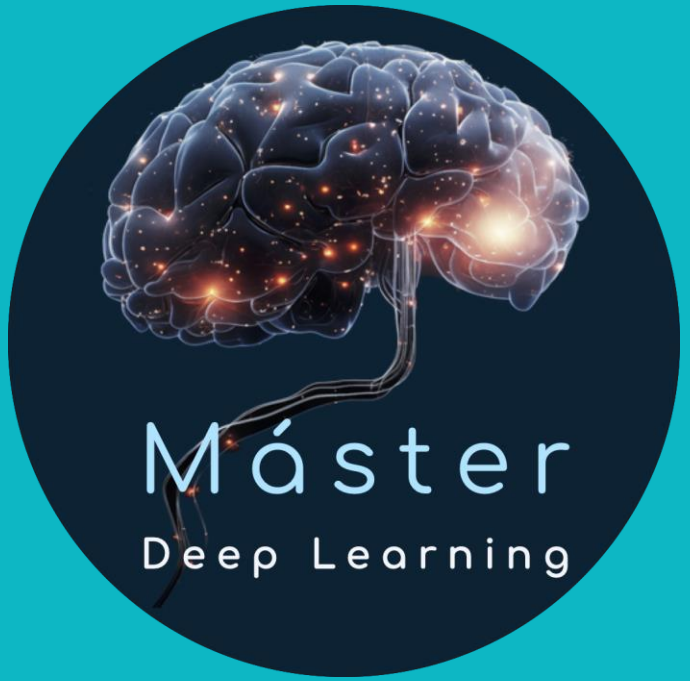
Redes Neuronales para grafos



Tema 1

Introducción a las redes neuronales para grafos

POLITÉCNICA
UNIVERSIDAD POLITÉCNICA DE MADRID

Máster
Deep Learning

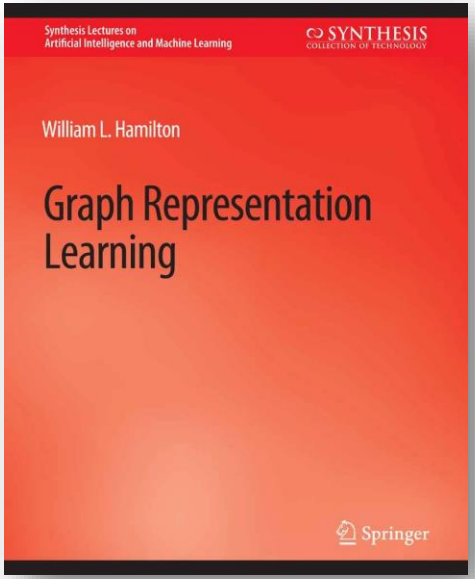
Máster
Deep Learning

Profesores

Ángel Panizo LLedot
angel.panizo@upm.es
Despacho: 1216

Adrian Giron Jimenez
adrian.giron@upm.es
Despacho: 12

Libro de la asignatura



Hamilton, W. L. (2020). Graph representation learning. Morgan & Claypool Publishers.

https://www.cs.mcgill.ca/~wlh/grl_book/

Calendario

semana	martes	jueves
1	14/01: Introducción	16/01: Shallow embeddings
2	21/01: GNN – paso de mensajes	23/01: GNN – arquitecturas avanzadas
3		30/01: GNN – consideraciones prácticas
4		**06/02: Torneo**
5		13/02: Métodos generativos

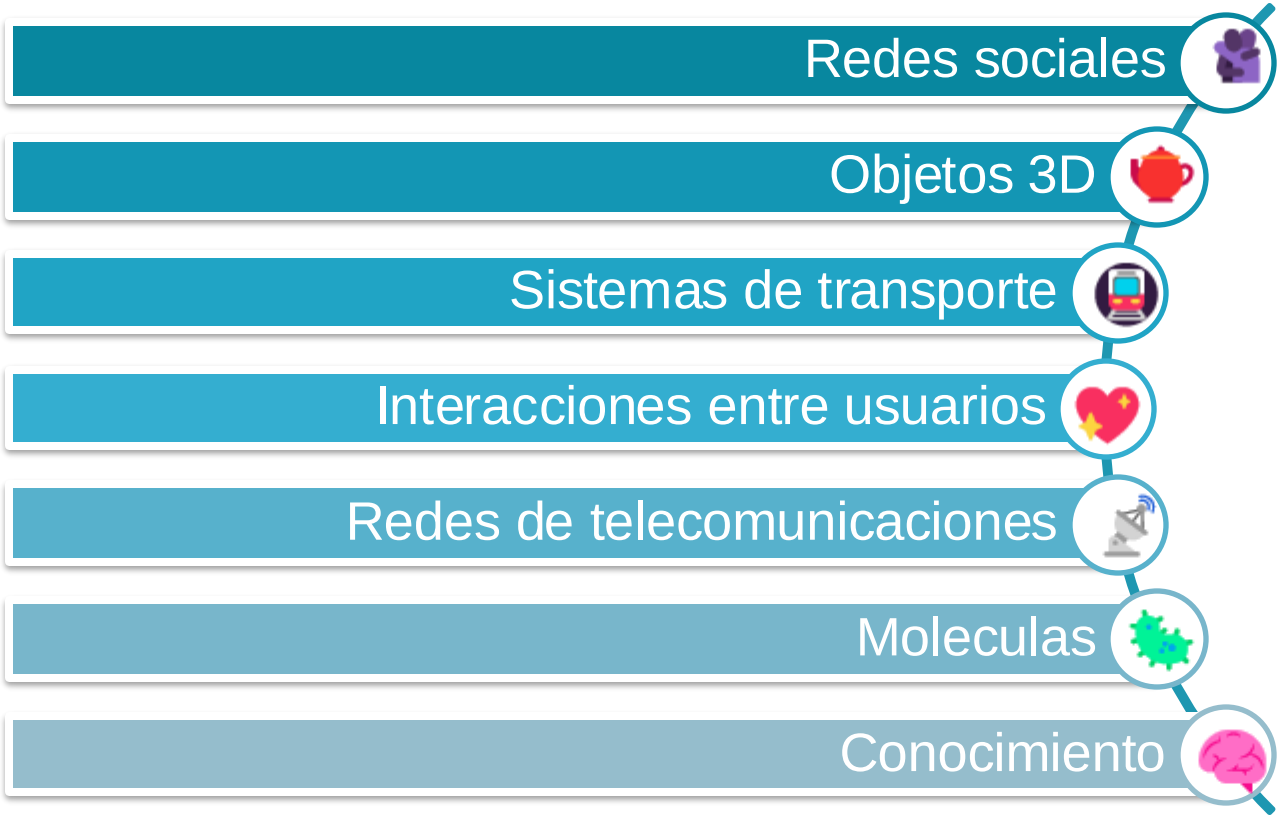
Evaluación

- Una única práctica.
- 85% entrega (Código + memoria).
- 15% actividad en clase (sesión del torneo del 6 de febrero)

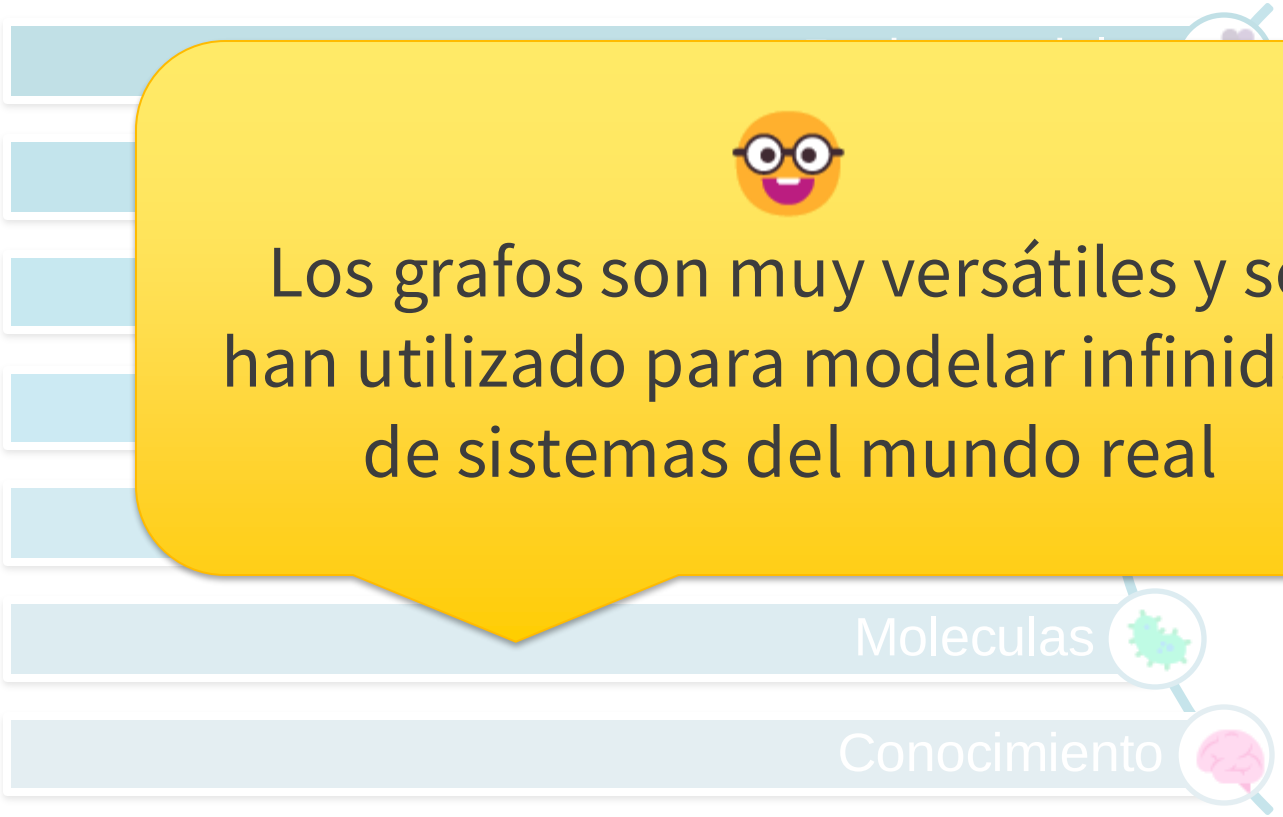
¿Qué son los Grafos?

