitalCamera',), ('MicrosoftOffice',), ('Printer',)], 3], [[('ExternalHard-Drive',), ('MicrosoftOffice',), ('Printer',)], 4], [[('DigitalCamera',), ('ExternalHard-Drive',), ('MicrosoftOffice',), ('Printer',)], 1], [[('LabTopCase',), ('MicrosoftOffice',), ('Printer',)], 5], [[('DigitalCamera',), ('LabTopCase',), ('MicrosoftOffice',), ('Printer',)], 2], [[('ExternalHard-Drive',), ('LabTopCase',), ('MicrosoftOffice',), ('Printer',)], 4], [[('DigitalCamera',), ('ExternalHard-Drive',), ('LabTopCase',), ('MicrosoftOffice',), ('Printer',)], 1], [[('LabTop',), ('MicrosoftOffice',), ('Printer',)], 4], [[('DigitalCamera',), ('LabTop',), ('MicrosoftOffice',), ('Printer',)], 2], [[('ExternalHard-Drive',), ('LabTop',), ('MicrosoftOffice',), ('Printer',)], 2], [[('DigitalCamera',), ('ExternalHard-Drive',), ('LabTop',), ('MicrosoftOffice',), ('Printer',)], 1], [[('LabTopCase',), ('LabTop',), ('MicrosoftOffice',), ('Printer',)], 3], [[('DigitalCamera',), ('LabTopCase',), ('LabTo
```

```
In [380]: #Second function to remove the items that do not meet the threshold
def remove_less thansupporttwo(Candidates, dataset):
    list1 = []
    for outlist in Candidates:
        if len(outlist) >= 2:
            second_object = outlist[1]
            if (second_object / len(new_list))*100 >= user_input_mins:
                list1.append(outlist[0])
                list1.append(outlist[1])
    return list1
```

```
In [381]: removed_second = remove_less_than_support_two(All_unique_combos, new_list)
print('\nThese are the candidates after the next pass\n', removed_second)
```

These are the candidates after the next pass

```
[[], 20, [['DigitalCamera',)], 9, [['ExternalHard-Drive',)], 9, [['LabTopCase',)], 14, [['DigitalCamera',), ('LabTopCase',)], 7, [['ExternalHard-Drive',), ('LabTopCase',)], 8, [['LabTop',)], 12, [['LabTopCase',), ('LabTop',)], 10, [['Anti-Virus',)], 14, [['ExternalHard-Drive',), ('Anti-Virus',)], 9, [['LabTopCase',), ('Anti-Virus',)], 12, [['ExternalHard-Drive',), ('LabTopCase',), ('Anti-Virus',)], 8, [['LabTop',), ('Anti-Virus',)], 10, [['LabTopCase',), ('LabTop',), ('Anti-Virus',)], 9, [['Speakers',)], 11, [['DigitalCamera',), ('Speakers',)], 7, [['LabTopCase',), ('Speakers',)], 9, [['Anti-Virus',), ('Speakers',)], 9, [['LabTopCase',), ('Anti-Virus',), ('Speakers',)], 8, [['MicrosoftOffice',)], 11, [['LabTopCase',), ('MicrosoftOffice',)], 7, [['Anti-Virus',), ('MicrosoftOffice',)], 8, [['LabTopCase',), ('Anti-Virus',), ('MicrosoftOffice',)], 7, [['FlashDrive',)], 13, [['LabTopCase',), ('FlashDrive',)], 9, [['LabTop',), ('FlashDrive',)], 7, [['Anti-Virus',), ('FlashDrive',)], 10, [['LabTopCase',), ('Anti-Virus',), ('FlashDrive',)], 9, [['MicrosoftOffice',), ('FlashDrive',)], 11, [['LabTopCase',), ('MicrosoftOffice',), ('FlashDrive',)], 7, [['Anti-Virus',), ('MicrosoftOffice',), ('FlashDrive',)], 8, [['LabTopCase',), ('Anti-Virus',), ('MicrosoftOffice',), ('FlashDrive',)], 7, [['Printer',)], 10, [['Anti-Virus',), ('Printer',)], 7, [['MicrosoftOffice',), ('Printer',)], 9, [['FlashDrive',), ('Printer',)], 10, [['Anti-Virus',), ('FlashDrive',), ('Printer',)], 7, [['MicrosoftOffice',), ('FlashDrive',), ('Printer',)], 9]
```

```
In [382]: def Rules(CandidateSet):
    CandidateRule = []
    for candidates in CandidateSet:
        if isinstance(candidates, list):
            if len(candidates) != 0:
                length_candidates = len(candidates) - 1
                while length_candidates > 0:
                    combos = list(itertools.combinations(candidates, length_candidates))
                    combolist = []
                    Left = []
                    for Right in combos:
                        Left = set(candidates) - set(Right)
                        combolist.append(list(Left))
                        combolist.append(list(Right))
                        CandidateRule.append(combolist)
                        combolist = []
                    length_candidates = length_candidates - 1

    return CandidateRule
```

```
In [383]: Associationrules = Rules(removed_second)
print('\nThese are the association rules\n\n',Associationrules )
```

These are the association rules

