Conceptos y Aplicaciones de Big Data

STREAM PROCESSING

Prof. Waldo Hasperué
whasperue@lidi.info.unlp.edu.ar

Temario

¿Qué es un flujo de datos?

Procesamiento de flujo de datos

Creación de modelos predictores

Flujo de datos

El flujo de datos es continuo

- La frecuencia de la llegada de los datos depende del problema

Los datos son recolectados en tiempo real

No se almacenan para entrenar el modelo

Flujos de datos

- Redes Sociales: Twitter, Facebook, Instagram.
- Flujos de transacciones: Bancarias o criptomonedas (Bitcoin).
- Monitoreo de redes: Detección de intrusiones en la red, logs de servidores.
- Monitoreo en tiempo real de sensores + Internet de las Cosas (IoT).
- Análisis climático
- Análisis de información generada por dispositivos wearable.

Estrategias para el tratamiento del flujo

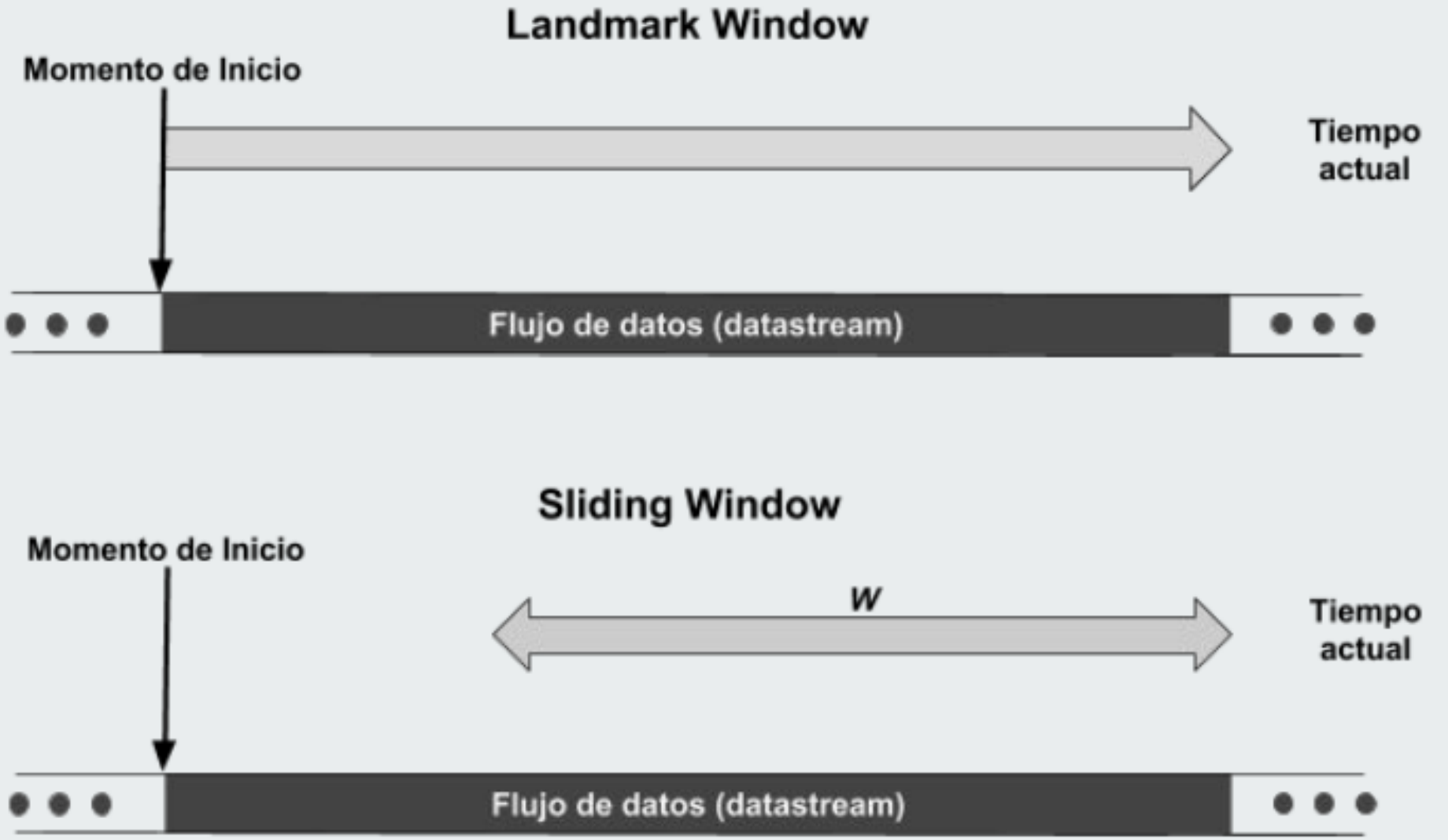
El dato se recibe, se utiliza y se descarta

