

Experiment – 10

Student Name: Vivek Kumar

UID: 21BCS8129

Branch: BE-CSE(LEET)

Section/Group: WM-20BCS-616/A

Semester: 5th

Date of Performance: 9/11/2022

Subject Name: Machine Learning Lab

Subject Code: 20CSP-317

1. Aim/Overview of the practical:

Implement Association Rule Mining.

2. Task to be done/ Which logistics used:

Implement Association Rule mining.

3. Steps for experiment/practical/Code:

```
data = [  
    ['T100', ['I1', 'I2', 'I5']],  
    ['T200', ['I2', 'I4']],  
    ['T300', ['I2', 'I3']],  
    ['T400', ['I1', 'I2', 'I4']],  
    ['T500', ['I1', 'I3']],  
    ['T600', ['I2', 'I3']],  
    ['T700', ['I1', 'I3']],  
    ['T800', ['I1', 'I2', 'I3', 'I5']],  
    ['T900', ['I1', 'I2', 'I3']]  
]  
  
init = []  
for i in data:  
    for q in i[1]:  
        if(q not in init):  
            init.append(q)  
init = sorted(init)  
print(init)  
  
sp = 0.4  
s = int(sp*len(init))  
s  
  
from collections import Counter  
c = Counter()  
for i in init:  
    for d in data:  
        if(i in d[1]):  
            c[i]+=1
```

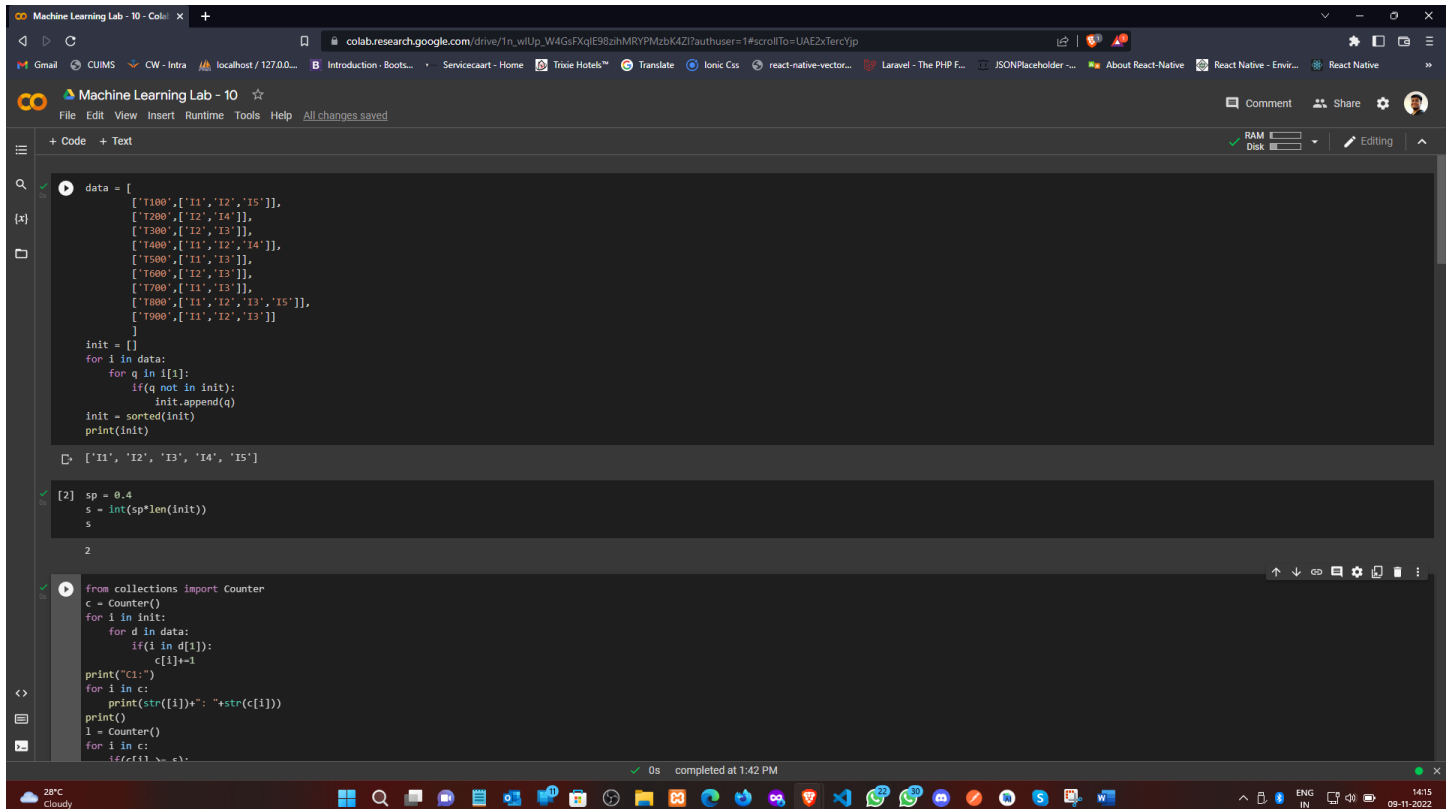
```
print("C1:")
for i in c:
    print(str([i])+"": "+str(c[i]))
print()
l = Counter()
for i in c:
    if(c[i] >= s):
        l[frozenset([i])]+=c[i]
print("L1:")
for i in l:
    print(str(list(i))+"": "+str(l[i]))
print()
pl = l
pos = 1
for count in range (2,1000):
    nc = set()
    temp = list(l)
    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()

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 = 1
        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 = 1
        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()
```

