

# Trabajo Práctico N°4: Aprendizaje no supervisado

Grupo 10



# Problema - Dataset



# Problema - Estandarización

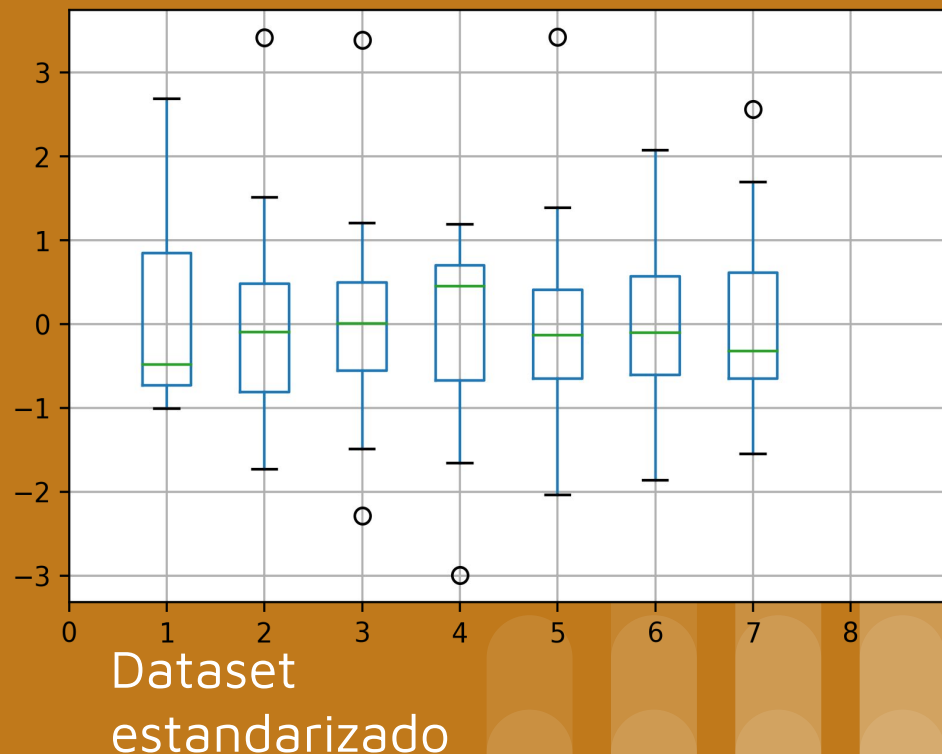
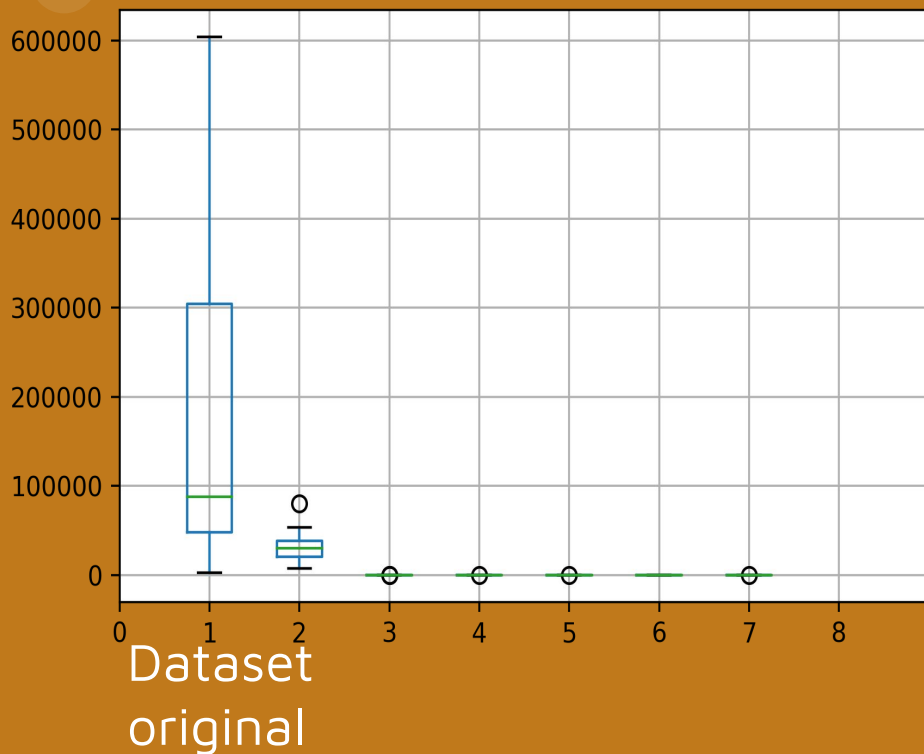
	Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0	Austria	83871	41600	3.5	79.91	0.80	0.03	4.2
1	Belgium	30528	37800	3.5	79.65	1.30	0.06	7.2
2	Bulgaria	110879	13800	4.2	73.84	2.60	-0.80	9.6
3	Croatia	56594	18000	2.3	75.99	2.39	-0.09	17.7
4	Czech Republic	78867	27100	1.9	77.38	1.15	-0.13	8.5

Dataset  
original

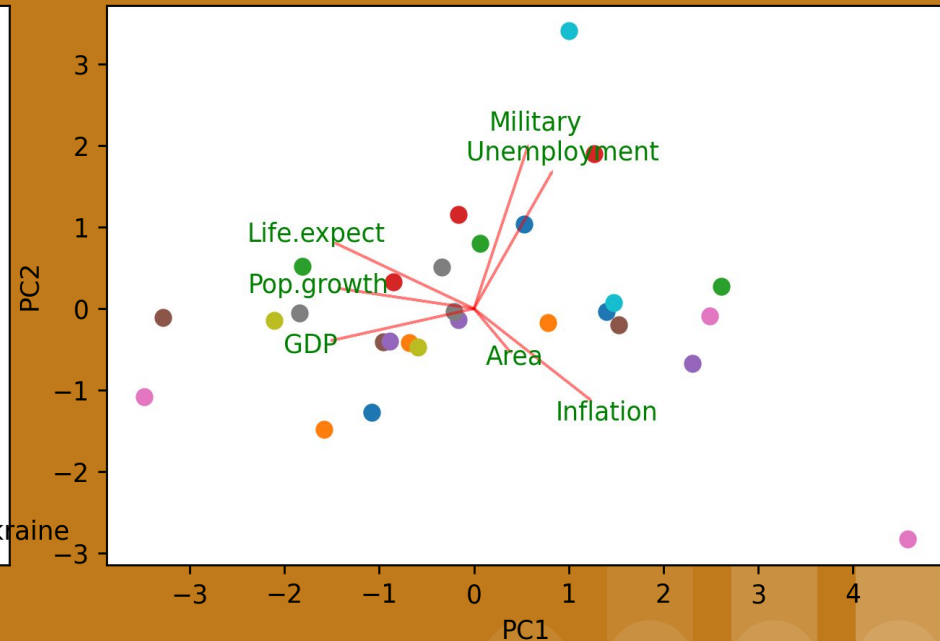
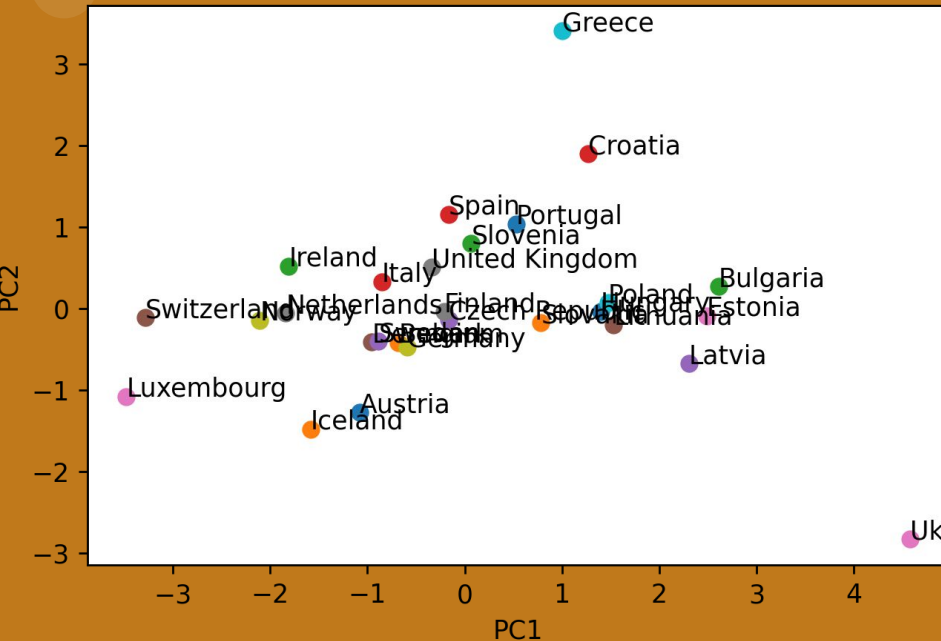
	Country	Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0	Austria	-0.507835	0.683900	0.114447	0.570778	-1.024347	-0.176789	-1.245527
1	Belgium	-0.835987	0.417061	0.114447	0.487756	-0.388952	-0.115927	-0.592442
2	Bulgaria	-0.341689	-1.268238	0.624255	-1.367473	1.263074	-1.860631	-0.069973
3	Croatia	-0.675636	-0.973310	-0.759511	-0.680942	0.996208	-0.420236	1.693357
4	Czech Republic	-0.538618	-0.334301	-1.050830	-0.237093	-0.579571	-0.501385	-0.309438

