# Universidad Nacional Autónoma de México


## Inteligencia Artificial

# Clustering Jerárquico-Particional

### Grupo 3

*Nombre:*
Barreiro Valdez Alejandro

## Práctica 6

**Profesor: Dr. Guillermo Gilberto Molero Castillo**

31 de marzo de 2022

## Introducción

En esta práctica se utilizará clustering jerárquico y particional para analizar grupos de pacientes que fueron diagnosticadas con un tumor de mama. Para esto se hará uso de dos algoritmos de agrupación para generar grupos donde se pueda observar a través de sus características qué tumores son potencialmente malignos y cuáles benignos. Se utilizarán todas las herramientas y métodos que se desarrollaron en la práctica 4 y 5.

## Objetivos

Obtener grupos de pacientes con características similares, diagnosticadas con un tumor de mama, a través de clustering jerárquico y particional.

## Desarrollo

Primero se importan las bibliotecas necesarias para la realización de la práctica. Se utilizan las mismas bibliotecas que en prácticas pasadas.

```
import pandas as pd                 # Para la manipulación y análisis de datos
import numpy as np                  # Para crear vectores y matrices n dimensionales
import matplotlib.pyplot as plt     # Para la generación de gráficas a partir de los datos
import seaborn as sns               # Para la visualización de datos basado en matplotlib
%matplotlib inline
```

Se lee el archivo csv de los datos que representan datos sobre pacientes para encontrar cáncer de mama. Se tienen 569 casos.

```
BCancer = pd.read_csv('WDBCOriginal.csv')
BCancer
```

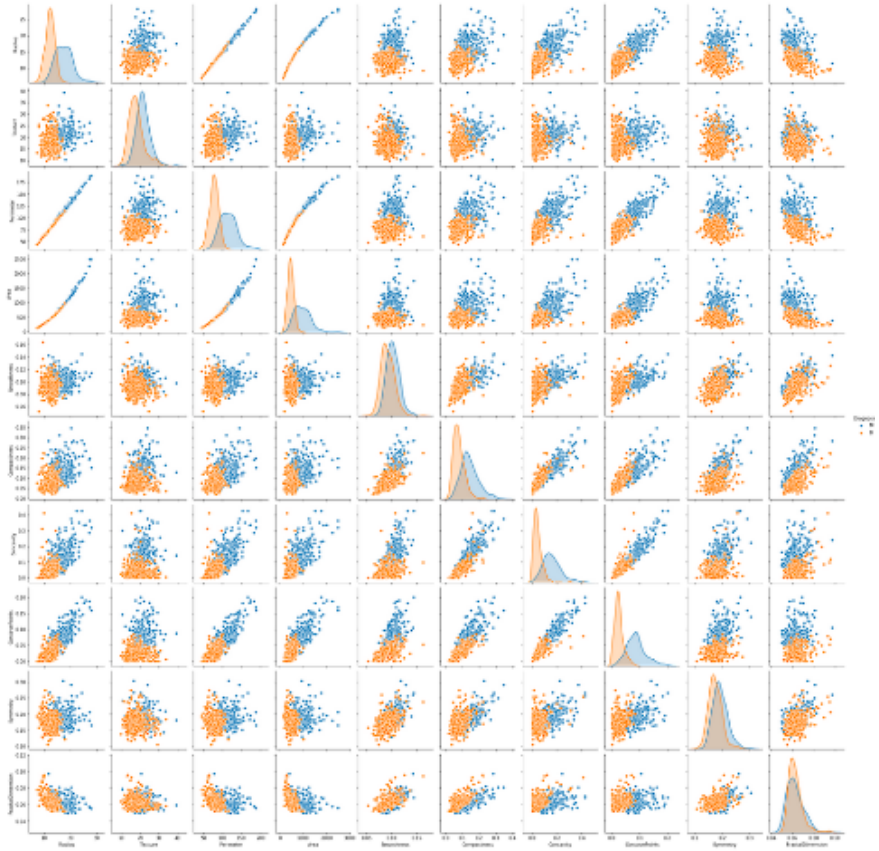
| | IDNumber | Diagnosis | Radius | Texture | Perimeter | Area | Smoothness | Compactness | Concavity | ConcavePoints | Symmetry | FractalDimension |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | P-842302 | M | 17.99 | 10.38 | 122.80 | 1001.0 | 0.11840 | 0.27760 | 0.30010 | 0.14710 | 0.2419 | 0.07871 |
| 1 | P-842517 | M | 20.57 | 17.77 | 132.90 | 1326.0 | 0.08474 | 0.07864 | 0.08690 | 0.07017 | 0.1812 | 0.05667 |
| 2 | P-84300903 | M | 19.69 | 21.25 | 130.00 | 1203.0 | 0.10960 | 0.15990 | 0.19740 | 0.12790 | 0.2069 | 0.05999 |
| 3 | P-84348301 | M | 11.42 | 20.38 | 77.58 | 386.1 | 0.14250 | 0.28390 | 0.24140 | 0.10520 | 0.2597 | 0.09744 |
| 4 | P-84358402 | M | 20.29 | 14.34 | 135.10 | 1297.0 | 0.10030 | 0.13280 | 0.19800 | 0.10430 | 0.1809 | 0.05883 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 564 | P-926424 | M | 21.56 | 22.39 | 142.00 | 1479.0 | 0.11100 | 0.11590 | 0.24390 | 0.13890 | 0.1726 | 0.05623 |
| 565 | P-926682 | M | 20.13 | 28.25 | 131.20 | 1261.0 | 0.09780 | 0.10340 | 0.14400 | 0.09791 | 0.1752 | 0.05533 |
| 566 | P-926954 | M | 16.60 | 28.08 | 108.30 | 858.1 | 0.08455 | 0.10230 | 0.09251 | 0.05302 | 0.1590 | 0.05648 |
| 567 | P-927241 | M | 20.60 | 29.33 | 140.10 | 1265.0 | 0.11780 | 0.27700 | 0.35140 | 0.15200 | 0.2397 | 0.07016 |
| 568 | P-92751 | B | 7.76 | 24.54 | 47.92 | 181.0 | 0.05263 | 0.04362 | 0.00000 | 0.00000 | 0.1587 | 0.05884 |

Se obtuvo la información de los datos para saber si existen valores nulos en el conjunto de datos. Se obtuvo el número de diagnosis malignos y benignos que existen en el conjunto de datos.

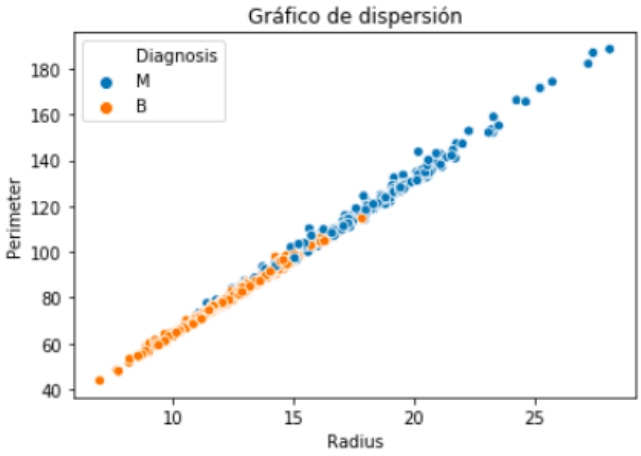
```
print(BCancer.groupby('Diagnosis').size())
```

```
Diagnosis
B    357
M    212
dtype: int64
```

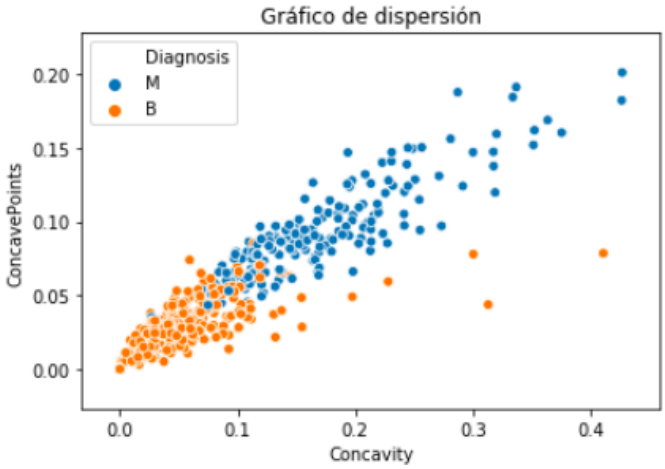
Para la selección de características primero se genera una evaluación visual de la correlación que existe entre las variables y apartir de esto se analizan dos gráficas que tienen interés para el modelo.