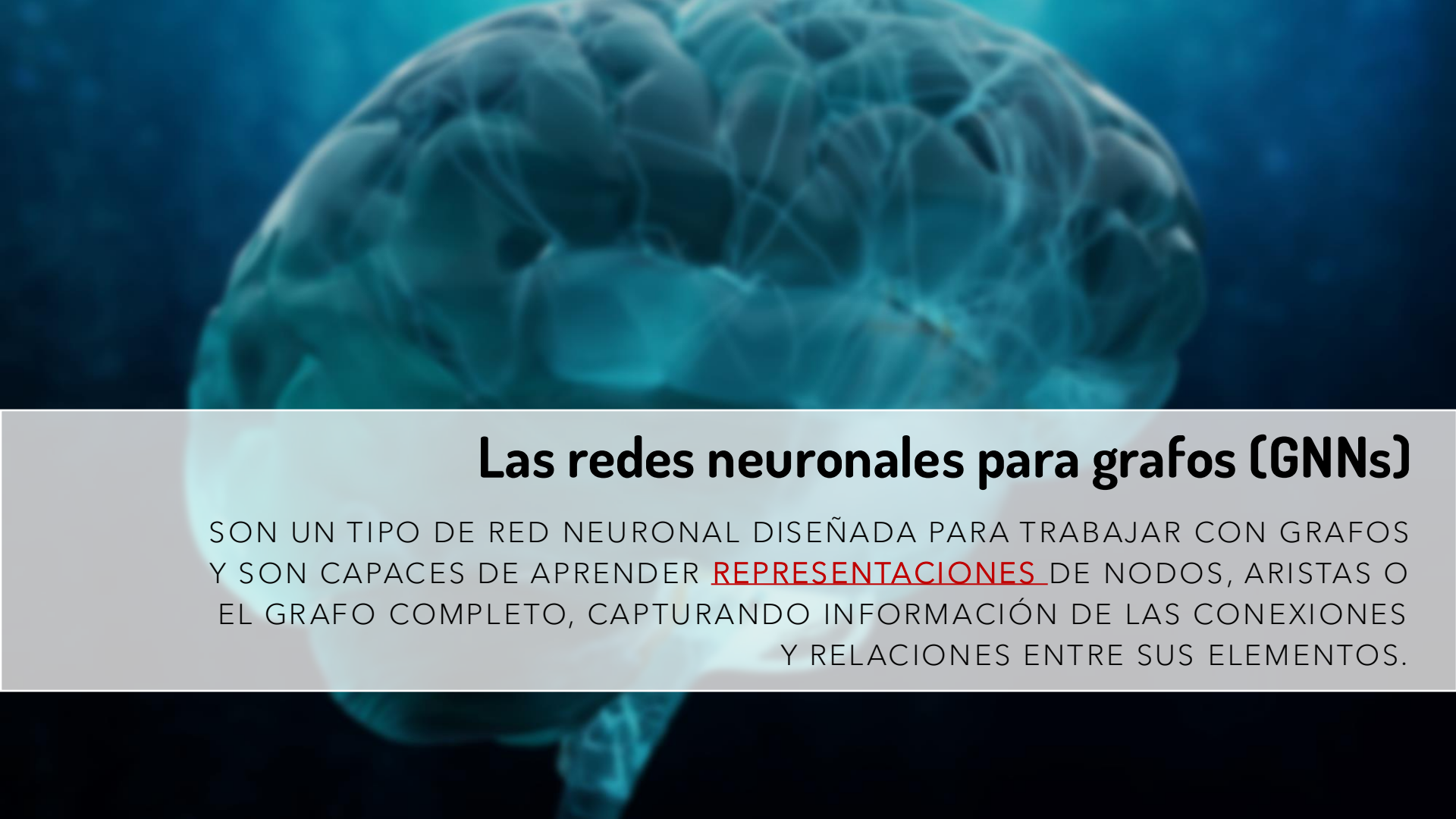
ESTRUCTURAS MATEMÁTICAS QUE REPRESENTAN CONJUNTOS DE OBJETOS (LLAMADOS NODOS) Y RELACIONES ENTRE ELLOS (LLAMADAS ARISTAS)

Ejemplos de grafos



Ejemplos de grafos





Las redes neuronales para grafos (GNNs)

SON UN TIPO DE RED NEURONAL DISEÑADA PARA TRABAJAR CON GRAFOS Y SON CAPACES DE APRENDER REPRESENTACIONES DE NODOS, ARISTAS O EL GRAFO COMPLETO, CAPTURANDO INFORMACIÓN DE LAS CONEXIONES Y RELACIONES ENTRE SUS ELEMENTOS.

¿Qué pueden hacer la GNNs?

- Sistemas de recomendación.
- Predicción de las propiedades de una molecula.
- Generación de mallas 3D.
- Detección de Código duplicado.
- Predicción del tráfico.
- …. etc

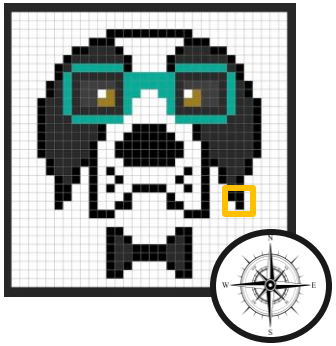
- EL DEEP LEARNING HA SOBRESALIDO EN OTROS DOMINIOS COMO **IMÁGENES** Y **TEXTO**.
- ¿POR QUÉ **NO SE APLICA** TANTO A LOS DATOS DE **RED**?


LAS REDES NO TIENEN UNA ESTRUCTURA HOMOGÉNEA COMO EL TEXTO O LAS IMÁGENES.

Texto

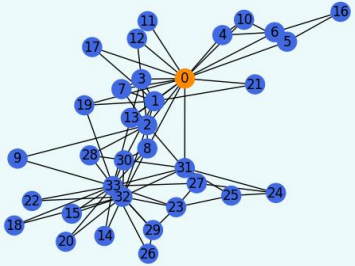
- Tiene principio y fin.
- Todas las palabras tiene sucesores y antecesores
- Se pueden dividir en frases

Imagen



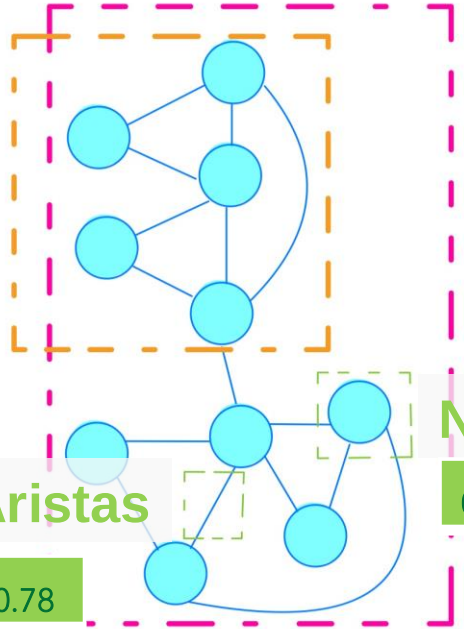
- Tiene un ancho y un alto.
- Se pueden redimensionar.
- Todos los pixels tienen vecinos en cada punto cardinal.

Grafo



- No tiene principio claro.
- No tiene un sistema de coordinadas.
- Cada nodo tiene una vecindad de tamaño diferente.

Tareas



Todo el grafo

0.7	0.34	0.95	0.41	0.08	0.64

Sub-grafos

0.4	0.33	0.35	0.11	0.81	0.45

Nodos

0.1	0.23	0.45	0.01	0.98	0.76

Aristas

0.9	0.54	0.78	0.05	0.08	0.78

Inductive

Transductive

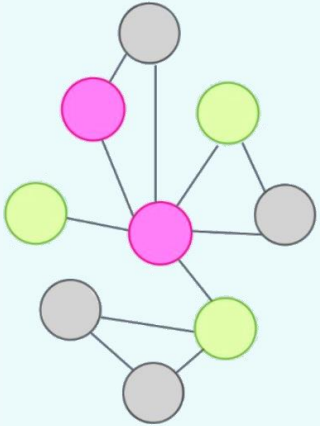
- Cuando aplicamos *Inductive learning* entrenamos un modelo para poder predecir casos nuevos que no tenemos previstos.
- Puede predecir datos nuevos sin necesidad de un reentrenamiento

- Cuando aplicamos *Transductive learning* entrenamos un modelo para predecir un conjunto de casos que ya conocemos de antemano.
- Si aparecen datos nuevos, es necesario entrenar de nuevo.

