Universidad de Córdoba

Escuela Politécnica Superior de Córdoba

Grado de Ingeniería Informática - Mención en computación

Tercer curso - Segundo cuatrimestre - 2020/2021

Introducción al Aprendizaje Automático

Práctica 4: Análisis cluster

Profesor: *Nicolás Emilio García Pedrajas*

Autor: *Ventura Lucena Martínez*

Córdoba, 2 de julio de 2021

Índice

Listings

Índice de figuras

1 Ejercicio 0

Ejecute los programas ejemplo facilitados junto con la práctica para familiarizarse con los conceptos de los algoritmos de clustering de scikit-learn.

Algunas de las ejecuciones propuestas nos dejan con los siguientes resultados:

1.1 *plot_affinity_propagation.py*

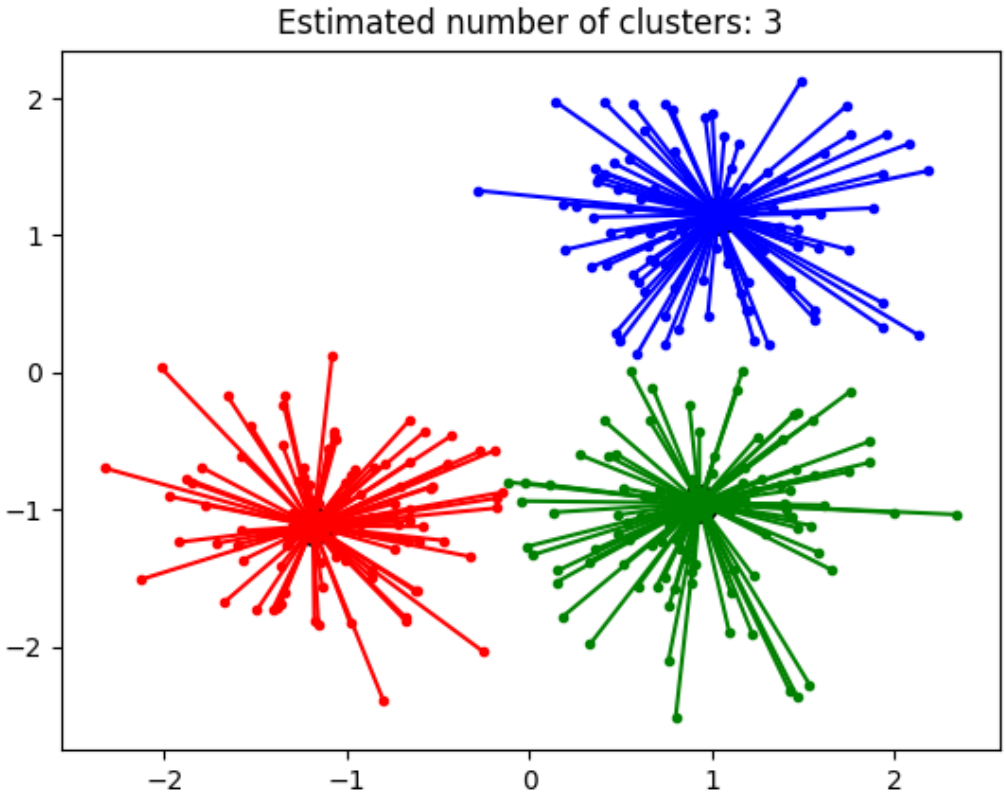

Figura 1: Resultados *plot_affinity_propagation.py*.

Listing 1: **Resultados de algoritmo de clúster *Affinity propagation.***

```
Estimated number of clusters: 3
Homogeneity: 0.872
```

```
Completeness: 0.872
V-measure: 0.872
Adjusted Rand Index: 0.912
Adjusted Mutual Information: 0.871
Silhouette Coefficient: 0.753
```

1.2 *plot_cluster_iris.py*

Los gráficos muestran en primer lugar lo que produciría un algoritmo de K-medias utilizando tres grupos. Luego se muestra cuál es el efecto de una mala inicialización en el proceso de clasificación: Al establecer *n_init* en solo 1 (el valor predeterminado es 10), se reduce la cantidad de veces que el algoritmo se ejecutará con diferentes semillas de centroide. El siguiente gráfico muestra lo que proporcionaría el uso de ocho grupos y, finalmente, el *ground truth*.

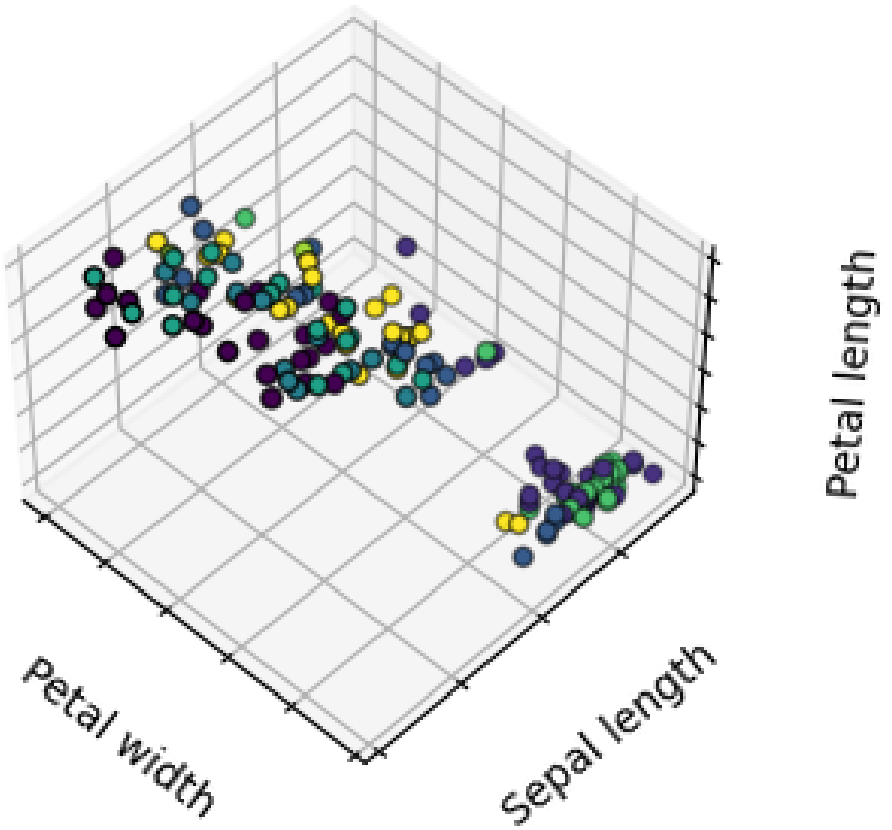

Figura 2: Resultados *plot_cluster_iris.py* (1).

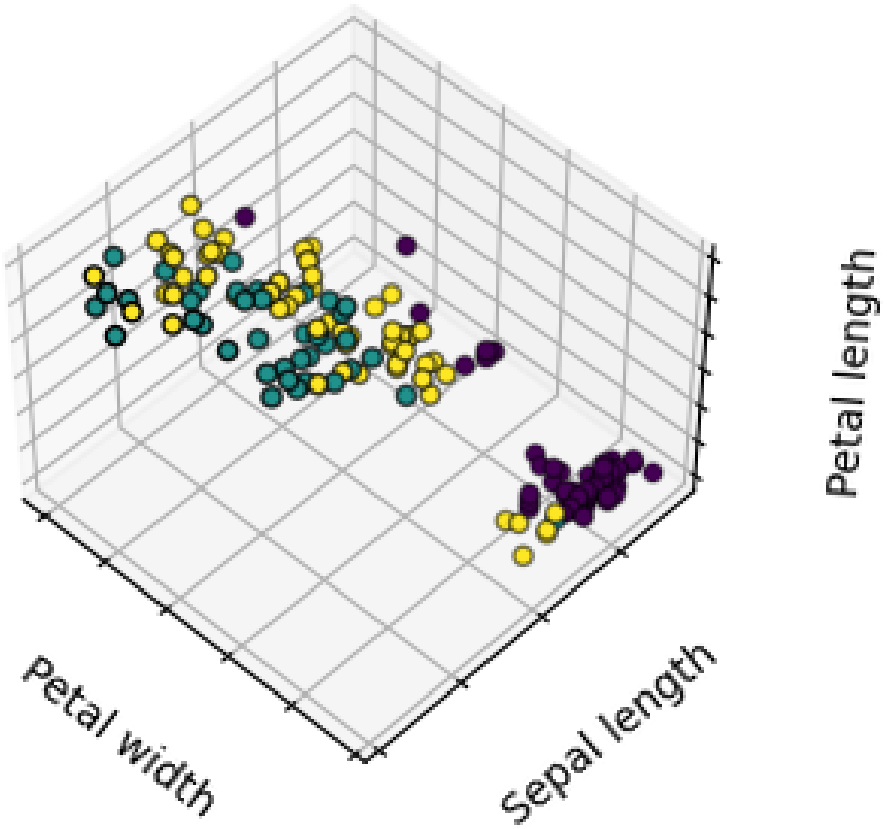

Figura 3: Resultados *plot_cluster_iris.py* (2).

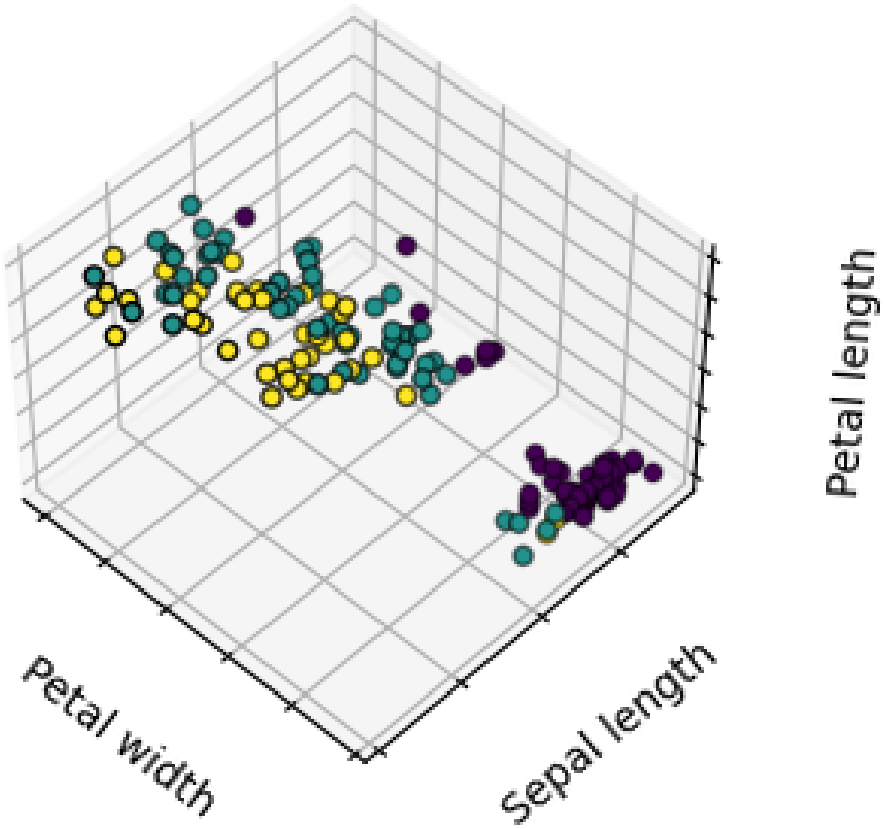

Figura 4: Resultados *plot_cluster_iris.py* (3).

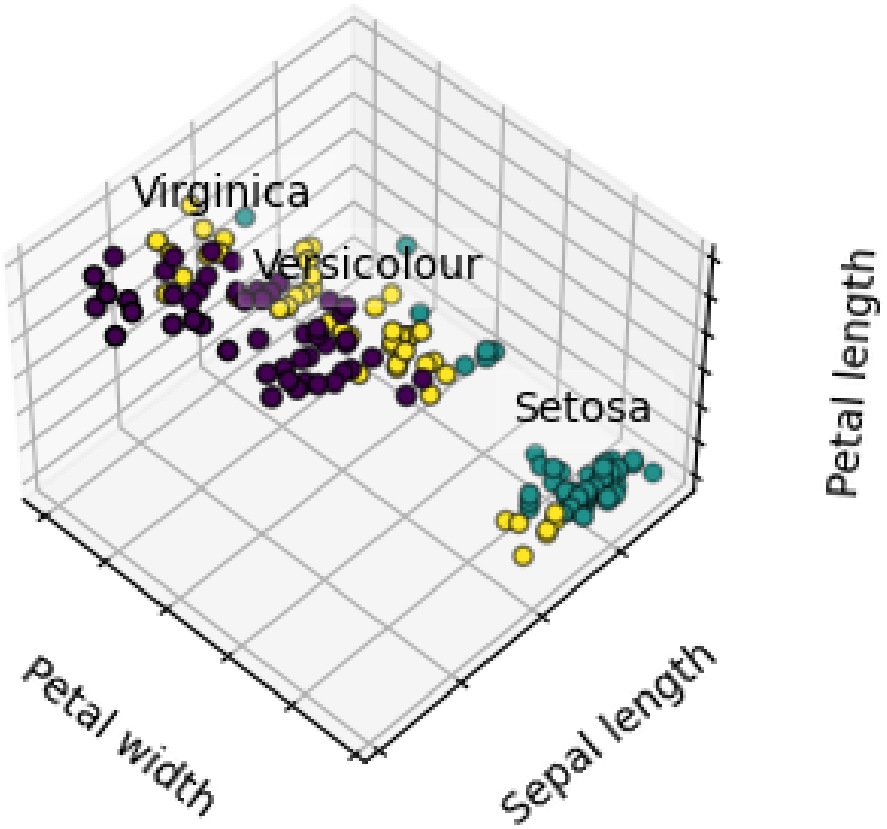

Figura 5: Resultados *plot_cluster_iris.py* (4).