4. Result/Output/Writing Summary:



The screenshot shows a Google Colab notebook titled "Machine Learning Lab - 10". The code in the notebook is as follows:

```
data = [
    ['1100', ['11', '12', '15']],
    ['1200', ['12', '14']],
    ['1300', ['12', '13']],
    ['1400', ['11', '12', '14']],
    ['1500', ['11', '13']],
    ['1600', ['12', '13']],
    ['1700', ['11', '13']],
    ['1800', ['11', '12', '13', '15']],
    ['1900', ['11', '12', '13']]
]

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

The output of the first cell is:

```
['11', '12', '13', '14', '15']
```

The second cell contains the following code:

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

The output of the second cell is:

```
2
```

The third cell contains the following code:

```
from collections import Counter
c = Counter()
for i in init:
    for d in data:
        if(i in d[1]):
            c[i] += 1
print("c:")
for i in c:
    print(str(i) + ": " + str(c[i]))
print()
l = Counter()
for i in c:
    l[c[i]] += 1
```

The output of the third cell is:

```
c:
11: 3
12: 4
13: 4
14: 2
15: 2
```

The bottom status bar shows "0s completed at 1:42 PM".

```
Machine Learning Lab - 10 - Colab
colab.research.google.com/drive/1n_wlUp_W4GuFXqE96zhMRYPmzbK4Z7tauthuser=1#scrollTo=UAE2xTercYjp
Machine Learning Lab - 10
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
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()
p1 = 1
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[i])
            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
    p1 = 1
    pos = count
print("Result: ")
0s completed at 1:42 PM
```

```
Machine Learning Lab - 10 - Colab
colab.research.google.com/drive/1n_wlUp_W4GuFXqE96zhMRYPmzbK4Z7tauthuser=1#scrollTo=UAE2xTercYjp
Machine Learning Lab - 10
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
print("L" + str(pos) + ":")
for i in p1:
    print(str(list(i)) + ": " + str(p1[i]))
print()
C1:
['11']: 6
['12']: 7
['13']: 6
['14']: 2
['15']: 2
L1:
['11']: 6
['12']: 7
['13']: 6
['14']: 2
['15']: 2
C2:
['13', '14']: 0
['13', '15']: 1
['13', '12']: 4
['13', '11']: 4
['12', '11']: 4
['14', '15']: 0
['14', '14']: 1
['14', '15']: 2
['14', '12']: 2
['12', '15']: 2
L2:
['13', '12']: 4
['13', '11']: 4
['12', '11']: 4
['11', '15']: 2
['14', '12']: 2
['12', '15']: 2
C3:
['13', '12', '14']: 0
['13', '12', '15']: 1
['13', '11', '15']: 1
['14', '12', '15']: 0
['14', '12', '11']: 1
['12', '12', '11']: 2
0s completed at 1:42 PM
```

```
Machine Learning Lab - 10 - Colab
colab.research.google.com/drive/1n_wlUp_W4GsFXqE96zhMRYPmzbK4Z7authuser=1#scrollTo=UAE2xTercYjp

Machine Learning Lab - 10
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
['I4', 'I2', 'I1']: 1
['I3', 'I2', 'I1']: 2
['I2', 'I1', 'I5']: 2

L3:
['I3', 'I2', 'I1']: 2
['I2', 'I1', 'I5']: 2

C4:
['I1', 'I3', 'I2', 'I5']: 1

L4:

Result:
L3:
['I3', 'I2', 'I1']: 2
['I2', 'I1', 'I5']: 2

from itertools import combinations
for l in pl:
    c = [frozenset(q) for q in combinations(1,len(l)-1)]
    mmax = 0
    for a in c:
        b = 1-a
        ab = 1
        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
```

0s completed at 1:42 PM

```
Machine Learning Lab - 10 - Colab
colab.research.google.com/drive/1n_wlUp_W4GsFXqE96zhMRYPmzbK4Z7authuser=1#scrollTo=UAE2xTercYjp

Machine Learning Lab - 10
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
[4] 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 = 1-a
    ab = 1
    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()

['I3', 'I2'] -> ['I1'] = 50.0%
['I1'] -> ['I3', 'I2'] = 33.33333333333333%
['I3', 'I1'] -> ['I2'] = 50.0%
['I2'] -> ['I3', 'I1'] = 28.57142857142857%
['I2', 'I1'] -> ['I3'] = 50.0%
['I3'] -> ['I2', 'I1'] = 33.33333333333333%
choosing: 1 3 5

['I2', 'I1'] -> ['I5'] = 50.0%
['I5'] -> ['I2', 'I1'] = 100.0%
['I2', 'I5'] -> ['I1'] = 100.0%
['I1'] -> ['I2', 'I5'] = 33.33333333333333%
['I1', 'I5'] -> ['I2'] = 100.0%
['I2'] -> ['I1', 'I5'] = 28.57142857142857%
choosing: 2 3 5
```

0s completed at 1:42 PM

Learning outcomes (What I have learnt):

1. Understood the concept of Association Rule Mining.
2. Learnt how to find the Association rule for the Subset.
3. Learnt the concept of Apriori.
4. Completed the Worksheet Successfully.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			