

Practica 2

Segmentación para Análisis Empresarial

Curso 2018-2019

David Castro Salazar

Índice

1. Introducción	3
2. Casos de estudio	4
2.1 Caso de estudio 1	4
2.1.1. Introducción de caso de estudio	4
2.1.2. Variables del caso de estudio	4
2.1.3. Algoritmos	5
- K-means	5
- Birch	8
- DBSCAN	9
- AffinityPropagation	11
- AgglomerativeClustering	13
2.1.4. Interpretación de la segmentación	16
2.2 Caso de estudio 2	17
2.2.1. Introducción de caso de estudio	17
2.2.2. Variables del caso de estudio	17
2.2.3. Algoritmos	18
- K-means	18
- Birch	19
- DBSCAN	20
- AffinityPropagation	21
- SpectralClustering	24
2.2.4. Interpretación de la segmentación	27
2.3 Caso de estudio 3	29
2.3.1. Introducción de caso de estudio	29
2.3.2. Variables del caso de estudio	29
2.3.3. Algoritmos	30
- K-means	31
- Birch	34
- DBSCAN	36
- AffinityPropagation	39
- AgglomerativeClustering	40
2.3.4. Interpretación de la segmentación	42
4. Referencias	43

Introducción

Para la segunda práctica vamos a usar técnicas de aprendizaje no supervisado para el análisis empresarial. Usaremos un conjunto de datos sobre los que aplicaremos algoritmos de clustering.

La base de datos que vamos a usar para esta práctica son microdatos del 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 tiene 83.499 casos y 142 variables.

Para la realización de la práctica vamos a escoger tres casos a los cuales vamos a aplicarles 5 algoritmos. Cada caso tendrá un conjunto filtrado de los datos que necesitamos, es decir, si queremos coger personas mayores de edad, antes de trabajar con el conjunto de datos filtramos para que la edad sea mayor que 18. Por último escogeremos las variables más relevantes para el caso que hemos escogido.

Los algoritmos que vamos a usar son:

- K-means
- Birch
- DBSCAN
- AgglomerativeClustering
- AffinityPropagation
- SpectralClustering
- MeanShift

Por último los tres casos de estudio. En el primero vamos a ver la situación de parejas que tiene un hijo menor de 25 años y que viven en un edificio en buen estado, para saber cómo son los edificios. En el segundo caso compruebo para las mujeres a partir de 16 años que están paradas desde hace tiempo o buscando un primer trabajo, y entonces saber el tipo de estudios que tiene y el número de hijos a su cargo. Y el último caso es para gente que no disponga de internet, pero tenga línea telefónica y el estado este en buen edificio.

Casos de estudio

Para esta práctica vamos a realizar tres casos de estudios usando distintas variables para agruparlas. Para cada uno de los casos se han escogido dos algoritmos para ser explicados. Las medidas que vamos a usar son: Número de clúster, índice Calinski-Harabaz, la métrica Silhouette, y el tiempo que se tarda en ejecutar el algoritmo.

Para visualizar los datos se van a usar: Dendrogramas, Heatmap, scatter-matrix y tablas con los datos para poder diferenciar las medidas.

Caso de estudio 1

- Introducción de caso de estudio

El primer caso de estudio se basaría en buscar en qué tipo de edificios viven las parejas que tenga la menos un hijo de menos de 25 años. Con este caso queremos observar si hay una gran diferencia entre los tipos de edificios y sobre todo saber el año de construcción.

- Variables del caso de estudio

- Categóricas:
 - o ESTADO: Estado del edificio. Para el caso vamos a querer que el estado del edificio sea bueno (por lo tanto el 'ESTADO'==4). Esta variable lo usaremos como un filtro inicial.
 - o ESTHOG: Estructura del hogar. Queremos solo fijarnos en pareja que tenga un hijo menor de 25 años (la variable quedaría 'ESTHOG'==8). Esta variable también se usa como filtro inicial.
- Numéricas:
 - o EDAD: Edad que tiene cada persona. Queremos que las personas sean mayores de edad para realizar este estudio (sería 'EDAD'>17). Esta es la última variable que se va a usar como filtro para el caso de estudio.
 - o ANOCOSN: Año de construcción del edificio.
 - o PLANTAS: Numero de plantas que tiene el edificio
 - o SUT: Superficie útil de la vivienda.
 - o NHAB: número de habitaciones que hay en la vivienda.

- Algoritmos

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means 3	3	15904.920	0.45934	0.09
k-Means 5	5	12960.183	0.32887	0.2
k-Means 8	8	11924.172	0.32965	0.44
Birch	5	8353.099	0.42294	0.42
DBSCAN	14	1691.004	0.09375	3.80
AgglomerativeClustering	5	11271.131	0.31202	24.61
AffinityPropagation d=0.93	3	196.001	0.14629	1.77
AffinityPropagation d=0.919	3	196.001	0.14629	1.77
AffinityPropagation d=0.9	69	516.882	0.32310	13.65

K-means

*****	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means 3	3	15904.920	0.45934	0.09
k-Means 5	5	12960.183	0.32887	0.2
k-Means 8	8	11924.172	0.32965	0.44

`k_means = KMeans(init='k-means++', n_clusters=N_Cluster, n_init=5)`

Para el K-means he decidido variar las características para hacer una comparación cambiando el numero de clúster para ver si daba un índice Calinski-Harabaz y métrica Silhouette distinta. Para ello he variado el número de clúster entre 3, 5 y 8.



Figura 1 Heatmap K-means con 3 clúster



Figura 2 Heatmap K-means con 5 clúster

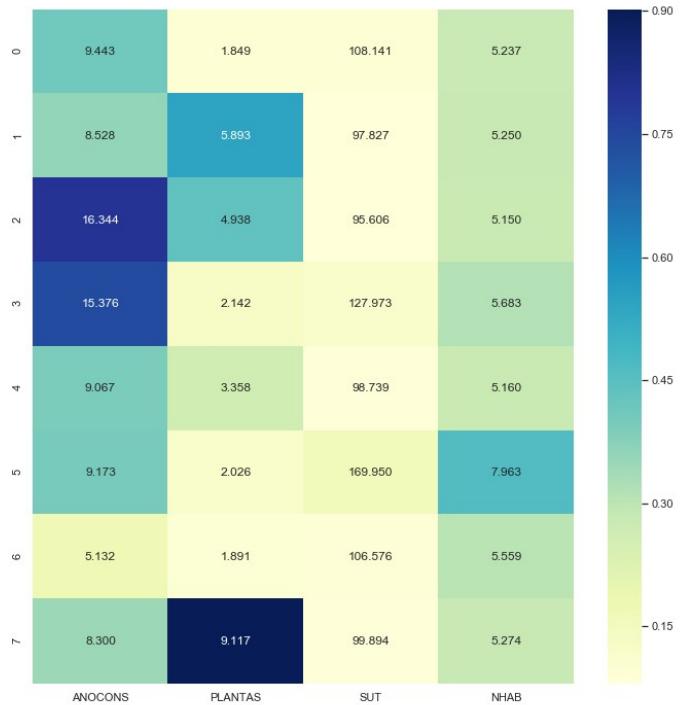


Figura 3 Heatmap K-means con 8 clúster

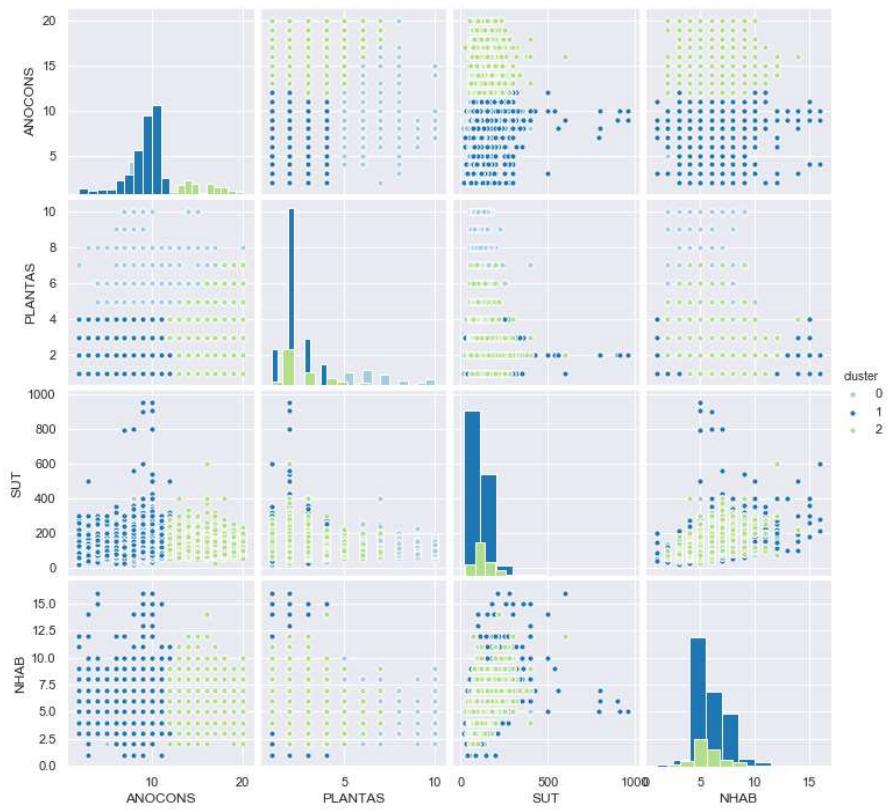


Figura 4 Scatter matrix K-means con 3

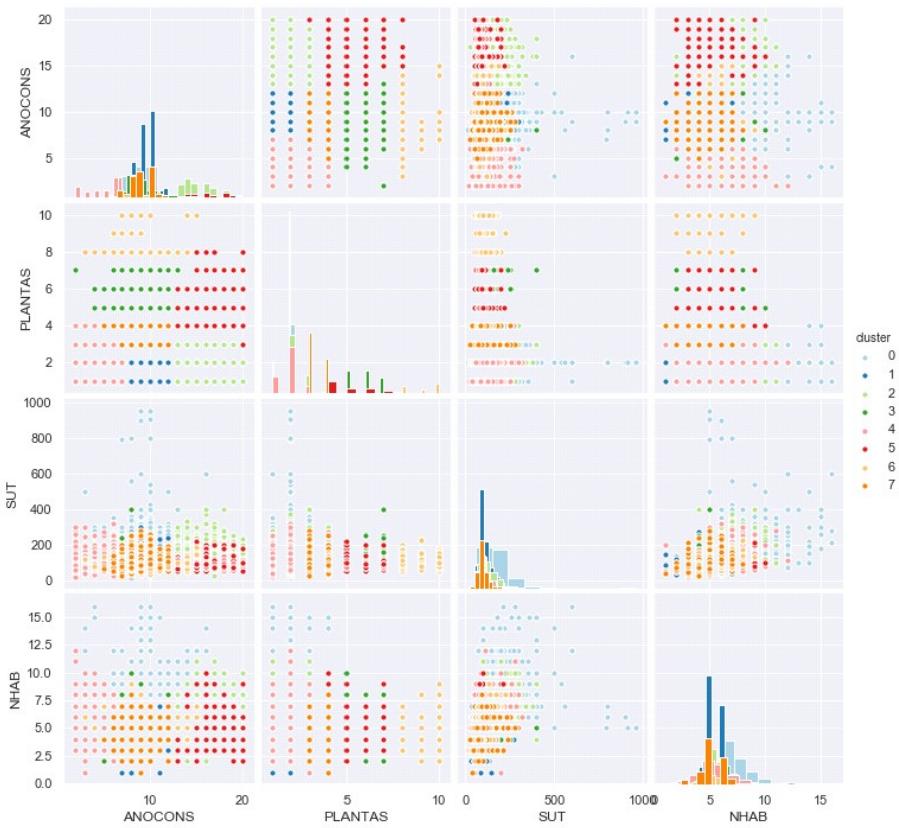


Figura 5 Scatter matrix K-means con 5

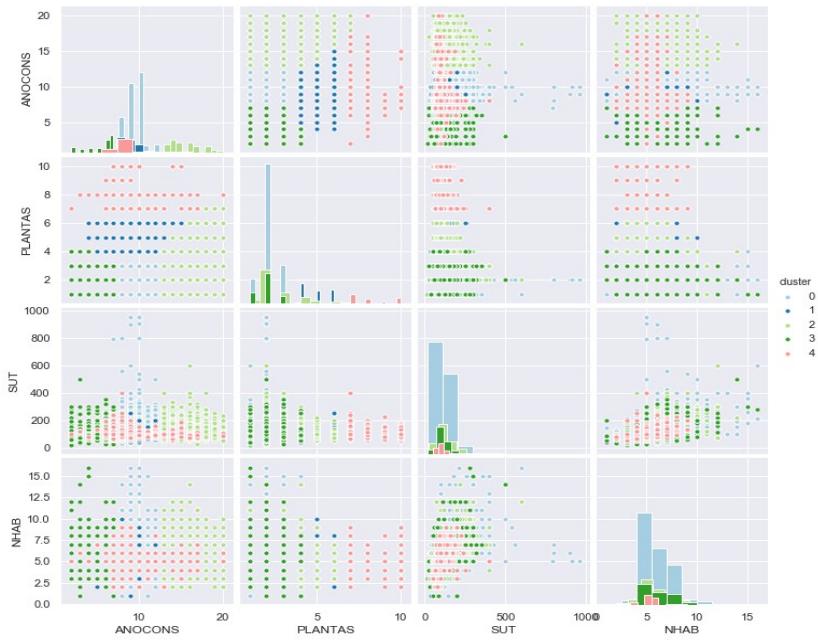


Figura 6 Scatter matrix K-means con 8

Birch

*****	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
Birch	5	8353.099	0.42294	0.42

Birch(threshold=0.2, branching_factor=50, n_clusters=5, compute_labels=True, copy=True)

Para realizar el heatmap en el algoritmo birch al no tener los centros como una variable lo que he hecho es sacar la media de las variables del clúster y mostrarla como centro. De esa forma aunque no sea realmente un heatmap, se pueden sacar bien las conclusiones.