1.3 *plot_dbscan.py*

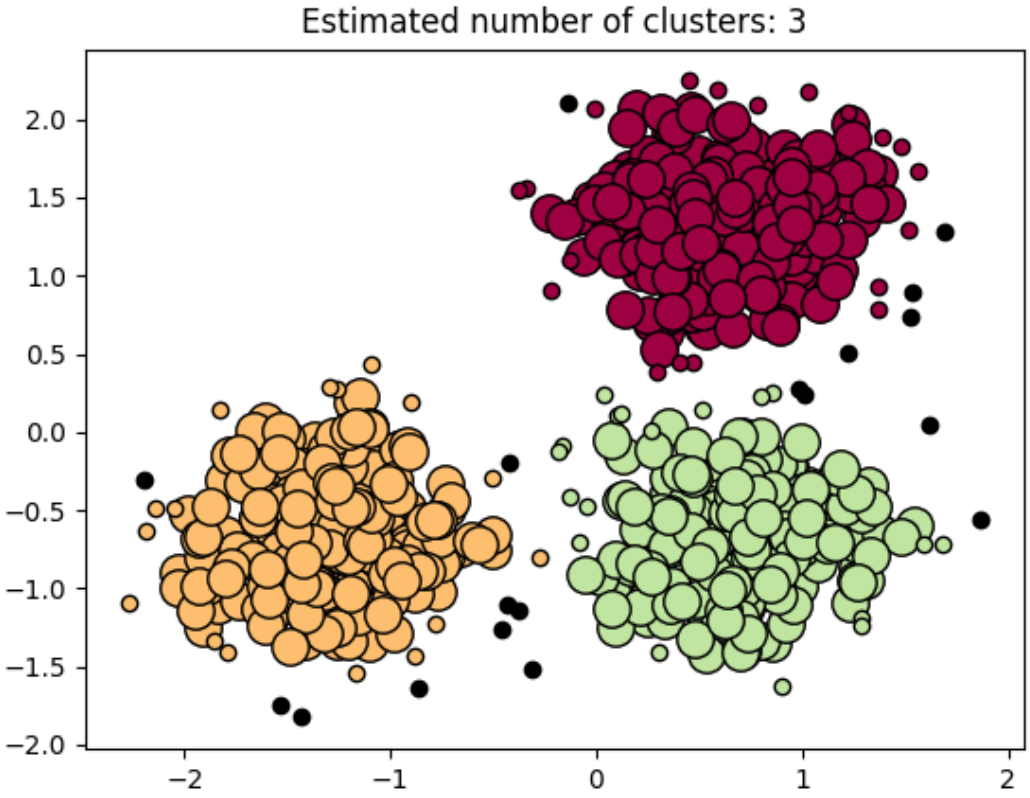

Figura 6: Resultados *plot_dbscan.py*.

Listing 2: **Resultados de algoritmo de clúster *DBSCAN*.**

```
Estimated number of clusters: 3
Estimated number of noise points: 18
Homogeneity: 0.953
Completeness: 0.883
V-measure: 0.917
Adjusted Rand Index: 0.952
Adjusted Mutual Information: 0.916
Silhouette Coefficient: 0.626
```

1.4 *plot_kmeans_assumptions.py*

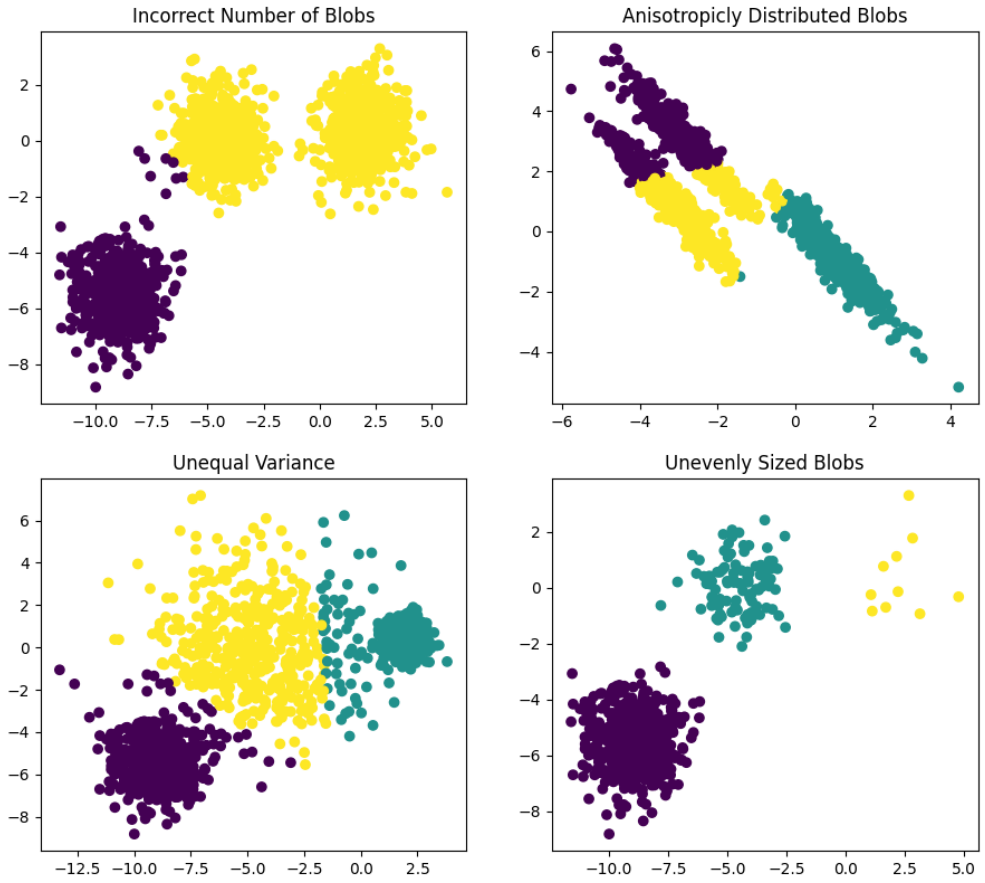

Figura 7: Resultados *plot_kmeans_assumptions.py*.

1.5 *plot_mean_shift.py*

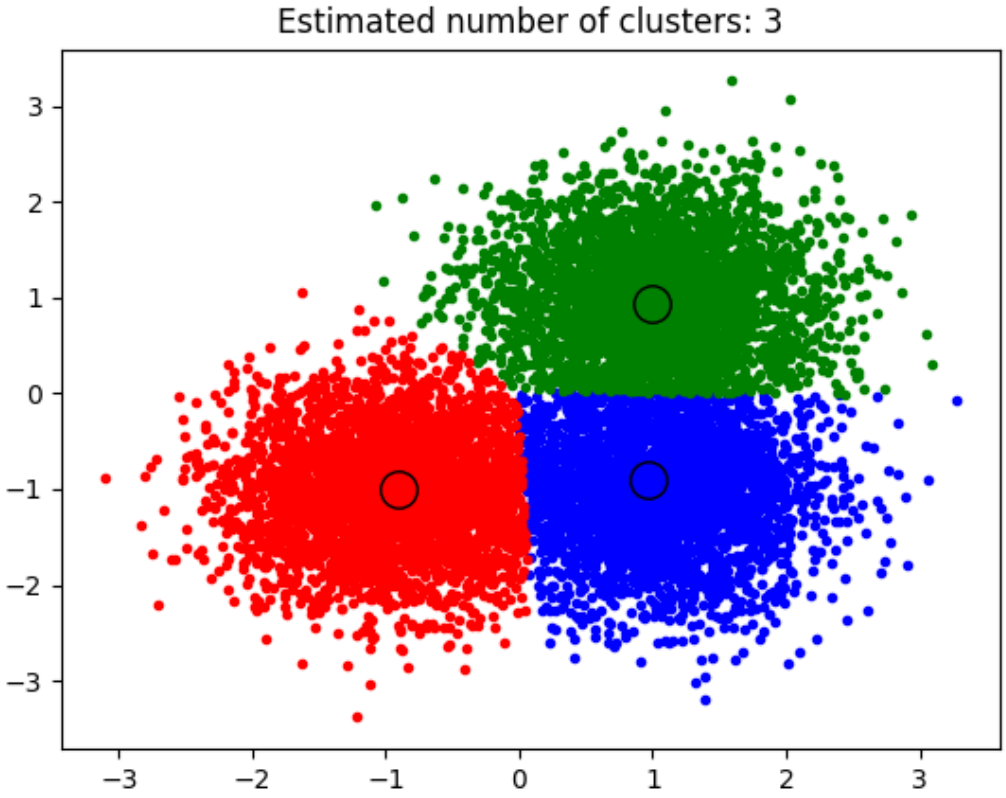

Figura 8: Resultados *plot_mean_shift.py*.

1.6 *plot_segmentation_toy*

En este ejemplo, se genera una imagen con círculos conectados y la agrupación espectral se utiliza para separar los círculos.

En estas configuraciones, el enfoque: ref: ‘spectral_clustering‘ resuelve el problema conocido como *normalized graph cuts*: la imagen se ve como un gráfico de vóxeles conectados, y el algoritmo de agrupamiento espectral equivale a elegir cortes de gráfico que definen regiones mientras se minimiza la relación del gradiente a lo largo del corte y el volumen de la región.

A medida que el algoritmo intenta equilibrar el volumen (es decir, equilibrar los tamaños

de las regiones), si tomamos círculos con diferentes tamaños, la segmentación falla.

Además, como no hay información útil sobre la intensidad de la imagen o su gradiente, se opta por realizar el agrupamiento espectral en un gráfico que solo está débilmente informado por el gradiente. Esto está cerca de realizar una partición Voronoi del gráfico.

Además, se hace uso de la máscara de los objetos para restringir el gráfico al contorno de los objetos. En este ejemplo, estamos interesados en separar los objetos unos de otros, y no del fondo.

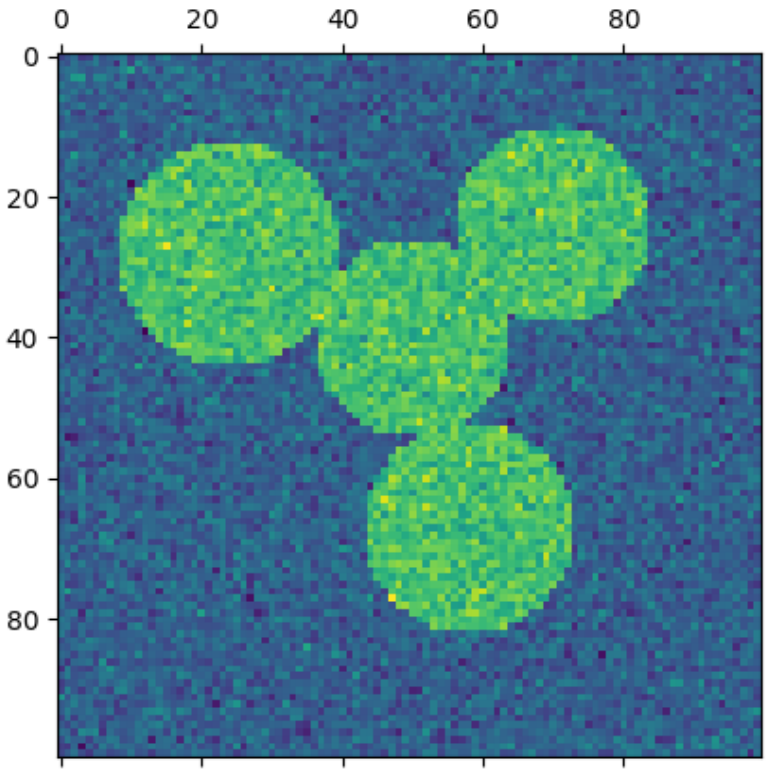

Figura 9: Resultados *plot_segmentation_toy* (1).

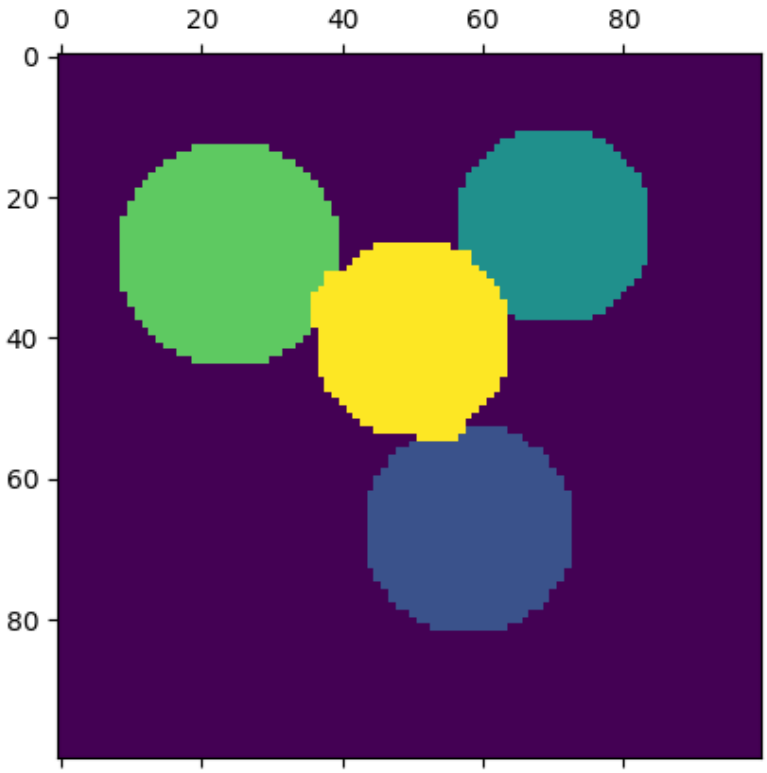

Figura 10: Resultados *plot_segmentation_toy* (2).

