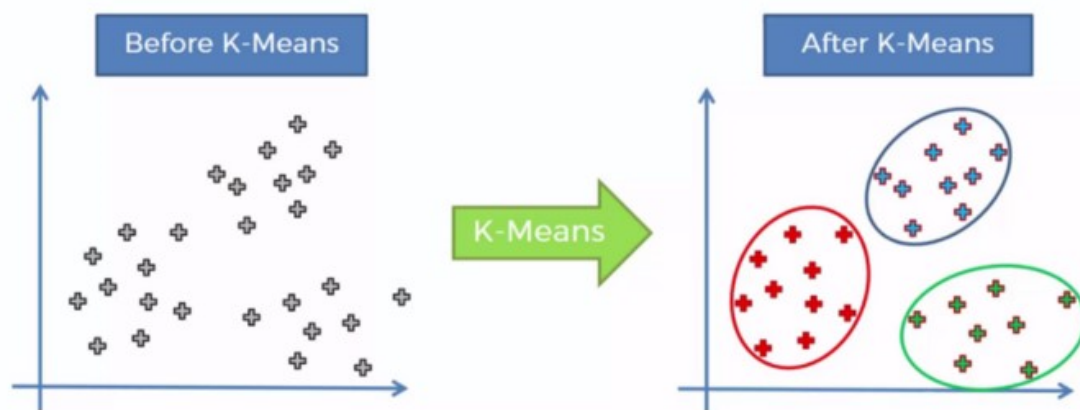


## Clustering con el Modelo k-means al Dataframe Forbes 2023

k-means es uno de los algoritmos de aprendizaje automático no supervisado más popular. El objetivo de k-means es simple: agrupa puntos de datos similares con el objetivo de descubrir patrones subyacentes. Para lograr este objetivo, k-means busca un número fijo (k) de agrupamientos (clústers) en el conjunto de datos.



Funcionamiento básico de k-means.

En k-means se define de inicio un número k, que se refiere al número de centroides en los que se dividirá el conjunto de datos. Cada centroide sería la ubicación que marca el centro de cada agrupación.

A cada punto se asigna uno de los grupos mediante la reducción de la suma de cuadrados en el grupo. Dicho de otra forma, el algoritmo k-means identifica k número de centroides, y luego asigna cada punto de los datos al grupo más cercano, mientras mantiene los centroides lo más pequeños posibles.

Una vez se tiene cada punto asociado a un clúster, se puede etiquetar en el dataframe original asociándolo a dicho grupo y "catalogando" por tanto los datos.

Datos de inicio: Dataframe forbes23.

Se va a aplicar el algoritmo sobre un conjunto de datos del Dataframe forbes 2023 revista de economía forbes para determinar cómo se agrupan usando k-means.

De dichos datos, se buscarán patrones de agrupamiento entre dos señales elegidas, en este caso se selecciona la señal SALES (ventas de las principales empresas en cifras en millones de dólares) y la señal ASSETS (ganancias de las principales empresas en cifras en millones de dólares). Se cuenta con un dataframe con un total de 2000 registros que ya pasaron por el proceso de limpieza.

NOTA: Se puede aplicar k-means sobre todas las señales/variables necesarias, pero para poder visualizar los agrupamientos k-means en una gráfica 2-D se aplica el algoritmo exclusivamente sobre las dos señales anteriormente detalladas.

En [ ]: *# Carga de librerías.*

```
import os # directorios de trabajo
import seaborn as sns
import numpy as np # metodos numericos y manipulación de array
from scipy.spatial import Distance
import pandas as pd # Data frame
import matplotlib.pyplot as plt # componente grafico
from sklearn.preprocessing import MinMaxScaler # Preprocesamiento para normaliz
de sklearn.cluster import KMeans # KMeans para realizar el clustering
```

En [ ]: *# Establecer directorio de trabajo e importar el Dataframe forbes23*

```
os . obtenercwd ()
```

Afuera[... 'c:\\Usuarios\\USUARIO\\OneDrive\\Documentos\\Uibero Ing de Software\\8 SEMESTRE  
\\Inteligencia\_artificial-main'

En [ ]: *# Mostrar Los Archivos que hay en el ditrectorio de trabajo*  
os . listadir ()

Afuera[... ['.git',  
'1. Introduccion.ipynb',  
'2. Numpy.ipynb',  
'3. PandasForbes.ipynb',  
'Aactividad 1.docx',  
'caracteristicas de vinos.csv',  
'cluster\_k-means.ipynb',  
'Explicación del modelo kmeans.ipynb',  
'Forbes',  
'forbes23.ipynb',  
'imagen-1.png',  
'imagen.png',  
'ipynb',  
'Mall\_Customers.csv',  
'Proyecto km significa cliente.ipynb',  
'pruebaforbes.ipynb']

En [ ]: *# Importación de La base de datos*

```
forbes23 = pd . read_csv ( 'Forbes\\Forbes2023.csv' , codificación = 'latin-1' ,
```

En [ ]: forbes23

Afuera[...

	Rango	nombre de empresa	País	Ventas\n(millones de dólares)	Beneficios\n(\$millones)	Activos\n(n de
0	1	JPMorgan Chase	Estados Unidos	179930	41800	
1	2	Compañía petrolera de Arabia Saudita (Saudi Aramco)	Arabia Saudita	589470	156360	
2	3	ICBC	Porcelana	216770	52470	
3	4	Banco de construccion de China	Porcelana	203080	48250	
4	5	Banco de Agricultura de China	Porcelana	186140	37920	
...	...	...	...	...	...	
1995	1996	Alfa Laval	Suecia	5350	489,5	
1996	1996	Brecha	Estados Unidos	15620	-202	
1997	1996	Si Banco	India	3340	91,6	
1998	1999	BEKB-BCBE	Suiza	556	167,1	
1999	1999	Ipsen	Francia	3180	681,7	

2000 filas × 7 columnas



```
En [ ]: # método que devuelve La dimensión del array en este caso serian 2000 empresas y 7
print ( forbes23 . shape )
# muestra las columnas del DataFrame.
imprimir ( forbes23 . columnas )
```

```
(2000, 7)
Index(['Rango', 'Nombre de la empresa', 'País', 'Ventas\n($millones)',
      'Beneficios\n($millones)', 'Activos\n($millones)',
      'Valor de mercado \nA partir del 05/05/23 ($m)'],
      dtype='object')
```

```
En [ ]: ### Proceso de Limpieza de Datos: Cambiar el nombre de las variables
forbes23 . columnas = [ 'Rango' , 'Empresa' , 'País' , 'Ventas' , 'Beneficios' ,
forbes23 . columnas
```

```
Afuera[... Index(['Rango', 'Empresa', 'País', 'Ventas', 'Beneficios', 'Activos',
'Valor de mercado'],
tipo d='objeto')
```

```
En [ ]: Forbes23 . head ( n = 10 ) # Top 10 de Las empresas de Forbes 2023
#forbes23.tail(n=9) #Las últimas 5
```