```
[[('LabTopCase',)], [('DigitalCamera',)], [('DigitalCamera',)],
[('LabTopCase',)], [('LabTopCase',)], [('ExternalHard-Drive',)],
[('ExternalHard-Drive',)], [('LabTopCase',)], [('LabTop',)], [('La
bTopCase',)], [('LabTopCase',)], [('LabTop',)], [('Anti-Viru
s',)], [('ExternalHard-Drive',)], [('ExternalHard-Drive',)], [('Ant
i-Virus',)], [('Anti-Virus',)], [('LabTopCase',)], [('LabTopCas
e',)], [('Anti-Virus',)], [('Anti-Virus',)], [('ExternalHard-Driv
e',)], ('LabTopCase',)], [('LabTopCase',)], [('ExternalHard-Driv
e',)], ('Anti-Virus',)], [('ExternalHard-Drive',)], [('LabTopCas
e',)], ('Anti-Virus',)], [('LabTopCase',)], ('Anti-Virus',)], [('Exte
rnalHard-Drive',)], [('ExternalHard-Drive',)], ('Anti-Virus',)], [('
LabTopCase',)], [('ExternalHard-Drive',)], ('LabTopCase',)], [('Anti
-Virus',)], [('Anti-Virus',)], [('LabTop',)], [('LabTop',)], [('A
nti-Virus',)], [('Anti-Virus',)], [('LabTopCase',)], ('LabTop',)],
[('LabTop',)], ('LabTopCase',)], ('Anti-Virus',)], [('LabTopCas
e',)], ('LabTop',)], ('Anti-Virus',)], [('LabTop',)], ('Anti-Viru
s',)], ('LabTopCase',)], [('LabTopCase',)], ('Anti-Virus',)], [('La
bTop',)], [('LabTop',)], ('LabTopCase',)], ('Anti-Virus',)], [('S
peakers',)], ('DigitalCamera',)], [('DigitalCamera',)], [('Speaker
s',)], [('Speakers',)], ('LabTopCase',)], [('LabTopCase',)], [('
Speakers',)], [('Speakers',)], ('Anti-Virus',)], [('Anti-Viru
s',)], ('Speakers',)], [('Speakers',)], ('LabTopCase',)], ('Anti-V
irus',)], [('Anti-Virus',)], ('LabTopCase',)], ('Speakers',)],
[('LabTopCase',)], ('Anti-Virus',)], ('Speakers',)], [('Speaker
s',)], ('Anti-Virus',)], ('LabTopCase',)], [('Speakers',)], ('LabTop
Case',)], ('Anti-Virus',)], [('LabTopCase',)], ('Anti-Virus',)],
[('Speakers',)], [('MicrosoftOffice',)], ('LabTopCase',)], [('La
bTopCase',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], [('An
ti-Virus',)], [('Anti-Virus',)], ('MicrosoftOffice',)], [('Micro
softOffice',)], ('LabTopCase',)], ('Anti-Virus',)], [('Anti-Viru
s',)], ('LabTopCase',)], ('MicrosoftOffice',)], [('LabTopCase',)],
[('Anti-Virus',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('A
nti-Virus',)], ('LabTopCase',)], [('MicrosoftOffice',)], ('LabTopCa
se',)], ('Anti-Virus',)], [('LabTopCase',)], ('Anti-Virus',)], [('M
icrosoftOffice',)], [('FlashDrive',)], ('LabTopCase',)], [('LabT
opCase',)], ('FlashDrive',)], [('FlashDrive',)], ('LabTop',)],
[('LabTop',)], ('FlashDrive',)], [('FlashDrive',)], ('Anti-Viru
s',)], [('Anti-Virus',)], ('FlashDrive',)], [('FlashDrive',)],
[('LabTopCase',)], ('Anti-Virus',)], [('Anti-Virus',)], ('LabTopCas
e',)], ('FlashDrive',)], [('LabTopCase',)], ('Anti-Virus',)], ('Flas
hDrive',)], [('FlashDrive',)], ('Anti-Virus',)], ('LabTopCase',)],
[('FlashDrive',)], ('LabTopCase',)], ('Anti-Virus',)], [('LabTopCa
se',)], ('Anti-Virus',)], ('FlashDrive',)], [('FlashDrive',)], [('M
icrosoftOffice',)], [('MicrosoftOffice',)], ('FlashDrive',)],
```

```

[['FlashDrive',)], [('LabTopCase',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], [('LabTopCase',)], ('FlashDrive',)], [('LabTopCase',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('FlashDrive',)], ('MicrosoftOffice',)], ('LabTopCase',)], [('FlashDrive',)], ('LabTopCase',)], ('MicrosoftOffice',)], [('FlashDrive',)], ('Anti-Virus',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('Anti-Virus',)], ('FlashDrive',)], [('Anti-Virus',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('FlashDrive',)], ('MicrosoftOffice',)], ('Anti-Virus',)], [('FlashDrive',)], ('Anti-Virus',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('Anti-Virus',)], ('FlashDrive',)], [('FlashDrive',)], ('LabTopCase',)], ('Anti-Virus',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('LabTopCase',)], ('Anti-Virus',)], ('FlashDrive',)], [('Anti-Virus',)], ('LabTopCase',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('LabTopCase',)], ('Anti-Virus',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('FlashDrive',)], ('MicrosoftOffice',)], ('LabTopCase',)], ('Anti-Virus',)], [('FlashDrive',)], ('Anti-Virus',)], ('LabTopCase',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('Anti-Virus',)], ('LabTopCase',)], ('FlashDrive',)], [('FlashDrive',)], ('LabTopCase',)], ('Anti-Virus',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('LabTopCase',)], ('Anti-Virus',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('LabTopCase',)], ('Anti-Virus',)], ('MicrosoftOffice',)], ('FlashDrive',)], ('Anti-Virus',)], ('LabTopCase',)], ('MicrosoftOffice',)], ('FlashDrive',)], ('Anti-Virus',)], ('LabTopCase',)], ('MicrosoftOffice',)], ('FlashDrive',)], ('Printer',)], ('Anti-Virus',)], [('Anti-Virus',)], ('Printer',)], [('Printer',)], ('MicrosoftOffice',)], [('MicrosoftOffice',)], ('Printer',)], [('Printer',)], ('FlashDrive',)], [('FlashDrive',)], ('Printer',)], [('Printer',)], ('Anti-Virus',)], ('FlashDrive',)], [('FlashDrive',)], ('Anti-Virus',)], ('Printer',)], [('Anti-Virus',)], ('FlashDrive',)], ('Printer',)], [('FlashDrive',)], ('Printer',)], ('Anti-Virus',)], [('Printer',)], ('Anti-Virus',)], ('FlashDrive',)], [('FlashDrive',)], ('Anti-Virus',)], ('Printer',)], [('Printer',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('FlashDrive',)], ('MicrosoftOffice',)], ('Printer',)], [('MicrosoftOffice',)], ('FlashDrive',)], ('Printer',)], [('FlashDrive',)], ('Printer',)], ('MicrosoftOffice',)], [('Printer',)], ('MicrosoftOffice',)], ('FlashDrive',)], [('FlashDrive',)], ('MicrosoftOffice',)], ('Printer',)]]

```

```
In [384]: def Apriori(Associationrules, new_list, user_input_minConfidence):
    AAlgorithm = []
    for rule in Associationrules:
        first = set(item[0] for item in rule[0])
        ASupport = 0
        ABsupport = 0
        for transaction in new_list:
            if first.issubset(set(transaction)):
                ASupport += 1
            if all(set(item) <= set(transaction) for each in rule for
                ABsupport += 1
        CalculateASupport = (ASupport * 1.0 / len(new_list)) * 100
        CalculateABSupport = (ABsupport * 1.0 / len(new_list)) * 100
        confidence = (CalculateABSupport / CalculateASupport) * 100
        if confidence >= user_input_minConfidence:
            OutputASupport = "A Support is: " + str(round(CalculateASu
            OutputABSupport = "\nA&B support is: " + str(CalculateABSu
            OutputConfidence = "\nConfidence is: " + str(round(confide
            AAlgorithm.append(OutputASupport)
            AAlgorithm.append(OutputABSupport)
            AAlgorithm.append(OutputConfidence)
            AAlgorithm.append(rule)
    return AAlgorithm
```

```
In [385]: Apriori = Apriori(Associationrules, new_list, user_input_minConfidence
print('\nApriori algorithm\n', Apriori )
```

Apriori algorithm