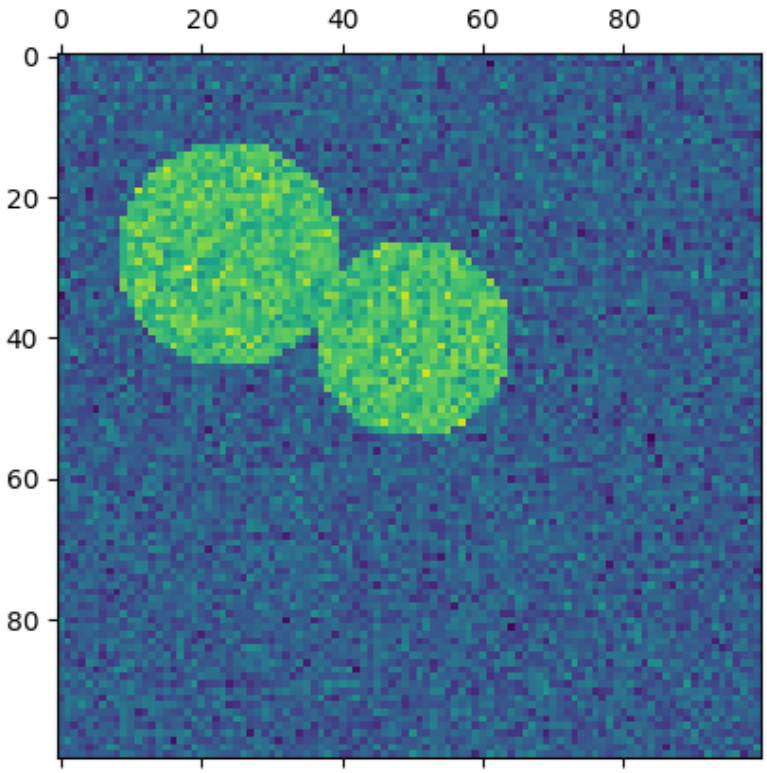

Figura 11: Resultados *plot_segmentation_toy* (3).

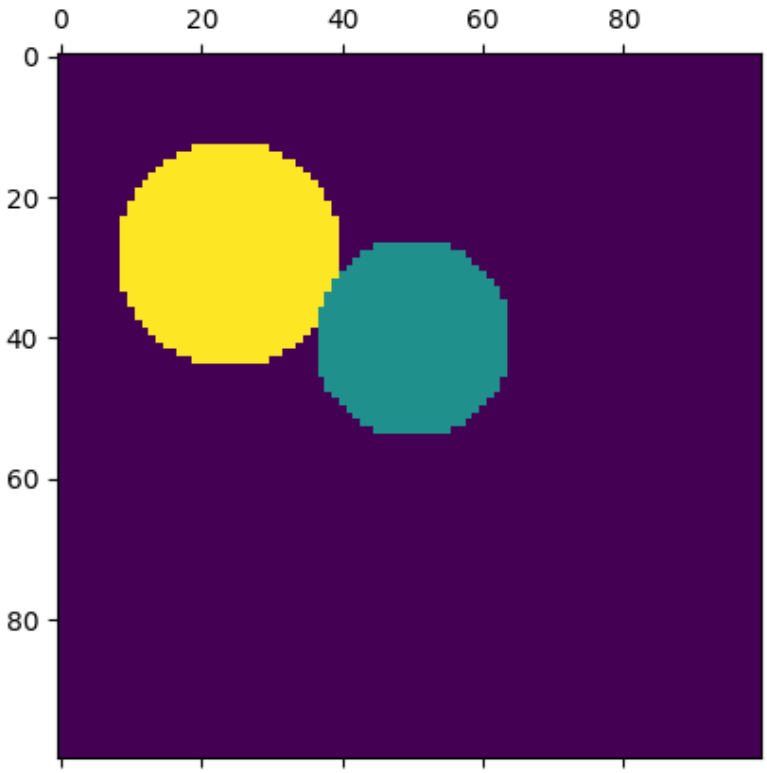

Figura 12: Resultados *plot_segmentation_toy* (4).

2 Ejercicio 1

Seleccione al menos cinco problemas de los disponibles en lo repositorios usados en las prácticas anteriores. Use problemas que solo contengan atributos numéricos. No olvide eliminar la información de la clase antes de ejecutar los algoritmos.

Los ficheros elegidos han sido los siguientes:

- cpu.arff.
- diabetes.arff.
- glass.arff.

- ionosphere.arff.
- iris.arff.

3 Ejercicio 2

Seleccione el algoritmo de clustering *k-means*.

Para llevar a cabo el algoritmo de clustering *K-Medias*, se ha procedido a realizar los siguientes pasos sobre un conjunto de datos cualquiera:

1. Importar el conjunto de datos.
2. Normalizar los datos.
3. Aplicar el método de Elbow para obtener el número óptimo de clústers.
4. Análisis de componentes principales.
5. Representación de resultados.

Los resultados obtenidos sobre el conjunto de datos *glass.arff* son los siguientes:

Listing 3: **Resultados de algoritmo de clúster *K-Medias*.**

```
            RI         Na         Mg         Al         Si         'K'         Ca    Ba
               Fe
0     0.297629   0.309774   0.779510   0.258567   0.575000   0.103060   0.310409   0.0
    0.000000
1     0.231782   0.215038   0.783964   0.330218   0.550000   0.091787   0.288104   0.0
    0.000000
2     0.297629   0.372932   0.775056   0.348910   0.505357   0.095008   0.278810   0.0
    0.000000
3     0.080773   0.551880   0.387528   0.389408   0.846429   0.000000   0.200743   0.0
    0.000000
4     1.000000   0.236090   0.000000   0.221184   0.062500   0.019324   1.000000   0.0
    0.470588
..         ...        ...        ...        ...        ...        ...        ...   ...
          ...
209   0.217296   0.404511   0.757238   0.289720   0.514286   0.095008   0.268587   0.0
    0.000000
210   0.209394   0.320301   0.783964   0.570093   0.508929   0.111111   0.236059   0.0
    0.000000
211   0.218613   0.479699   0.783964   0.299065   0.548214   0.059581   0.233271   0.0
    0.274510
212   0.251975   0.291729   0.641425   0.442368   0.607143   0.117552   0.289033   0.0
    0.000000
```

```
213   0.323529   0.505263   0.487751   0.426791   0.510714   0.000000   0.361524   0.0
    0.000000

[214 rows x 9 columns]
```

```
                RI           Na           Mg           Al           Si          'K'
                             Ca           Ba           Fe
count   214.000000   214.000000   214.000000   214.000000   214.000000   214.000000
    214.000000   214.000000   214.000000
mean      0.316744     0.402684     0.597891     0.359784     0.507310     0.080041
       0.327785     0.055570     0.111783
std       0.133313     0.122798     0.321249     0.155536     0.138312     0.105023
       0.132263     0.157847     0.191056
min       0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
       0.000000     0.000000     0.000000
25%       0.235843     0.327444     0.471047     0.280374     0.441071     0.019726
       0.261152     0.000000     0.000000
50%       0.286655     0.386466     0.775056     0.333333     0.532143     0.089372
       0.294610     0.000000     0.000000
75%       0.351514     0.465414     0.801782     0.417445     0.585268     0.098229
       0.347816     0.000000     0.196078
max       1.000000     1.000000     1.000000     1.000000     1.000000     1.000000
       1.000000     1.000000     1.000000
```