	Rango	Compañía	País	Ventas	Beneficios	Activos	Valor de mercado
<b>0</b>	1	JPMorgan Chase	Estados Unidos	179930	41800	3744300	399590
<b>1</b>	2	Compañía petrolera de Arabia Saudita (Saudi Aramco)	Arabia Saudita	589470	156360	660990	2055220
<b>2</b>	3	ICBC	Porcelana	216770	52470	6116820	203010
<b>3</b>	4	Banco de construccion de China	Porcelana	203080	48250	4977480	172990
<b>4</b>	5	Banco de Agricultura de China	Porcelana	186140	37920	5356860	141820
<b>5</b>	6	Banco de America	Estados Unidos	133840	28620	3194660	220820
<b>6</b>	7	Alfabeto	Estados Unidos	282850	58590	369490	1340530
<b>7</b>	8	ExxonMobil	Estados Unidos	393160	61690	369370	439390
<b>8</b>	9	microsoft	Estados Unidos	207590	69020	380090	2309840
<b>9</b>	10	Manzana	Estados Unidos	385100	94320	332160	2746210

```
En [ ]: # Se Modifica el formato de Los datos (Coerción de datos, quitar comas)

forbes23 = forbes23 . reemplazar ( ',' , '' , regex = True ) # regex: expresión
forbes23
```

Afuera[...

	Rango	Compañía	País	Ventas	Beneficios	Activos	Valor de mercado
<b>0</b>	1	JPMorgan Chase	Estados Unidos	179930	41800	3744300	399590
<b>1</b>	2	Compañía petrolera de Arabia Saudita (Saudi Aramco)	Arabia Saudita	589470	156360	660990	2055220
<b>2</b>	3	ICBC	Porcelana	216770	52470	6116820	203010
<b>3</b>	4	Banco de construccion de China	Porcelana	203080	48250	4977480	172990
<b>4</b>	5	Banco de Agricultura de China	Porcelana	186140	37920	5356860	141820
...	...	...	...	...	...	...	...
<b>1995</b>	1996	Alfa Laval	Suecia	5350	4895	7820	15600
<b>1996</b>	1996	Brecha	Estados Unidos	15620	-202	11390	3170
<b>1997</b>	1996	Si Banco	India	3340	916	43220	5600
<b>1998</b>	1999	BEKB-BCBE	Suiza	556	1671	42970	2490
<b>1999</b>	1999	Ipsen	Francia	3180	6817	5990	10210

2000 filas × 7 columnas

En [ ]: *#preguntar si hay datos vacios o pedir suma, resultados sospechosos*  
forbes23 . isna () . suma ()

Afuera[...

Rango 0  
Empresa 0  
País 0  
Ventas 0  
Beneficios 0  
Activos 0  
Valor\_de\_mercado 0  
tipo de letra: int64

En [ ]: *#sumar*  
forbes23 [ 'Ventas' ] . suma ()

Afuera[...

50844922

Normalización de los datos.

Como ocurre con cualquier algoritmo de Machine Learning que utilice funciones de distancia, los datos deben ser normalizados antes de aplicarles el algoritmo. En este caso se utilizará la función `MinMaxScaler()` que normaliza todos los datos entre `[0, 1]`.

```
En [ ]: # Se eliminan filas que tengan valor NaN.
df = forbes23 . dropna ()
df = df . reset_index ( soltar = Verdadero )
```

```
En [ ]: # se pregunta si alguno de los elementos de una lista esta vacio anidando con la fu
forbes23 . isna () . cualquiera () . cualquiera ()
```

Afuera[... FALSO

```
En [ ]: # filtrar datos mediante índices con la función iloc y cambiar el tipo de dato a fl
Forbes23 . iloc[:, 3:] = forbes23 . iloc[:, 3:] . como tipo ( flotante )
```

```
En [ ]: # Se pregunta si hay datos vacios.
Forbes23 . isna () . suma ()
```

Afuera[... Rango 0  
Empresa 0  
País 0  
Ventas 0  
Beneficios 0  
Activos 0  
Valor\_de\_mercado 0  
tipo de letra: int64

```
En [ ]: forbes23 [ 'Beneficios' ] . suma ()
```

Afuera[... 7666268.0

```
En [ ]: Forbes23 . tipos
```

Afuera[... Rango int64  
Objeto social  
Objeto de país  
Flotación de ventas64  
Objeto de ganancias  
Los activos flotan64  
Valor de mercado flotante64  
tipo de objeto: objeto

```
En [ ]: #Coercionar datos

Forbes23 . iloc[:, 3:] . como tipo ( flotante )
```

Afuera[...

	Ventas	Beneficios	Activos	Valor de mercado
<b>0</b>	179930.0	41800.0	3744300.0	399590.0
<b>1</b>	589470.0	156360.0	660990.0	2055220.0
<b>2</b>	216770.0	52470.0	6116820.0	203010.0
<b>3</b>	203080.0	48250.0	4977480.0	172990.0
<b>4</b>	186140.0	37920.0	5356860.0	141820.0
...	...	...	...	...
<b>1995</b>	5350.0	4895.0	7820.0	15600.0
<b>1996</b>	15620.0	-202.0	11390.0	3170.0
<b>1997</b>	3340.0	916.0	43220.0	5600.0
<b>1998</b>	556.0	1671.0	42970.0	2490.0
<b>1999</b>	3180.0	6817.0	5990.0	10210.0

2000 filas × 4 columnas

```
En [ ]: # Coercionar los datos de str a float
forbes23 . iloc[:, 3:] = forbes23 . iloc[:, 3:] . como tipo ( flotante )
```

```
En [ ]: # Preguntar si hay datos vacios

Forbes23 . isna () . cualquiera () # Método de Pandas
```

Afuera[... Rango Falso  
 Empresa Falso  
 país falso  
 Ventas Falso  
 Beneficios falsos  
 Activos Falso  
 Valor\_de\_mercado Falso  
 tipo de letra: bool

```
En [ ]: # se pregunta si alguno de los elementos de una lista esta vacio anidando con la fu
forbes23 . isna () . cualquiera () . cualquiera ()
```

Afuera[... FALSO

```
En [ ]: # Se Modifica el formato de los datos (Coerción de datos, quitar la coma)

forbes23 = forbes23 . reemplazar ( ',' , '' , regex = True ) # regex: expresión
forbes23
```

Afuera[...

