Web Ming Lab Assignment

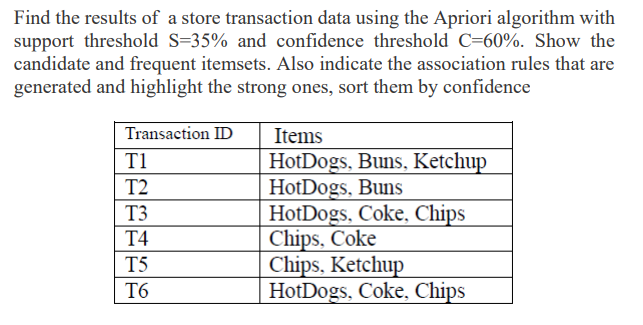
Apriori

## Name: Om Ashish Mishra

## Registration Number: 16BCE0789

## Slot: F2

# The Question



# The Code:

sp\_thr=int(input("Please Enter the support threshold percentage"))

terms=int(input("Enter the number of terms"))

min\_sp=sp\_thr\*6/100

print("Minimum Support = ",min\_sp)

ms=min\_sp

from itertools import combinations

items = [["HotDogs", "Buns", "Ketchup"],["HotDogs", "Buns"],["HotDogs", "Coke", "Chips"],["Chips, Coke"],["Chips", "Ketchup"],["HotDogs", "Coke", "Chips"]]

s = []

def apriori(lname, paramater):

second = list(combinations(lname, paramater))

# print(second)

slist = []

for l in items:

st = list(combinations(l, paramater))

slist.append(st)

# print(slist)

qlist = []

for p in slist:

for q in p:

qlist.append(q)

# print(qlist)

scounts = {}

for i in qlist:

if i in second:

if i in scounts:

scounts[i] = scounts[i] + 1

else:

scounts[i] = 1

# print(scounts)

mv = ms

for k in scounts:

if scounts[k] < mv:

mv = scounts[k]

for k in scounts.copy():

if scounts[k] == mv:

scounts.pop(k)

#print(scounts)

return scounts

#first

for i in items:

for j in i:

s.append(j)

counts = {}

for i in s:

if i in counts:

counts[i] = counts[i] + 1

else:

counts[i] = 1

result = sorted(counts.items(), key=lambda t: t[0], reverse=False)

new\_dict = {}

for k, v in result:

new\_dict[k] = v

m = ms

for k in new\_dict:

if new\_dict[k] > ms:

m = new\_dict[k]

for k in new\_dict.copy():

if new\_dict[k]<ms:

new\_dict.pop(k)

print(new\_dict)

# second

nitem = []

for k in new\_dict:

nitem.append(k)

support2=apriori(nitem,2)

print(support2)

# third

support3 = apriori(nitem,3)

print(support3)

# The Output:

