20BCE529

Naman Thaker

PRACTICAL 7

Data Mining

Aim : Identify the frequent patterns and generate strong association rule from the frequent pattern for the following data set(using Apriori or FP growth algorithm). Keep minimum 40% support and 40% confidence.

APRIORI ALGORITHM

1. Inputting the dataset

-> This code snippet will give the output as a list of given different items

```
data = [
        ['10', ['Beer', 'Nuts', 'Diaper']],
        ['20', ['Beer', 'Coffee', 'Diaper']],
        ['30', ['Beer', 'Diaper', 'Eggs']],
        ['40', ['Nuts', 'Eggs', 'Milk']],
        ['50', ['Nuts', 'Coffee', 'Diaper', 'Eggs', 'Milk']],
        ]

init = []
for i in data:
    for q in i[1]:
        if(q not in init):
            init.append(q)
init = sorted(init)
print(init)
```

```
     ['Beer', 'Coffee', 'Diaper', 'Eggs', 'Milk', 'Nuts']
```

2. Support Count

-> Here, we chose a support to be 50%

```
sp = 0.4 # 0.5
s = int(sp*len(init))
s
```

```
     2
```

3. Algorithm

-> Apriori Algorithm will be applied and the k-frequent itemsets are prnted as the output

```
from collections import Counter
c = Counter()
for i in init:
    for d in data:
        if(i in d[1]):
            c[i]+=1
print("C1:")
for i in c:
    print(str([i])+": "+str(c[i]))
print()
l = Counter()
for i in c:
    if(c[i] >= s):
        l[frozenset([i])]+=c[i]
print("L1:")
for i in l:
    print(str(list(i))+": "+str(l[i]))
print()
pl = l
pos = 1
for count in range (2,1000):
    nc = set()
    temp = list(l)
    for i in range(0,len(temp)):
        for j in range(i+1,len(temp)):
            t = temp[i].union(temp[j])
            if(len(t) == count):
                nc.add(temp[i].union(temp[j]))
    nc = list(nc)
    c = Counter()
    for i in nc:
        c[i] = 0
        for q in data:
            temp = set(q[1])
            if(i.issubset(temp)):
                c[i]+=1
    print("C"+str(count)+":")
    for i in c:
        print(str(list(i))+": "+str(c[i]))
    print()
    l = Counter()
    for i in c:
        if(c[i] >= s):
            l[i]+=c[i]
    print("L"+str(count)+":")
    for i in l:
        print(str(list(i))+": "+str(l[i]))
    print()
    if(len(l) == 0):
        break
    pl = l
    pos = count
print("Result: ")
print("L"+str(pos)+":")
for i in pl:
    print(str(list(i))+": "+str(pl[i]))
print()
```

```
['Nuts']: 3

L1:
['Beer']: 3
['Coffee']: 2
['Diaper']: 4
['Eggs']: 3
['Milk']: 2
['Nuts']: 3

C2:
['Diaper', 'Coffee']: 2
['Milk', 'Beer']: 0
['Eggs', 'Coffee']: 1
['Beer', 'Eggs']: 1
['Milk', 'Coffee']: 1
['Beer', 'Nuts']: 1
['Nuts', 'Coffee']: 1
['Milk', 'Nuts']: 2
['Beer', 'Coffee']: 1
['Milk', 'Diaper']: 1
['Milk', 'Eggs']: 2
['Nuts', 'Eggs']: 2
['Eggs', 'Diaper']: 2
['Nuts', 'Diaper']: 2
['Beer', 'Diaper']: 3

L2:
['Diaper', 'Coffee']: 2
['Milk', 'Nuts']: 2
['Milk', 'Eggs']: 2
['Nuts', 'Eggs']: 2
['Eggs', 'Diaper']: 2
['Nuts', 'Diaper']: 2
['Beer', 'Diaper']: 3

C3:
['Eggs', 'Diaper', 'Coffee']: 1
['Beer', 'Nuts', 'Diaper']: 1
['Beer', 'Eggs', 'Diaper']: 1
['Nuts', 'Diaper', 'Coffee']: 1
['Beer', 'Diaper', 'Coffee']: 1
['Milk', 'Eggs', 'Diaper']: 1
['Milk', 'Nuts', 'Diaper']: 1
['Nuts', 'Eggs', 'Diaper']: 1
['Milk', 'Nuts', 'Eggs']: 2

L3:
['Milk', 'Nuts', 'Eggs']: 2

C4:
```

```
L4:

Result:
L3:
['Milk', 'Nuts', 'Eggs']: 2
```

4. Finding the Association Rules

```
from itertools import combinations
for l in pl:
    c = [frozenset(q) for q in combinations(l,len(l)-1)]
    mmax = 0
    for a in c:
        b = l-a
        ab = l
        sab = 0
        sa = 0
        sb = 0
        for q in data:
            temp = set(q[1])
            if(a.issubset(temp)):
                sa+=1
            if(b.issubset(temp)):
                sb+=1
            if(ab.issubset(temp)):
                sab+=1
        temp = sab/sa*100
        if(temp > mmax):
            mmax = temp
        temp = sab/sb*100
        if(temp > mmax):
            mmax = temp
        print(str(list(a))+" -> "+str(list(b))+" = "+str(sab/sa*100)+"%")
        print(str(list(b))+" -> "+str(list(a))+" = "+str(sab/sb*100)+"%")
    curr = 1
    print("choosing:", end=' ')
    for a in c:
        b = l-a
        ab = l
        sab = 0
        sa = 0
        sb = 0
        for q in data:
            temp = set(q[1])
            if(a.issubset(temp)):
                sa+=1
            if(b.issubset(temp)):
                sb+=1
            if(ab.issubset(temp)):
                sab+=1
        temp = sab/sa*100
        if(temp == mmax):
            print(curr, end = ' ')
        curr += 1
        temp = sab/sb*100
        if(temp == mmax):
            print(curr, end = ' ')
        curr += 1
    print()
    print()
```

```
['Milk', 'Nuts'] -> ['Eggs'] = 100.0%
['Eggs'] -> ['Milk', 'Nuts'] = 66.66666666666666%
['Milk', 'Eggs'] -> ['Nuts'] = 100.0%
['Nuts'] -> ['Milk', 'Eggs'] = 66.66666666666666%
['Nuts', 'Eggs'] -> ['Milk'] = 100.0%
['Milk'] -> ['Nuts', 'Eggs'] = 100.0%
choosing: 1 3 5 6
```