Ventana temporal para guardar los últimos *n* datos recibidos

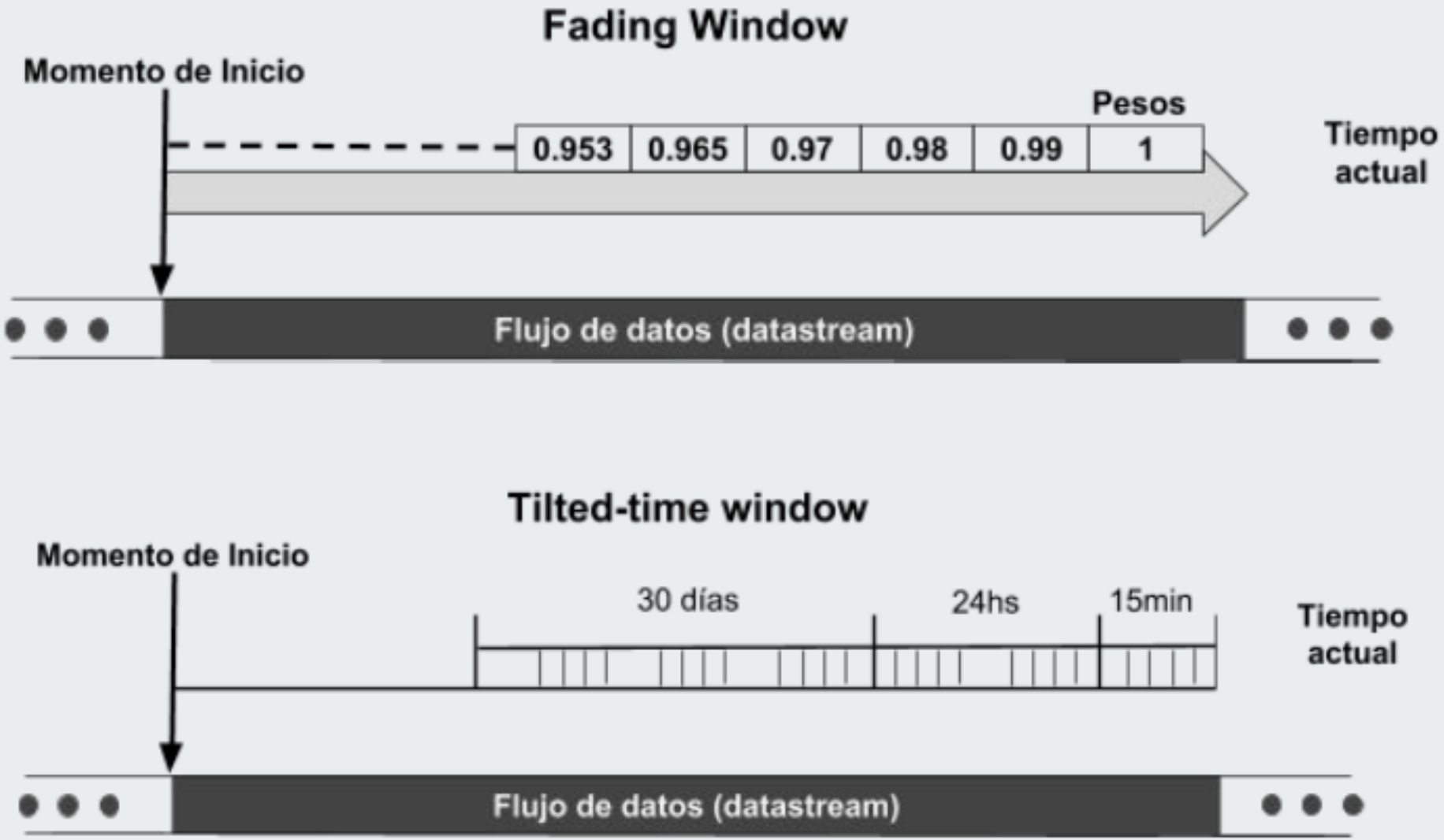
Ventanas de tiempo

- Landmark Window
- Sliding Window
- Fading Window (Damped Window)
- Tilted Time Window

