Experiment No 9

Aim: Implementation of Association Rule Mining algorithm (Apriori)

Theory:

Apriori algorithm refers to the algorithm which is used to calculate the association rules between objects. It means how two or more objects are related to one another. In other words, we can say that the apriori algorithm is an association rule leaning that analyzes that people who bought product A also bought product B.

Apriori algorithm refers to an algorithm that is used in mining frequent products sets and relevant association rules. Generally, the apriori algorithm operates on a database containing a huge number of transactions. For example, the items customers but at a Big Bazar.

Apriori algorithm helps the customers to buy their products with ease and increases the sales performance of the particular store.

Components of the Apriori algorithm

The given three components comprise the apriori algorithm.

Support

Support refers to the default popularity of any product. You find the support as a quotient of the division of the number of transactions comprising that product by the total number of transactions. Hence, we get

Support (Biscuits) = (Transactions relating to biscuits) / (Total transactions)

= 400/4000 = 10 percent.

Confidence

Confidence refers to the possibility that the customers bought both biscuits and chocolates together. So, you need to divide the number of transactions that comprise both biscuits and chocolates by the total number of transactions to get confidence.

Hence,

Confidence = (Transactions relating to both biscuits and Chocolate) / (Total transactions involving Biscuits)

= 200/400

= 50 percent.

It means that 50 per cent of customers who bought biscuits bought chocolates also.

