Inteligencia de negocio(2018-2019)

Grado en Ingeniería Informática

Universidad de Granada

Memoria de la práctica 2

David Peinado Perea

Grupo de prácticas 1

davidpeinado@correo.ugr.es

Índice

1 Introducción

En esta práctica se utilizarán técnicas de aprendizaje no supervisado para analizar un conjunto de datos y extraer información sobre el mismo. Dicho conjunto de datos será el proporcionado por el INE((http://www.ine.es/censos2011_datos/ cen11_datos_microdatos.htm).
El conjunto, procesado a partir de la fuente original, se compone de 142 variables sobre sexo, edad, nacionalidad, estudios, situación laboral, migraciones y movilidad, situación familiar, etc.
Estos datos son relativos a la provincia de Granada, un total de 83.499 casos.

El objetivo es coger a unos grupos determinados de personas(mujeres de una edad determinada, extranjeros, etc.) y, aplicando algoritmos de clustering, analizar los resultados obtenidos y interpretarlos para explicar distintos perfiles o grupos encontrados.

En este caso nos centraremos en:

- Mujeres entre 20 y 50 años
- Nacidos en el extranjero
- Personas pertenecientes a familias numerosas

Para el análisis de los resultados se usarán diversos gráficos(dendrogramas, mapas de calor...), así como las métricas de rendimiento tales como Silhouette y el índice Calinski-Harabaz.

Los algoritmos utilizados para cada caso de uso serán:

- K-Means
- MiniBatchK-Means
- MeanShift
- AffinityPropagation
- DBSCAN
- Ward (clustering aglomerativo)

2 Casos de estudio

2.1. Caso de estudio 1

2.1.1. Introducción

En este caso de uso nos centraremos en las mujeres de entre 20 y 50 años de edad.
Un grupo de gran importancia, como se podrá comprobar por el número de instancias que contiene, que representa un alto porcentaje del total.
Reduciremos el número de instancias de 17996 obtenidas a 2500, para poder obtener los resultados en un período de tiempo bastante menor(algoritmos como MeanShift o AffinityPropagation tardarían varios minutos en ejecutarse, incluso horas, con tal cantidad de instancias).

En este caso analizaremos las variables correspondientes al número de hijos de la mujer, estado civil, número de generaciones en el hogar y número de miembros en el hogar.

```
censo = pd.read_csv('censo_granada.csv')
censo = censo.replace(np.NaN,0) #los valores en blanco realmente son otra categoría que nombramos como 0

#       MUJERES ENTRE 20 Y 50 AÑOS(AMBOS INCLUIDOS)
subset = censo.loc[censo['SEXO']==6]
subset = subset.loc[censo['EDAD']>=20]
subset = subset.loc[censo['EDAD']<=50]
#seleccionar variables de interés para clustering
usadas = ['NHIJOS', 'ECIVIL', 'NGENER', 'NMIEM']
X = subset[usadas]
print(X.shape) # numero de instancias antes del sample
#selecciona 1000 instancias del caso de uso
X = X.sample(2500, random_state=123456)
#normaliza
X_normal = X.apply(norm_to_zero_one)
```

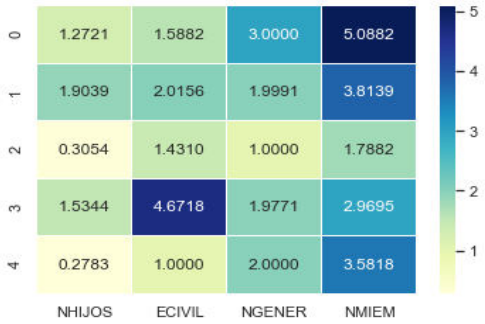
2.1.2. Parámetros de los algoritmos

Hemos seleccionado un k=5 para los algoritmos K-Means y MiniBatchK-Means. En el algoritmo DBSCAN ajustaremos la densidad mediante el párametro eps= 0.1. Para el clustering aglomerativo “ward” usaremos un k=6.

2.1.3. Resultados obtenidos

```
K-means             : k:   5,  0.02 segundos, CH index:  2847.789, SC: 0.59058
MiniBatchKMeans     : k:   5,  0.01 segundos, CH index:  2847.051, SC: 0.59058
MeanShift           : k:   8,  2.70 segundos, CH index:  1070.737, SC: 0.61653
AffinityPropagation: k: 997, 26.87 segundos, E:\Anaconda\lib\site-packages\sklearn\metrics\cluster
\unsupervised.py:205: RuntimeWarning: invalid value encountered in true_divide
  sil_samples /= np.maximum(intra_clust_dists, inter_clust_dists)
CH index:   124.376, SC: nan
DBSCAN              : k:  37,  0.08 segundos, CH index:   643.795, SC: 0.64940
Ward                : k:   6,  0.15 segundos, CH index:  2498.559, SC: 0.59112
```