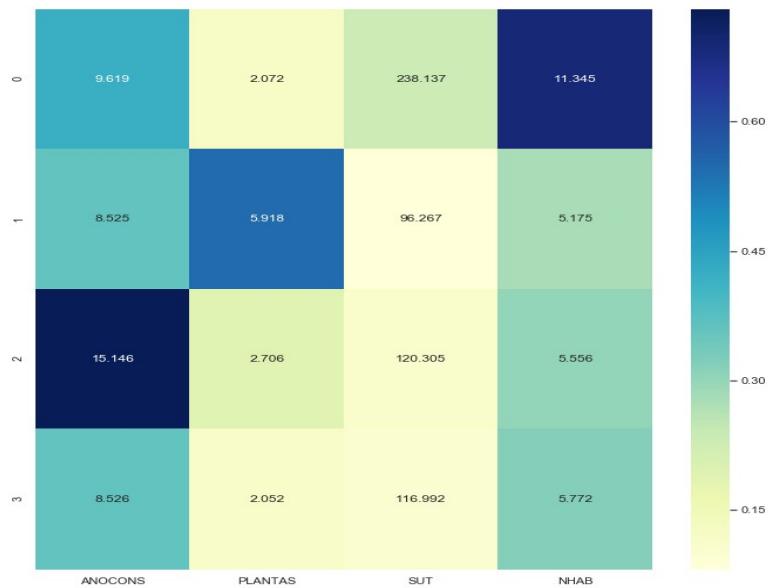


Figura 7 Heatmap Birch

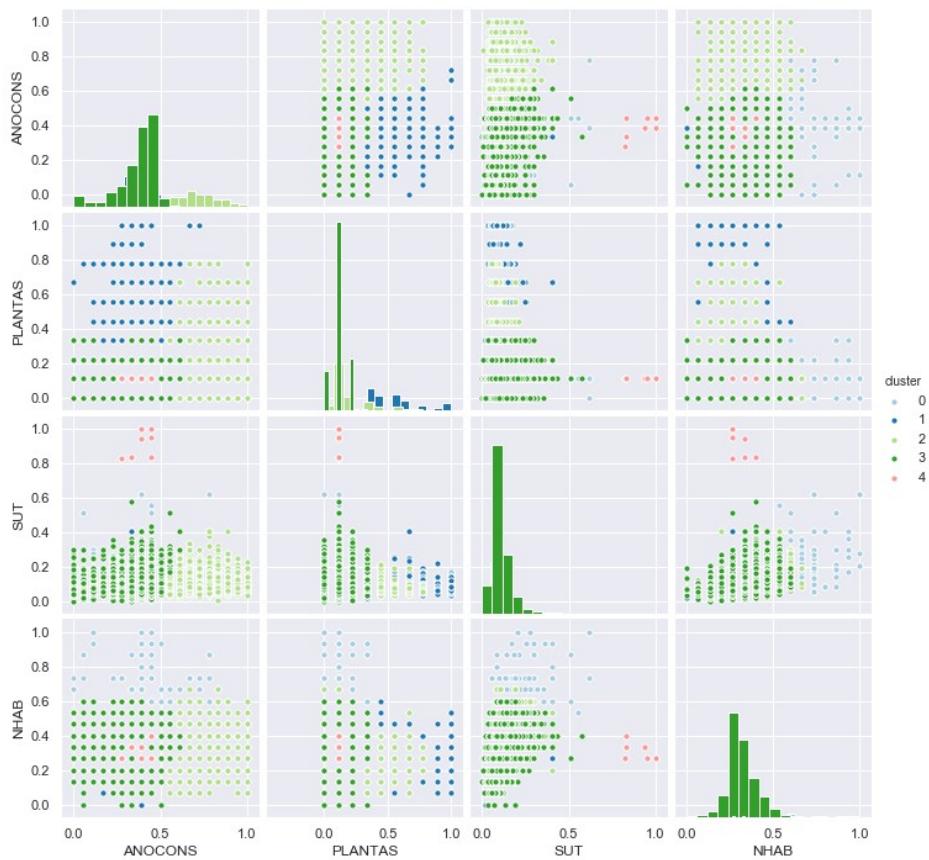


Figura 8 Scatter matrix Birch

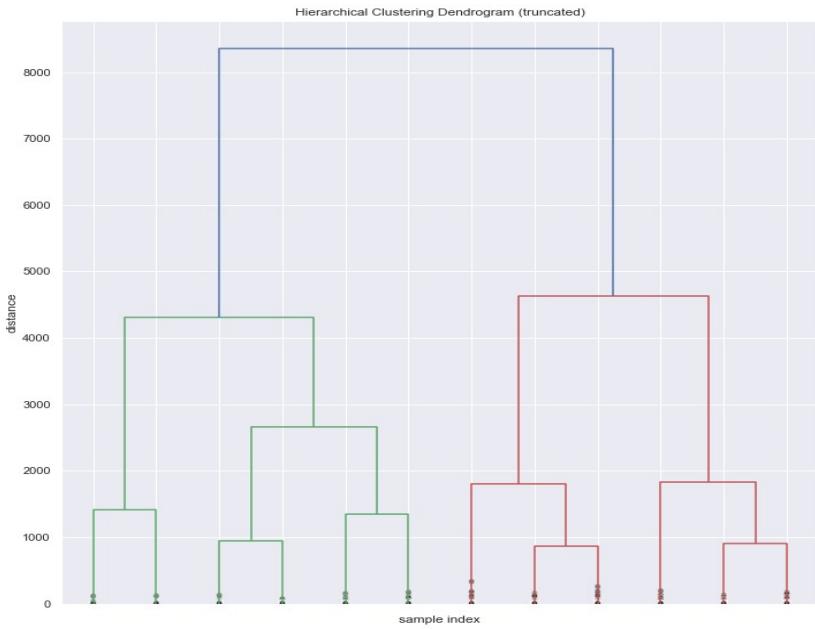


Figura 9 Dendrograma Birch

DBSCAN

*****	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
DBSCAN	14	1691.004	0.09375	3.80

DBSCAN(eps=0.111, min_samples=15, metric='euclidean', metric_params=None, algorithm='auto', leaf_size=30, p=None, n_jobs=None)

Para el DBSCAN tambien he usado las madias para hacer el heatmap.

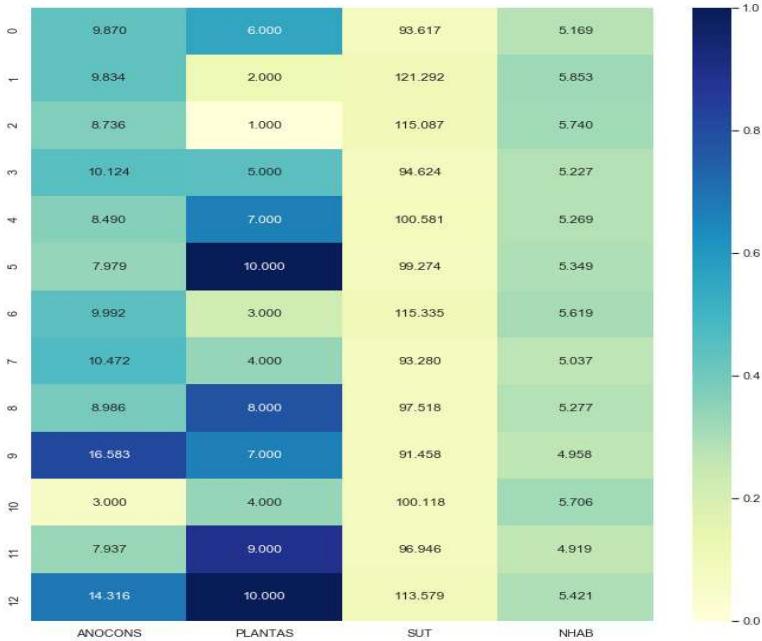


Figura 10 Heatmap DBSCAN

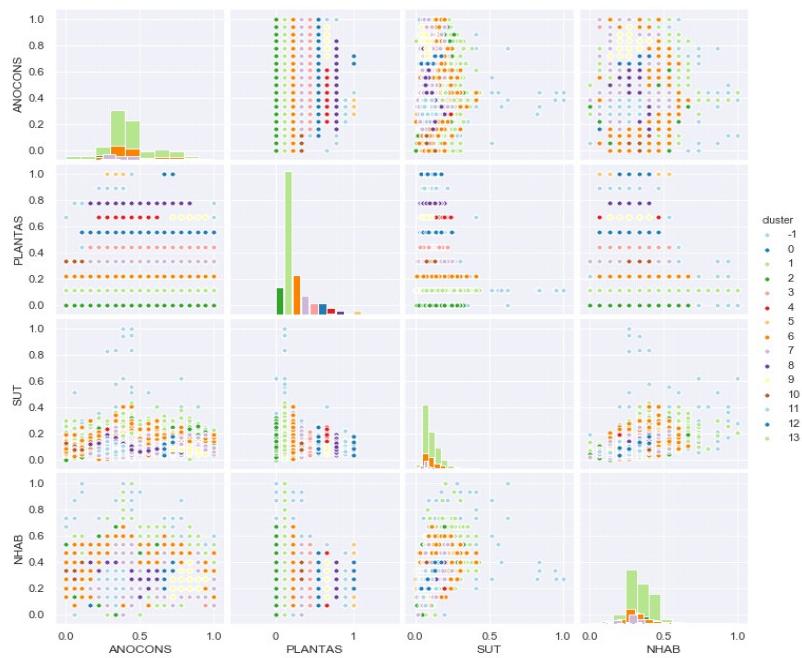


Figura 11 Scatter DBSCAN

AgglomerativeClustering

*****	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
AgglomerativeClustering	5	11271.131	0.31202	24.61

```
AgglomerativeClustering(n_clusters=5, affinity='euclidean', memory=None, connectivity=None,
compute_full_tree='auto', linkage='ward', pooling_func='deprecated')
```

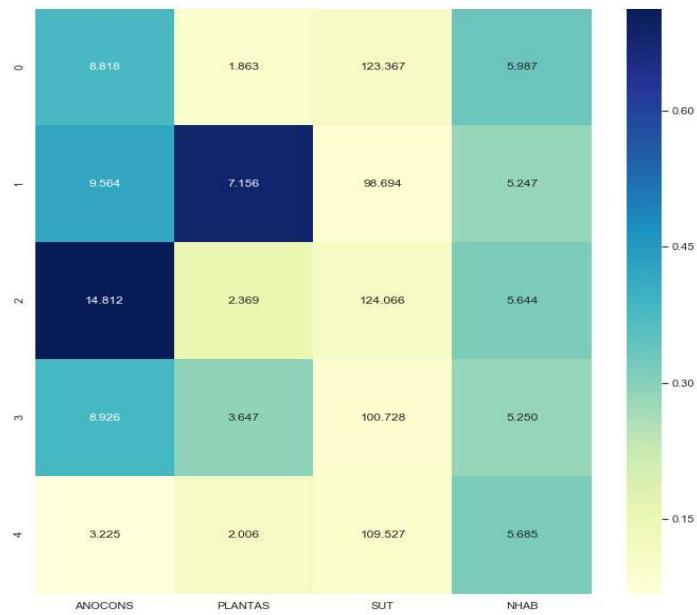


Figura 12 Heatmap AgglomerativeClustering

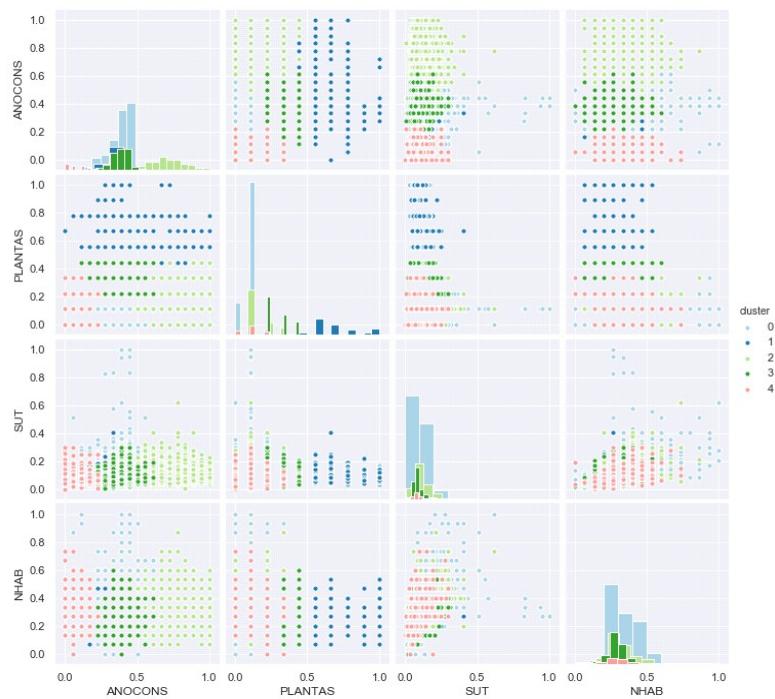


Figura 13 Scatter matrix AgglomerativeClustering

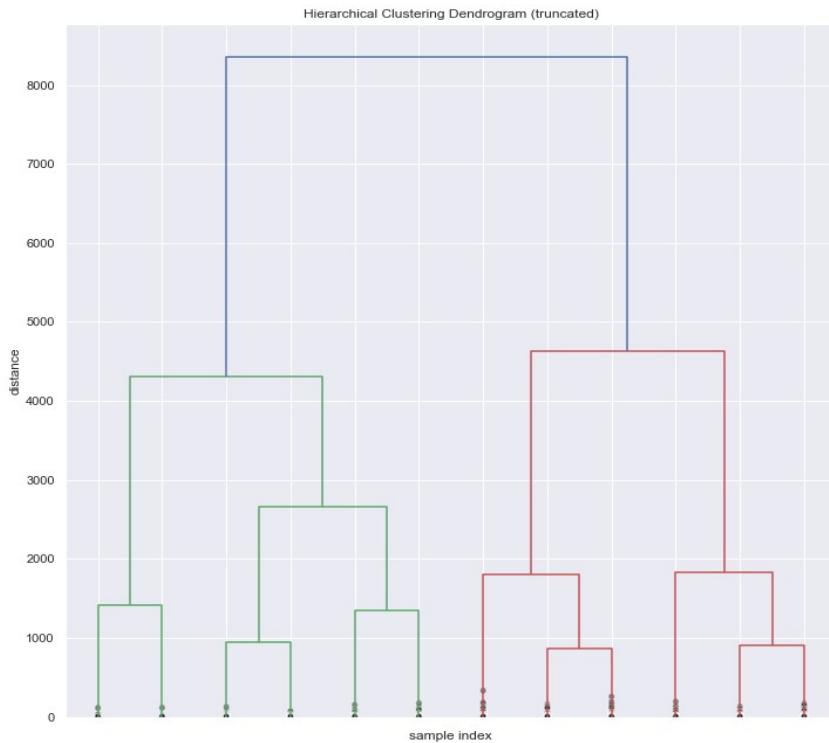


Figura 14 Dendrograma AgglomerativeClustering

AffinityPropagation

*****	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
AffinityPropagation d=0.93	3	196.001	0.14629	1.77
AffinityPropagation d=0.919	3	196.001	0.14629	1.77
AffinityPropagation d=0.9	69	516.882	0.32310	13.65

```
AFC = AffinityPropagation(damping=d, max_iter=200, convergence_iter=15, copy=True,
preference=None, affinity='euclidean', verbose=False);
```

Para el AffinityPropagation también he variado la variable damping que se encarga de juntar los datos y sacar el numero de clúster. En este caso lo he puesto en 0.93, 0.919 y 0.9. Podemos ver que no hay diferencia entre 0.93 y 0.919 en cambio el 0.9 hace que la distancia disminuya y se cree una mayor cantidad de clúster.