```
['A Support is: 45.0', '\nA&B support is: 35.0', '\nConfidence is: 7
8', [[('DigitalCamera',)], [('LabTopCase',)]]], 'A Support is: 45.0',
'\nA&B support is: 40.0', '\nConfidence is: 89', [[('ExternalHard-Dri
ve',)], [('LabTopCase',)]]], 'A Support is: 60.0', '\nA&B support is:
50.0', '\nConfidence is: 83', [[('LabTop',)], [('LabTopCase',)]]], 'A
Support is: 45.0', '\nA&B support is: 45.0', '\nConfidence is: 100',
[[('ExternalHard-Drive',)], [('Anti-Virus',)]]], 'A Support is: 70.0',
'\nA&B support is: 60.0', '\nConfidence is: 86', [[('Anti-Virus',)],
[('LabTopCase',)]]], 'A Support is: 70.0', '\nA&B support is: 60.0', '
\nConfidence is: 86', [[('LabTopCase',)], [('Anti-Virus',)]]], 'A Supp
ort is: 45.0', '\nA&B support is: 40.0', '\nConfidence is: 89', [[('E
xternalHard-Drive',)], [('LabTopCase',), ('Anti-Virus',)]]], 'A Suppor
t is: 45.0', '\nA&B support is: 40.0', '\nConfidence is: 89', [[('Ext
ernalHard-Drive',), ('Anti-Virus',)], [('LabTopCase',)]]], 'A Support
is: 40.0', '\nA&B support is: 40.0', '\nConfidence is: 100', [[('Exte
rnalHard-Drive',), ('LabTopCase',)], [('Anti-Virus',)]]], 'A Support i
s: 60.0', '\nA&B support is: 50.0', '\nConfidence is: 83', [[('LabTo
p',)], [('Anti-Virus',)]]], 'A Support is: 60.0', '\nA&B support is: 4
```



```

5.0', '\nConfidence is: 75', [[('LabTop',)], [('LabTopCase',)], ('Anti-
-Virus',)]], 'A Support is: 50.0', '\nA&B support is: 45.0', '\nConfi
dence is: 90', [[('LabTop',)], ('Anti-Virus',)], [('LabTopCase',)]], '
A Support is: 60.0', '\nA&B support is: 45.0', '\nConfidence is: 75',
[('LabTopCase',)], ('Anti-Virus',)], [('LabTop',)]], 'A Support is: 5
0.0', '\nA&B support is: 45.0', '\nConfidence is: 90', [[('LabTop',)],
('LabTopCase',)], [('Anti-Virus',)]], 'A Support is: 45.0', '\nA&B su
pport is: 35.0', '\nConfidence is: 78', [[('DigitalCamera',)], [('Spe
akers',)]], 'A Support is: 55.0', '\nA&B support is: 45.0', '\nConfid
ence is: 82', [[('Speakers',)], [('LabTopCase',)]], 'A Support is: 5
5.0', '\nA&B support is: 45.0', '\nConfidence is: 82', [[('Speaker
s',)], [('Anti-Virus',)]], 'A Support is: 45.0', '\nA&B support is: 4
0.0', '\nConfidence is: 89', [[('Speakers',)], ('Anti-Virus',)], [('La
bTopCase',)]], 'A Support is: 45.0', '\nA&B support is: 40.0', '\nCon
fidence is: 89', [[('Speakers',)], ('LabTopCase',)], [('Anti-Viru
s',)]], 'A Support is: 40.0', '\nA&B support is: 35.0', '\nConfidence
is: 88', [[('MicrosoftOffice',)], ('Anti-Virus',)], [('LabTopCas
e',)]], 'A Support is: 35.0', '\nA&B support is: 35.0', '\nConfidence
is: 100', [[('MicrosoftOffice',)], ('LabTopCase',)], [('Anti-Viru
s',)]], 'A Support is: 65.0', '\nA&B support is: 50.0', '\nConfidence
is: 77', [[('FlashDrive',)], [('Anti-Virus',)]], 'A Support is: 50.
0', '\nA&B support is: 45.0', '\nConfidence is: 90', [[('FlashDriv
e',)], ('Anti-Virus',)], [('LabTopCase',)]], 'A Support is: 45.0', '\n
A&B support is: 45.0', '\nConfidence is: 100', [[('FlashDrive',)], ('L
abTopCase',)], [('Anti-Virus',)]], 'A Support is: 60.0', '\nA&B suppo
rt is: 45.0', '\nConfidence is: 75', [[('LabTopCase',)], ('Anti-Viru
s',)], [('FlashDrive',)]], 'A Support is: 65.0', '\nA&B support is: 5
5.000000000000001', '\nConfidence is: 85', [[('FlashDrive',)], [('Micr
osoftOffice',)]], 'A Support is: 55.0', '\nA&B support is: 55.0000000
0000001', '\nConfidence is: 100', [[('MicrosoftOffice',)], [('FlashDr
ive',)]], 'A Support is: 45.0', '\nA&B support is: 35.0', '\nConfiden
ce is: 78', [[('FlashDrive',)], ('LabTopCase',)], [('MicrosoftOffic
e',)]], 'A Support is: 35.0', '\nA&B support is: 35.0', '\nConfidence
is: 100', [[('MicrosoftOffice',)], ('LabTopCase',)], [('FlashDriv
e',)]], 'A Support is: 50.0', '\nA&B support is: 40.0', '\nConfidence
is: 80', [[('FlashDrive',)], ('Anti-Virus',)], [('MicrosoftOffic
e',)]], 'A Support is: 40.0', '\nA&B support is: 40.0', '\nConfidence
is: 100', [[('MicrosoftOffice',)], ('Anti-Virus',)], [('FlashDriv
e',)]], 'A Support is: 40.0', '\nA&B support is: 35.0', '\nConfidence
is: 88', [[('MicrosoftOffice',)], ('Anti-Virus',)], [('LabTopCase',)],
('FlashDrive',)]], 'A Support is: 45.0', '\nA&B support is: 35.0', '\n
Confidence is: 78', [[('FlashDrive',)], ('LabTopCase',)], [('Anti-Vir
us',)], ('MicrosoftOffice',)]], 'A Support is: 35.0', '\nA&B support i
s: 35.0', '\nConfidence is: 100', [[('MicrosoftOffice',)], ('LabTopCas
e',)], [('Anti-Virus',)], ('FlashDrive',)]], 'A Support is: 40.0', '\n
A&B support is: 35.0', '\nConfidence is: 88', [[('FlashDrive',)], ('Mi
crosoftOffice',)], ('Anti-Virus',)], [('LabTopCase',)]], 'A Support i
s: 35.0', '\nA&B support is: 35.0', '\nConfidence is: 100', [[('Flash
Drive',)], ('MicrosoftOffice',)], ('LabTopCase',)], [('Anti-Virus',)]],
'A Support is: 45.0', '\nA&B support is: 35.0', '\nConfidence is: 7

```