Ventanas de tiempo



Ventanas de tiempo

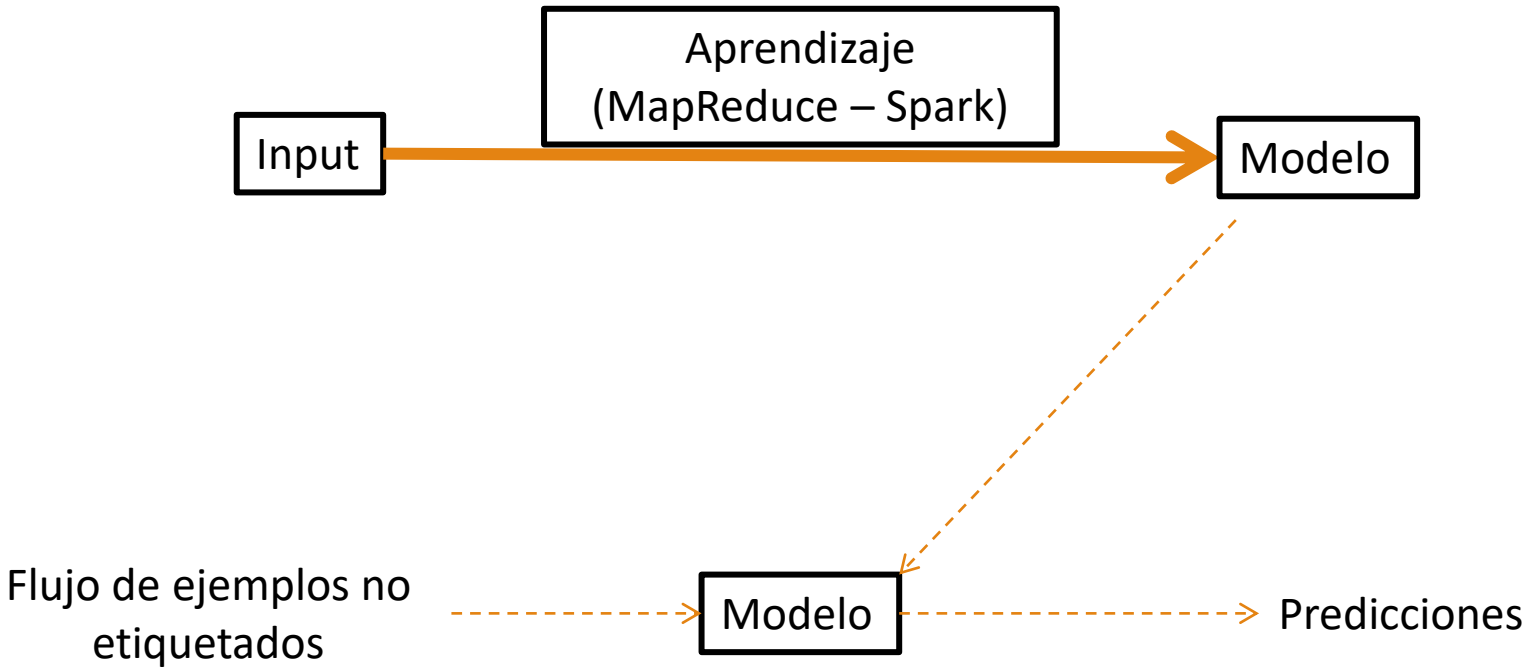


Uso de algoritmos de streaming

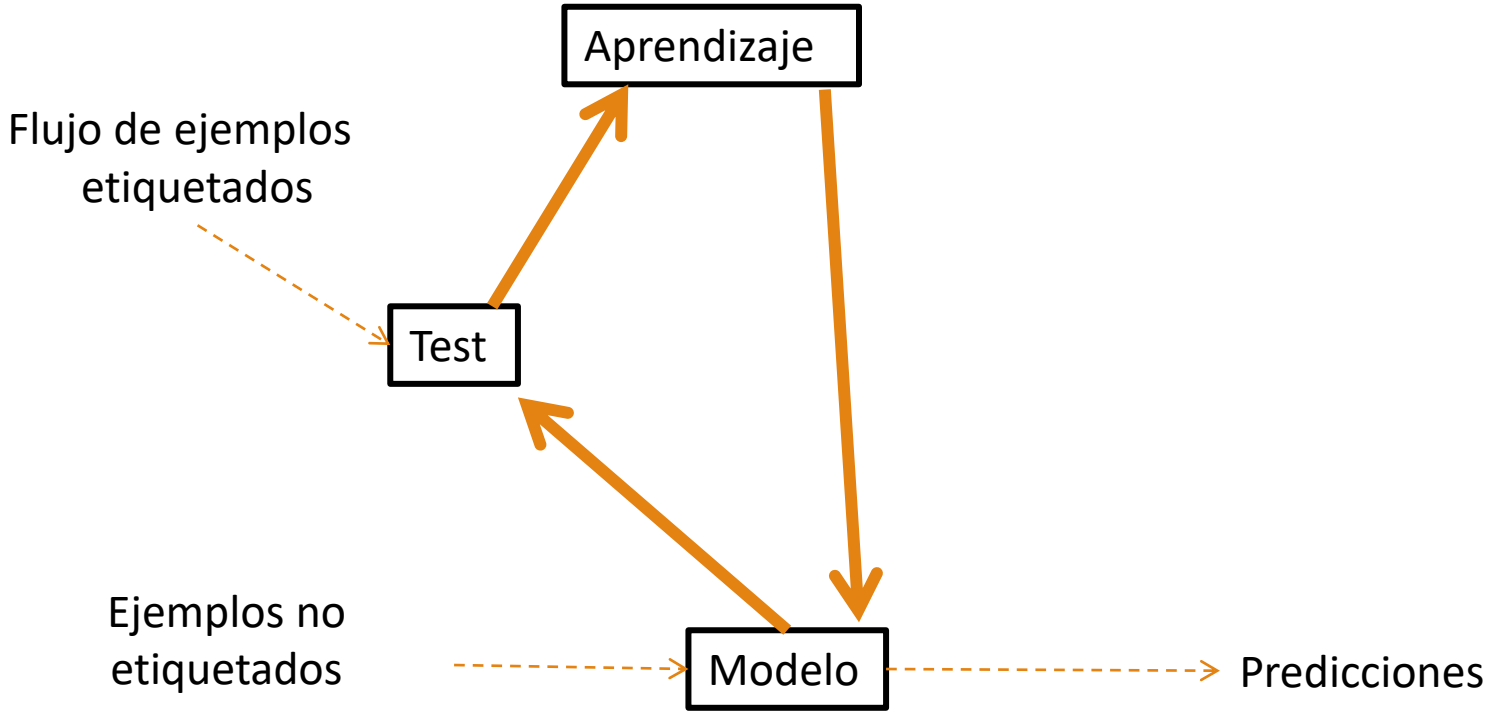
Por lo general se utilizan como modelos predictores:

- El modelo puede estar entrenado de entemano y usarlo sobre el streaming
- El modelo se entrena con el propio streaming
  - Esta variante puede seguir entrenando y actuar como predictor al mismo tiempo

Entrenamiento off-line



Entrenamiento on-line



Stream processing

Un algoritmo de streaming debe cuidar tres aspectos:

- Velocidad. Debe poder operar un nuevo dato en el menor tiempo posible.
- Memoria. Debe ocupar la menor cantidad de memoria RAM.
- Eficacia. Debe poder clasificar nuevos datos con la mayor eficacia posible.

Ejemplo – Cálculo del promedio

Se desea calcular el valor promedio de diferentes valores que se leen de un stream

Todo el tiempo debe ser posible devolver el promedio actual

Ejemplo – Cálculo del promedio

1. ¿Cuál es el modelo?

```
n = 0
suma = 0
```

Ejemplo – Cálculo del promedio

2. ¿Cómo lo actualizo?

```
def newExample(value):
    n = n + 1
    acumulado = acumulado + value
```

Ejemplo – Cálculo del promedio

3. ¿Cómo respondo?

```
def getResponse():
    return acumulado / n
```

Ejemplo – Recomendador

Un sitio web de venta de productos desea recomendar productos relacionados a la visita de otro producto.

Cuando un usuario ve un producto el sitioweb debe mostrarle tres productos relacionados

El flujo de datos está compuesto por tuplas de la forma <user, page, time>

Ejemplo – Recomendador

Para cada usuario del sitio web que visita un producto iremos almacenando el tiempo de permanencia en esa página, de una manera de almacenar el interés de ese usuario con ese producto.

Cuanto más tiempo esté ese usuario visitando esa página, interpretaremos que está interesado en ese producto.

Productos

Usuarios

0	0	5	6	0	0	0	10	15	1
0	0	0	4	0	0	3	4	16	3
0	25	0	0	0	0	90	3	2	1
32	45	12	32	6	0	3	0	2	9
1	0	0	2	0	0	20	0	0	0
1	4	8	0	3	7	0	0	0	0
0	0	11	30	0	3	45	0	12	0
0	2	3	9	12	0	0	1	0	8

Ejemplo – Recomendador

1. ¿Cuál es mi modelo?

```
for i in 1..cantUsuarios:
    for j in 1..cantProductos:
        visitas[i][j] = 0
```