	Rango	Compañía	País	Ventas	Beneficios	Activos	Valor de mercado
<b>0</b>	1	JPMorgan Chase	Estados Unidos	179930.0	41800.0	3744300.0	399590.0
<b>1</b>	2	Compañía petrolera de Arabia Saudita (Saudi Aramco)	Arabia Saudita	589470.0	156360.0	660990.0	2055220.0
<b>2</b>	3	ICBC	Porcelana	216770.0	52470.0	6116820.0	203010.0
<b>3</b>	4	Banco de construccion de China	Porcelana	203080.0	48250.0	4977480.0	172990.0
<b>4</b>	5	Banco de Agricultura de China	Porcelana	186140.0	37920.0	5356860.0	141820.0
...	...	...	...	...	...	...	...
<b>1995</b>	1996	Alfa Laval	Suecia	5350.0	4895.0	7820.0	15600.0
<b>1996</b>	1996	Brecha	Estados Unidos	15620.0	-202.0	11390.0	3170.0
<b>1997</b>	1996	Si Banco	India	3340.0	916.0	43220.0	5600.0
<b>1998</b>	1999	BEKB-BCBE	Suiza	556.0	1671.0	42970.0	2490.0
<b>1999</b>	1999	Ipsen	Francia	3180.0	6817.0	5990.0	10210.0

2000 filas × 7 columnas

En [ ]: Forbes23 . es nulo ( ) . suma ( )

Afuera[...

Rango 0  
 Empresa 0  
 País 0  
 Ventas 0  
 Beneficios 0  
 Activos 0  
 Valor\_de\_mercado 0  
 tipo de letra: int64

En [ ]: *### Proceso de Limpieza de Datos: Cambiar el nombre de las variables*  
 forbes23 . columnas = [ 'Rango' , 'Empresa' , 'País' , 'Ventas' , 'Beneficios' ,  
 forbes23 . columnas



```
Afuera[...] Index(['Rango', 'Empresa', 'País', 'Ventas', 'Beneficios', 'Activos',
                    'Valor de mercado'],
                    tipo d='objeto')
```

```
En [ ]: forbes23
```

```
Afuera[...]
```

	Rango	Compañía	País	Ventas	Beneficios	Activos	Valor de mercado
<b>0</b>	1	JPMorgan Chase	Estados Unidos	179930.0	41800.0	3744300.0	399590.0
<b>1</b>	2	Compañía petrolera de Arabia Saudita (Saudi Aramco)	Arabia Saudita	589470.0	156360.0	660990.0	2055220.0
<b>2</b>	3	ICBC	Porcelana	216770.0	52470.0	6116820.0	203010.0
<b>3</b>	4	Banco de construccion de China	Porcelana	203080.0	48250.0	4977480.0	172990.0
<b>4</b>	5	Banco de Agricultura de China	Porcelana	186140.0	37920.0	5356860.0	141820.0
...	...	...	...	...	...	...	...
<b>1995</b>	1996	Alfa Laval	Suecia	5350.0	4895.0	7820.0	15600.0
<b>1996</b>	1996	Brecha	Estados Unidos	15620.0	-202.0	11390.0	3170.0
<b>1997</b>	1996	Si Banco	India	3340.0	916.0	43220.0	5600.0
<b>1998</b>	1999	BEKB-BCBE	Suiza	556.0	1671.0	42970.0	2490.0
<b>1999</b>	1999	Ipsen	Francia	3180.0	6817.0	5990.0	10210.0

2000 filas × 7 columnas

```
En [ ]: # La función describe de Pandas regresa estadísticas descriptivas incluyendo: media
Forbes23 . columns = [ 'Rango' , 'Empresa' , 'País' , 'Ventas' , 'Beneficios' ,
forbes23 . columns
forbes23 . describir ()
```

Afuera[...

	Rango	Ventas	Beneficios	Activos	Valor de mercado
<b>contar</b>	2000.000000	2000.000000	2000.000000	2.000000e+03	2.000000e+03
<b>significar</b>	1000.309500	25422.461000	3833.134000	1.158463e+05	3.699562e+04
<b>enfermedad de transmisión sexual</b>	577.395083	46517.172959	6919.101006	3.722422e+05	1.142306e+05
<b>mín.</b>	1.000000	-109990.000000	-76170.000000	1.050000e+02	3.200000e+01
<b>25%</b>	500.750000	5770.000000	1220.000000	1.472000e+04	6.860000e+03
<b>50%</b>	1000.500000	12500.000000	2575.500000	3.165500e+04	1.531000e+04
<b>75%</b>	1499.500000	25170.000000	5872.750000	7.697250e+04	3.404250e+04
<b>máximo</b>	1999.000000	611290.000000	156360.000000	6.116820e+06	2.746210e+06

```
En [ ]: print ( forbes23 . groupby ( 'Sales' ) . size () ) #Para saber cuantos registros t
```

Ventas

-109990.0 1

-74520.0 1

-38600.0 1

-25510.0 1

-9090.0 1

..

453560.0 1

457400.0 1

524900.0 1

589470.0 1

611290.0 1

Longitud: 1568, tipo d: int64

```
En [ ]: Forbes23 . soltar ([ 'Ventas' ], eje = 1 ) . hist () #Ver gráficamente los datos
plt . mostrar ()
```



```
En [ ]: importar seaborn como sb
sb . pairplot ( forbes23 . dropna () , hue = 'Sales' , size = 4 , vars = [ "Rank" ,
# En este caso se seleccionan 3 dimensiones: Rank , Assets y Market_value y se cruz
```

```

c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\axisgrid.py:2095: UserWarning: El parámetro `size` ha sido renombrado a `height`; por favor actualice su código.
    advertencias.warn(msg, Advertencia de usuario)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: la opción use_inf_as_na está obsoleta y se eliminará en una versión futura. Convierta los valores inf a NaN antes de operar.
    con pd.option_context('mode.use_inf_as_na', Verdadero):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):

