Inteligencia de Negocio
Práctica 2. Análisis Relacional mediante Segmentación

Pablo Jesús Jiménez Ortiz
Grupo IN2 (Jueves)

Diciembre de 2019

Contents

1 Introducción

Bajo la premisa de introducirnos en el uso de técnicas de aprendizaje no supervisado para análisis relacional mediante segmentación, en esta práctica estudiaremos cómo aplicar distintos algoritmos de agrupamiento (o **clustering**) a un conjunto de datos dado.

El conjunto de datos a analizar contiene variables de tipo categórico que nos serán útiles a la hora de delimitar diferentes casos de estudio y distinguir de manera adecuada unos de otros. Sin embargo, no debemos tener en cuenta este tipo de variables a la hora de aplicar un análisis de clustering. A su vez, también contiene variables numéricas y ordinales que si serán de utilidad en el momento de aplicar el agrupamiento de variables.

Las técnicas de clustering que se usarán en esta práctican pertenecen a lo que se conoce como técnicas de aprendizaje no supervisado y en esta línea, se han considerado algoritmos de clustering clásicos como Kmeans así como algunos más novedosos para aportar diversidad a los resultados y de esta forma poder interpretar y comparar el comportamiento de dichos algoritmos de forma más adecuada, además de destacar con esto la necesidad de seguir avanzando en el desarrollo de este tipo de algoritmos.

1.1 Algoritmos usados

- **MeanShift**: El desplazamiento medio es una técnica no paramétrica de análisis de características y espacio para localizar los máximos de una función de densidad. Los dominios de aplicación incluyen el análisis de conglomerados en la visión por computador y el procesamiento de imágenes. [1]

- **DBSCAN**: Se basa en noción intuitiva de "clusters" y "ruido". La idea clave es que para cada punto de un cluster, la vecindad de un radio dado tiene que contener al menos un número mínimo de puntos, marcando como puntos atípicos los que se encuentran solos en regiones de baja densidad. Se utiliza habitualmente en minería de datos y aprendizaje automático. [2]

- **KMeans**: K-means almacena $k$centroides que utiliza para definir clusters. Se considera que un punto está en un cluster en particular si está más cerca del centroide de ese cluster que de cualquier otro centroide. Es un método utilizado en minería de datos y es considerada una de las técnicas de aprendizaje no supervisado más simples y populares. [3]

- **Agglomerative Clustering**: realiza una agrupación jerárquica utilizando un enfoque ascendente: cada observación comienza en su propio cluster, y los clusters se fusionan sucesivamente. El criterio de enlace en nuestro caso es **ward**, que minimiza la suma de las diferencias al cuadrado dentro de todos los clusters. [4]

- **Gaussian Mixture (GMM)**: Los modelos de mezcla gaussianos son un modelo probabilístico para representar clusters distribuidos normalmente

dentro de una población total. Los modelos mixtos en general no requieren saber a qué cluster pertenece un punto de los datos, permitiendo que el modelo aprenda los clusters automáticamente. Dado que no se conoce la asignación de clusters, se trata de una forma de aprendizaje no supervisado. [5]

2 Caso de estudio 1. Mujeres de más de 40, españolas, que no tienen trabajo ni ella ni la pareja y tienen dificultad para llegar a fin de mes

Para este caso de estudio se ha escogido un grupo de gente que recoge las características de personas mayores de 40 años y que por diferentes situaciones se han quedado sin trabajo tanto ella como su pareja y debido a esto tienen dificultad para llegar a fin de mes.

Se pretende mostrar con este caso de estudio a la gente que por lo general se encuentra en la mitad de su vida laboral, que tiene experiencia y a pesar de esto no consigue una estabilidad económica.

A continuación se presentan los resultados obtenidos para cada uno de los algoritmos.

2.1 MeanShift

En este caso Meanshift separa el conjunto de datos en 2 clusters diferentes.