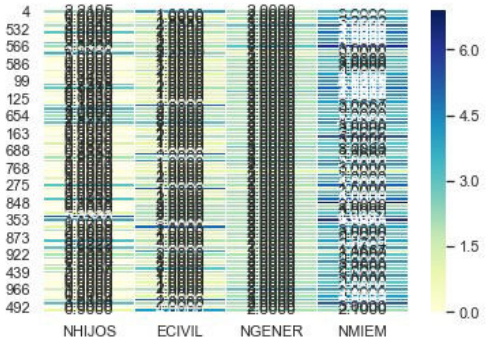
- K-Means



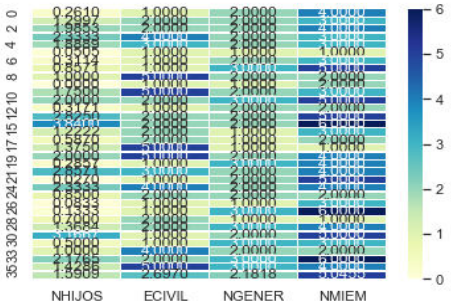
- MiniBatchK-Means

- MeanShift

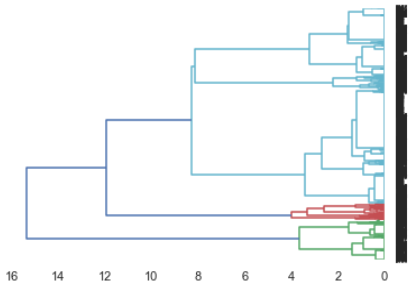
- AffinityPropagation

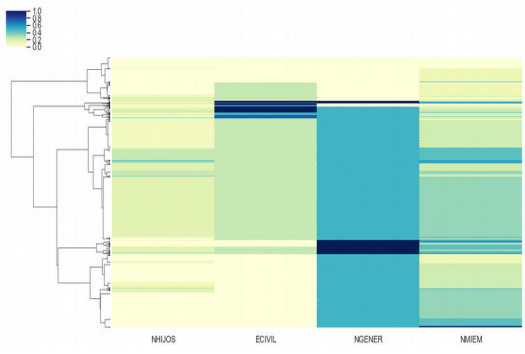


- DBSCAN



- Ward (clustering aglomerativo)





2.1.3.1 K-Means

Hemos escogido diversos valores de k para este algoritmo, se puede ver que el pico de rendimiento se alcanza en k=6, lo que indica la realidad existente, este grupo tiene una segmentación heterogénea, el algoritmo actúa mejor con un k=6 que con un k=3 por esto mismo.

```
K-means          : k:   3,  0.02 segundos, CH index:  1876.510, SC: 0.58796
K-means          : k:   4,  0.02 segundos, CH index:  2144.704, SC: 0.52679
K-means          : k:   6,  0.02 segundos, CH index:  2690.177, SC: 0.59149
```

2.1.3.2 DBSCAN

Al aumentar el parámetro eps(0.1, 0.2 y 0.3 respectivamente), se observa como el número de clusters decrece, pues la densidad disminuye, así pues, el SC disminuye lógicamente cuanto menor es la densidad, ya que esta medida muestra como de similar es un objeto a su propio cluster, haciendo los clusters más grandes se agrupan objetos que no definen tan bien a ese cluster, pues hay más diversidad intracluster.

```
DBSCAN               : k:  37,  0.07 segundos, CH index:   643.795, SC: 0.64940
DBSCAN               : k:  14,  0.10 segundos, CH index:  1474.797, SC: 0.59729
DBSCAN               : k:   4,  0.13 segundos, CH index:   877.858, SC: 0.57259
```

2.1.4. Interpretación de la segmentación

Como es lógico, un grupo tan grande como es este, tiene una segmentación heterogénea, aunque los algoritmos no se ajusten totalmente al problema se pueden identificar patrones claros como el caso de las mujeres viviendo solas o con su pareja únicamente(cluster 2 en K-Means, cluster 1 en MeanShift), también se puede apreciar el caso de los hogares donde conviven abuelos, padres y nietos (cluster 0 en K-Means, cluster 5 en MeanShift). También se puede observar que en pocos clusters, el centroide del número de hijos sea superior a 2, llevándonos a la conclusión de que el número de mujeres con 3 hijos o más es muy reducido.

2.2. Caso de estudio 2

2.2.1. Introducción

En este caso de uso nos centraremos en los nacidos en el extranjero, seleccionando las instancias en las que la provincia de nacimiento de la persona no pertenece al territorio español.
Un grupo que siempre suele estar en el punto de mira, en el caso de extranjeros, subconjunto de este grupo, sobre el que rondan muchos prejuicios y del que los medios de comunicación no suelen hablar en profundidad, realizando estadísticas sobre ellos y analizando los resultados.
Se podrán analizar grupos como el de nacidos en el extranjero no nacionalizados, o los ya nacionalizados.

En este caso obtendremos 3740 instancias, no realizaremos sampling ya que este número no es lo suficientemente grande como para dar problemas con el tiempo de la ejecución del programa.