Dataset  
estandarizado

# Problema - Estandarización



# Problema - Primera componente



Primera componente frente a  
segunda

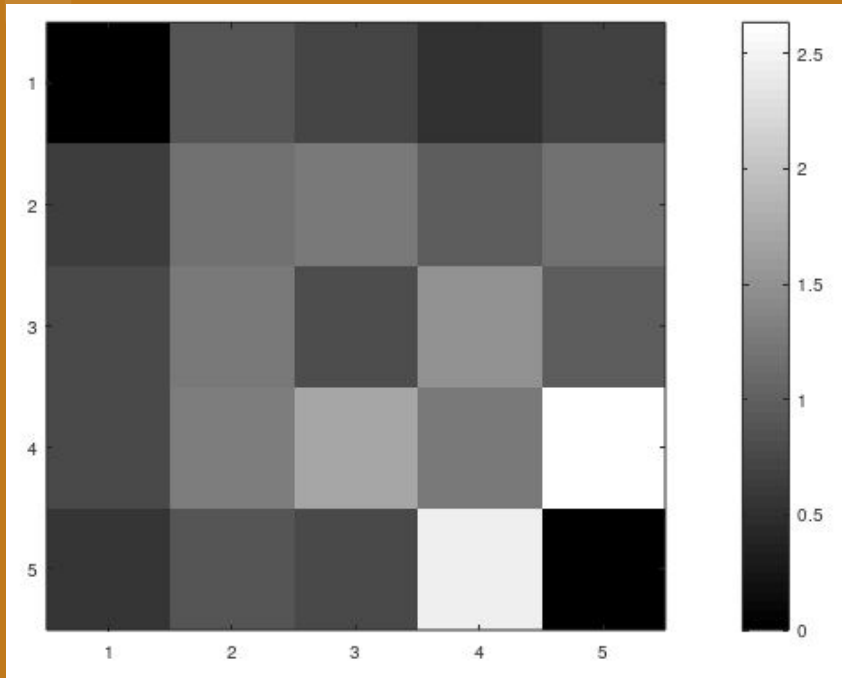
# Red de Kohonen



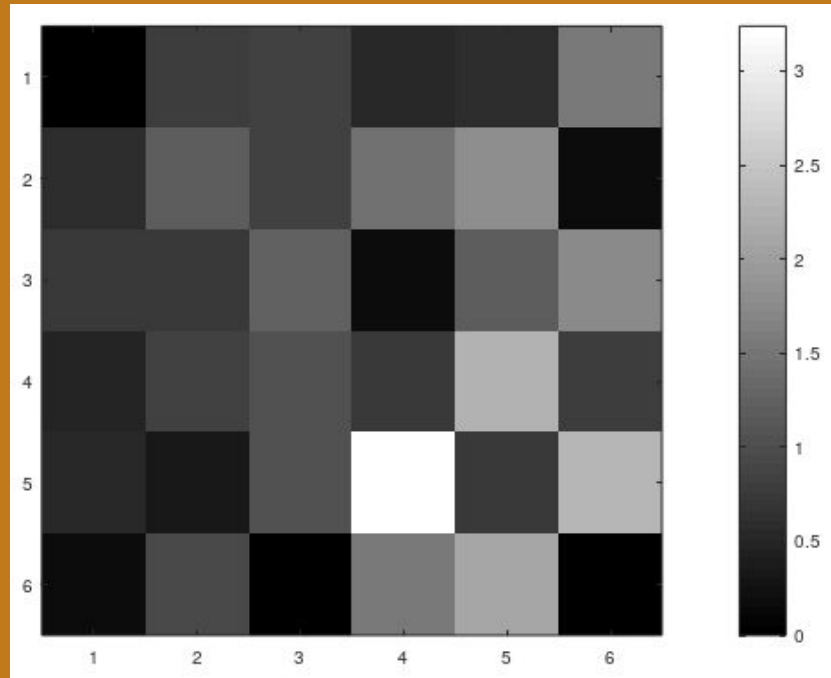
# Red de Kohonen - Implementación

- Radio inicial y número de iteraciones en función del tamaño de red.
- Inicialización de pesos con valores de entrada
- Radio disminuye tras cierta cantidad de iteraciones hasta el mínimo.
- $\eta$  adaptativo en función de las iteraciones ( $1/it$ ), decreciente.
- Tamaño grilla 5x5.

