

Naive-Bayes

1. Agregar un valor en cada clase
2. N = número de grupos B_i
3. Calcular soporte de los valores que vamos a usar: $SOP(B_i C)$:

```
for(int i= 0; i< n ; i++){  
    sopBC[i] =(SELECT COUNT (*) FROM <table> WHERE Bi=1 & C=1)/(SELECT COUNT * FROM <table>)  
}
```

4. Calcular el Soporte de C(lo que queremos que pase):

```
sopC =(SELECT COUNT (*) FROM <table> WHERE C=1)/(SELECT COUNT * FROM <table>)
```

5. Calcular las probabilidades condicionales de cada B_i dado C:

```
for(int i = 0; i< n; i++){  
    CondB[i] = sopBC[i]/sopC  
}
```

6. Obtenemos el valor que queremos verificar de que clase es

Nuevo = dato a verificar

7. Calculmos su Probabilidad para cada Clase C

```
For(int l = 0; l < m; i++){  
    For(int j = 0; j<n; i++){  
        If(nuevo cumple Bi)  
            PCN[i][j] = CondB[i][j]  
        Else  
            PCN[i][j]=1  
    }  
    PC[i]=1  
    For(int j = 0; j<n; i++){  
        PC[i] = PC[i]*PCN[i][j]  
    }
```

8. Vemos cual es el más probable

```
Mp=-1  
Pmax=0  
For(int i =0;i<m;i++){  
    If(PC[i]>pmax)  
        Pmax = PC[i]  
        MP = i  
}  
Print: Nuevo es de clase i
```

A priori

Apriori(T, ϵ)

$L_1 \leftarrow \{\text{large 1-itemsets}\}$

$k \leftarrow 2$

while $L_{k-1} \neq \emptyset$

$C_k \leftarrow \{a \cup \{b\} \mid a \in L_{k-1} \wedge b \notin a\} - \{c \mid \{s \mid s \subseteq c \wedge |s| = k-1\} \not\subseteq L_{k-1}\}$

for transactions $t \in T$

$C_t \leftarrow \{c \mid c \in C_k \wedge c \subseteq t\}$

for candidates $c \in C_t$

$count[c] \leftarrow count[c] + 1$

$L_k \leftarrow \{c \mid c \in C_k \wedge count[c] \geq \epsilon\}$

$k \leftarrow k + 1$

return $\bigcup_k L_k$

1. Calculo el Soporte de A (que mi tupla cumpla algo)
2. Quito de mi lista los A que su soporte es menor a SopMin
3. Calculo Soporte de Todos los $A_i A_j$ donde A_i y A_j siguen en la lista
4. Quito todos los que no cumplan SopMin
5. Sigo hasta no agregar nada