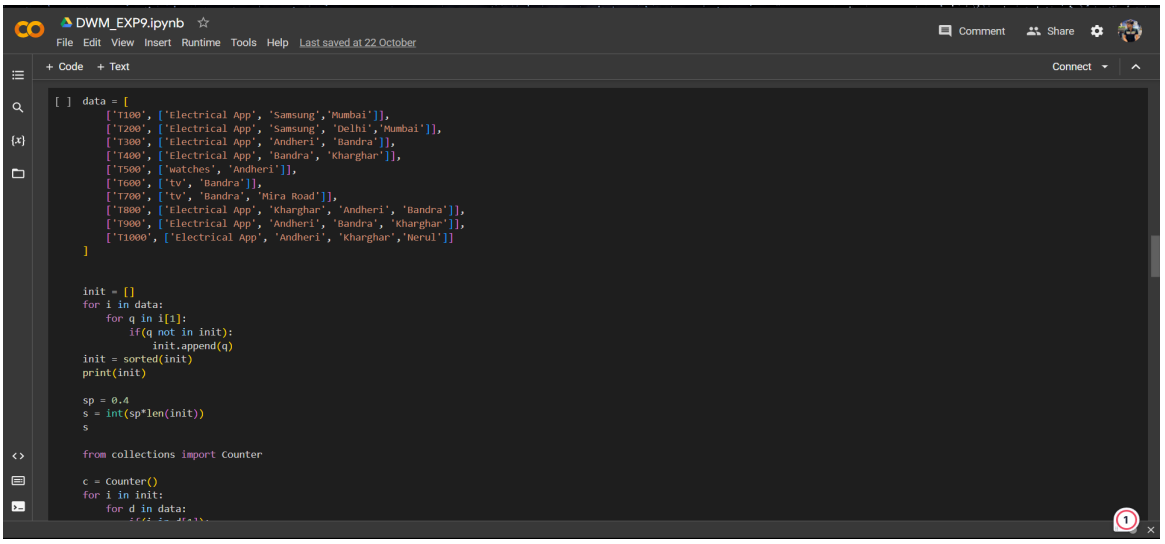
Lift

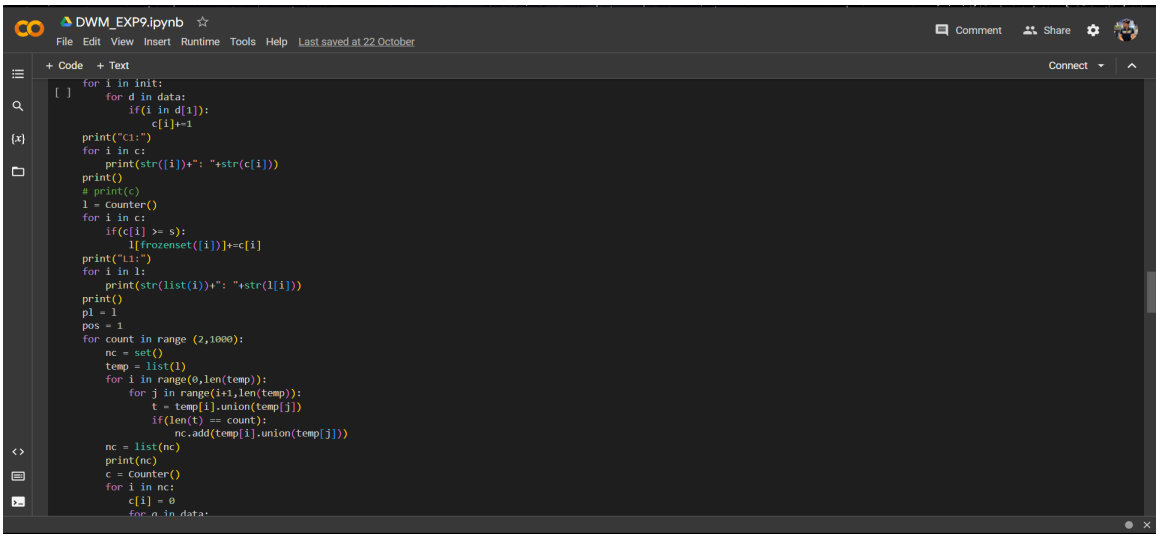
Consider the above example; lift refers to the increase in the ratio of the sale of chocolates when you sell biscuits. The mathematical equations of lift are given below.

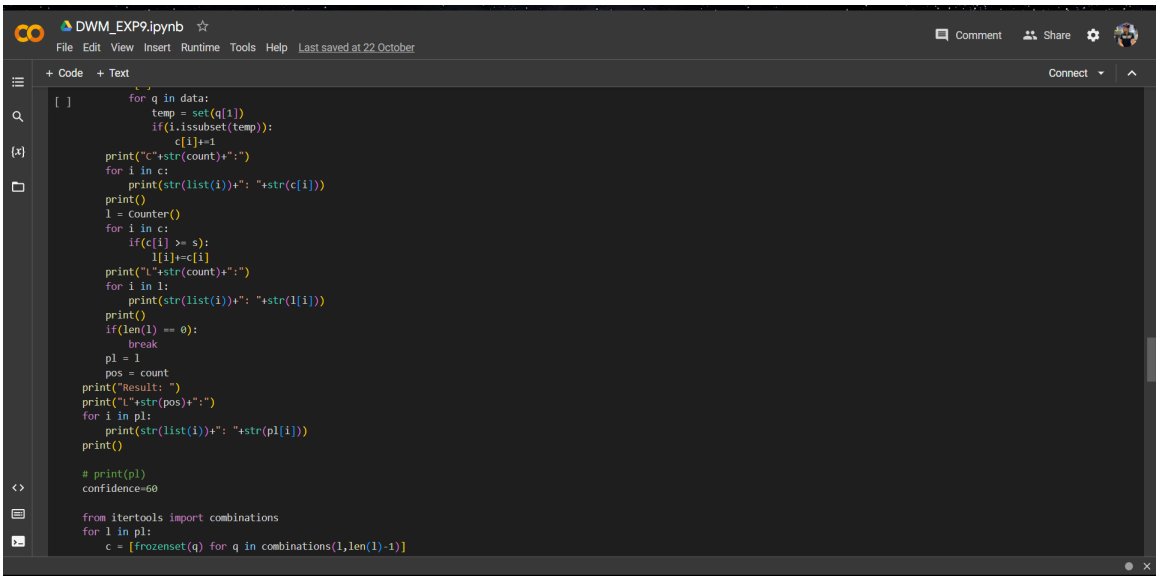
Lift = (Confidence (Biscuits - chocolates)/ (Support (Biscuits)

= 50/10 = 5

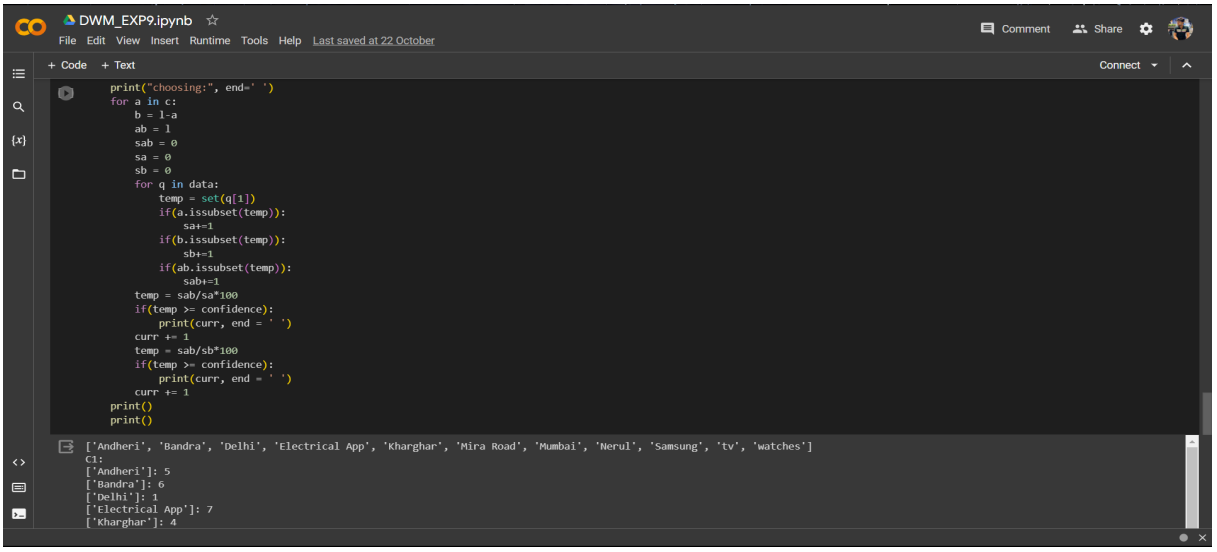
Output:

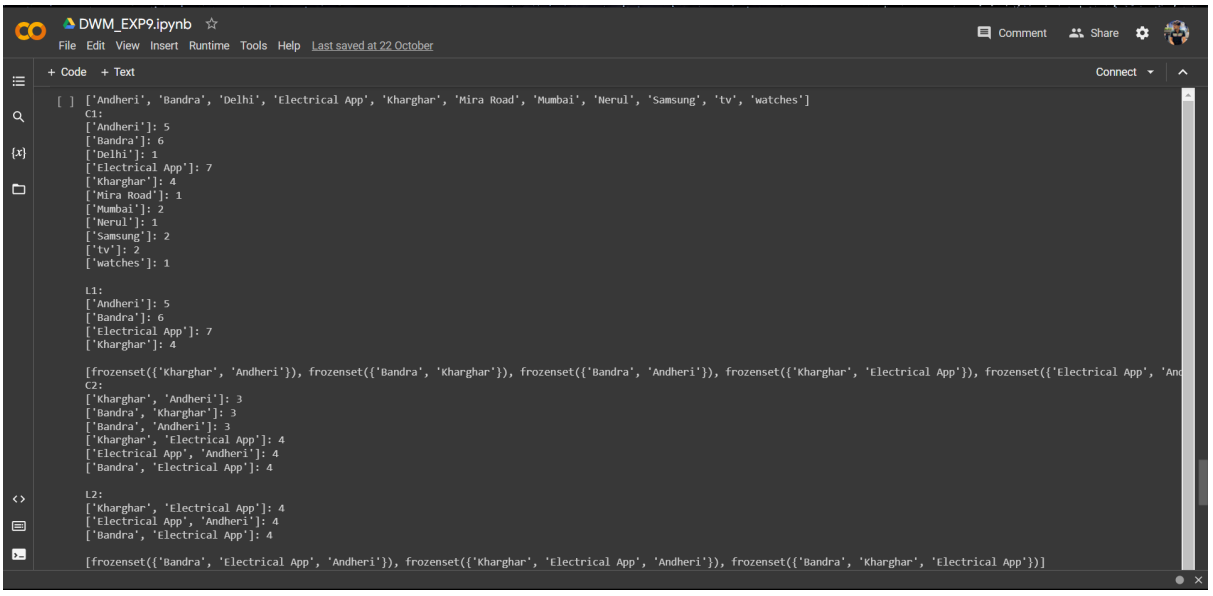


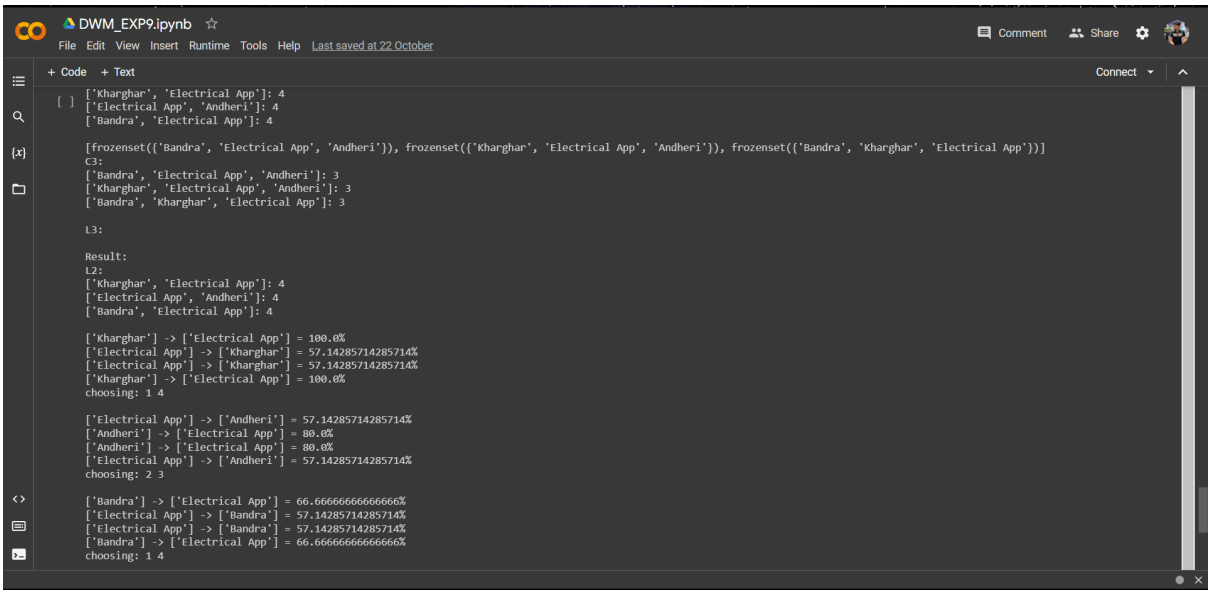

DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment Share
+ Code + Text
Connect
[ ]
for i in init:
for d in data:
if(i in d[1]):
c[i]+=1
print("C1:")
for i in c:
print(str([i])+": "+str(c[i]))
print()
# print(c)
l = Counter()
for i in c:
if(c[i] >= s):
l[frozenset([i])]+=c[i]
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)
print(nc)
c = Counter()
for i in nc:
c[i] = 0


DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment Share
+ Code + Text
Connect
[ ]
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()
# print(pl)
confidence=60
from itertools import combinations
for l in pl:
c = [frozenset(q) for q in combinations(l,len(l)-1)]


DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment Share
+ Code + Text
Connect
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 > confidence):
mmax = temp
temp = sab/sb*100
if(temp > confidence):
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


DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment
Share
+ Code + Text
Connect
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 >= confidence):
print(curr, end = ' ')
curr += 1
temp = sab/sb*100
if(temp >= confidence):
print(curr, end = ' ')
curr += 1
print()
print()
['Andheri', 'Bandra', 'Delhi', 'Electrical App', 'Kharghar', 'Mira Road', 'Mumbai', 'Nerul', 'Samsung', 'tv', 'watches']
C1:
['Andheri']: 5
['Bandra']: 6
['Delhi']: 1
['Electrical App']: 7
['Kharghar']: 4


DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment
Share
+ Code + Text
Connect
[ ] ['Andheri', 'Bandra', 'Delhi', 'Electrical App', 'Kharghar', 'Mira Road', 'Mumbai', 'Nerul', 'Samsung', 'tv', 'watches']
C1:
['Andheri']: 5
['Bandra']: 6
['Delhi']: 1
['Electrical App']: 7
['Kharghar']: 4
['Mira Road']: 1
['Mumbai']: 2
['Nerul']: 1
['Samsung']: 2
['tv']: 2
['watches']: 1
L1:
['Andheri']: 5
['Bandra']: 6
['Electrical App']: 7
['Kharghar']: 4
[frozenset({'Kharghar', 'Andheri'}), frozenset({'Bandra', 'Kharghar'}), frozenset({'Bandra', 'Andheri'}), frozenset({'Kharghar', 'Electrical App'}), frozenset({'Electrical App', 'And
C2:
['Kharghar', 'Andheri']: 3
['Bandra', 'Kharghar']: 3
['Bandra', 'Andheri']: 3
['Kharghar', 'Electrical App']: 4
['Electrical App', 'Andheri']: 4
['Bandra', 'Electrical App']: 4
L2:
['Kharghar', 'Electrical App']: 4
['Electrical App', 'Andheri']: 4
['Bandra', 'Electrical App']: 4
[frozenset({'Bandra', 'Electrical App', 'Andheri'}), frozenset({'Kharghar', 'Electrical App', 'Andheri'}), frozenset({'Bandra', 'Kharghar', 'Electrical App'})]


DWM_EXP9.ipynb
File Edit View Insert Runtime Tools Help Last saved at 22 October
Comment
Share
+ Code + Text
Connect
[ ] ['Kharghar', 'Electrical App']: 4
['Electrical App', 'Andheri']: 4
['Bandra', 'Electrical App']: 4
[frozenset({'Bandra', 'Electrical App', 'Andheri'}), frozenset({'Kharghar', 'Electrical App', 'Andheri'}), frozenset({'Bandra', 'Kharghar', 'Electrical App'})]
C3:
['Bandra', 'Electrical App', 'Andheri']: 3
['Kharghar', 'Electrical App', 'Andheri']: 3
['Bandra', 'Kharghar', 'Electrical App']: 3
L3:
Result:
L2:
['Kharghar', 'Electrical App']: 4
['Electrical App', 'Andheri']: 4
['Bandra', 'Electrical App']: 4
['Kharghar'] -> ['Electrical App'] = 100.0%
['Electrical App'] -> ['Kharghar'] = 57.14285714285714%
['Electrical App'] -> ['Kharghar'] = 57.14285714285714%
['Kharghar'] -> ['Electrical App'] = 100.0%
choosing: 1 4
['Electrical App'] -> ['Andheri'] = 57.14285714285714%
['Andheri'] -> ['Electrical App'] = 80.0%
['Andheri'] -> ['Electrical App'] = 80.0%
['Electrical App'] -> ['Andheri'] = 57.14285714285714%
choosing: 2 3
['Bandra'] -> ['Electrical App'] = 66.66666666666666%
['Electrical App'] -> ['Bandra'] = 57.14285714285714%
['Electrical App'] -> ['Bandra'] = 57.14285714285714%
['Bandra'] -> ['Electrical App'] = 66.66666666666666%
choosing: 1 4