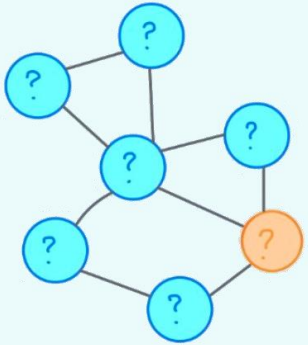
Inductive

VS

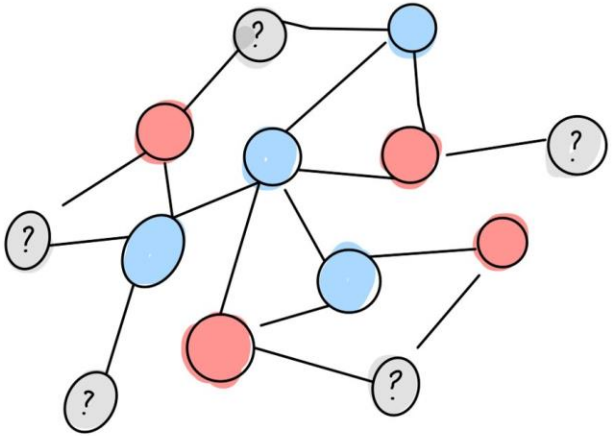
Transductive



Entrenar



Predecir



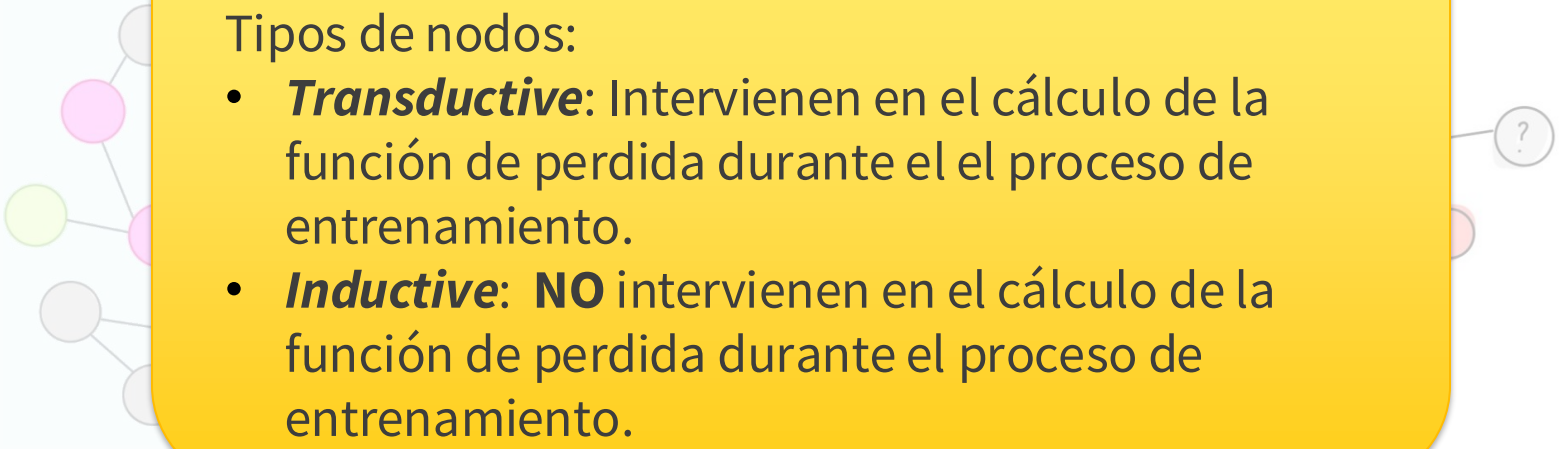
Entrenar y predecir

Inductive

Transductive

Tipos de nodos:

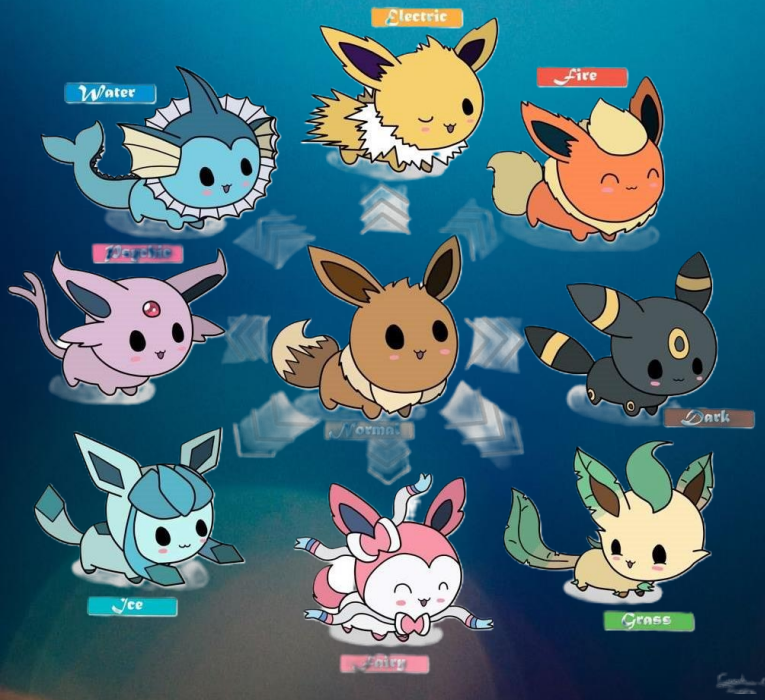
- ***Transductive***: Intervienen en el cálculo de la función de perdida durante el el proceso de entrenamiento.
- ***Inductive***: **NO** intervienen en el cálculo de la función de perdida durante el proceso de entrenamiento.



Entrenar y predecir

Tipos de aprendizaje

Supervisado:	Semi-supervisado:	No-supervisado:
• Se realiza el proceso de testing con nodos de tipo Transductive.	• Se realiza el proceso de testing con nodos de tipo Inductive.	• No tengo etiquetas.



Evolución del machine learning para grafos

DE LOS ATRIBUTOS MANUALES A LAS GNNS

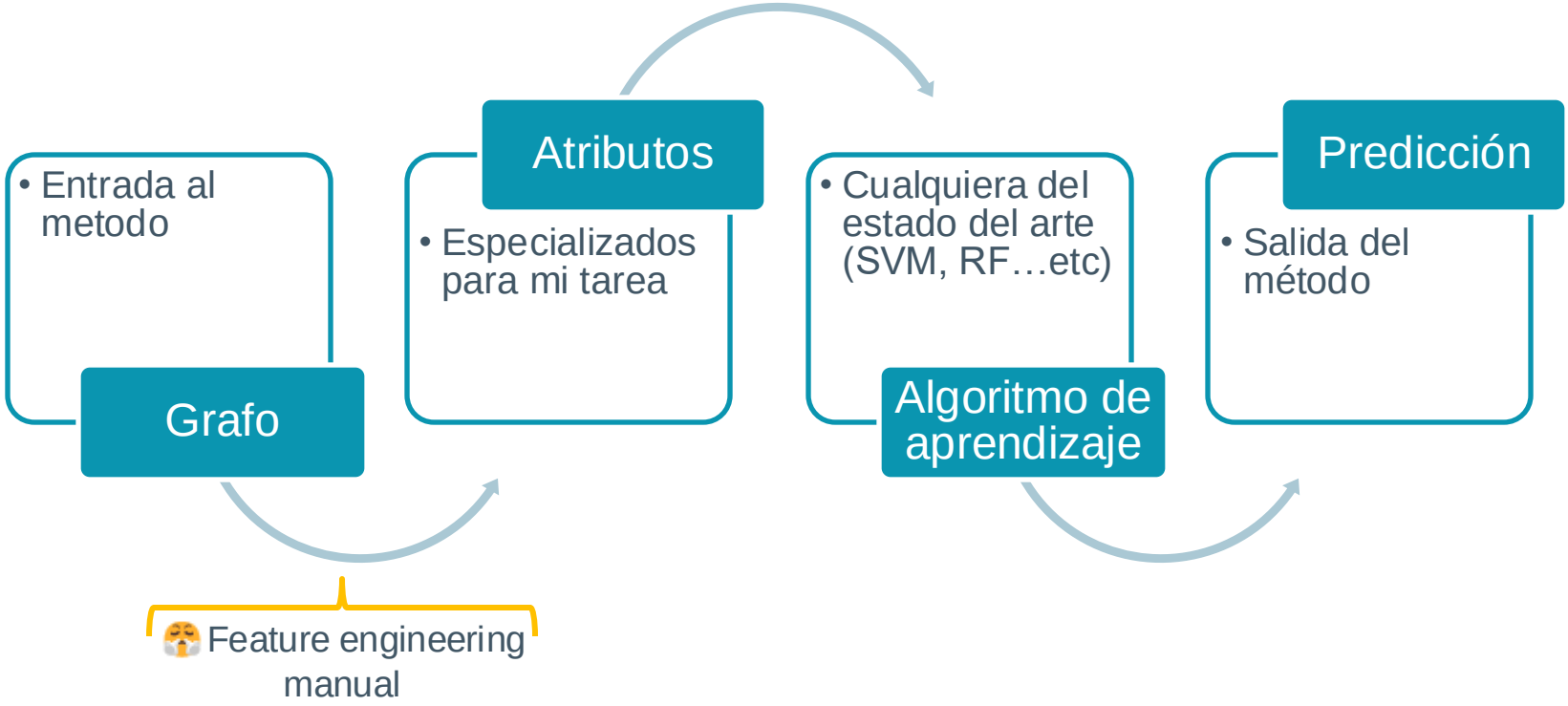
Evolución

Pipeline tradicional
Se crean atributos adhoc para cada problema de manera manual