# Red de Kohonen



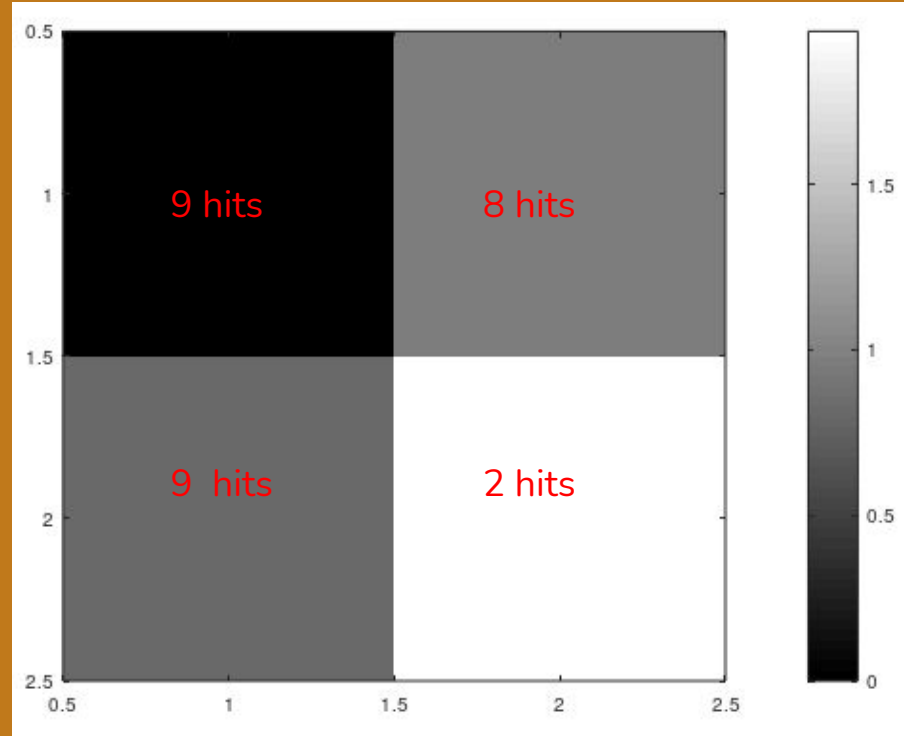
Matriz U: red 5x5



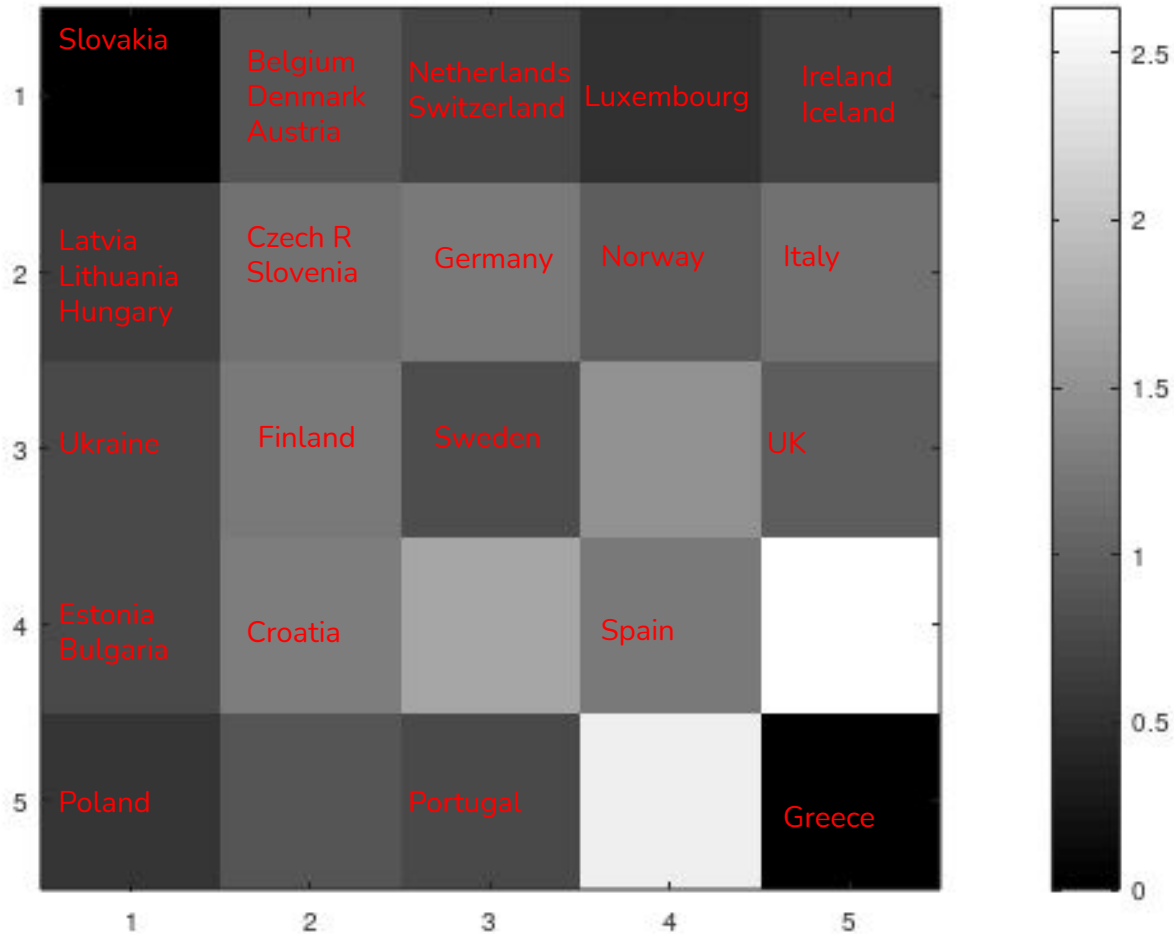
Matriz U: red 6x6



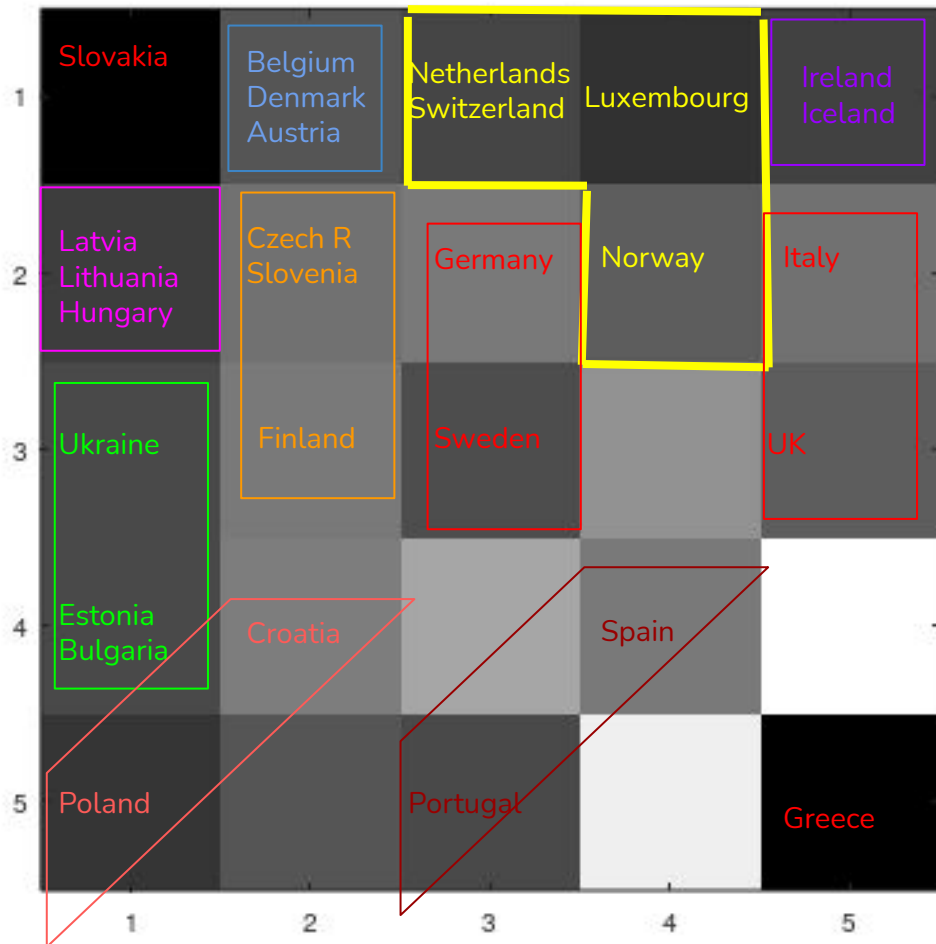
# Red de Kohonen



Matriz  $U$  red 2x2 con hits finales



Matriz U y hits  
finales



-3.478434956704641 Luxembourg  
 -3.2815861288159915 Switzerland  
 -2.1065108336056704 Norway  
 -1.8400534050753625 Netherlands  
 -1.808917610939201 Ireland  
 -1.583719702463114 Iceland  
 -1.081747661209592 Austria  
 -0.9551907971097199 Denmark  
 -0.8851053096517608 Sweden  
 -0.853223962394838 Italy  
 -0.681094068633844 Belgium  
 -0.5923936529244829 Germany  
 -0.34081935066733804 United Kingdom  
 -0.2105631563634406 Finland  
 -0.1672094937905111 Czech Republic  
 -0.16376695684389816 Spain  
 0.06754337538258412 Slovenia  
 0.5264933349515386 Portugal  
 0.7829659710814719 Slovakia  
 1.0004719569396427 Greece  
 1.2701488485858352 Croatia  
 1.3968983127833716 Hungary  
 1.4717738276291152 Poland  
 1.530099909952709 Lithuania  
 2.3060594085931676 Latvia  
 2.487735218104109 Estonia  
 2.609878816420479 Bulgaria  
 4.580268066769382 Ukraine

# Red de Kohonen - Conclusiones

- En general, países con valores similares caen en la misma neurona (ej Irlanda e Islandia).
  - Se pudo agrupar a los países con neuronas vecinas, asimilando el ordenamiento de la primera componente.
  - Algunos países con valores similares quedan más alejados, (ej. Bélgica e Italia) posiblemente por la manera de inicializar los pesos.
- El tamaño de la red puede obtenerse a través de prueba y error, para minimizar la distancia promedio pero mantener agrupaciones.