Las variables que se tendrán en cuenta serán las correspondientes a: edad, estado civil, código del país de nacimiento, nacionalidad y el año de llegada a España.

```
censo = pd.read_csv('censo_granada.csv')
censo = censo.replace(np.NaN,0) #los valores en blanco realmente son otra categoría que nombramos como 0

#       NACIDOS EN EL EXTRANJERO
subset = censo.loc[censo['CPRON']==0]

#seleccionar variables de interés para clustering
usadas = ['EDAD', 'ECIVIL', 'CPAISN', 'NACI', 'ANOE']
X = subset[usadas]
print(X.shape)
#selecciona 2500 instancias del caso de uso
#X = X.sample(2500, random_state=123456)
#normaliza
X_normal = X.apply(norm_to_zero_one)
```

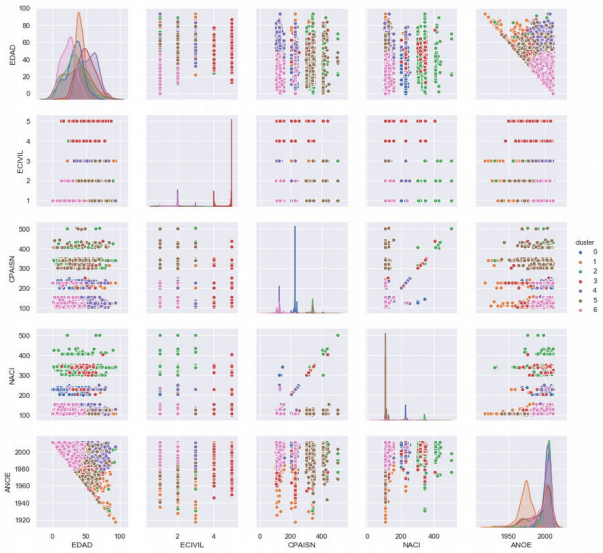
2.2.2. Parámetros de los algoritmos

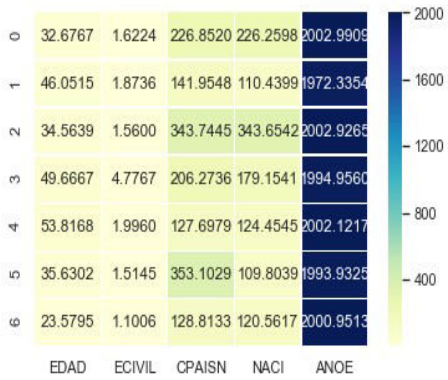
Hemos seleccionado un k=7 para los algoritmos K-Means y MiniBatchK-Means.
En el algoritmo DBSCAN ajustaremos la densidad mediante el párametro eps= 0.1.
Para el clustering aglomerativo "ward" usaremos un k=5.

2.2.3. Resultados obtenidos

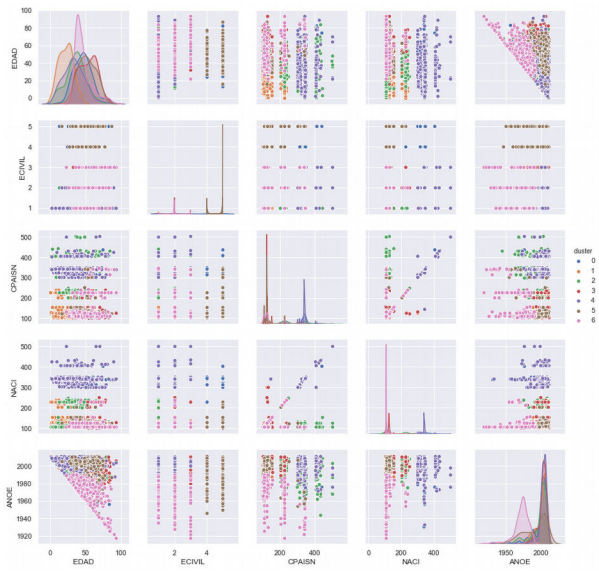
```
K-means             : k:   7,  0.06 segundos, CH index:  1904.243, SC: 0.36427
MiniBatchKMeans     : k:   7,  0.02 segundos, CH index:  1890.669, SC: 0.36896
MeanShift           : k:   4,  6.59 segundos, CH index:  1612.618, SC: 0.36113
AffinityPropagation: k:  78, 35.13 segundos, CH index:  1388.031, SC: 0.34153
DBSCAN              : k:  27,  0.09 segundos, CH index:   497.129, SC: 0.25870
Ward                : k:  10,  0.41 segundos, CH index:  1669.452, SC: 0.35429
```

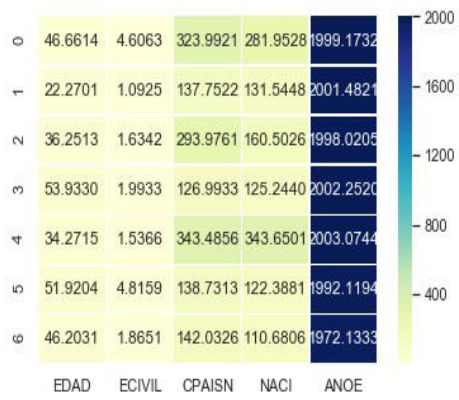
- K-Means



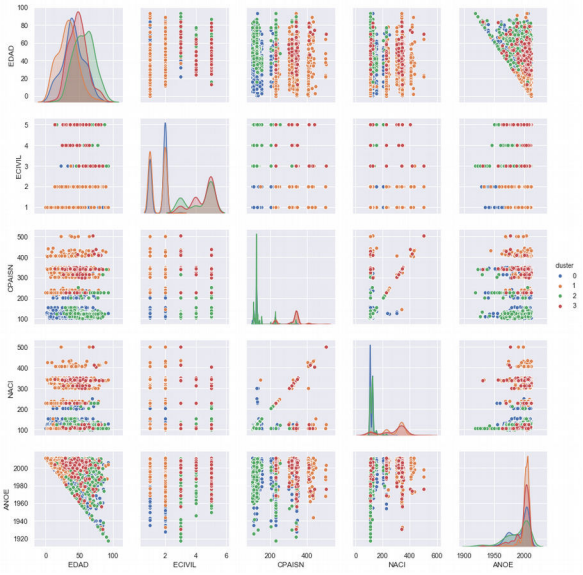


- MiniBatchK-Means

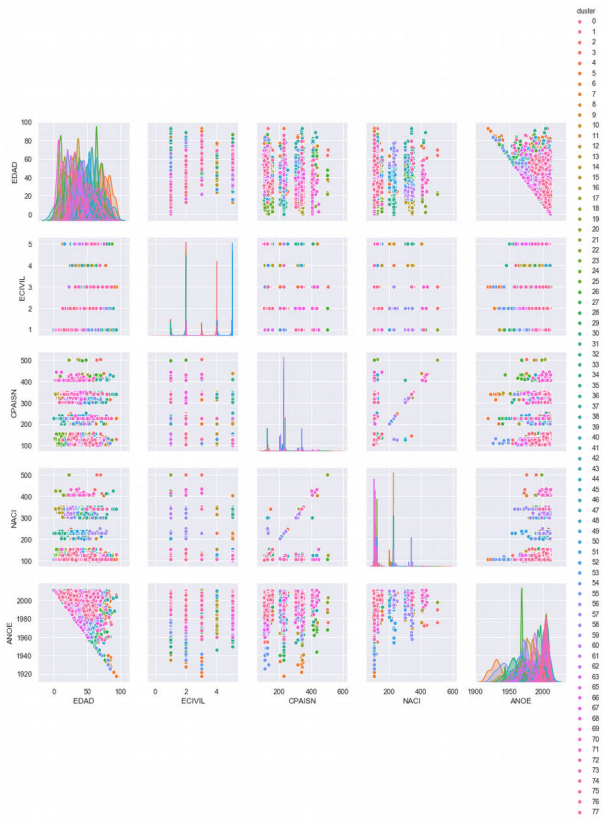


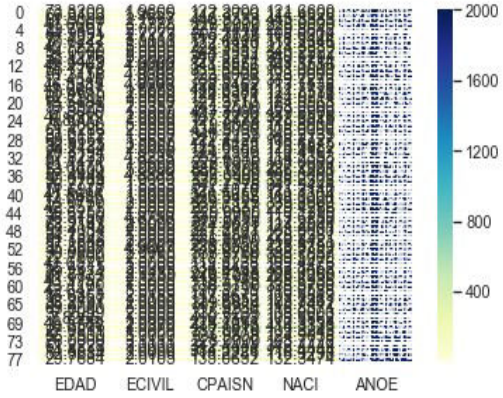


- MeanShift

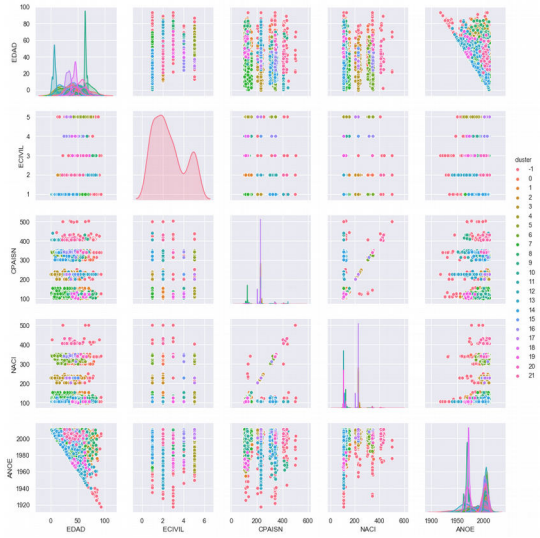


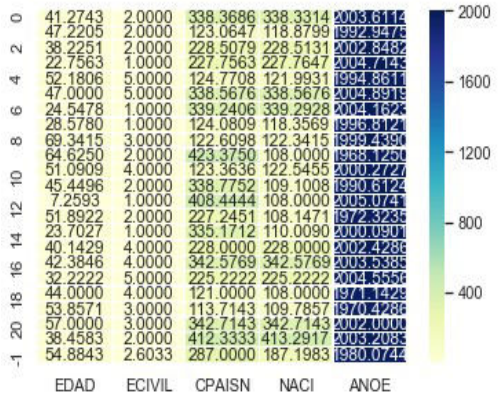
- AffinityPropagation



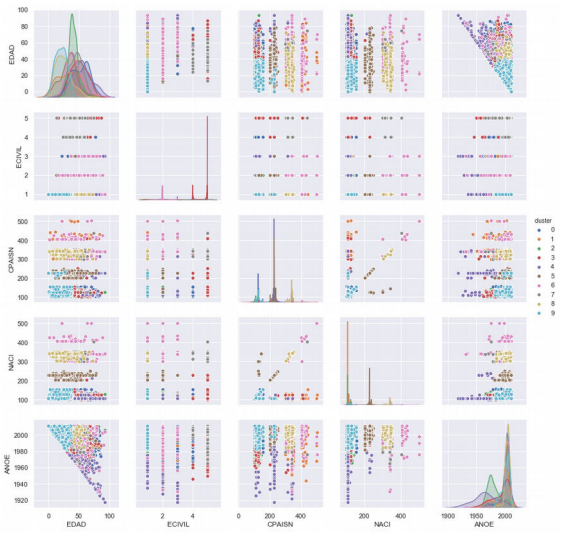


- DBSCAN

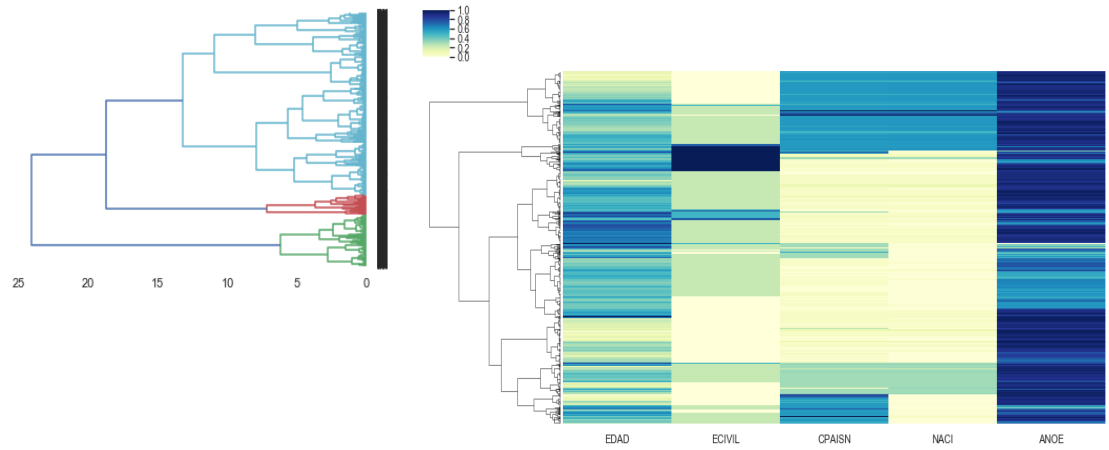




- Ward (clustering aglomerativo)



	EDAD	ECIVIL	CPAISN	NACI	ANOE
0	52.5059	2.1825	126.0795	123.6597	2000.5841
1	35.3909	1.5244	353.2248	109.8274	1994.3909
2	42.1017	1.6455	119.9730	111.5262	1981.0509
3	50.8762	4.9333	172.8333	132.4524	1993.4714
4	55.6646	1.9207	229.9329	114.5061	1967.5305
5	33.5301	1.6325	225.8012	228.7771	2003.1054
6	42.5508	1.9976	347.5296	347.3215	2001.9882
7	45.1923	4.6538	339.8846	339.4359	2000.6795
8	24.9023	1.0000	339.1063	339.1580	2004.0489
9	20.7946	1.0000	127.9232	119.9419	2001.5809



2.2.3.1 K-Means

Hemos escogido diversos valores de k para este algoritmo, se puede ver que el mejor rendimiento que se alcanza en k=4.

```
K-means                : k:   4,  0.04 segundos, CH index:  1981.246, SC: 0.31368
K-means                : k:   5,  0.05 segundos, CH index:  1976.471, SC: 0.32926
K-means                : k:   7,  0.06 segundos, CH index:  1904.243, SC: 0.36427
```

2.2.3.2 MiniBatchK-Means

Hemos escogido los mismos valores de k para este algoritmo que para K-Means, se puede ver que el mejor rendimiento que se alcanza en k=7

```