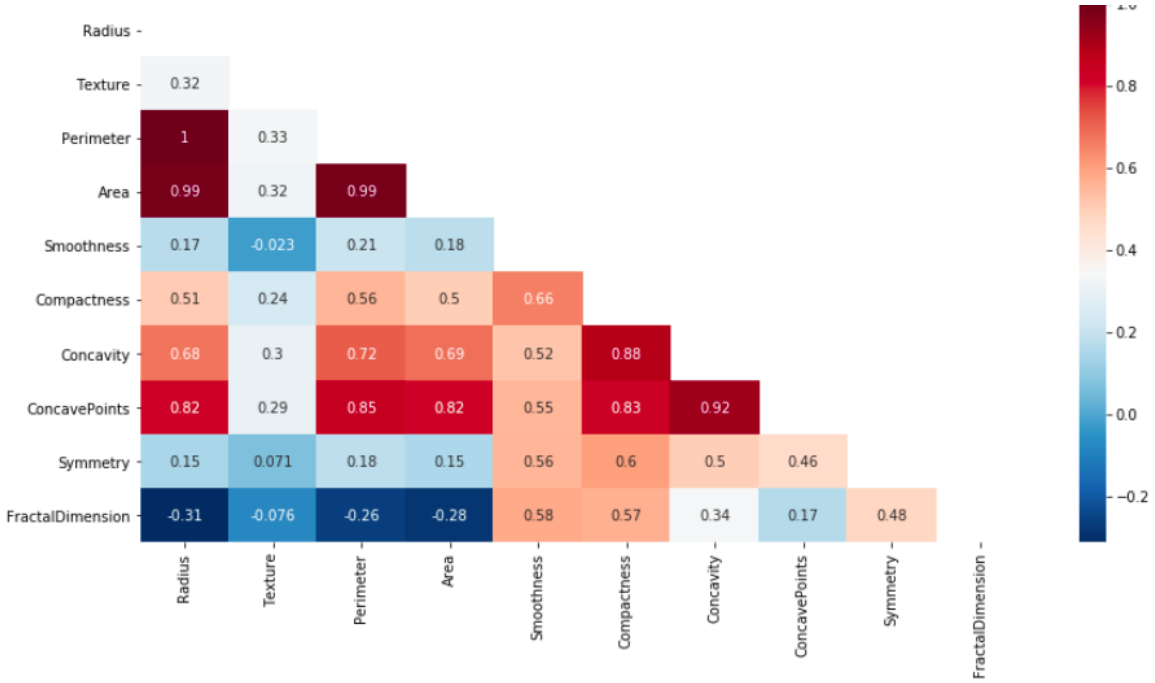
El radio y perímetro tiene una alta correlación. Esto tiene sentido porque son datos que dependen uno del otro. Se podría eliminar alguno de estos.




Otros datos que tienen alta correlación son concavidad y puntos cóncavos como se muestra en la siguiente gráfica.




Utilizando la matriz de correlaciones se analizó la relación de las variables. Con un mapa de calor se generó una selección de aquellas variables que se utilizarán para el clustering. Se seleccionan las variables que tienen el menor número de correlación con otras variables. Las gráficas anteriores muestran ejemplos de variables con mucha correlación pero existen variables adicionales que también está relacionado así. Las características seleccionadas son: textura, área, smoothness, compactness, symmetry y FractalDimension.




La selección de variables se realizó mediante un arreglo y un DataFrame. Se seleccionaron las columnas a utilizar. Aquellas variables que están altamente correlacionadas con otras se quitaron del modelo. Se muestra el nuevo conjunto de datos.

```
MatrizVariables = np.array(BCancer[['Texture', 'Area', 'Smoothness', 'Compactness', 'Symmetry', 'FractalDimension']]
pd.DataFrame(MatrizVariables)
#MatrizVariables = BCancer.iloc[:, [3, 5, 6, 7, 10, 11]].values  #iloc para seleccionar filas y columnas
```

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| **0** | 10.38 | 1001.0 | 0.11840 | 0.27760 | 0.2419 | 0.07871 |
| **1** | 17.77 | 1326.0 | 0.08474 | 0.07864 | 0.1812 | 0.05667 |
| **2** | 21.25 | 1203.0 | 0.10960 | 0.15990 | 0.2069 | 0.05999 |
| **3** | 20.38 | 386.1 | 0.14250 | 0.28390 | 0.2597 | 0.09744 |
| **4** | 14.34 | 1297.0 | 0.10030 | 0.13280 | 0.1809 | 0.05883 |
| **...** | ... | ... | ... | ... | ... | ... |
| **564** | 22.39 | 1479.0 | 0.11100 | 0.11590 | 0.1726 | 0.05623 |
| **565** | 28.25 | 1261.0 | 0.09780 | 0.10340 | 0.1752 | 0.05533 |
| **566** | 28.08 | 858.1 | 0.08455 | 0.10230 | 0.1590 | 0.05648 |
| **567** | 29.33 | 1265.0 | 0.11780 | 0.27700 | 0.2397 | 0.07016 |
| **568** | 24.54 | 181.0 | 0.05263 | 0.04362 | 0.1587 | 0.05884 |

569 rows × 6 columns

Se genera una estandarización de datos como se ha realizado en prácticas pasadas utilizando StandardScaler. Se muestra el nuevo conjunto de datos

estandarizado. Todos los pasos anteriores servirán para el clustering jerárquico y particional.