# Regla de Oja



# Regla de Oja - Implementación

- Datos estandarizados
- $\eta$  definido dentro de 0.0001 y 0.001
- Inicialización de los pesos uniforme dentro de  $[0,1]$

# Regla de Oja

PCA según la librería sklearn

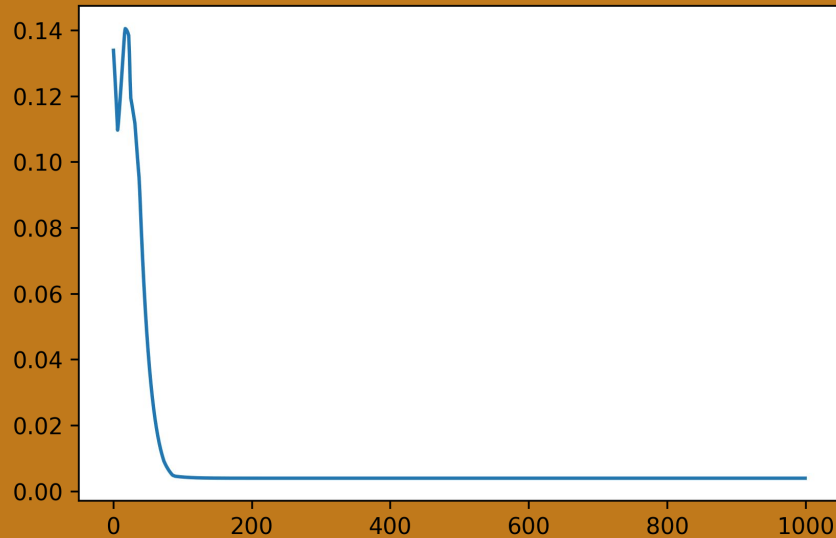
Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
0.124874	-0.500506	0.406518	-0.482873	0.188112	-0.475704	0.271656

Regla de Oja, con 1000 epochs y  $\eta=0.0001$

Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
-0.128375	0.499834	-0.409759	0.483279	-0.185001	0.474607	-0.269718

# Regla de Oja

Error medio, según los epochs con  $\eta=0.001$

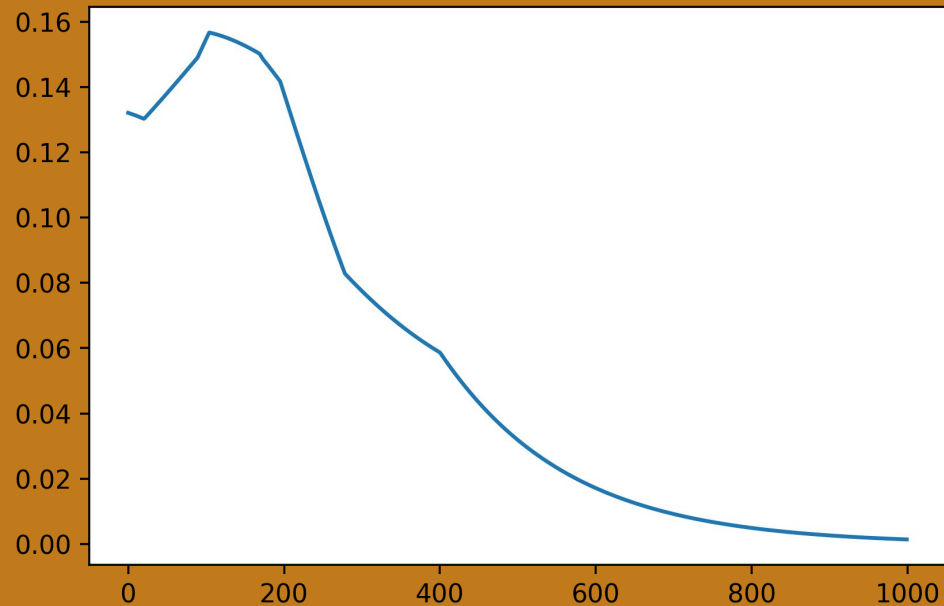


Area	GDP	Inflation	Life.expect	Military	Pop.growth	Unemployment
-0.131902	0.499125	-0.412998	0.483684	-0.181898	0.473477	-0.267751