MiniBatchKMeans        : k:   4,  0.01 segundos, CH index:  1609.447, SC: 0.37697
MiniBatchKMeans        : k:   5,  0.01 segundos, CH index:  1882.242, SC: 0.34378
MiniBatchKMeans        : k:   7,  0.02 segundos, CH index:  1890.669, SC: 0.36896
```

Como era de esperar en este caso, MiniBatchK-Means se ha comportado peor que K-means si tomamos en cuenta el índice CH, esto es debido a que este algoritmo es una variante de K-Means que usa “mini-lotes” con el objetivo de reducir el tiempo de computación, no obstante dando ligeramente peores resultados en esta métrica. Se puede ver que esta variante, nos hace clusteres en los que el objeto es más similar a su propio cluster que en K-Means ya que la métrica Silhouette es superior para todo k analizado.

2.2.4. Interpretación de la segmentación

Podemos ver que tanto K-Means como MiniBatchK-Means hacen una distinción por estado civil a los separados y divorciados, creando un cluster que los contenga a la casi totalidad de ellos. El resto de los algoritmos no hacen esta distinción.

También se puede observar el grupo que distingen tanto K-Means y MiniBatchK-Means como DBSCAN y Ward, que es el de llegados a España alrededor de los 70, mientras otros algoritmos se centran en años alrededor del 2000.

En cuanto a la nacionalidad, se puede ver que la gran mayoría de nacidos fuera de España goza de nacionalidad española, mientras que variando según el algoritmo, se aprecian diferencias en las nacionalidades mayoritarias que no son la española.

