In [1]:
```
from itertools import combinations #importamos  la funcion para hacer combinacior
```

In [2]:
```
lista=["a","b","c","d","e"]
```

In [3]:
```
#Creamos las listas que usaremos
com2=[] #Lista vacia, aqui iran las combinaciones para k=2
com3=[] #Lista vacia, aqui iran las combinaciones para k=3
contador1=[] #lista de casos ocurridos de cada combinacion de k=1
contador2=[] #Lista de casos ocurridos de cada combinacion de k=2
contador3=[] #Lista de casos ocurridos de cada combinacion de k=3
ap1=[] #almacena a los conjuntos con soporte mayor a 0.5 (k=1)
ap2=[] #(k=2)
ap3=[] #(k=3)
```

In [4]:
```
tabla=[["a","b","c","e"],["b","e"],["c","d","e"],["a","c","d"],["a","c","e"]] #es
```

In [5]:
```
for a in combinations(lista,2):
    com2.append(a) #con append le agregamos a la lista la combinacion de letras c
com2 #se imprimen la lista con las combinaciones de 2 letras (Deben ser 10 combir
```

Out[5]:
```
[('a', 'b'),
 ('a', 'c'),
 ('a', 'd'),
 ('a', 'e'),
 ('b', 'c'),
 ('b', 'd'),
 ('b', 'e'),
 ('c', 'd'),
 ('c', 'e'),
 ('d', 'e')]
```

In [6]:
```
for a in combinations(lista,3):
    com3.append(a) #con append le agregamos a la lista la combinacion de letras c
com3 #se imprime la lista con las combinaciones de 3 letras (deben ser 10 combina
```

Out[6]:
```
[('a', 'b', 'c'),
 ('a', 'b', 'd'),
 ('a', 'b', 'e'),
 ('a', 'c', 'd'),
 ('a', 'c', 'e'),
 ('a', 'd', 'e'),
 ('b', 'c', 'd'),
 ('b', 'c', 'e'),
 ('b', 'd', 'e'),
 ('c', 'd', 'e')]
```

In [7]:
```
for i in range(0,len(lista)):
    contador1.append(0)

for x in range(0,len(com2)):
    contador2.append(0)

for y in range(0,len(com3)):
    contador3.append(0)
```

In [8]:
```
contador1 #verificamos que contenga 5 veces el 0
```

Out[8]:
```
[0, 0, 0, 0, 0]
```

In [9]:
```
contador2 #verificamos que contega 10 veces el 0
```

Out[9]:
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [10]:
```
 contador3 #verificamos que contenga 10 veces el 0
```

Out[10]:
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

In [11]:
```
for x in range(0,len(tabla)):
    for i in range(0,len(tabla[x])):
            for j in range(0,len(lista)):
                    if lista[j]==tabla[x][i]:
                        contador1[j]=contador1[j]+1

print(contador1)
```

```
[3, 2, 4, 2, 4]
```

In [12]:
```
pasa1=0 #verificamos cuantos itemset pasan encima del soporte minimo de 0.5
for i in range(0,len(contador1)):
    if (contador1[i]/5)>0.5:
        ap1.append([lista[i],contador1[i]/5]) #almacenamos el itemset y su soport
        pasa1=pasa1+1

pasa1 #al correrlo podemos ver que solo 3 elementos estan por encima del soporte
```

Out[12]:
```
3
```

In [13]:
```
ap1
```

Out[13]:
```
[['a', 0.6], ['c', 0.8], ['e', 0.8]]
```

In [14]:
```
for x in range(0,len(com2)): #Nos movemos primero entre las combinaciones de letr
    acum1=0 #acumulador que nos ayudará
    acum2=0 #acumulador que nos ayudará
    for i in range(0,len(tabla)): #entramos en la tabla
        for j in range(0,len(tabla[i])): #nos movemos dentro del conjunto i de la
            if tabla[i][j]==com2[x][0]: #Si el elemento j del conjunto i es igual
                acum1=acum1+1 #se acumula un 1 en acum1
            if tabla[i][j]==com2[x][1]: #Si el elemento j del conjunto i es igual
                acum2=acum2+1 #se acumula un 1 en acum2
        if acum1==1 and acum2==1: #si los dos acumuladores tienen valor de 1 entc
            contador2[x]=contador2[x]+1 #acumulamos su ocurrencia
        acum1=0 #los reiniciamos para verificar con el siguiente conjunto de la t
        acum2=0

contador2 #imprime el contador de ocurrencias
```

Out[14]:
```
[1, 3, 1, 2, 1, 0, 2, 2, 3, 1]
```

In [15]:
```
for i in range(0,len(contador2)):
    if contador2[i]/5>0.5:
        ap2.append([com2[i],contador2[i]/5])
ap2 #verificamos que haya elementos con soporte mayor a 0.5 y si es asi pasamos c
```

Out[15]:
```
[[('a', 'c'), 0.6], [('c', 'e'), 0.6]]
```

In [16]:
```
for x in range(0,len(com3)):
    acum1=0
    acum2=0
    acum3=0
    for i in range(0,len(tabla)):
        for j in range(0,len(tabla[i])):
            if tabla[i][j]==com3[x][0]:
                acum1=acum1+1
            if tabla[i][j]==com3[x][1]:
                acum2=acum2+1
            if tabla[i][j]==com3[x][2]:
                acum3=acum3+1
        if acum1+acum2+acum3==3:
            contador3[x]=contador3[x]+1
        acum1=0
        acum2=0
        acum3=0

contador3
```

Out[16]:
```
[1, 0, 1, 1, 2, 0, 0, 1, 0, 1]
```

In [17]:
```
pasa=0
for i in range(0,len(contador3)):
    if contador3[i]/5>0.5:
        ap3.append([com3[i],contador[i]/5])
        pasa=pasa+1

if pasa==0:
    print("Ningun itemset tiene un soporte mayor a 0.5")
```

```
Ningun itemset tiene un soporte mayor a 0.5
```

In [18]:
```
print("Dado a que no existe ningun itemset para k=3 con un soporte mayor o igual
```

```
Dado a que no existe ningun itemset para k=3 con un soporte mayor o igual a 0.
5, entonces el nivel maximo de soporte es k=2
```

In [ ]: