Universidad de Granada

Ingeniería Informática

Practica 2: Segmentación para Análisis Empresarial.

Autor: José Antonio Ruiz Millán
email: jantonioruiz@correo.ugr.es
Curso: 2018-2019
Asignatura: Inteligencia de Negocio
1 de diciembre de 2018

Índice

1. Introduccion

En este problema se analizarán perfiles de la población granadina.

A partir de los microdatos publicados en el último censo de población realizado por el Instituto Nacional de Estadística (INE) en 2011 (`http://www.ine.es/censos2011_datos/cen11_datos_microdatos.htm`). El conjunto de datos se compone de 142 variables sobre sexo, edad, nacionalidad, estudios, situación laboral, migraciones y movilidad, situación familiar, etc. Trabajaremos con los datos relativos a la provincia de Granada, un total de 83.499 casos.

Algunas variables son categóricas como, por ejemplo, estado civil (soltero, casado, divorciado, viudo...). Estas variables no sirven para aplicar un análisis de agrupamiento, pero sí son útiles para fijar casos de estudio donde centrar el análisis. Hay otras variables numéricas como, por ejemplo, tamaño del núcleo (2, 3, 4, 5, 6 o más) o edad que sí se pueden usar para clustering. Finalmente, hay también variables que, aunque no son numéricas, sí son ordinales (por ejemplo, nivel de estudios) y, por tanto, también se pueden usar para clustering.

El objetivo de la práctica es definir algunos casos de estudio de interés (fijando condiciones en algunas variables), aplicar distintos algoritmos de clustering, analizar la calidad de las soluciones obtenidas y, finalmente, interpretar los resultados para explicar los distintos perfiles o grupos encontrados.

2. Caso de estudio 1

2.1. Conjunto de datos

Para este caso de estudio he analizado la población granadina mayor de 17 años, que es estudiante y de los cuales tenemos información sobre los estudios de los padres.

Este subconjunto tiene un enfoque sobre el nivel de estudios en la ramificación familiar, es decir, comprobar si por lo general los hijos suelen seguir el camino de los padres a la hora de estar estudiando, no obstante, estos datos son sobre estudios que están realizando actualmente por lo que decidí acotar a mayores de edad.

He decidido crear los clusters utilizando las variables *"ESCUR1" , "ESTUPAD" , "ESTUMAD" , "TENEN" , "EDAD"* que identifican los estudios actuales, los estudios del padre, los estudios de la madre, en que tipo de vivienda viven y la edad, respectivamente. Con este conjunto de datos obtenemos 3760 elementos.

2.2. Resultados algoritmos

En primer lugar comentar que he hecho dos ejecuciones, una ejecución de todos los algoritmos con los parámetros por defecto para cada uno de ellos, y una segunda ejecucion con los algoritmos *KMeans* y *AgglomerativeClustering* modificando sus parámetros.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Kmeans	0,19	1052,49	0,25	8
AgglomerativeClustering	0,52	934,44	0,43	2
DBSCAN	0,15	94,38	-0,042	27
Birch	0,09	824,07	0,36	3
SpectralClustering	2,05	1212,4	0,23	4

Tabla 1: Tabla comparativa parametros por defecto caso de estudio 1

Como apreciamos en la tabla, la columna izquierda nos muestra el nombre de los algoritmos que he decidido utilizar y en las demás columnas podemos ver el resultado de cada una de las métricas medidas como el tiempo que ha tardado y el número de clusters

2.2.1. Kmeans:

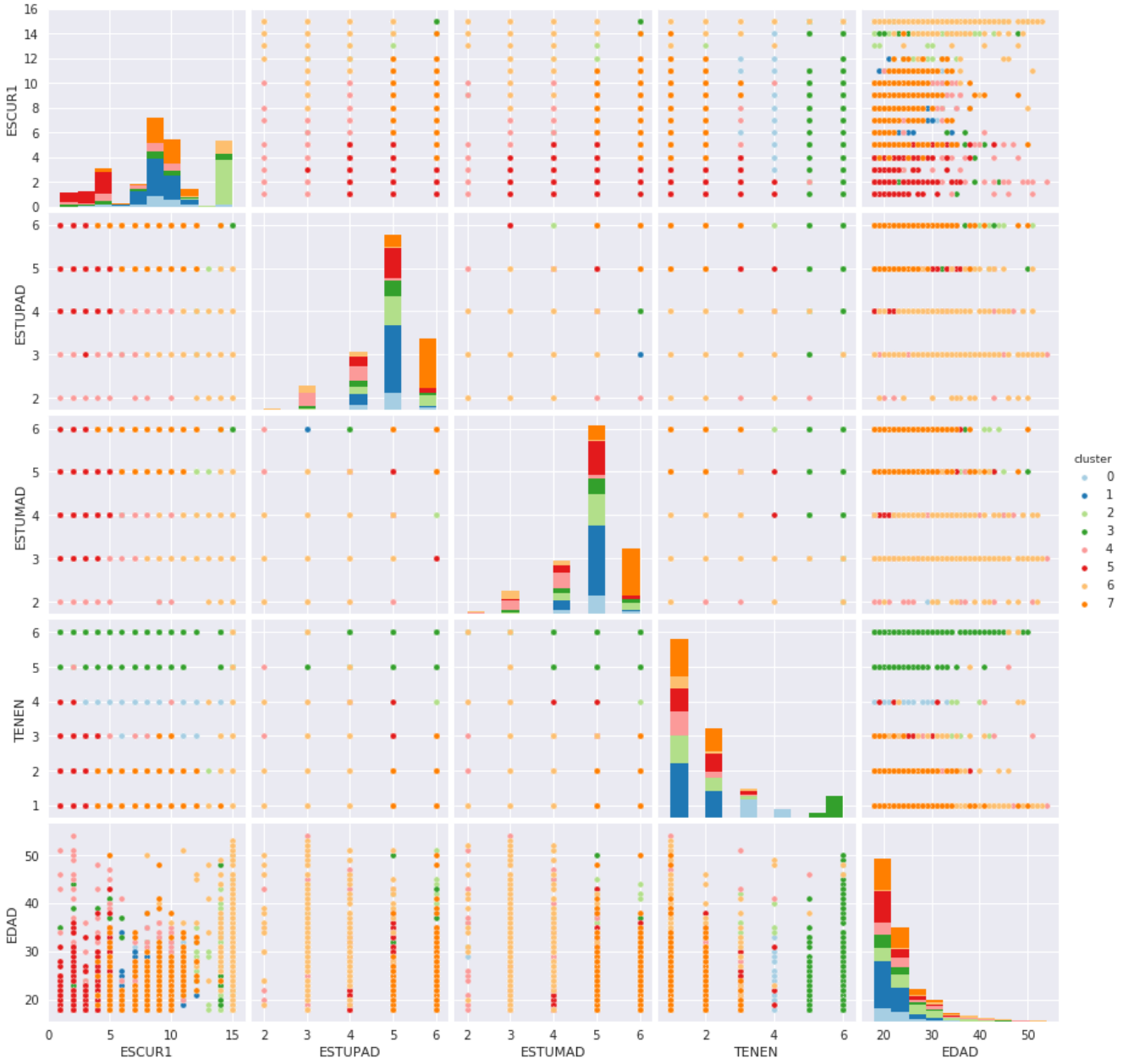
Para este algoritmo concretamente he decidido utilizar tanto los parámetros por defecto como una modificación de ellos, obteniendo:

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Kmeans-8	0,19	1052,49	0,25	8
Kmeans-3	0,08	1210,37	0,29	3

Tabla 2: Tabla comparativa parametros KMeans caso de estudio 1

Podemos ver como hemos obtenido una mejor solución modificando estos parámetros y poniendo que nos agrupe en un número menor de clusters que el caso por defecto que tiene Kmeans.

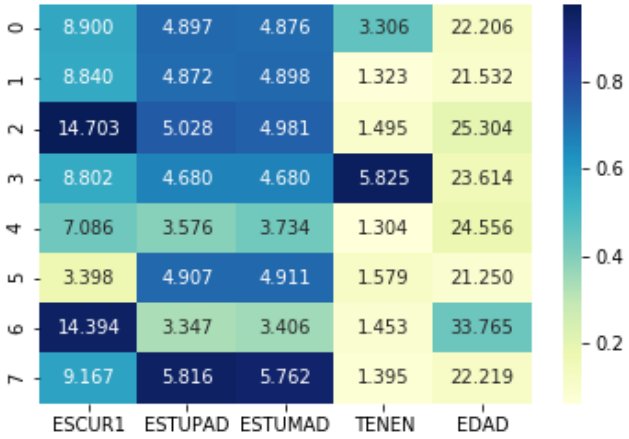
Pasamos ahora a observar el resultado de **Kmeans-8**:



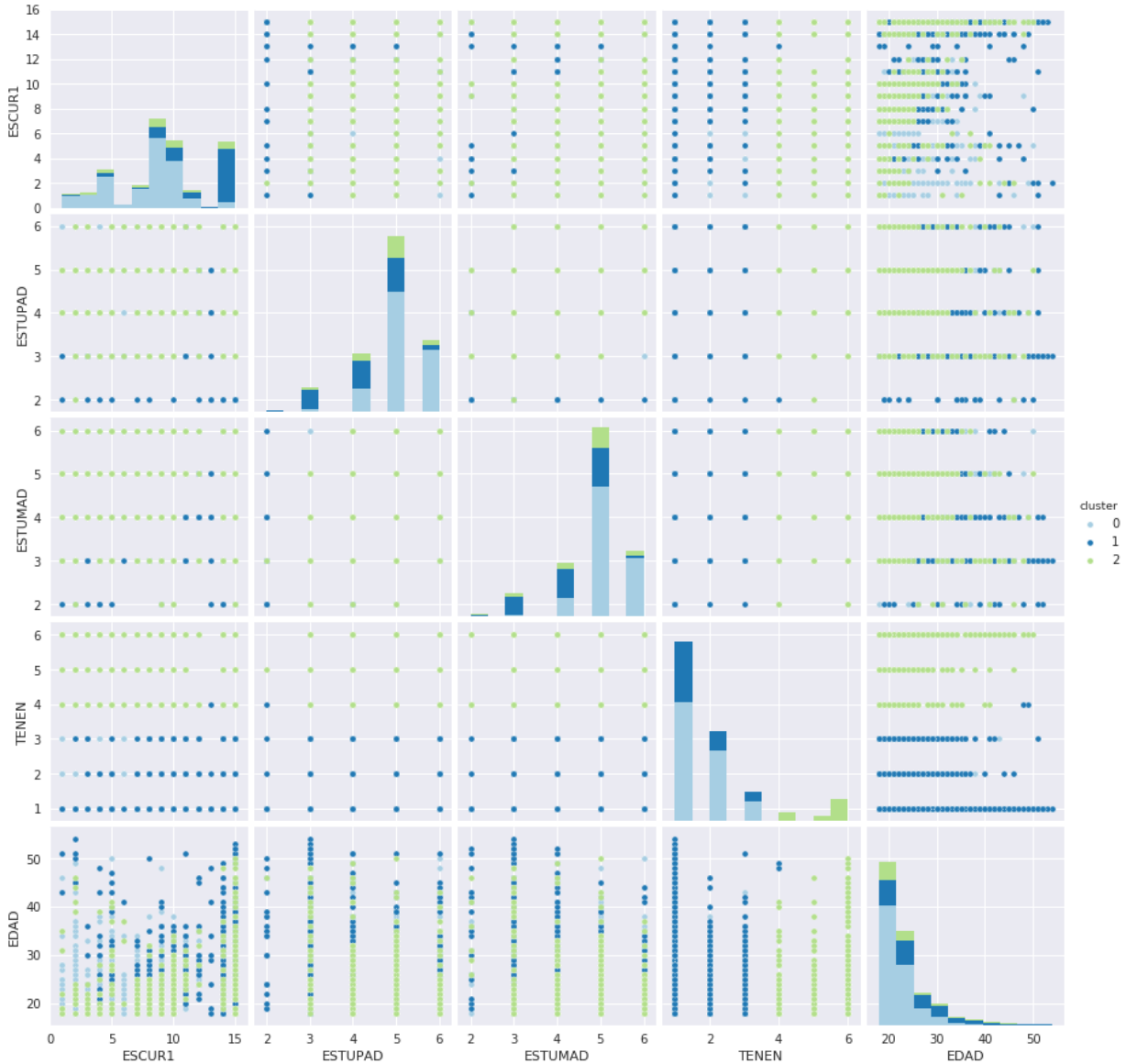
	Elementos	%
0	291	7,74
1	909	24,18
2	529	14,07
3	303	8,06
4	349	9,28
5	515	13,7
6	170	4,52
7	694	18,46

Tabla 3: Tabla comparativa clusters KMeans-8 caso de estudio 1

Con esto, podemos ver tambien el siguiente gráfico que nos da información sobre los centroides.

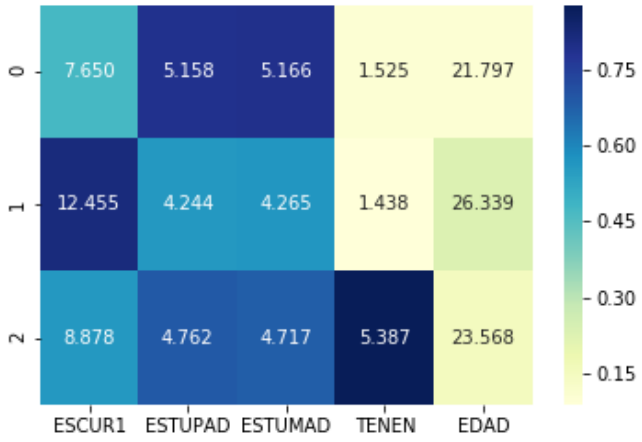


Por otra parte, analizaremos los mismos gráficos para el segundo caso, el caso de **KMeans-3**:



	Elementos	%
0	2349	62,47
1	1008	26,81
2	403	10,72

Tabla 4: Tabla comparativa clusters KMeans-3 caso de estudio 1



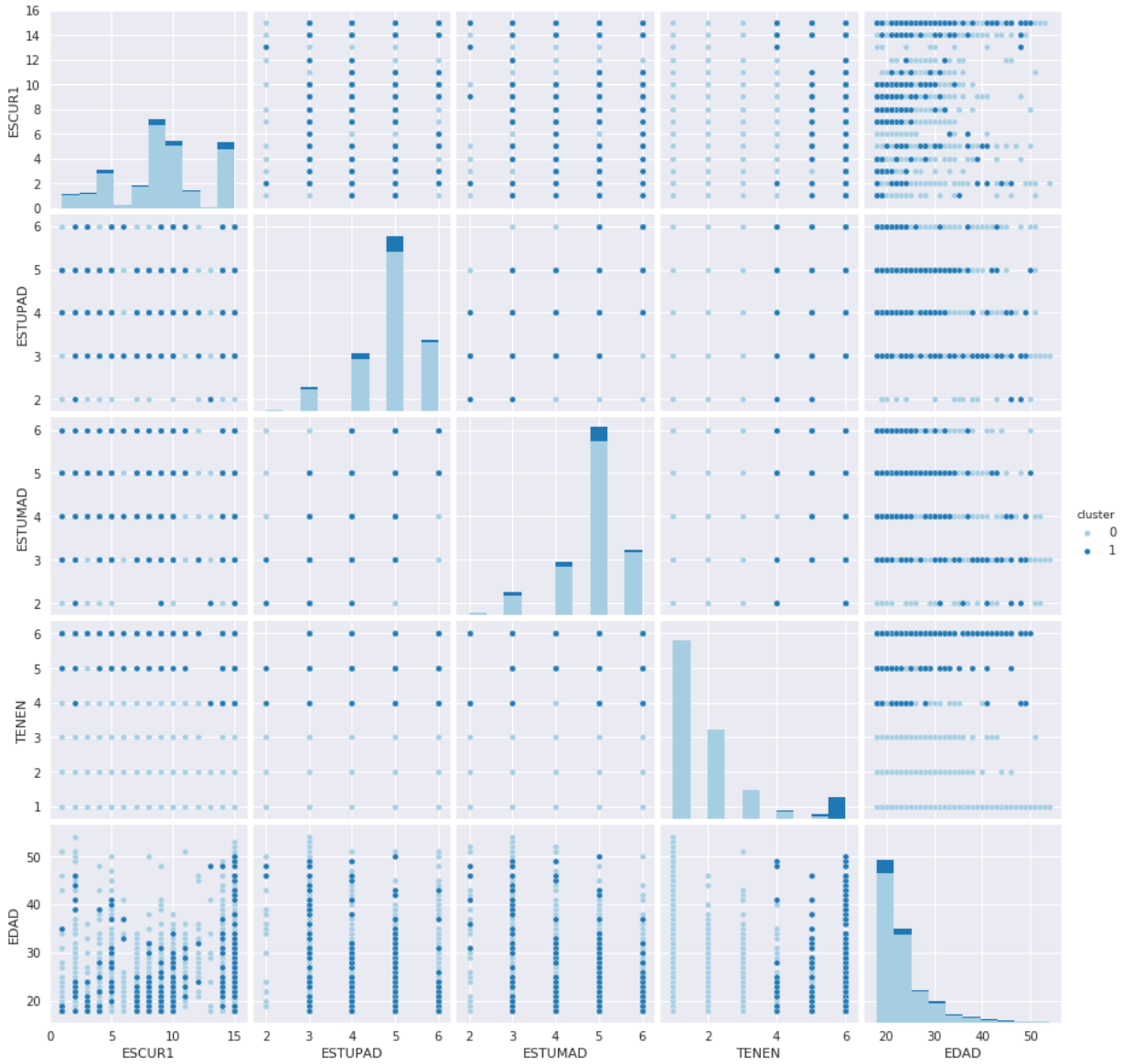
2.2.2. AgglomerativeClustering

Como en el caso anterior, he decidido hacer una modificación de los parámetros de este algoritmo, utilizando en número de clusters.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
AgglomerativeClustering-2	0,52	934,44	0,43	2
AgglomerativeClustering-12	0,52	771,31	0,22	12

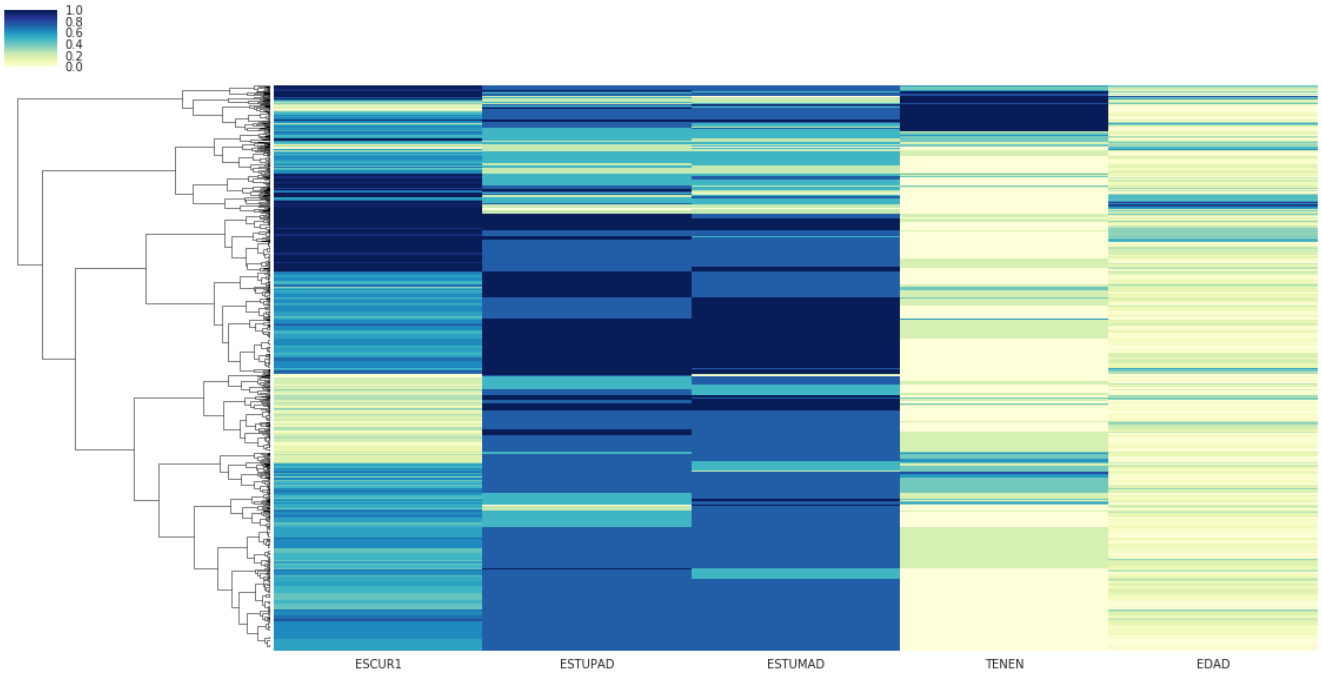
Tabla 5: Tabla comparativa parametros AgglomerativeClustering caso de estudio 1

Pasamos ahora a mostrar los resultados para **AgglomerativeClustering-2**:

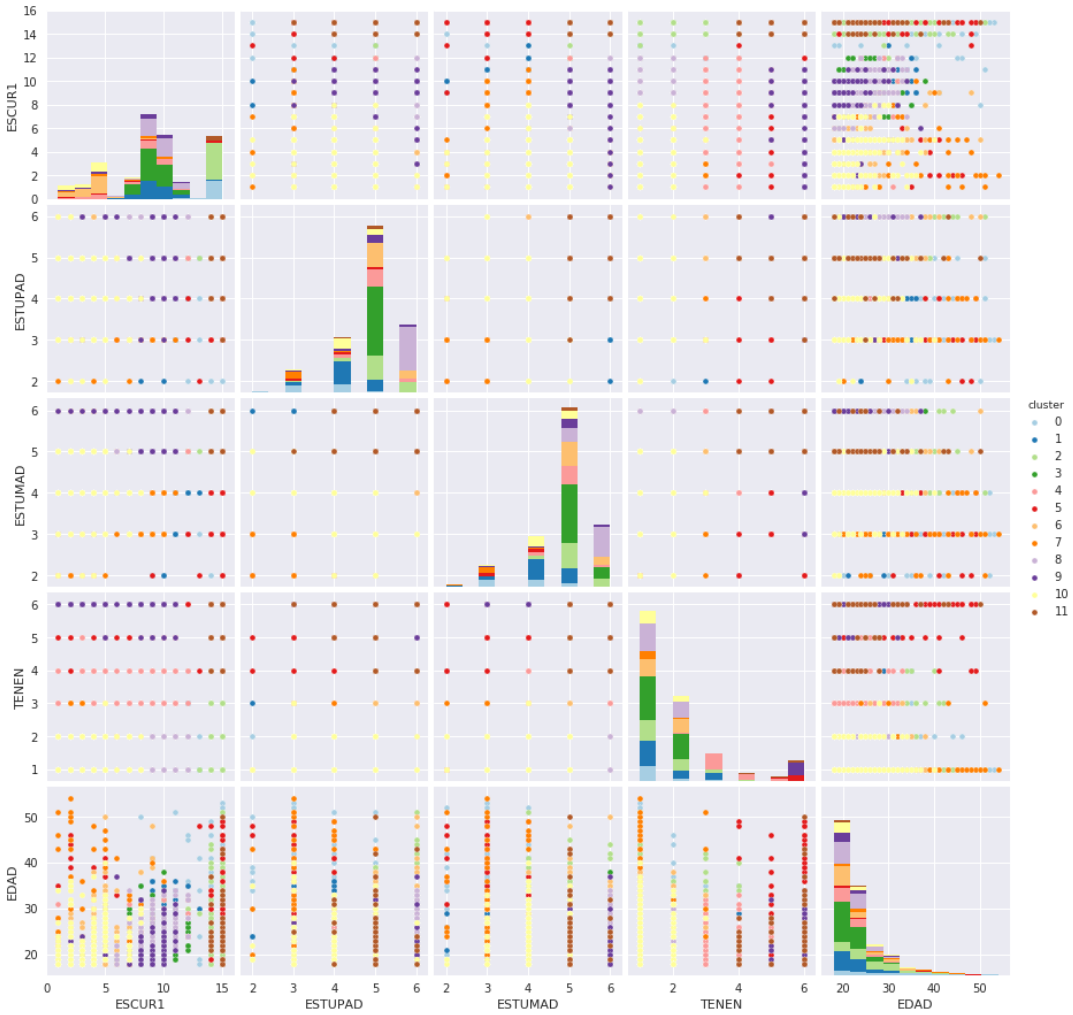


	Elementos	%
0	3464	92,13
1	296	7,87

Tabla 6: Tabla comparativa clusters AgglomerativeClustering-2 caso de estudio 1

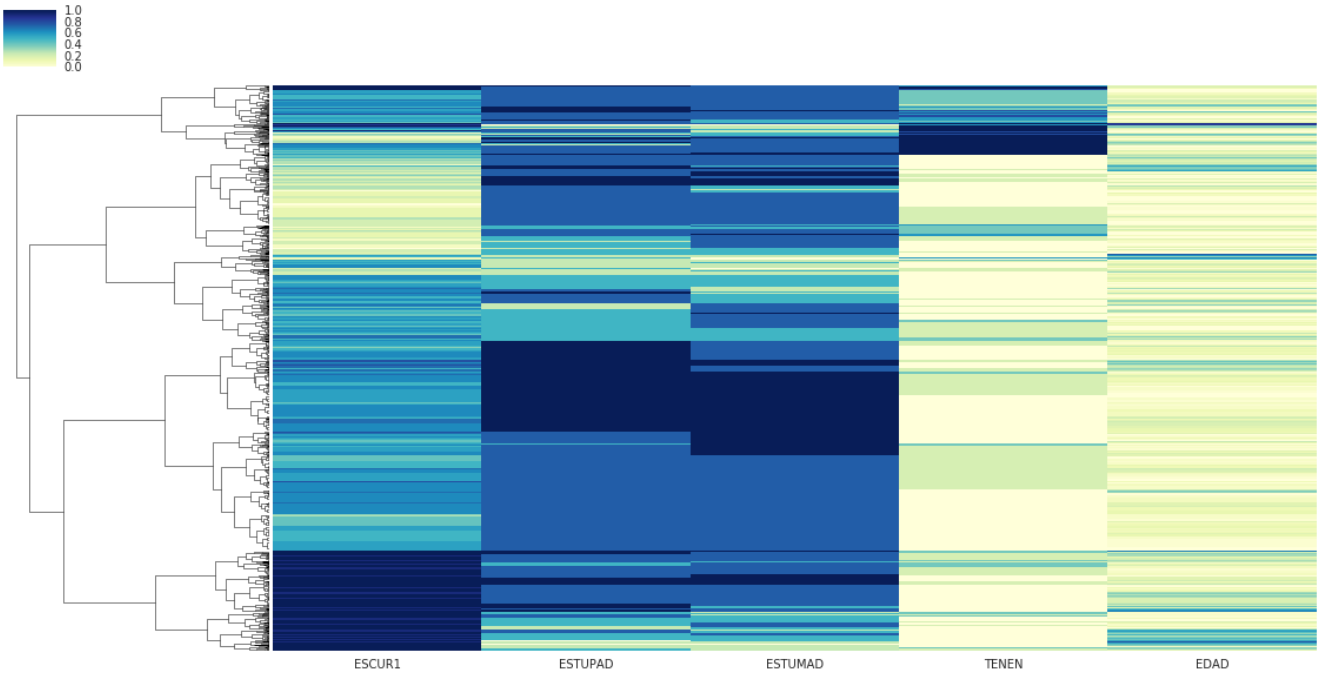


Una vez visto estos resultados, pasamos a ver los resultados del **AgglomerativeClustering-12**



	Elementos	%
0	223	5,93
1	463	12,31
2	446	11,86
3	821	21,84
4	291	7,74
5	82	2,18
6	384	10,21
7	104	2,77
8	511	13,53
9	161	4,28
10	221	5,88
11	53	1,41

Tabla 7: Tabla comparativa clusters AgglomerativeClustering-12 caso de estudio 1



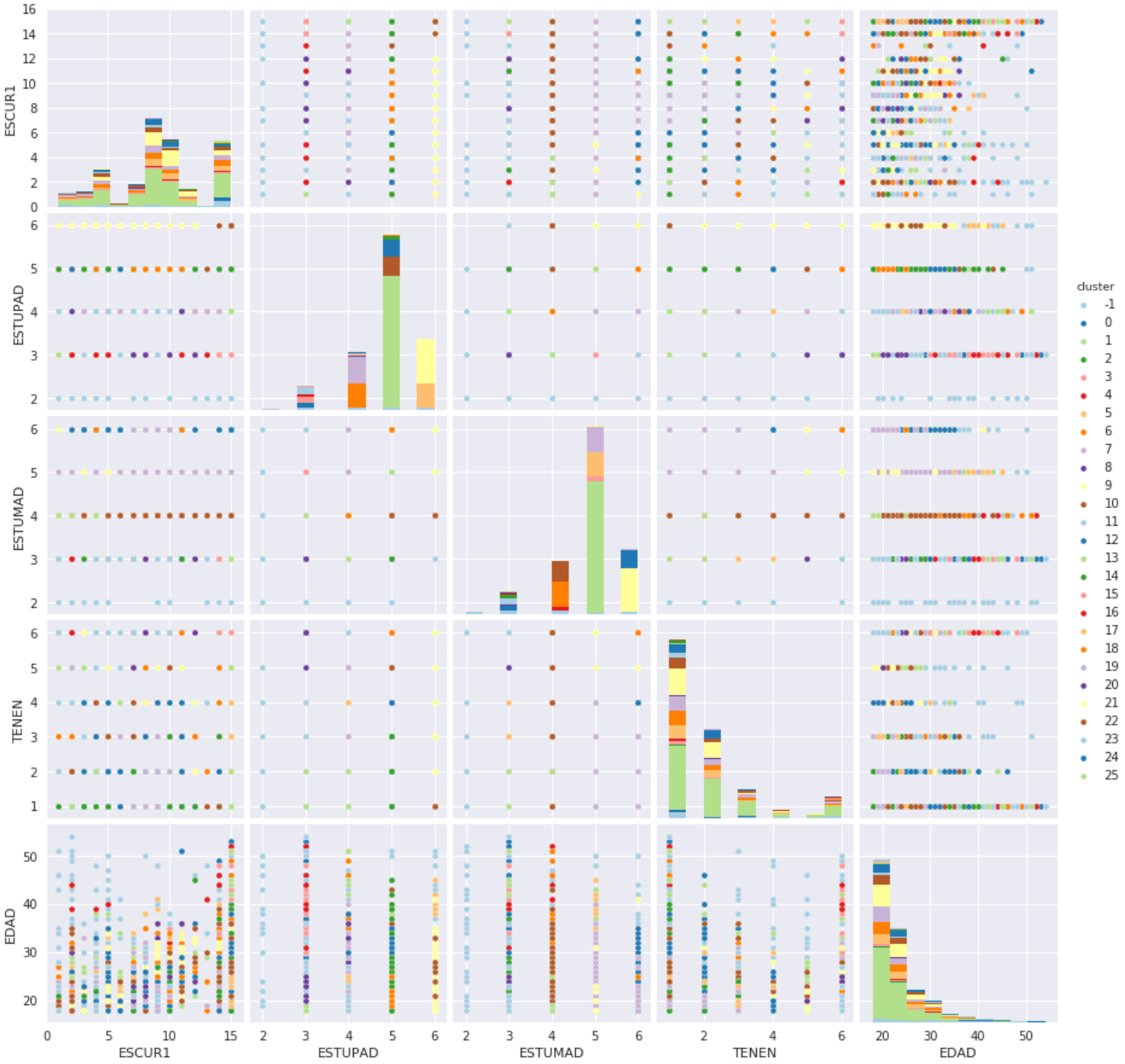
2.2.3. DBSCAN

: Como ya he realizado dos modificaciones a 2 algoritmos, a partir de este, utilizo los parametros por defecto para cada uno de los algoritmos.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
DBSCAN	0,15	94,38	-0,042	27

Tabla 8: Tabla DBSCAN caso de estudio 1

Vamos ahora a ver los resultados obtenidos:


ESCUR1
ESTUPAD
ESTUMAD
TENEN
EDAD
cluster
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

	Elementos	%
0	52	1,38
1	1482	39,41
2	8	0,21
3	62	1,65
4	38	1,01
5	268	7,13
6	279	7,42
7	293	7,79
8	18	0,48
9	499	13,27
10	224	5,96
11	73	1,94
12	194	5,16
13	6	0,16
14	34	0,9
15	7	0,19
16	5	0,13
17	9	0,24
18	10	0,27
19	12	0,32
20	7	0,19
21	6	0,16
22	6	0,16
23	8	0,21
24	2	0,05
25	4	0,11
-1	154	4,1

Tabla 9: Tabla clusters DBSCAN caso de estudio 1

2.2.4. Birch

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Birch	0,09	824,07	0,36	3

Tabla 10: Tabla Birch caso de estudio 1

Ahora vamos a ver los resultados de este algoritmo.

	Elementos	%
0	399	10,61
1	3216	85,53
2	145	3,86

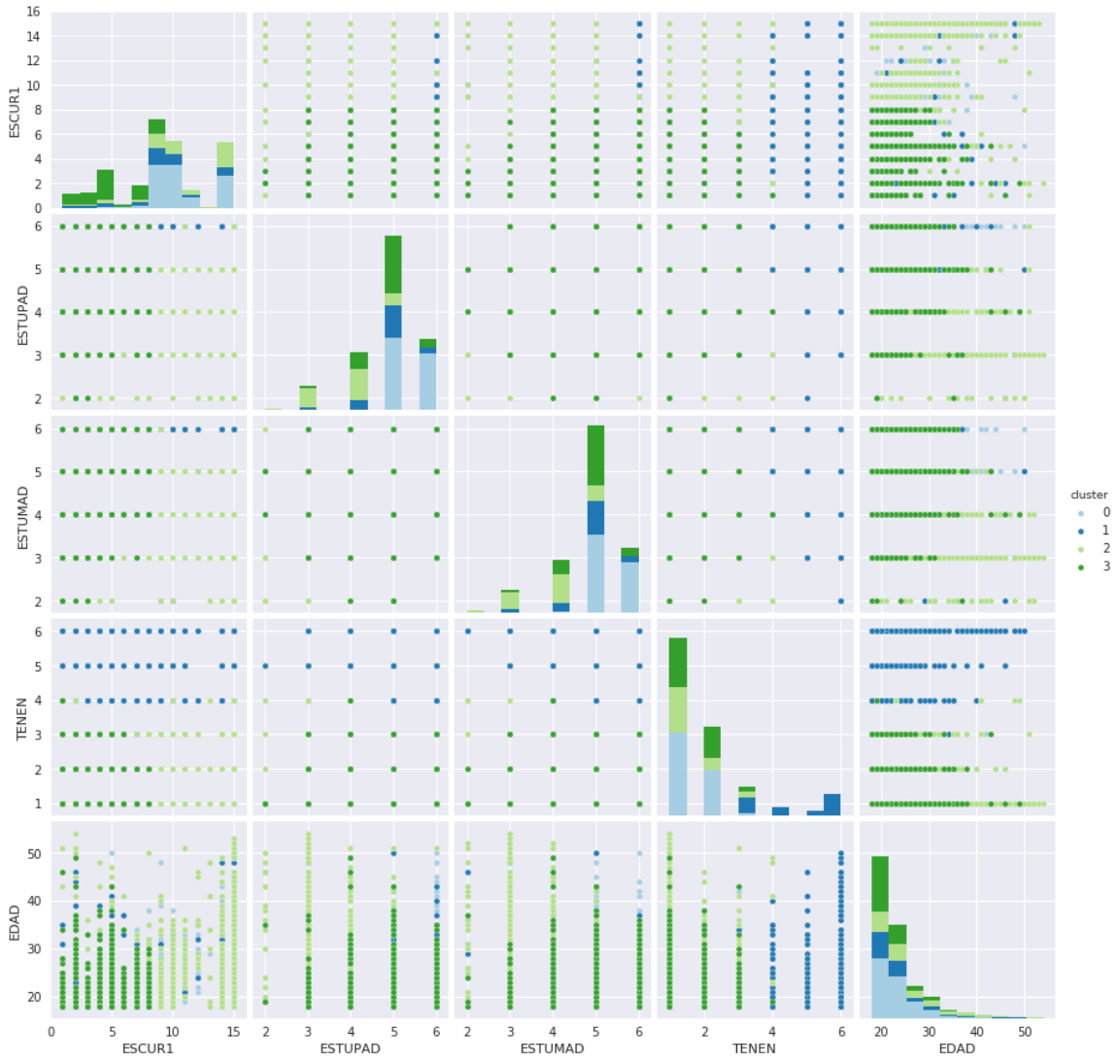
Tabla 11: Tabla clusters Birch caso de estudio 1

2.2.5. SpectralClustering

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
SpectralClustering	2,05	1212,4	0,23	4

Tabla 12: Tabla SpectralClustering caso de estudio 1

Ahora pasamos a ver los resultados:



	Elementos	%
0	1478	39,31
1	588	15,64
2	726	19,31
3	968	25,74

Tabla 13: Tabla clusters SpectralClustering caso de estudio 1

2.3. Interpretación de la segmentación

Podemos ver[2.2] como en términos generales, el mejor resultado fijándonos en las métricas nos los da *AgglomerativeClustering*, sin embargo, este algoritmo no es el más rapido pero tiene un tiempo medio razonable respecto a los demás. Sin embargo, este algoritmo está utilizando únicamente 2 clusters en lo que vemos que utiliza la variable *"TENEN"* para hacer esta separación, es decir, separa entre personas que vivien en casas propias a las que viven de alquiler. Aunque consigamos una mejor valoración en las métricas quiero comentar por ejemplo el *SpectralClustering* que también nos da buenos resultados y al ser 4 clusters, tenemos una visión más amplia de los distintos valores. Como el anterior, también hace una clara diferencia entre clusters con viviendas propias y de alquiler, pero luego, dentro de cada uno de estos podemos ver por ejemplo como el cluster 3, nos concentra a personas que están estudiando niveles "bajos" de estudios, que sus padres sí que tienen niveles superiores de estudios y que vivien en casa propia. Por lo que aunque el *AgglomerativeClustering* tenga unas mejores métricas, el *SpectralClustering* es algo más interesante a la hora de poder definir los distintos clusters.

Por caso contrario, tenemos el DBSCAN que podemos v er que no está haciendo una solución para nada buena. Esto se debe claramente al funcionamiento del mismo y a los parámetros por defecto. Modificando este parámetro conseguía que obtuviera menor número de cluster pero pasaba de 27 a 2, cosa que es demasiado brusta y decidí dejarlo así para que tenga un algoritmo con un número grande de clusters, aunque no he conseguido buenos resultados como se puede comprobar, ya que realiza los clusters mas o menos como el resto de algoritmos pero luego mete clusters muy pequeños con elementos aislados que hacen empeorar las métricas.

3. Caso de estudio 2

3.1. Conjunto de datos

Para este caso de estudio he analizado la población granadina mayor de 29 años, la cual tiene pareja que tiene algun tipo de estudios, viven en casa propia, tienen hijos y es empresari@.

Este subconjunto tiene un enfoque sobre personas que ya estan centradas, es decir, son una familia con hijos que vive en casa propia, para poder comprobar como en el caso anterior, si los estudios realizados de una persona durante su vida tiene algún tipo de relación con su pareja y para comprobar si dependiendo del nivel de estudios una familia suele ser mas o menos numerosa,

Para ello, he decidido crear los clusters utilizando las variables *"EDAD" , "ESREAL" , "ESTUCON" ,"NMIEM" , "NOCU"* que nos muestran la edad, los estudios realizados tantos por la persona como por la pareja, el numero de miembros en la familia y el número de ocupados en la casa. Con este conjunto de datos obtenemos 1270 elementos.

3.2. Resultados algoritmos

En primer lugar comentar que he hecho dos ejecuciones, una ejecución de todos los algoritmos con los parámetros por defecto para cada uno de ellos, y una segunda ejecucion con los algoritmos

KMeans y *AgglomerativeClustering* modificando sus parámetros.

	Tiempo	**Calinski-Harabaz**	**Silhouette Coefficient**	**Clusters**
Kmeans	0,08	368,58	0,26	8
AgglomerativeClustering	0,06	422,88	0,23	2
DBSCAN	0,03	121,33	0,2	15
Birch	0,05	239,06	0,25	3
SpectralClustering	0,22	438,78	0,25	4

Tabla 14: Tabla comparativa parametros por defecto caso de estudio 2

Como apreciamos en la tabla, la columna izquierda nos muestra el nombre de los algoritmos que he decidido utilizar y en las demás columnas podemos ver el resultado de cada una de las métricas medidas como el tiempo que ha tardado y el número de clusters

3.2.1. Kmeans:

Para este algoritmo concretamente he decidido utilizar tanto los parámetros por defecto como una modificación de ellos, obteniendo:

	Tiempo	**Calinski-Harabaz**	**Silhouette Coefficient**	**Clusters**
Kmeans-8	0,08	368,58	0,26	8
Kmeans-3	0,05	494,95	0,3	3

Tabla 15: Tabla comparativa parametros KMeans caso de estudio 2

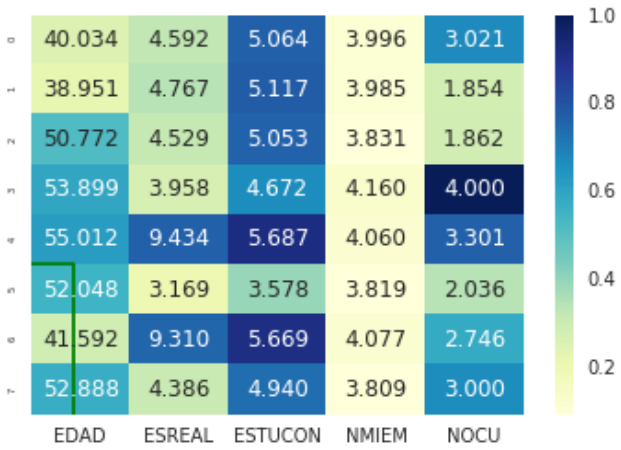
Podemos ver como hemos obtenido una mejor solución modificando estos parámetros y poniendo que nos agrupe en un número menor de clusters que el caso por defecto que tiene Kmeans.

Pasamos ahora a observar el resultado de **Kmeans-8**:

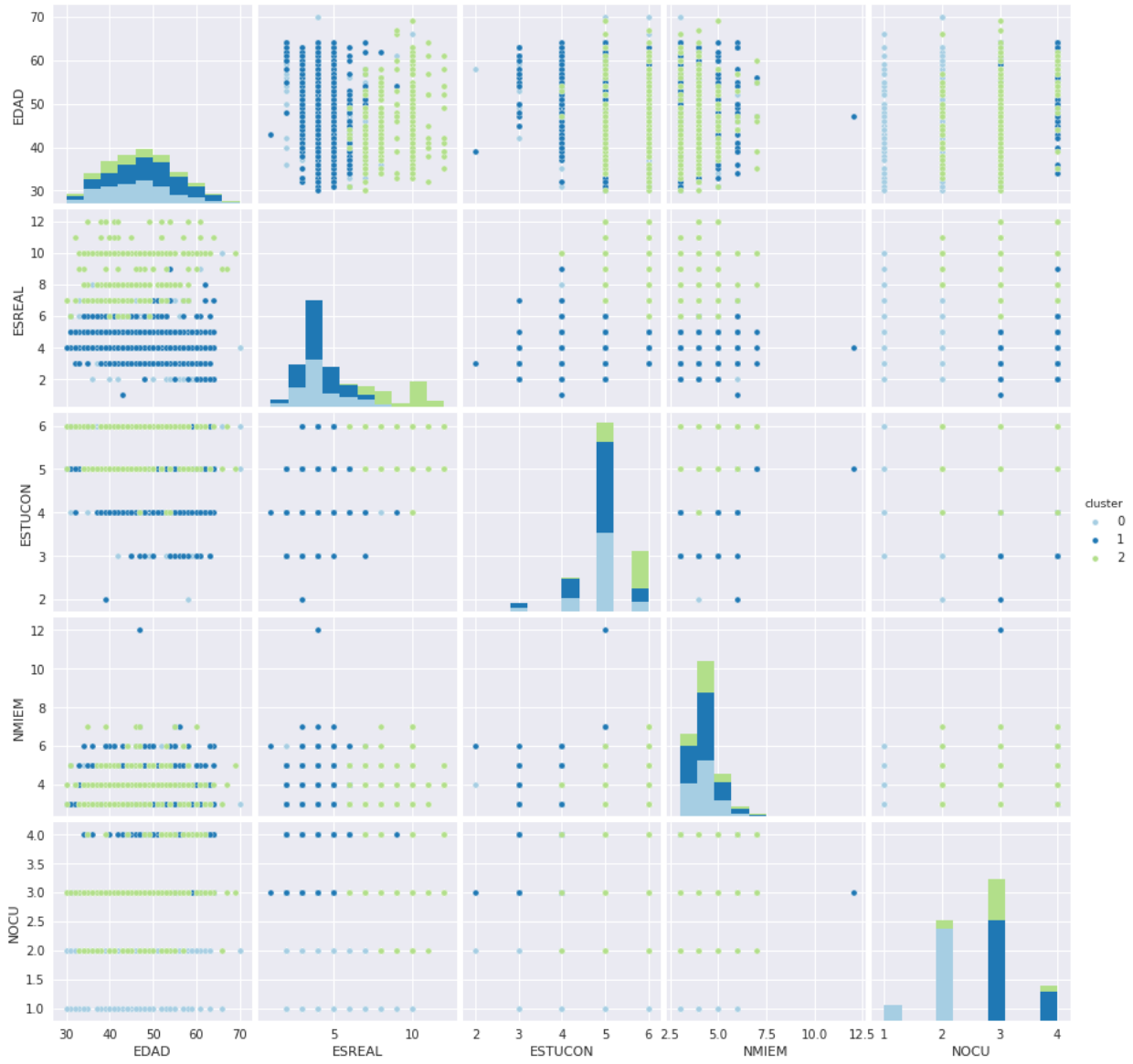
	Elementos	%
0	233	18,35
1	206	16,22
2	189	14,88
3	119	9,37
4	83	6,54
5	83	6,54
6	142	11,18
7	215	16,93

Tabla 16: Tabla comparativa clusters KMeans-8 caso de estudio 2

Con esto, podemos ver tambien el siguiente gráfico que nos da información sobre los centroides.

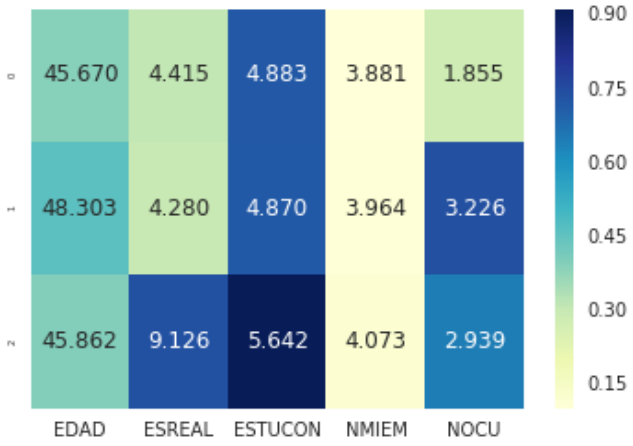


Por otra parte, analizaremos los mismos gráficos para el segundo caso, el caso de **KMeans-3**:



	Elementos	%
0	463	36,46
1	561	44,17
2	246	19,37

Tabla 17: Tabla comparativa clusters KMeans-3 caso de estudio 2



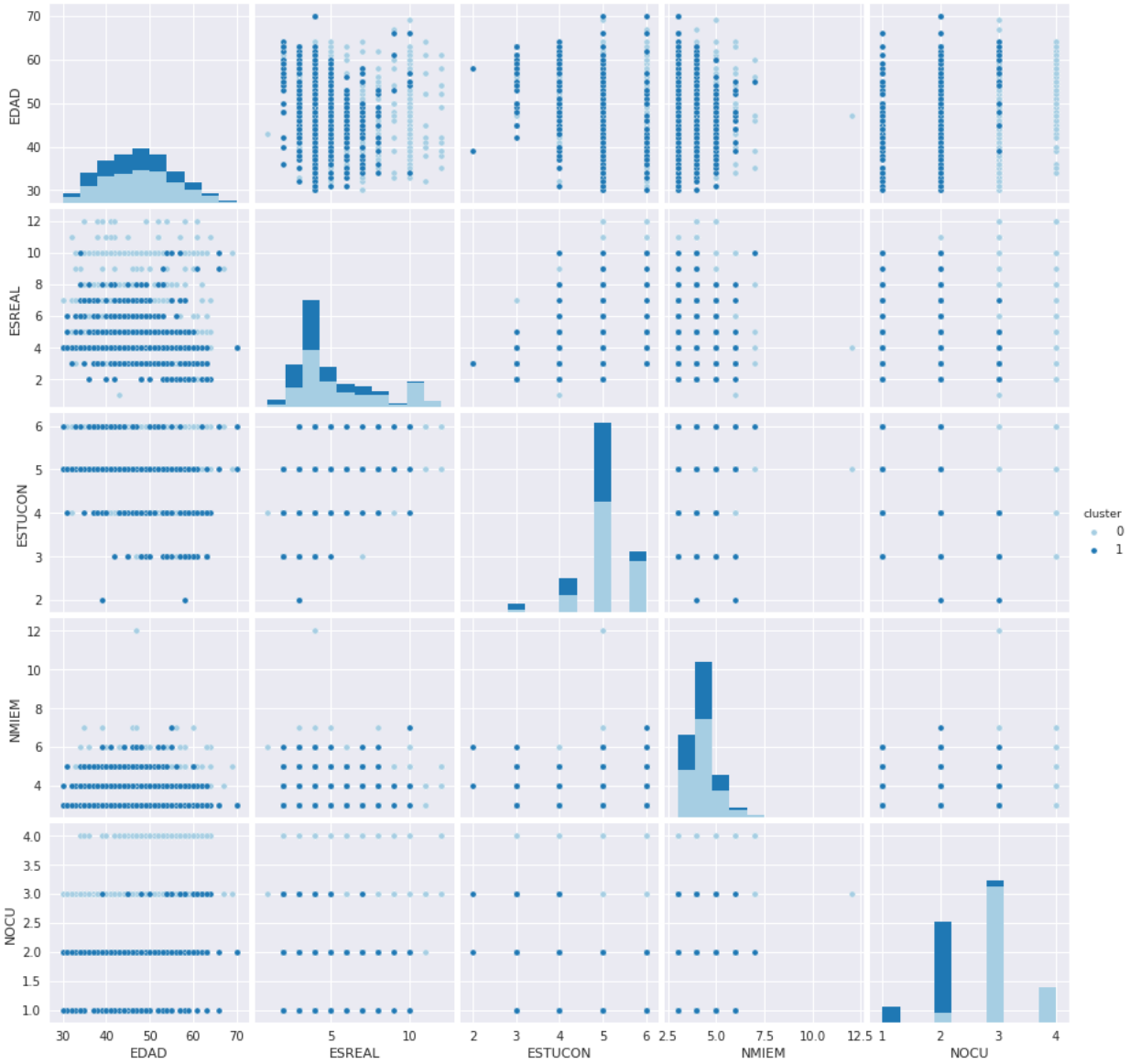
3.2.2. AgglomerativeClustering

Como en el caso anterior, he decidido hacer una modificación de los parámetros de este algoritmo, utilizando en número de clusters.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
AgglomerativeClustering-2	0,06	422,88	0,23	2
AgglomerativeClustering-12	0,06	270,68	0,22	12

Tabla 18: Tabla comparativa parametros AgglomerativeClustering caso de estudio 2

Pasamos ahora a mostrar los resultados para **AgglomerativeClustering-2**:



	Elementos	%
0	781	61,5
1	489	38,5

Tabla 19: Tabla comparativa clusters AgglomerativeClustering-2 caso de estudio 2

Una vez visto estos resultados, pasamos a ver los resultados del **AgglomerativeClustering-12**