No se aprecia una clara diferenciación intercluster entre diversas rangos de edades, si bien se tratan distintamente según el algoritmo, no se aprecia un claro patrón en ninguno de ellos.

2.3. Caso de estudio 3

2.3.1. Introducción

En este caso de uso nos centraremos en personas que tienen familias numerosas.
Este grupo lo he seleccionado ya que yo pertenezco a una, y tenía interés por saber que información podría obtener.
Reduciremos el número de instancias de 4993 obtenidas a 2500, para poder obtener los resultados en un período de tiempo bastante menor(AffinityPropagation tarda varios minutos en ejecutarse, y al tener que ejecutar múltiples veces el programa supondría un problema).

En este caso analizaremos las variables correspondientes al número de hijos, régimen de tenencia del hogar, superficie útil del hogar y número de habitaciones en el hogar.

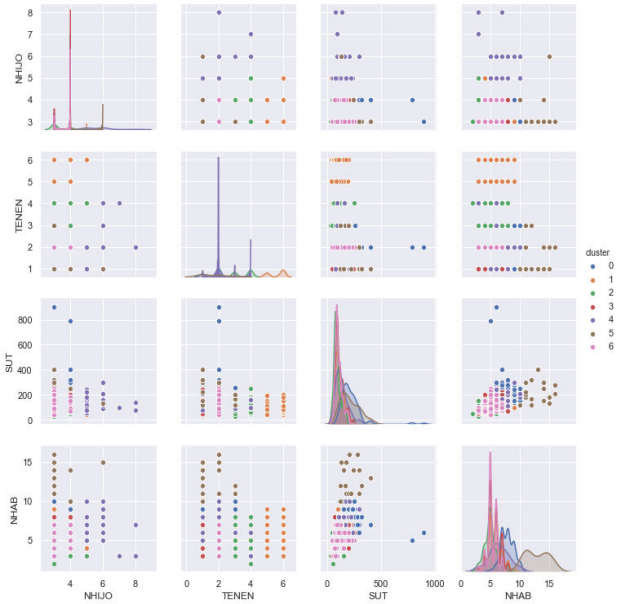
```
censo = pd.read_csv('censo_granada.csv')
censo = censo.replace(np.NaN,0) #los valores en blanco

#       FAMILIAS NUMEROSAS
subset = censo.loc[censo['FAMNUM']==2]

#seleccionar variables de interés para clustering
usadas = ['NHIJO', 'TENEN', 'SUT', 'NHAB']
X = subset[usadas]
print(X.shape)
#selecciona 1000 instancias del caso de uso
X = X.sample(2500, random_state=123456)
```

2.3.2. Parámetros de los algoritmos

Hemos seleccionado un k=7 para los algoritmos K-Means y k= 5 para MiniBatchK-Means.
En el algoritmo DBSCAN ajustaremos la densidad mediante el párametro eps= 0.3.
Para el clustering aglomerativo “ward” usaremos un k=4.

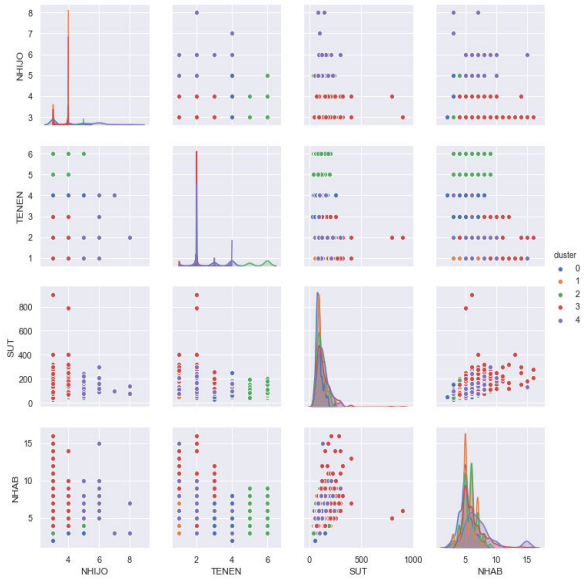
2.3.3. Resultados obtenidos

- K-Means



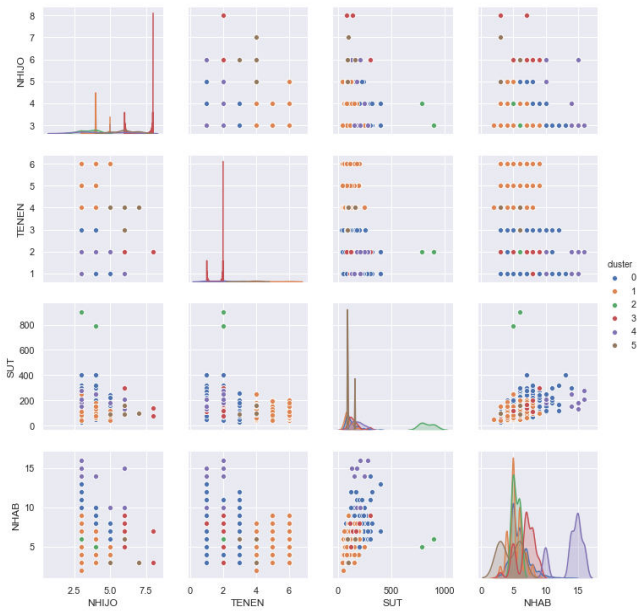
	NHIJO	TENEN	SUT	NHAB
0	3.1342	1.7668	222.9137	7.9073
1	3.1905	5.6349	113.2024	5.7540
2	3.3238	3.5960	94.7163	5.0888
3	3.1686	1.0000	114.2455	5.6530
4	5.8101	2.1139	136.9494	6.5063
5	3.4464	1.6964	215.3750	12.9107
6	3.1131	2.0000	108.2988	5.2845

