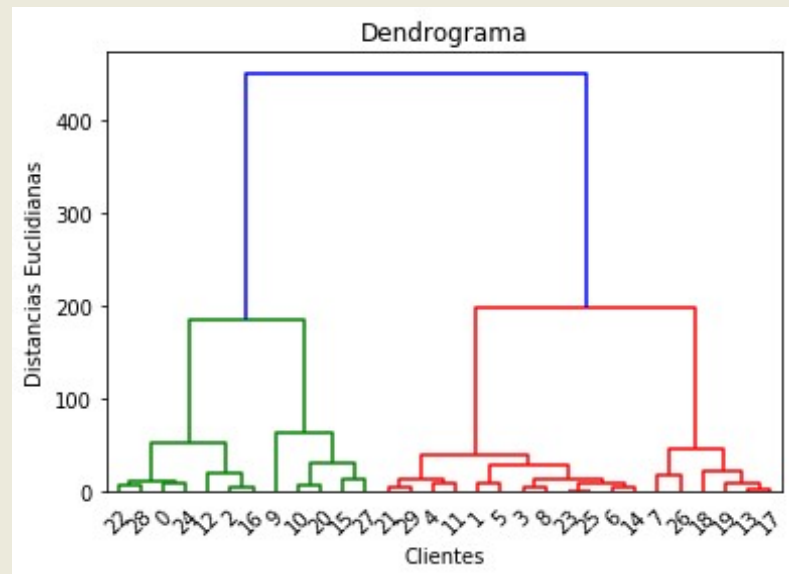


ML\_CLUSTERING\_Jerarquico\_01

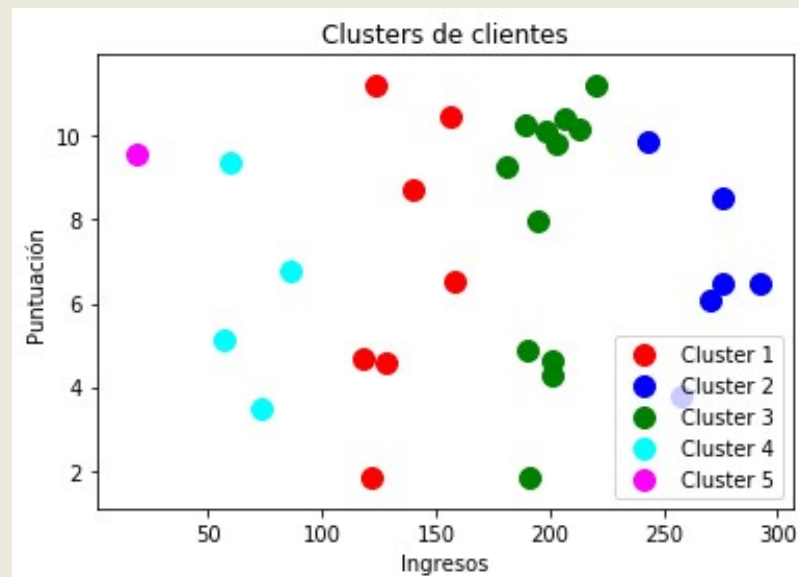
Segmentación mediante clustering jerárquico

ML

En esta práctica se realiza un dendrograma utilizando clustering jerárquico. Se parte de una base de datos de clientes de una tienda y en función de los ingresos y su puntuación se realiza una clasificación por categorías.



Como resultado se obtiene la clasificación de los clientes en diferentes clusters.



## SOLUCIÓN

Definir las librerías a utilizar

```
# Importar librerías
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Cargar el conjunto de datos y filtrar los datos necesarios (ingresos y puntuación)

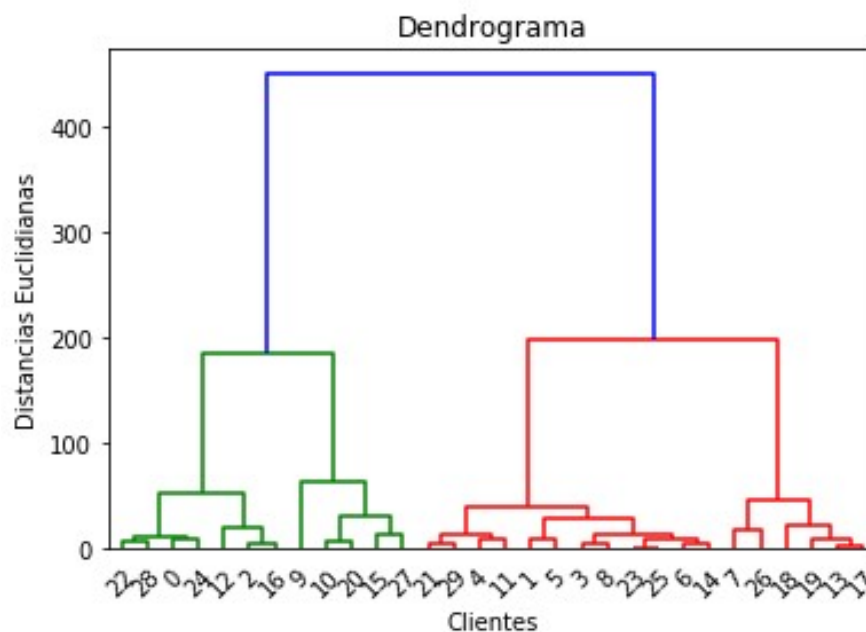
```
# Cargar el conjunto de datos
dataset = pd.read_csv('Clientes_Tienda.csv')

# Seleccionar por ingresos y puntuación
X = dataset.iloc[:, [2, 3]].values
```

Crear el dendrograma y visualizarlo

```
# Creamos el dendrograma para encontrar el número óptimo de clusters
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))

# Visualizar el dendrograma
plt.title('Dendrograma')
plt.xlabel('Clientes')
plt.ylabel('Distancias Euclidianas')
plt.show()
```



### Realizar el clustering jerárquico

```
# Realizar el clustering jerárquico ajustando al conjunto de datos
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
```

### Hacer la predicción sobre el conjunto X

```
# Predicción del cluster
y_hc = hc.fit_predict(X)
```

### Visualizar los clusters

```
# Visualizar los clusters
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')

plt.title('Clusters de clientes')
plt.xlabel('Ingresos')
plt.ylabel('Puntuación')
plt.legend()
plt.show()
```