```

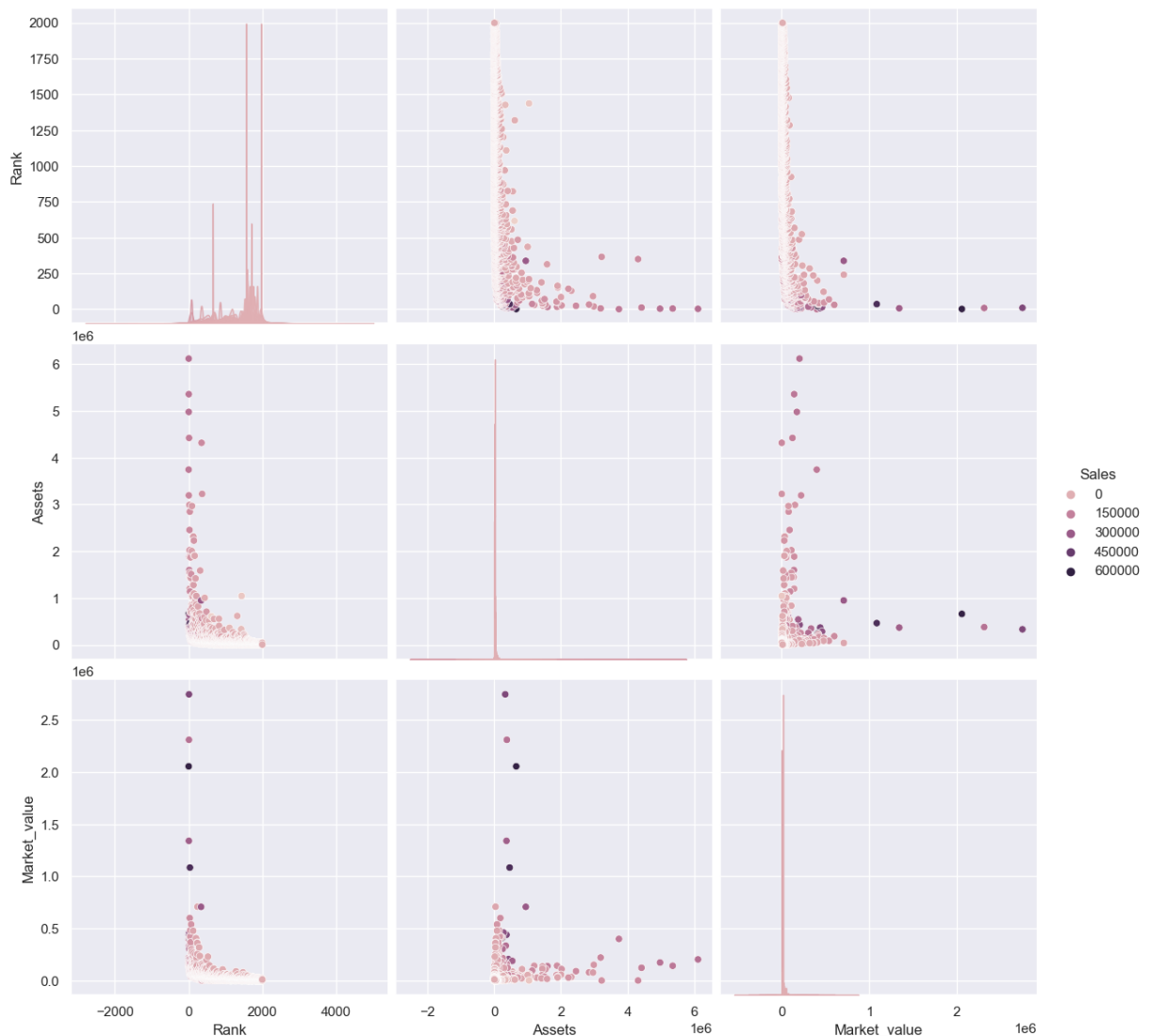
```
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: la opción use_inf_as_na está obsoleta y se eliminará en una versión futura. Convierta los valores inf a NaN antes de operar.  
    con pd.option_context('mode.use_inf_as_na', Verdadero):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: la opción use_inf_as_na está obsoleta y se eliminará en una versión futura. Convierta los valores inf a NaN antes de operar.  
    con pd.option_context('mode.use_inf_as_na', Verdadero):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):  
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar  
    si pd.api.types.is_categorical_dtype(vector):
```

```

c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\seaborn\_
oldcore.py:1498: FutureWarning: is_categorical_dtype está obsoleto y se eliminará en
una versión futura. Utilice isinstance(dtype, CategoricalDtype) en su lugar
    si pd.api.types.is_categorical_dtype(vector):

```

Afuera[... <seaborn.axisgrid.PairGrid en 0x27c4399ab50>



Revisando la gráfica no pareciera que haya algún tipo de agrupación o compensación entre las ventas (ventas) y sus variables.

```
En [ ]: # Definir la entrada
# Se Concreta la estructura de datos que se utilizará para alimentar el algoritmo.

X = np . array ( forbes23 [ [ "Rango" , "Activos" , "Valor_de_mercado" ] ])
y = np . matriz ( forbes23 [ 'Ventas' ] )
X . forma
```

Afuera[... (2000, 3)

Representación gráfica de los datos.

Ahora se va a echar un vistazo a la representación gráfica de los datos. En el eje x se representa las ventas (Sales) y en el eje y Assets (activos):

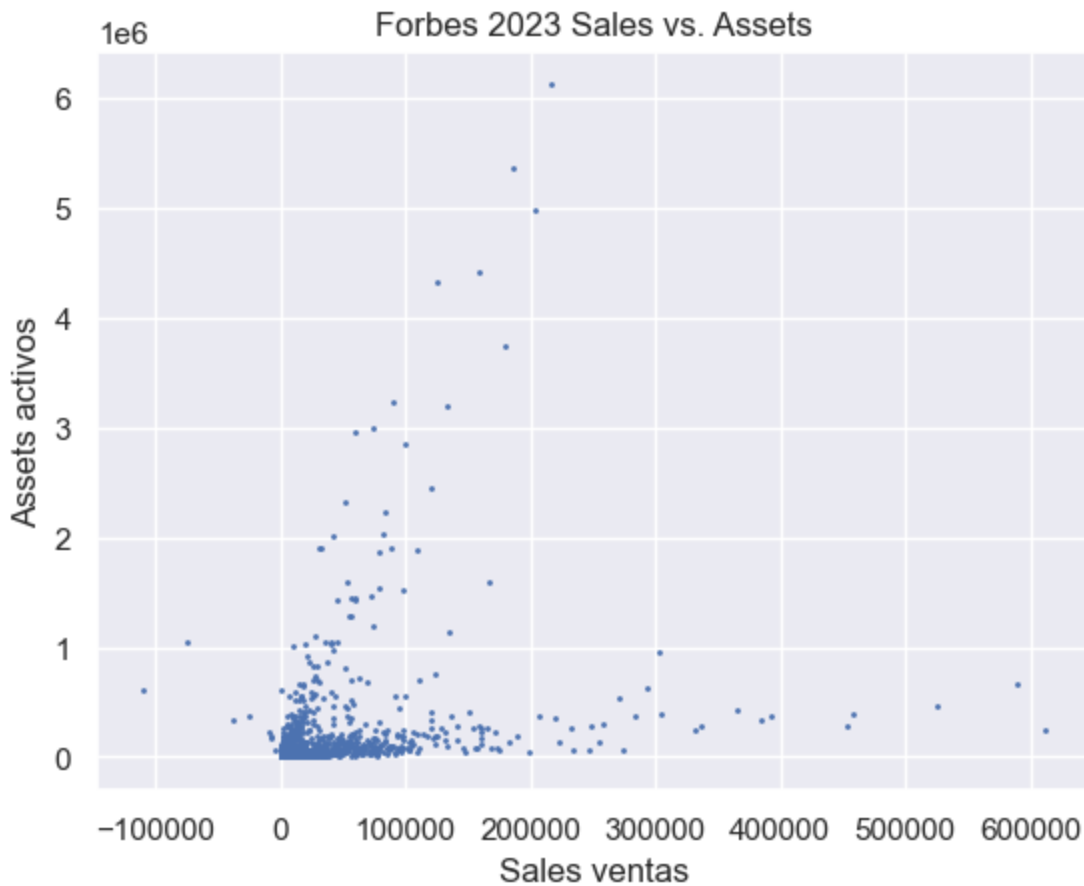
```
En [ ]: #%% Representación gráfica de los datos.
df_escalado = forbes23
x = df_escalado [ 'Ventas' ] . valores
```

```

y = df_escalado [ 'Activos' ] . valores
plt . xlabel ( 'Ventas ventas' )
plt . ylabel ( 'Activos activos' )
plt . título ( 'Ventas frente a activos de Forbes 2023' )
plt . trazar ( x , y , 'o' , tamaño del marcador = 1 )