```
8', [[('FlashDrive',), ('LabTopCase',), ('Anti-Virus',)], [('MicrosoftOffice',)]], 'A Support is: 35.0', '\nA&B support is: 35.0', '\nConfidence is: 100', [[('LabTopCase',), ('MicrosoftOffice',), ('Anti-Virus',)], [('FlashDrive',)]], 'A Support is: 50.0', '\nA&B support is: 45.0', '\nConfidence is: 90', [[('Printer',)], [('MicrosoftOffice',)]], 'A Support is: 55.0', '\nA&B support is: 45.0', '\nConfidence is: 82', [[('MicrosoftOffice',)], [('Printer',)]], 'A Support is: 50.0', '\nA&B support is: 50.0', '\nConfidence is: 100', [[('Printer',)], [('FlashDrive',)]], 'A Support is: 65.0', '\nA&B support is: 50.0', '\nConfidence is: 77', [[('FlashDrive',)], [('Printer',)]], 'A Support is: 35.0', '\nA&B support is: 35.0', '\nConfidence is: 100', [[('Printer',), ('Anti-Virus',)], [('FlashDrive',)]], 'A Support is: 50.0', '\nA&B support is: 45.0', '\nConfidence is: 90', [[('Printer',)], [('MicrosoftOffice',), ('FlashDrive',)]], 'A Support is: 55.0', '\nA&B support is: 45.0', '\nConfidence is: 82', [[('MicrosoftOffice',)], [('FlashDrive',), ('Printer',)]], 'A Support is: 50.0', '\nA&B support is: 45.0', '\nConfidence is: 90', [[('FlashDrive',), ('Printer',)], [('MicrosoftOffice',)]], 'A Support is: 45.0', '\nA&B support is: 45.0', '\nConfidence is: 100', [[('Printer',), ('MicrosoftOffice',)], [('FlashDrive',)]], 'A Support is: 55.0', '\nA&B support is: 45.0', '\nConfidence is: 82', [[('FlashDrive',), ('MicrosoftOffice',)], [('Printer',)]]]
```

```
In [386]: counter = 1
for i in Apriori:
    if counter == 4:
        print("\n"+str(i[0]) + "----->" + str(i[1])+"\n")
        counter = 0
    else:
        print(i, end=' ')
        counter = counter + 1
elapsed_time = time.time() - start_time
print("---- %s seconds ----" % (elapsed_time))
```

```
A Support is: 45.0
A&B support is: 35.0
Confidence is: 78
[('DigitalCamera',)]----->[('LabTopCase',)]
```

```
A Support is: 45.0
A&B support is: 40.0
Confidence is: 89
[('ExternalHard-Drive',)]----->[('LabTopCase',)]
```

```
A Support is: 60.0
A&B support is: 50.0
Confidence is: 83
[('LabTop',)]----->[('LabTopCase',)]
```

```
A Support is: 45.0
```

A&B support is: 45.0
 Confidence is: 100
 [('ExternalHard-Drive',)]-----> [('Anti-Virus',)]

A Support is: 70.0
 A&B support is: 60.0
 Confidence is: 86
 [('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 70.0
 A&B support is: 60.0
 Confidence is: 86
 [('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 45.0
 A&B support is: 40.0
 Confidence is: 89
 [('ExternalHard-Drive',)]-----> [('LabTopCase',), ('Anti-Virus',)]

A Support is: 45.0
 A&B support is: 40.0
 Confidence is: 89
 [('ExternalHard-Drive',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 40.0
 A&B support is: 40.0
 Confidence is: 100
 [('ExternalHard-Drive',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 60.0
 A&B support is: 50.0
 Confidence is: 83
 [('LabTop',)]-----> [('Anti-Virus',)]

A Support is: 60.0
 A&B support is: 45.0
 Confidence is: 75
 [('LabTop',)]-----> [('LabTopCase',), ('Anti-Virus',)]

A Support is: 50.0
 A&B support is: 45.0
 Confidence is: 90
 [('LabTop',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 60.0
 A&B support is: 45.0
 Confidence is: 75
 [('LabTopCase',), ('Anti-Virus',)]-----> [('LabTop',)]

A Support is: 50.0

A&B support is: 45.0
 Confidence is: 90
 [('LabTop',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 45.0
 A&B support is: 35.0
 Confidence is: 78
 [('DigitalCamera',)]-----> [('Speakers',)]

A Support is: 55.0
 A&B support is: 45.0
 Confidence is: 82
 [('Speakers',)]-----> [('LabTopCase',)]

A Support is: 55.0
 A&B support is: 45.0
 Confidence is: 82
 [('Speakers',)]-----> [('Anti-Virus',)]

A Support is: 45.0
 A&B support is: 40.0
 Confidence is: 89
 [('Speakers',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 45.0
 A&B support is: 40.0
 Confidence is: 89
 [('Speakers',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 40.0
 A&B support is: 35.0
 Confidence is: 88
 [('MicrosoftOffice',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 35.0
 A&B support is: 35.0
 Confidence is: 100
 [('MicrosoftOffice',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 65.0
 A&B support is: 50.0
 Confidence is: 77
 [('FlashDrive',)]-----> [('Anti-Virus',)]

A Support is: 50.0
 A&B support is: 45.0
 Confidence is: 90
 [('FlashDrive',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 45.0

A&B support is: 45.0
 Confidence is: 100
 [('FlashDrive',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 60.0
 A&B support is: 45.0
 Confidence is: 75
 [('LabTopCase',), ('Anti-Virus',)]-----> [('FlashDrive',)]

A Support is: 65.0
 A&B support is: 55.000000000000001
 Confidence is: 85
 [('FlashDrive',)]-----> [('MicrosoftOffice',)]

A Support is: 55.0
 A&B support is: 55.000000000000001
 Confidence is: 100
 [('MicrosoftOffice',)]-----> [('FlashDrive',)]

A Support is: 45.0
 A&B support is: 35.0
 Confidence is: 78
 [('FlashDrive',), ('LabTopCase',)]-----> [('MicrosoftOffice',)]

A Support is: 35.0
 A&B support is: 35.0
 Confidence is: 100
 [('MicrosoftOffice',), ('LabTopCase',)]-----> [('FlashDrive',)]

A Support is: 50.0
 A&B support is: 40.0
 Confidence is: 80
 [('FlashDrive',), ('Anti-Virus',)]-----> [('MicrosoftOffice',)]

A Support is: 40.0
 A&B support is: 40.0
 Confidence is: 100
 [('MicrosoftOffice',), ('Anti-Virus',)]-----> [('FlashDrive',)]

A Support is: 40.0
 A&B support is: 35.0
 Confidence is: 88
 [('MicrosoftOffice',), ('Anti-Virus',)]-----> [('LabTopCase',), ('FlashDrive',)]

A Support is: 45.0
 A&B support is: 35.0
 Confidence is: 78
 [('FlashDrive',), ('LabTopCase',)]-----> [('Anti-Virus',), ('MicrosoftOffice',)]

A Support is: 35.0
 A&B support is: 35.0
 Confidence is: 100
 [('MicrosoftOffice',), ('LabTopCase',)]-----> [('Anti-Virus',), ('FlashDrive',)]

A Support is: 40.0
 A&B support is: 35.0
 Confidence is: 88
 [('FlashDrive',), ('MicrosoftOffice',), ('Anti-Virus',)]-----> [('LabTopCase',)]

A Support is: 35.0
 A&B support is: 35.0
 Confidence is: 100
 [('FlashDrive',), ('MicrosoftOffice',), ('LabTopCase',)]-----> [('Anti-Virus',)]

A Support is: 45.0
 A&B support is: 35.0
 Confidence is: 78
 [('FlashDrive',), ('LabTopCase',), ('Anti-Virus',)]-----> [('MicrosoftOffice',)]

A Support is: 35.0
 A&B support is: 35.0
 Confidence is: 100
 [('LabTopCase',), ('MicrosoftOffice',), ('Anti-Virus',)]-----> [('FlashDrive',)]

A Support is: 50.0
 A&B support is: 45.0
 Confidence is: 90
 [('Printer',)]-----> [('MicrosoftOffice',)]

A Support is: 55.0
 A&B support is: 45.0
 Confidence is: 82
 [('MicrosoftOffice',)]-----> [('Printer',)]

A Support is: 50.0
 A&B support is: 50.0
 Confidence is: 100
 [('Printer',)]-----> [('FlashDrive',)]

A Support is: 65.0
 A&B support is: 50.0
 Confidence is: 77
 [('FlashDrive',)]-----> [('Printer',)]

