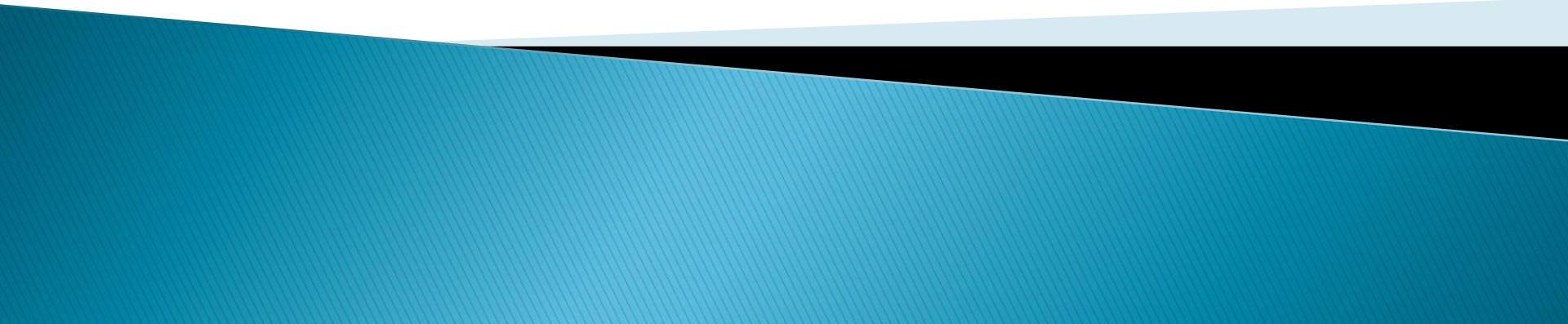


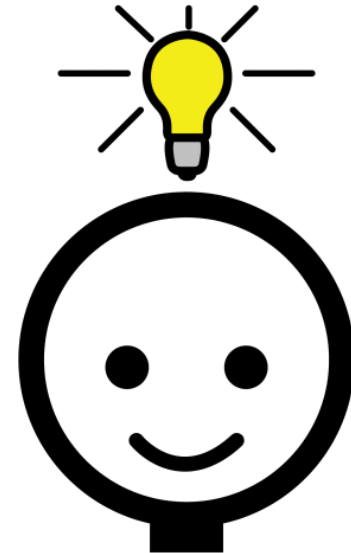
# Sistemas de recomendación

Joaquín Pineda Gutiérrez



# Algoritmo a priori

- ▶ Se encarga de buscar reglas que nos permiten realizar recomendaciones.
- ▶ Las reglas nos dicen que frecuente es un hecho a partir de otro.
- ▶ Se basa en la idea “Los clientes que compraron X también compraron Y”.



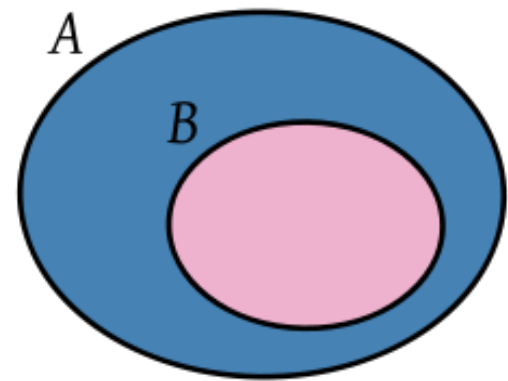
# Conjunto de datos

- ▶ Para la practica usamos un conjunto de datos que representan transacciones de un supermercado (retail.dat).
- ▶ Cada línea del archivo es una transacción (una compra) y cada numero un producto.
- ▶ También usamos un conjunto mucho menor de pruebas.



# Conjuntos frecuentes

- ▶ Para generar las reglas primero tenemos que localizar los conjuntos de objetos que aparecen juntos frecuentemente.
- ▶ Un conjunto frecuente está formado por subconjuntos que también son frecuentes, pero no tiene por qué cumplirse al revés.
- ▶ Los filtramos según el soporte objetivo que queramos



# Creación de reglas

- ▶ Cuando tenemos los conjuntos frecuentes buscamos reglas de asociación en los mismos partiendo el conjunto en dos grupos.
- ▶ Buscamos la confianza y el soporte de dicha regla.
- ▶ Una regla se considera buena si tiene un soporte y una confianza altos.
- ▶ Como no se indica nada, se obtienen todas las reglas para todos los tamaños de  $k$

- $\{2, 3\} \rightarrow \{5\}$  ( $s = 2/4, c = 2/2$ )
- $\{3, 5\} \rightarrow \{2\}$  ( $s = 2/4, c = 2/2$ )
- $\{2, 5\} \rightarrow \{3\}$  ( $s = 2/4, c = 2/3$ )

# Ejemplos

- ▶ Conjunto de test:
  - Soporte de al menos 50% de las transacciones
  - Confianza de al menos 50%

```
[[1], [3], [2], [5], [1, 3], [2, 3], [3, 5], [2, 5], [2, 3, 5]]
[3] -> 1 (s=0.5, c=0.6666666666666666)
[1] -> 3 (s=0.5, c=1.0)
[3] -> 2 (s=0.5, c=0.6666666666666666)
[2] -> 3 (s=0.5, c=0.6666666666666666)
[5] -> 3 (s=0.5, c=0.6666666666666666)
[3] -> 5 (s=0.5, c=0.6666666666666666)
[5] -> 2 (s=0.75, c=1.0)
[2] -> 5 (s=0.75, c=1.0)
[3, 5] -> 2 (s=0.5, c=1.0)
[2, 5] -> 3 (s=0.5, c=0.6666666666666666)
[2, 3] -> 5 (s=0.5, c=1.0)
```

# Conclusiones test.dat

- ▶ Al ser un conjunto muy pequeño el soporte y la confianza es muy alto en todos los casos.
- ▶ Podemos observar perfectamente como las reglas no son conmutativas.
  - Que se compre 2 y 3 garantiza que se va a comprar 5 al 100% de confianza
  - Que se compre 2 y 5 no tiene la misma garantía, es solo del 66,6%

`[2, 5] -> 3 (s=0.5, c=0.6666666666666666)`

`[2, 3] -> 5 (s=0.5, c=1.0)`

# Ejemplos

- ▶ Conjunto retail.dat:
  - Soporte de al menos el 7,9% de las transacciones.
  - Confianza de al menos el 50%.

```
[[32], [38], [39], [41], [48], [32, 39], [32, 48], [38, 39], [38, 48], [39, 41], [39, 48], [41, 48], [39, 41, 48]]
[32] -> 39 (s=0.09590299675597196, c=0.5574602755983386)
[32] -> 48 (s=0.0911276967400921, c=0.5297026438979363)
[38] -> 39 (s=0.1173408044282117, c=0.6633111054116441)
[38] -> 48 (s=0.09010684875569973, c=0.5093613747114645)
[41] -> 39 (s=0.12946620993171662, c=0.7637336901973905)
[48] -> 39 (s=0.33055057734624893, c=0.6916340334638661)
[39] -> 48 (s=0.33055057734624893, c=0.5750764676862358)
[41] -> 48 (s=0.10228896803611533, c=0.603412512546002)
[41, 48] -> 39 (s=0.0835507361448243, c=0.8168108227988468)
[39, 41] -> 48 (s=0.0835507361448243, c=0.6453478184685474)
```



# Conclusiones retail.dat

- ▶ Los objetos 39 y 48 aparecen juntos en muchas de las transacciones (muchas más que el resto)
  - ▶ La mayoría de conjuntos frecuentes no llega a estar presentes en más del 5% de las transacciones.
- 