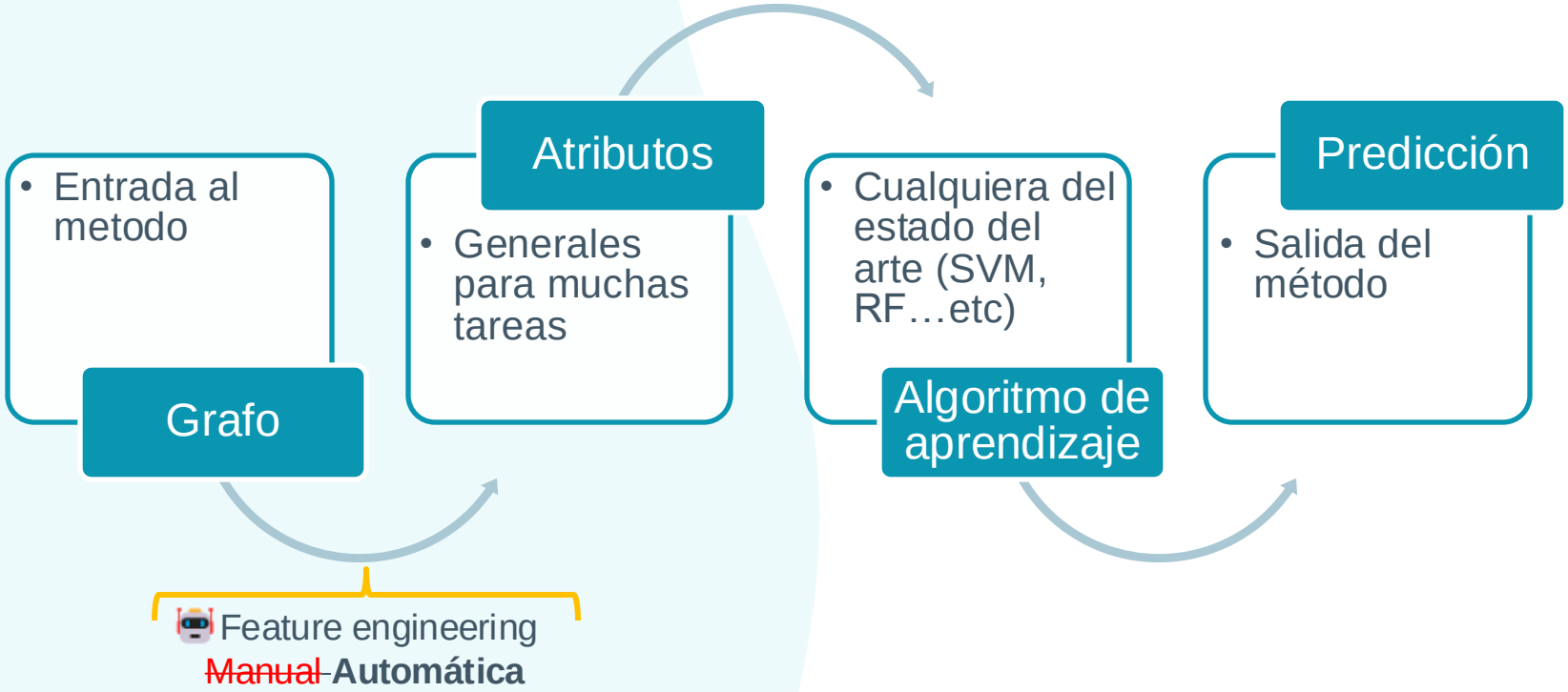
Representational learning
Se crean métodos capaces de aprender atributos agnosticos a las tareas de manera automática

GNNs
Son capaces de crear atributos adhoc para un problema especifico de manera automática