	Elementos	%
0	229	18,03
1	99	7,8
2	85	6,69
3	145	11,42
4	90	7,09
5	114	8,98
6	39	3,07
7	132	10,39
8	205	16,14
9	53	4,17
10	41	3,23
11	38	2,99

Tabla 20: Tabla comparativa clusters AgglomerativeClustering-12 caso de estudio 2

3.2.3. DBSCAN

: Como ya he realizado dos modificaciones a 2 algoritmos, a partir de este, utilizo los parametros por defecto para cada uno de los algoritmos.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
DBSCAN	0,03	121,33	0,2	15

Tabla 21: Tabla DBSCAN caso de estudio 2

Vamos ahora a ver los resultados obtenidos:


EDAD
ESREAL
ESTUCON
NMIEM
NOCU
cluster
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
13

	Elementos	%
0	306	24,09
1	29	2,28
2	380	29,92
3	84	6,61
4	8	0,63
5	162	12,76
6	24	1,89
7	53	4,17
8	44	3,46
9	60	4,72
10	15	1,18
11	49	3,86
12	9	0,71
13	12	0,94
14	35	2,76

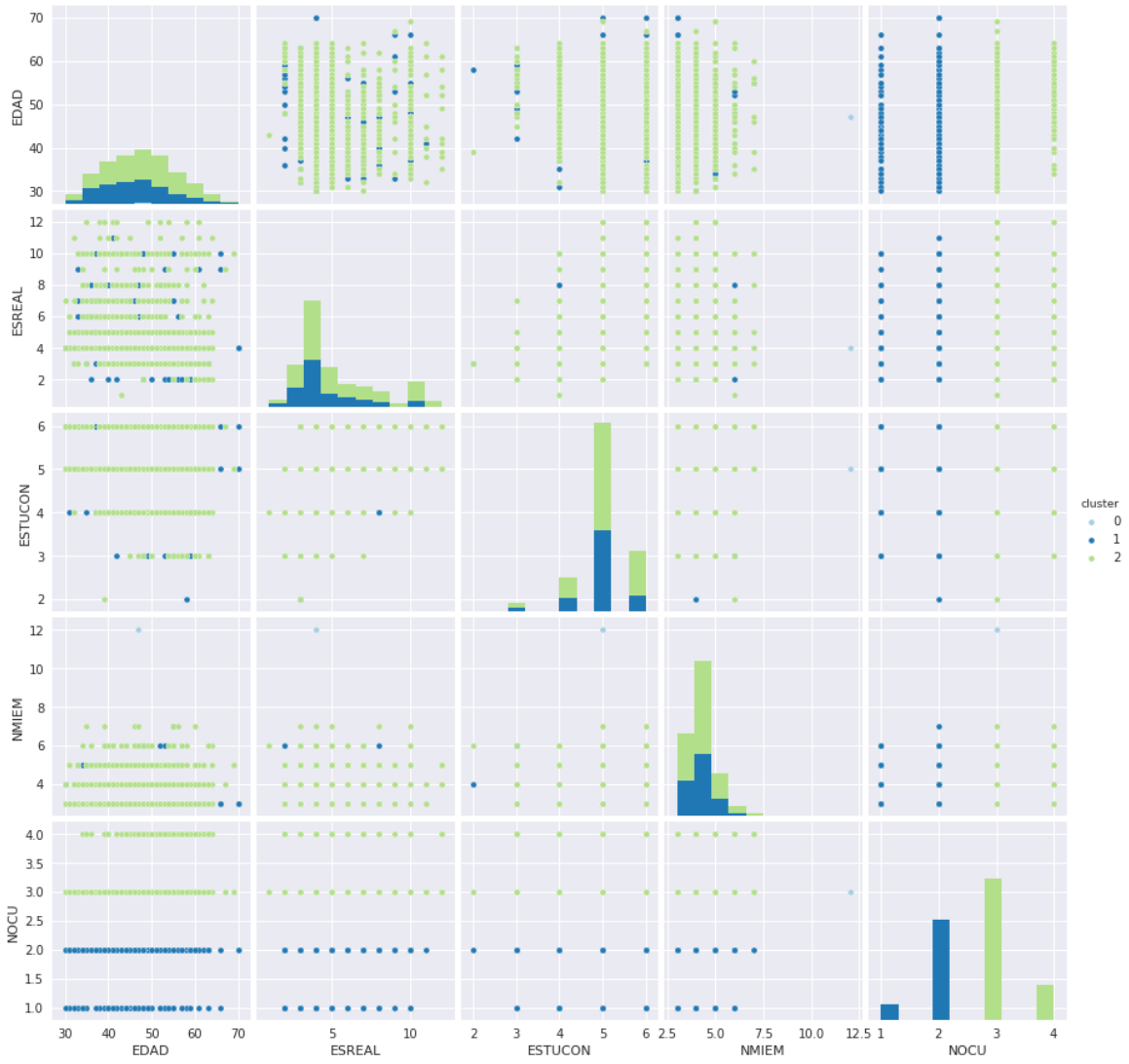
Tabla 22: Tabla clusters DBSCAN caso de estudio 2

3.2.4. Birch

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Birch	0,05	239,06	0,25	3

Tabla 23: Tabla Birch caso de estudio 2

Ahora vamos a ver los resultados de este algoritmo.



	Elementos	%
0	1	0,08
1	504	39,69
2	765	60,24

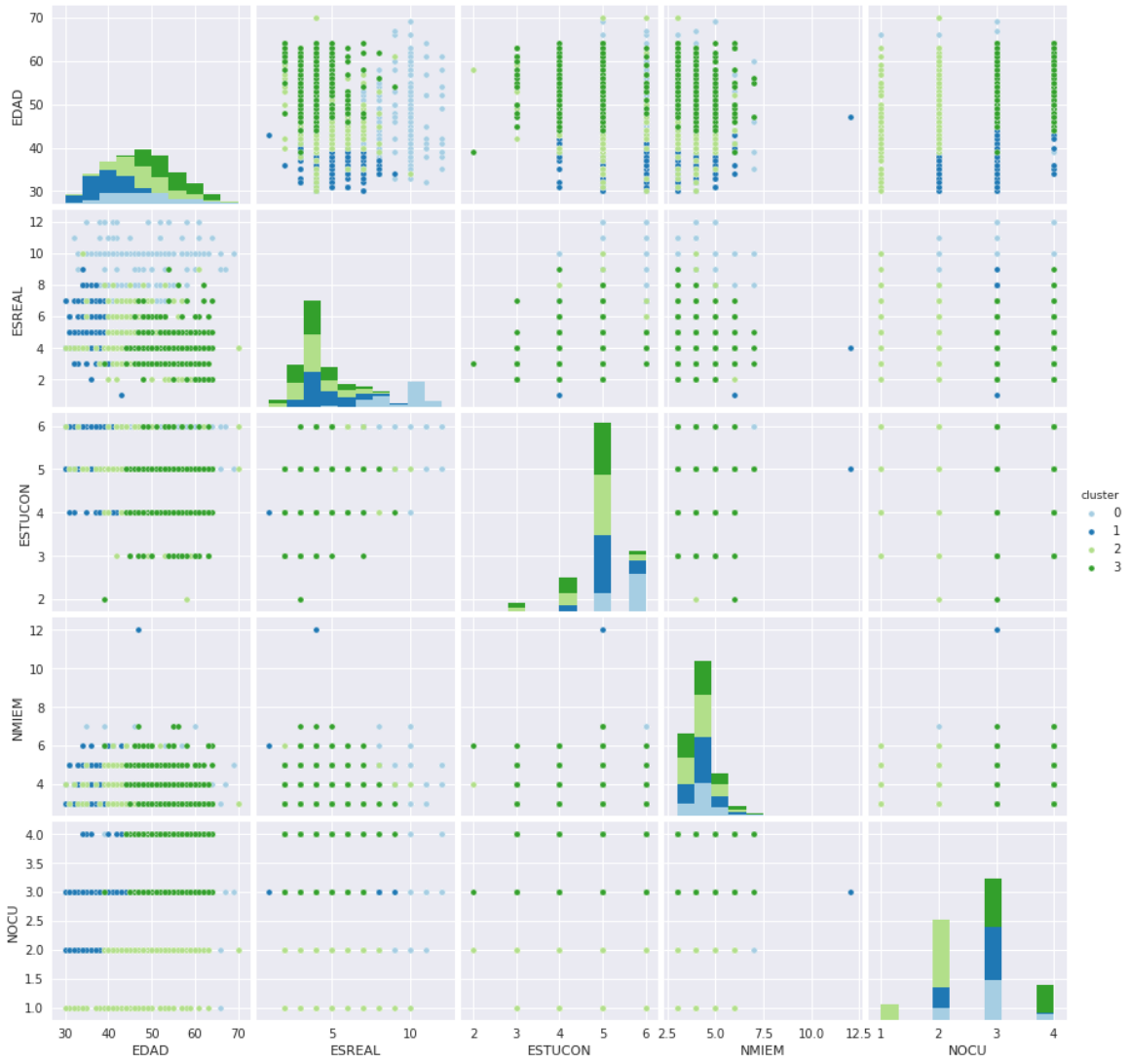
Tabla 24: Tabla clusters Birch caso de estudio 2

3.2.5. SpectralClustering

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
SpectralClustering	0,22	438,78	0,25	4

Tabla 25: Tabla SpectralClustering caso de estudio 2

Ahora pasamos a ver los resultados:



	Elementos	%
0	247	19,45
1	334	26,3
2	361	28,43
3	328	25,83

Tabla 26: Tabla clusters SpectralClustering caso de estudio 2

3.3. Interpretación de la segmentación

Podemos ver[3.2] como en este caso, el algoritmo *SpectralClustering* nos hace en media un mejor resultado respecto a las métricas, sin embargo, con la modificación realizada sobre el *KMeans* se ha conseguido una mejora aún mas significativa que este algoritmo. Voy a comentar sobre el *KMeans-3*[3.2.1]. Podemos ver como el cluster 0 y el cluster 1 están claramente diferenciados con la variable *"NOCU"* que nos dice el número de personas ocupadas en el hogar, es decir, estos clusters se diferencian esencialmente en el número de ocupados en el hogar. Por otra parte, podemos ver

como el cluster 2 se diferencia del resto por la variable *"ESREAL"*, por lo que este cluster se diferencia del resto porque las personas que están en él tienen estudios relativamente altos con respecto a los otros dos clusters. Con estas principales diferenciaciones podemos mirar el resto de variables y comprobar más características sobre ellos, como por ejemplo, que para todos los casos donde la persona tiene estudios relativamente altos, su pareja/cónyugue también los tiene.

Por otra parte, tenemos que el peor algoritmo en este caso sigue siendo *DBSCAN*, como he comentado anteriormente, he estado modificando los parámetros del mismo para intentar obtener un resultado aceptable respecto al resto, pero el algoritmo es demasiado crítico con los algoritmos y a mímino cambio que realice, cambia mucho su resultado y por ello dejé un valor por defecto. Como he comentado anteriormente aunque éste sea el peor, si miramos el scatter, veremos que los grupos principales también los crea, únicamente que mete muchos grupos pequeños y esto ensucia las métricas. Una posible solución podría ser poner un umbral de un número de miembros mínimos y eliminar esos clusters que no superen ese umbral. Esto mismo se puede hacer cambiando un parámetro del algoritmo y decir que para que un elemento se considere centroide, su cluster tiene que ser de al menos x miembros, no obstante, modificando este parámetro tampoco conseguí los resultados deseados.

4. Caso de estudio 3

4.1. Conjunto de datos

Para este caso de estudio he analizado la población granadina que cuidan de personas con problemas y que trabajan en otro municipio diferente al suyo.

Este subconjunto tiene un enfoque sobre personas que tienen que estar al cargo de personas con problemas y que por cuestiones de trabajo tiene que viajar a otro municipio, lo que esto hace que le quite tiempo para cuidar a estas personas.

Para ello, he decidido crear los clusters utilizando las variables *"EDAD"* ,*"NVIAJE"* , *"TDESP"* , *"NMIEM"* , *"H6584"* que nos muestran la edad, el número de viajes que hace al día, el tiempo que tarda en desplazarse, el número de miembros en la familia y el numero de miembros que tienen entre 65 y 84 años. Con este conjunto de datos obtenemos 976 elementos.

4.2. Resultados algoritmos

En primer lugar comentar que he hecho dos ejecuciones, una ejecución de todos los algoritmos con los parámetros por defecto para cada uno de ellos, y una segunda ejecucion con los algoritmos *KMeans* y *AgglomerativeClustering* modificando sus parámetros.