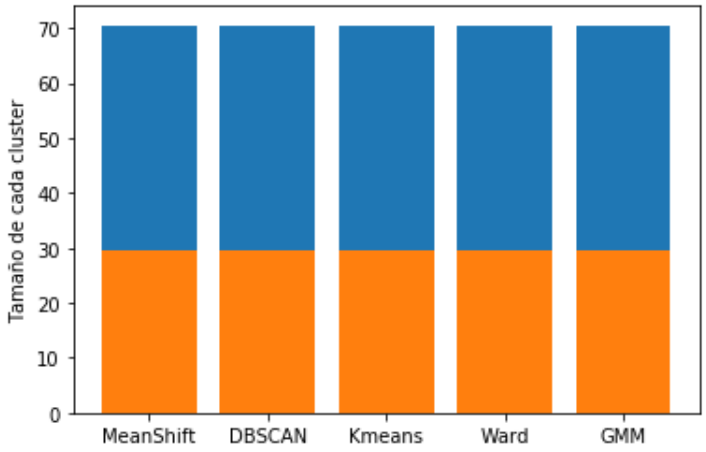

Figure 1: Tamaño de cada cluster

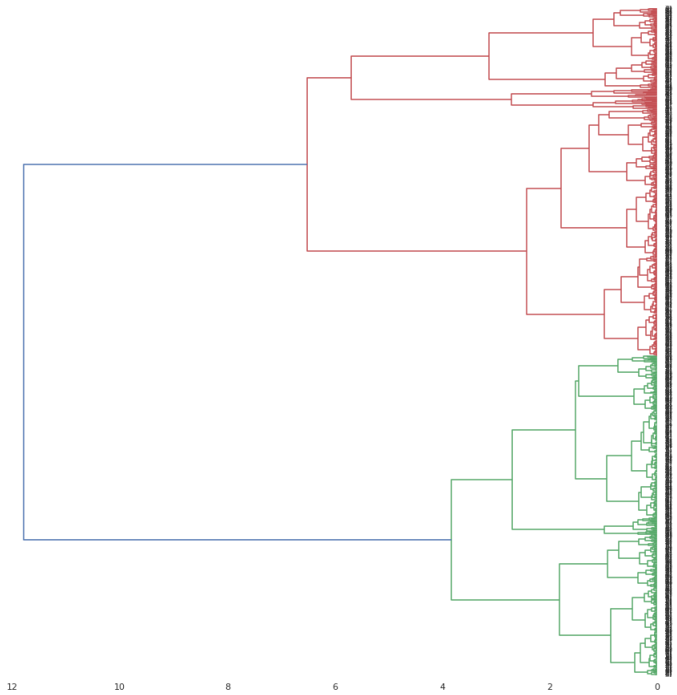

Figure 2: Dendrograma obtenido para el algoritmo Meanshift

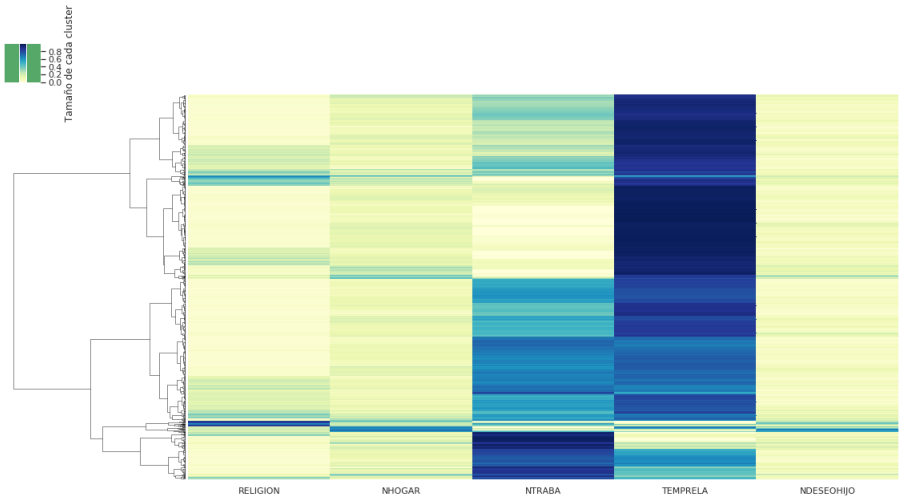

Figure 3: Clustermap obtenido para el algoritmo MeanShift

2.2 DBSCAN

