

In [2]:

```
1 import time as tt
2 t0=tt.time()
3 dats=[['google','amazon'],['amazon','google','python','cse'],['cse','gc
4       ['cse','amazon','python','google'],['amazon','google','cse','data'
5 sf=[]
6 from itertools import combinations
7 import pandas as pd
8 def pruned(it,k,l):
9     subs=combinations(sorted(it),k)
10    for itm in subs:
11        if itm not in l:
12            return False
13    return True
14 def apriori_gen(c,data,minsp,k,l):
15     C=[]
16     for i in range(0,len(c)):
17         itm=[]
18         for j in range(i+1,len(c)):
19             if c[i][0:(k-1)]==c[j][0:(k-1)]:
20                 if type(c[i][0:(k-1)])!=tuple:
21                     itm=(c[i][0:(k-1)]+(c[i][(k-1)])+(c[j][(k-1)]))
22                 else:
23                     itm=c[i][0:(k-1)]+(c[i][(k-1)],)+(c[j][(k-1)],)
24                 itm=tuple(sorted(itm))
25                 if pruned(itm,k,l)!=False and itm not in C:
26                     C.append(itm)
27
28     return C
29 def apriori(data,minsp):
30     global sf
31     kdc={}
32     fi,sfi=[],[]
33     for dat in data:
34         for i in range(0,len(dat)):
35             if dat[i] not in kdc:
36                 kdc[dat[i]]=1
37             elif dat[i] in dat[0:i]:
38                 continue
39             else:
40                 kdc[dat[i]]+=1
41     for k,v in kdc.items():
42         if v>=minsp:
43             fi.append((k,v))
44             sfi.append(k)
45     sfi=sorted(sfi)
46     sf=fi
47     for i in range(len(data)):
48         data[i]=tuple(set(sfi).intersection(set(data[i])))
49     k=2
50     l=[]
51     c=[]
52     c=combinations(sfi,2)
53     for it in c:
54         l.append(it)
55     c=l
56     dic={}
57     while len(l)!=0:
58         if k>2:
59             c=apriori_gen(l,dats,minsp,k-1,l)
60             for it in data:
61                 sub=combinations(it,k)
```

```

62         subs=[]
63         for itms in sub:
64             subs.append(tuple(sorted(itms)))
65         for itm in c:
66             itm=tuple(sorted(itm))
67             if itm in subs:
68                 #print(itm,"----",subs)
69                 if itm not in dic:
70                     dic[itm]=1
71                 else:
72                     dic[itm]+=1
73     l=[]
74     for key,v in dic.items():
75         if v>=minsp:
76             fi.append((key,v))
77             l.append(key)
78     dic={}
79     k=k+1
80     freq=fi
81     data=pd.DataFrame(columns=["items","support"])
82     its=[]
83     sps=[]
84     for i in range(0,len(freq)):
85         its.append(str(freq[i][0]))
86         sps.append(freq[i][1])
87     data["items"]=its
88     data["support"]=sps
89     return data
90 import requests as rq
91 res=rq.get(r"https://raw.githubusercontent.com/stedy/Machine-Learning-wi
92 datass=[]
93 for it in res.iter_lines(1000000):
94     li=[]
95     for itm in it.split():
96         li.append(itm)
97     datass.append(li)
98 dii=datass[0:100]
99 print(apriori(dats,2).sort_values(ascending=False,by="support"))

```

	items	support
0	google	5
1	amazon	5
3	cse	4
4	('amazon', 'google')	4
7	('cse', 'google')	4
2	python	3
5	('amazon', 'cse')	3
6	('amazon', 'python')	3
10	('amazon', 'cse', 'google')	3
8	('cse', 'python')	2
9	('google', 'python')	2
11	('amazon', 'google', 'python')	2
12	('amazon', 'cse', 'python')	2
13	('cse', 'google', 'python')	2
14	('amazon', 'cse', 'google', 'python')	2

In [6]:

1