Como apreciamos en la tabla, la columna izquierda nos muestra el nombre de los algoritmos que he decidido utilizar y en las demás columnas podemos ver el resultado de cada una de las métricas medidas como el tiempo que ha tardado y el número de clusters

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Kmeans	0,06	323,81	0,33	8
AgglomerativeClustering	0,03	263,02	0,31	2
DBSCAN	0,03	147,18	0,43	15
Birch	0,03	199,37	0,41	3
SpectralClustering	0,12	373,5	0,29	4

Tabla 27: Tabla comparativa parametros por defecto caso de estudio 3

4.2.1. Kmeans:

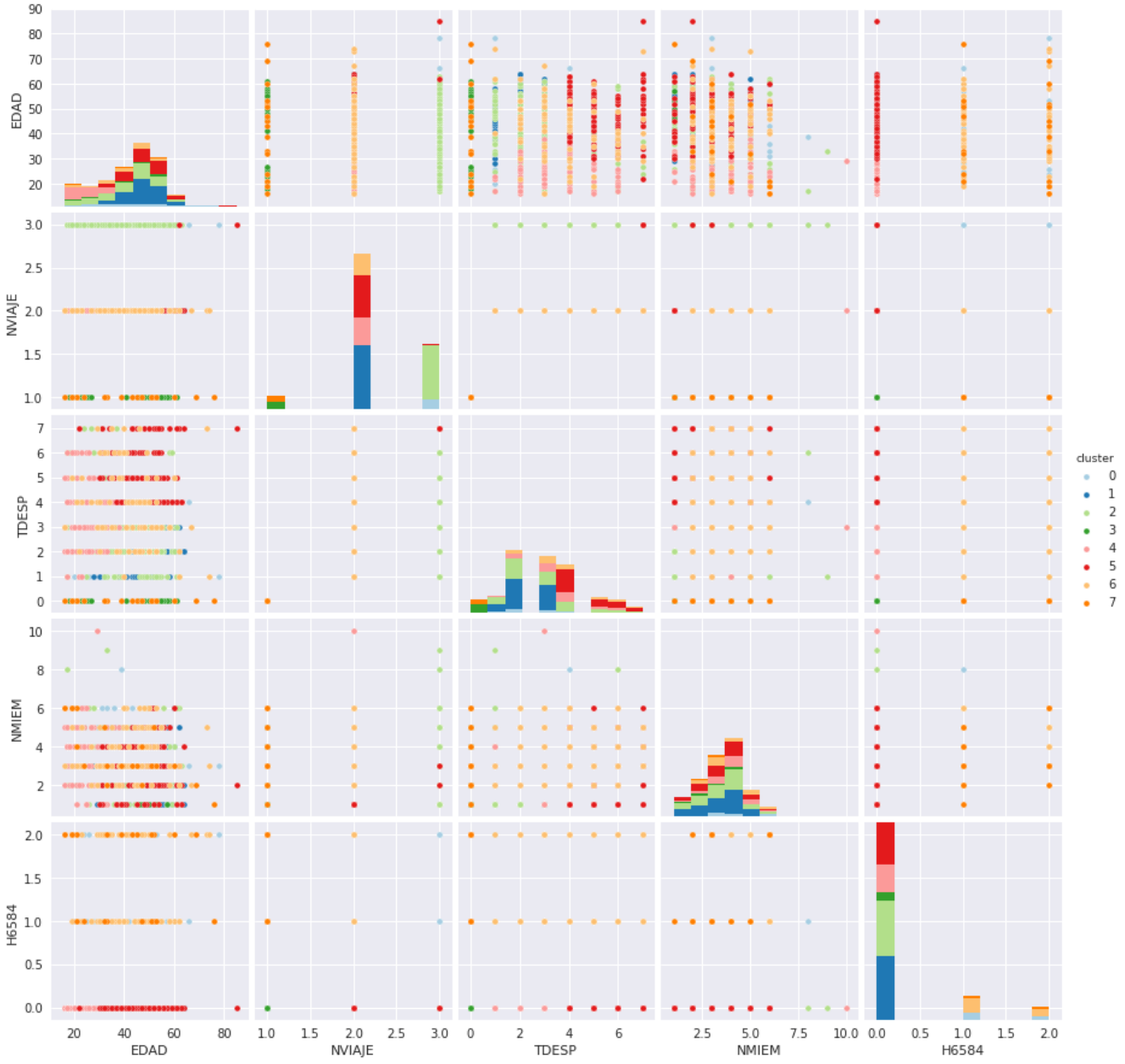
Para este algoritmo concretamente he decidido utilizar tanto los parámetros por defecto como una modificación de ellos, obteniendo:

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Kmeans-8	0,06	323,81	0,33	8
Kmeans-3	0,02	411,9	0,4	3

Tabla 28: Tabla comparativa parametros KMeans caso de estudio 3

Podemos ver como hemos obtenido una mejor solución modificando estos parámetros y poniendo que nos agrupe en un número menor de clusters que el caso por defecto que tiene Kmeans.

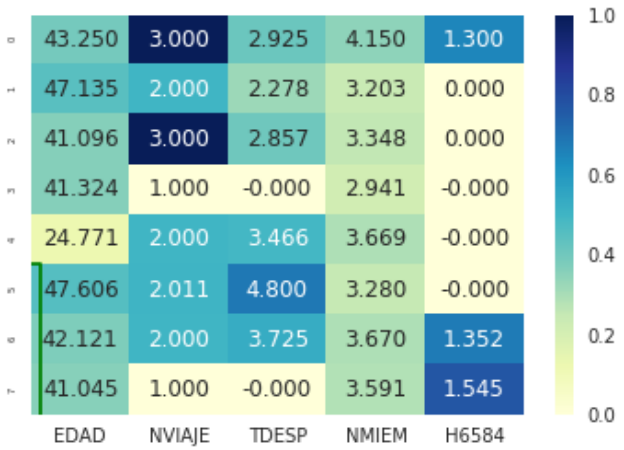
Pasamos ahora a observar el resultado de **Kmeans-8**:



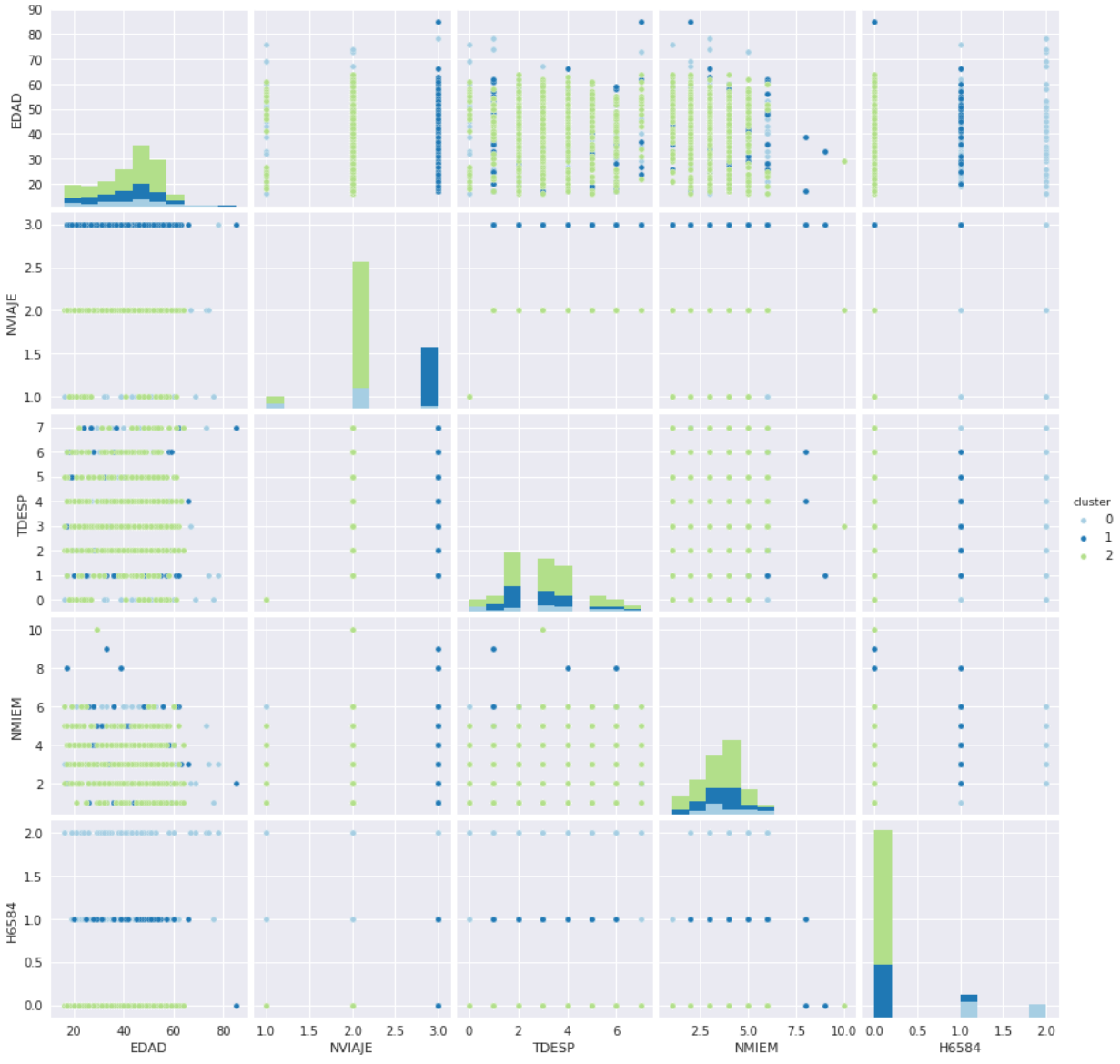
	Elementos	%
0	40	4,1
1	266	27,25
2	230	23,57
3	34	3,48
4	118	12,09
5	175	17,93
6	91	9,32
7	22	2,25

Tabla 29: Tabla comparativa clusters KMeans-8 caso de estudio 3

Con esto, podemos ver tambien el siguiente gráfico que nos da información sobre los centroides.

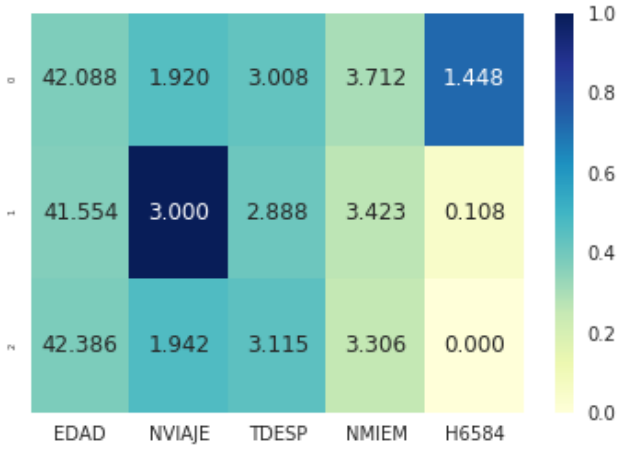


Por otra parte, analizaremos los mismos gráficos para el segundo caso, el caso de **KMeans-3**:



	Elementos	%
0	125	12,81
1	260	26,64
2	591	60,55

Tabla 30: Tabla comparativa clusters KMeans-3 caso de estudio 3



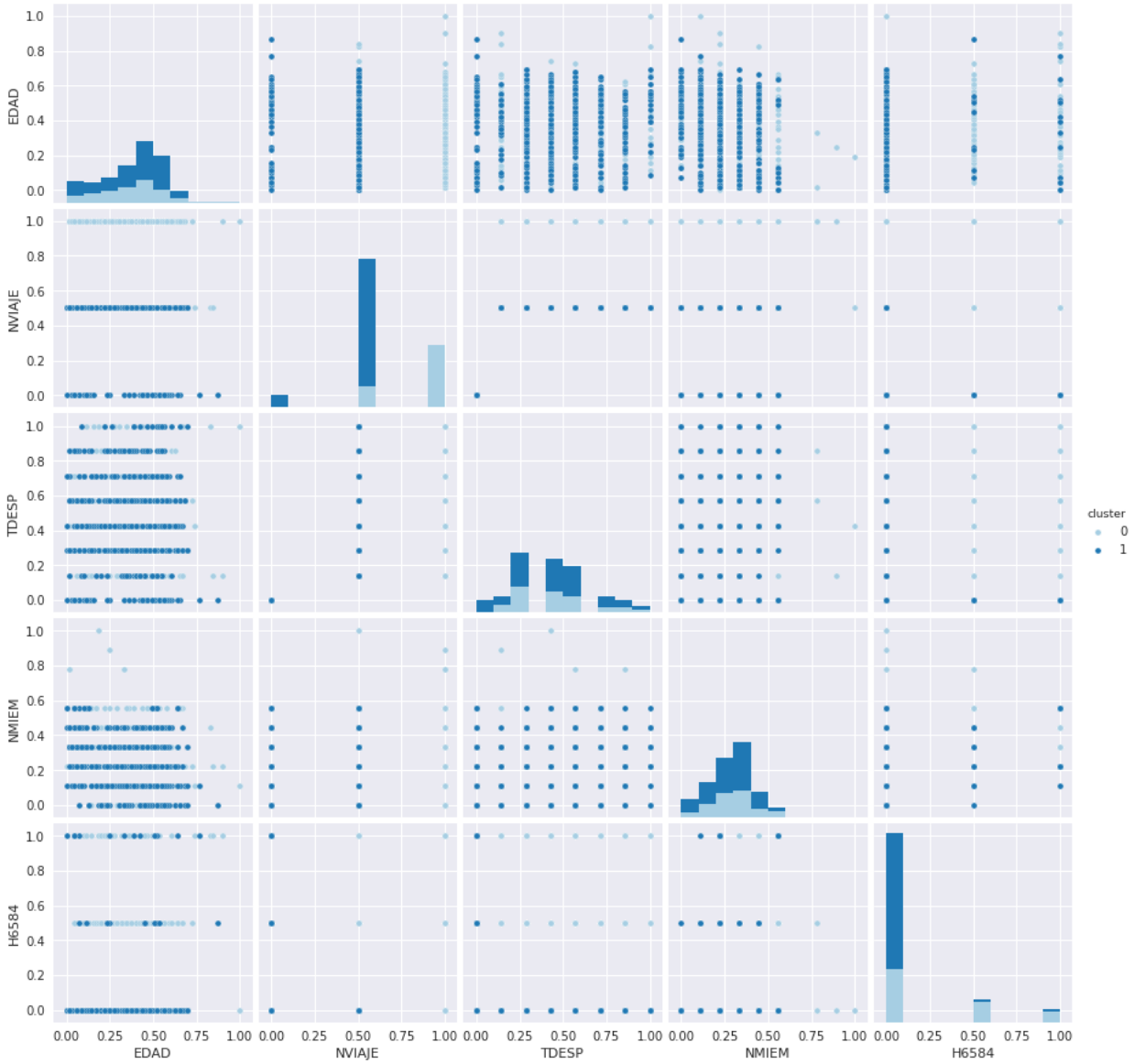
4.2.2. AgglomerativeClustering

Como en el caso anterior, he decidido hacer una modificación de los parámetros de este algoritmo, utilizando en número de clusters.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
AgglomerativeClustering-2	0,03	263,02	0,31	2
AgglomerativeClustering-12	0,03	282,05	0,25	12