```

Afuera[... [`<matplotlib.lines.Line2D en 0x27c2dfeeed0>`]



Obtener el valor K Se hallara el valor de K haciendo una gráfica e intentando encontrar el "punto de codo" que se comento antes. Este es el resultado:

```

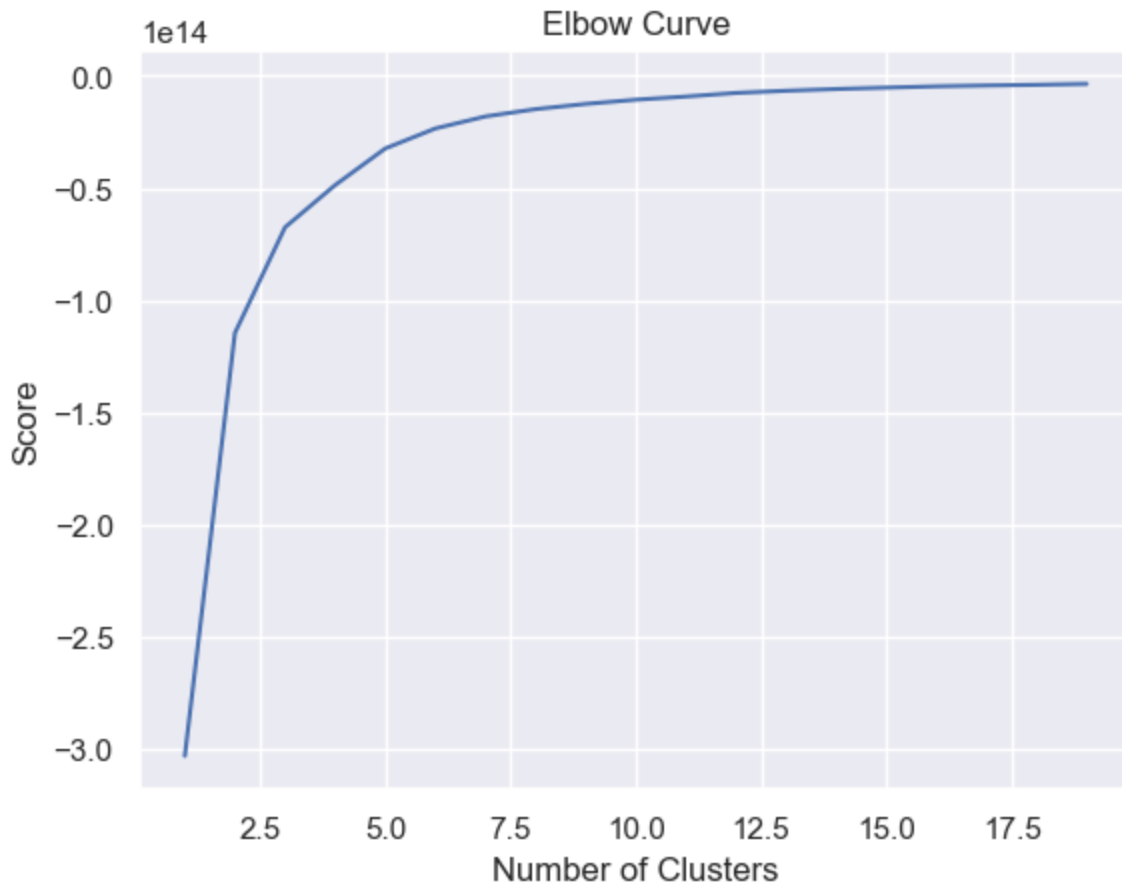
En [ ]: Nc = rango ( 1 , 20 )
kmeans = [ KMeans ( n_clusters = i ) para i en Nc ]
kmeans
puntuación = [ kmeans [ i ] . encajar ( X ) . puntuación ( X ) para i en el r
puntuación
plt . trama ( NC , puntuación )
plt . xlabel ( 'Número de clústeres' )
plt . ylabel ( 'Puntuación' )
plt . título ( 'Curva del codo' )
plt . mostrar ()

```



```
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
```

```
luster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super()._check_params_vs_input(X, default_n_init=10)
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
    super().check_params_vs_input(X, default_n_init=10)
```



En realidad la curva es bastante "suave". Considere a 5 como un buen número para K.

Se ejecuta K-Means. Se ejecuta el algoritmo para 5 clusters y se obtienen las etiquetas y los centroides.

```
En [ ]: kmeans = KMeans ( n_clusters = 5 ) . ajustar ( X )
        centroides = kmeans . cluster_centers_
        print ( centroides )
```

```
c:\Users\USUARIO\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\cluster\_kmeans.py:1416: FutureWarning: El valor predeterminado de `n_init` cambiará de 10 a 'auto' en 1.4. Establezca el valor de `n_init` explícitamente para suprimir la advertencia
```

```
super()._check_params_vs_input(X, default_n_init=10)
[[1.05685376e+03 4.92861640e+04 2.95193817e+04]
 [6.25000000e+01 4.82244667e+06 1.73431833e+05]
 [9.21538462e+01 1.98921154e+06 7.61021923e+04]
 [3.08384615e+02 5.54072308e+05 7.32128942e+04]
 [7.00000000e+00 4.35682500e+05 2.11295000e+06]]
```

Representación gráfica de los clústeres k-means

Una vez con los datos etiquetados, podemos visualizar gráficamente en dos dimensiones el clustering realizado por k-means, ya que se usaron sólo dos variables.