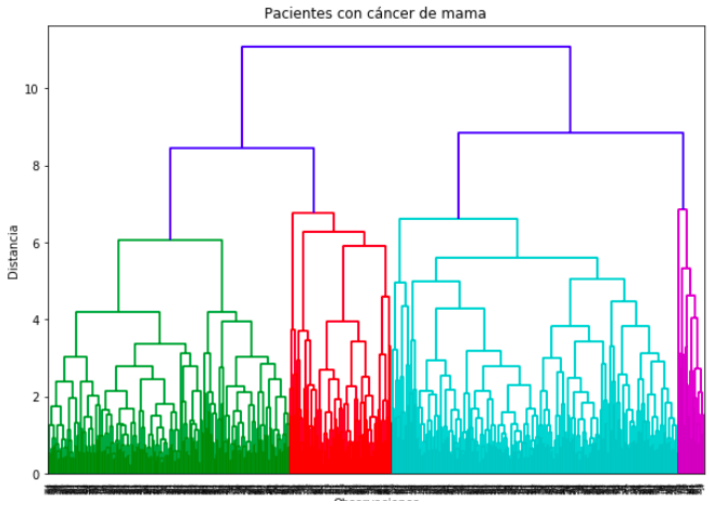
```
from sklearn.preprocessing import StandardScaler, MinMaxScaler
estandarizar = StandardScaler()
MEstandarizada = estandarizar.fit_transform(MatrizVariables)
pd.DataFrame(MEstandarizada)
```

| | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| **0** | -2.073335 | 0.984375 | 1.568466 | 3.283515 | 2.217515 | 2.255747 |
| **1** | -0.353632 | 1.908708 | -0.826962 | -0.487072 | 0.001392 | -0.868652 |
| **2** | 0.456187 | 1.558884 | 0.942210 | 1.052926 | 0.939685 | -0.398008 |
| **3** | 0.253732 | -0.764464 | 3.283553 | 3.402909 | 2.867383 | 4.910919 |
| **4** | -1.151816 | 1.826229 | 0.280372 | 0.539340 | -0.009560 | -0.562450 |
| **...** | ... | ... | ... | ... | ... | ... |
| **564** | 0.721473 | 2.343856 | 1.041842 | 0.219060 | -0.312589 | -0.931027 |
| **565** | 2.085134 | 1.723842 | 0.102458 | -0.017833 | -0.217664 | -1.058611 |
| **566** | 2.045574 | 0.577953 | -0.840484 | -0.038680 | -0.809117 | -0.895587 |
| **567** | 2.336457 | 1.735218 | 1.525767 | 3.272144 | 2.137194 | 1.043695 |
| **568** | 1.221792 | -1.347789 | -3.112085 | -1.150752 | -0.820070 | -0.561032 |

569 rows × 6 columns

Se genera un árbol jerárquico como se realizó en la práctica 4. Se utilizó una métrica de distancia euclidiana. El algoritmo de tipo jerárquico generó 4 clústeres.

```
#Se importan las bibliotecas de clustering jerárquico para crear el árbol
import scipy.cluster.hierarchy as shc
from sklearn.cluster import AgglomerativeClustering
plt.figure(figsize=(10, 7))
plt.title("Pacientes con cáncer de mama")
plt.xlabel('Observaciones')
plt.ylabel('Distancia')
Arbol = shc.dendrogram(shc.linkage(MEstandarizada, method='complete', metric='euclidean'))
#plt.axhline(y=7, color='orange', linestyle='--')
#Probar con otras medciones de distancia (euclidean, chebyshev, cityblock)
```




Posteriormente se creó el etiquetado asignado a cada uno de los pacientes un número de clúster. Se muestran las etiquetas generadas. Estas etiquetas se ponen en los datos originales y se muestra cuántas personas existen en cada