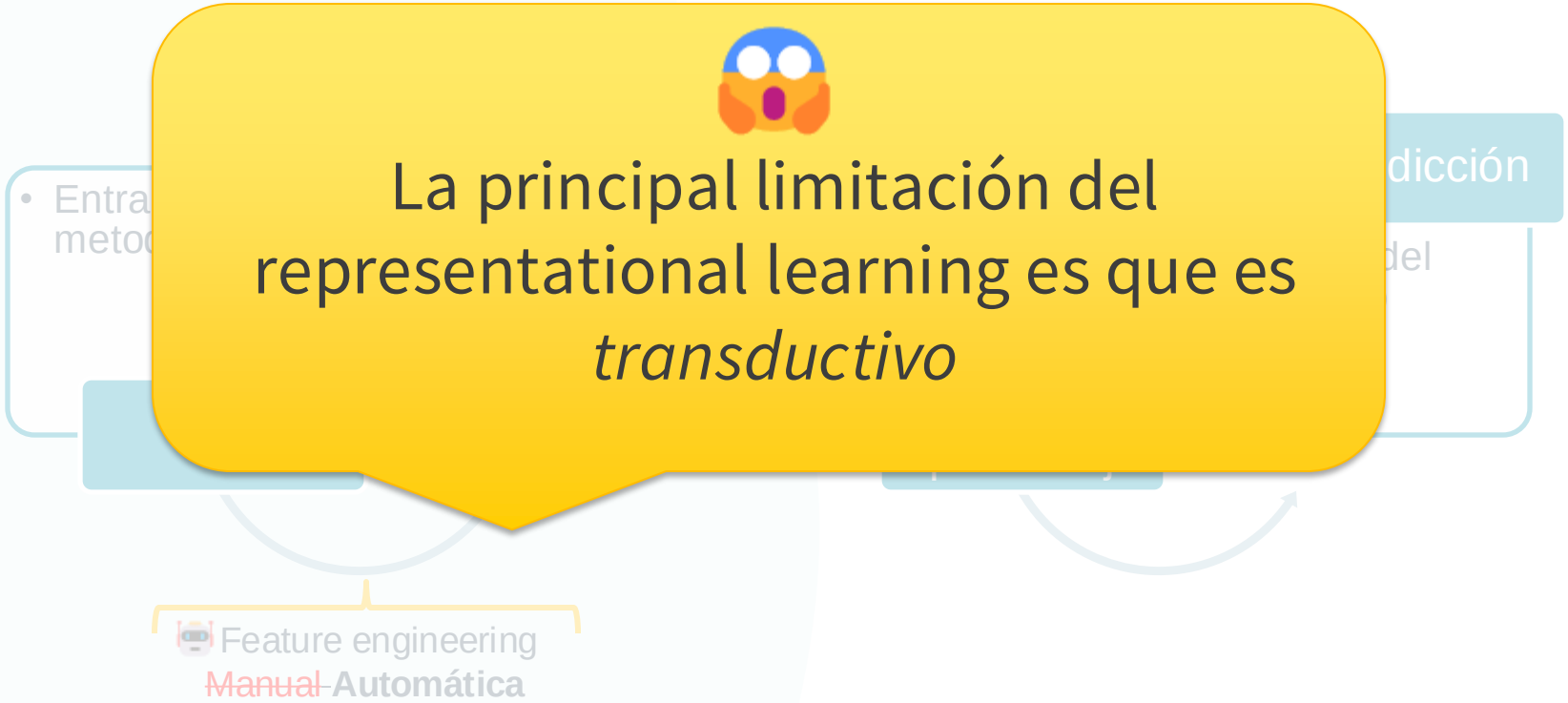
Pipeline tradicional



Representational learning

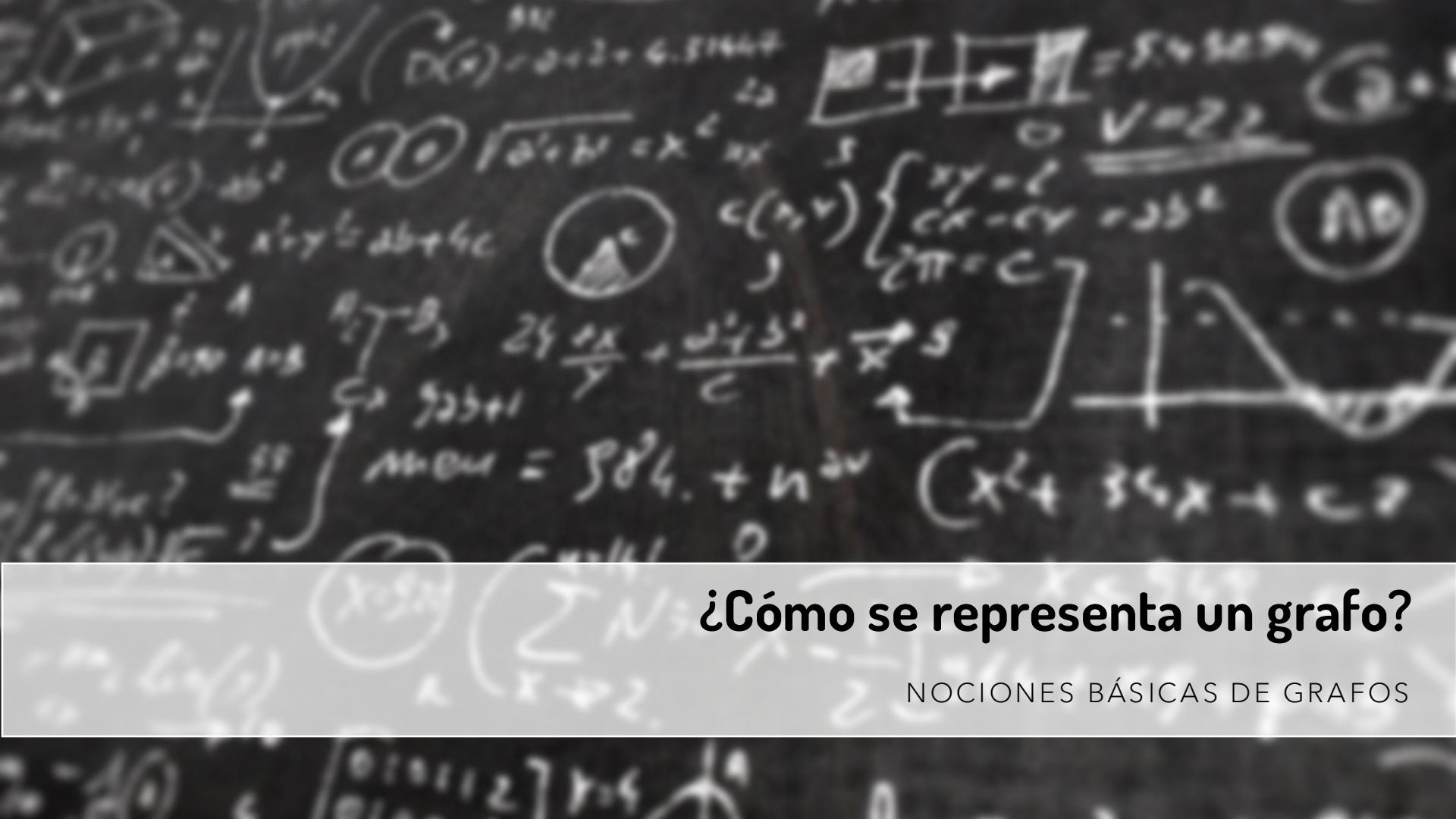


Representational learning



GNNs


• Entrada al metodo
Grafo
Capa paso de mensajes
• Crean representaciones específicas para el problema
• Usan la representación para realizar la tarea
Capa clasificación
Predicción
• Salida del método



¿Cómo se representa un grafo?

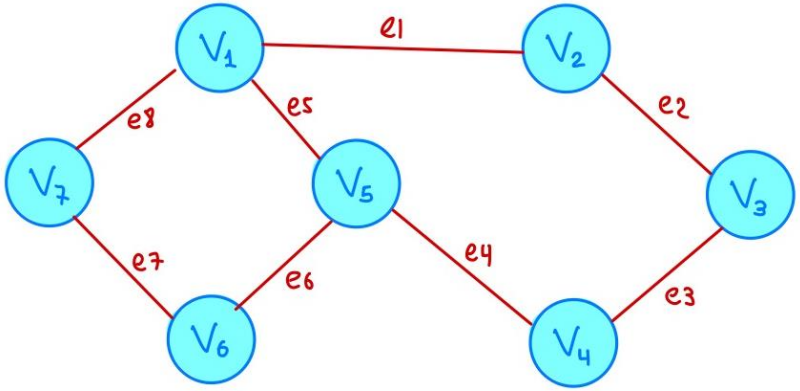
NOCIONES BÁSICAS DE GRAFOS

Nomenclatura

- Grafo: G(V,E)
  - Nodos: V = $\{v_0, \ldots, v_N\}$
  - Nº de nodos => N = |V|
  - Aristas: E = {(u,v) | u∈V ∧ v∈V}

Nomenclatura

- Grafo: G(V,E)
  - V = $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$
  - N = 7
  - E = $\{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_1), (v_5, v_6), (v_6, v_7), (v_7, v_1)\}$



Cómo represento un grafo

- Grafo: G(V,E)
  - X: Matriz de atributos
    - (N x NºAtributos)
  - A: Matriz de adyacencia
    - (N x N)
    - $A_{ij} = \begin{cases} 0 \; si \; e_{ij} \notin E \\ 1 \; si \; e_{ij} \in E \end{cases}$

Cómo represento un grafo

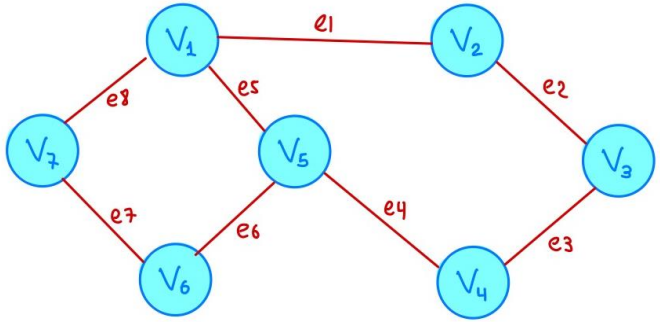
- Grafo: G(V,E)

$A =$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	0	1	0	1
v_2	1	0	1	0	0	0	0
v_3	0	1	0	1	0	0	0
v_4	0	0	1	0	1	0	0
v_5	1	0	0	1	0	1	0
v_6	0	0	0	0	1	0	1
v_7	1	0	0	0	0	1	0

$X =$

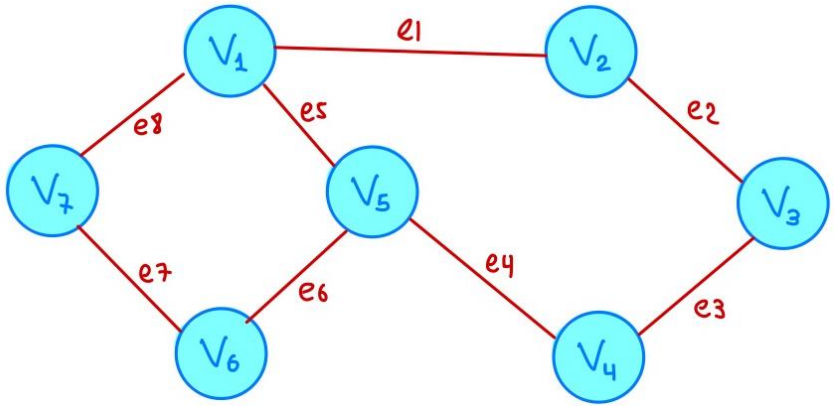
	x_1	x_2
v_1	1	1
v_2	2	2
v_3	3	3
v_4	4	4
v_5	5	5
v_6	6	6
v_7	7	7



Grado de un nodo

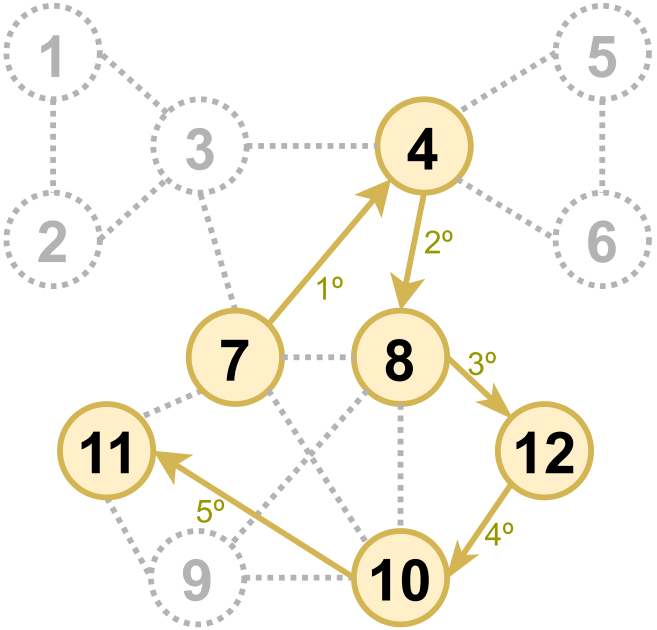
- Nº de enlaces que tiene un nodo
  - $d_i = \sum_{j=0}^{N} A_{ij}$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	d
v_1	0	1	0	0	1	0	1	3
v_2	1	0	1	0	0	0	0	2
v_3	0	1	0	1	0	0	0	2
v_4	0	0	1	0	1	0	0	2
v_5	1	0	0	1	0	1	0	3
v_6	0	0	0	0	1	0	1	2
v_7	1	0	0	0	0	1	0	2



Caminos en un grafo

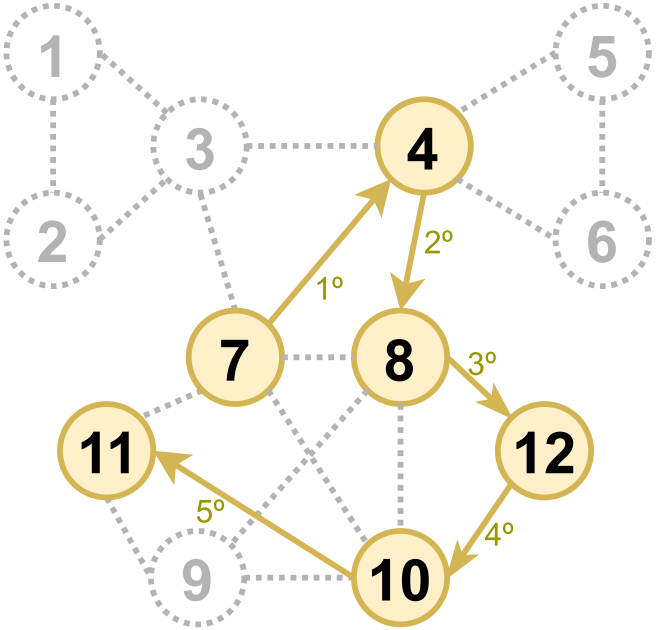
- Un **camino** es una secuencia de aristas donde cada elemento de la secuencia comparte una arista con el siguiente y el anterior nodo de la secuencia.
- Un **camino aleatorio** o “Random walk”, es un camino dónde cada arista de la secuencia se elige al azar.



Máster
Deep Learning

Caminos en un grafo

- Se dice que dos nodos n_i y n_j son **accesibles** o “reachables”, si existen al menos un camino que partiendo de n_i termine en n_j
- La **distancia (geodésica)** entre dos nodos n_i y n_j se define como el camino de menor longitud que empieza en n_i y termina en n_j.

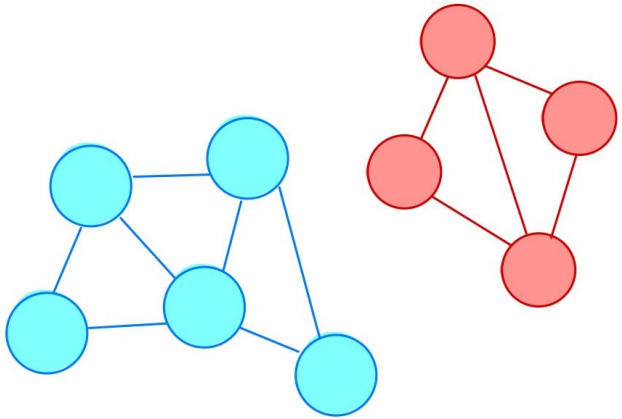


Caminos y la matriz de adyacencia

- Las potencias de la matriz de adyacencia me indican el número de caminos de longitud N que hay entre dos nodos.
- La distancia más larga que puede existir entre dos nodos en un grafo es N -1
- $\sum_{i=1}^{N-1} A^i$ Me indica si dos nodos son alcanzables o no.

Componentes conexas

- **Subgrafo:** $S \subseteq G \Rightarrow S(V', E')$ es un **subgrafo** de $G(V, E)$ si $V' \subseteq V \wedge E' \subseteq E$
- Una **componente conexa (CC)** de G es un Subgrafo de G donde todos los pares de nodos son accesibles entre sí y no se pueden añadir más nodos de G sin que se deje de cumplir la propiedad.

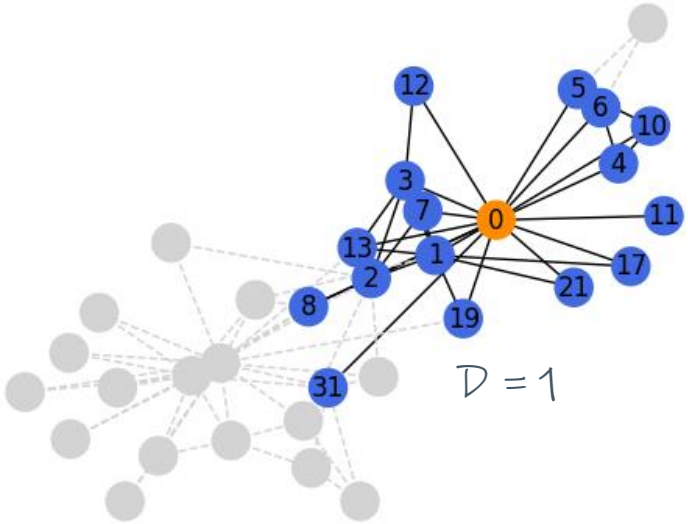


Componentes conexas

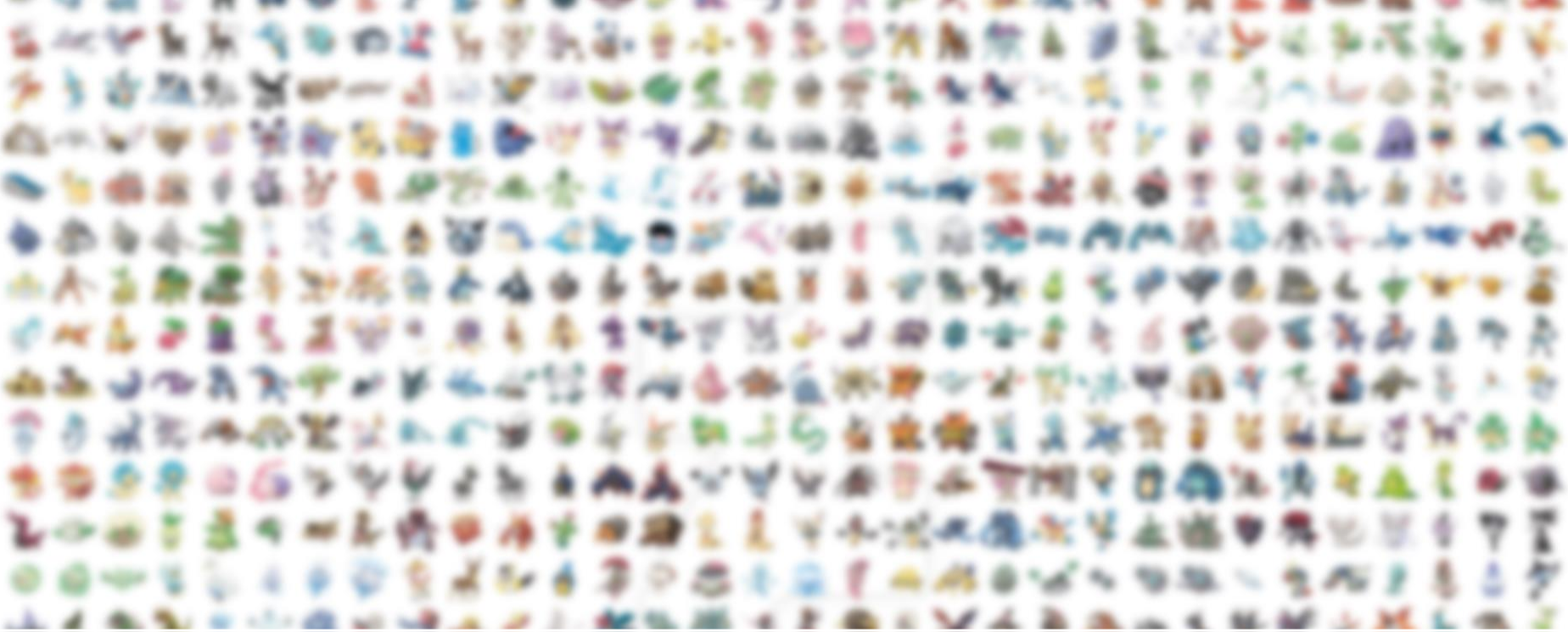
- **Su**
  ur
  V
- U
  G
  lo
  entre *u* y *v* existan más
  nodos de G sin que se deje de
  cumplir la propiedad.

Un grafo con una única componente conexa se le conoce como ***grafo conexo***

Ego Network



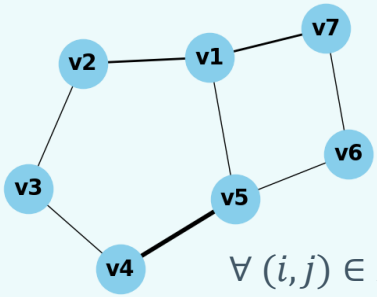
- Subgrafo generado sobre un nodo u que contienen todos sus vecinos a una distancia D y las aristas entre ellos.