- MiniBatchK-Means

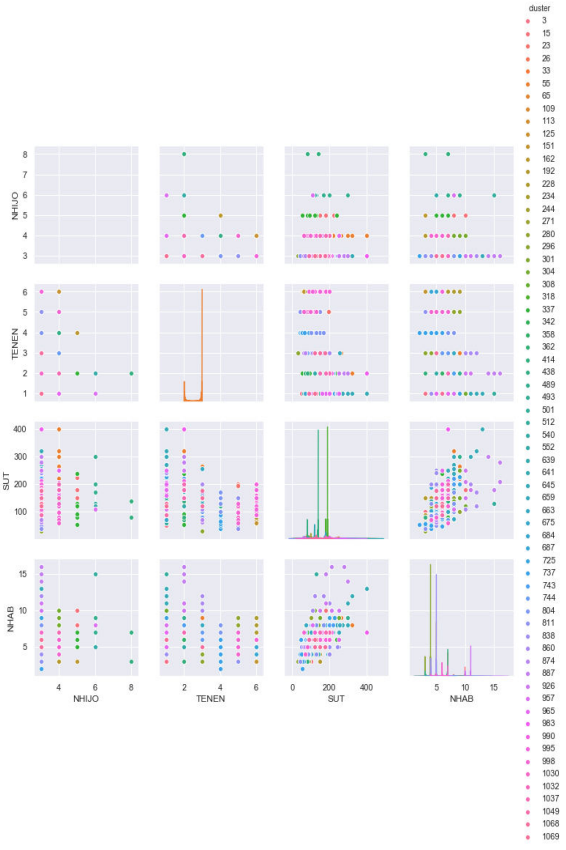


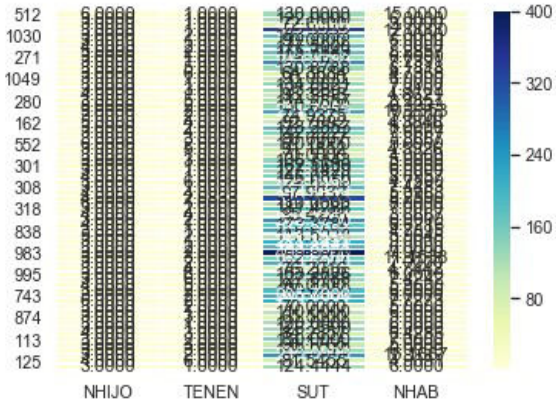
	NHIJO	TENEN	SUT	NHAB
0	3.3238	3.5960	94.7163	5.0888
1	3.1602	1.0389	113.8212	5.5754
2	3.1905	5.6349	113.2024	5.7540
3	3.1230	1.9342	144.0145	6.3330
4	5.8235	2.0353	136.4588	7.1059