```

A Support is: 35.0
A&B support is: 35.0
Confidence is: 100
[('Printer',), ('Anti-Virus',)]----->[('FlashDrive',)]

A Support is: 50.0
A&B support is: 45.0
Confidence is: 90
[('Printer',)]----->[('MicrosoftOffice',), ('FlashDrive',)]

A Support is: 55.0
A&B support is: 45.0
Confidence is: 82
[('MicrosoftOffice',)]----->[('FlashDrive',), ('Printer',)]

A Support is: 50.0
A&B support is: 45.0
Confidence is: 90
[('FlashDrive',), ('Printer',)]----->[('MicrosoftOffice',)]

A Support is: 45.0
A&B support is: 45.0
Confidence is: 100
[('Printer',), ('MicrosoftOffice',)]----->[('FlashDrive',)]

A Support is: 55.0
A&B support is: 45.0
Confidence is: 82
[('FlashDrive',), ('MicrosoftOffice',)]----->[('Printer',)]

--- 0.09882998466491699 seconds ---

```

Brute Force

```

In [387]: import pandas as pd
import time
from itertools import combinations

```

```

In [388]: Transactiondata = input("Enter the file name: ")
minsupport = float(input('Please enter the minimum support value'))

Enter the file name: Transactions_2.csv
Please enter the minimum support value35

```

```
In [389]: start_time = time.time()
transaction = pd.read_csv(Transactiondata, header =None)
TransactionforSum = pd.get_dummies(transaction.unstack().dropna()).groupby(transaction.index).sum()
UniqueItems = TransactionforSum.sum()
```

```
In [390]: print('\nThese are all unique one item sets:\n\n', UniqueItems)
```

These are all unique one item sets:

Anti-Virus	14
DeskTop	6
DigitalCamera	9
ExternalHard-Drive	9
FlashDrive	13
LabTop	12
LabTopCase	14
MicrosoftOffice	11
Printer	10
Speakers	11
dtype: int64	

```
In [391]: OneItemSets = pd.DataFrame((UniqueItems / len(transaction) * 100), columns=['support'])
OneFrequentItems = OneItemSets[OneItemSets['support'] >= minsupport]
print('These are the Frequent One Item sets:\n', OneFrequentItems)
```

These are the Frequent One Item sets:

	support
Anti-Virus	70.0
DigitalCamera	45.0
ExternalHard-Drive	45.0
FlashDrive	65.0
LabTop	60.0
LabTopCase	70.0
MicrosoftOffice	55.0
Printer	50.0
Speakers	55.0

```
In [392]: import itertools
items = UniqueItems.index
combos = list(itertools.combinations(items, 2))
combinations = []
for combo in combos:
    combinations.append(combo)
```

```
In [393]: combo_counts = {}
for i in range(len(combinations)):
    combo = combinations[i]
```



```

count = 0
for index, row in transaction.iterrows():
    if set(combo).issubset(row):
        count += 1
    combo_counts[i+1] = count
print('Thesea are all the two possible combinations:\n\n')
for combo_num, count in combo_counts.items():
    print(f"({combinations[combo_num-1]}) , Number of repetitions {count}")

```

Thesea are all the two possible combinations:

```

(('Anti-Virus', 'DeskTop')) , Number of repetitions 4.
(('Anti-Virus', 'DigitalCamera')) , Number of repetitions 5.
(('Anti-Virus', 'ExternalHard-Drive')) , Number of repetitions 9.
(('Anti-Virus', 'FlashDrive')) , Number of repetitions 10.
(('Anti-Virus', 'LabTop')) , Number of repetitions 10.
(('Anti-Virus', 'LabTopCase')) , Number of repetitions 12.
(('Anti-Virus', 'MicrosoftOffice')) , Number of repetitions 8.
(('Anti-Virus', 'Printer')) , Number of repetitions 7.
(('Anti-Virus', 'Speakers')) , Number of repetitions 9.
(('DeskTop', 'DigitalCamera')) , Number of repetitions 2.
(('DeskTop', 'ExternalHard-Drive')) , Number of repetitions 3.
(('DeskTop', 'FlashDrive')) , Number of repetitions 5.
(('DeskTop', 'LabTop')) , Number of repetitions 2.
(('DeskTop', 'LabTopCase')) , Number of repetitions 3.
(('DeskTop', 'MicrosoftOffice')) , Number of repetitions 5.
(('DeskTop', 'Printer')) , Number of repetitions 5.
(('DeskTop', 'Speakers')) , Number of repetitions 4.
(('DigitalCamera', 'ExternalHard-Drive')) , Number of repetitions 3.
(('DigitalCamera', 'FlashDrive')) , Number of repetitions 4.
(('DigitalCamera', 'LabTop')) , Number of repetitions 5.
(('DigitalCamera', 'LabTopCase')) , Number of repetitions 7.
(('DigitalCamera', 'MicrosoftOffice')) , Number of repetitions 4.
(('DigitalCamera', 'Printer')) , Number of repetitions 3.
(('DigitalCamera', 'Speakers')) , Number of repetitions 7.
(('ExternalHard-Drive', 'FlashDrive')) , Number of repetitions 6.
(('ExternalHard-Drive', 'LabTop')) , Number of repetitions 6.
(('ExternalHard-Drive', 'LabTopCase')) , Number of repetitions 8.
(('ExternalHard-Drive', 'MicrosoftOffice')) , Number of repetitions
5.
(('ExternalHard-Drive', 'Printer')) , Number of repetitions 5.
(('ExternalHard-Drive', 'Speakers')) , Number of repetitions 6.
(('FlashDrive', 'LabTop')) , Number of repetitions 7.
(('FlashDrive', 'LabTopCase')) , Number of repetitions 9.
(('FlashDrive', 'MicrosoftOffice')) , Number of repetitions 11.
(('FlashDrive', 'Printer')) , Number of repetitions 10.
(('FlashDrive', 'Speakers')) , Number of repetitions 6.
(('LabTop', 'LabTopCase')) , Number of repetitions 10.
(('LabTop', 'MicrosoftOffice')) , Number of repetitions 5.

```