Productos

Usuarios

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Ejemplo – Recomendador

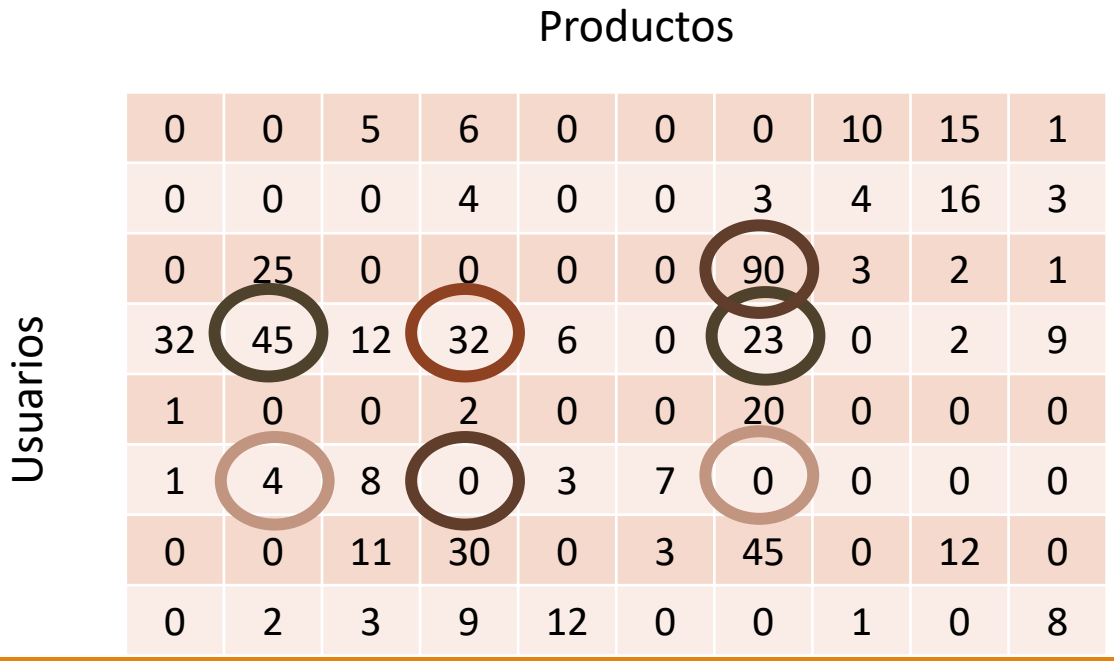
2. ¿Cómo lo actualizo?

```
def newExample(user, page, time):
    visitas[user][page]+= time
```

Ejemplo – Recomendador

3. ¿Cómo respondo una "predicción"?

```
def getResponse(user, page):
    ...
```

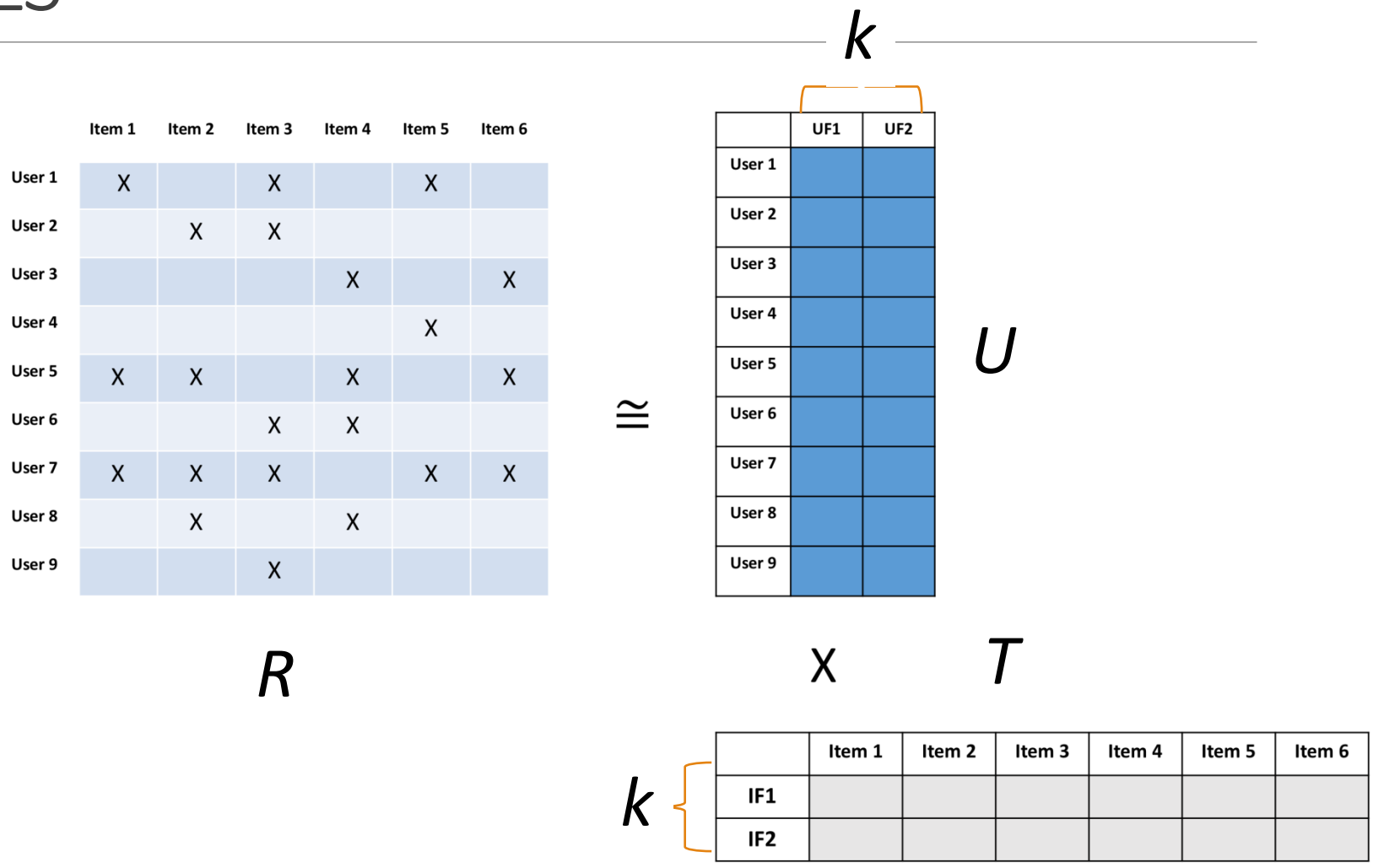


ALS

ALS (Alternating least square) es un algoritmo de recomendación

	i_1	i_2	. . .	i_k	. . .	i_n
U_1	5	?	. . .	3	. . .	4
U_2	?	?	. . .	4	. . .	5
.	. . .	. . .	. . .	. . .	. . .	. . .
U_k	2	5	. . .	?	. . .	3
.	. . .	. . .	. . .	. . .	. . .	. . .
U_m	5	4	. . .	2	. . .	?