agrupación. Para la primera agrupación se tienen 23 personas, para el segundo clúster son 88 personas, para el tercer clúster se tienen 248 personas y para el último clúster se tienen 210.

```
#Se crean las etiquetas de los elementos en los clusters
MJerarquico = AgglomerativeClustering(n_clusters=4, linkage='complete', affinity='euclidean')
MJerarquico.fit_predict(MEstandarizada)
MJerarquico.labels_
```

```
array([0, 1, 1, 0, 1, 2, 1, 2, 0, 0, 3, 2, 1, 3, 0, 2, 3, 2, 1, 2, 2, 2,
       0, 1, 2, 0, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2, 3, 3, 2, 3, 2, 1, 2,
       2, 2, 3, 2, 2, 3, 3, 3, 3, 2, 3, 3, 1, 2, 3, 2, 2, 2, 2, 2, 2, 2,
       2, 3, 2, 2, 1, 2, 2, 2, 2, 1, 2, 2, 0, 2, 2, 2, 1, 1, 2, 2, 2, 2,
       3, 2, 3, 3, 3, 2, 2, 2, 2, 2, 3, 2, 3, 2, 3, 2, 2, 2, 2, 3, 0, 3,
       2, 2, 2, 2, 2, 3, 2, 2, 2, 1, 3, 2, 0, 2, 3, 3, 3, 1, 2, 1, 2, 2,
       2, 2, 1, 3, 3, 2, 2, 2, 3, 2, 2, 2, 3, 2, 0, 3, 2, 3, 2, 2, 2, 2,
       2, 3, 2, 1, 3, 3, 2, 1, 2, 3, 1, 3, 3, 1, 1, 2, 2, 3, 2, 2, 3, 3,
       2, 2, 3, 3, 1, 0, 1, 3, 3, 3, 1, 2, 2, 3, 0, 3, 3, 2, 2, 3, 2, 1,
       1, 2, 2, 1, 1, 0, 3, 3, 2, 1, 2, 3, 1, 3, 1, 1, 2, 2, 3, 3, 2, 1,
       3, 2, 2, 2, 3, 2, 2, 2, 3, 2, 2, 3, 3, 1, 3, 3, 1, 1, 3, 1, 2, 3,
       2, 3, 2, 2, 3, 2, 2, 2, 1, 3, 1, 1, 1, 2, 1, 0, 0, 1, 1, 3, 2, 3,
       2, 1, 3, 3, 2, 2, 3, 2, 1, 3, 3, 2, 3, 1, 3, 2, 1, 3, 1, 2, 3, 3,
       3, 3, 2, 3, 2, 2, 2, 3, 2, 3, 3, 2, 3, 3, 1, 3, 1, 2, 3, 3, 3, 3,
       3, 3, 3, 3, 3, 3, 2, 3, 3, 1, 2, 3, 2, 1, 2, 1, 3, 2, 3, 3, 2, 2,
       2, 2, 2, 3, 3, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 3, 0,
       1, 1, 3, 3, 2, 3, 2, 2, 3, 3, 2, 1, 3, 1, 1, 2, 1, 1, 2, 3, 1, 1,
       3, 2, 2, 3, 3, 0, 2, 3, 3, 2, 3, 3, 3, 3, 2, 2, 2, 2, 2, 1, 2, 3,
       2, 3, 3, 3, 0, 3, 3, 2, 3, 2, 2, 3, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2,
       3, 3, 3, 2, 2, 3, 2, 3, 2, 3, 3, 3, 2, 2, 1, 2, 3, 2, 3, 3, 3, 3,
       2, 3, 3, 2, 2, 2, 1, 2, 3, 1, 3, 1, 3, 2, 3, 3, 3, 3, 3, 3, 1, 1,
       3, 3, 3, 3, 3, 3, 2, 2, 2, 3, 3, 3, 2, 2, 3, 3, 2, 2, 3, 3, 2, 2,
       2, 2, 3, 1, 2, 1, 3, 3, 2, 3, 3, 3, 2, 3, 2, 1, 2, 2, 2, 1, 0, 0,
       2, 2, 2, 2, 2, 3, 2, 2, 3, 2, 1, 1, 2, 2, 2, 1, 3, 2, 2, 2, 2, 3,
       2, 2, 2, 2, 2, 2, 2, 1, 2, 0, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
       3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 1, 3, 1, 1, 1, 1, 3, 0, 3])
```

Clúster 0: Conformado por 23 pacientes con indicios de cáncer maligno por el tamaño del tumor, con un área promedio de tumor de 775 píxeles y una desviación estándar de textura de 20 pixeles. Aparentemente es un tumor compacto (0.24 pixeles), cuya suavidad alcanza 0.12 pixeles, una simetría de 0.24 y una aproximación de frontera, dimensión fractal, promedio de 0.077 pixeles.

Clúster 1: Conformado por 88 pacientes con indicios de cáncer maligno por el tamaño del tumor, con un área promedio de tumor de 1243 píxeles y una desviación estándar de textura de 22.5 píxeles. Aparentemente es un tumor compacto (0.13 pixeles), cuya suavidad alcanza 0.09 pixeles, una simetría de 0.18 y una aproximación de frontera, dimensión fractal, promedio de 0.05 píxeles.

Clúster 2: Conformado por 248 pacientes con un área promedio de tumor de 561 píxeles y una desviación estándar de textura de 18 píxeles. Aparentemente es un tumor no tan compacto (0.11 pixeles), cuya suavidad alcanza 0.10 píxeles, una simetría de 0.19 y una aproximación de frontera, dimensión fractal, promedio de 0.06 píxeles.