```
(('LabTop', 'Printer')) , Number of repetitions 5.
(('LabTop', 'Speakers')) , Number of repetitions 5.
(('LabTopCase', 'MicrosoftOffice')) , Number of repetitions 7.
(('LabTopCase', 'Printer')) , Number of repetitions 6.
(('LabTopCase', 'Speakers')) , Number of repetitions 9.
(('MicrosoftOffice', 'Printer')) , Number of repetitions 9.
(('MicrosoftOffice', 'Speakers')) , Number of repetitions 6.
(('Printer', 'Speakers')) , Number of repetitions 5.
```

```
In [394]: print('These are the 2 frequent itemsets:\n\n')
for combo_num, count in combo_counts.items():
    if (count / len(transaction) * 100) >= minsupport:
        print(f" ({{combinations[combo_num-1]}})    number of repetition:
```

These are the 2 frequent itemsets:

```
(('Anti-Virus', 'ExternalHard-Drive'))    number of repetition: 9.
(('Anti-Virus', 'FlashDrive'))    number of repetition: 10.
(('Anti-Virus', 'LabTop'))    number of repetition: 10.
(('Anti-Virus', 'LabTopCase'))    number of repetition: 12.
(('Anti-Virus', 'MicrosoftOffice'))    number of repetition: 8.
(('Anti-Virus', 'Printer'))    number of repetition: 7.
(('Anti-Virus', 'Speakers'))    number of repetition: 9.
(('DigitalCamera', 'LabTopCase'))    number of repetition: 7.
(('DigitalCamera', 'Speakers'))    number of repetition: 7.
(('ExternalHard-Drive', 'LabTopCase'))    number of repetition: 8.
(('FlashDrive', 'LabTop'))    number of repetition: 7.
(('FlashDrive', 'LabTopCase'))    number of repetition: 9.
(('FlashDrive', 'MicrosoftOffice'))    number of repetition: 11.
(('FlashDrive', 'Printer'))    number of repetition: 10.
(('LabTop', 'LabTopCase'))    number of repetition: 10.
(('LabTopCase', 'MicrosoftOffice'))    number of repetition: 7.
(('LabTopCase', 'Speakers'))    number of repetition: 9.
(('MicrosoftOffice', 'Printer'))    number of repetition: 9.
```

```
In [395]: combinations = []
for r in range(3,4):
    combos = list(itertools.combinations(items, r))
    combinations.extend(combos)

# Filter out empty tuples
combinations = [combo for combo in combinations if combo]
```

```
In [396]: combo_counts = {}
for i in range(len(combinations)):
    combo = combinations[i]
    count = 0
    for index, row in transaction.iterrows():
        if set(combo).issubset(row):
            count += 1
    combo_counts[i+1] = count
print('These are the 3 possible combinations\n\n')
for combo_num, count in combo_counts.items():
    print(f"({combinations[combo_num-1]}) number of repetitions {count}")
```

These are the 3 possible combinations

((('Anti-Virus', 'DeskTop', 'DigitalCamera')) number of repetitions
1

((('Anti-Virus', 'DeskTop', 'ExternalHard-Drive')) number of repetit
ions 3

((('Anti-Virus', 'DeskTop', 'FlashDrive')) number of repetitions 3

((('Anti-Virus', 'DeskTop', 'LabTop')) number of repetitions 2

((('Anti-Virus', 'DeskTop', 'LabTopCase')) number of repetitions 3

((('Anti-Virus', 'DeskTop', 'MicrosoftOffice')) number of repetition
s 3

((('Anti-Virus', 'DeskTop', 'Printer')) number of repetitions 3

```
In [397]: print('These are the 3 Frequent item set:\n\n')
for combo_num, count in combo_counts.items():
    if (count / len(transaction) * 100) >= minsupport:
        print(f"({combinations[combo_num-1]}) number of repetitions {c
```

These are the 3 Frequent item set:

(('Anti-Virus', 'ExternalHard-Drive', 'LabTopCase')) number of repetitions 8

(('Anti-Virus', 'FlashDrive', 'LabTopCase')) number of repetitions 9

(('Anti-Virus', 'FlashDrive', 'MicrosoftOffice')) number of repetitions 8

(('Anti-Virus', 'FlashDrive', 'Printer')) number of repetitions 7

(('Anti-Virus', 'LabTop', 'LabTopCase')) number of repetitions 9

(('Anti-Virus', 'LabTopCase', 'MicrosoftOffice')) number of repetitions 7

(('Anti-Virus', 'LabTopCase', 'Speakers')) number of repetitions 8

(('FlashDrive', 'LabTopCase', 'MicrosoftOffice')) number of repetitions 7

(('FlashDrive', 'MicrosoftOffice', 'Printer')) number of repetitions 9

```
In [398]: combinations = []
for r in range(4,20):
    combos = list(itertools.combinations(items, r))
    combinations.extend(combos)

# Filter out empty tuples
combinations = [combo for combo in combinations if combo]
```

```
In [399]: combo_counts = {}
for i in range(len(combinations)):
    combo = combinations[i]
    count = 0
    for index, row in transaction.iterrows():
        if set(combo).issubset(row):
            count += 1
            #print(f"Combo {i+1} ({combo}) is a subset of row {index}")
    combo_counts[i+1] = count
print('These are the 4 and 4+ Frequent item sets:\n\n')
for combo_num, count in combo_counts.items():
    print(f"({combinations[combo_num-1]}) number of repetitions {count}")
```

These are the 4 and 4+ Frequent item sets:

(('Anti-Virus', 'DeskTop', 'DigitalCamera', 'ExternalHard-Drive'))
number of repetitions 1.

(('Anti-Virus', 'DeskTop', 'DigitalCamera', 'FlashDrive')) number o
f repetitions 1.

(('Anti-Virus', 'DeskTop', 'DigitalCamera', 'LabTop')) number of re
petitions 1.

(('Anti-Virus', 'DeskTop', 'DigitalCamera', 'LabTopCase')) number o
f repetitions 1.

(('Anti-Virus', 'DeskTop', 'DigitalCamera', 'MicrosoftOffice')) num

```
In [400]: print('These are all the other frequent item sets:\n')
for combo_num, count in combo_counts.items():
    if (count / len(transaction) * 100) >= minsupport:
        print(f"({combinations[combo_num-1]}) number of repetitions: {count}")
```

These are all the other frequent item sets:

(('Anti-Virus', 'FlashDrive', 'LabTopCase', 'MicrosoftOffice')) numbe
r of repetitions: 7 .