Tipos de grafos

EL ZOO DE LOS GRAFOS

Ponderado

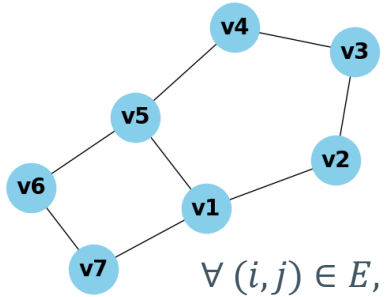


$\forall\,(i,j) \in E, A_{ij} \geq 1$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	2	0	0	1	0	2
v_2	2	0	1	0	0	0	0
v_3	0	1	0	1	0	0	0
v_4	0	0	1	0	3	0	0
v_5	1	0	0	3	0	1	0
v_6	0	0	0	0	1	0	1
v_7	2	0	0	0	0	1	0

VS

No Ponderado

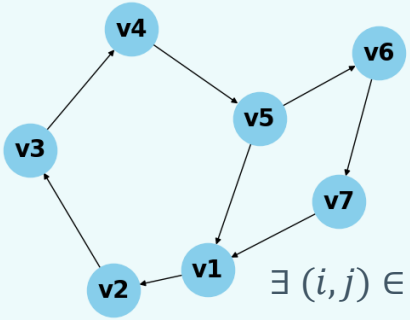


$\forall\,(i,j) \in E, A_{ij} = 1$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	0	1	0	1
v_2	1	0	1	0	0	0	0
v_3	0	1	0	1	0	0	0
v_4	0	0	1	0	1	0	0
v_5	1	0	0	1	0	1	0
v_6	0	0	0	0	1	0	1
v_7	1	0	0	0	0	1	0

Máster
Deep Learning

Dirigido

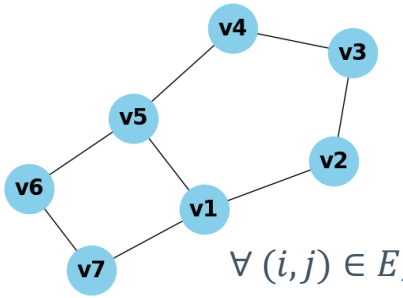


$\exists\ (i,j) \in E, A_{ij} \neq A_{ji}$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	0	0	0	1
v_2	0	0	0	0	0	0	0
v_3	0	1	0	0	0	0	0
v_4	0	0	1	0	1	0	0
v_5	1	0	0	0	0	1	0
v_6	0	0	0	0	0	0	1
v_7	0	0	0	0	0	0	0

VS

No Dirigido



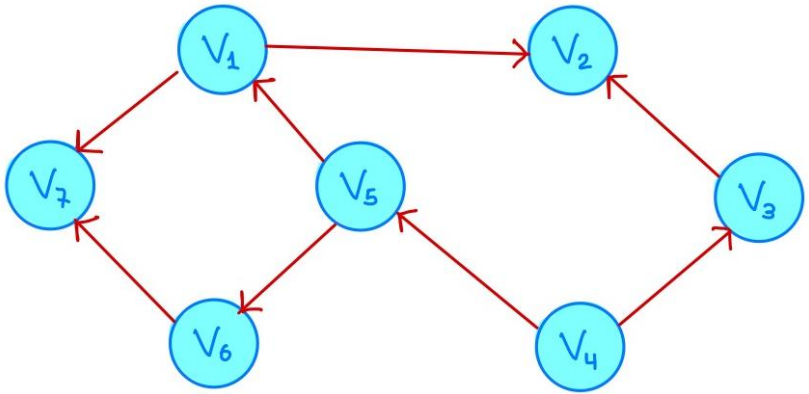
$\forall\ (i,j) \in E, A_{ij} = A_{ji}$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
v_1	0	1	0	0	1	0	1
v_2	1	0	1	0	0	0	0
v_3	0	1	0	1	0	0	0
v_4	0	0	1	0	1	0	0
v_5	1	0	0	1	0	1	0
v_6	0	0	0	0	1	0	1
v_7	1	0	0	0	0	1	0

Grado en grafos dirigidos

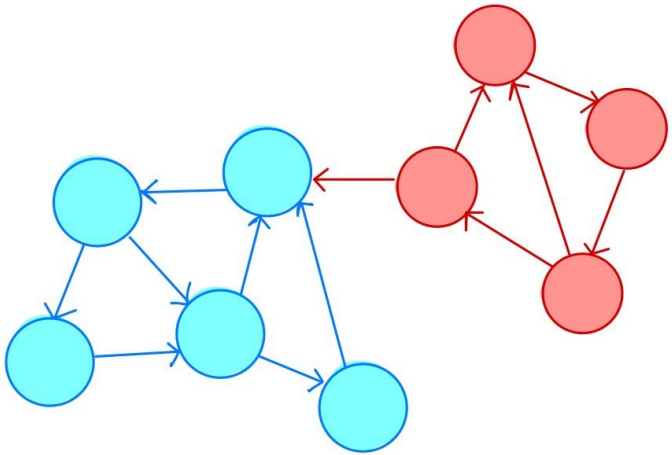
- In degree: $d^{in}(x) = \sum_{i=0}^{N-1} A_{xi}$
- Out degree: $d^{out}(x) = \sum_{i=0}^{N-1} A_{ix}$

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	d^{out}
v_1	0	1	0	0	0	0	1	2
v_2	0	0	0	0	0	0	0	0
v_3	0	1	0	0	0	0	0	1
v_4	0	0	1	0	1	0	0	2
v_5	1	0	0	0	0	1	0	2
v_6	0	0	0	0	0	0	1	1
v_7	0	0	0	0	0	0	0	0
d^{in}	1	2	1	0	1	1	2	



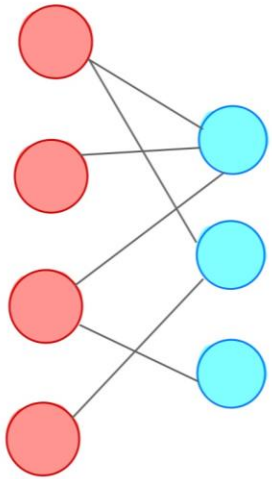
Componentes conexas en grafos dirigidos

- En grafos dirigidos existen dos variantes la **fuerte** y la **débil:**
- En la **fuerte**, para cada par de nodos n_i y n_j debe haber un camino tanto de n_i a n_j , como de n_j y n_i.
- En la **débil** basta con que existe al menos un camino, de n_i y n_j o de n_j y n_i



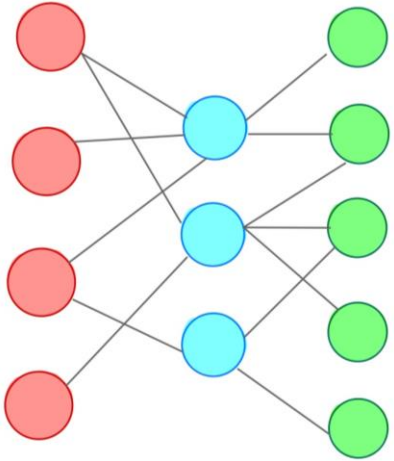
Grafos k-partidos

- Tengo k tipos de nodos
- Tengo k-1 tipos de relación
- Las relaciones solo unen nodos de tipos distintos.
- Todos los tipos de nodos tiene algún tipo de relación

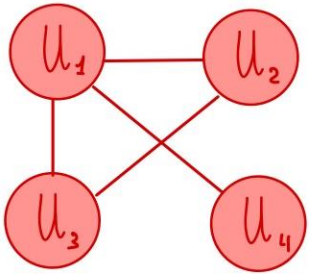


Bi-partito

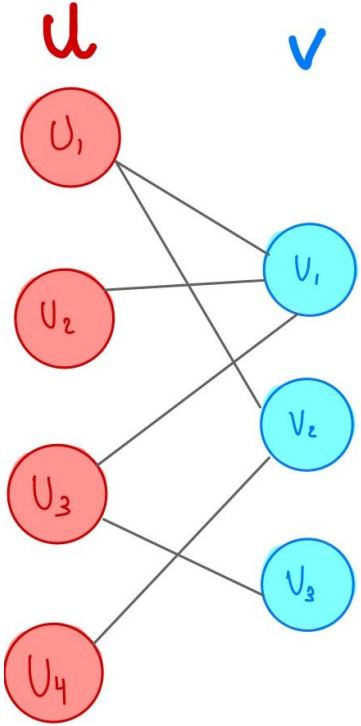
tri-partito



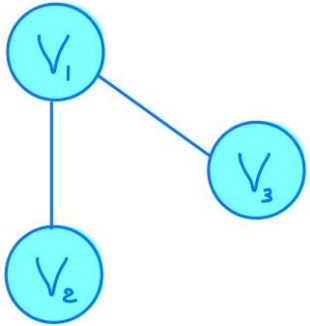
Proyecciones en grafos partidos



Proyección en U



Bi-partito

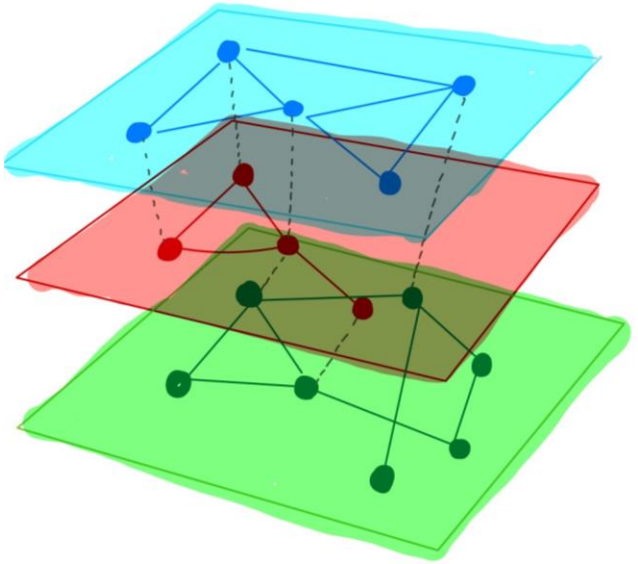


Proyección en V

Grafos multicapa

- Los nodos y las aristas tienen diferentes tipos.
- *Puedo tener conexiones en la misma capa y conexiones entre capas*

	v_1	v_2	v_1	v_2	v_3	v_1	v_4
v_1	0	1	0	0	0	0	1
v_2	1	0	0	0	1	0	0
v_1	0	0	0	1	0	0	0
v_2	0	0	1	0	1	0	0
v_3	0	1	0	1	0	0	0
v_1	0	0	0	0	0	0	1
v_4	1	0	0	0	0	1	0