ALS


k
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
User 1
User 2
User 3
User 4
User 5
User 6
User 7
User 8
User 9
UF1
UF2
U
≅
R
X
T
k
IF1
IF2

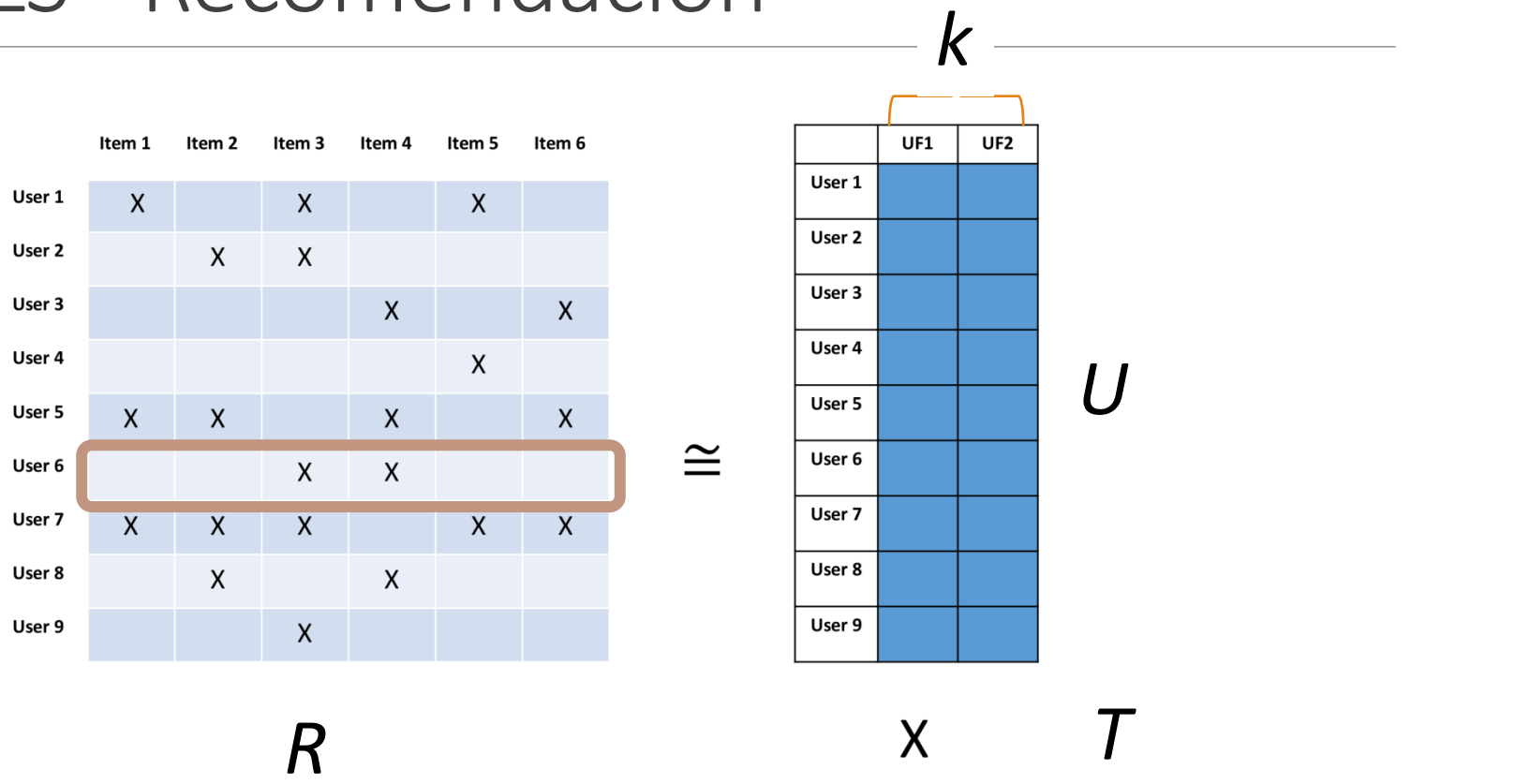
ALS

Las matrices U y T se consiguen de manera iterativa

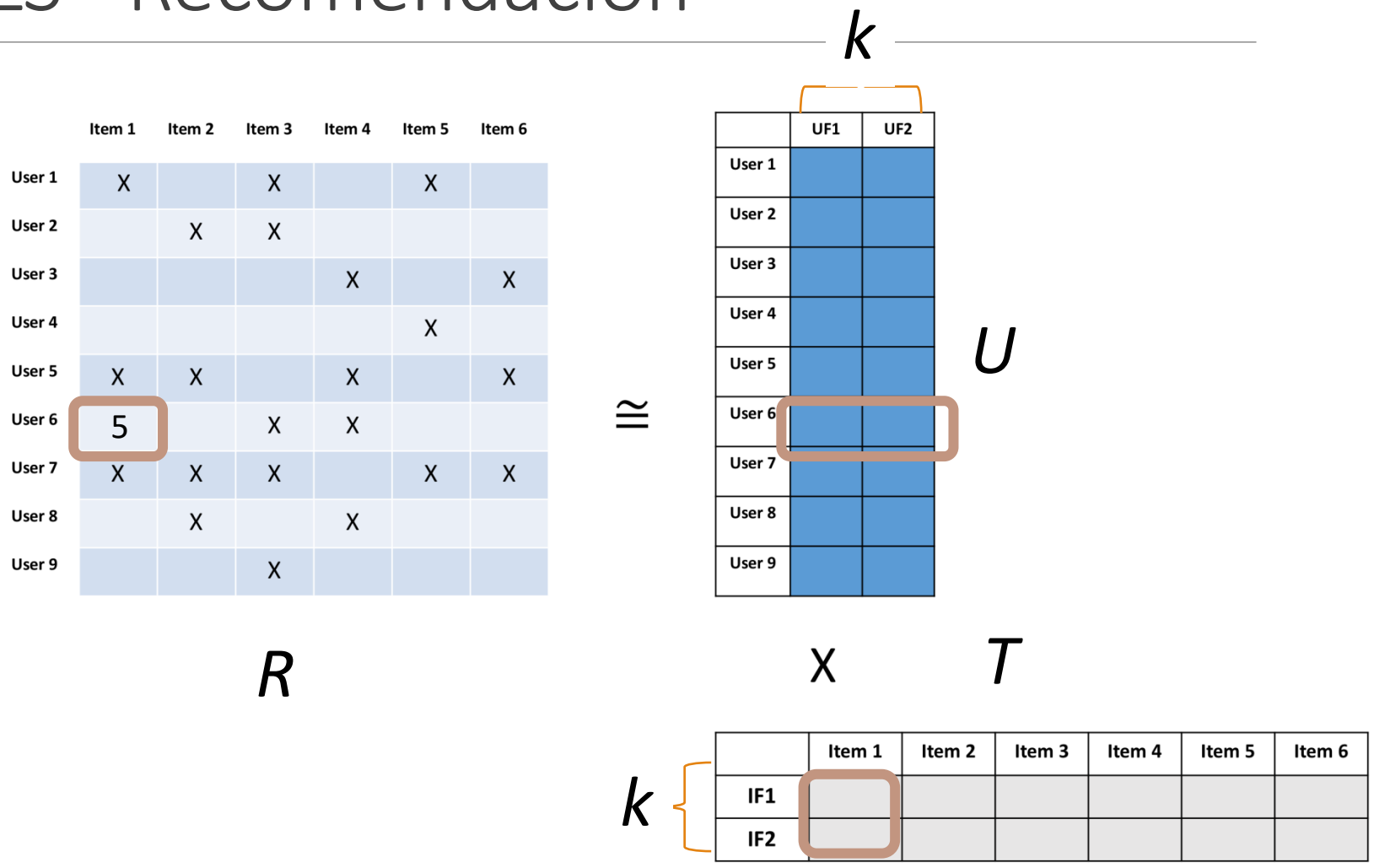
$$U_j = \left[\sum_{i;r_{ji}>0} \left(T_i T_i^t \right) + \lambda I_k \right]^{-1} \sum_{i;r_{ji}>0} \left(R_{ji} T_i \right)$$

$$T_j = \left[\sum_{i;r_{ji}>0} \left(U_i U_i^t \right) + \lambda I_k \right]^{-1} \sum_{i;r_{ji}>0} \left(R_{ji} U_i \right)$$