	RI	Na	Mg	Al	Si	'K'	Ca	Ba	Fe	KMeans_Clusters
0	1.51793	12.79	3.50	1.12	73.03	0.64	8.77	0.0	0.00	1
1	1.51643	12.16	3.52	1.35	72.89	0.57	8.53	0.0	0.00	1
2	1.51793	13.21	3.48	1.41	72.64	0.59	8.43	0.0	0.00	1
3	1.51299	14.40	1.74	1.54	74.55	0.00	7.59	0.0	0.00	1
4	1.53393	12.30	0.00	1.00	70.16	0.12	16.19	0.0	0.24	0

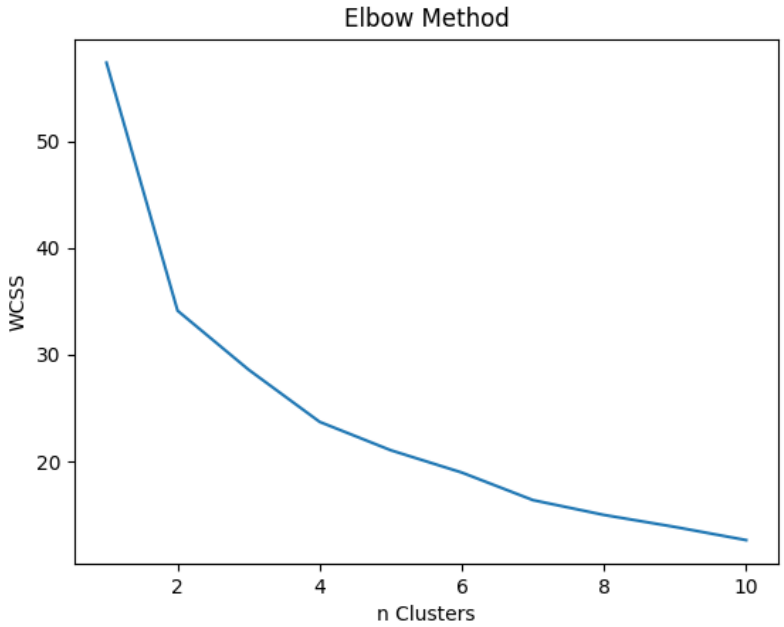

Figura 13: Resultados aplicación de *K-Medias* sobre el conjunto de datos *glass.arff* (1).

Observando la gráfica, se puede observar que el número óptimo de clústers es 2.

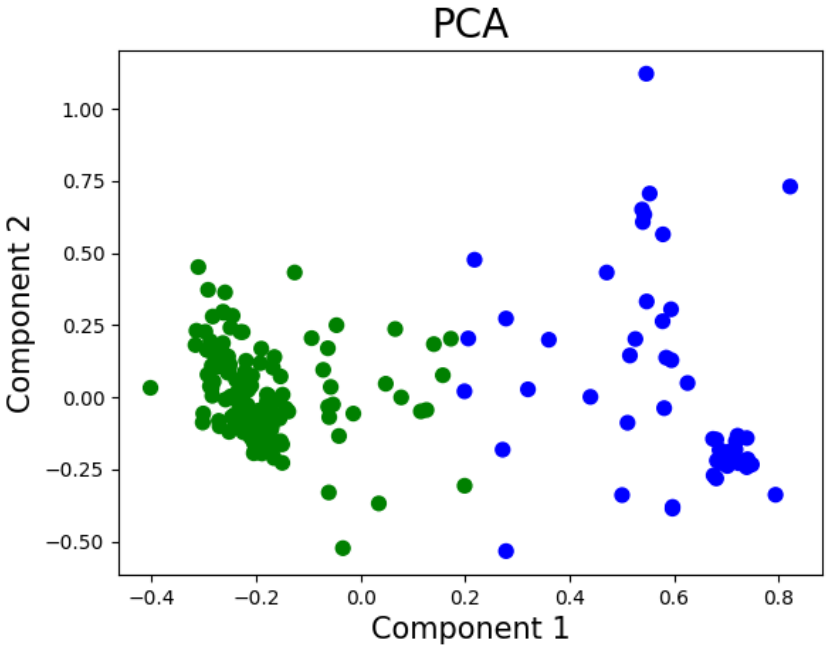

Figura 14: Resultados aplicación de *K-Medias* sobre el conjunto de datos *glass.arff* (2).

4 Ejercicio 3

Selecciones los algoritmos de clustering jerárquicos *single link*, *complete link* y *average link*.

La aplicación del algorítmo, según el método, sobre el conjunto de datos utilizado anteriormente queda como sigue, teniendo en cuenta que la aplicación del diagrama de dispersión se realiza sobre dos columnas del dataset seleccionadas aleatoriamente:

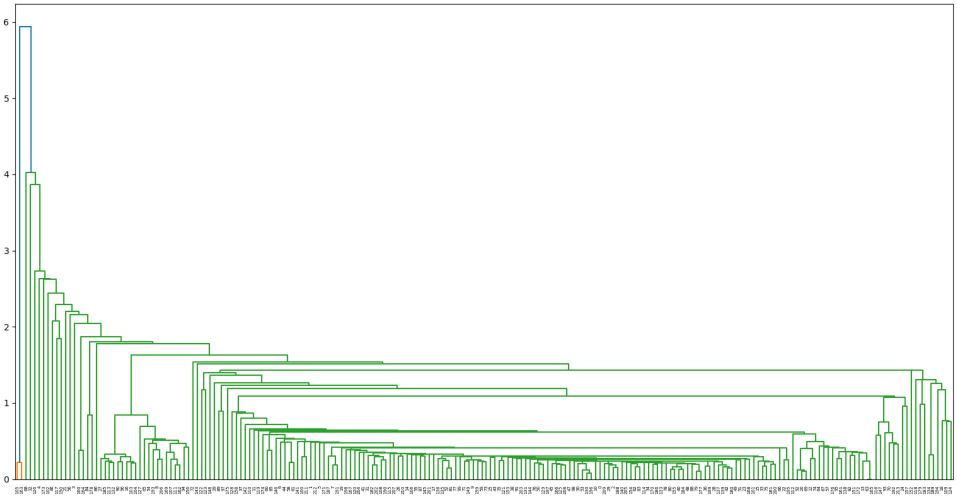

Figura 15: Resultados aplicación de *single link* sobre el conjunto de datos *glass.arff* (1).