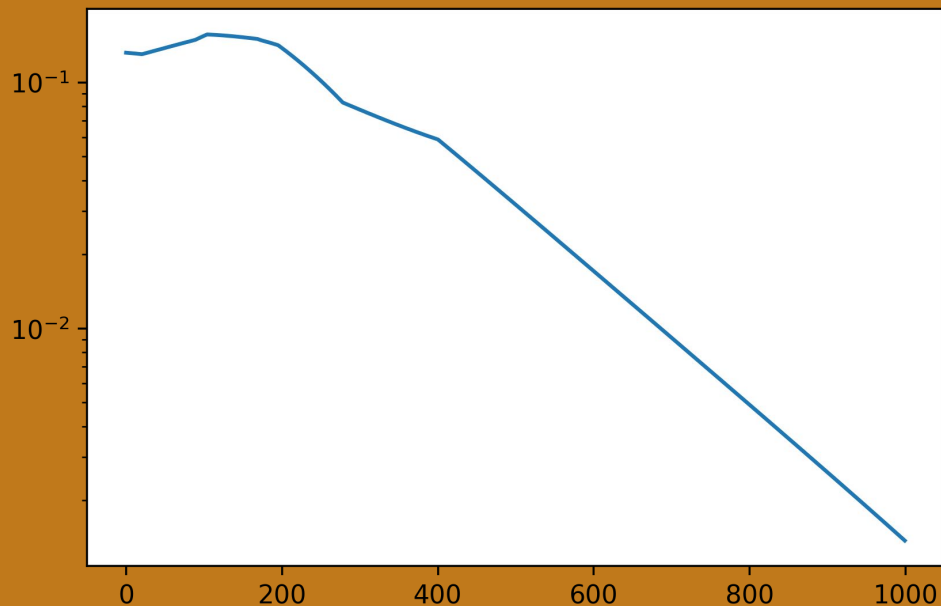
# Regla de Oja

Error medio, según los epochs con  $\eta=0.0001$



# Regla de Oja

Error medio, según los epochs con  $\eta=0.0001$  (en escala log)



# Regla de Oja - Conclusiones

- Utilizando la Regla de Oja, se logró llegar a una aproximación muy similar al cálculo de la primera componente.
- Es necesario estandarizar los datos para obtener una buena aproximación de la PC, utilizando librerías y también en el caso de la red neuronal.