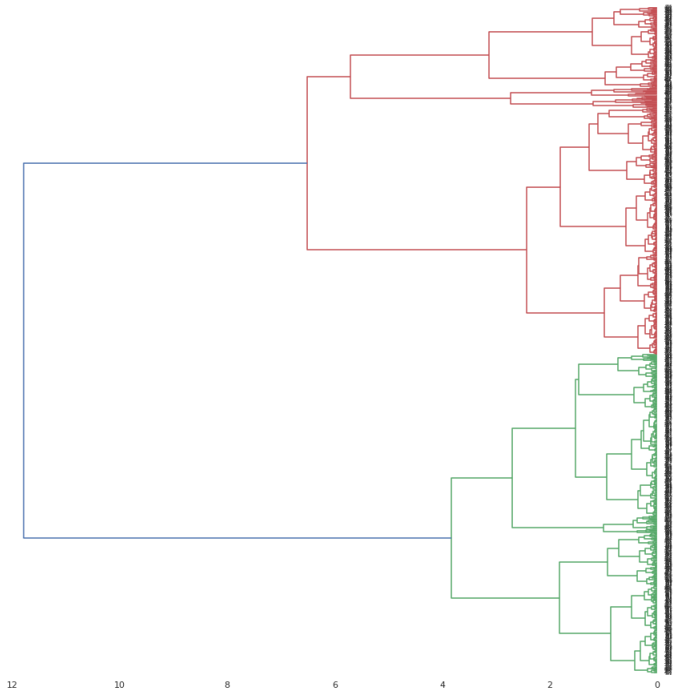

Figure 4: Dendrograma obtenido para el algoritmo DBSCAN

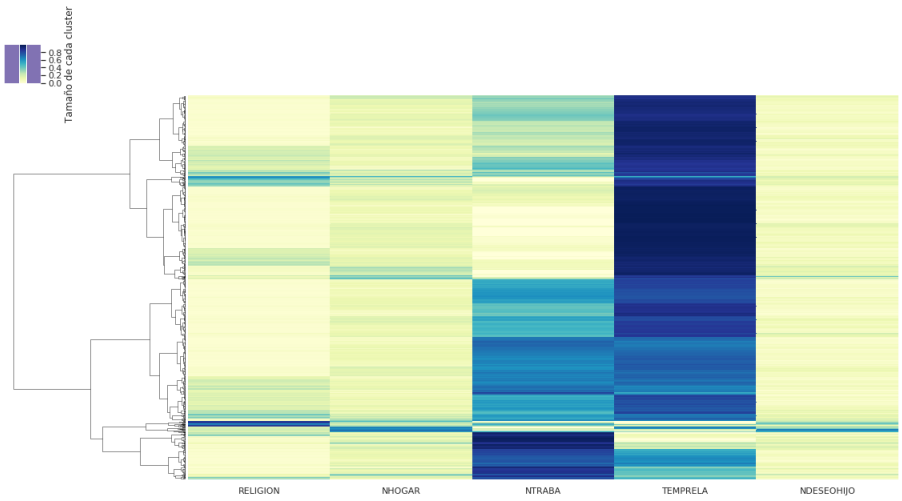

Figure 5: Clustermap obtenido para el algoritmo DBSCAN

Número de cluster	Número de filas	Porcentaje de los datos
2:	257	(27.58%)
1:	212	(22.75%)
3:	189	(20.28%)
0:	164	(17.60%)