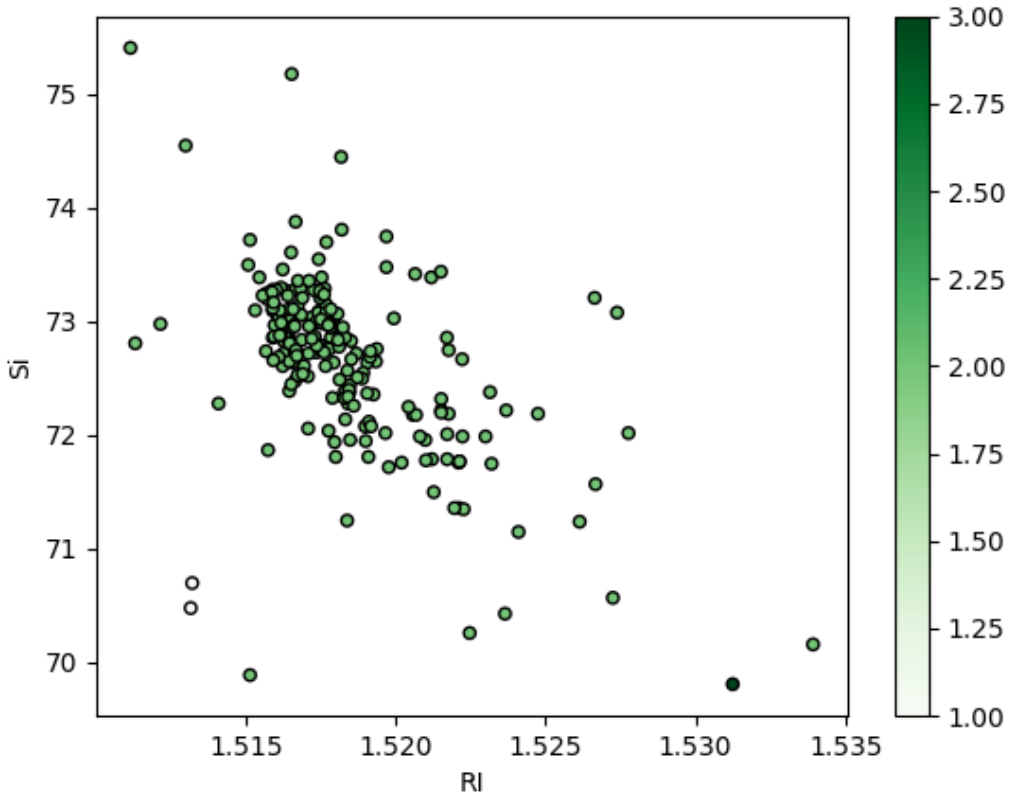

Figura 16: Resultados aplicación de *single link* sobre el conjunto de datos *glass.arff* (2).

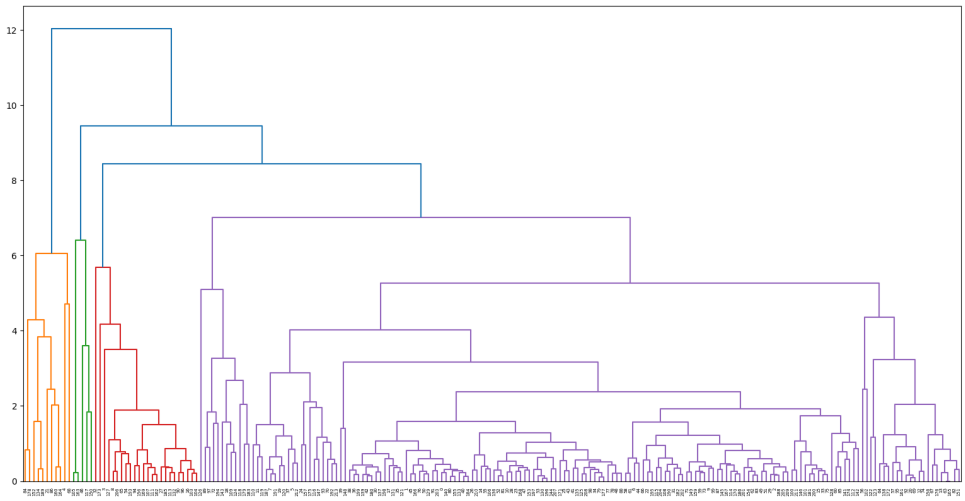

Figura 17: Resultados aplicación de *complete link* sobre el conjunto de datos *glass.arff* (1).

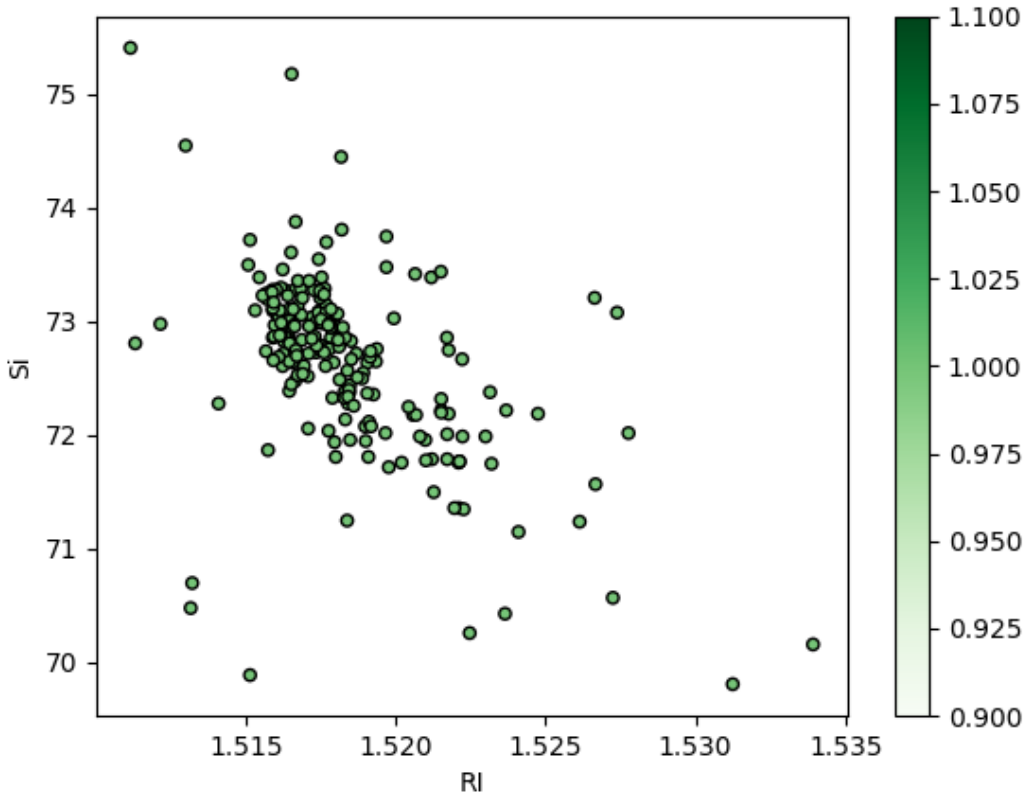

Figura 18: Resultados aplicación de *complete link* sobre el conjunto de datos *glass.arff* (2).