- MeanShift

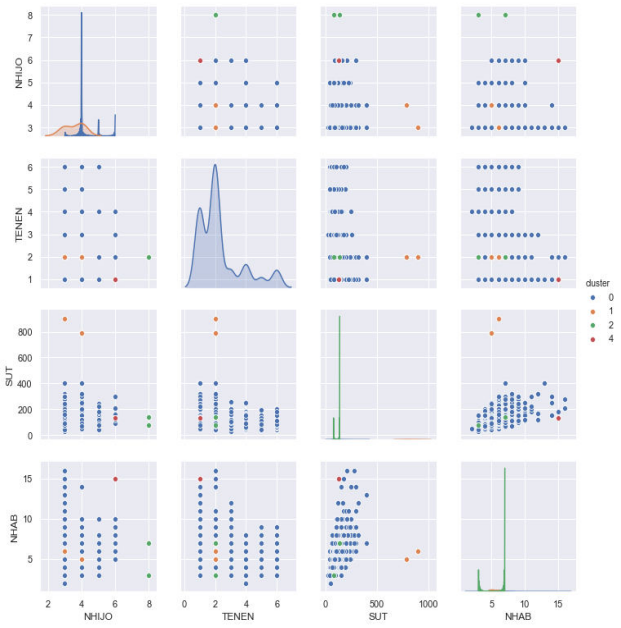


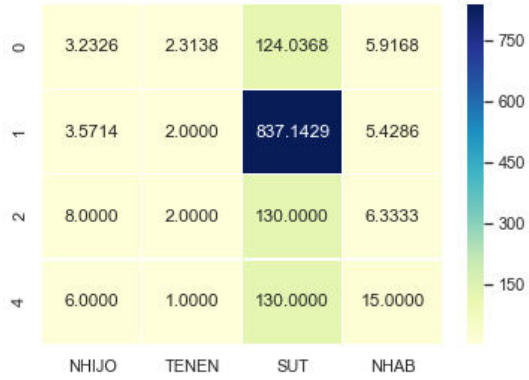
- AffinityPropagation



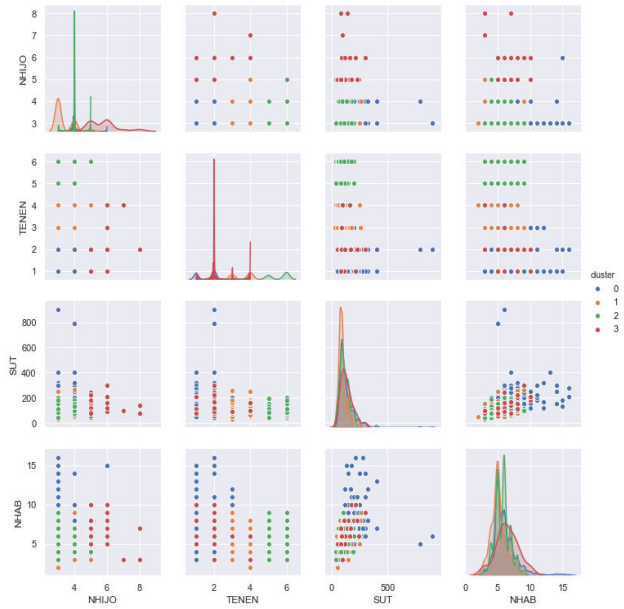


- DBSCAN





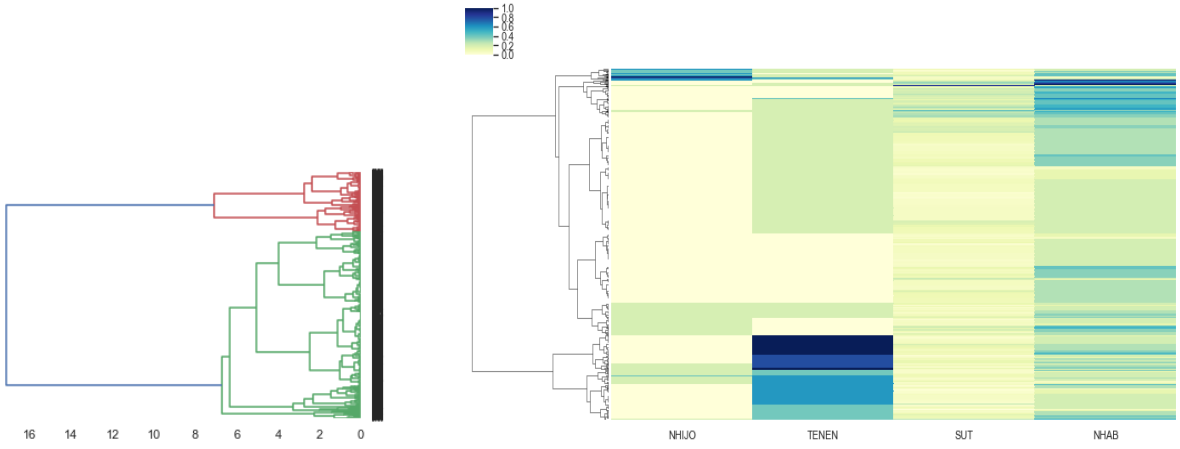
- Ward (clustering aglomerativo)



2.3.3.1 K-Means

```
K-means          : k:   3,  0.02 segundos, CH index:  2004.692, SC: 0.38132
K-means          : k:   5,  0.02 segundos, CH index:  1765.909, SC: 0.40390
K-means          : k:   7,  0.03 segundos, CH index:  1851.169, SC: 0.45514
```

CH y SC muestran una correlación negativa en este caso.



Cuanto menor es el número de clusters, un objeto es menos similar a su propio cluster, mientras que el ratio de dispersión intra-cluster / inter-cluster aumenta. Cuanto mayor es el número de clusters, un objeto es más similar a su propio cluster y el ratio de dispersión intra-cluster / inter-cluster disminuye.

2.3.3.2 Ward

```
Ward                : k:   4,  0.13 segundos, CH index:  1687.275, SC: 0.48545
Ward                : k:   6,  0.13 segundos, CH index:  1795.486, SC: 0.39675
Ward                : k:   9,  0.14 segundos, CH index:  1646.742, SC: 0.45418
```

En este caso no se aprecia una tendencia creciente o decreciente hacia k mayores o menores.

2.3.4. Interpretación de la segmentación

Se puede distinguir un subgrupo, el cluster 4 de K-Means aglomera las familias numerosas que más hijos tienen, de 5 en adelante.
DBSCAN, al ser el parámetro “eps” algo elevado, sólo genera 5 clusters, siendo la mayor parte de instancias de uno sólo. Otro cluster aglomera los hogares con superficies útiles más grandes. Otro se centra en los hogares que tienen un número de habitaciones elevado.
MiniBatchK-Means hace una distinción con el grupo de familias que han obtenido su hogar cedida gratis, a bajo precio, o de otra forma no estipulada en los criterios.

3 Contenido adicional

4 Bibliografía

https://scikit-learn.org/stable/modules/clustering.html
https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html
https://en.wikipedia.org/wiki/Silhouette_(clustering)
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabaz_score.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html#sklearn.metrics.silhouette_score