Table 1: Porcentaje de datos para cada cluster

2.3 KMeans

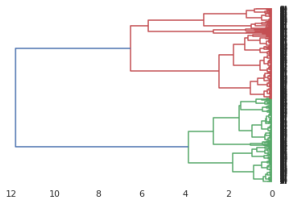

Figure 6: Dendrograma obtenido para el algoritmo KMeans

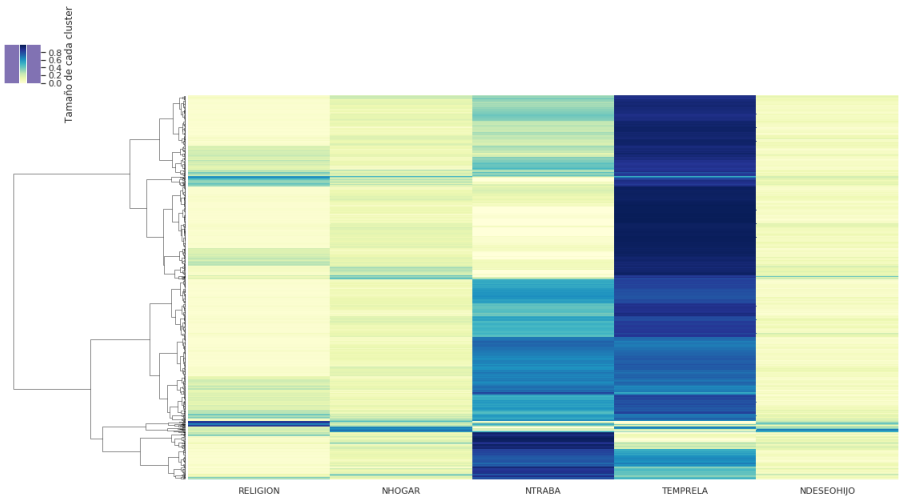

Figure 7: Clustermap obtenido para el algoritmo KMeans

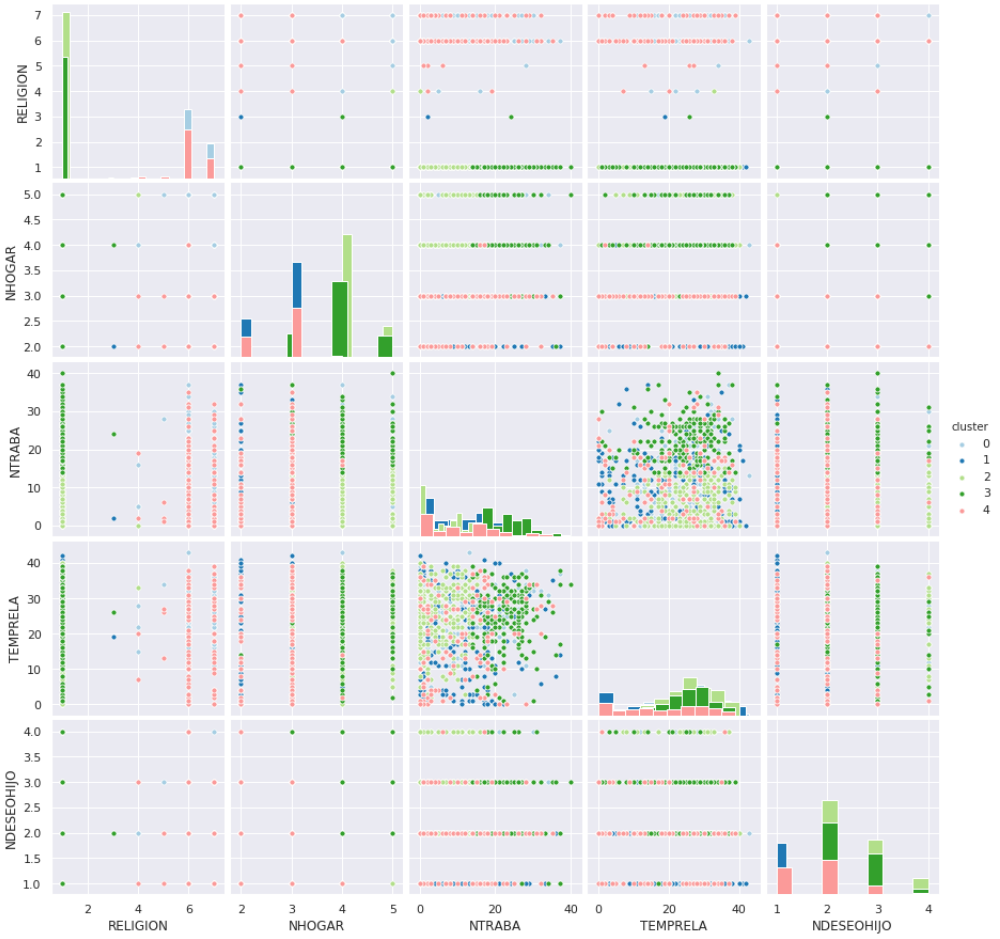

Figure 8: Scatter Matrix obtenida para el algoritmo KMeans

2.4 Agglomerative Clustering

Número de cluster	Número de filas	Porcentaje de los datos
0:	272	(29.18%)
2:	262	(28.11%)
1:	157	(16.85%)
4:	122	(13.09%)