Grafos multicapa

- Lo
- *Pu* *capas*

Los grafos k-partidos se usan para almacenar conocimiento en forma de hechos

v_1							
v_2							
v_1							
v_2							
v_3	0	1	0	1			
v_1	0	0	0	0	0	0	1
v_4	1	0	0	0	0	1	0

Prácticar

- Hacer grupos y pensar que tipos de grafos usarias en cada caso.
- Tienes las instrucciones detalladas en la hoja de ejercicios ""

Social Network Análisis (SNA)

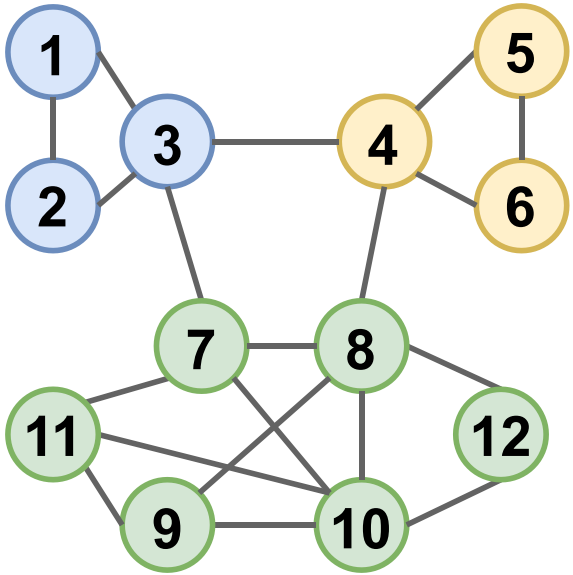
TAREAS CLÁSICAS QUE PODEMOS RESOLVER USANDO GRAFOS

Metricas para nodos: Centralidad

- Miden como de importante es un nodo en la red
- Destaca el *page rank* (o algoritmo de google) que indica los nodos que están conectados a otros nodos bien conectados.

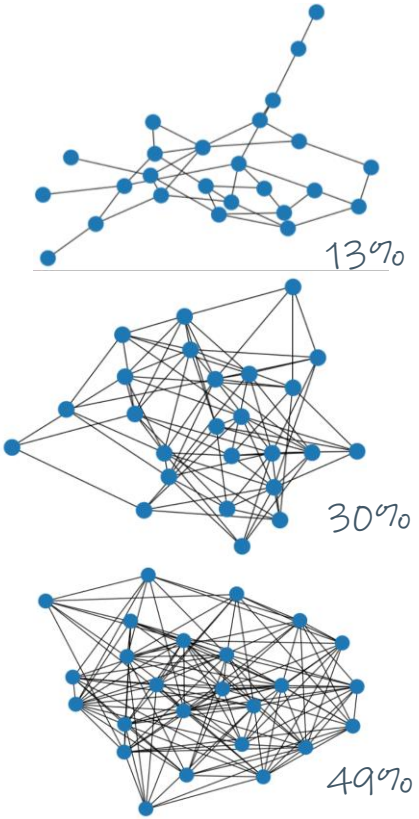
Detección de comunidades

- Tarea que consiste en dividir un grafo en grupos de nodos con más conexiones entre los nodos del grupo comparados con el resto de nodos de la red.
- Cabe destacar el algoritmo de *Louvain*.



Densidad de un grafo

- El número máximo de aristas de un grafo viene fijado por su número de nodos.
  - Dirigidos : $N * (N - 1)$
  - No-Dirigidos: $\frac{N*(N-1)}{2}$
- La densidad de un grafo se mide como el porcentaje de aristas que tiene del total possible.



¿Qué librerías vamos a utilizar?

NETWORKX Y PYTORCH GEOMETRIC

Pytorch geometric (PyG)

- Libreria basada en pytroch que permite hacer deep learning sobre grafos.
- Framework potente que permite implementar muchos tipos de arquitecturas con poco esfuerzo.
- Tiene un sinfin de arquitecturas ya programadas y lista para usar.

https://pytorch-geometric.readthedocs.io/en/latest/

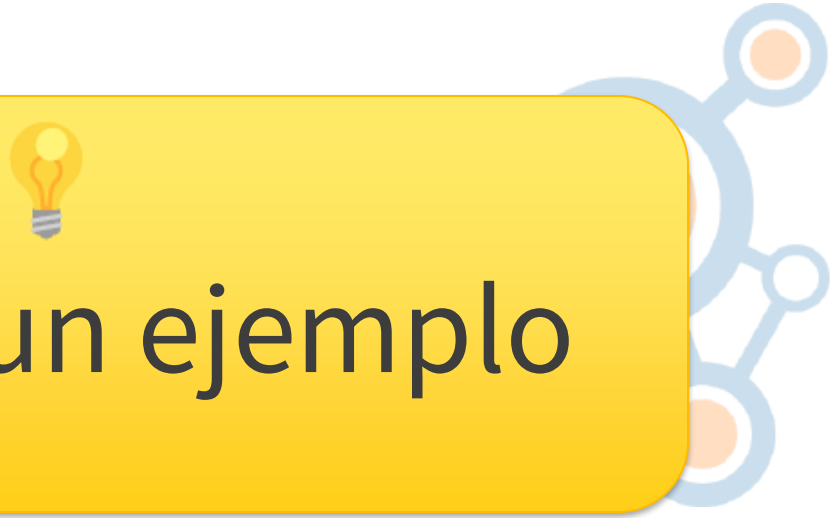
NetworkX

- Librería para la creación, manipulación y análisis de grafos.
- Proporciona funciones para calcular métricas de grafos, encontrar caminos y detectar comunidades.
- Ampliamente utilizada tanto en análisis como en la visualización de grafos.

https://networkx.org/documentation/stable/index.html

NetworkX

- Librería para la creación, ma
- Pro cal en co
- Ampliamente utilizada tanto en análisis como en la visualización de grafos.



Veamos un ejemplo

https://networkx.org/documentation/stable/index.html

¿Qué representa mi grafo?

EL MODELO DEBAJO DE LAS MATEMÁTICAS

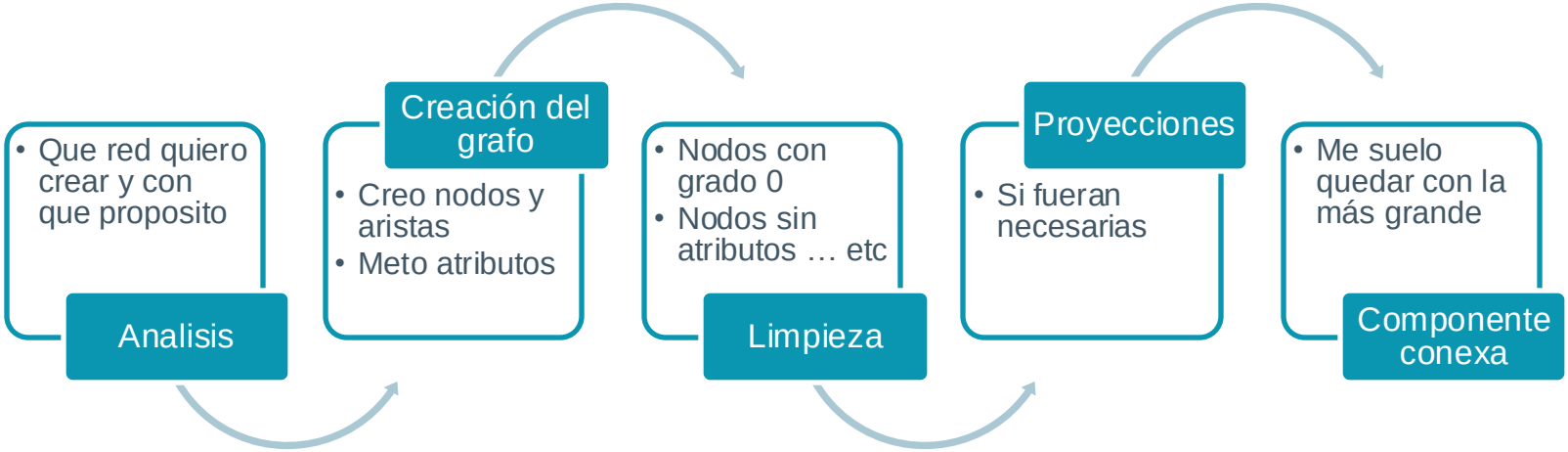
Escoger la representación adecuada

Siempre tenemos que pensar que representa nuestra red.

- **Red profesional:** Conecto individuos que trabajan juntos.
- **Red sexual:** Conecto individuos que han tenido relaciones sexuales.
- **Red de citas científicas**: Conecto papers que se citan uno al otro.
- **Red ASOIAF:** Conecto personajes que aparecen a menos de 15 palabras de otro personaje en los libros de “canción de hielo y fuego”.

https://github.com/mathbeveridge/asoiaf

Proceso de creación de un grafo



Prácticar

- Vamos a crear un grafo a partir de datos en bruto, que es el proceso que tengo que hacer antes de realizar cualquier tarea de machine learning sobre grafos.
- Tienes toda la información el la hoja de ejercicios "".