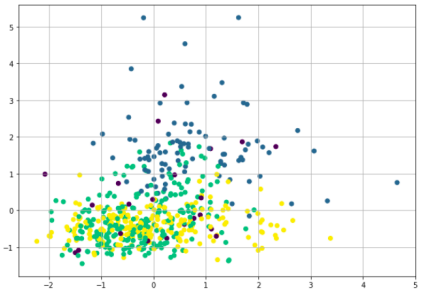
Clúster 3: Es un grupo formado por 210 pacientes con el menor tamaño de tumor (posiblemente benigno), con un área promedio de tumor de 505 píxeles y una

desviación estándar de textura de 19 pixeles. Es un tumor compacto (0.06 pixeles), cuya suavidad alcanza 0.08 píxeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 píxeles.
Estos datos se obtuvieron a partir de la tabla generada de las medias que tiene cada uno de los clústeres. Esta tabla representa las diferentes características que presenta cada uno de los grupos.

| clusterH | Texture | Area | Smoothness | Compactness | Symmetry | FractalDimension |
|---|---|---|---|---|---|---|
| **0** | 20.133478 | 775.543478 | 0.124274 | 0.242200 | 0.240830 | 0.077839 |
| **1** | 22.540568 | 1243.728409 | 0.098441 | 0.137140 | 0.182560 | 0.058889 |
| **2** | 18.167540 | 561.336694 | 0.103316 | 0.114235 | 0.190486 | 0.065737 |
| **3** | 19.160095 | 505.403810 | 0.084217 | 0.063813 | 0.163030 | 0.059317 |

Se graficó la manera en que se dividiría a los diferentes clústeres en una gráfica de dos dimensiones donde los ejes son las primeras dos variables del conjunto de datos. Parece que no existe ningún patrón porque para representar gráficamente los conjuntos se debería hacer en otra dimensión y esto no es posible.

```
plt.figure(figsize=(10, 7))
plt.scatter(MEstandarizada[:,0], MEstandarizada[:,1], c=MJerarquico.labels_
plt.grid()
plt.show()
```

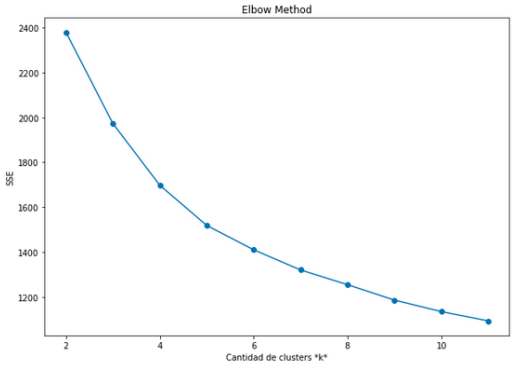


Se generó un modelo de clustering utilizando K-medias correspondiente a la parte del clustering particional. Primero se importaron las mismas bibliotecas que la práctica anterior. Se generó el SSE de diferentes modelos de K-medias con una *k* diferente. Se generó una gráfica que corresponde al método del codo para definir

qué *k* utilizar. Puede que no exista un punto claro o afilado del codo para elegir un número por lo que pueden existir ambigüedades en esta parte.

```
#Definición de k clusters para K-means
#Se utiliza random_state para inicializar el generador interno de números aleatorios
SSE = []
for i in range(2, 12):
    km = KMeans(n_clusters=i, random_state=0)
    km.fit(MEstandarizada)
    SSE.append(km.inertia_)

#Se grafica SSE en función de k
plt.figure(figsize=(10, 7))
plt.plot(range(2, 12), SSE, marker='o')
plt.xlabel('Cantidad de clusters *k*')
plt.ylabel('SSE')
plt.title('Elbow Method')
plt.show()
```

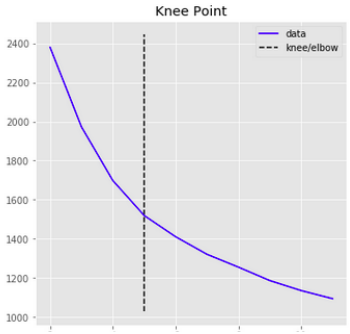



Se utiliza kneed para determinar cuál es el número de *k* que se debe utilizar para el algoritmo. En este caso, el módulo genera un *k*=5.

```
from kneed import KneeLocator
kl = KneeLocator(range(2, 12), SSE, curve="convex", direction="decreasing")
kl.elbow
```

```
5
```

```
plt.style.use('ggplot')
kl.plot_knee()
```