Table 2: Porcentaje de datos de cada cluster

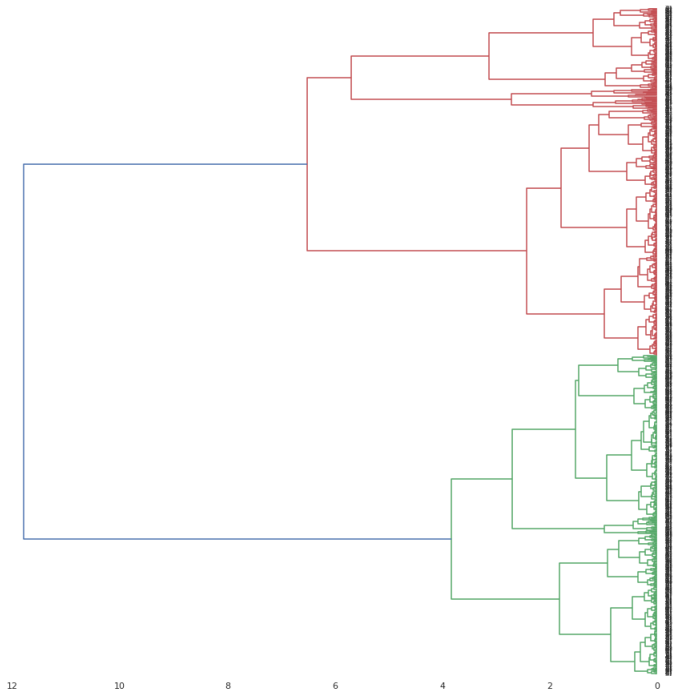

Figure 9: Dendrograma obtenido para el algoritmo Agglomerative Clustering

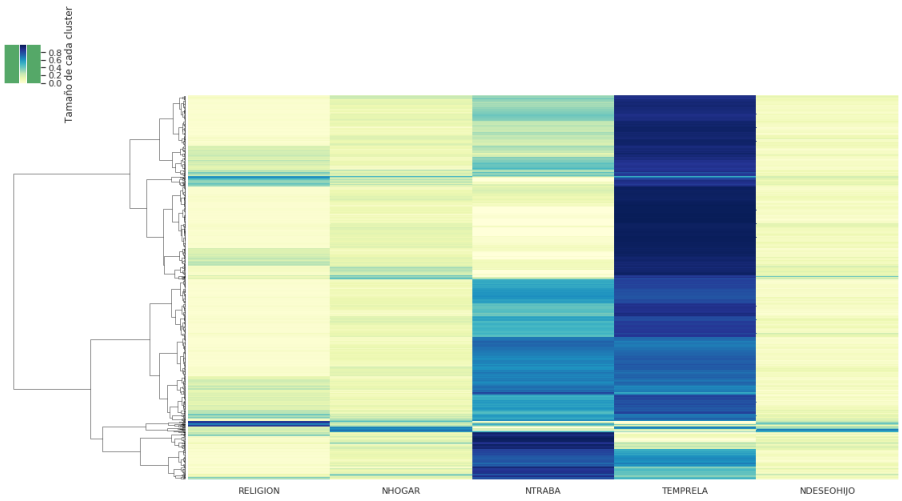

Figure 10: Clustermap obtenido para el algoritmo Agglomerative Clustering

2.5 Gaussian Mixture

Número de cluster	Número de filas	Porcentaje de los datos
1:	393	(42.17%)
2:	277	(29.72%)
0:	262	(28.11%)