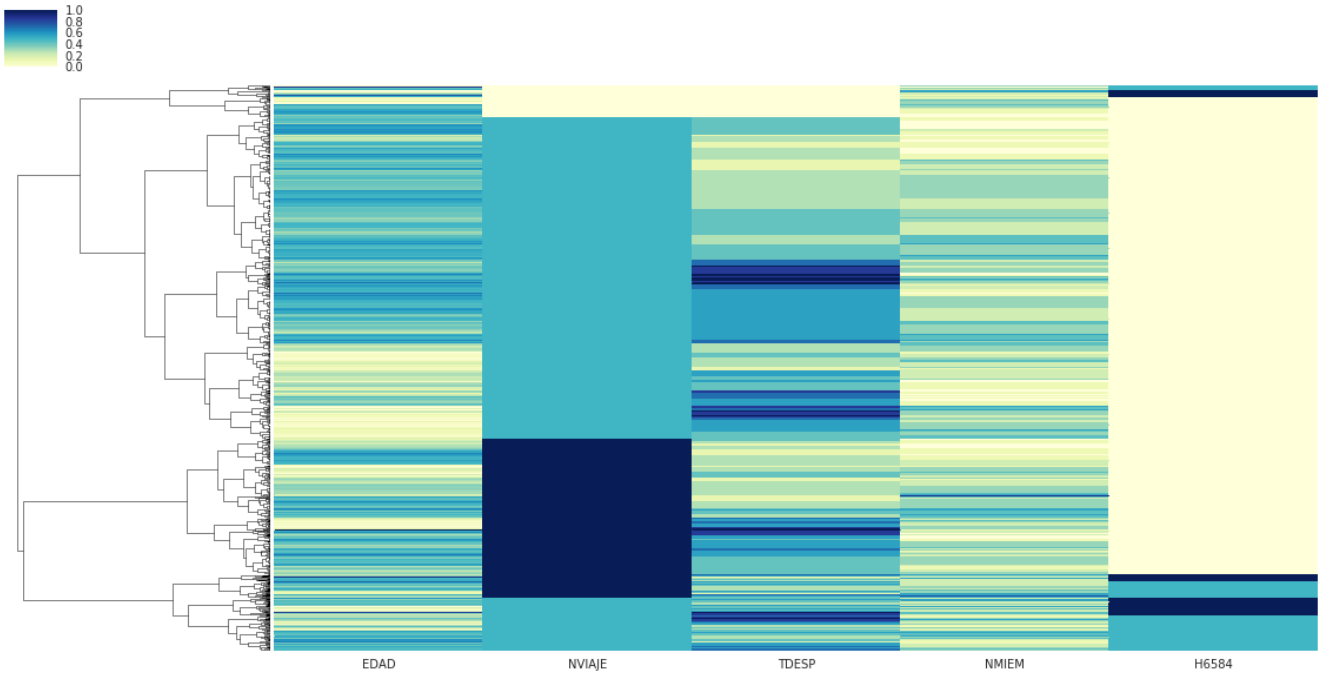
Tabla 31: Tabla comparativa parametros AgglomerativeClustering caso de estudio 3

Pasamos ahora a mostrar los resultados para **AgglomerativeClustering-2**:

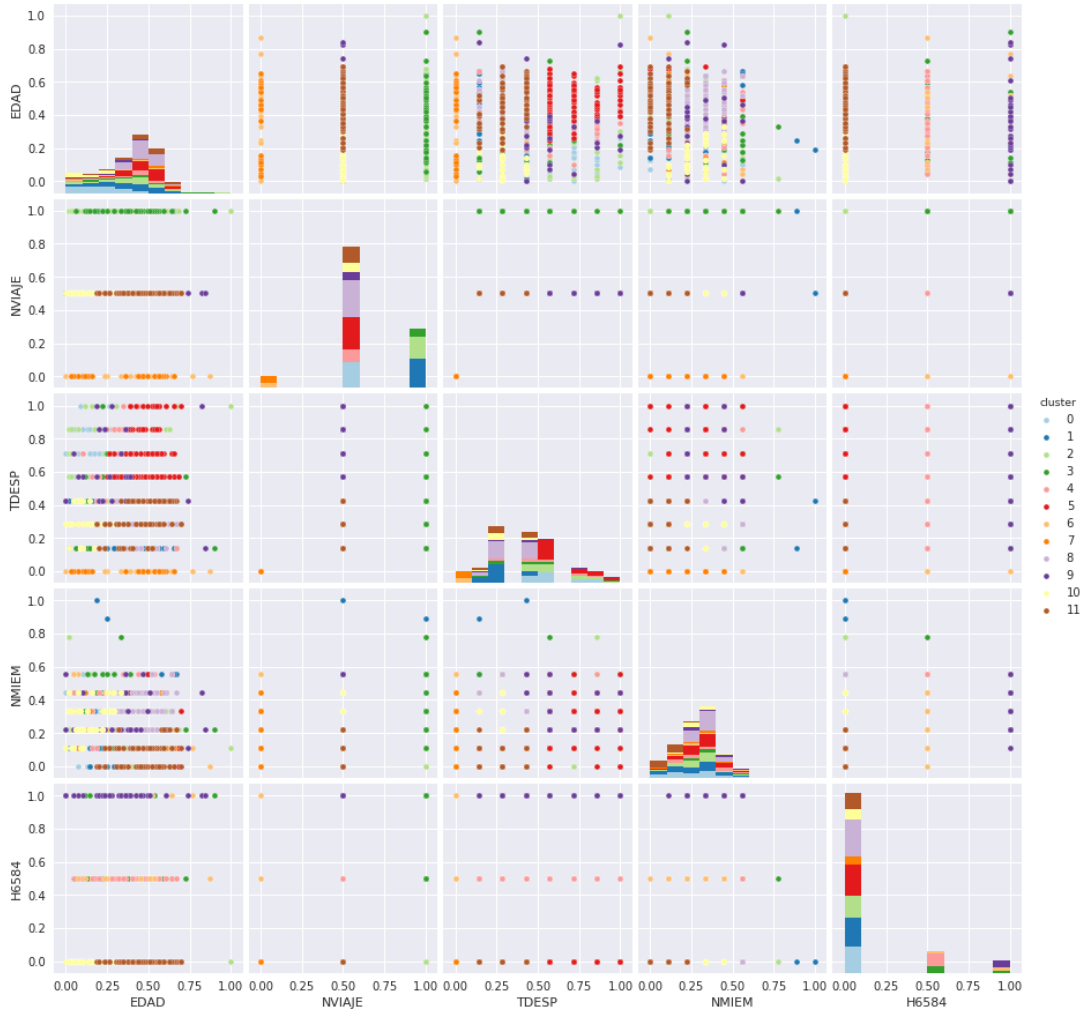


	Elementos	%
0	364	37,3
1	612	62,7

Tabla 32: Tabla comparativa clusters AgglomerativeClustering-2 caso de estudio 3

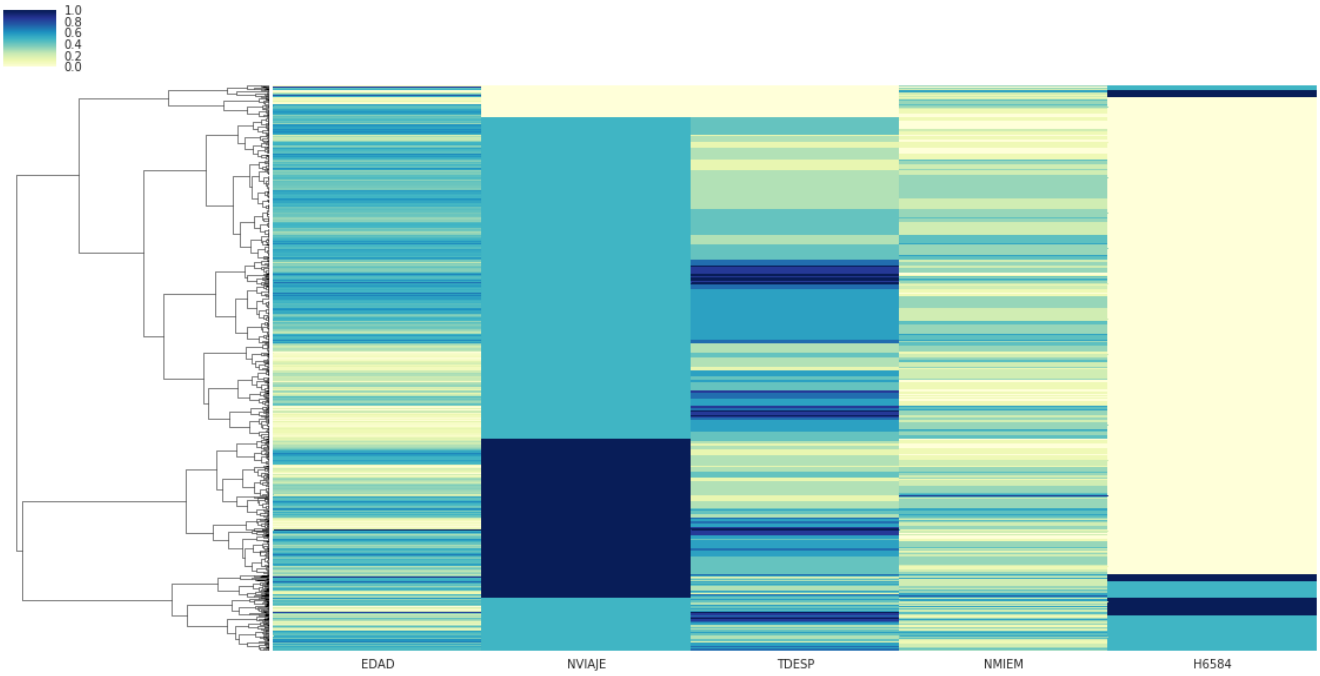


Una vez visto estos resultados, pasamos a ver los resultados del **AgglomerativeClustering-12**



	Elementos	%
0	118	12,09
1	135	13,83
2	98	10,04
3	40	4,1
4	59	6,05
5	145	14,86
6	22	2,25
7	34	3,48
8	172	17,62
9	32	3,28
10	47	4,82
11	74	7,58

Tabla 33: Tabla comparativa clusters AgglomerativeClustering-12 caso de estudio 3



