HENRY


# Aprendizaje no supervisado

Data Science

# Agenda

- Aprendizaje no supervisado
- Segmentación (Clustering)
- Métricas de evaluación
- Reducción de dimensionalidad
- Sistemas de Recomendación

# *OBJETIVOS* DE LA CLASE

***Al finalizar esta lecture estarás en la capacidad de...***

- ➜ Comprender el Aprendizaje No Supervisado
- ➜ Entender el concepto de Segmentación
- ➜ Usar las técnicas de evaluación de modelos de aprendizaje no supervisado
- ➜ Utilizar el concepto de Reducción de la Dimensionalidad
- ➜ Conocer el funcionamiento de los Sistemas de Recomendación

Al **finalizar** cada uno de los temas, tendremos un **espacio de consultas**.

Hay un **mentor** asignado para responder el **Q&A**.

¡Pregunta, pregunta, pregunta! :D

# Aprendizaje **no supervisado**

# ¿Qué es?

En los problemas de aprendizaje no supervisado, ya no tenemos la clase o valor de salida de nuestros datos.

Por lo que se usan diferentes algoritmos para encontrar patrones en los datos y hacer que nuestro dataset sea el que nos indica cómo está compuesto, subgrupos dentro de él y diferentes características que pueden presentarse.

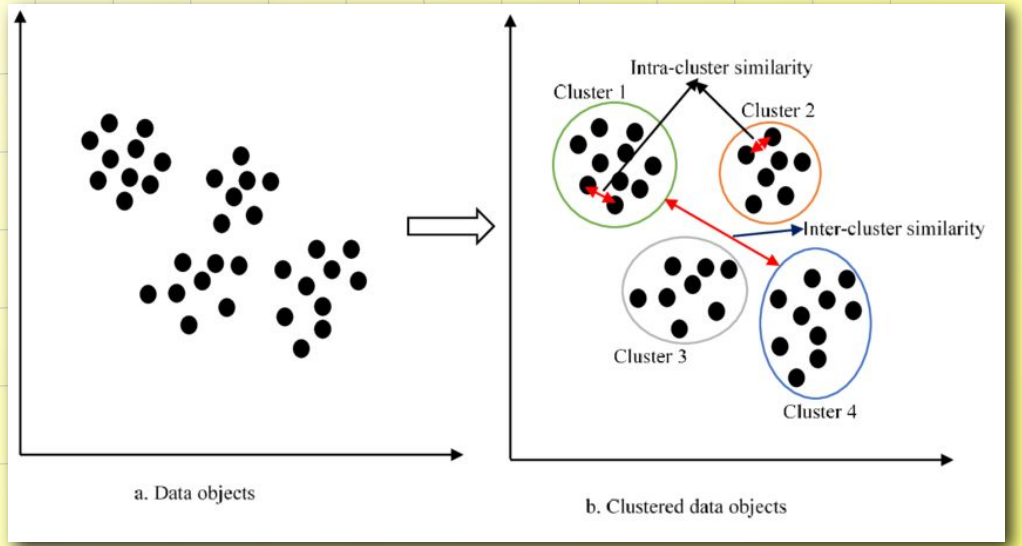
Veremos dos técnicas de aprendizaje no supervisado: clustering y reducción de dimensionalidad.

# clustering

# ¿Qué es?

El clustering es una técnica utilizada para agrupar datos de acuerdo a cuánto se parecen entre sí.

Dado un set de datos, la meta del clustering será encontrar clústers en los cuales las instancias pertenecientes sean parecidas.



# Algoritmos de Clustering

Entre los algoritmos que nos facilitan la tarea de medir que tan cerca están las instancias y armar los grupos están:

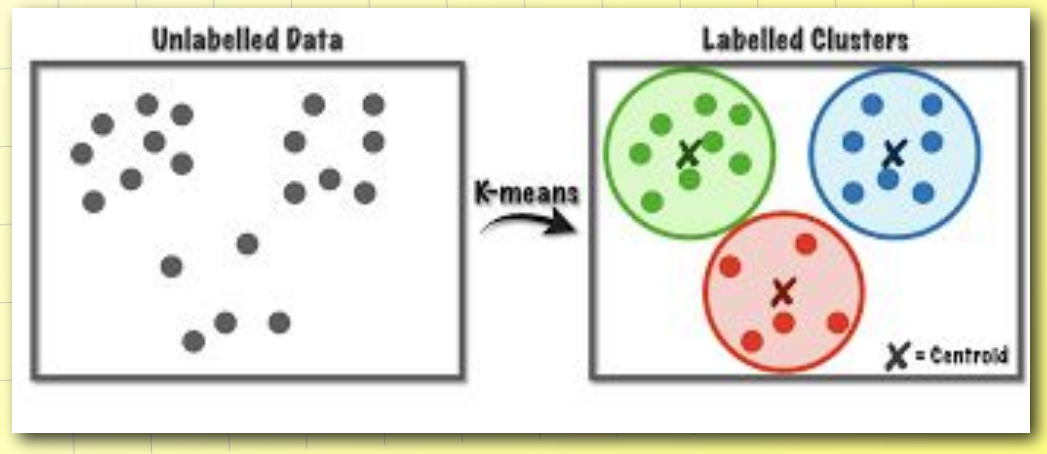
- K-means
- DBSCAN
- Hierarchical clustering
- Fuzzy C-means

El alcance de este curso cubrirá los dos primeros: k-means y DBSCAN.

# K-means

Este algoritmo intenta separar los datos en K clústers, agrupando instancias que se encuentren cercanas.

El centro de cada clúster es llamado centroide y es el promedio de todos los puntos pertenecientes al clúster.



# K-means

Este algoritmo trabaja de manera iterativa:

1. Se inicializan los K centroides.
2. Se asigna cada instancia al centroide más cercano.
3. Se actualizan los centroides (media).
4. Se repiten los pasos 2 y 3 (hasta que la posición del centroide no varíe).

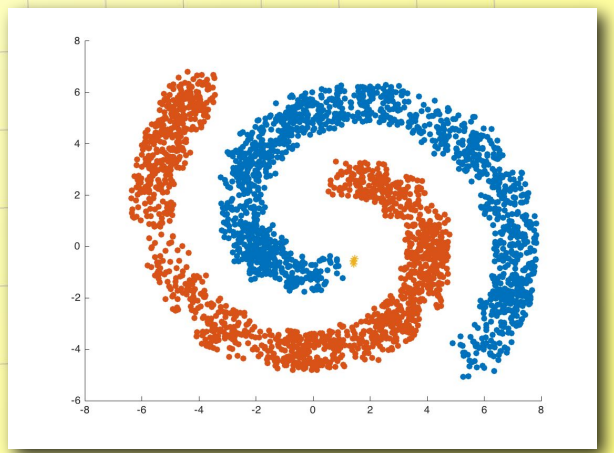
# DBSCAN

*Density-based spatial clustering of applications with noise* es un algoritmo en el que NO hace falta seleccionar la cantidad de clústers de antemano, ya que los define automáticamente a partir de la densidad de puntos y no de centroides.

Adicionalmente, incorpora el análisis de outliers, ya que no los clasifica.

Para este algoritmo, los clústers son regiones densas en el espacio de datos.

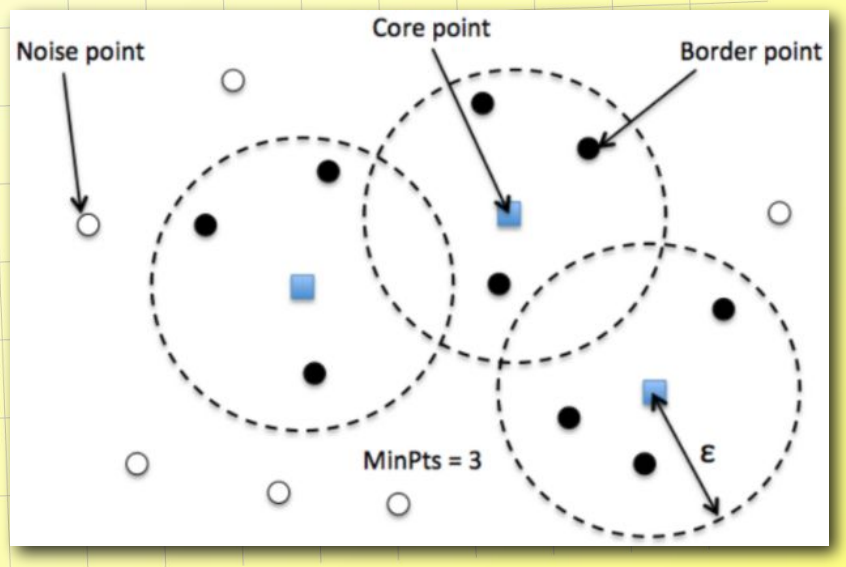
Cada punto del clúster debe tener un mínimo de vecinos en un radio determinado para no ser considerado outlier.



# DBSCAN

Existen 3 tipos de puntos:

- Core point: tiene al menos m puntos a una distancia n de él.
- Border point: tiene al menos un Core point a una distancia n
- Noise: no es ni Core ni Border y tiene menos de m puntos a una distancia n.



# DBSCAN

Los hiperparámetros clave de este algoritmo son:

- ➜ Epsilon: magnitud del radio considerado.
- ➜ MinPoints: cantidad mínima de vecinos para no ser considerado outlier.

A diferencia de K-means, DBSCAN es más flexible y permite adaptarse a formas de clúster más complejas, mientras k-means funciona mejor con cluster alejados, bien agrupados y globulares.

# K-means Vs. DBSCAN

| K-Means | DBSCAN |
| --- | --- |
| Muy Rápido | Es computacionalmente más costoso |
| No tiene parámetros | Hay que elegir bien los parámetros |
| Fácil de asignar nuevas instancias | |
| Hay que definir el número de clusters | No hay que elegir el número de clusters |
| Sólo funciona bien con clusters tipo esferas | Detecta cualquier forma de clusters |
| Sensible a outliers | Determina automáticamente los outliers |
| | No anda bien si hay clusters de diferentes densidades |

# Métricas de **evaluación**

# ¿Cuáles?

Existen métricas para evaluar los modelos de aprendizaje no supervisado.

A diferencia del paradigma del aprendizaje supervisado, aquí no hay etiquetas para comparar cuán alejado de ellas estuvo nuestro valor predicho.

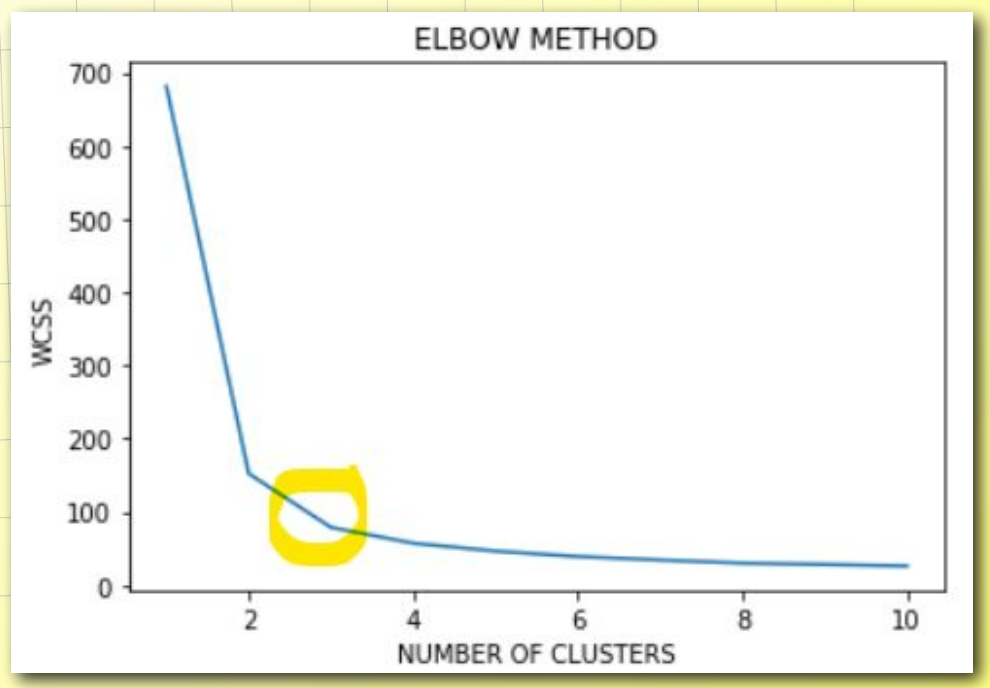
Desarrollaremos dos métodos: Elbow y Silhoutte.

# Elbow

Esta métrica se utiliza para el algoritmo k-means.

El K óptimo puede encontrarse en el codo de la curva.

Utiliza la suma de errores cuadrados dentro de clúster como medida para decidir K.



# Silhoutte

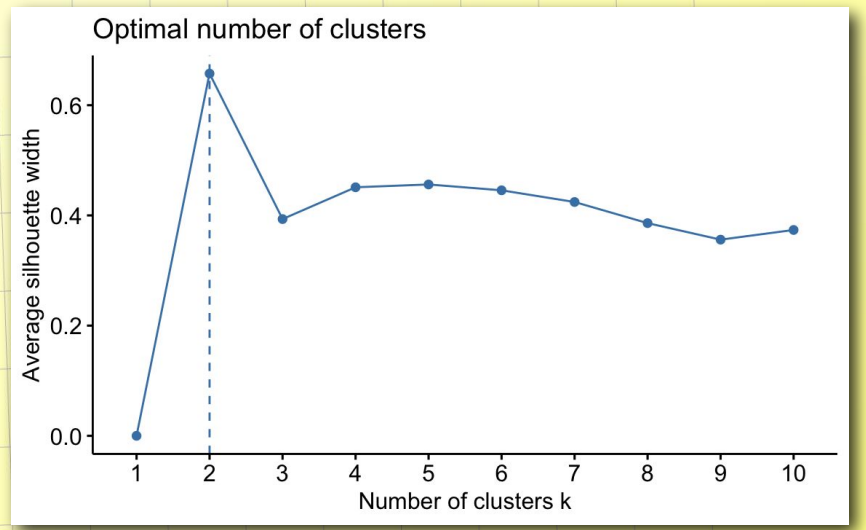
Esta métrica mide qué tan parecidos son los datos con su propio clúster:

$$s(i) = \frac{b(i) - a(i)}{\max\left(a(i), b(i)\right)}$$

Se utiliza para cualquier técnica de clustering.

# Silhoutte

- Su valor oscila entre -1 y 1.
- Un valor de 1 indica que la instancia está bien emparejada con su propio clúster y mal emparejada con clústers vecinos.
- Si la mayoría de las instancias tienen valor alto, entonces la configuración del cluster es apropiado.



# **Reducción** de Dimensionalidad

# ¿Qué es?

La reducción de dimensionalidad busca disminuir la cantidad de features de un dataset, siempre reteniendo la mayor cantidad de información posible. Esto tiene diversos usos, como por ejemplo:

- ➜ Mejorar la eficiencia en modelos de regresión y clasificación.
- ➜ Disminuir el ruído, facilitar la visualización.
- ➜ Detectar features relevantes en dataset.

Como primer aproximación al preprocesamiento de datos.

# Reducción de la dimensionalidad

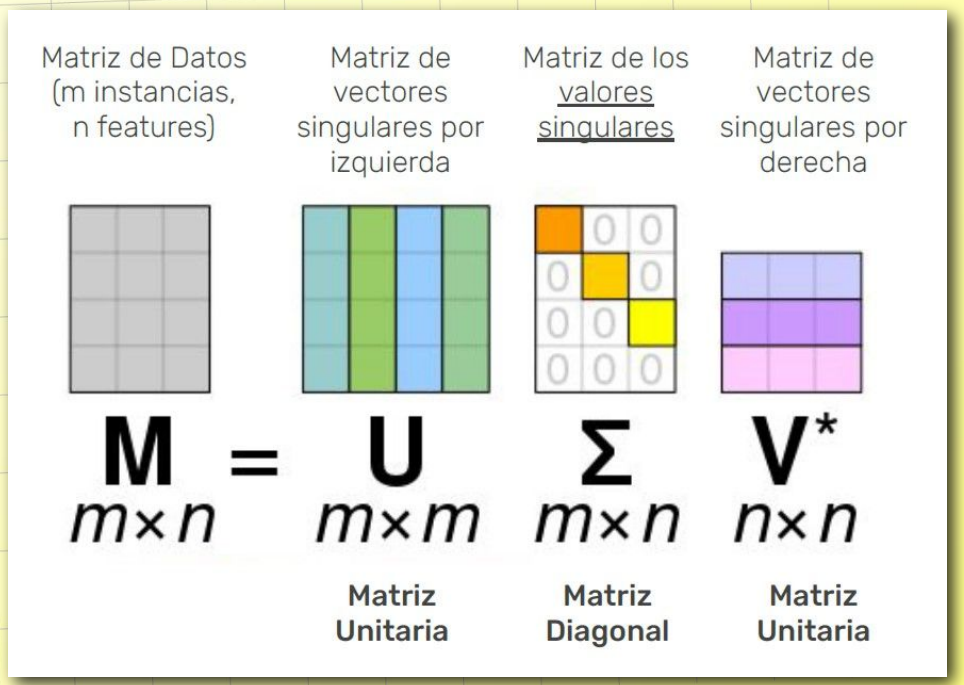
Entre las técnicas más usadas encontramos:

- ➜ SVD
- ➜ PCA
- ➜ MDS
- ➜ t.-SNA
- ➜ LDA
- ➜ GDA
- ➜ Autoencoder

Desarrollaremos SVD y PCA.

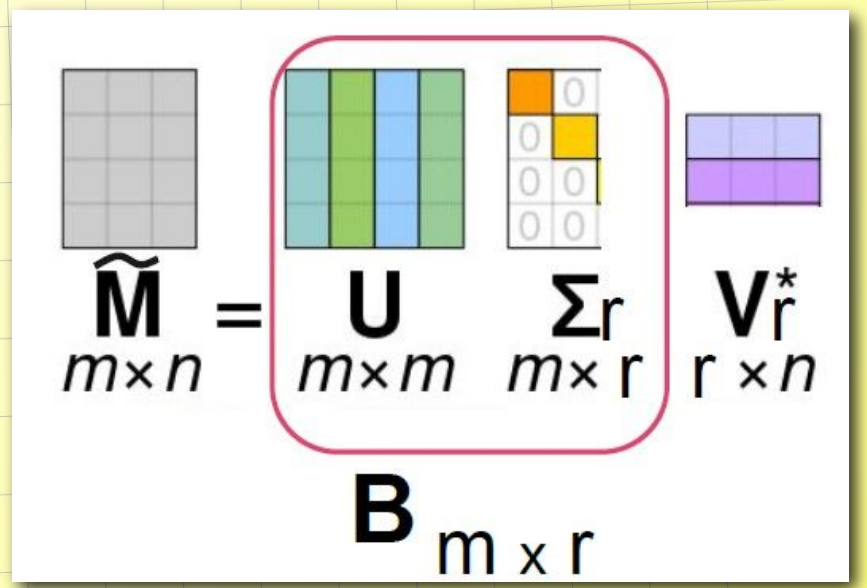
# SVD

SVD (*Singular Value Decomposition*) es un método de álgebra lineal que nos permite representar cualqueir matriz en términos de la multiplicación de otras tres.



# SVD

El objetivo consiste en reducir la cantidad de features. Para lograr esto, buscamos crear una nueva matriz B que reemplace a M, para que tenga menos columnas.
Esto se conoce como SVD truncado.



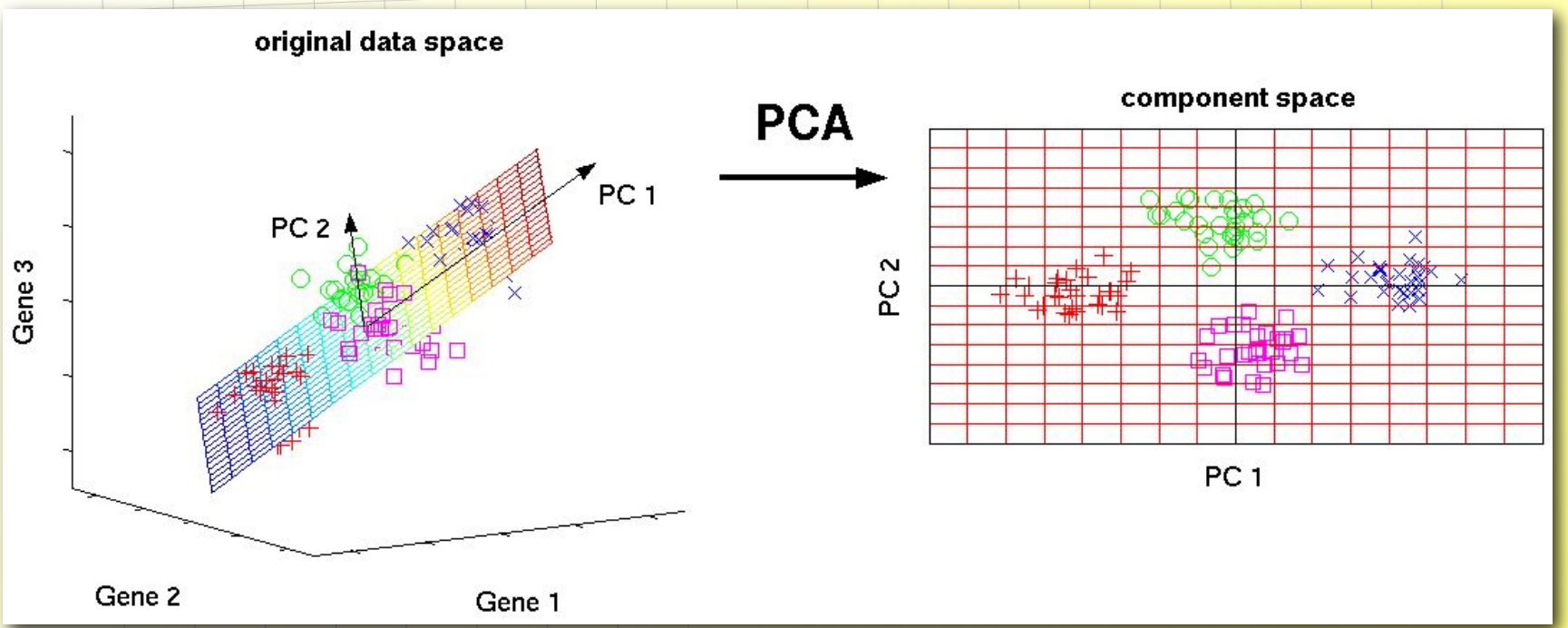
# PCA

PCA (*Análisis de Componentes Principales*) permite realizar una descomposición de d variables correlacionadas en d variables no correlacionadas.

A través de combinaciones lineales de las variables originales que maximizan la varianza explicada, se consiguen estos llamados componentes principales.

Entonces, el primer componente principal estará proyectado en la dirección que representa la mayor varianza explicada, el segundo en la segunda dirección en términos de varianza y así sucesivamente.

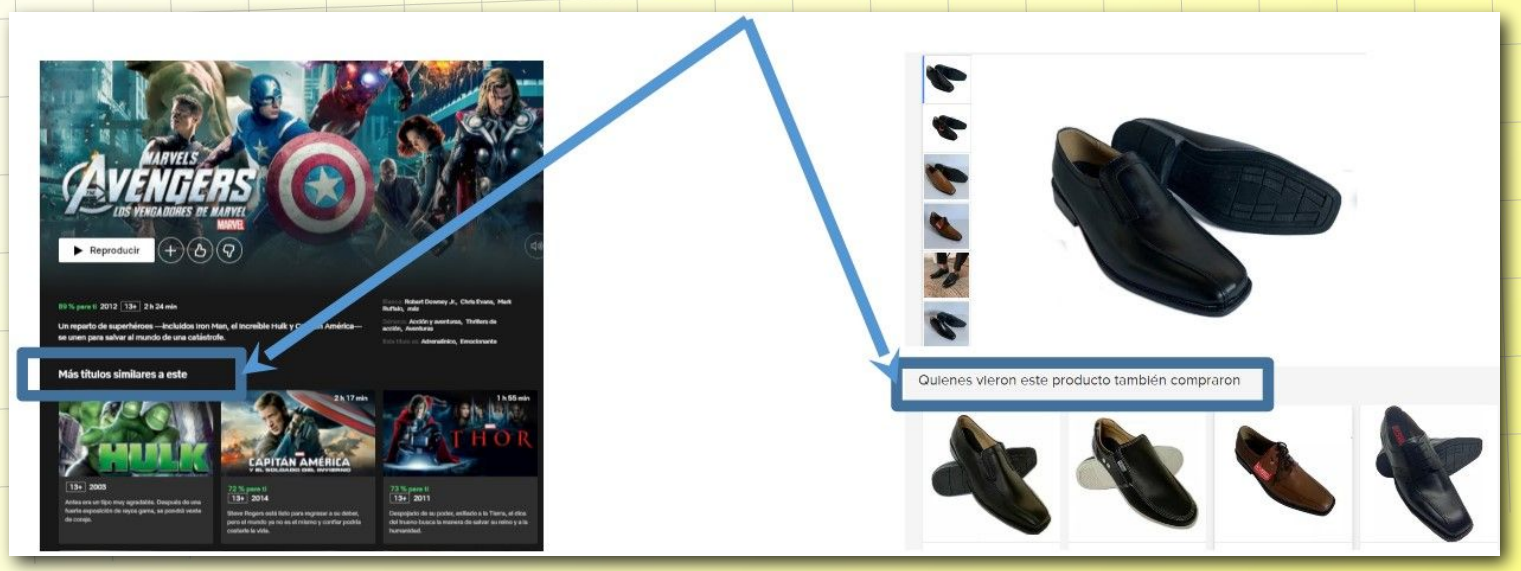
# PCA


original data space
PC 1
PC 2
Gene 3
Gene 2
Gene 1
PCA
component space
PC 2
PC 1

# sistemas de **Recomendación**

# ¿Dónde?

Es muy común encontrar en diversas plataformas, recomendaciones de productos para consumo, en base al producto seleccionado.



# Sistemas de Recomendación

- Existen usuarios e ítems. Los usuarios prefieren algunos ítems por sobre otros.
- Ejemplo: Usuarios de Netflix y Películas. De 1 a 5 estrellas.
- El objetivo del sistema de recomendación es poblar la matriz de utilidad de una manera inteligente y bajo los requisitos que imponga cada entorno.

Matríz de Utilidad

| | P1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Usuario 1** | 5 | 4 | ? | ? | 2 | ? | ... | 1 |
| **Usuario 2** | 2 | 1 | ? | 5 | ? | ? | ... | 5 |
| **Usuario 3** | ? | 1 | 5 | ? | 4 | 3 | ... | 2 |
| **Usuario 4** | 4 | ? | ? | 2 | 1 | ? | ... | ? |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **Usuario *n*** | 1 | 2 | 5 | ? | 5 | ? | ... | 3 |

# Sistemas de Recomendación

- Por ejemplo, Netflix tiene 150 millones suscriptores y 5 mil películas. La matriz tiene 750 mil millones de espacios, de los cuales la mayoría están vacíos.
- Cuando buscamos recomendar, interesa más recomendar ítems que van a gustar que aquellos que no van a gustar.
- En algunos casos, interesa mostrar a los usuarios novedades.