Se generaron las etiquetas como en la práctica pasada y se adjuntan al conjunto de datos con el que se trabaja. Estas etiquetas se adjuntan al conjunto de datos con el que se trabaja. Las etiquetas generadas fueron las siguientes.

```
#Se crean las etiquetas de los elementos en los clusters
MParticional = KMeans(n_clusters=5, random_state=0).fit(MEstandarizada)
MParticional.predict(MEstandarizada)
MParticional.labels_
```

```
array([2, 1, 1, 2, 1, 2, 1, 2, 2, 2, 4, 3, 2, 4, 2, 2, 0, 2, 1, 0, 3, 3,
       2, 1, 1, 2, 2, 1, 1, 3, 1, 2, 2, 1, 3, 1, 3, 0, 4, 3, 4, 3, 1, 3,
       4, 1, 0, 3, 3, 4, 4, 0, 0, 1, 4, 0, 1, 3, 0, 3, 3, 3, 2, 3, 3, 3,
       3, 0, 2, 0, 1, 2, 1, 3, 0, 0, 3, 2, 2, 3, 3, 3, 1, 1, 3, 1, 4, 1,
       4, 3, 4, 4, 0, 0, 3, 1, 3, 3, 0, 3, 4, 3, 0, 3, 3, 2, 3, 0, 2, 4,
       3, 3, 2, 3, 3, 4, 3, 2, 2, 1, 0, 1, 2, 3, 0, 0, 4, 1, 3, 1, 3, 3,
       1, 0, 1, 4, 0, 0, 3, 3, 0, 3, 3, 0, 0, 3, 2, 0, 3, 0, 3, 2, 2, 0,
       0, 0, 1, 0, 0, 0, 3, 1, 1, 3, 1, 0, 0, 0, 1, 0, 3, 0, 3, 0, 0, 0,
       3, 1, 4, 0, 1, 2, 4, 0, 4, 0, 0, 0, 0, 0, 2, 4, 0, 3, 3, 0, 3, 4,
       1, 3, 3, 1, 1, 2, 3, 0, 3, 1, 3, 0, 1, 0, 1, 4, 3, 3, 3, 0, 1, 4,
       0, 3, 3, 3, 0, 0, 3, 0, 4, 2, 1, 4, 4, 1, 0, 4, 1, 1, 4, 4, 0, 0,
       3, 4, 1, 3, 0, 0, 4, 3, 1, 0, 1, 1, 1, 3, 1, 2, 2, 1, 1, 4, 1, 0,
       1, 1, 3, 4, 0, 3, 0, 0, 1, 0, 4, 3, 0, 0, 0, 3, 1, 0, 1, 3, 0, 0,
       4, 0, 3, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 4, 1, 0, 2, 3, 0, 4, 0, 0,
       0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 2, 0, 3, 1, 3, 2, 0, 3, 0, 0, 3, 3,
       3, 3, 3, 0, 0, 1, 3, 1, 3, 1, 3, 3, 3, 1, 3, 3, 0, 0, 0, 3, 0, 2,
       1, 4, 0, 0, 3, 0, 0, 3, 0, 4, 3, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 3, 2, 4, 0, 2, 3, 0, 4, 3, 0, 4, 0, 0, 3, 1, 3, 3, 3, 1, 3, 0,
       3, 0, 0, 0, 2, 0, 0, 3, 0, 3, 0, 4, 1, 0, 0, 3, 4, 4, 4, 3, 3, 2,
       0, 4, 0, 2, 3, 3, 3, 4, 3, 4, 0, 0, 2, 3, 1, 1, 0, 3, 3, 0, 0, 0,
       0, 4, 0, 0, 1, 3, 1, 0, 0, 1, 4, 1, 4, 3, 0, 4, 4, 4, 4, 4, 1, 1,
       4, 0, 0, 4, 4, 0, 1, 3, 3, 4, 0, 4, 3, 0, 4, 0, 3, 2, 0, 0, 3, 0,
       3, 3, 0, 1, 3, 4, 4, 0, 1, 0, 4, 0, 3, 0, 1, 1, 3, 2, 3, 1, 2, 2,
       3, 3, 0, 2, 0, 0, 2, 0, 0, 3, 1, 1, 3, 3, 2, 1, 0, 3, 0, 3, 3, 0,
       3, 3, 3, 3, 0, 1, 3, 1, 3, 2, 4, 3, 3, 4, 4, 4, 3, 4, 0, 0, 0, 4,
       4, 3, 4, 4, 4, 4, 3, 4, 4, 4, 4, 4, 2, 2, 1, 1, 4, 2, 4],
      dtype=int32)
```

Se generó la tabla de medias y se contó cuántos elementos existe en cada una de las agrupaciones para poder generar un análisis de cada uno de los clústeres.

| clusterP | Texture | Area | Smoothness | Compactness | Symmetry | FractalDimension |
|---|---|---|---|---|---|---|
| **0** | 16.297442 | 514.286628 | 0.085941 | 0.062736 | 0.164908 | 0.059056 |
| **1** | 21.837500 | 1228.067000 | 0.100036 | 0.140695 | 0.187407 | 0.059186 |
| **2** | 20.364643 | 705.283929 | 0.115617 | 0.204721 | 0.226070 | 0.075936 |
| **3** | 17.734615 | 476.337179 | 0.104744 | 0.107066 | 0.188042 | 0.066356 |
| **4** | 24.492706 | 559.569412 | 0.085045 | 0.074626 | 0.164491 | 0.059430 |

Clúster 0: Es un grupo formado por 85 pacientes con un menor tamaño de tumor (potencialmente benigno), con un área promedio de tumor de 514 píxeles y una desviación estándar de textura de 16 pixeles. Es un tumor compacto (0.06 pixeles), cuya suavidad alcanza 0.08 píxeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 píxeles.