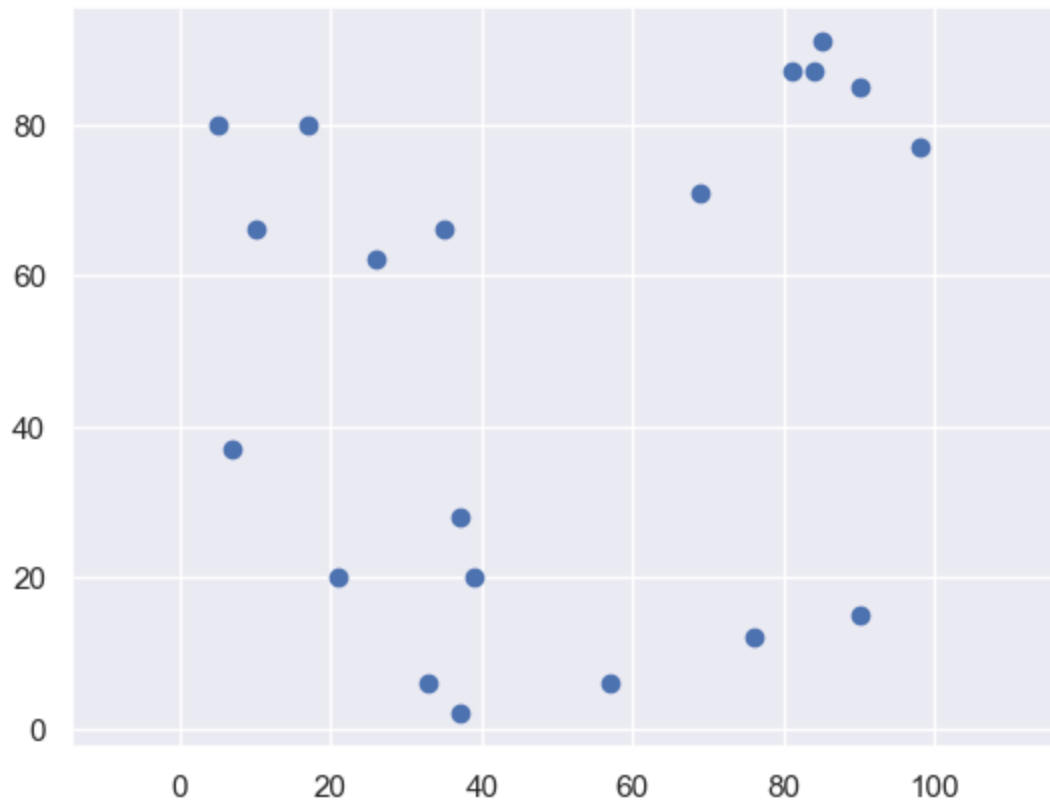
```
En [ ]: # df Conjunto de datos de entrada
forbes23 = np . aleatorio . randint ( 1 , 101 , tamaño = ( 20 , 2 ))
imprimir ( forbes23 )
```

```
[[37 28]
 [26 62]
 [21 20]
 [17 80]
 [76 12]
 [81 87]
 [69 71]
 [10 66]
 [57 6]
 [84 87]
 [33 6]
 [37 2]
 [5 80]
 [90 15]
 [85 91]
 [35 66]
 [90 85]
 [98 77]
 [39 20]
 [7 37]]
```

```
En [ ]: # Realicemos un scatterplo
sns . conjunto ()

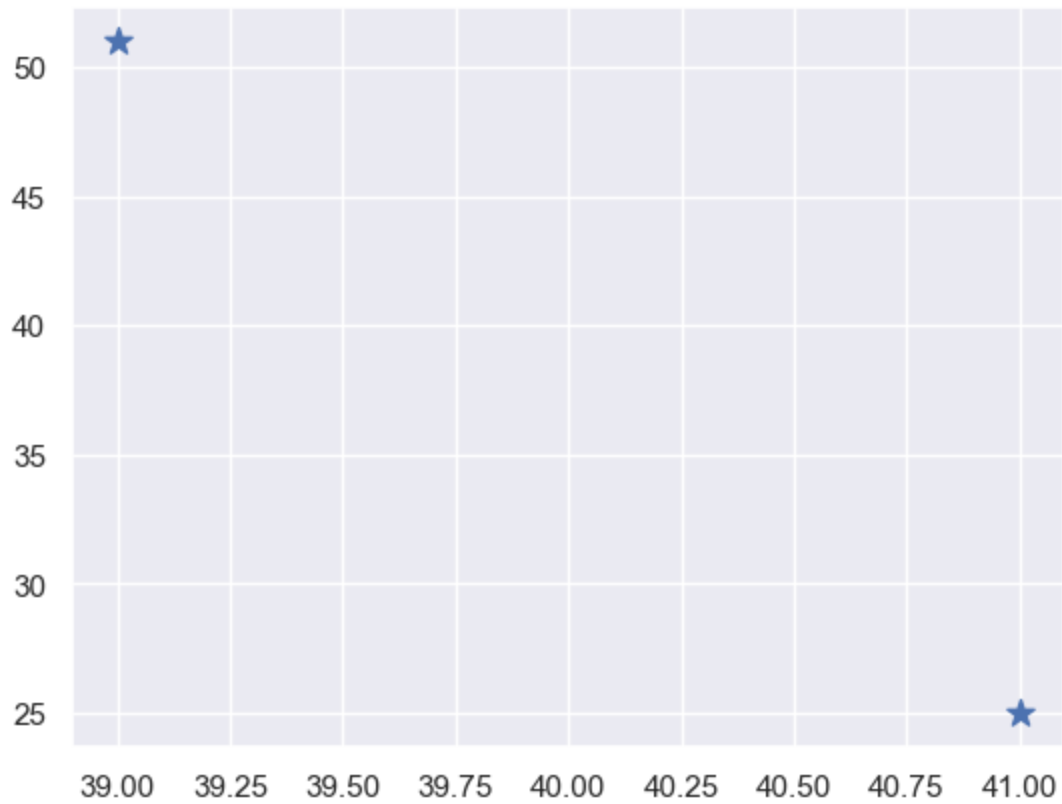
pl . axis ( 'equal' ) #Misma escala de los ejes
plt . dispersión ( forbes23[:, 0 ], forbes23[:, 1 ])
```