# Sistemas de Recomendación

- Algunas veces, ni siquiera hay calificaciones, solamente si vio o no (o escuchó, leyó, compró, etc.).
- Históricamente, las recomendaciones se hacían por medio de crítica de expertos, listas de favoritos, listas de clásicos, más populares, recientes, etc. Hoy las recomendaciones son específicas para cada usuario.

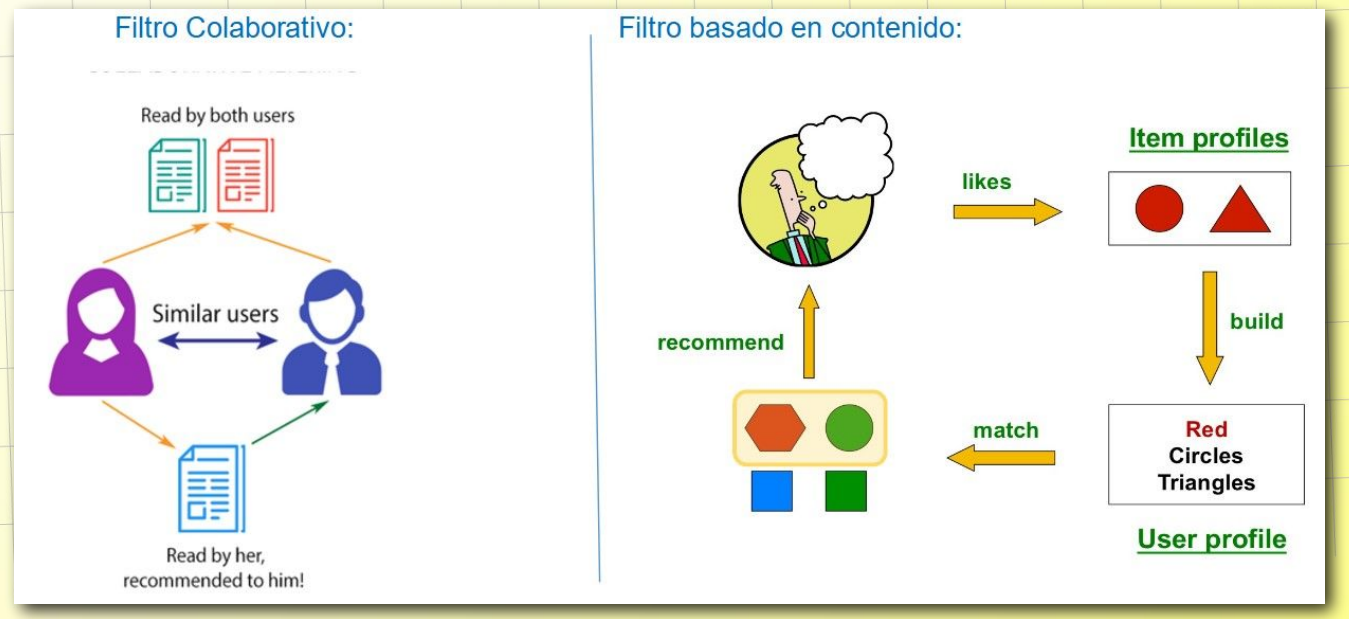
# Formas

Es posible diferenciar dos formas de hacer las recomendaciones:

1. Pedir a los usuarios que puntúen los ítems.
   ➜Los usuarios no suelen hacerlo
   ➜Si lo hacen, puede estar sesgado (gente que prefiere puntuar cosas que no le gustan a puntuar cosas que sí, etc.).
2. Inferir a partir de acciones
   ➜Ejemplo: compra muchas cosas de camping → le gusta el camping, aire libre, etc.
   ➜¿Qué pasa con las cosas que no le gustan?

# Modelos híbridos

En ocasiones, los modelos híbridos que utilizan ambos métodos, son la opción más conveniente.



# ¿PREGUNTAS?

# ¿Alguien dijo Homework?

HENRY


Próxima lecture

# Modelos de ensambles

# ¡Feedback!

Click on me

**Dispones de un formulario en:**

- Homeworks
- Guías de clase
- Slack


HENRY