```
In [401]: elapsed_time = time.time() - start_time
print("--- %s seconds ---" % (elapsed_time))
```

--- 0.4260389804840088 seconds ---

Applying Apriori Library

```
In [402]: import csv
import pandas as pd
import time
from itertools import combinations
from apyori import apriori
from mlxtend.frequent_patterns import apriori, fpmmax
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
```

```
In [403]: #The below will allow us to read the file
def load_data(filename):
    full_transaction_list= []
    with open(filename, encoding = 'utf-8-sig') as data:
        transaction_data = csv.reader(data, delimiter = ',')
        for row in transaction_data:
            filtered_rows = [value for value in row if value != '']
            full_transaction_list.append(filtered_rows)
    return full_transaction_list
```

```
In [404]: new_list = load_data(input('please enter file name\n\n'))

please enter file name

Transactions_2.csv
```

```
In [405]: new_list
```

```
Out[405]: [['DeskTop',
'Printer',
'FlashDrive',
'MicrosoftOffice',
'Speakers',
'Anti-Virus'],
['LabTop', 'FlashDrive', 'MicrosoftOffice', 'LabTopCase', 'Anti-Virus'],
['LabTop',
'Printer',
'FlashDrive',
'MicrosoftOffice',
'Anti-Virus',
'LabTopCase',
'ExternalHard-Drive'],
['LabTop',
'Printer',
```

```

'FlashDrive',
'Anti-Virus',
'ExternalHard-Drive',
'LabTopCase'],
['LabTop', 'FlashDrive', 'LabTopCase', 'Anti-Virus'],
['LabTop', 'Printer', 'FlashDrive', 'MicrosoftOffice'],
['DeskTop', 'Printer', 'FlashDrive', 'MicrosoftOffice'],
['LabTop', 'ExternalHard-Drive', 'Anti-Virus'],
['DeskTop',
'Printer',
'FlashDrive',
'MicrosoftOffice',
'LabTopCase',
'Anti-Virus',
'Speakers',
'ExternalHard-Drive'],
['DigitalCamera',
'LabTop',
'DeskTop',
'Printer',
'FlashDrive',
'MicrosoftOffice',
'LabTopCase',
'Anti-Virus',
'ExternalHard-Drive',
'Speakers'],
['LabTop',
'DeskTop',
'LabTopCase',
'ExternalHard-Drive',
'Speakers',
'Anti-Virus'],
['DigitalCamera',
'LabTop',
'LabTopCase',
'ExternalHard-Drive',
'Anti-Virus',
'Speakers'],
['DigitalCamera', 'Speakers'],
['DigitalCamera', 'DeskTop', 'Printer', 'FlashDrive', 'MicrosoftOffi
ce'],
['Printer',
'FlashDrive',
'MicrosoftOffice',
'Anti-Virus',
'LabTopCase',
'Speakers',
'ExternalHard-Drive'],
['DigitalCamera',
'FlashDrive',

```

```
'MicrosoftOffice',  
'Anti-Virus',  
'LabTopCase',  
'ExternalHard-Drive',  
'Speakers'],  
['DigitalCamera', 'LabTop', 'LabTopCase'],  
['DigitalCamera', 'LabTopCase', 'Speakers'],  
['DigitalCamera',  
 'LabTop',  
 'Printer',  
 'FlashDrive',  
 'MicrosoftOffice',  
 'Speakers',  
 'LabTopCase',  
 'Anti-Virus'],  
['DigitalCamera', 'LabTop', 'Speakers', 'Anti-Virus', 'LabTopCase']]
```

```
In [406]: TranEn = TransactionEncoder()  
TranEn_ary=TranEn.fit(new_list).transform(new_list)
```



```
In [407]: Dataframe = pd.DataFrame(TranEn_ary, columns=TranEn.columns_)
Dataframe
```

Out[407]:

	Anti-Virus	DeskTop	DigitalCamera	ExternalHard-Drive	FlashDrive	LabTop	LabTopCase	MicrosoftK
0	True	True	False	False	True	False	False	
1	True	False	False	False	True	True	True	
2	True	False	False	True	True	True	True	
3	True	False	False	True	True	True	True	
4	True	False	False	False	True	True	True	
5	False	False	False	False	True	True	False	
6	False	True	False	False	True	False	False	
7	True	False	False	True	False	True	False	
8	True	True	False	True	True	False	True	
9	True	True	True	True	True	True	True	
10	True	True	False	True	False	True	True	
11	True	False	True	True	False	True	True	
12	False	False	True	False	False	False	False	
13	False	True	True	False	True	False	False	
14	True	False	False	True	True	False	True	
15	True	False	True	True	True	False	True	
16	False	False	True	False	False	True	True	
17	False	False	True	False	False	False	True	
18	True	False	True	False	True	True	True	
19	True	False	True	False	False	True	True	