Afuera[... <matplotlib.collections.PathCollection en 0x27c4207d190>



```
En [ ]: # Iniciamos los centroides
ck = np . aleatorio . randint ( np . min ( forbes23 ), np . max ( forbes23 ), ta

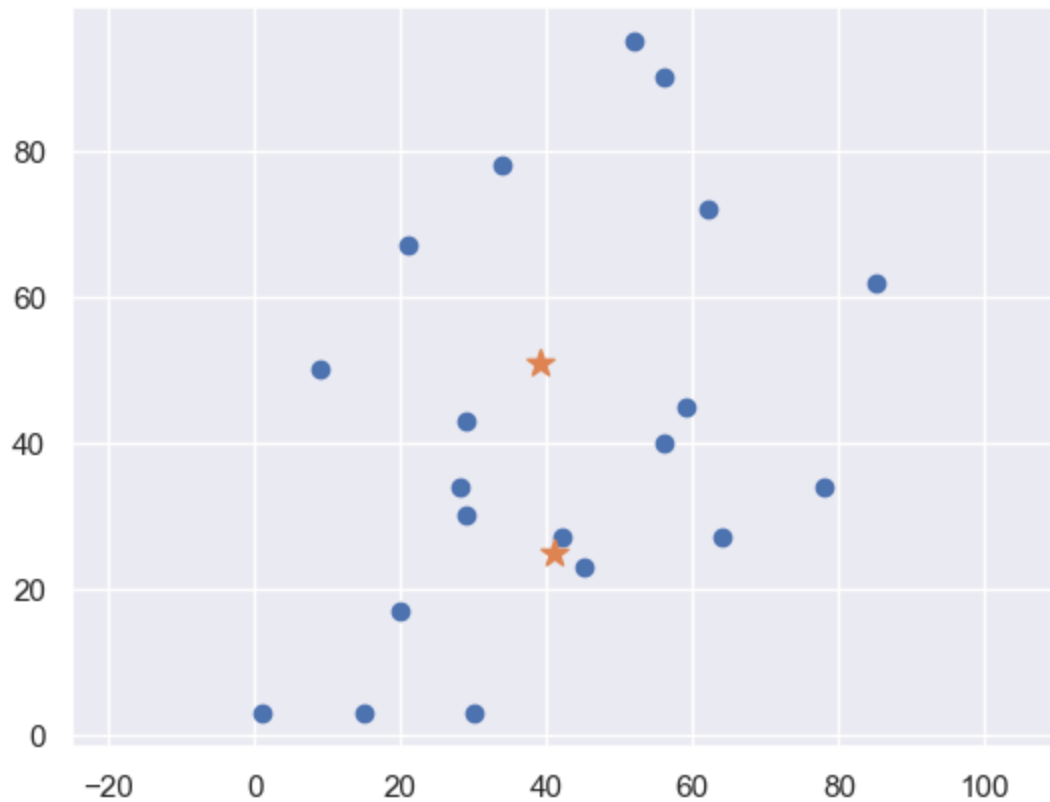
pl . dispersión ( ck[:, 0 ], ck[:, 1 ], s = 100 , marcador = '*' )
plt . mostrar ()
imprimir ( ck [ 0 ,:], ck [ 1 ,:])
```



[41 25] [39 51]

```
En [ ]: pl . axis ( 'equal' ) #Misma escala de los ejes
        plt . dispersión ( forbes23 [:, 0 ], forbes23 [:, 1 ])

        pl . dispersión ( ck [:, 0 ], ck [:, 1 ], s = 100 , marcador = '*' )
        plt . mostrar ()
        imprimir ( ck [ 0 ,:], ck [ 1 ,:])
```



[41 25] [39 51]

En [ ]: *# Calcular las distancias de cada dato al centroide y reasignar a los conjuntos C1*

```

para iteración en el rango ( 5 ):
    C1 = []
    C2 = []
    i = 0
    print ( ck [ 0 ,:], ck [ 1 ,:])
    para x en forbes23 :
        dst1 = distancia . euclidiana ( forbes23 [ i ,:], ck [ 0 ,:])
        dst2 = distancia . euclidiano ( forbes23 [ i ,:], ck [ 1 ,:])
        si dst1 < dst2 :
            C1 . agregar ( forbes23 [ i ,:])
        más :
            C2 . agregar ( forbes23 [ i ,:])
        i = i + 1

    imprimir ( 'iteración:' , iteración )

AC1 = np . matriz ( C1 )
AC2 = np . matriz ( C2 )

pl . eje ( 'igual' )
plt . dispersión ( ck[:, 0 ], ck[:, 1 ], s = 100 , marcador = '*' )
plt . dispersión ( AC1[:, 0 ], AC1[:, 1 ], etiqueta = 'Posición verdadera' )
plt . dispersión ( AC2[:, 0 ], AC2[:, 1 ], etiqueta = 'Posición verdadera' )
plt . mostrar ()

# Se actualizan las coordenadas de los centroides
ck [ 0 , 0 ] = np . media ( AC1[:, 0 ])

```

```
ck [ 0 , 1 ] = np . media ( AC1 [ :, 1 ] )

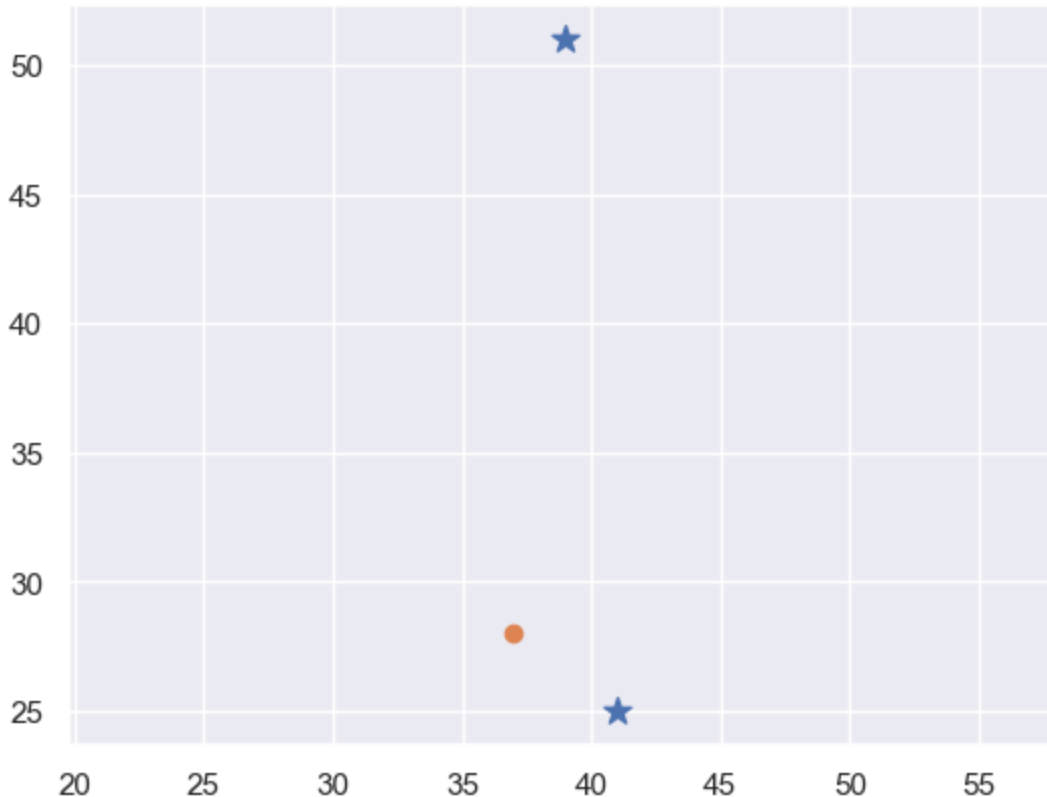
ck [ 1 , 0 ] = notario público . media ( AC2 [ :, 0 ] )
ck [ 1 , 1 ] = np . media ( AC2 [ :, 1 ] )
```