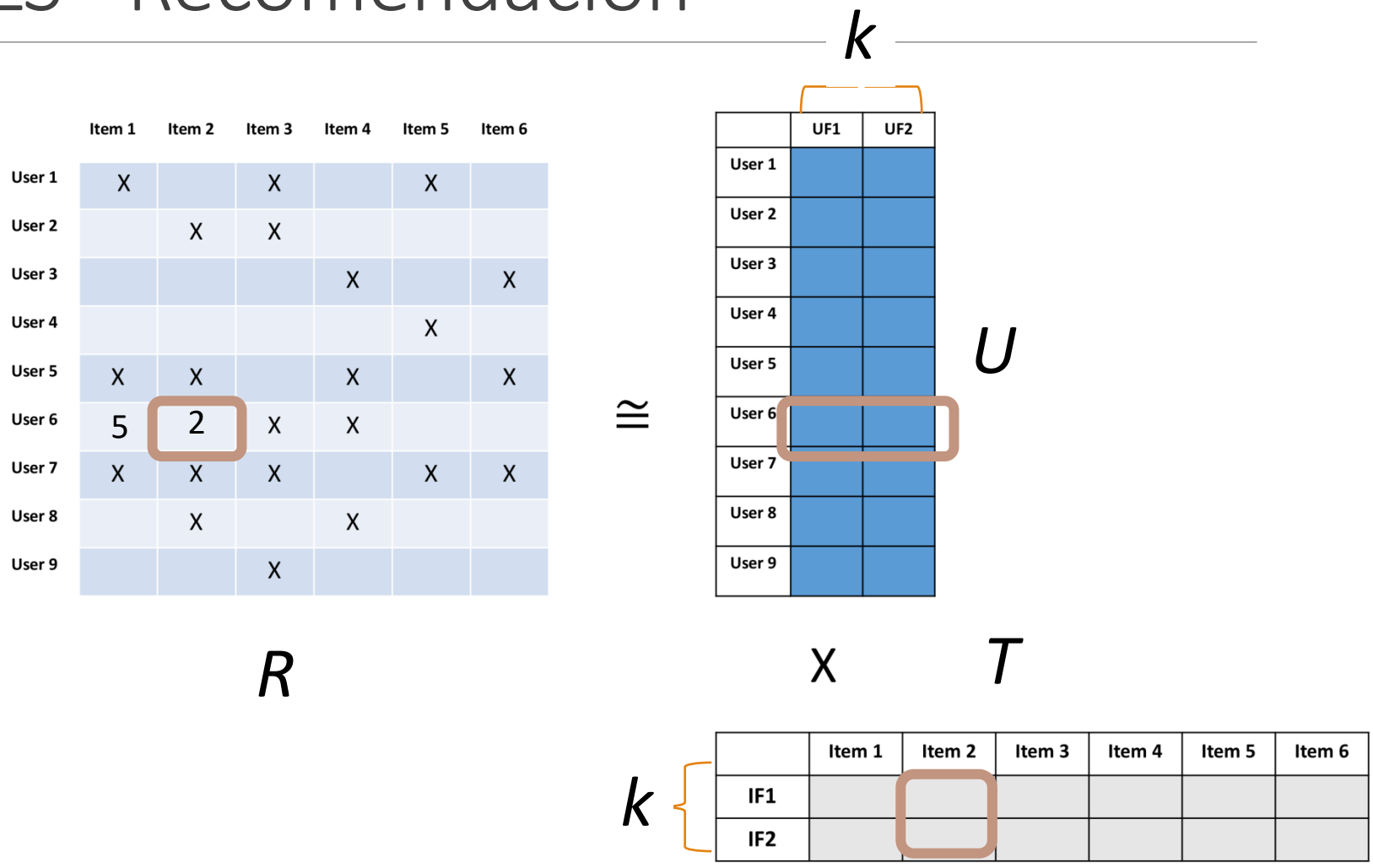
ALS - Recomendación



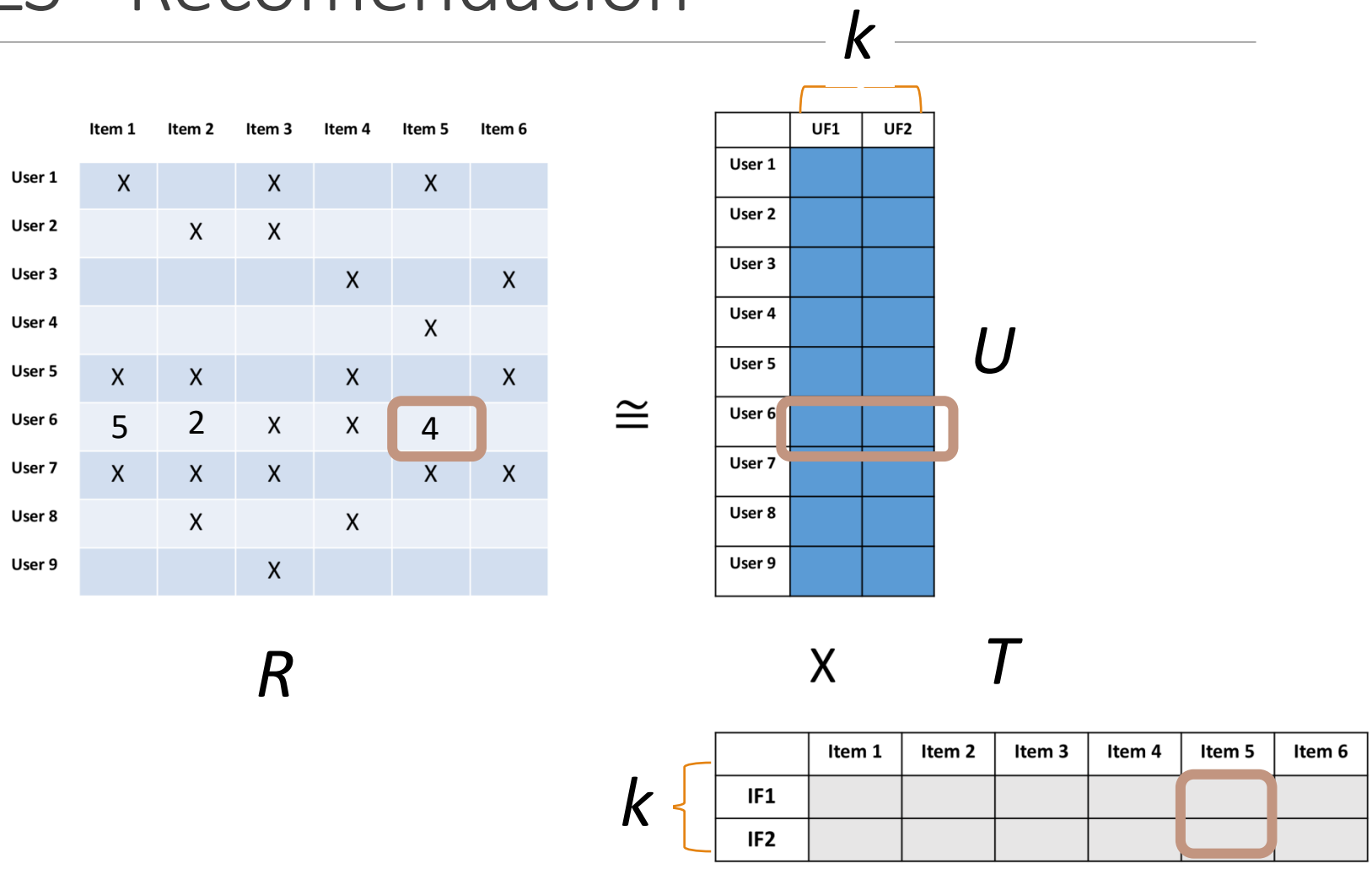
ALS - Recomendación



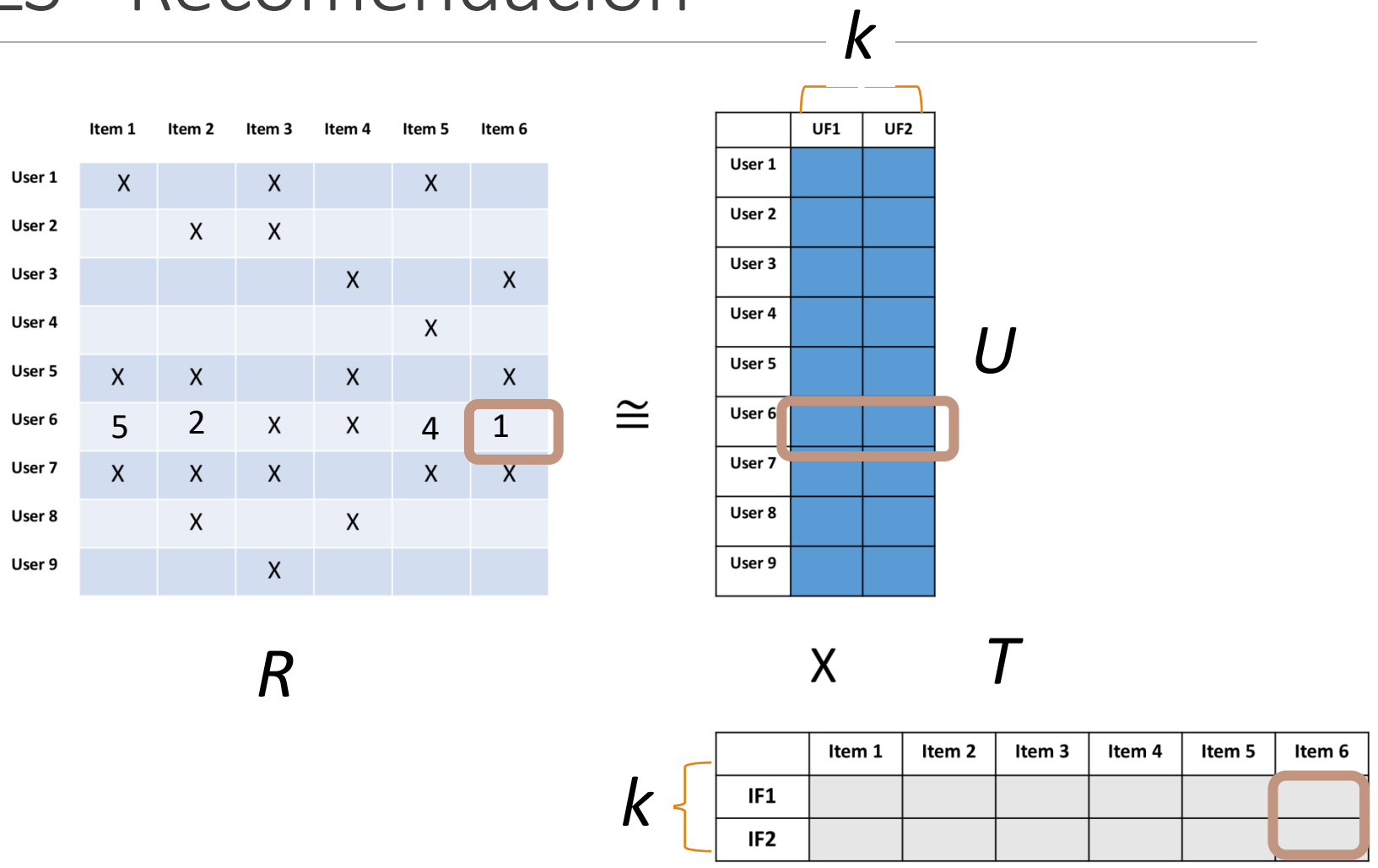