Table 3: Porcentaje de datos de cada cluster

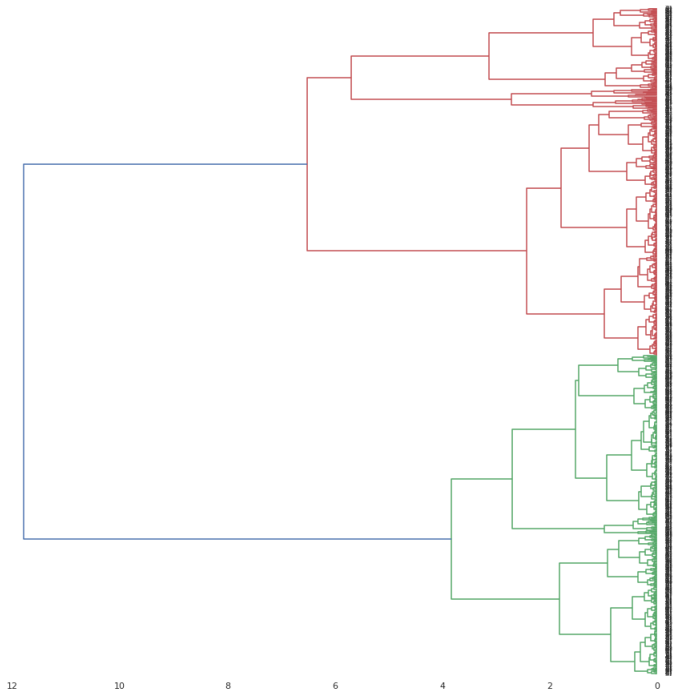

Figure 11: Dendrograma obtenido para el algoritmo Gaussian Mixture

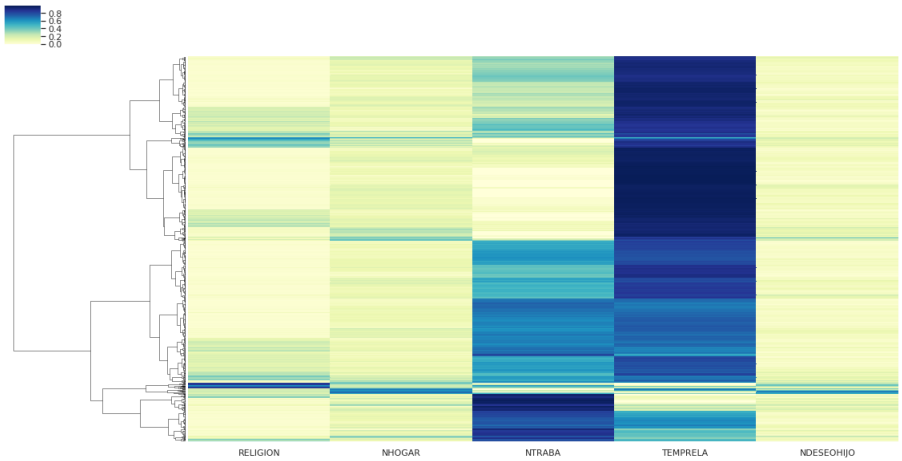

Figure 12: Clustermap obtenido para el algoritmo Gaussian Mixture

2.6 Interpretación de resultados

Algoritmo	Tiempo	Calinski-Harabaz	Silhouette Coefficient
DBSCAN	0.01 segundos	93.766	-0.01197
MeanShift	2.14 segundos	614.508	0.41637
KMeans	0.02 segundos	347.956	0.22886
Ward	0.02 segundos	329.499	0.21258
GMM	0.01 segundos	476.597	0.29375

Table 4: Resultados obtenidos en el caso de Estudio 1

En este análisis vemos que los mejores resultados los obtiene el algoritmo KMeans, obteniendo mejores marcas en las métricas **Calinski-Harabaz** y **Silhouette** que el resto de algoritmos sin embargo se ve penalizado por su elevado tiempo de ejecución si lo comparamos con sus competidores.

3 Caso de estudio 2

4 Caso de estudio 3

5 Contenido adicional

Con el objetivo de mejorar los resultados y en esta línea, de mejorar la visualización de resultados, se han eliminado los posibles outliers que no se encuentran entre el cuartil 25 y cuartil 75. De esta forma, si tenemos un caso concreto que se sale de lo normal y dentro de un conjunto de datos grande podemos eliminar esos pocos casos concretos ya que no aportan mucha información. En esta situación y a modo de resumen, lo que se ha hecho ha sido quitar estos valores "raros" y después aplicar clustering.

Para los datos usados, estos outliers eliminados representan un porcentaje pequeño de los datos. A continuación se muestra el resultado de la aplicación de dicho filtro en el caso de estudio 1:

```
Aplicando filtro anti outliers...

Resultados:
Tamaño del dataset con outliers: 32070
Tamaño del dataset sin outliers: 31790
```

Figure 13: Eliminación de Outliers

References

[1] Meanshift Algorithm
`https://en.wikipedia.org/wiki/Mean_shift`

[2] DBSCAN Algorithm
https://www.geeksforgeeks.org/dbscan-clustering-in-ml-density-based-clustering/

[3] KMeans Algorithm
https://stanford.edu/~cpiech/cs221/handouts/kmeans.html

[4] Agglomerative Clustering
https://scikit-learn.org/stable/modules/clustering.html#
hierarchical-clustering

[5] Gaussian Mixture Clustering Algorithm
https://brilliant.org/wiki/gaussian-mixture-model/