Clúster 1: Es un grupo formado por 85 pacientes con un mayor tamaño de tumor (potencialmente maligno), con un área promedio de tumor de 1228 píxeles y una desviación estándar de textura de 21 pixeles. Es un tumor no compacto (0.14 pixeles), cuya suavidad alcanza 0.1 pixeles, una simetría de 0.18 y una aproximación de frontera, dimensión fractal, promedio de 0.059 píxeles.

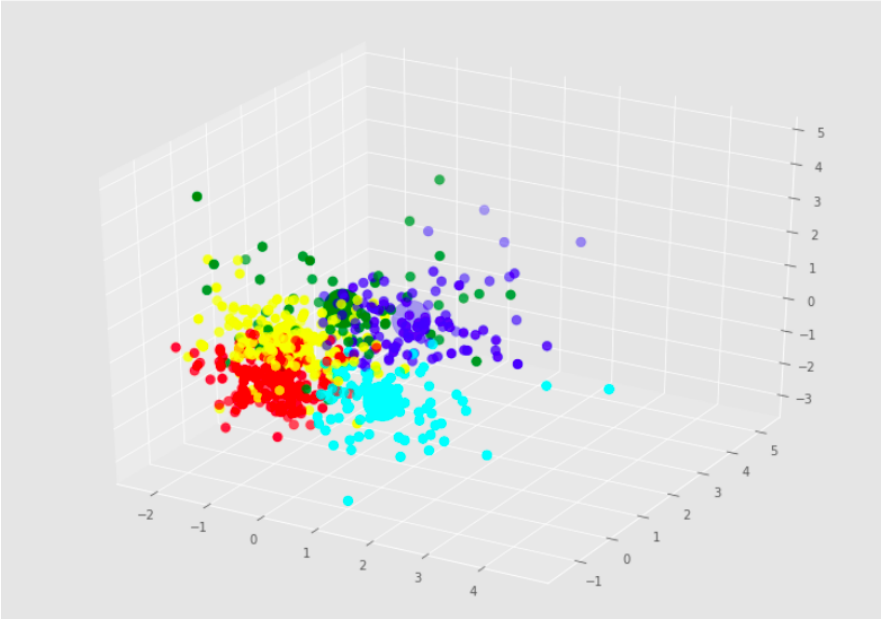
Clúster 2: Es un grupo formado por 85 pacientes con un mayor tamaño de tumor (potencialmente maligno), con un área promedio de tumor de 705 píxeles y una desviación estándar de textura de 20 píxeles. Es un tumor no compacto (0.2

pixeles), cuya suavidad alcanza 0.11 píxeles, una simetría de 0.22 y una aproximación de frontera, dimensión fractal, promedio de 0.075 píxeles.

Clúster 3: Es un grupo formado por 85 pacientes con un menor tamaño de tumor (potencialmente benigno), con un área promedio de tumor de 476 píxeles y una desviación estándar de textura de 17 pixeles. Es un tumor no compacto (0.1 pixeles), cuya suavidad alcanza 0.10 pixeles, una simetría de 0.18 y una aproximación de frontera, dimensión fractal, promedio de 0.06 píxeles.

Clúster 4: Es un grupo formado por 85 pacientes con un menor tamaño de tumor (potencialmente benigno), con un área promedio de tumor de 559 píxeles y una desviación estándar de textura de 24 pixeles. Es un tumor compacto (0.07 pixeles), cuya suavidad alcanza 0.08 píxeles, una simetría de 0.16 y una aproximación de frontera, dimensión fractal, promedio de 0.059 píxeles.

Por último se generó una gráfica tridimensional, como en la práctica 6, sobre las cinco agrupaciones que se generaron utilizando este algoritmo.



**Conclusiones**

Se cumplieron los objetivos al generar los dos modelos de agrupaciones utilizando clustering jerárquico y particional a partir de conjunto de datos de pacientes diagnosticadas con un tumor de mama. Se realizó la estandarización y selección de variables como se lleva haciendo en prácticas pasadas y este proceso sirvió

para ambos algoritmos. Posteriormente, se realizó el algoritmo jerárquico obteniendo un modelo que relaciona el conjunto de datos en cuatro clústeres. Se analizó cada conjunto para determinar si era potencialmente maligno o benigno. Después, se realizó la parte del clustering particional donde a través del método del codo se obtuvieron cinco grupos para el clustering. Con estos resultados se puede concluir que aunque ambos algoritmos buscan el clustering no siempre dan los mismos resultados. Se analizaron también los resultados que dieron los 5 clusters en el K-medias. Con esto también se concluye que la parte más significativa para este tipo de algoritmos es el análisis de los clusters generados. Gracias a esta práctica se reforzaron los algoritmos de las dos prácticas pasadas y se pudo observar las diferencias de los resultados que existen entre ambas. El procedimiento que ambas siguen no es el mismo por lo que tiene sentido que los resultados difieran.