[41 25] [39 51]

iteración: 4

```
-----
IndexError                                Traceback (última llamada más reciente)
  c:\Users\USUARIO\OneDrive\Documentos\Uibero Ing de Software\8 SEMESTRE\Inteligencia
  _artificial-main\cluster_k-means.ipynb Celda 48 línea 2
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
  0Ing%20de%20Software/8%20SEMESTRE /Inteligencia_artificial-main/cluster_k-means.ipyn
  b#X65sZmlsZQ%3D%3D?line=22'>23</a> plt.scatter(ck[:,0],ck[:,1],s=100,marcador = '*')
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
  0Ing%20de%20Software/8%20SEMESTRE/Inteligencia_artificial-main/cluster_k-means.ipynb
  #X65sZmlsZQ%3D%3D?line=23'>24</a>plt.scatter(AC1[:,0],AC1[:,1], label='Posición verd
  adera') --->
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive /Documentos/Uibero%20In
  g%20de%20Software/8%20SEMESTRE/Inteligencia_artificial-main/cluster_k-means.ipynb#X6
  5sZmlsZQ%3D%3D?line=24'>25</a> plt.scatter(AC2[:,0 ],AC2[:,1], label='Posición Verda
  dera')
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
  0Ing%20de%20Software/8% 20SEMESTRE/Inteligencia_artificial-main/cluster_k-means.ipyn
  b#X65sZmlsZQ%3D%3D?line=25'>26</a> plt.show()
    <a href='vscode-notebook-cell:/c%3A/Users /USUARIO/OneDrive/Documentos/Uibero%
  20Ing%20de%20Software/8%20SEMESTRE/Inteligencia_artificial-main/cluster_k-means.ipyn
  b#X65sZmlsZQ%3D%3D?line=27'>28</a># Se actualizan las coordenadas de los centroides

IndexError : demasiados índices para la matriz: la matriz es unidimensional, pero se
  indexaron 2
```





```

En [ ]: # Predicción del Modelo
# Dado un nuevo dato xn, se asigna al conjunto C1 o C2 teniendo en cuenta la distan
# del elemento a los centroides de cada grupo

# xn, nuevo dato
xn = ( 20 , 30 )

# Distancia de xn a cada centroide
dst1 = distancia . euclidiana ( xn , ck [ 0 ,:])
dst2 = distancia . euclidiana ( xn , ck [ 1 ,:])

if dst1 < dst2 :
    print ( 'El dato Xn pertenece al cluster: {} con centroide {} ' . format ( 1 ,
else :
    print ( 'El dato Xn pertenece al cluster: {} con centroide {} ' .formato ( 2 ,

pl . eje ( 'igual' )
plt . title ( 'Predicción del modelo k-means' )
plt . dispersión ( xn [ 0 ], xn [ 1 ], s = 100 , marcador = 'o' )
plt . dispersión ( ck[:, 0 ], ck[:, 1 ], s = 100 , marcador = '*' )
plt .dispersión ( AC1[:, 0 ], AC1[:, 1 ], etiqueta = 'Posición verdadera' )
plt . dispersión ( AC2[:, 0 ], AC2[:, 1 ], etiqueta = 'Posición verdadera' )
plt . mostrar ()

```

El dato Xn pertenece al cluster:2 con centroide [70 57]

```

-----
IndexError                                Traceback (última llamada más reciente)
c:\Users\USUARIO\OneDrive\Documentos\Uibero Ing de Software\8 SEMESTRE\Inteligencia
_artificial-main\cluater_k-means.ipynb Celda 51 línea 2
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
0Ing%20de%20Software/8%20SEMESTRE /Inteligencia_artificial-main/cluater_k-means.ipyn
b#Y101sZmlsZQ%3D%3D?line=18'>19</a> plt.scatter(xn[0],xn[1],s=100,marker='o' )
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
0Ing%20de%20Software/8%20SEMESTRE/Inteligencia_artificial-main/cluater_k-means.ipynb
#Y101sZmlsZQ%3D%3D?line=19'>20</a>plt.scatter(ck[:,0],ck[:,1],s=100,marker='*') --->
<a href
    ='vscode-notebook-cell:/c%3A/Users/ USUARIO/OneDrive/Documentos/Uibero%20Ing%20de%2
0Software/8%20SEMESTRE/Inteligencia_artificial-main/cluater_k-means.ipynb#Y101sZmlsZ
Q%3D%3D?line=20'>21</a> plt.scatter(AC1 [ :,0],AC1[:,1], label='Posición Verdadera')
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
0Ing%20de%20Software /8%20SEMESTRE/Inteligencia_artificial-main/cluater_k-means.ipyn
b#Y101sZmlsZQ%3D%3D?line=21'>22</a> plt.scatter(AC2[:,0],AC2[:,1], etiqueta = 'Posici
ón verdadera')
    <a href='vscode-notebook-cell:/c%3A/Users/USUARIO/OneDrive/Documentos/Uibero%2
0Ing%20de%20Software/8%20SEMESTRE/Inteligencia_artificial-main/cluater_k-means.ipynb
#Y101sZmlsZQ%3D%3D?line=22'>23</a>plt.mostrar()

IndexError : demasiados índices para la matriz: la matriz es unidimensional, pero se
indexaron 2

```