ALS - Recomendación



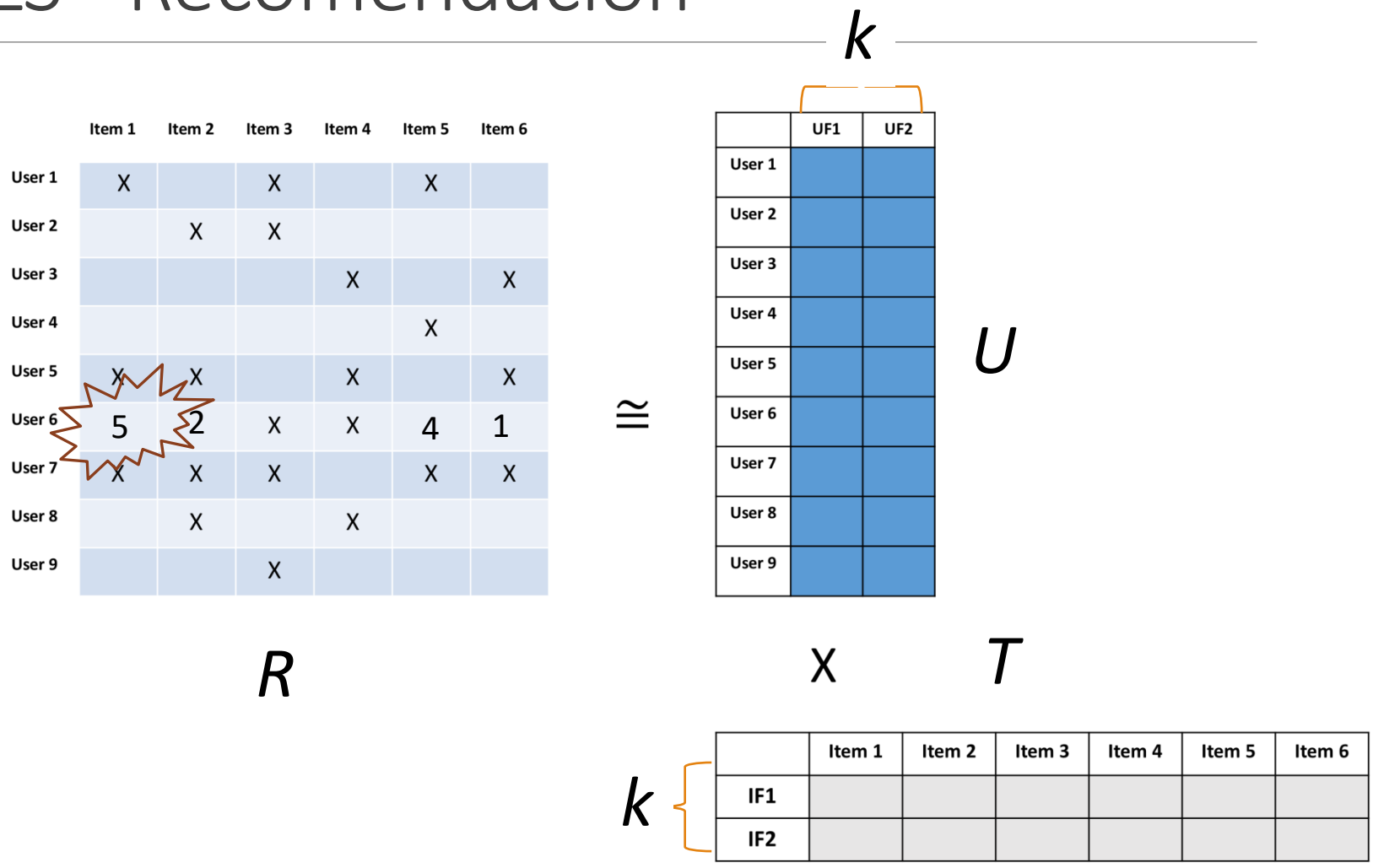
ALS - Recomendación



ALS - Recomendación



ALS - Recomendación



Stream processing

- Clustering
- Clasificación
- Minado de patrones frecuentes
- Detección de cambios
- Reducción de dimensionalidad
- Predicción
- Consultas
- Join de streams