Para este algoritmo he estado intentando variar el damping de forma que no hubiera un gran cambio. Pero da salto muy grandes incluso trabajando con milésimas. Así que he decidido mostrarlos con los siguientes valores.

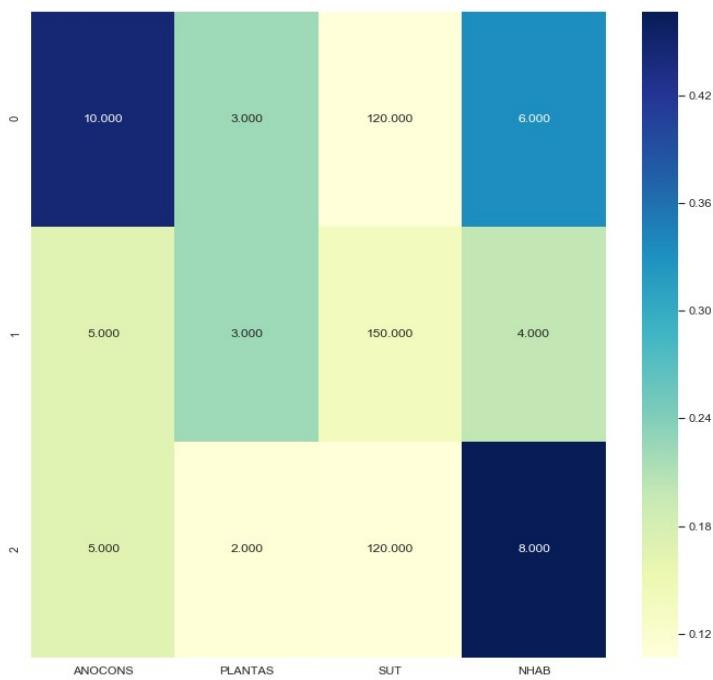


Figura 15 Heatmap AffinityPropagation con $d= 0.919$ y $d=0.93$

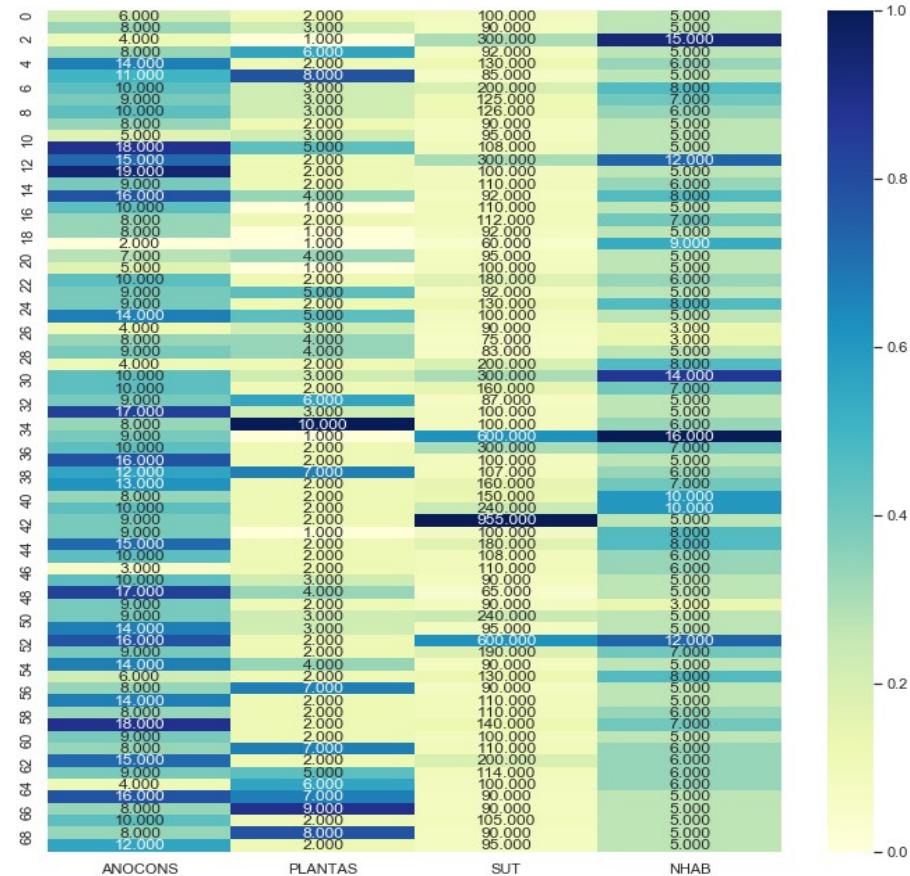


Figura 16 Heatmap AffinityPropagation con $d=0.9$

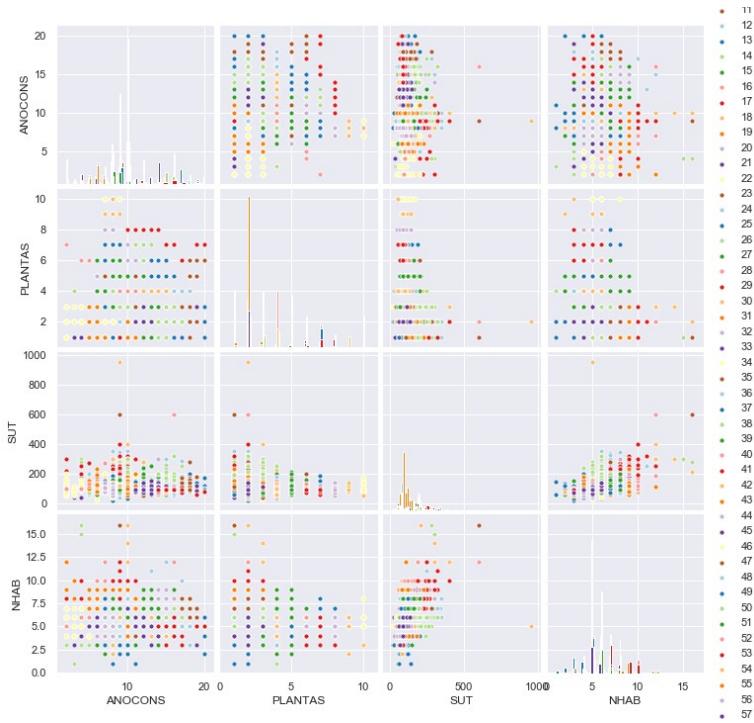


Figura 17 Scatter matrix AffinityPropagation con $d= 0.919$ y $d=0.93$

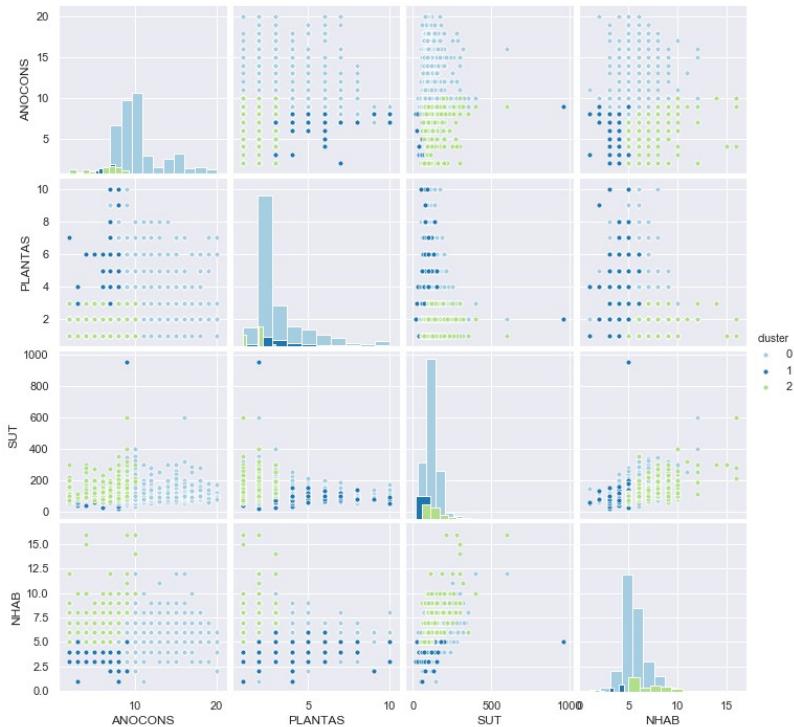


Figura 18 Scatter matrix AffinityPropagation con $d=0.9$

- Interpretación de la segmentación

Para este caso se han tomado una muestra de 23620 personas. Para el algoritmo de AffinitiPropagation se ha reducido el número de datos de la muestra a otra más pequeña de 2000 datos. Esto es debido a la complejidad del algoritmo a la hora de ejecutarse que es de $O(N^2)$.

En primer lugar nos podemos fijar en la tabla que hemos sacado con las distintas medidas. En la que podemos observar como K-means obtiene los mejores índices de Calinski-Harabaz y en especial el mejor Silhouette en el que solo se usan 3 clúster para este caso. Simplemente viendo estos datos y sin tener en cuenta el tiempo, podemos decir que es el algoritmo de clustering mas eficiente trabajando con pocos clúster. Por otro lado podemos observar que el algoritmo birch y el AgglomerativeClustering son otros también muy buenos. La diferencia es que el birch no tiene el índice de Calinski-Harabaz tan alto por lo tanto las agrupaciones no son tan buenas, y el AgglomerativeClustering tarda mucho tiempo en ejecutarse con respecto a los otros. Por último podemos ver que tanto DBSCAN como AffinitiPropagation no tienen un buen índice. Con esto podemos sacar en claro que las para sacar las conclusiones sobre el caso es mejor fijarse en el K-means, birch y AgglomerativeClustering.

Clústeres que voy a usar para el análisis de los datos.

Nombre algoritmo	N_cluster	N_datos
K-means 3	3	0 : 2997
		1 : 16777
		2 : 3846
K-means 8	8	0 : 7660
		1 : 2150
		2 : 835
		3 : 2954
		4 : 3393
		5 : 3349
		6 : 2542
		7 : 737
Birch	5	0 : 307
		1 : 4056
		2 : 4328
		3 : 14910
		4 : 19

Si nos fijamos en la figura 1 podemos comprobar que hay un bloque principal que es el clúster 0. Que se podría describir como Personas que viven en bloques de pisos de muchas plantas. Podemos observar también que la forma principal de dividir los clúster ha sido por fecha de construcción y después por el número de plantas, ya que el tanto el número de habitaciones con el de la superficie útil es muy similar. Para ver el clúster 2 vamos a hacer referencia también a los clúster 2 y 3 de la figura 3. Que es un sub esquema si lo dividiésemos en mas clúster, en los que podemos ver que la casa que fueron construidas hace meno tiempo hay diferencia entre el número de plantas que tienen aunque como se puede apreciar hay menos pisos nuevos con mayor número de plantas.

En general todos los clúster muestran los mismos datos. Pero hay dos que han sacado algunos datos que a mí me han parecido más llamativos aunque son casos aislados. En el caso del clúster 4 de la figura 7 hay 19 personas que viven en casas en las que hay un gran número de habitaciones y de superficie útil es mayor. Se podría decir que son casas más antiguas dirigidas a familias con una gran cantidad de hijos. También hay un caso extraños en el clúster 41 de la figura 16. En este caso El terreno útil es excesivamente grande. Este caso se da para 21 personas en la muestra.

Caso de estudio 2

- Introducción de caso de estudio

Para este segundo caso de estudio queremos determinar el número de mujeres que están paradas y las posibles razones por la que lo están. Para eso vamos a usar el tipo de estudios que tienen, la edad y el número de hijos para ver si hay algún tipo de relación

- Variables del caso de estudio

- Categóricas:
 - SEXO: Determina el sexo de la persona. Para este caso vamos a querer que sean solo mujeres (por lo tanto el 'SEXO ==6). Esta variable lo usaremos como un filtro inicial.
 - RELA: Relación con la actividad. Queremos solo fijarnos en mujeres que estén paradas o que estén buscando trabajo por primera vez (la variable quedaría 'RELA'>1 y 'RELA'<4). Esta variable también se usa como filtro inicial.
- Numéricas:
 - EDAD: Edad que tiene cada persona. Queremos que las personas puedan trabajar y para ello tienen que tener más de 16 años (sería 'EDAD'>16). Esta es la última variable que se va a usar como filtro para

el caso de estudio. Pero también la vamos a usar para poder diferenciar las edades dentro del análisis.

- ESREAL: Nivel de estudios completados.
- NIHIOS: Numero de hijos de la mujer.