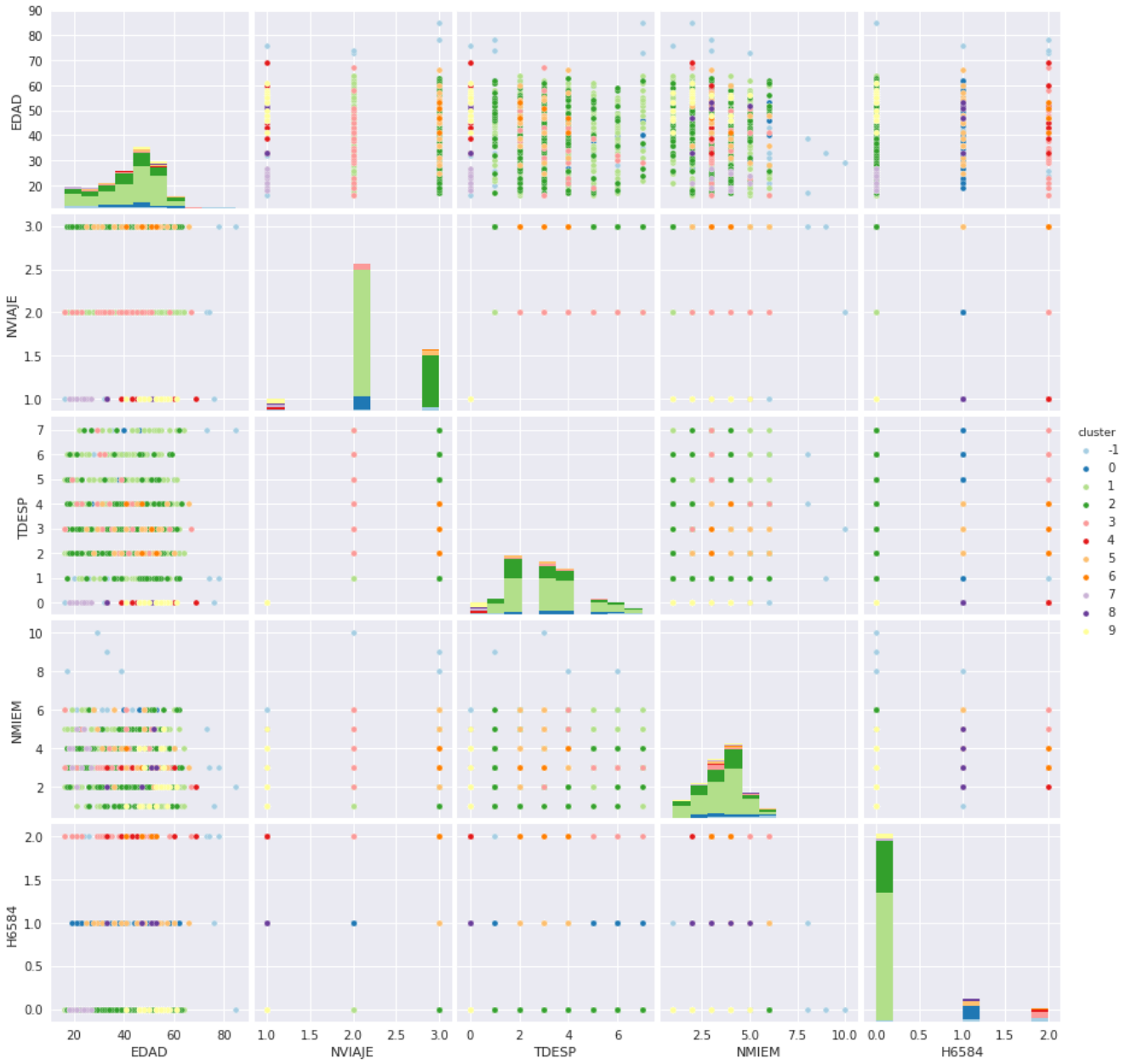
4.2.3. DBSCAN

: Como ya he realizado dos modificaciones a 2 algoritmos, a partir de este, utilizo los parametros por defecto para cada uno de los algoritmos.

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
DBSCAN	0,03	147,18	0,43	15

Tabla 34: Tabla DBSCAN caso de estudio 3

Vamos ahora a ver los resultados obtenidos:



	Elementos	%
0	59	6,05
1	556	56,97
2	228	23,36
3	28	2,87
4	8	0,82
5	23	2,36
6	5	0,51
7	12	1,23
8	6	0,61
9	22	2,25
10	29	2,97

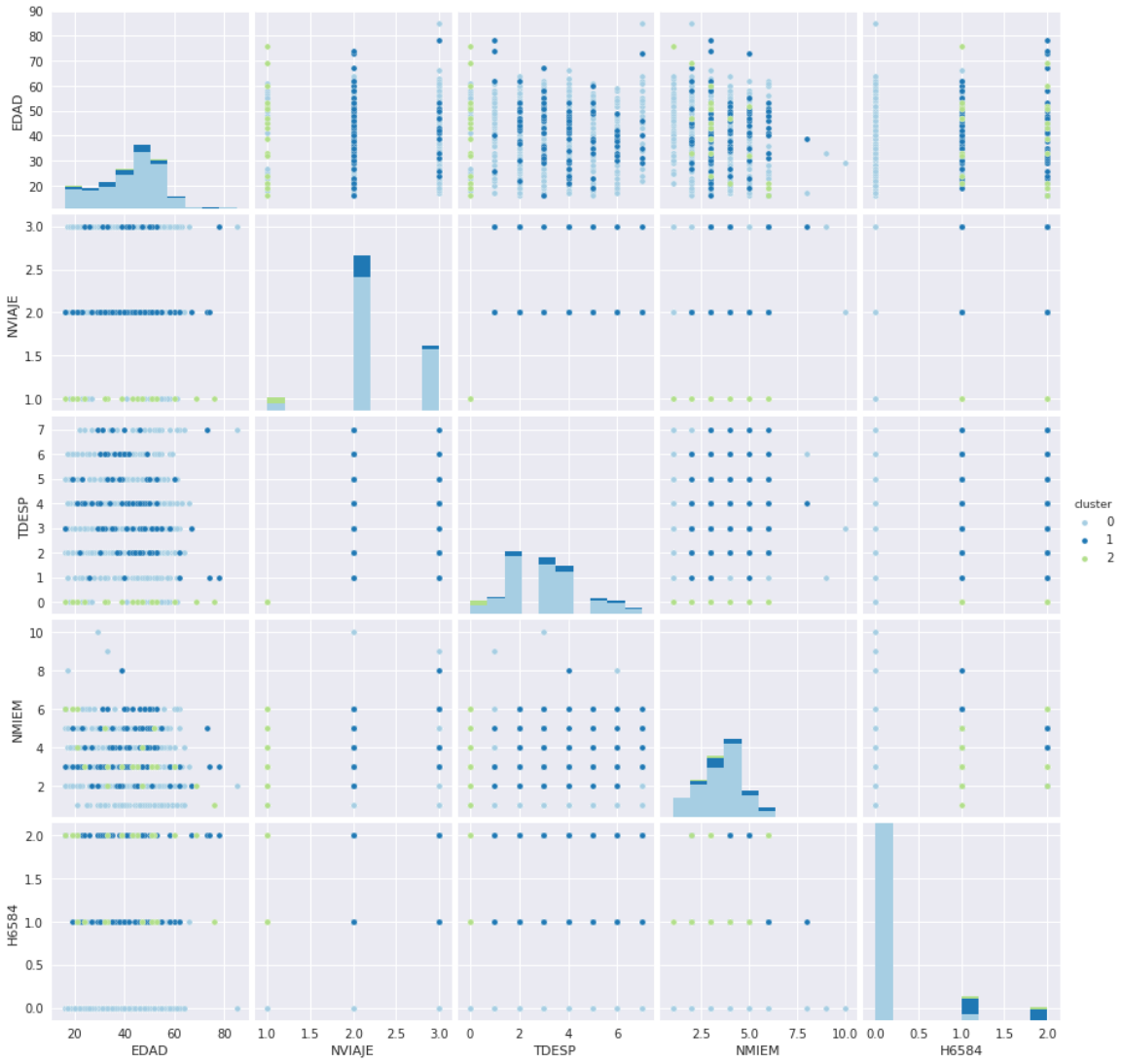
Tabla 35: Tabla clusters DBSCAN caso de estudio 3

4.2.4. Birch

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
Birch	0,03	199,37	0,41	3

Tabla 36: Tabla Birch caso de estudio 3

Ahora vamos a ver los resultados de este algoritmo.



	Elementos	%
0	848	86,89
1	106	10,86
2	22	2,25

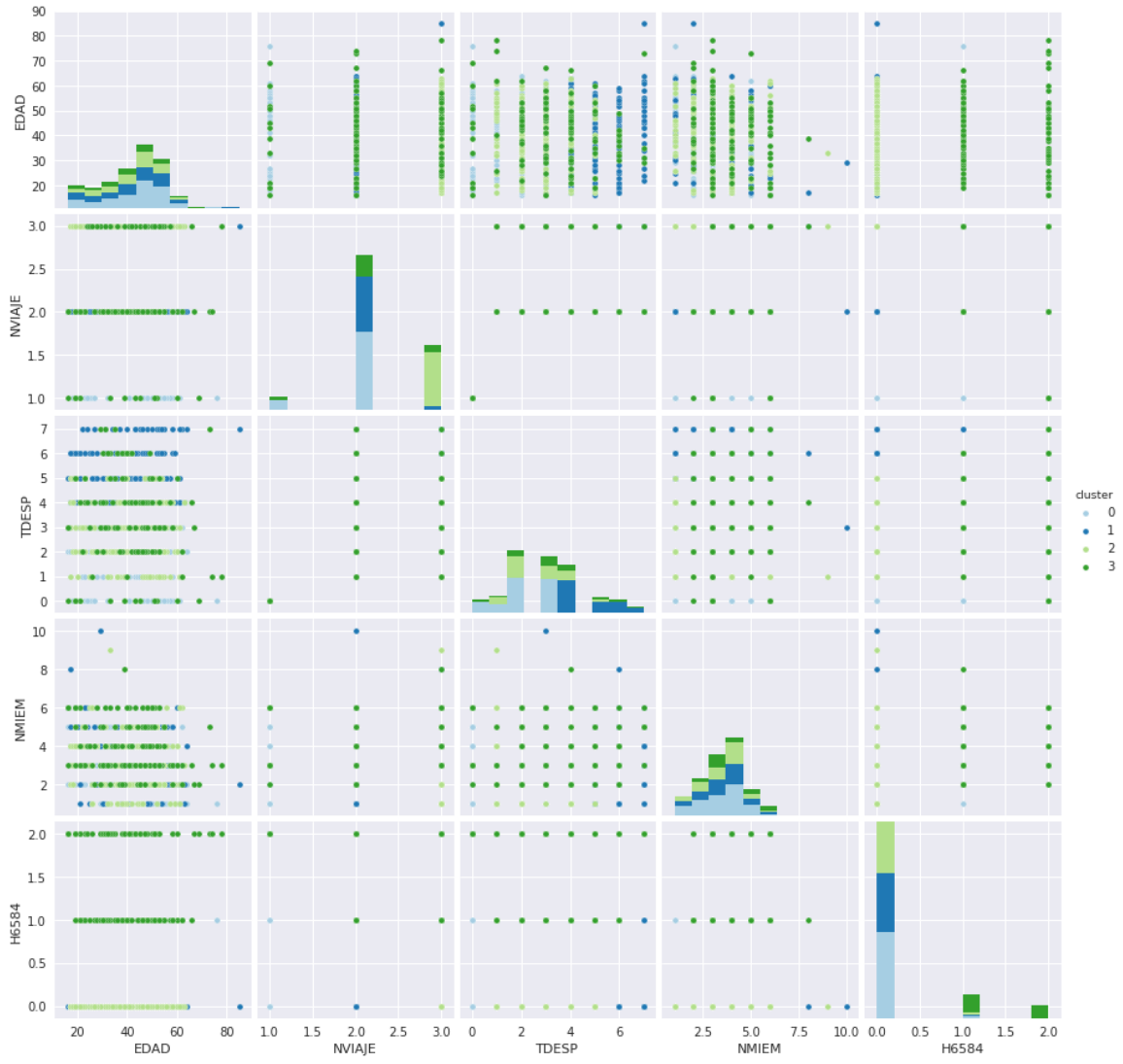
Tabla 37: Tabla clusters Birch caso de estudio 3

4.2.5. SpectralClustering

	Tiempo	Calinski-Harabaz	Silhouette Coefficient	Clusters
SpectralClustering	0,12	373,5	0,29	4

Tabla 38: Tabla SpectralClustering caso de estudio 3

Ahora pasamos a ver los resultados:



	Elementos	%
0	370	37,91
1	248	25,41
2	227	23,26
3	131	13,42

Tabla 39: Tabla clusters SpectralClustering caso de estudio 3

4.3. Interpretación de la segmentación

Para este caso podemos ver[4.2] como tanto *KMeans* como *SpectralClustering* hacen unas métricas mas o menos parecidas a excepción del tiempo que gana *Kmeans*. No obstante, de nuevo, con el cambio de parámetros de *KMeans* he vuelto a obtener unas mejores métricas, por lo que voy a comentar este algorimto[4.2.1]. Apriori, vemos que no tenemos mucha diferencia ni unas conclusiones muy claras sobre este tema, sin embargo, podemos apreciar como el cluster 0 nos diferencia del resto en que las personas de este cluster, a parte de tener que cuidar a una persona enferma que ya está especificado en el conjunto de datos, tienen a una o dos personas entre 65 y 84 años en la casa. Es facil comprobar gracias al cluster 1 que diferencia al resto de el número de viajes que realizan, que un conjunto de personas que tienen personas mayores en su casa realizan menos viajes al día que las personas que tienen al menos 1 persona mayor en su casa. Podemos refinal más cada uno de los casos si analizamos algoritmos con más clusters, pero he intentado hacer un consenso entre la calidad que devuelven las métricas y el número de clusters para que sea visible y poder sacar razonamientos.

Por otra parte como en los casos anteriores, *DBSCAN* obtiene las peores métricas, de nuevo recalco que creo que es por los motivos que he expresado en el caso 2 y en el 1.

Referencias

[1] Web asignatura, sci2s.ugr.es, `https://sci2s.ugr.es/graduateCourses/in`, Accedido el 1 de diciembre de 2018.

[2] Sklearn references, scikit-learn.org, `https://scikit-learn.org/stable/modules/clustering.html`, Accedido el 1 de diciembre de 2018.