# Red de Hopfield



# Red de Hopfield

-Conjunto de 4 letras de tamaño 5x5

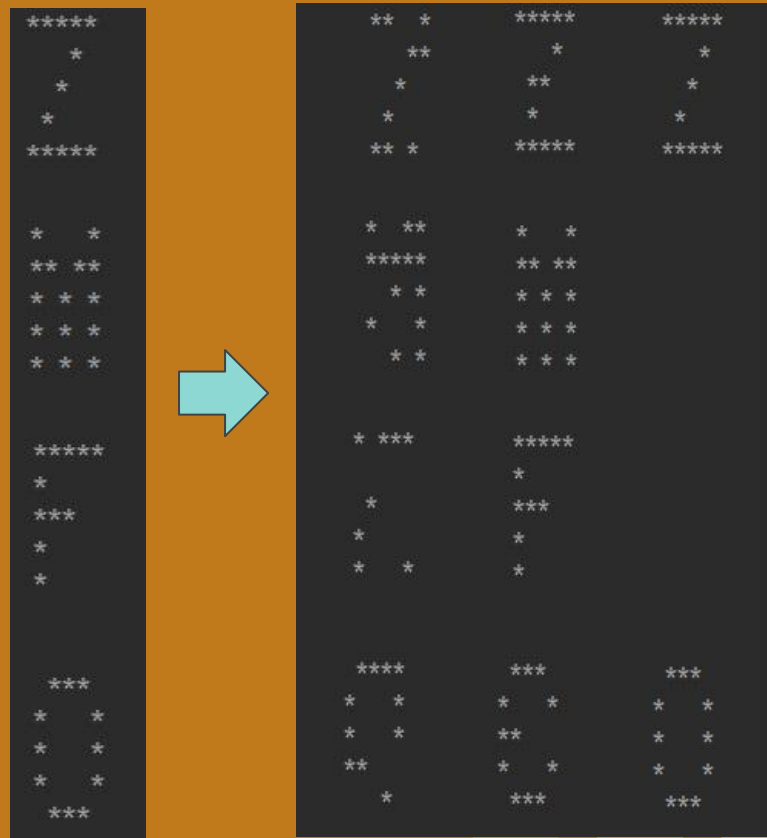


-Elección de conjunto en base a la ortogonalidad

$$\sum_{A=1}^{N-1} \sum_{B=A+1}^N |A \cdot B|$$

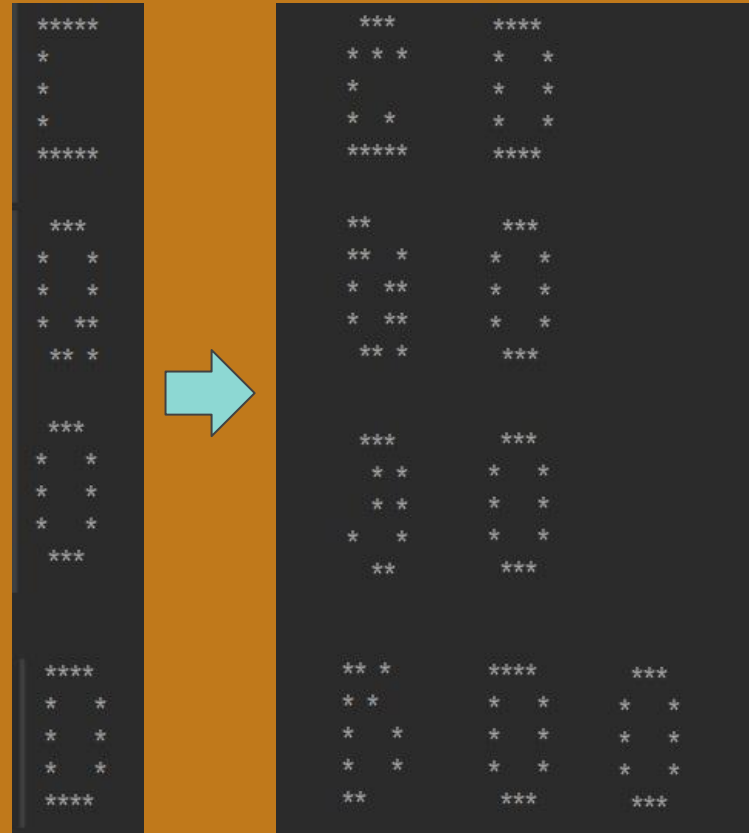
# Red de Hopfield

- Conjunto Z-M-F-O
- Ortogonalidad: 14
- 20% ruido



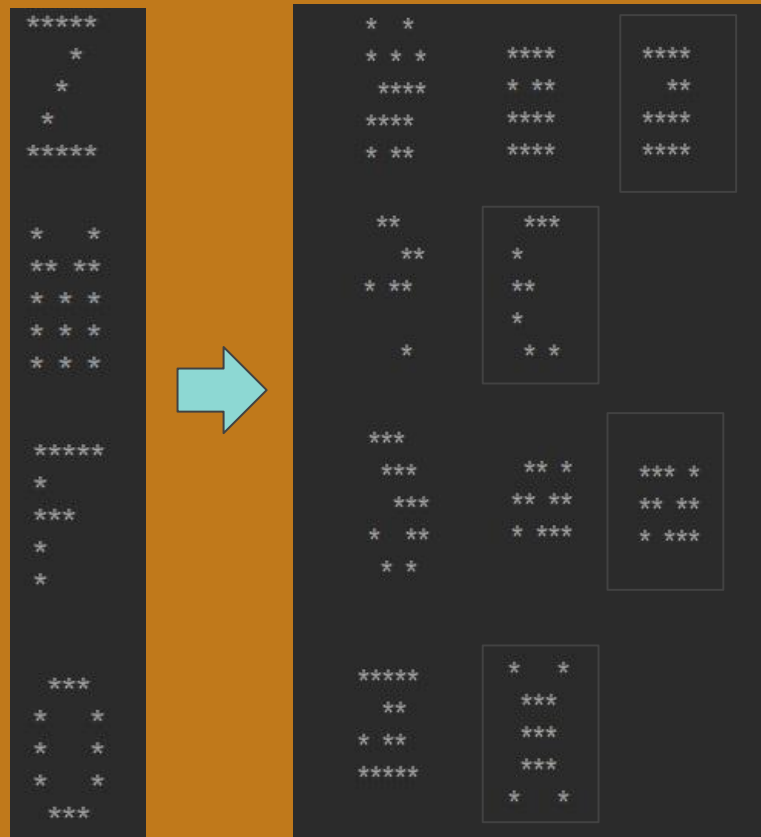
# Red de Hopfield

- Conjunto C-Q-O-D
- Ortogonalidad: 90
- 20% ruido



# Red de Hopfield

- Estados espúreos
- Conjunto Z-M-F-O
- 60% ruido





# Red de Hopfield - Conclusiones

- Con estados más o menos ortogonales, la red funciona correctamente.
- Añadir una mayor cantidad de ruido produce atracción a patrones espúreos.
- La ortogonalidad de los patrones puede afectar el resultado obtenido, o generar asociaciones incorrectas.
- Dependiendo de la distribución del ruido, aún con bajo porcentaje podría asociarse incorrectamente el patrón o llegar a un espúreo.