- Algoritmos

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means	5	6347.484	0.32085	0.09
Birch	5	6224.584	0.31873	0.16
DBSCAN	11	1078.081	0.01127	0.44
AgglomerativeClustering con complete	5	2257.523	0.22633	2.57
AgglomerativeClustering con ward	5	5134.067	0.23529	2.30
AgglomerativeClustering con average	5	1849.065	0.33944	3.17
SpectralClustering con 5 cluster y 5 n_neighbors	5	4167.344,	0.26767	12.70
SpectralClustering con 5 cluster y 10 n_neighbors	5	4211.711	0.27130	12.82
SpectralClustering con 8 cluster y 10 n_neighbors	8	3597.268	0.21866	13.49

K-means

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means	5	6347.484	0.32085	0.09

`KMeans(init='k-means++', n_clusters=5, n_init=5)`

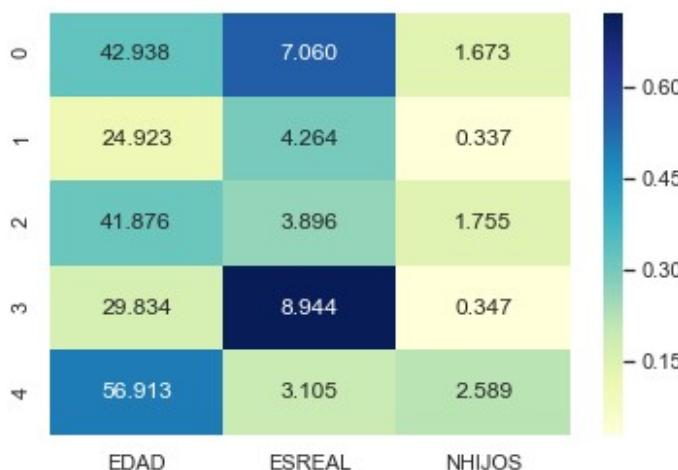


Figura 19 Heatmap K-means

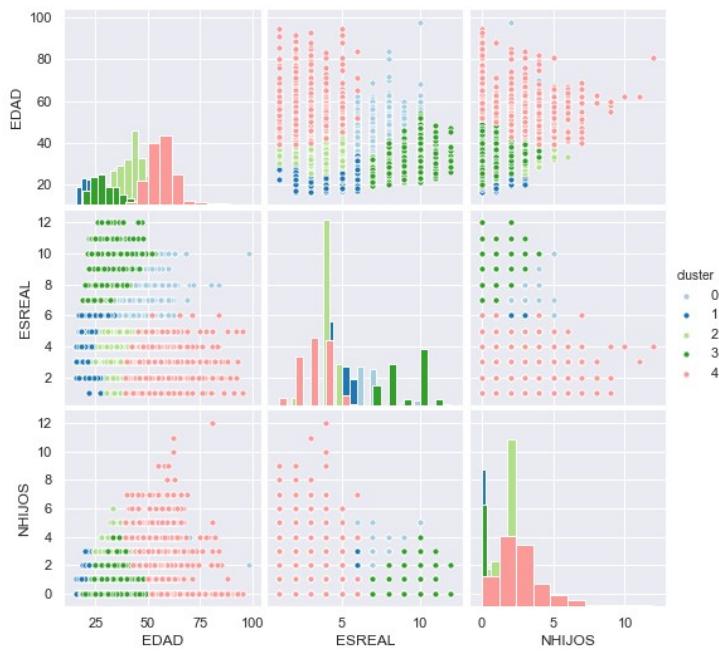


Figura 20 Scatter matrix K-means

Birch

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
Birch	5	6224.584	0.31873	0.16

Birch(threshold=0.2, branching_factor=50, n_clusters=5, compute_labels=True, copy=True)

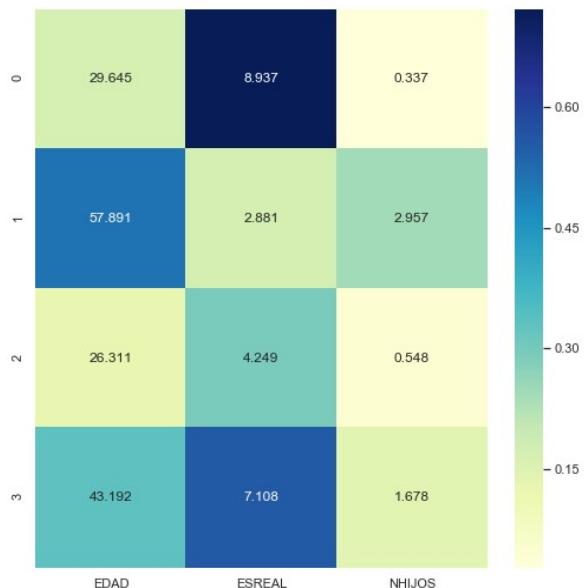


Figura 21 Heatmap Birch

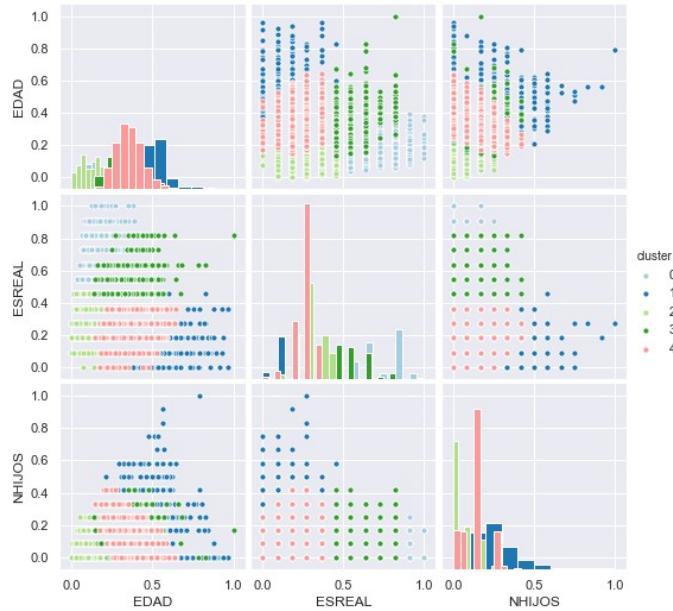


Figura 22 Scatter matrix Birch

DBSCAN

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
DBSCAN	11	1078.081	0.01127	0.44

```
DBSCAN(eps=0.09, min_samples=15, metric='euclidean', metric_params=None,
algorithm='auto', leaf_size=30, p=None, n_jobs=None)
```

Igual que en el caso anterior he estado intentado ajustar el eps para que de una cantidad de de clúster razonable. Se ha quedado en 0.09 ya que con el cambio de las milésimas seguía sin variar.

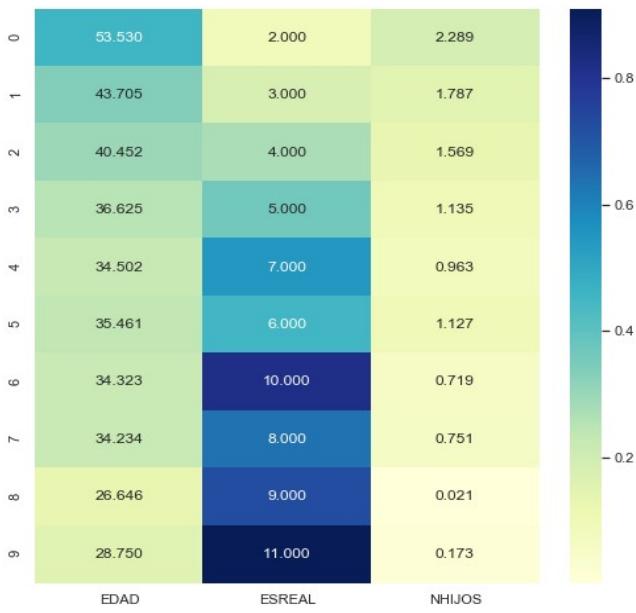


Figura 23 Heatmap DBSCAN

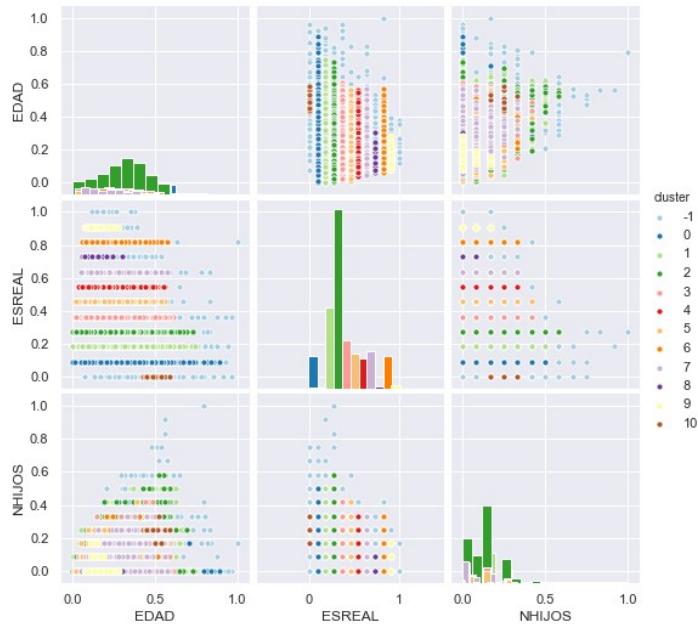


Figura 24 Scatter matrix DBSCAN

AgglomerativeClustering

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
AgglomerativeClustering con complete	5	2257.523	0.22633	2.57
AgglomerativeClustering con ward	5	5134.067	0.23529	2.30
AgglomerativeClustering con average	5	1849.065	0.33944	3.17

```
AgglomerativeClustering(n_clusters=5, affinity='euclidean', memory=None, connectivity=None,
compute_full_tree='auto', linkage=TipoDeLink, pooling_func='deprecated')
```

Este es el primer algoritmo que he querido hacer cambios para este caso para comprobar si mejoraba la calidad de los cluster. En este caso he escogido variar la forma de hacer link entre los datos de la muestra. Las tres formas serian ‘complete’, ‘ward’ y ‘average’.

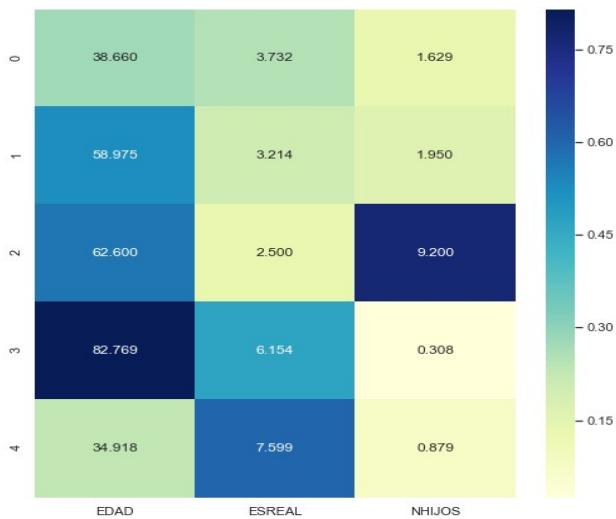


Figura 25 Heatmap AgglomerativeClustering con complete

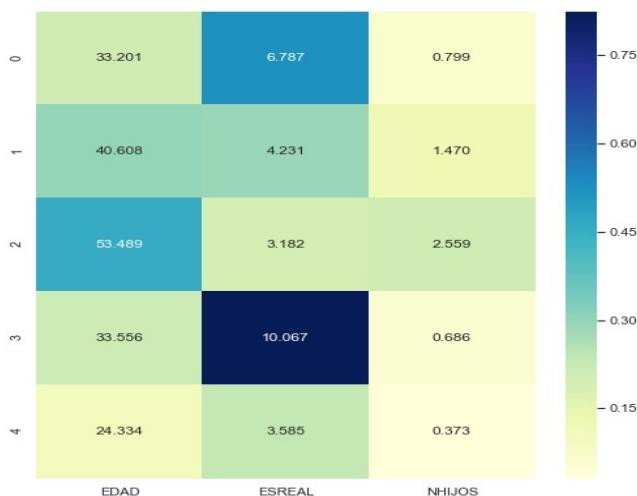


Figura 26 Heatmap AgglomerativeClustering con ward

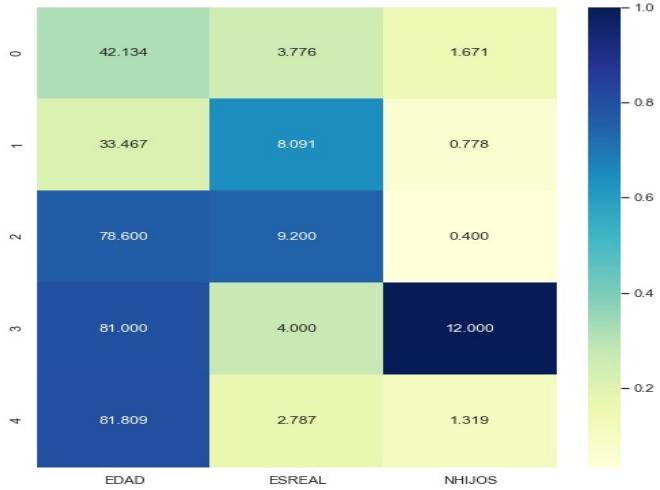


Figura 27 Heatmap AgglomerativeClustering con average

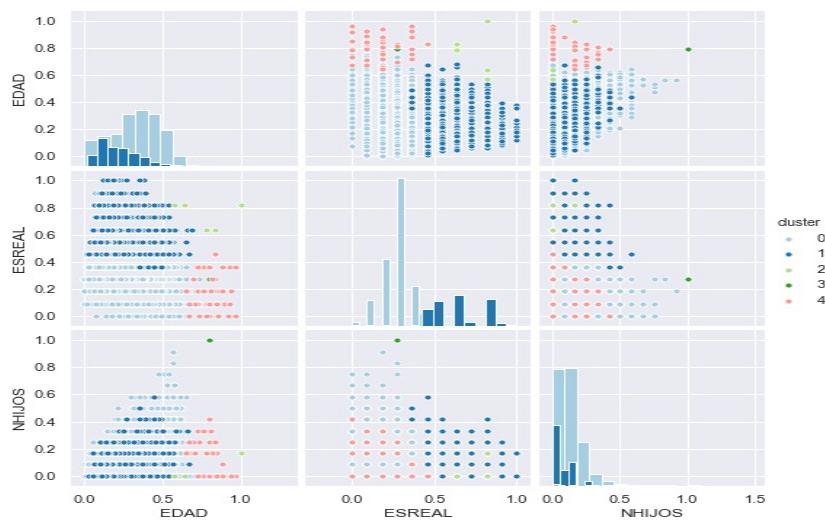


Figura 28 Scatter matrix AgglomerativeClustering con complete

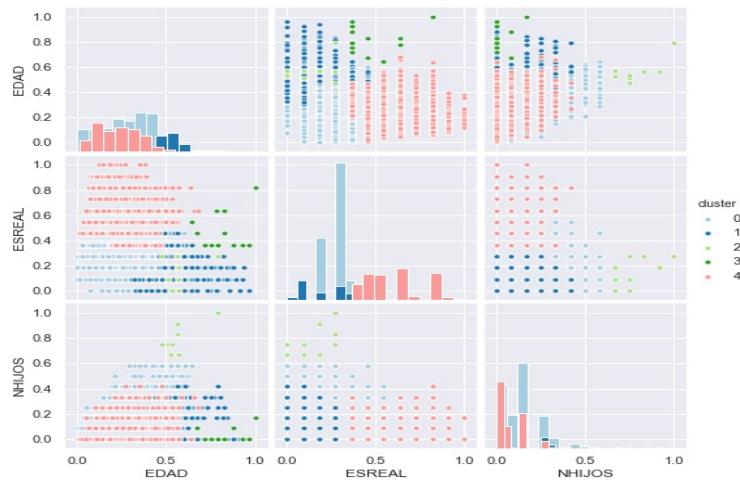


Figura 29 Scatter matrix AgglomerativeClustering con ward

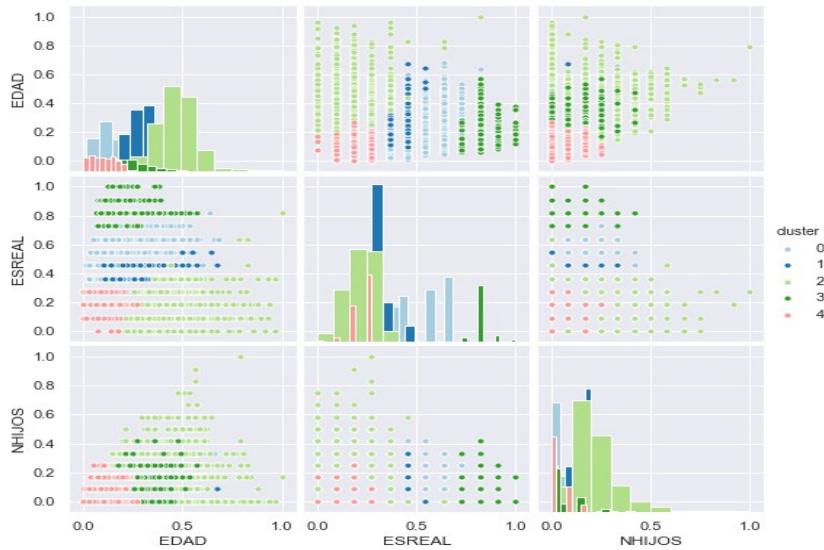


Figura 30 Scatter matrix AgglomerativeClustering con average

SpectralClustering

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
SpectralClustering con 5 cluster y 5 n_neighbors	5	4167.344,	0.26767	12.70
SpectralClustering con 5 cluster y 10 n_neighbors	5	4211.711	0.27130	12.82
SpectralClustering con 8 cluster y 10 n_neighbors	8	3597.268	0.21866	13.49

*SpectralClustering(n_clusters=**N_Cluster**, eigen_solver=None, random_state=None, n_init=20, gamma=1.0, affinity='rbf', n_neighbors=**n_neighbors**, eigen_tol=0.0, assign_labels='kmeans', degree=3, coef0=1, kernel_params=None, n_jobs=None)*

Este ha sido el último algoritmo y también he decidido hacer las modificaciones para comprobar los resultados que daba. En este caso ha dado buenos resultados comparados con los otros. El cambio que he decidido hacer es el numero de cluster y el numero de vecinos.

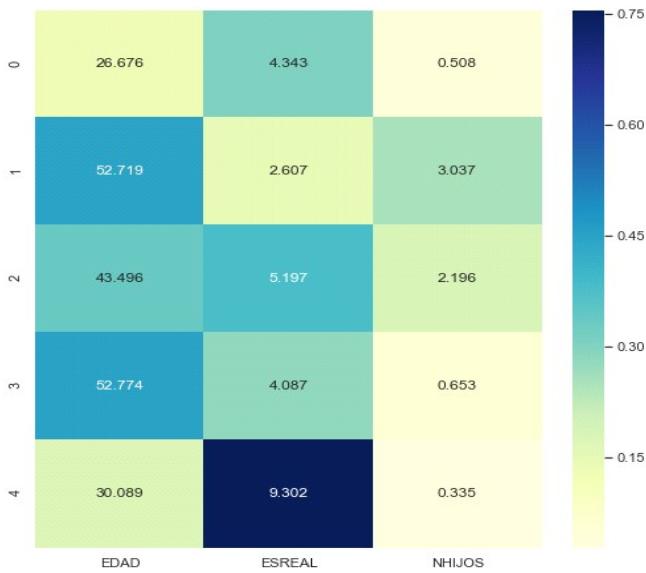


Figura 31 Heatmap SpectralClustering con 5 cluster y 5 n_neighbors

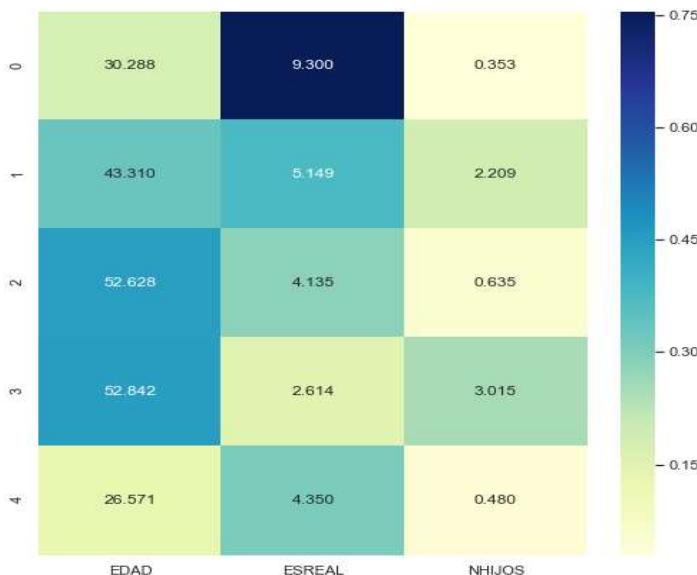


Figura 32 Heatmap SpectralClustering con 5 cluster y 10 n_neighbors

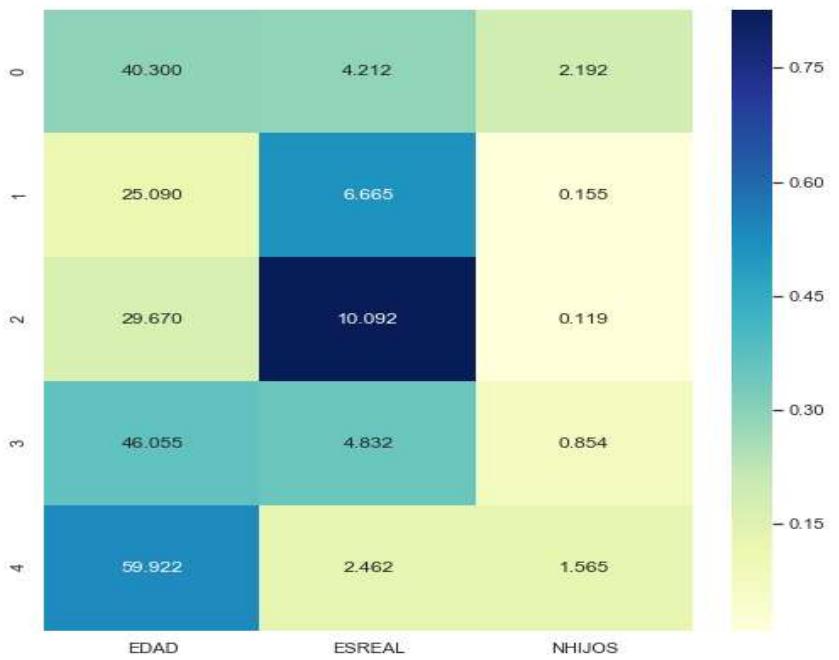


Figura 33 Heatmap SpectralClustering con 8 cluster y 10 n_neighbors

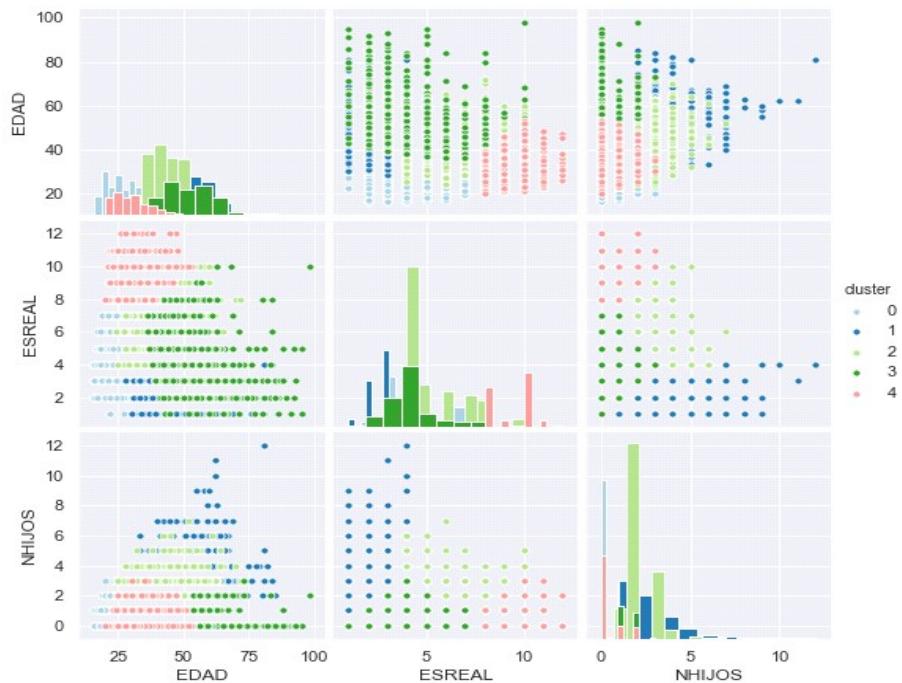


Figura 34 Scatter matrix SpectralClustering con 5 cluster y 5 n_neighbors

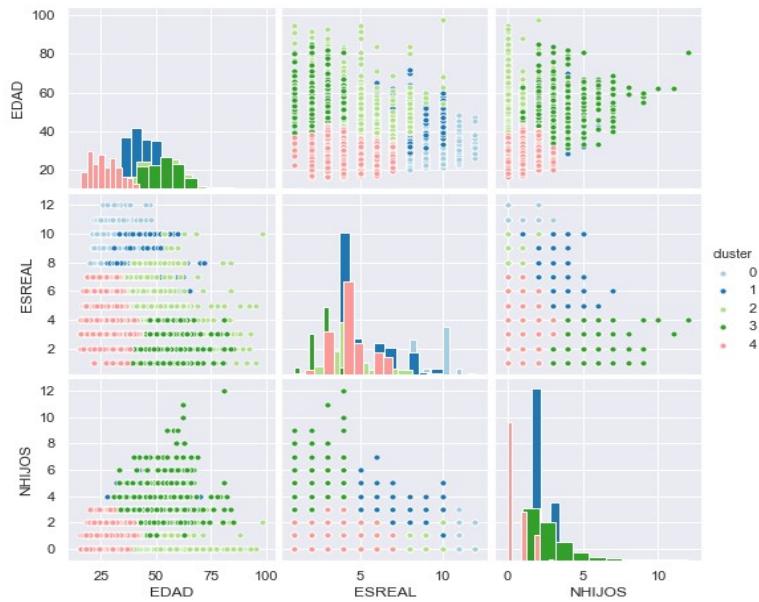


Figura 35 Scatter matrix SpectralClustering con 5 cluster y 10 n_neighbors

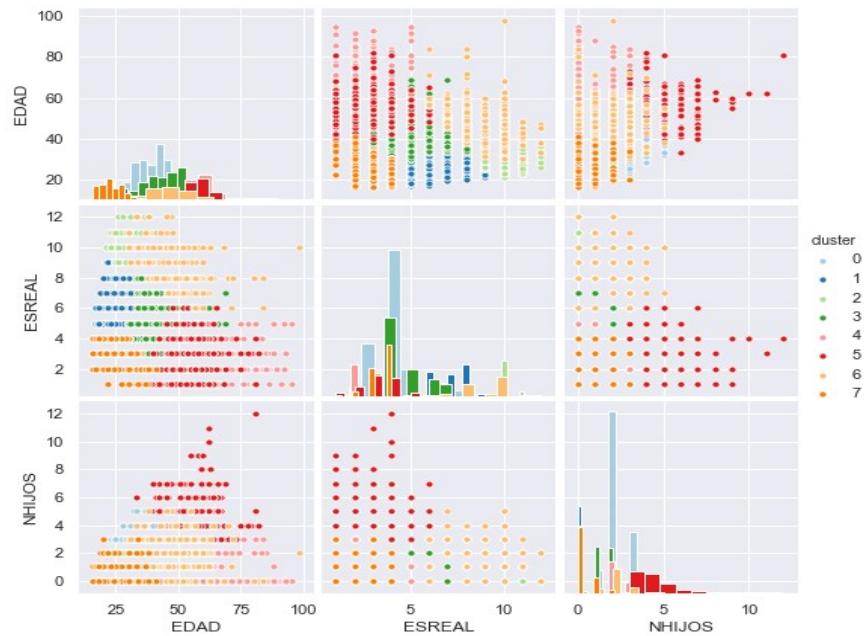


Figura 36 Scatter matrix SpectralClustering con 8 clúster y 10 n_neighbors

- Interpretación de la segmentación

Para este caso la muestra que ha quedado después de hacer el filtrado es de 9078 datos. Por lo tanto para este caso no ha sido necesario hacer una reducción de datos para el algoritmo de clúster SpectralClustering.

Como se puede observar en la tabla de este caso esta vez la mayoría de los algoritmos dan unos resultados muy similares tanto en el índice de Calinski-Harabaz,

como en el Silhouette. El único que se ha diferenciado es el DBSCAN dando los peores resultados. Pero esto se puede deber a la cantidad de clúster que genera. También nos podemos fijar es que como en el caso anterior el K-means da los mejores resultados pegado al birch.

Por otro lado podemos observar que el algoritmo AgglomerativeClustering da mejor resultado en el caso en el que se usa la forma de hacer link que viene por defecto. Sobre todo el ‘average’ da un mal resultado. Se podría decir que hay diferencias grandes a la hora de los resultados dependiendo del tipo link que usemos.

En cambio en el algoritmo SpectralClustering al hacer los cambios no ha habido unas grandes diferencias. Sobre todo parece que el número de clúster hace que afecte más al algoritmo. Lo más llamativo es que es el algoritmo de clustering que más tiempo tarda en ejecutarse.

Como he hecho en el caso anterior voy a tomar los clúster más representativos para realizar la conclusión, que son:

Nombre algoritmo	N_cluster	N_datos
K-means	5	0 : 1089
		1 : 1850
		2 : 2869
		3 : 1293
		4 : 1977
AgglomerativeClustering con ward	5	0 : 2113
		1 : 2432
		2 : 2684
		3 : 719
		4 : 1130
SpectralClustering con 5 cluster y 10 n_neighbors	5	0 : 1057
		1 : 2986
		2 : 1269
		3 : 1367
		4 : 2399

Si comprobamos la figura 19 podemos ver que como se muestra en la tabla de los clúster del algoritmo k-means los grupos están muy igualados en cantidad de datos por clúster. Podemos ver los grupos con edades más jóvenes como el clúster 1 y 3 que nos muestran mujeres que posiblemente estén buscando su primer trabajo porque todavía se están formando. Esta conclusión se puede sacar por la diferencia de edad que hay entre ellos. También podemos ver que las mujeres jóvenes paradas o estudiando todavía no tienen realmente intención de tener hijos. Por otro lado podemos observar 3 grupos con edad más superior el grupo 4 y 2 se pueden tomar como grupos que han sido afectados por el tipo de sociedad que había en esa época.

Además, son los dos grupos más numerosos. También lo podemos notar porque el grupo 0 es menor, que se podría considerar como mujeres superiores a los 40 años con un nivel de estudios más alto. Por lo tanto podemos observar un cambio en la sociedad viendo que no hay tanta diferencia entre la cantidad de estudiantes que de ahora y de mujeres sin estudios. Es decir, hay más mujeres jóvenes que han podido estudiar en comparación con antes.

Podemos fijarnos también en que en los algoritmos de SpectralClustering y AgglomerativeClustering el valor al que más prioridad le ha dado ha sido al número de hijos que tienen las personas. Sobre todo si nos fijamos en el AgglomerativeClustering los dos grupos con más datos son el 2 y el 1, que corresponden a los grupos con mayor edad pero con más hijos. Aun así en este algoritmo también podemos observar que cuanto más joven es la mujer más tiende a no haber tenido hijos todavía y sobre todo a tener más estudios. El caso más llamativo de este algoritmo es el 3 ya que son 719 mujeres con un nivel de estudios muy alto hasta el momento. En el algoritmo de SpectralClustering también podemos ver como el grupo 0 tiene también una gran cantidad de estudios. También en los dos clúster existe un grupo más joven que parece que todavía se está formando.

Como conclusión se podría decir que se ha producido un cambio en la sociedad en el paso de los años. Por el cual las mujeres más jóvenes se forman más que hace 50 años. Y debido a la formación tienen una menor cantidad de hijos.

Caso de estudio 3

- Introducción de caso de estudio

Para el último caso he decidido basarlo en la creación de perfiles para una empresa que de servicio de internet. Para ello queremos saber de personas que no tenga ningún servicio de internet pero que tengan línea telefónica en el edificio y que este en buen estado el edificio. Así se facilitaría la instalación de la red. Una vez creado los perfiles se puede hacer publicidad personalizada para cada uno de ellos. También vamos a usar los niños de menor edad y los jóvenes como incentivos para que contraten el servicio de red. Por último también se va mirar la superficie útil de donde viven para saber qué tipo de red hay que instalar.

- Variables del caso de estudio

- Categóricas:
 - o INTERNET: Determina si la persona tienen internet o no. Para este caso solo queremos personas que no dispongan de internet (por lo tanto el 'INTERNET ==2'). Esta variable lo usaremos como un filtro inicial.

- RELA: Relación con la actividad. Queremos crear perfiles para todos los tipos de personas posibles, pero he decidido quitar el caso 6 por la ambigüedad que tenia. (la variable quedaría 'RELA'<6). Esta variable también se usa como filtro inicial.
 - ESTADO: Estado del edificio. Queremos que los edificios estén en buen estado para que no haya problemas con la instalación (por lo tanto el 'ESTADO'>3). Tambien se usara como filtro.
 - TELEF: Tendido telefónico. También queremos que se tengan instalado el tendido telefónico por los mismo motivos que el anterior (por lo tanto el 'TELEF' ==2).
- Numéricas:
- EDAD: Edad que tiene cada persona. Queremos que las personas puedan contratar el servicio así que la edad será mayor de 18 (sería 'EDAD'>17). Esta es la última variable que se va a usar como filtro para el caso de estudio. Pero también la vamos a usar para poder diferenciar las edades dentro del análisis.
 - NOCU: Numero de personas ocupadas en el hogar.
 - NPARAIN: Número de parados o inactivos en el hogar.
 - H0515: Numero de personas en la casa entre 5 y 15 años.
 - H1624: Numero de personas en la casa entre 16 y 24 años.
 - SUT: Superficie útil de la vivienda.

- Algoritmos

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means 3	3	16633.160	0.42654	0.16
k-Means 5	5	13112.099	0.39091	0.27
k-Means 8	8	11430.116	0.40809	0.48
Birch	5	7082.637	0.35889	0.76
MeanShift bandwidth=0.45	5	4833.900	0.27573	0.45
MeanShift con estimate_bandwidth (X_normal, n_samples=300)	3	3181.497	0.27643	2.14
MeanShift con estimate_bandwidth(X_normal, n_samples=10000)	3	3181.497	0.27643	2.14
AgglomerativeClustering	5	11083.571	0.36803	26.57
SpectralClustering	5	922.479	0.36574	0.54

K-means

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
k-Means 3	3	16633.160	0.42654	0.16
k-Means 5	5	13112.099	0.39091	0.27
k-Means 8	8	11430.116	0.40809	0.48

*KMeans(init='k-means++', n_clusters= **N_Cluster**, n_init=5)*

Para el K-means he decidido variar las características para hacer una comparación como en el caso anterior cambiando el numero de clúster para ver si daba un índice Calinski-Harabaz y métrica Silhouette distinta. Para ello he variado el número de clúster entre 3, 5 y 8.

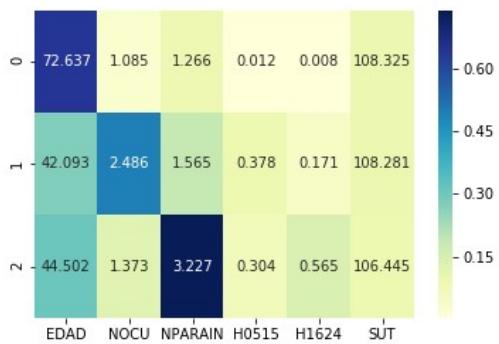


Figura 37 Heatmap K-means con 3 clúster

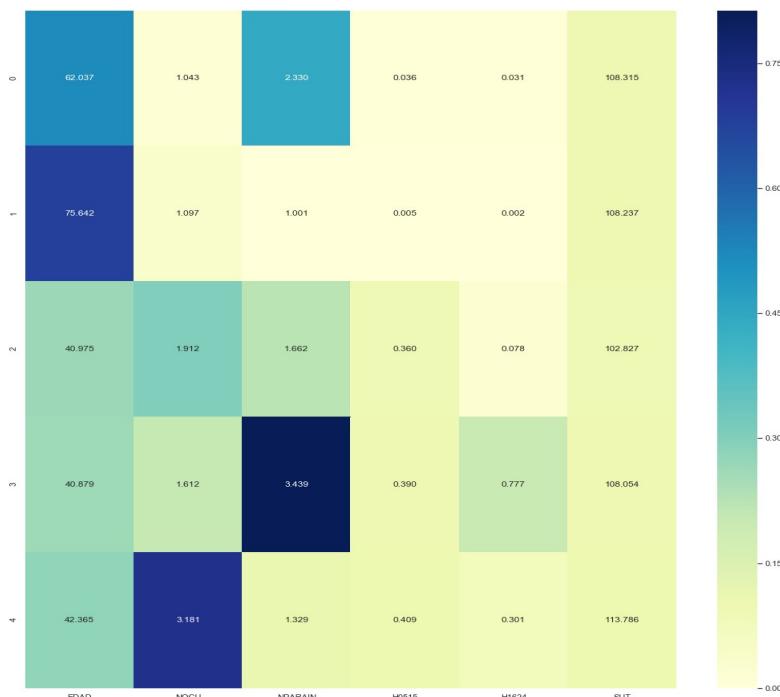


Figura 38 Heatmap K-means con 5 clúster

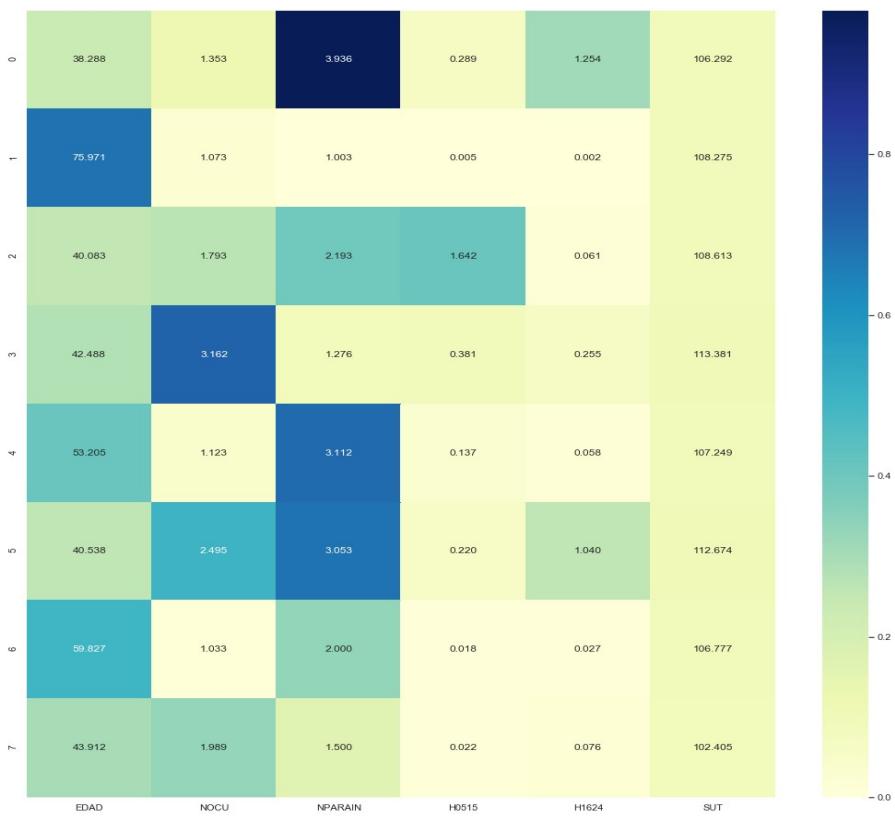


Figura 39 Heatmap K-means con 8 clúster

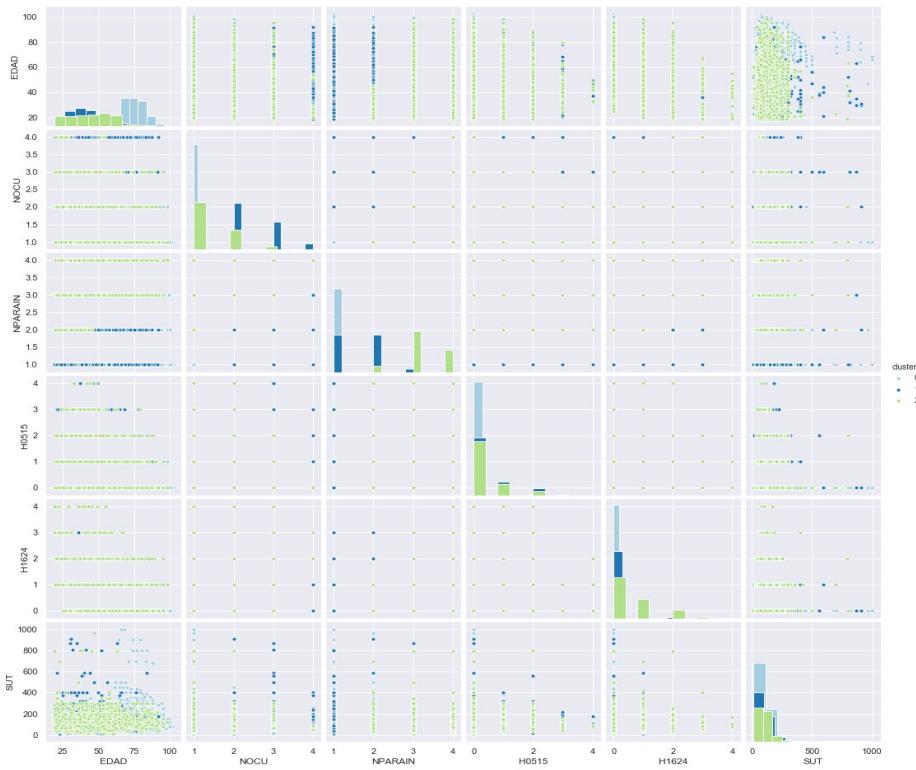


Figura 40 Scatter matrix K-means con 3

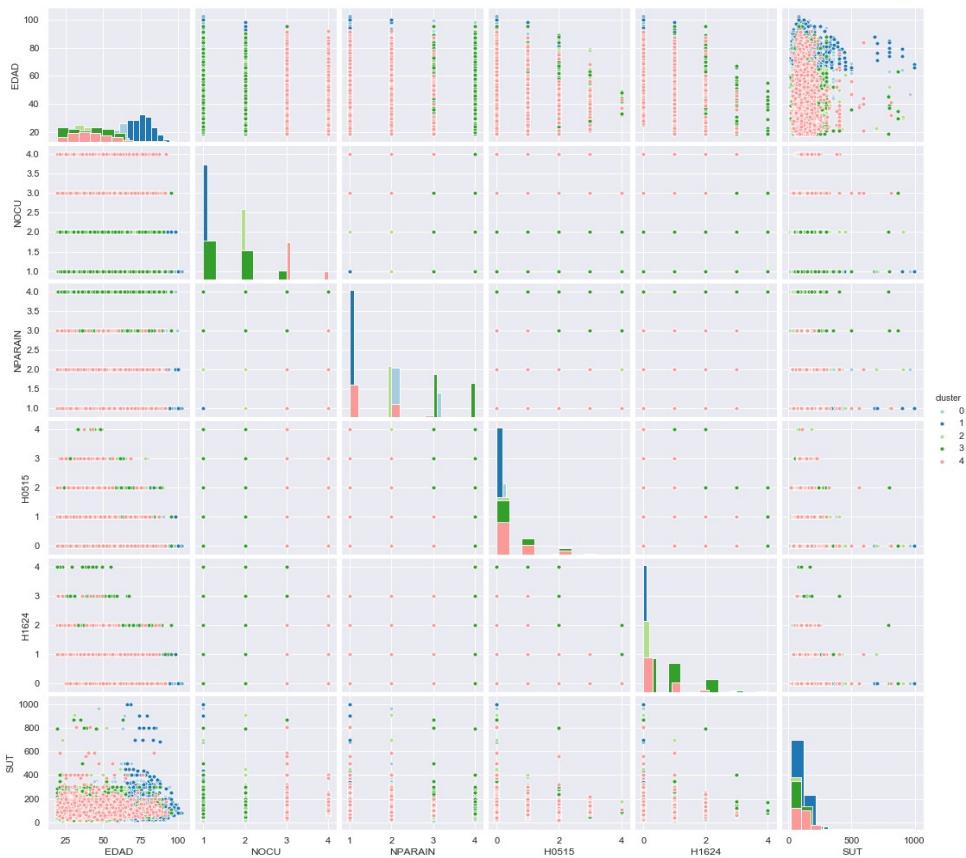


Figura 41 Scatter matrix K-means con 5

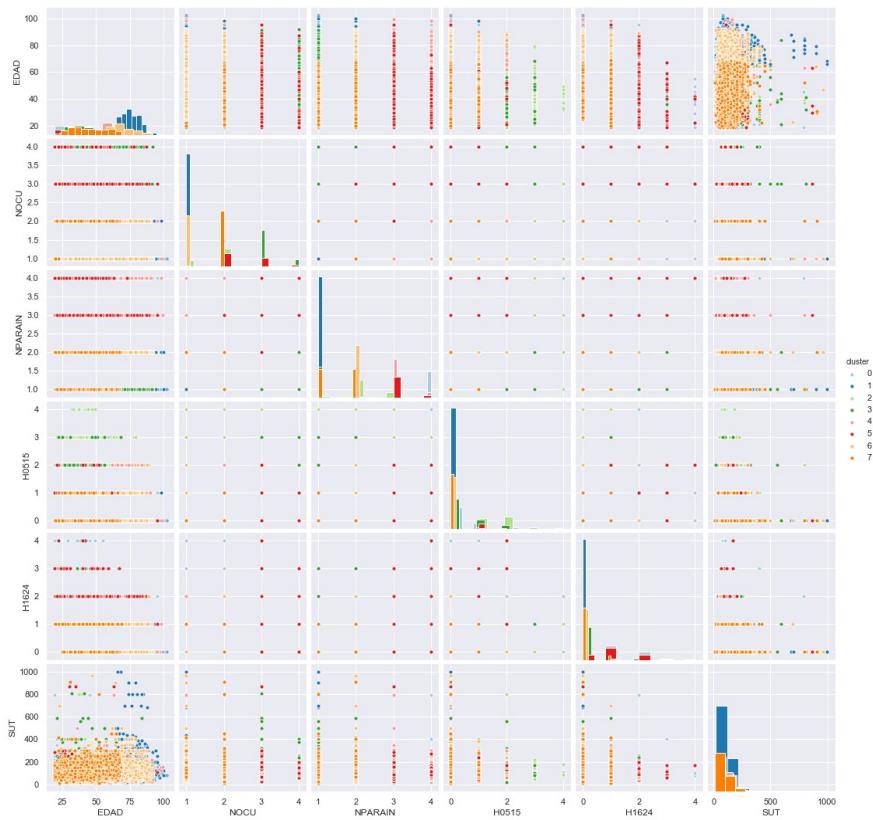


Figura 42 Scatter matrix K-means con 8

Birch

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
Birch	5	7082.637	0.35889	0.76

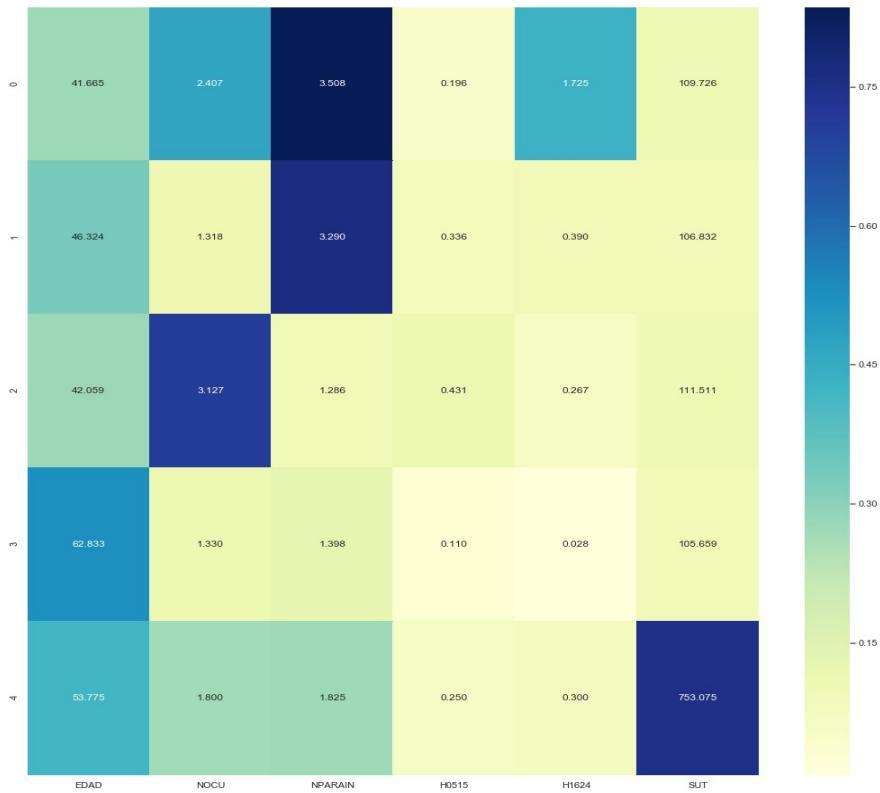


Figura 43 Heatmap Birch

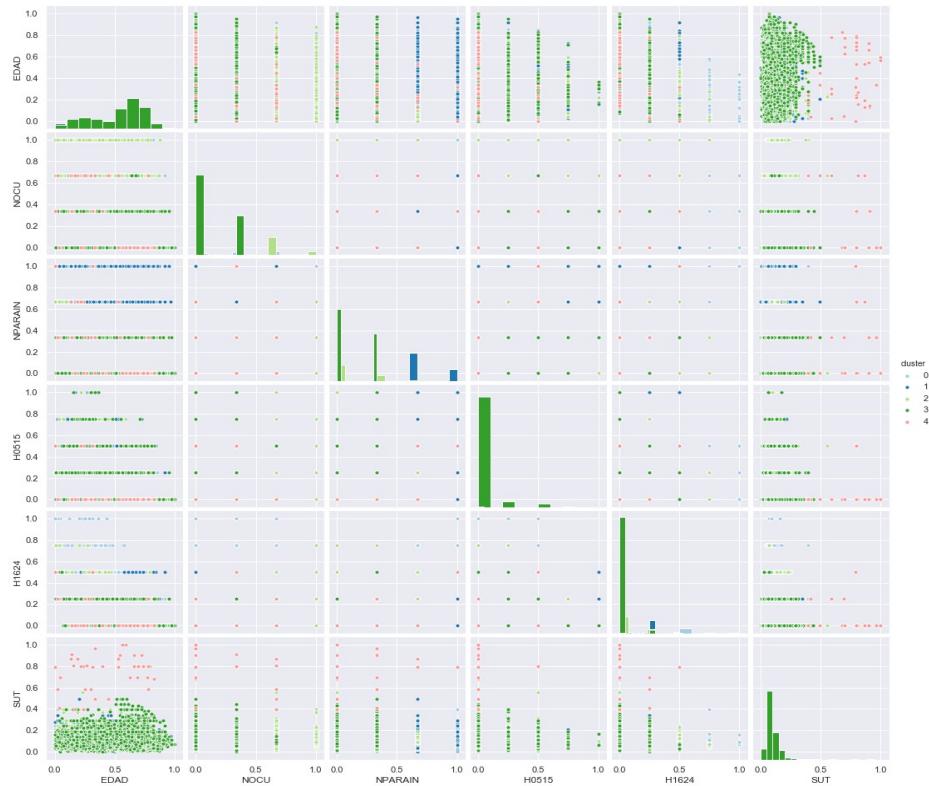


Figura 44 Scatter matrix Birch

Meanshift

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
MeanShift bandwidth=0.45	5	4833.900	0.27573	0.45
MeanShift con estimate_bandwidth (X_normal, n_samples=300)	3	3181.497	0.27643	2.14
MeanShift con estimate_bandwidth(X_normal, n_samples=10000)	3	3181.497	0.27643	2.14

band=estimate_bandwidth(X_normal, n_samples=N_Samples)

MeanShift(bandwidth=band, bin_seeding=True)

He querido aplicar también cambios en este algoritmo en el ancho de banda que sirve para sacar el número de clúster final. En el primer algoritmo lo he aproximado yo manualmente para sacar el número de clúster que quería. En los dos siguientes he querido usar una función la cual saca unos datos del conjunto de muestras y con ellos estima el ancho de banda. Como podemos comprobar no hay mucho cambio en el numero de clúster.

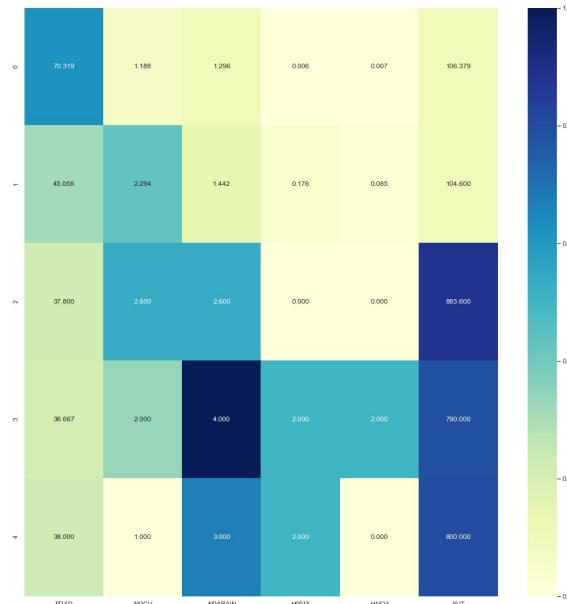


Figura 45 Heatmap MeanShift bandwidth=0.45

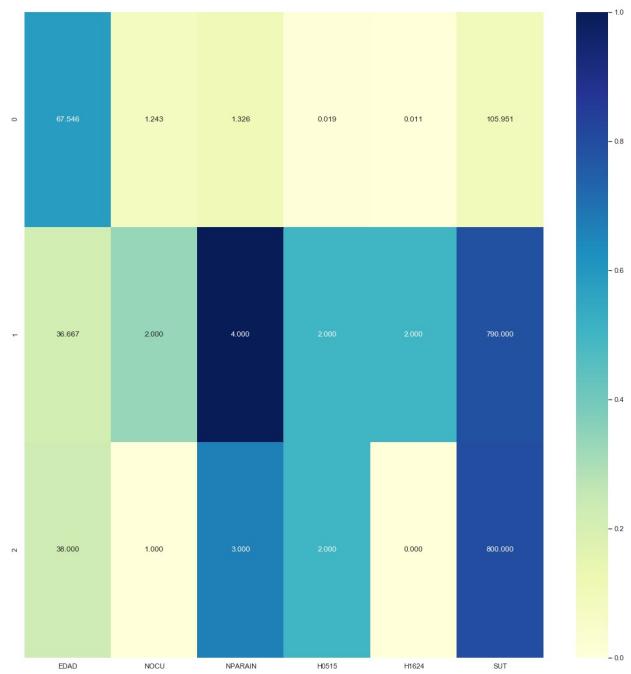


Figura 46 Heatmap MeanShift con estimate_bandwidth (X_normal, n_samples=300 y 5000)

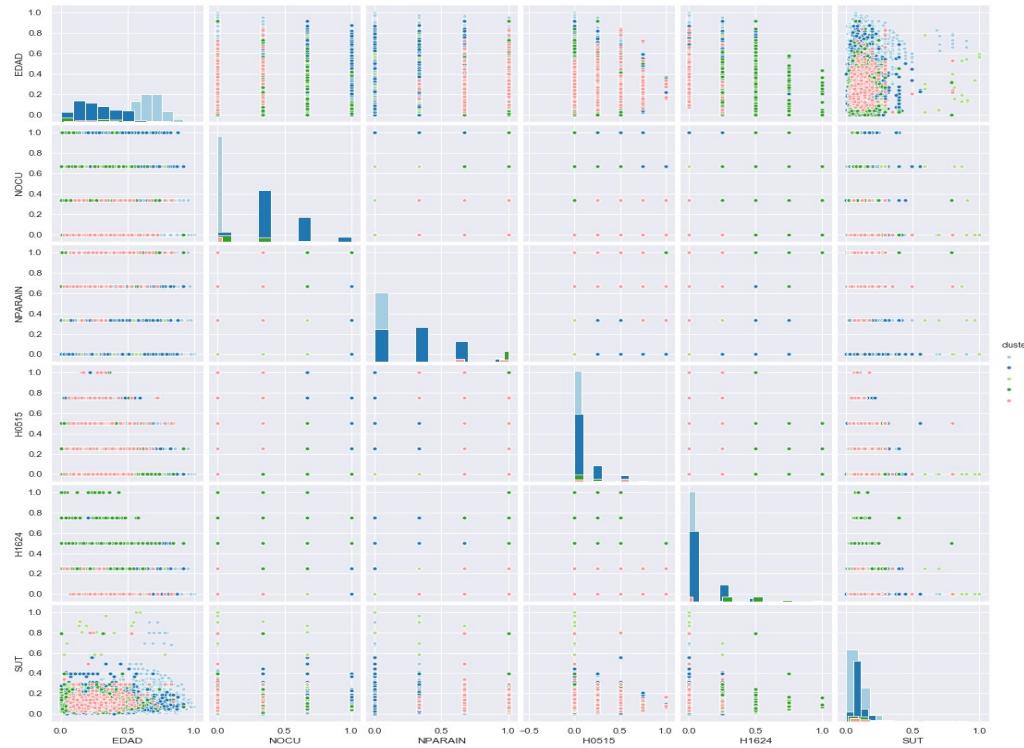


Figura 47 Scatter matrix MeanShift bandwidth=0.45

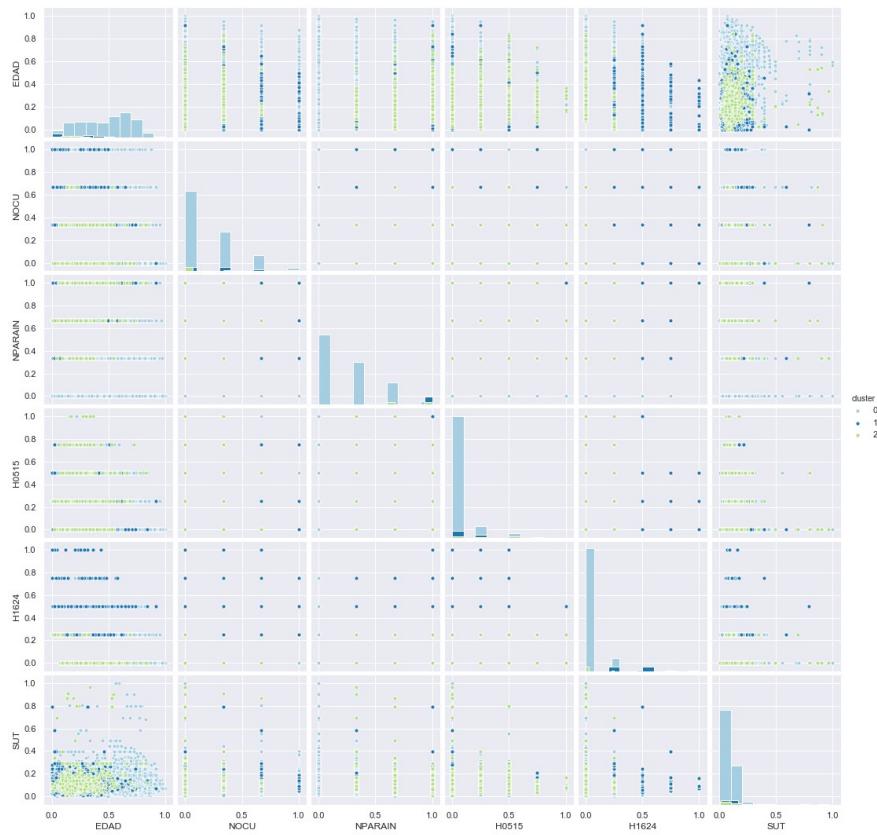


Figura 48 Scatter matrix MeanShift con estimate_bandwidth (X_normal, n_samples=300 y 5000)

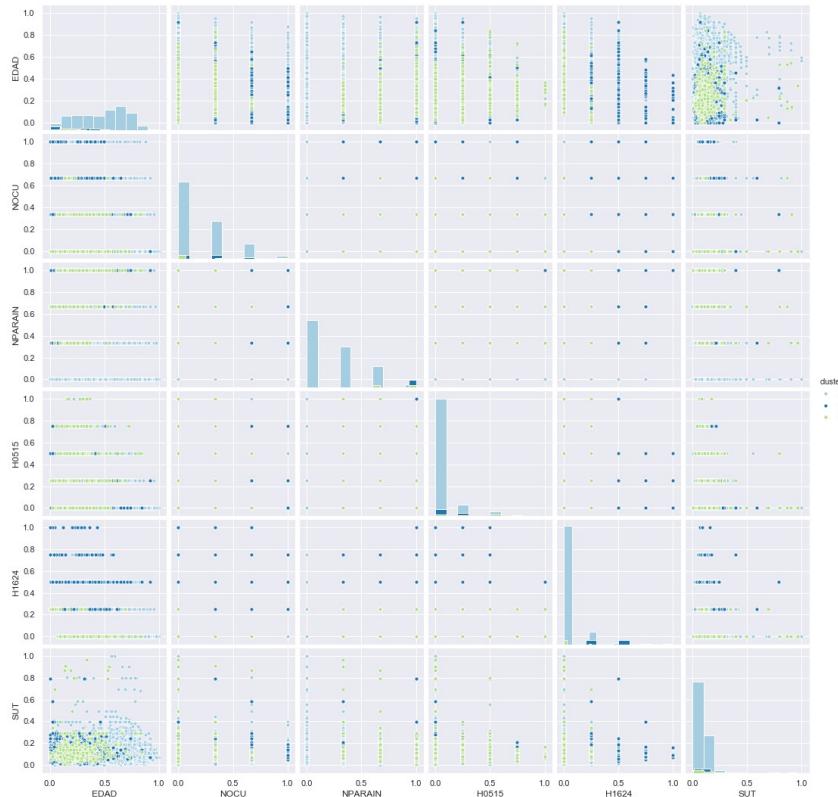


Figura 49 Scatter matrix

AgglomerativeClustering

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
AgglomerativeClustering	5	11083.571	0.36803	26.57

AgglomerativeClustering(n_clusters=5, affinity='euclidean', memory=None, connectivity=None, compute_full_tree='auto', linkage='ward', pooling_func='deprecated')

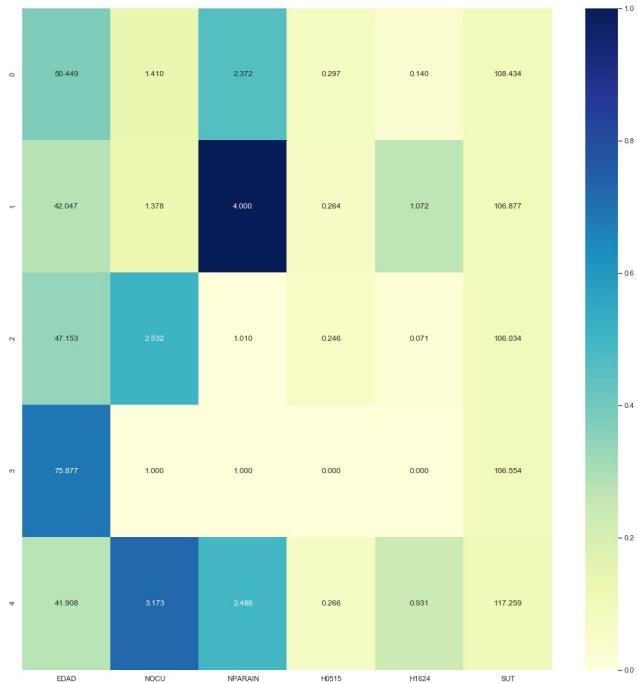


Figura 50 Heatmap AgglomerativeClustering



Figura 51 Scatter matrix AgglomerativeClustering

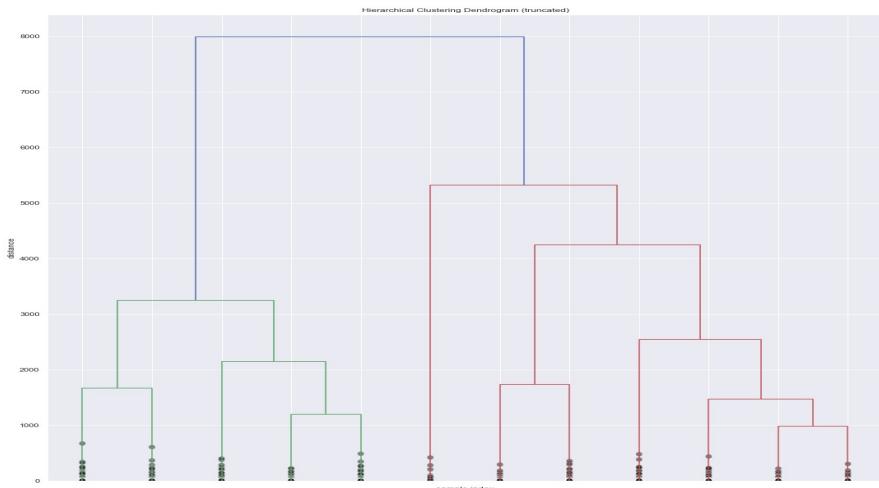


Figura 52 Dendrogramma AgglomerativeClustering

SpectralClustering

Algoritmo	N_Cluster	Calinski-Harabaz	Silhouette	Tiempo
SpectralClustering	5	922.479	0.36574	0.54

```
SpectralClustering(n_clusters=5, eigen_solver=None, random_state=None, n_init=20,
gamma=1.0, affinity='rbf', n_neighbors=10, eigen_tol=0.0, assign_labels='kmeans', degree=3,
coef0=1, kernel_params=None, n_jobs=None)
```

Al igual que algoritmo AffinityPropagation necesitamos reducir el número de datos de la muestra ya que si no por si complejidad tardara un tiempo excesivo.

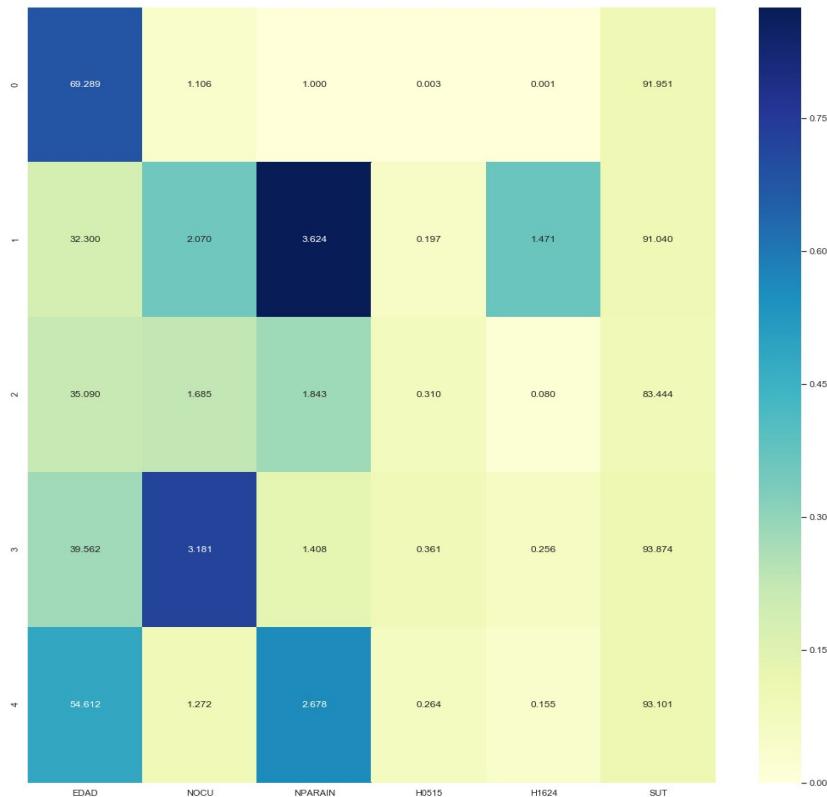


Figura 53 Heatmap SpectralClustering



Figura 54 Scatter matrix SpectralClustering

- Interpretación de la segmentación

Para este caso la muestra que ha quedado después de hacer el filtrado es de 24452 datos. Por ello para este caso sí que he tenido que hacer una reducción del número de datos para el algoritmo SpectralClustering. Su muestra se ha quedado en 2000 datos.

Como era de esperar en este caso también el algoritmo K-means ha dado los mejores resultados. Los resultados son mejores cuantos menos clúster tenga. En este caso sí que hay una gran diferencia entre los resultados de los algoritmos. Esta vez he decidido cambiar el DBSCAN debido a los malos resultados por el MeanShift, pero los resultados no han salido tampoco buenos. Este algoritmo da el Silhouette más malo. Aunque el índice Calinski-Harabaz mas malos es el del algoritmo SpectralClustering que ha sacado el peor con diferencia. Por otro lado esta vez el birch no ha sido tan bueno como en los casos anteriores. En cambio el AgglomerativeClustering ha dado un resultado bastante bueno aunque como siempre y debido al tamaño de la muestra ha sido el que más tiempo ha tardado.

Para el último caso los clúster que he escogido para explicar son:

Nombre algoritmo	N_cluster	N_datos
		0 : 1756

K-means 8	8	1 : 7593
		2 : 1510
		3 : 2670
		4 : 2715
		5 : 1436
		6 : 3263
		7 : 3509

La primera conclusión que podemos sacar es que la gente joven o de edad de menos de 35 años suele tener una línea de internet. Incluso viendo la figura 38 podemos fijarnos que la mayor cantidad de gente sin internet reside en las personas de mayor edad como podemos ver en los clúster 1, 4, 6 y 7 del algoritmo K-means. También podemos ver que en las casas en las que hay un mayor número de niños entre 5 y 15 años son un clúster minoritario. Hoy en día la conexión a internet es importante a la hora del aprendizaje de los niños por lo tanto la mayoría de las familias con menores suelen tener internet. De hecho según la muestra filtrada no es ni un 10% si tenemos en cuenta que la mayoría están en el clúster 2.

También podemos ver que hay una relación dependiendo del número de parados en la casa con respecto al número de personas que están ocupados que hace que la gente no se pueda permitir tener una red en casa. Como el clúster 4 en el que la cantidad de parados o inactivos es mucho más grande. Además, es un 10% de la muestra. También existe el caso contrario que sería el clúster 3.

La realidad es que hoy en día las personas jóvenes necesitan estar conectados a internet para mejorar su nivel de aprendizaje. En cambio personas de mayor edad parece que no quieren aprender las utilidades que se pueden sacar de tener internet. A la hora de ofrecer internet intentaría hacerlo a aquellas familias que tuvieran menores en casa ya que puede ser un trampolín para que compren el servicio.

Referencias

- [1] <https://scikit-learn.org/stable/modules/clustering.html>
- [2] <https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/>