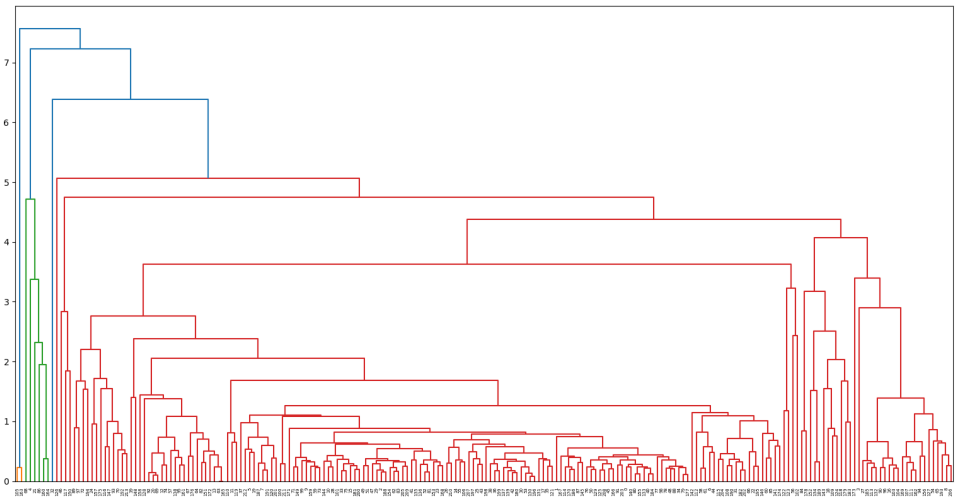

Figura 19: Resultados aplicación de *average link* sobre el conjunto de datos *glass.arff* (1).

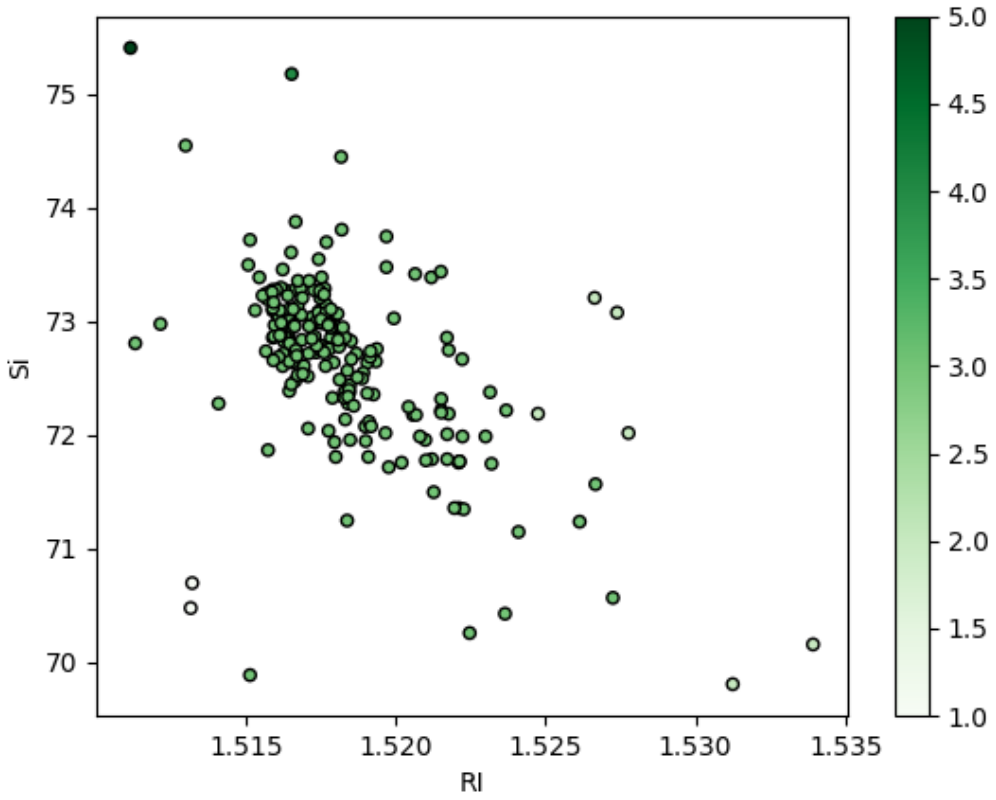

Figura 20: Resultados aplicación de *average link* sobre el conjunto de datos *glass.arff* (2).

Para ello, se ha modificado el atributo *method* de la siguiente línea de código:

Listing 4: **Clústering jerárquico.**

```
h_clustering = sch.linkage(attributes, method="single")
h_clustering = sch.linkage(attributes, method="complete")
h_clustering = sch.linkage(attributes, method="average")
```

5 Ejercicio 4

Implemente la medida de evaluación de la calidad de un método de agrupación basada en la correlación entre la matriz de incidencia y la de proximidad. Seleccione una de las medidas de evaluación no supervisada disponibles en scikit-learn.

La medida seleccionada ha sido el coeficiente de Silhouette. Se ha utilizado el ejemplo provisto en la práctica para su testeo. La línea de código a aplicar sería la siguiente, donde *df* hace mención al conjunto de datos y *clusters* a los distintos grupos o etiquetas:

Listing 5: **Coeficiente de Silhouette.**

```
print("Silhouette Coefficient: %0.3f"% metrics.silhouette_score(df, clusters,
    metric='euclidean'))
```

6 Ejercicio 5

Para cada uno de los problemas seleccionados realice las siguientes tareas:

1. **Ejecute el algoritmo *k-means* y evalúe su rendimiento para un rango de valores alrededor del número conocido de clases.**
2. **Ejecute los algoritmos jerárquicos y evalúe su rendimiento al nivel en el cual tengan el mismo número de grupos que el número de clases del problema.**

Referencias

[1] Moodle Universidad de Córdoba - Enunciado Práctica 4.

[2] Moodle Universidad de Córdoba - Introducción al clústering con scikit-learn.