```
In [410]: frequentItemsets = apriori(Dataframe, min_support = 0.35, use_colnames=frequentItemsets)
```

Out[410]:

	support	itemsets
0	0.70	(Anti-Virus)
1	0.45	(DigitalCamera)
2	0.45	(ExternalHard-Drive)

3	0.65	(FlashDrive)
4	0.60	(LabTop)
5	0.70	(LabTopCase)
6	0.55	(MicrosoftOffice)
7	0.50	(Printer)
8	0.55	(Speakers)
9	0.45	(ExternalHard-Drive, Anti-Virus)
10	0.50	(FlashDrive, Anti-Virus)
11	0.50	(LabTop, Anti-Virus)
12	0.60	(LabTopCase, Anti-Virus)
13	0.40	(MicrosoftOffice, Anti-Virus)
14	0.35	(Anti-Virus, Printer)
15	0.45	(Anti-Virus, Speakers)
16	0.35	(LabTopCase, DigitalCamera)
17	0.35	(DigitalCamera, Speakers)
18	0.40	(LabTopCase, ExternalHard-Drive)
19	0.35	(LabTop, FlashDrive)
20	0.45	(LabTopCase, FlashDrive)
21	0.55	(FlashDrive, MicrosoftOffice)
22	0.50	(FlashDrive, Printer)
23	0.50	(LabTopCase, LabTop)
24	0.35	(LabTopCase, MicrosoftOffice)
25	0.45	(LabTopCase, Speakers)
26	0.45	(MicrosoftOffice, Printer)
27	0.40	(LabTopCase, ExternalHard-Drive, Anti-Virus)
28	0.45	(LabTopCase, FlashDrive, Anti-Virus)
29	0.40	(MicrosoftOffice, FlashDrive, Anti-Virus)
30	0.35	(FlashDrive, Anti-Virus, Printer)
31	0.45	(LabTopCase, LabTop, Anti-Virus)
32	0.35	(LabTopCase, MicrosoftOffice, Anti-Virus)
33	0.40	(LabTopCase, Anti-Virus, Speakers)

34 0.35 (LabTopCase, FlashDrive, MicrosoftOffice)

35 0.45 (FlashDrive, MicrosoftOffice, Printer)

36 0.35 (LabTopCase, MicrosoftOffice, FlashDrive, Anti...

In [411]: Rules = association_rules(frequentItemsets, metric="confidence", min_t
Rules

Out[411]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
0	(ExternalHard-Drive)	(Anti-Virus)	0.45	0.70	0.45	1.000000	1.428571
1	(FlashDrive)	(Anti-Virus)	0.65	0.70	0.50	0.769231	1.098901
2	(LabTop)	(Anti-Virus)	0.60	0.70	0.50	0.833333	1.190476
3	(LabTopCase)	(Anti-Virus)	0.70	0.70	0.60	0.857143	1.224490
4	(Anti-Virus)	(LabTopCase)	0.70	0.70	0.60	0.857143	1.224490
5	(Speakers)	(Anti-Virus)	0.55	0.70	0.45	0.818182	1.168831
6	(DigitalCamera)	(LabTopCase)	0.45	0.70	0.35	0.777778	1.111111
7	(DigitalCamera)	(Speakers)	0.45	0.55	0.35	0.777778	1.414141
8	(ExternalHard-Drive)	(LabTopCase)	0.45	0.70	0.40	0.888889	1.269841
9	(FlashDrive)	(MicrosoftOffice)	0.65	0.55	0.55	0.846154	1.538462
10	(MicrosoftOffice)	(FlashDrive)	0.55	0.65	0.55	1.000000	1.538462
11	(FlashDrive)	(Printer)	0.65	0.50	0.50	0.769231	1.538462
12	(Printer)	(FlashDrive)	0.50	0.65	0.50	1.000000	1.538462
13	(LabTop)	(LabTopCase)	0.60	0.70	0.50	0.833333	1.190476
14	(Speakers)	(LabTopCase)	0.55	0.70	0.45	0.818182	1.168831
15	(MicrosoftOffice)	(Printer)	0.55	0.50	0.45	0.818182	1.636364
16	(Printer)	(MicrosoftOffice)	0.50	0.55	0.45	0.900000	1.636364
17	(LabTopCase, ExternalHard-Drive)	(Anti-Virus)	0.40	0.70	0.40	1.000000	1.428571
18	(ExternalHard-Drive, Anti-Virus)	(LabTopCase)	0.45	0.70	0.40	0.888889	1.269841
19	(ExternalHard-Drive)	(LabTopCase, Anti-Virus)	0.45	0.60	0.40	0.888889	1.481481
20	(LabTopCase,	(Anti-Virus)	0.45	0.70	0.45	1.000000	1.428571

		FlashDrive)					
21	(LabTopCase, Anti-Virus)	(FlashDrive)	0.60	0.65	0.45	0.750000	1.153846
22	(FlashDrive, Anti-Virus)	(LabTopCase)	0.50	0.70	0.45	0.900000	1.285714
23	(Anti-Virus, MicrosoftOffice)	(FlashDrive)	0.40	0.65	0.40	1.000000	1.538462
24	(FlashDrive, Anti-Virus)	(MicrosoftOffice)	0.50	0.55	0.40	0.800000	1.454545
25	(Anti-Virus, Printer)	(FlashDrive)	0.35	0.65	0.35	1.000000	1.538462
26	(LabTopCase, LabTop)	(Anti-Virus)	0.50	0.70	0.45	0.900000	1.285714
27	(LabTopCase, Anti-Virus)	(LabTop)	0.60	0.60	0.45	0.750000	1.250000
28	(LabTop, Anti- Virus)	(LabTopCase)	0.50	0.70	0.45	0.900000	1.285714
29	(LabTop)	(LabTopCase, Anti-Virus)	0.60	0.60	0.45	0.750000	1.250000
30	(LabTopCase, MicrosoftOffice)	(Anti-Virus)	0.35	0.70	0.35	1.000000	1.428571
31	(Anti-Virus, MicrosoftOffice)	(LabTopCase)	0.40	0.70	0.35	0.875000	1.250000
32	(LabTopCase, Speakers)	(Anti-Virus)	0.45	0.70	0.40	0.888889	1.269841
33	(Anti-Virus, Speakers)	(LabTopCase)	0.45	0.70	0.40	0.888889	1.269841
34	(LabTopCase, FlashDrive)	(MicrosoftOffice)	0.45	0.55	0.35	0.777778	1.414141
35	(LabTopCase, MicrosoftOffice)	(FlashDrive)	0.35	0.65	0.35	1.000000	1.538462
36	(FlashDrive, MicrosoftOffice)	(Printer)	0.55	0.50	0.45	0.818182	1.636364
37	(FlashDrive, Printer)	(MicrosoftOffice)	0.50	0.55	0.45	0.900000	1.636364
38	(MicrosoftOffice, Printer)	(FlashDrive)	0.45	0.65	0.45	1.000000	1.538462
39	(MicrosoftOffice)	(FlashDrive, Printer)	0.55	0.50	0.45	0.818182	1.636364
40	(Printer)	(FlashDrive, MicrosoftOffice)	0.50	0.55	0.45	0.900000	1.636364
	(LabTopCase, Anti-Virus)	(FlashDrive)	0.60	0.65	0.45	0.750000	1.153846

41	FlashDrive, MicrosoftOffice)	(Anti-Virus)	0.35	0.70	0.35	1.000000	1.428571
42	(LabTopCase, Anti-Virus, MicrosoftOffice)	(FlashDrive)	0.35	0.65	0.35	1.000000	1.538462
43	(LabTopCase, FlashDrive, Anti- Virus)	(MicrosoftOffice)	0.45	0.55	0.35	0.777778	1.414141
44	(Anti-Virus, FlashDrive, MicrosoftOffice)	(LabTopCase)	0.40	0.70	0.35	0.875000	1.250000
45	(LabTopCase, MicrosoftOffice)	(FlashDrive, Anti-Virus)	0.35	0.50	0.35	1.000000	2.000000
46	(LabTopCase, FlashDrive)	(Anti-Virus, MicrosoftOffice)	0.45	0.40	0.35	0.777778	1.944444
47	(Anti-Virus, MicrosoftOffice)	(LabTopCase, FlashDrive)	0.40	0.45	0.35	0.875000	1.944444

Conclusion

In conclusion, for the Frist transaction, it was very noticable how faster Apriori (time: 0.09882998466491699 seconds) was compared to brute force method (Time: 0.4260389804840088 seconds). Also, by using an existing library, I was able to compare the results to my implementation and they both had the same number of Rules meaning that the implementation was working correctly. Also, by implementing the brute force method, I was able to see that the One, two, and three itemsets matched to ones from the apriori implementation meaning that the Brute Force even though it took more time, it returned the same results as the Apriori Algorithm

In []: