Práctica 2:
Análisis relacional mediante segmentación

Curso 2019/2020

SOFÍA ALMEIDA BRUNO
sofialmeida@correo.ugr.es

Grupo IN 2
Jueves 9:30-10:30

Índice

1. Introducción **2**

2. Caso de estudio 1 - Primer hijo con más de 30 años **3**
2.1. Análisis KMeans 4
2.1.1. Análisis de parámetros 8
2.2. Birch 9
2.2.1. Análisis de parámetros 13
2.3. Interpretación de la segmentación 14

3. Caso de estudio 2 - Sin deseo de tener hijos **14**
3.1. Análisis KMeans 16
3.1.1. Análisis de parámetros 20
3.2. Análisis DBSCAN 21
3.2.1. Análisis de parámetros 21
3.3. Birch 23
3.4. Interpretación de la segmentación 27

4. Caso de estudio 3 - Tratamiento de reproducción asistida **27**
4.1. Análisis KMeans 28
4.1.1. Análisis de parámetros 31
4.2. Análisis Ward 32

5. Birch **35**
5.1. Análisis de parámetros 35
5.2. Interpretación de la segmentación 36

1. Introducción

Tras estudiar, en la práctica anterior, algoritmos de clasificación para un problema de aprendizaje supervisado, pasamos ahora a un problema de aprendizaje no supervisado. Utilizaremos diferentes algoritmos de agrupamiento o *clustering* para realizar un análisis relacional.

En este caso partiremos de los microdatos publicados por el Instituto Nacional de Estadística (INE) en 2018 sobre la última encuesta de fecundidad. De este conjunto, que contiene a personas entre 18 y 55 años, nos quedamos con los datos relativos a las mujeres. Elegiremos tres casos de estudio diferentes, en los que seleccionaremos las variables a estudiar (en general, de tipo categórico) y las variables sobre las que aplicar *clustering* (no tiene interés aplicarlo sobre el resto de variables ya que podrían no ser relevantes), compararemos distintos algoritmos y analizaremos los resultados obtenidos de forma comparada.

El conjunto original dispone de 14556 respuestas a la encuesta, con 463 variables sobre identificación, datos biográficos, hogar, vivienda, padres, relaciones de pareja actual e historia, hijos, embarazo actual y anteriores,...

En cada caso de estudio utilizaremos y compararemos varios algoritmos de agrupamiento, centrándonos en dos de ellos para hacer un análisis más profundo y tratando de mejorar sus resultados mediante el ajuste de sus parámetros ([1]). Los algoritmos elegidos y los parámetros con los que usarlo inicialmente son los siguientes:

- **K-Means**. Agrupa los ejemplos en tantos grupos como se le haya indicado. Parte de unos centros iniciales, asigna a cada ejemplo el *cluster* correspondiente al centro más cercano, recalcula los centros y vuelve al paso anterior. Este proceso se repite hasta que no haya más cambios en los clusters. Al estar basado en centroides generará *clusters* convexos. Parámetro inicial: `n_clusters = 5`.
- **MeanShift**. Este algoritmo está basado en la densidad de las muestras. A partir de un radio fijado, determina un número de *clusters* y va desplazando sus centros hacia las regiones más densas. Es otro ejemplo de algoritmo basado en centroides, los *clusters* generados serán también convexos. Preliminarmente estima `bandwidth` por defecto.
- **Birch**. Agrupa los objetos según se vayan recibiendo, es un algoritmo de *clustering* incremental. Crea un árbol con las características de los *clusters* guardando solo la información necesaria para poder determinarlos. Cada nodo tendrá un número de *subclusters* acotado por el factor de ramificación y un umbral que determina si un nuevo ejemplo está lo suficientemente cerca del *cluster* para pertenecer a él. Cuando llega un objeto, desciende por el árbol escogiendo en cada nodo el de características similares. Si no pertenece a ningún *cluster* y no se ha superado el factor de ramificación, se crea un nuevo *subcluster*, en caso contrario es absorbido por el *cluster* en cuestión. El último parámetro a fijar es el número de *clusters* con el que quedarnos finalmente. Inicialmente `branching_factor=25`, `threshold=0.25`, `n_clusters=5`.
- **DBSCAN**. Es un algoritmo basado en densidad que no utiliza centroides, por lo que los grupos generados pueden tener diversas formas. Utiliza dos parámetros para definir la densidad: un valor ε que determina cuándo un objeto es densamente alcanzable a partir de otro, un número mínimo de objetos por los que podemos alcanzar a otro para afirmar que pertenecen al mismo cluster. Generará en ocasiones un grupo etiquetado como "-1" que contiene a aquellos puntos que no pueden ser agrupados. Se comienza con `eps=0.2`[1], `min_samples=5`.
- **Ward**. Este método se distingue de los demás en que es jerárquico, es decir, genera una jerarquía que según el nivel de corte dará distintos agrupamientos. En este caso se ha elegido un algoritmo que en cada paso va agrupando dos *clusters* (inicialmente cada objeto es un *cluster*) de forma que el agrupamiento minimice la varianza (media de la distancia al cuadrado de cada elemento al centroide). Especificaremos como parámetro en qué nivel de la jerarquía nos quedamos, en este caso, `n_clusters=5`.

[1] Se partió de un valor inicial $\varepsilon = 0.5$, pero al ver que no conseguía buenos resultados en los casos a estudiar se disminuyó tratando de conseguir más variedad en los grupos.

Para comparar los resultados y rendimiento de los diferentes algoritmos según distintos índices:

- **Tiempo**. Mediremos el tiempo que tarda cada algoritmo en agrupar el mismo conjunto de datos. Aunque no es el factor más importante en clustering, puede ser determinante para decantarnos por los algoritmos más rápidos entre aquellos que consigan resultados similares lo suficientemente buenos.
- **Calinski-Harabasz**. Es una ratio entre la dispersión intracluster e intercluster para todos los clusters, donde la dispersión se mide como la suma de las distancias al cuadrado. Un valor alto indica que los grupos son densos y están bien separados.
- **Silhouette**. Este coeficiente mide cómo de similares son los objetos de un mismo *cluster* comparado con los de otros clusters. Toma valores en el intervalo $[-1, 1]$. Un valor cercano a 1 indica que los *clusters* están muy concentrados y distantes a los demás, uno cercano a 0 señala que los *clusters* están superpuestos y uno cercano a -1 que el agrupamiento es incorrecto.
- **Número de clusters**. Comparando este valor observaremos si algún algoritmo no logra agrupar correctamente, por hacer demasiados *clusters* o, por el contrario, por no agrupar lo suficiente. En los algoritmos que sea un parámetro fijo podemos variarlo y ver cuál es el número de *clusters* que mejor se adapta a nuestro conjunto.

Se busca obtener información con algún tipo de interés social. Por ejemplo, el retraso en la maternidad puede tener consecuencias negativas para el bebé. Así, en el caso de estudio 1, se estudia a las mujeres que tuvieron su primer hijo con más de treinta años, se busca conocer a los grupos de mujeres que se quedan embarazadas después de cierta edad para comprender mejor su situación y los motivos que provocan este retraso en la maternidad. El caso de estudio 2 se centra en una cuestión contraria, las mujeres que no desean tener hijos, con el objetivo de entender qué factores son los que afectan en la intención de tener hijos de estas mujeres. Un caso delicado es el de las mujeres que se someten a tratamientos de reproducción asistida (por las complicaciones médicas añadidas que pudiera tener). El caso de estudio 3 se centra en este conjunto de mujeres con el fin de conocer las necesidades de los grupos de este conjunto.

En todos los casos, para apoyar el análisis se han realizado gráficas de distintos tipos con la ayuda de `matplotlib` ([6]), como gráficos de barras ([5]) o mapas de calor ([2]), y de `seaborn` ([3]).

2. Caso de estudio 1 - Primer hijo con más de 30 años

En el resumen del INE sobre esta encuesta no se realizó un análisis de agrupamiento como el desarrollado en esta práctica, como mucho se relacionaron dos variables. Tomando como inspiración el resumen [4] del INE, que muestra algunos resultados destacados obtenidos a partir de dicha encuesta, enfocaremos el primer caso de estudio al retraso en la maternidad. Para ello, calculamos (a partir de los datos dados) la edad media en la que las mujeres tienen su primer hijo: a los 24 años. Así, fijamos como primer caso de estudio el conjunto de *mujeres que tuvieron su primer hijo con 30 años o más*. Este subconjunto está formado por 1545 objetos.

Para realizar el agrupamiento en este conjunto selecciono cuatro variables que pueden ayudarnos a distinguir la situación y preocupaciones de las mujeres con una maternidad tardía:

- **ANOVI** es el año en que empezó a vivir en la vivienda actual.
- **ANORELACION** indica el año en el que comenzó la relación sentimental actual.
- **ANOTRABACT** representa el año en el que la mujer comenzó a trabajar en su puesto actual.
- **EDADIDEAL** como su nombre indica, es la edad que la entrevistada considera oportuna para tener el primer hijo.

Una vez fijado el caso de estudio y las variables mediante las que agrupar, utilizamos el *script* `caso1.py` para ejecutar los cinco algoritmos y obtener tanto la tabla de resultados como los diferentes tipos de gráficos asociados. Podemos observar en la Tabla 1 los índices obtenidos en este caso de estudio.

Tabla 1: Resultados caso de estudio 1

Algoritmo	Tiempo (s)	Calinski-Harabasz	Silhouette	Número de clusters
KMeans	0.063	580.785	0.27533	5
MeanShift	9.857	9.770	0.32542	2
Birch	0.087	321.158	0.26552	5
DBSCAN	0.035	28.496	0.38585	2
Ward	0.099	501.202	0.29108	5

En una primera lectura detectamos que el tiempo de ejecución del algoritmo MeanShift es considerablemente mayor que el del resto de algoritmos. Esto se debe a que cuando lo ejecutamos, como no especificamos el valor de los parámetros, los calcula usando `sklearn.cluster.estimate_bandwidth` que, como indica su documentación [**bandwidth**], tarda un tiempo cuadrático sobre el número de objetos. El tiempo de ejecución del resto de algoritmos es similar, siendo ligeramente inferior el de KMeans y DBSCAN.

También destaca el bajo número de *clusters* generados por los algoritmos DBSCAN y MeanShift: 2. Ambos formaron un único grupo (DBSCAN con el 98.83 % de los objetos, MeanShift con el 99.68 %) y juntaron en el otro *cluster* los objetos que no encajaban en él. Esto provoca que los objetos del *cluster* único estén muy separados de los del otro cluster, esta es la posible causa de que el índice Silhouette sea más elevado que en el resto de casos. Sin embargo, el índice Calinski-Harabasz es inferior en estos casos pues no se ha realizado un buen agrupamiento.

El resto de algoritmos consigue mayor valor del coeficiente Calinski-Harabasz y valores similares del coeficiente Silhouette. Para realizar un análisis de los grupos me centraré en los algoritmos KMeans y Birch, aunque Ward consiguió mejores resultados, lo descarto para realizar un análisis en profundidad en un conjunto de menor tamaño.

2.1. Análisis KMeans

El algoritmo KMeans genera 5 grupos, como habíamos indicado, vemos en la Figura 1 la distribución de los objetos en los distintos grupos.

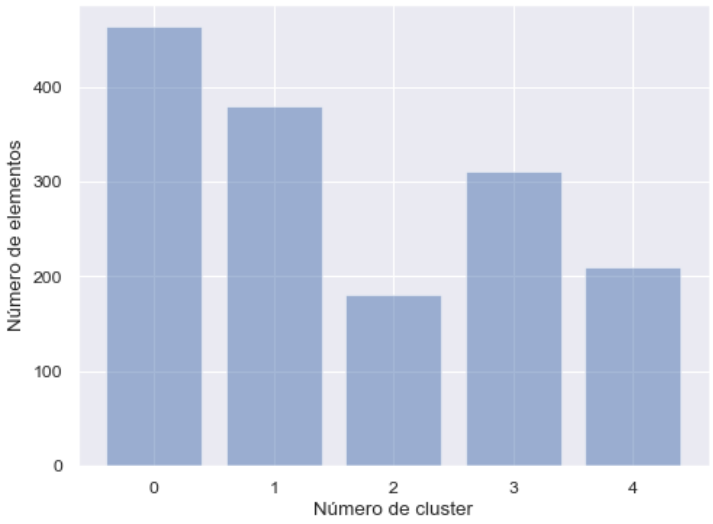

Figura 1: Caso 1 - Tamaño de los clusters- KMeans

Los grupos tienen tamaños en rangos similares, entre 300 y 450 los tres mayoritarios, seguidos por un *cluster* con 209 elementos y el menor con 180. Para analizar los grupos, observamos en la Figura 2 la matriz de dispersión de las variables dos a dos y el histograma de cada una de ellas en la diagonal.

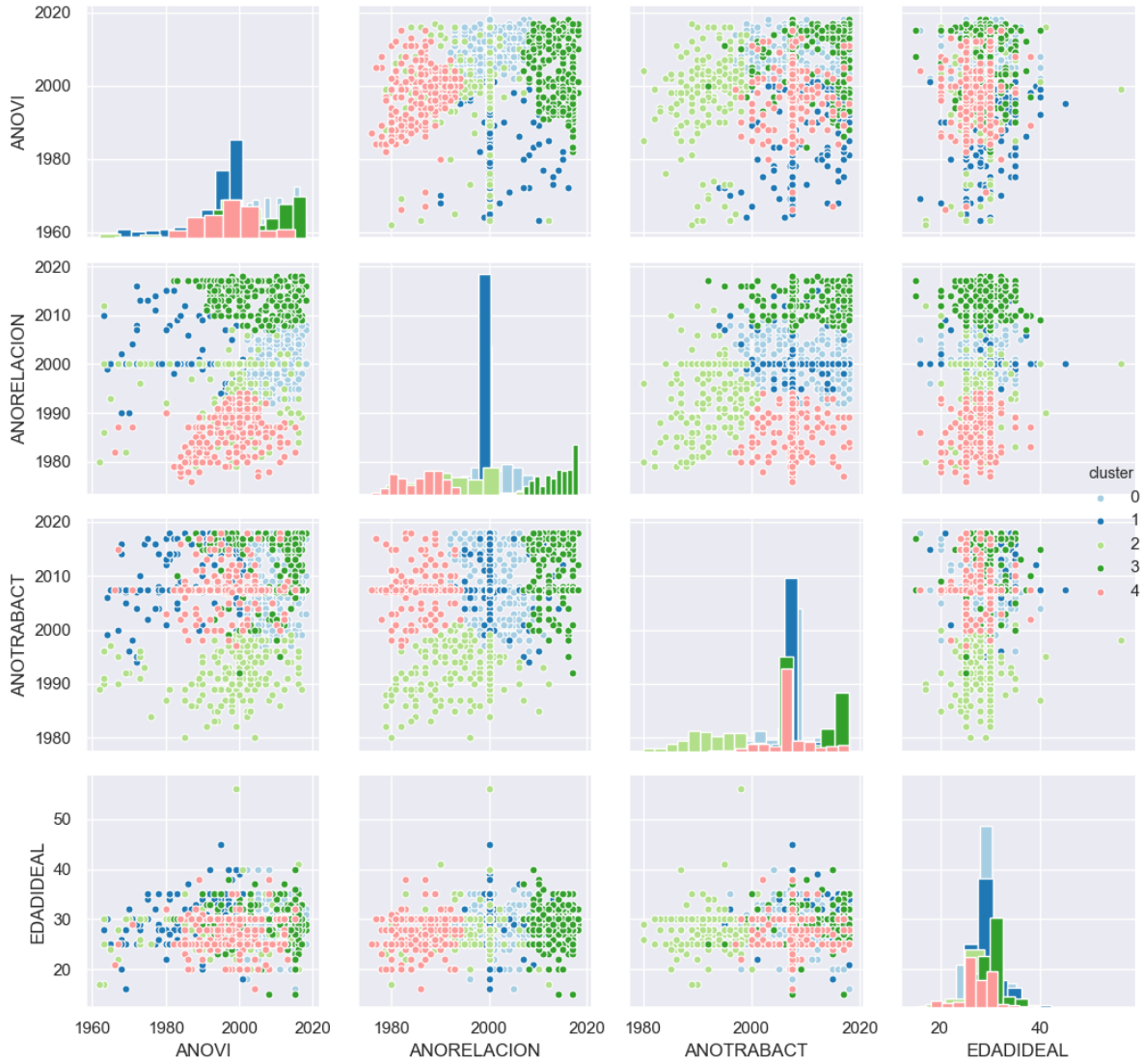

Figura 2: Caso 1 - Matriz de dispersión - KMeans

Las variables ANORELACION y ANOTRABACT sirven para separar bastante bien cuatro de los grupos, pero el resto también aporta información relevante. Aparentemente tenemos 4 grupos bien definidos y uno un poco más disperso, el *cluster* 1. Al cambiar el valor del parámetro `n_clusters` descubriremos si al reagrupar se consiguen *clusters* más compactos. Rellenaremos la Tabla 2 con ayuda del diagrama de cajas mostrado en la Figura 3 para hacernos una idea de las características principales de cada grupo.

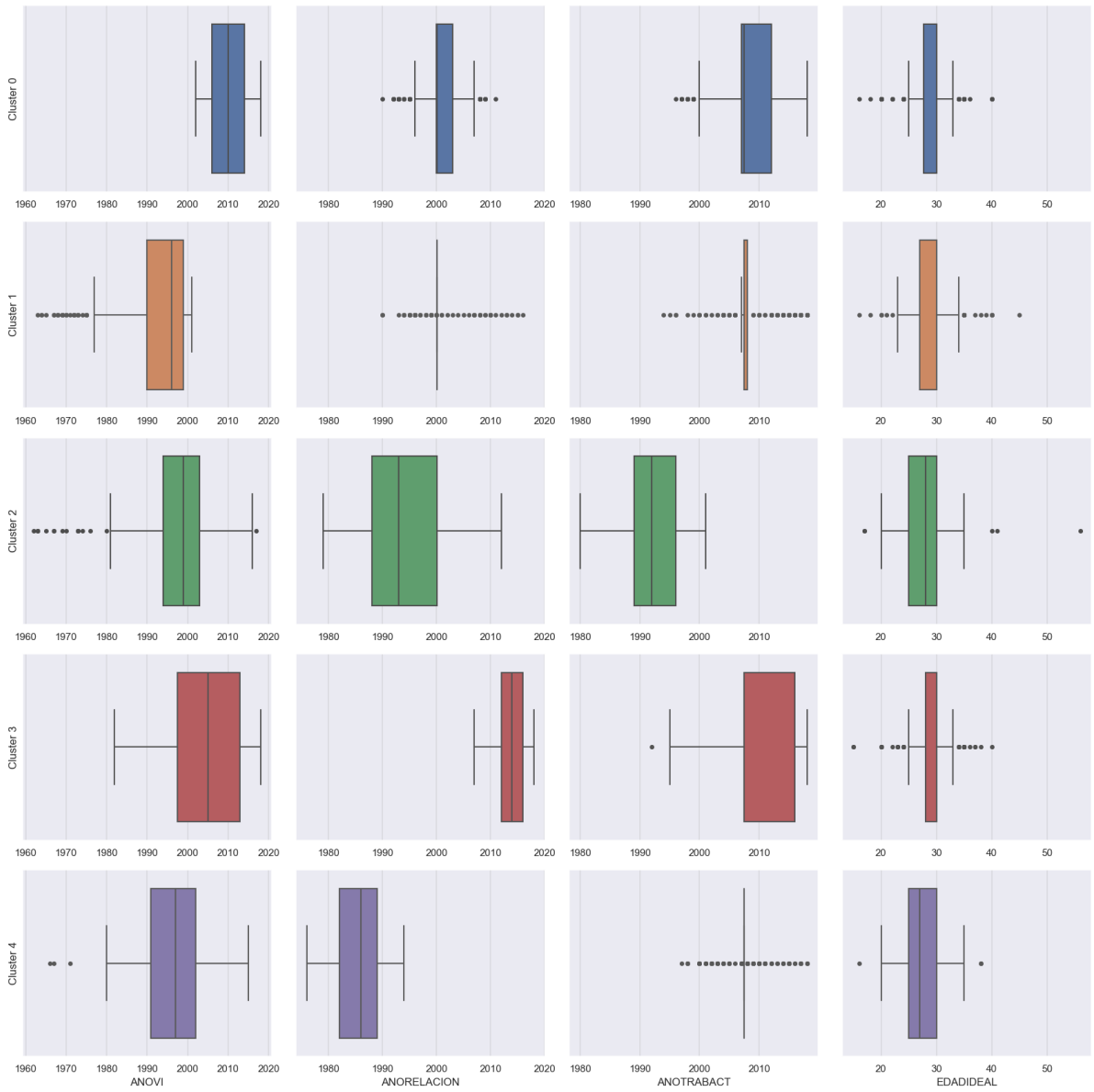

Figura 3: Caso 1 - Diagrama de cajas - KMeans

Tabla 2: Caso 1 - Características de los grupos - KMeans

Cluster	ANOVI	ANORELACION	ANOTRABACT	EDADIDEAL
0	2005-2015	2000-2003	2007-2012	28-30
1	1990-1999	2000	2008	27-30
2	1995-2003	1987-2000	1988-1996	25-30
3	1997-2012	2012-2016	2007-2016	28-30
4	1991-2002	1983-1989	2008	25-30

Destaca que la mayor parte de las entrevistadas consideran que la edad ideal para tener un hijo es, como mucho, 30 años. Aunque en cada grupo el rango de edad que se considera apropiado para tener un hijo varía, considerando el mayor subconjunto, en general podemos afirmar que las mujeres que tuvieron el primer hijo con más de 30 años consideran que lo ideal habría sido tenerlo antes.

El *cluster* 0, mayoritario, está caracterizado por las mujeres que comenzaron su relación en torno al año

2000 y pasaron a su vivienda actual a partir del año 2005, probablemente se mudaran a una vivienda común con su pareja. Comenzaron a trabajar en el puesto actual a partir del año 2007, posiblemente después de tener ese primer hijo.

El *cluster* 1, como ya habíamos observado en la matriz de dispersión, es un *cluster* más disperso, contiene *outliers*, o valores anómalos, en todas las variables.

La variable que más nos ayuda a distinguir el *cluster* 2 es ANOTRABACT, las mujeres de este grupo llevan trabajando desde el siglo anterior en el mismo empleo. Además, es característico que comenzaron su relación actual en la década de los 90.

El *cluster* 3 está compuesto por aquellas mujeres que comenzaron su relación actual más recientemente, entre 2012 y 2016. La fecha de comienzo de su trabajo actual es unos años antes de estos comienzos de relación.

Para diferenciar al *cluster* 4 es el año de comienzo de la relación el que nos dará la clave. Las mujeres de este *cluster* comenzaron su relación actual en la década de los 80. Pasaron a la vivienda actual en los 90.

Los grupos 0 y 3 son similares en el año de comienzo del trabajo actual, pero se distinguen en ANOVI y ANORELACION. En el grupo 0, la entrevistada comenzó la relación y posteriormente se mudó al domicilio actual. Mientras que en el grupo 3 las mujeres estudiadas vivían en la vivienda actual desde antes de comenzar la relación.

2.1.1. Análisis de parámetros

En el algoritmo KMeans el parámetro decisivo es el número de clusters. A priori, dado un conjunto de datos tan grande y sin conocimiento experto sobre el problema, es difícil acertar con el número de grupos en los que dividir el subconjunto. Podríamos aumentar este número enormemente para conseguir *clusters* muy compactos, el límite sería que cada objeto fuera su propio cluster, pero entonces no cumpliríamos el objetivo del análisis mediante segmentación: obtener información de los datos al agruparlos según una serie de variables.

Para probar con distintas opciones de este parámetro implementamos el *script* `caso1-kmeans`, donde se ejecuta el algoritmo KMeans variando el parámetro `n_clusters` en el rango de enteros $[2, 15]$. En la Tabla 3 vemos los resultados obtenidos para cada valor del parámetro.

Tabla 3: Resultados cambio de parámetros KMeans

Número de *clusters*	Tiempo (s)	Calinski-Harabasz	Silhouette
2	0.023	680.214	0.32456
3	0.024	580.263	0.26207
4	0.038	560.781	0.24693
5	0.052	587.885	0.31232
6	0.064	588.794	0.28092
7	0.079	583.712	0.29898
8	0.128	578.210	0.30893
9	0.159	559.847	0.30801
10	0.160	538.586	0.30408
11	0.205	524.341	0.30575
12	0.192	514.181	0.31093
13	0.251	508.801	0.31454
14	0.261	493.823	0.31312
15	0.289	484.230	0.30280

Como es natural, al aumentar el número de clusters, aumentan los cálculos a realizar. Como para

reasignar los *clusters* vamos comprobando si cada objeto pertenece al mismo, al aumentar el número de clusters, crece también el número de operaciones a realizar y con ello el tiempo de ejecución, como se refleja en la Tabla 3.

Para simplificar la comparación del valor de los dos coeficientes en función del número de grupos se han realizado las gráficas lineales mostradas en la Figura 4.

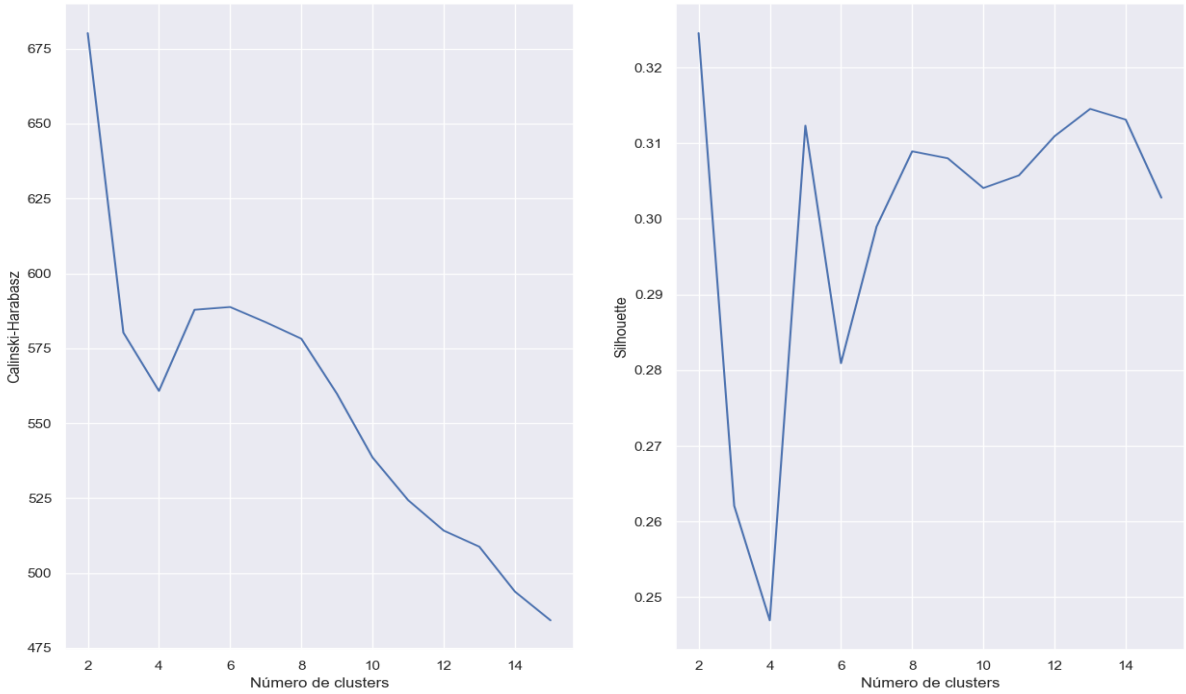

Figura 4: Caso 1 - Comparación coeficientes según el número de *clusters* - KMeans

Para un número muy bajo de clusters, 2, ambos índices alcanzan sus valores máximos. Esto se debe a que al distinguir entre tan solo dos grupos es más fácil conseguir que los elementos queden cerca de su centro. Al ir aumentando el número de grupos, será más complicado diferenciar los elementos de la frontera, disminuyendo los coeficientes.

Observamos que ambos coeficientes alcanzan un máximo local en 5 clusters, por lo que nuestra estimación por defecto fue acertada.

El coeficiente Calinski-Harabasz disminuye a medida que aumenta el número de clusters, pues aunque estén más concentrados, las diferencias entre ellos serán menores. Este hecho afecta más a este coeficiente que al coeficiente Silhouette, que a pesar de sufrir un mínimo local para 6 grupos, aumenta y se mantiene en el rango $[0.30, 0.32]$ a partir de 8 clusters.

2.2. Birch

Analizamos los grupos obtenidos por el algoritmo Birch. Veremos si son semejantes a los que proporcionó KMeans. Comenzamos observando el tamaño de los grupos creados. Los podemos ver en la Figura 5.

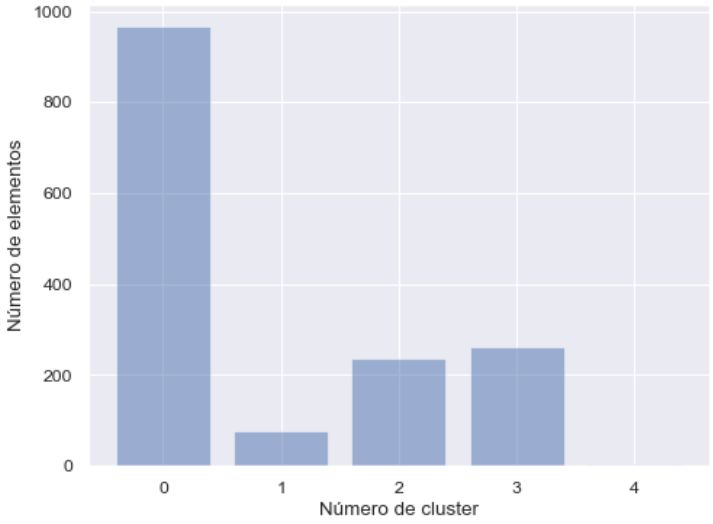

Figura 5: Caso 1 - Tamaño de los *clusters* - Birch

Los grupos creados son bastante desiguales en tamaño, de hecho el grupo 4 es prácticamente inexistente, no lo tendremos en cuenta en el análisis.

Aprovechamos la información mostrada por la matriz de dispersión en la Figura 6 para comprender las características de estos grupos.

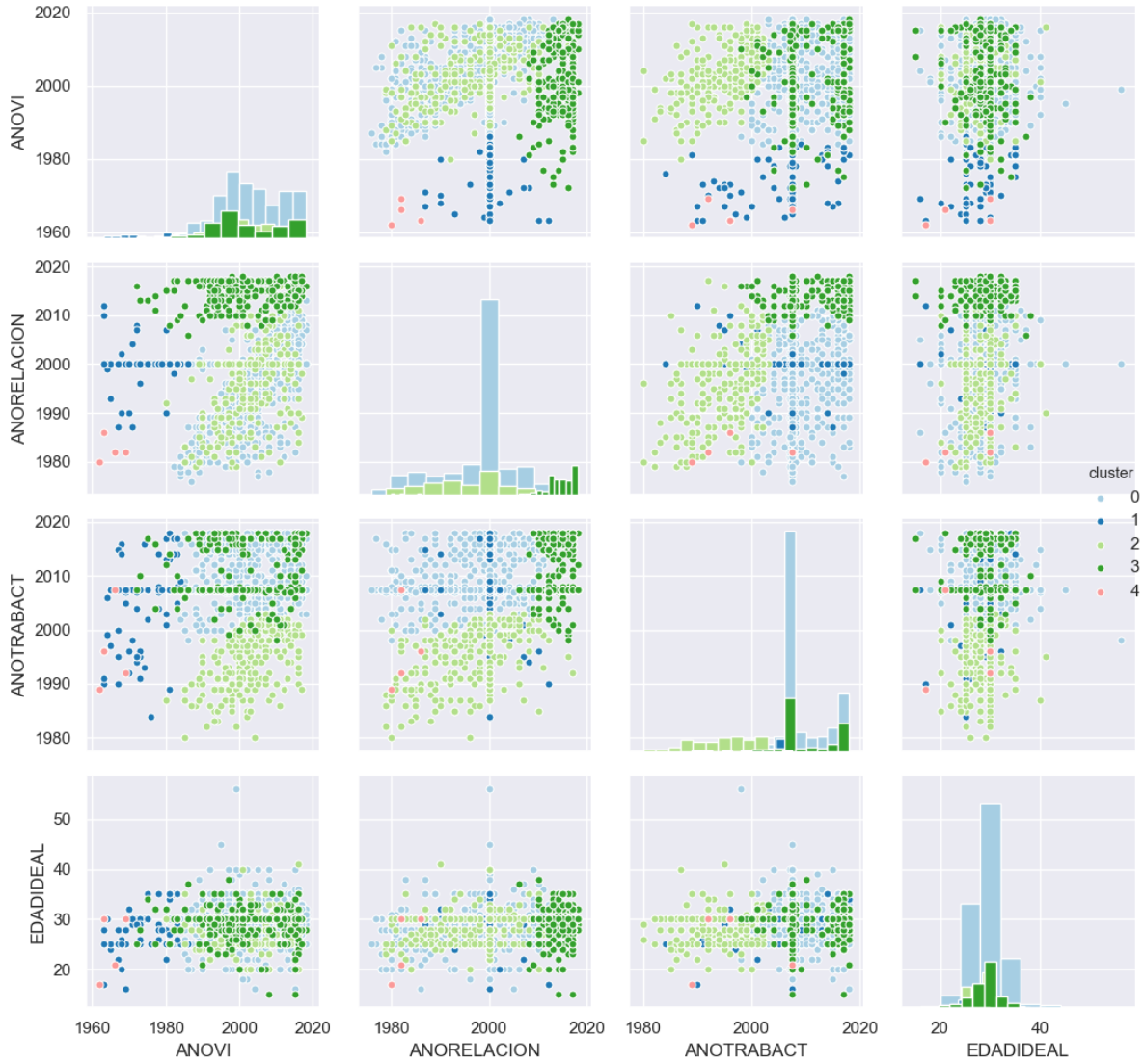

Figura 6: Caso 1 - Matriz de dispersión - Birch

Aparentemente los años en el trabajo actual y los años en la relación nos permiten distinguir tres de los grupos, mientras que para el cuarto debemos atender a otras variables como ANOVI. Pasamos a completar la Tabla 4 con las características de cada grupo. Para ello nos ayudamos del diagrama de cajas correspondiente, que podemos ver en la Figura 7.

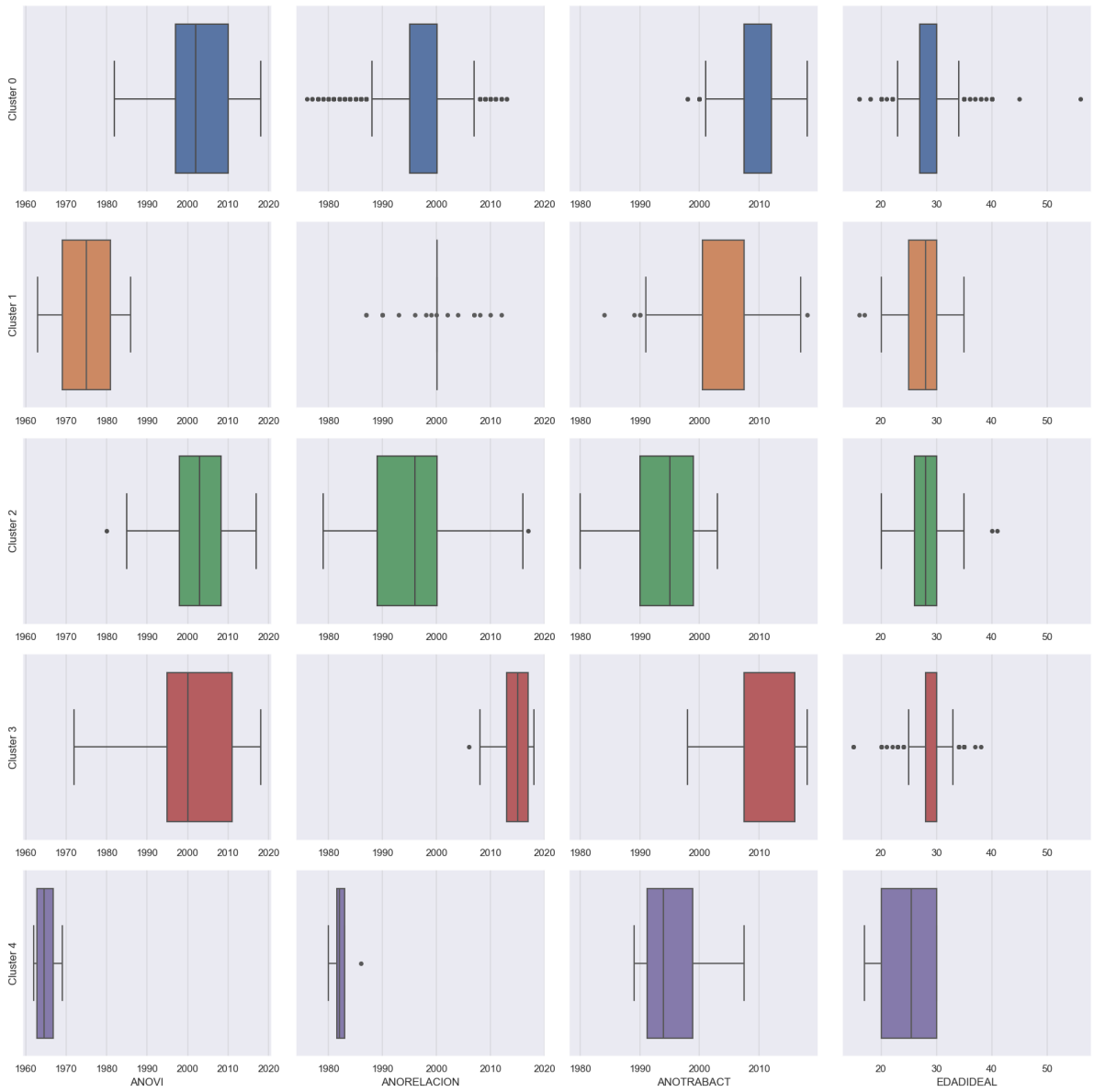

Figura 7: Caso 1 - Diagrama de cajas - Birch

Antes de comenzar el análisis de los grupos encontrados, notamos que el *cluster* 4 está formado por los valores anómalos del resto de grupos, sus valores no encajan con los de ningún grupo.

Tabla 4: Caso 1 - Características de los grupos - Birch

Cluster	ANOVI	ANORELACION	ANOTRABACT	EDADIDEAL
0	1997-2010	1995-2000	2007-2012	28-30
1	1967-1981	2000	2000-2008	27-30
2	1997-2009	1989-2000	1990-1999	28-30
3	1995-2011	2013-2017	2007-2016	29-30

Al igual que en KMeans la variable EDADIDEAL varía en un pequeño rango (25-30 años) en todos los grupos, ninguno se diferencia especialmente de los demás por el valor que esta variable pueda tomar.

El *cluster* 0 se caracteriza por contener a las entrevistadas que empezaron a trabajar alrededor del año

2010, pero que comenzaron su relación actual antes del 2000.

En el *cluster* 1 se caracteriza por el año en el que comenzaron las mujeres a vivir en su vivienda actual (entre 1967 y 1981) y por el año en que consiguieron su trabajo actual (entre 2000 y 2008).

Las mujeres que comenzaron a trabajar en su empleo actual en la década final del siglo pasado son las que componen el *cluster* 2, caracterizado por la variable ANOTRABACT. Al igual que en los grupos comentados anteriormente, la relación actual comenzó antes del 2000. El último cluster, el 3, precisamente en este hecho. Las relaciones actuales de las mujeres de este grupo comenzaron entre 2013 y 2017.

2.2.1. Análisis de parámetros

Pasamos ahora a realizar un análisis paramétrico de este algoritmo, modificaremos los valores del umbral y el factor de ramificación para ver qué efectos tiene sobre los coeficientes.

Tabla 5: Caso 1 - Resultados cambio de parámetros Birch

Umbral	Factor de ramificación	Tiempo (s)	Calinski-Harabasz	Silhouette	Nº de *clusters*
0.10	15	0.168	387.010	0.21037	5
0.15	15	0.132	387.292	0.22184	5
0.20	15	0.121	359.545	0.28990	5
0.25	15	0.073	321.158	0.26552	5
0.30	15	0.075	354.234	0.30612	5
0.10	20	0.162	355.875	0.19445	5
0.15	20	0.144	437.151	0.26688	5
0.20	20	0.125	326.572	0.30217	5
0.25	20	0.083	321.158	0.26552	5
0.30	20	0.072	354.234	0.30612	5
0.10	25	0.132	435.010	0.20884	5
0.15	25	0.129	383.414	0.27909	5
0.20	25	0.093	346.460	0.29634	5
0.25	25	0.075	321.158	0.26552	5
0.30	25	0.074	354.234	0.30612	5
0.10	30	0.135	427.033	0.25244	5
0.15	30	0.141	406.054	0.27413	5
0.20	30	0.087	341.472	0.29464	5
0.25	30	0.113	321.158	0.26552	5
0.30	30	0.082	354.234	0.30612	5
0.10	35	0.132	336.961	0.21058	5
0.15	35	0.119	385.373	0.23533	5
0.20	35	0.072	341.472	0.29464	5
0.25	35	0.074	321.158	0.26552	5
0.30	35	0.074	354.234	0.30612	5

En primer lugar, notamos que para un factor de ramificación fijo, el tiempo decrece a medida que aumentamos el umbral. Esto se debe a que al aumentar el umbral, la probabilidad de que un objeto pertenezca a un *cluster* es mayor así que no hay que recorrer tantos nodos del árbol.

Sin embargo, los coeficientes Calinski-Harabasz y Silhouette no tienen una tendencia clara. Ni fijando el factor de ramificación ni el valor umbral siguen una tendencia creciente o decreciente, se mueven a

saltos, sin coincidir en sus movimientos (ascendentes o descendentes) los coeficientes. Aunque sí que podemos destacar que los mayores (y por tanto mejores) coeficientes Calinski-Harabasz se obtienen para umbrales bajos, por generar grupos más semejantes.

2.3. Interpretación de la segmentación

Tras realizar la segmentación, nos damos cuenta de que la variable EDADIDEAL, que a priori parecía que fuera a ser de interés y nos ayudaría a dividir en diferentes grupos, toma valores en un rango pequeño en este conjunto particular. Sorprende que este rango sea inferior a los 30 años. Las mujeres que tuvieron su primer hijo con más de 30 años coinciden en que lo ideal habría sido tenerlo antes.

La variable más determinante a la hora de segmentar fue ANOTRABACT que, para los dos algoritmos, nos permitió distinguir varios grupos. Además, los años en la vivienda y en la relación actual nos ayudaron a diferenciar grupos concretos.

3. Caso de estudio 2 - Sin deseo de tener hijos

Para el segundo caso de estudio, decidimos analizar el conjunto formado por las *mujeres que no desean tener hijos*. Así, nos quedamos con las mujeres que respondieron no a la pregunta "¿Le hubiera gustado o le gustaría tener hijos?", esto es, contiene un 6 en la columna `DESEOHIJOS`. El subconjunto elegido está formado por 1806 objetos que deseamos agrupar para obtener más información.

Podemos observar el histograma de la variable `M_NOHIJOS1` en la Figura 8. Esta variable contiene el primer motivo por el que la mujer en cuestión no ha tenido hijos.

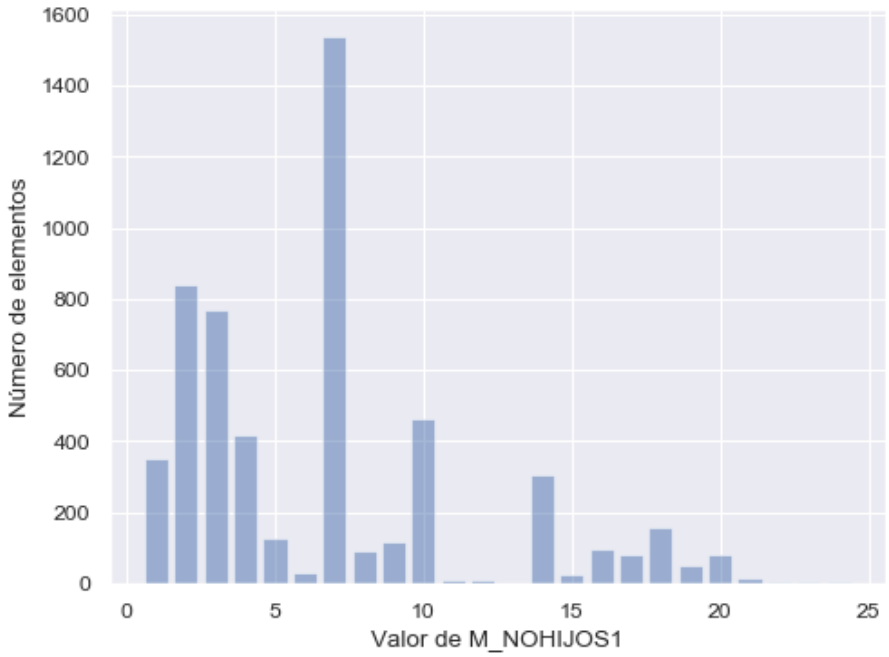

Figura 8: Caso 2 - Opción elegida en la encuesta en la pregunta "Primer motivo por el que no ha tenido hijos"

A continuación se lista el significado de cada una de estas opciones:

0. No me quedaba embarazada o no conseguí llevar un embarazo a término.

1. No he tenido una pareja o ésta no era adecuada.
2. No quiero ser madre o aún no quiero.
3. Deseaba seguir estudiando.
4. Problemas o molestias de salud.
5. Los embarazos, partos y cuidado de los hijos son duros para la mujer.
6. Demasiado joven para tener hijos.
7. Demasiada edad para tener hijos.
8. Entraría en conflicto con mi carrera profesional.
9. Insuficiencia de recursos económicos.
10. Malas condiciones de la vivienda.
11. Exceso de trabajo en el hogar.
12. Carencia o carestía de escuelas infantiles.
13. Por la situación laboral (propia o de la pareja).
14. Temor a que el hijo/a nazca con problemas de salud.
15. Supone perder libertad y no tener tiempo para realizar otras actividades.
16. Por las preocupaciones y problemas que entraña criar a los hijos.
17. Dificultad para conciliar la vida laboral y familiar .
18. Mi pareja no ha querido.
19. No me gusta el modelo de sociedad actual para un niño.
20. Otros motivos.
21. Cuidar de otros familiares.
22. No convivo con mi pareja.
23. No he tenido la oportunidad de formar una familia.

Apoyándonos en los motivos fundamentales que llevan a las mujeres a no desear tener hijos elegiremos las variables utilizadas para el agrupamiento. La principal razón que lleva a las mujeres a no tener hijos es que piensan que son demasiado mayores para ello (motivo 7), seguido de que no quieran ser madre aún o desean seguir estudiando (motivos 2 y 3), las malas condiciones en la vivienda (motivo 10), problemas de salud o preocupación por la salud del bebé (motivos 4 y 14) y que no han tenido pareja o no era adecuada (motivo 1). Las variables elegidas para realizar *clustering* son:

- **EDAD** en años de la mujer entrevistada.
- **SATISFACEVI** representa el grado de satisfacción con la vivienda. Es una escala del 0 al 10, donde 10 significa completamente satisfecho y 5 medianamente satisfecho.
- **EDADIDEAL** es la edad que la entrevistada considera más adecuada para tener el primer hijo.
- **ESTUDIOSA** simboliza el nivel de estudios alcanzados. Es una variable categórica que se representa con valores del 1 al 9, donde 1 denota menos que primaria, 5 educación postsecundaria no superior, 7 y 8 grados universitarios y 9 enseñanzas de doctorado.

Descartamos los motivos relativos al asunto de salud de la madre y del bebé porque no se encuentran variables adecuadas que puedan representar esta información. Para el motivo 1 ("no he tenido una pareja o ésta no era la adecuada") se podría añadir la variable NPARANT por ejemplo, pero no aporta información relevante.

El código necesario para ejecutar los diferentes algoritmos filtrando el conjunto inicial al de las mujeres que no desean tener hijos y agrupando a partir de las variables anteriores se encuentra en el archivo `caso2.py`. Tras ejecutarlo obtenemos los resultados recogidos en la Tabla 6.

Tabla 6: Resultados caso de estudio 2

Algoritmo	Tiempo (s)	Calinski-Harabasz	Silhouette	Número de clusters
KMeans	0.139	920.360	0.29447	5
MeanShift	19.005	56.152	0.32798	2
Birch	0.119	522.766	0.32363	5
DBSCAN	0.037	19.834	0.26498	3
Ward	0.133	743.780	0.22985	5

En cuanto a tiempos de ejecución vuelve a destacar el elevado valor que alcanza MeanShift en comparación al resto de algoritmos, los motivos son los explicados en la Sección 2.

Atendiendo al número de clusters, observamos que los algoritmos en los que no fijamos este valor crean un número muy bajo de clusters. De hecho, atendiendo a la salida del programa comprobamos que en ambos casos están realizando un único *cluster* con el 98 % de los valores. Concluimos que necesitamos adecuar los parámetros de estos algoritmos para que se ajusten al conjunto de datos y puedan generar algún tipo de segmentación.

Destaca también en la Tabla 6 que el índice Silhouette del algoritmo Birch supera al de KMeans y Ward, mientras que su índice Calinski-Harabasz es inferior al de los otros dos. Esto se debe a que de los 5 grupos creados por Birch, dos de ellos son muy pequeños y se ajustan muy bien a los datos que recogen, aunque la distancia entre los elementos del *cluster* sea baja. Que los *clusters* no estén muy poblados perjudica al coeficiente Calinski-Harabasz. El algoritmo que obtiene mayor valor para este coeficiente es KMeans.

A raíz de estas conclusiones iniciales decidimos estudiar el algoritmo KMeans y DBSCAN en el que trataremos de realizar un ajuste de parámetros con el que obtenga un desempeño adecuado.

3.1. Análisis KMeans

Comenzamos estudiando la segmentación obtenida tras aplicar el algoritmo KMeans. En primer lugar, vemos en la Figura 9 que el reparto entre los grupos está igualado, excepto para el *cluster* 0, cuyo tamaño es aproximadamente la mitad que el resto.

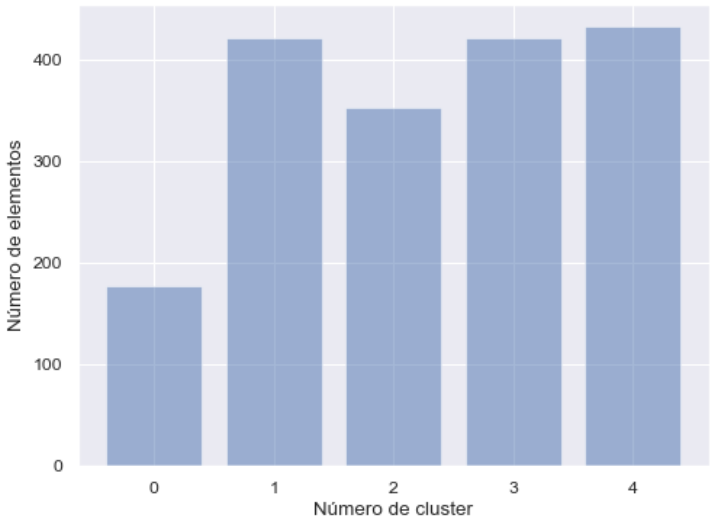

Figura 9: Caso 2 - Tamaño de *cluster* - KMeans

Para hacernos una idea de cómo ha sido el reparto y qué importancia ha tenido cada variable para determinar los grupos, atendemos a la matriz de dispersión, la podemos encontrar en la Figura 10. Las variables EDAD Y ESTUDIOSA nos permiten separar 4 de estos grupos, el quinto se ve separado de los por su valor de SATISFACEVI.

Figura 10: Caso 2 - Matriz de dispersión - KMeans

Para examinar cuáles son las características que determinan la pertenencia a cada grupo, además de a la matriz de dispersión, atenderemos al mapa de calor de las variables que encontramos en la Figura 11. El mapa de calor nos da información sobre el centroide del grupo. Cada cuadrado contiene el valor que toma el centroide del *cluster* correspondiente y el color indica lo alto que es este valor, en función a los valores tomados por esta variable. Sabemos que los *clusters* son convexos, luego, asumiendo concentración, la información obtenida de este gráfico será significativa.

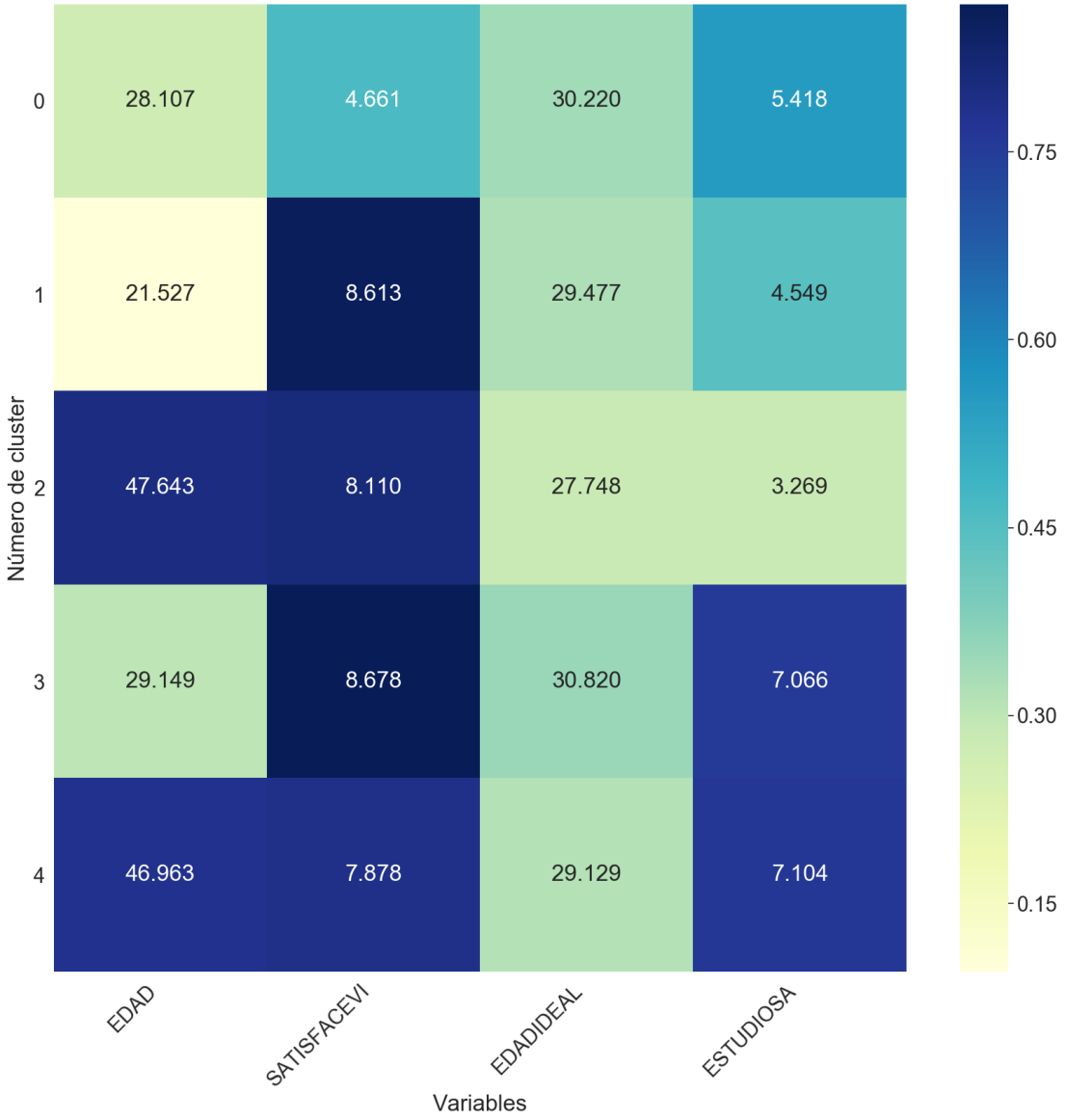

Figura 11: Caso 2 - Mapa de calor - KMeans

Rellenaremos una tabla con las propiedades de cada grupo, sabiendo qué tipo de valores toma para cada variable, podremos comprender mejor la naturaleza de los mismos. En la Tabla 7 se recoge esta información, como el heatmap no nos da rangos, podemos establecerlos a partir de este valor. Por ejemplo, diremos que la edad baja será la que ronde los 20 años (menor que 25), media entre 25 y 35 y alta más de 35. El grado de satisfacción con la vivienda lo consideraremos bajo cuando sea menor que 5 y alto cuando sea mayor que 5. La edad ideal para tener el primer hijo se dirá baja cuando sea menor que 28 años, media cuando esté entre 28 y 35 y alta cuando supere los 35 años.

Tabla 7: Caso 2 - Características de los grupos - KMeans

Cluster	EDAD	SATISFACEVI	EDADIDEAL	ESTUDIOSA
0	Media	Bajo	Media	Educación postsecundaria
1	Baja	Alto	Media	Educación secundaria
2	Alta	Alto	Baja	Primera etapa de educación secundaria
3	Media	Alto	Media	Grado universitario
4	Alta	Alto	Media	Grado universitario

Hay valores que son similares para cuatro de los grupos excepto uno, estos son determinantes para distinguir a dicho grupo de los demás. El grado de satisfacción con la vivienda es en general alto, excepto en el *cluster* 0, formado por mujeres a las que no satisface su vivienda. Notamos que este era el *cluster* de menor tamaño. Por otro lado, la edad considerada ideal para tener el primer hijo, aunque ronde valores similares, es significativamente más baja en el *cluster* 2. Las mujeres de este grupo no finalizaron la educación secundaria. Así, vemos una relación entre nivel de estudios y edad ideal para tener el primer hijo.

Los *clusters* 3 y 4 son parecidos, agrupan a mujeres con un grado universitario, para diferenciarlos atendemos a la variable edad, que en el caso del *cluster* 3 será menor que 35 años y en el 4 mayor. Por último, el *cluster* 1 es el que une a las mujeres más jóvenes, que han estudiado la educación secundaria.

3.1.1. Análisis de parámetros

Aunque los grupos obtenidos son, aparentemente, significativos, podemos comprobar qué habría ocurrido si en lugar de en 5 grupos hubiéramos realizado una segmentación en menos y más valores y ver cómo esto habría afectado a los coeficientes.

Para variar las opciones de este parámetro se utiliza el *script* `caso2-kmeans.py`, en el que el parámetro `n_clusters` se mueve en el rango de enteros $[2, 15]$. En la Tabla 8 vemos los resultados obtenidos para los distintos valores del parámetro.

Tabla 8: Resultados cambio de parámetros KMeans

Número de *clusters*	Tiempo (s)	Calinski-Harabasz	Silhouette
2	0.262	1282.168	0.37326
3	0.024	1106.125	0.35204
4	0.056	996.314	0.28637
5	0.079	921.725	0.29347
6	0.117	852.970	0.27377
7	0.125	809.926	0.26623
8	0.145	784.366	0.26365
9	0.203	755.025	0.25821
10	0.194	724.466	0.24667
11	0.262	692.991	0.24599
12	0.365	670.809	0.23216
13	0.543	643.188	0.22928
14	0.374	629.141	0.23531
15	0.440	612.722	0.22976

Destaca que el tiempo de ejecución no es creciente con el valor del parámetro, como lo fue en el caso de estudio 1. Para solo 2 *clusters* el algoritmo tarda lo mismo en ejecutarse que para crear 11 clusters. Para comparar el desempeño observaremos en la Figura 12 como variaron los coeficientes Calinski-Harabaz y Silhouette al cambiar el número de clusters.

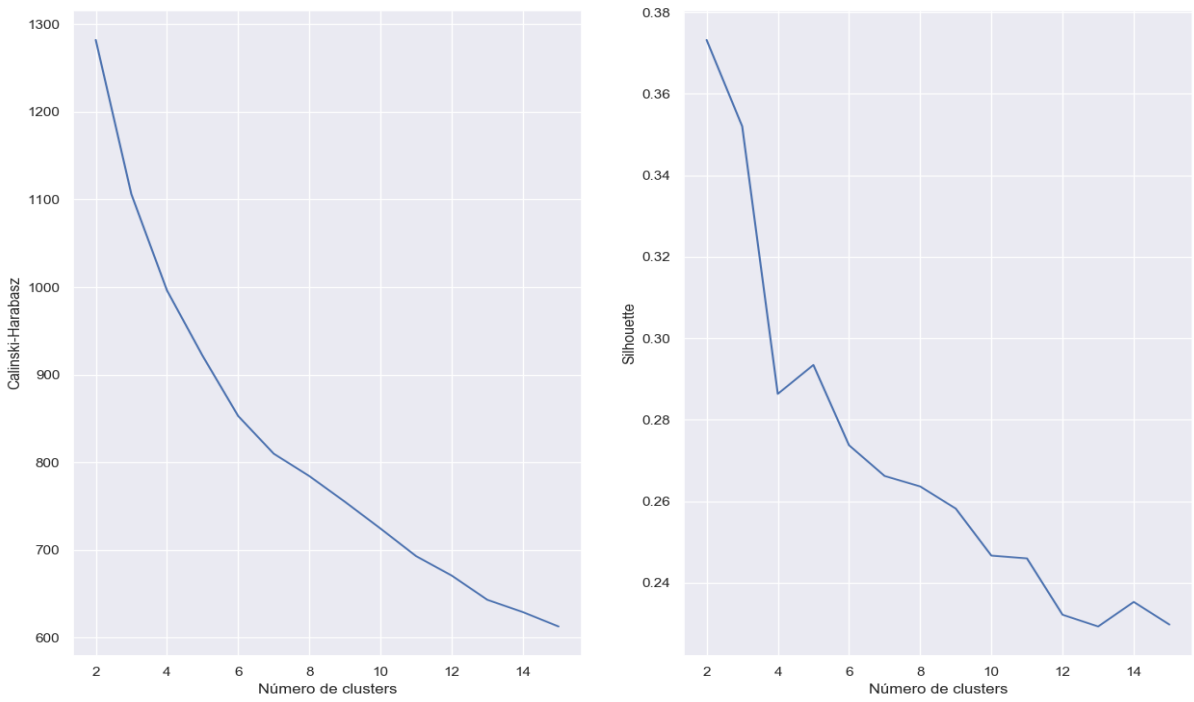

Figura 12: Caso 2 - Comparación coeficientes según el número de *clusters* - KMeans

En esta ocasión la tendencia es similar para ambos coeficientes. A menor número de cluster, mayor valor del coeficiente y mejor agrupamiento. Podemos concluir que podemos dividir a las mujeres que no desean tener hijos en tan solo dos grupos para hacer una buena separación del conjunto. A medida que aumentemos el número de grupos a crear, las diferencias entre ellos no son lo suficientemente grandes y forzar esta división artificial en los grupos perjudica a los coeficientes consultados.

3.2. Análisis DBSCAN

3.2.1. Análisis de parámetros

En este caso, comenzamos con un análisis de parámetros, buscando aquel valor del radio que se adapte mejor al conjunto a estudiar. Para ello se ha implementado el *script* `caso2-dbscan.py`, donde se prueba este algoritmo para diferentes valores del parámetro `eps` y `min_samples`. En la Tabla 9 podemos observar los valores de los índices obtenidos para las distintas combinaciones de estos valores.

Tabla 9: Resultados cambio de parámetros DBSCAN

ε	min_samples	Tiempo (s)	Calinski-Harabasz	Silhouette	Nº de *clusters*	Tam. 1^{er} cluster
0.10	5	0.018	45.981	-0.17517	47	26.19 %
0.15	5	0.033	47.571	0.25514	2	95.68 %
0.20	5	0.033	19.834	0.26498	3	98.23 %
0.25	5	0.042	17.014	0.35890	2	99.17 %
0.30	5	0.050	8.448	0.39872	2	99.72 %
0.10	10	0.016	58.114	-0.18197	30	44.96 %
0.15	10	0.029	53.289	0.21496	2	93.13 %
0.20	10	0.044	44.728	0.31036	2	97.67 %
0.25	10	0.053	21.548	0.35057	2	98.95 %
0.30	10	0.062	11.614	0.41289	2	99.61 %
0.10	15	0.017	59.806	-0.24084	15	65.12 %
0.15	15	0.036	97.678	0.18985	2	88.82 %
0.20	15	0.043	47.849	0.29252	2	97.23 %
0.25	15	0.048	29.719	0.35252	2	98.73 %
0.30	15	0.051	13.665	0.40924	2	99.56 %
0.10	20	0.018	51.446	-0.24913	12	76.08 %
0.15	20	0.036	71.282	0.06423	3	84.27 %
0.20	20	0.033	51.880	0.28787	2	96.90 %
0.25	20	0.040	31.183	0.34573	2	98.62 %
0.30	20	0.050	13.665	0.40924	2	99.56 %

Incluso conociendo el valor de ambos coeficientes para unos parámetros fijos, es complicado decidir qué valores de los mismos tomar.

Descartamos el valor $\varepsilon = 0.1$, pues para cualquier número de muestras mínimo consigue un coeficiente Silhouette negativo. Al ser tan pequeña la distancia que determina cuándo un objeto es densamente alcanzable a partir de otro, se pueden alcanzar menos objetos a partir de uno dado y por ello aumenta el número de clusters, hasta 47 en el peor caso, generando un agrupamiento incorrecto.

La tónica general es la misma para diferente número mínimo de muestras: al aumentar el valor ε crece el coeficiente Silhouette y disminuye el coeficiente Calinski-Harabasz. Intentando conseguir una buena relación entre ambos coeficientes escojo como valores de los parámetros: `eps = 0.2, min_samples = 10`.

Como DBSCAN suele agrupar los objetos que no corresponden a ningún grupo en particular en un único grupo, esto provoca que en el peor caso tengamos dos grupos: uno mayoritario y otro con los *outliers*. Observamos que el valor $\varepsilon = 0.1$ es muy pequeño y genera demasiados grupos, pero $\varepsilon = 0.15$ es muy grande y en la mayoría de los casos genera tan solo dos grupos.

Como la información obtenida es insuficiente, volvemos a ejecutar el *script*, ahora para algunos valores ε en el rango $[0.1, 0.15]$. Los resultados obtenidos esta vez los podemos encontrar en la Tabla 10.

Tabla 10: Resultados cambio de parámetros DBSCAN, $\varepsilon \in [0.1, 0.15]$

ε	min_samples	Tiempo (s)	Calinski-Harabasz	Silhouette	Número de *clusters*	Tam. 1^{er} cluster
0.10	5	0.018	45.981	-0.17517	47	26.19 %
0.11	5	0.022	84.449	-0.15446	19	19.71 %
0.12	5	0.022	94.623	-0.13734	18	20.38 %
0.13	5	0.024	25.566	0.03605	3	92.41 %
0.14	5	0.028	46.138	0.24058	2	94.91 %
0.15	5	0.028	47.571	0.25514	2	95.68 %
0.10	10	0.017	58.114	-0.18197	30	44.96 %
0.11	10	0.022	83.977	-0.17966	16	33.89 %
0.12	10	0.021	131.384	-0.12640	11	25.08 %
0.13	10	0.024	48.759	0.04495	3	85.60 %
0.14	10	0.026	60.746	0.20641	2	91.92 %
0.15	10	0.026	53.289	0.21496	2	93.13 %
0.10	15	0.017	59.806	-0.24084	15	65.12 %
0.11	15	0.019	90.783	-0.17799	16	45.07 %
0.12	15	0.022	123.793	-0.13657	10	34.99 %
0.13	15	0.024	68.439	-0.04772	3	79.01 %
0.14	15	0.032	64.467	0.05823	3	85.27 %
0.15	15	0.025	97.678	0.18985	2	88.82 %
0.10	20	0.016	51.446	-0.24913	12	76.08 %
0.11	20	0.030	90.455	-0.16712	12	60.30 %
0.12	20	0.020	110.723	-0.20185	12	46.23 %
0.13	20	0.023	237.450	0.16647	2	69.27 %
0.14	20	0.033	151.269	0.16536	2	80.40 %
0.15	20	0.029	71.282	0.06423	3	84.27 %

En este caso obtenemos incluso peores resultados, aparecen más casos con coeficiente Silhouette negativo, buscamos acercarnos a esa combinación ε, `min_samples` que proporcione un coeficiente Silhouette positivo y unos *clusters* de tamaño más equilibrado. Seguimos haciendo pruebas (que no se incluyen por falta de interés) hasta decidir que los valores de los parámetros a tomar serán `eps = 0.128, min_samples = 20`.

Sin embargo, aunque los resultados pudieran parecer más esperanzadores, pues para estos parámetros se crean 4 *clusters* diferentes y el primero de ellos contiene al 60 % del conjunto, de los otros tres hay dos que contienen el 2 y 3 % del conjunto, mientras que el que contiene al 36 % es el *cluster* -1 que agrupa a los elementos que no se ajustan a ningún grupo.

Concluimos que DBSCAN no se ajusta a este conjunto de datos. Además, para afinar la búsqueda de parámetros sería necesario algún otro tipo de algoritmo que de forma automática probara con una cantidad suficiente de opciones para conseguir los mejores resultados.

3.3. Birch

Para que el análisis de este caso de estudio no quede incompleto, tras las dificultades con DBSCAN, se decide analizar los resultados aportados por el algoritmo Birch. Ya habíamos notado que los grupos generados son desiguales, en la Figura 13 observamos cómo, aunque nos devuelva 5 clusters, en la práctica solo se han realizado 3 grupos: los correspondientes a los *clusters* 1, 2 y 4.

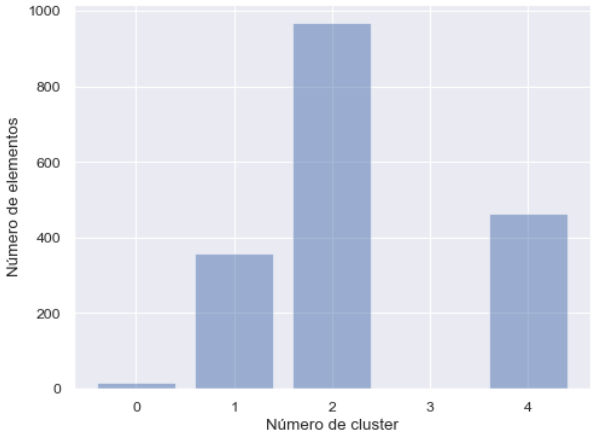

Figura 13: Caso 2 - Tamaño de cada *cluster* - Birch

Comenzamos, como en el resto de casos, observando la matriz de dispersión, que encontramos en la Figura 14. En ella los grupos que destacan son los que ya habíamos señalado el 1, 2 y 4. En una primera observación destacamos que se ven distinguidos por la variable EDAD y EDADESTUDIOSA.

Figura 14: Caso 2 - Matriz de dispersión - Birch

Para determinar las características de los grupos nos apoyamos en un histograma[2], encontrado en la Figura 15, que nos permitirá crear la tabla de características.

[2]Nos decantamos por un histograma en vez de una función de distribución porque varias de las variables son discretas. A pesar de todo se incluye una línea que muestra la distribución seguida.

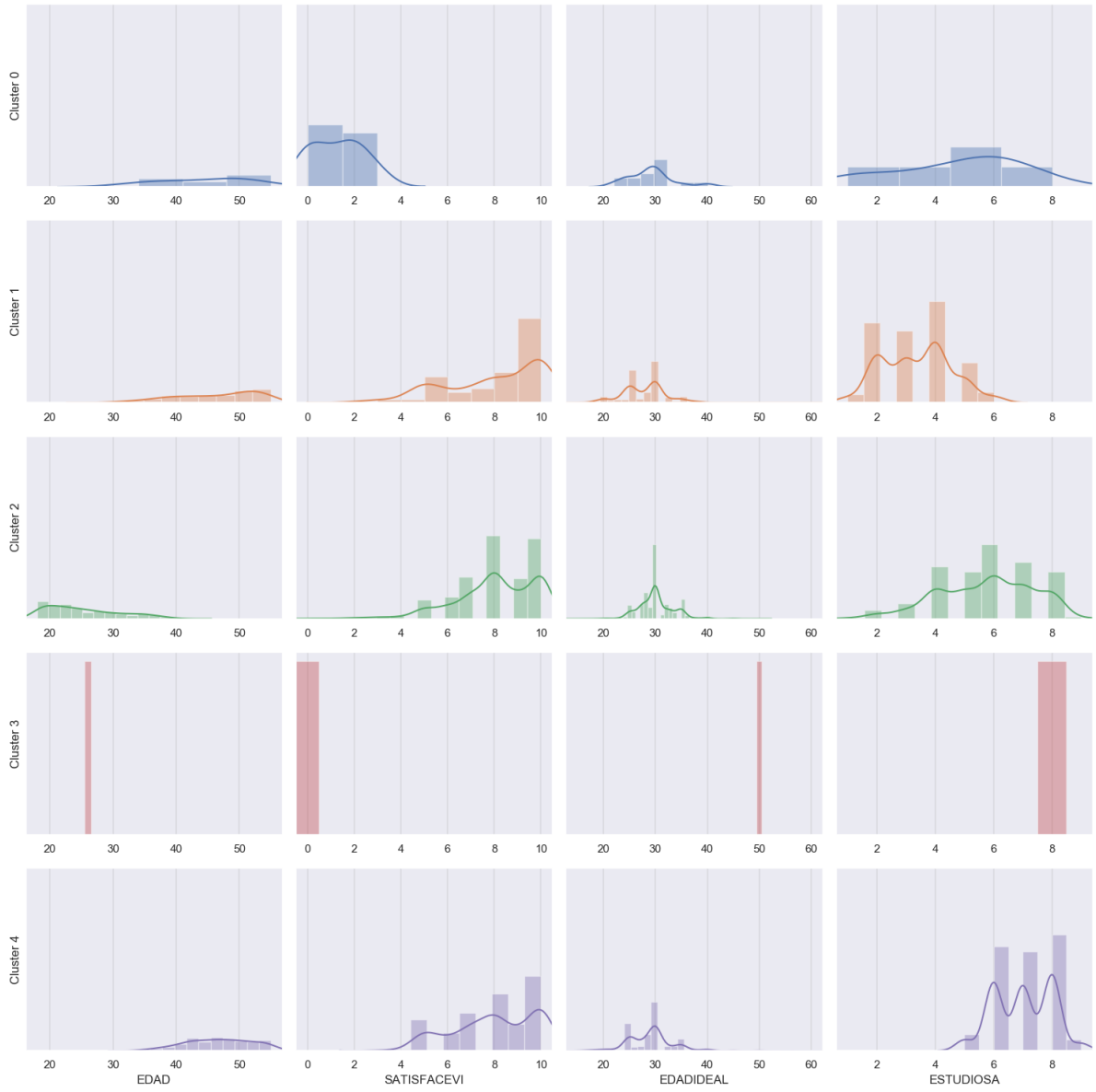

Figura 15: Caso 2 - Histograma - Birch

En el histograma destaca que el *cluster* 3, que contenía muy pocos elementos, se ajusta perfectamente a ellos, es una barra vertical que en un mismo valor representa a todos los elementos del cluster. Sin embargo, no estamos interesados en estos valores.

En la Tabla 11 encontramos las características de los grupos, agrupando los valores de las variables igual que se hizo en el caso del algoritmo KMeans.

Tabla 11: Caso 2 - Características de los grupos - KMeans

Cluster	EDAD	SATISFACEVI	EDADIDEAL	ESTUDIOSA
1	Alta	Alto	Media-baja	Hasta educación secundaria
2	Media - baja	Media	Baja	Educación secundaria - Grado universitario
4	Alta	Alto	Media-baja	Formación profesional - Grado universitario

Así, podemos distinguir los grupos 1 y 4 por el nivel de estudios de las mujeres que pertenecen a ellos,

ambos están formados por mujeres mayores de 35 que están satisfechas con su vivienda y consideran que la edad apropiada para tener el primer hijo es menor a 30 años. El grupo restante tiene un grado de agrado con su vivienda medio, las mujeres son más jóvenes y los niveles de estudio son diferentes en todas ellas, teniendo menor importancia para determinarlo.

3.4. Interpretación de la segmentación

En general, consideramos que ha sido una mala idea elegir tantas variables discretas cuando los algoritmos elegidos se basan en distancias. Los resultados de los algoritmos no han sido demasiado buenos en ningún caso. Además, destaca el esfuerzo empleado en tratar de ajustar DBSCAN y MeanShift (aunque no se incluya, se realizó un análisis similar al realizado con DBSCAN sin obtener ningún progreso).

En cuanto los *clusters* obtenidos en los algoritmos estudiados, aunque los dos algoritmos no generen los mismos grupos, sí que los diferencian a partir de las variables estudiadas, teniendo más peso el nivel de estudios en el caso del algoritmo Birch. El grado de satisfacción con la vivienda, que era de los primeros motivos para que algunas mujeres no desearan tener un hijo, resultó no tener tanta importancia como habíamos supuesto.

Aunque hayamos logrado segmentar y conseguir diversos grupos dentro del conjunto (por nivel de estudios, edad y edad considerada ideal) y lo conozcamos un poco mejor, esta información no nos dice nada sobre el motivo para el que las mujeres no deseen tener hijos.

4. Caso de estudio 3 - Tratamiento de reproducción asistida

En este último caso de estudio queremos analizar el algoritmo jerárquico utilizado, por ello se busca un conjunto inicial más pequeño que nos permita visualizar las tendencias en el dendrograma. Del conjunto total, nos quedamos con aquellas mujeres que se han quedado embarazadas alguna vez antes de la realización de la encuesta (EMBANT == 1) consiguiendo un conjunto formado por 8405 objetos. Dentro de él, nos quedamos con las mujeres que se han sometido a algún tratamiento de reproducción asistida (TRAREPRO == 1). Tras filtrar con las dos condiciones el conjunto a estudiar está compuesto por 598 objetos. Así, podemos reumir este caso de estudio como *mujeres que ya han estado embarazadas y se han sometido a algún tratamiento de reproducción asistida*.

Las variables utilizadas para agrupar son variadas, tienen relación con el propio tratamiento, la situación de la pareja y la posición laboral[3]. Son las siguientes:

- **EDADTRAREPRO** es la edad en la que la entrevistada comenzó a someterse a tratamientos de fecundidad.
- **NTRABA** representa el número de años que la mujer lleva en el empleo actual.
- **TEMPRELA** es el número de años de la relación de pareja actual.
- **TDYO** indica el porcentaje de tareas domésticas que realiza la entrevistada.

Tras ejecutar el código encontrado en `caso3.py` obtenemos los resultados encontrados en la Tabla 12.

[3]Hay variables que podrían parecer también de interés, como la edad o los ingresos, pero presentan indicios de correlación con las escogidas. Es por ello que se eligen solo estas cuatro variables por considerarlas lo suficientemente significativas.

Tabla 12: Resultados caso de estudio 3

Algoritmo	Tiempo (s)	Calinski-Harabasz	Silhouette	Número de clusters
KMeans	0.062	141.581	0.18194	5
MeanShift	2.335	-	-	1
Birch	0.027	84.632	0.15428	5
DBSCAN	0.009	10.655	0.32264	2
Ward	0.011	112.646	0.14388	5

Destaca negativamente que MeanShift se queda con un único grupo, por lo que no tiene sentido el cálculo de los coeficientes que indicarán cómo de buena o mala fue la segmentación. Se trató de conseguir mejores resultados con este algoritmo variando el radio a partir del valor `bandwidth` estimado (para ello se utilizó el *script* `caso3-meanshift.py`) y no se logró segmentar el grupo. Es posible que por la distribución de los datos, los puntos estén muy cerca y fueran densamente alcanzables con los radios probados.

Notamos que DBSCAN tampoco consigue buenos resultados, genera dos grupos (uno de los cuales sabemos que contendrá a todos los objetos que no se ajusten a ningún grupo).

En los algoritmos restantes, como hemos fijado el número de *clusters* no se produce el problema anterior. Entre ellos, decidimos analizar en profundidad KMeans, pues obtiene mejores resultados en ambos coeficientes, y Ward, la elección de un conjunto pequeño tenía como finalidad facilitar el análisis mediante este algoritmo jerárquico.

4.1. Análisis KMeans

Para comenzar a estudiar los grupos generados, nos planteamos en primer lugar cómo es el tamaño de estos grupos. Para conocerlo, atenderemos a la gráfica de barras de la Figura 16.

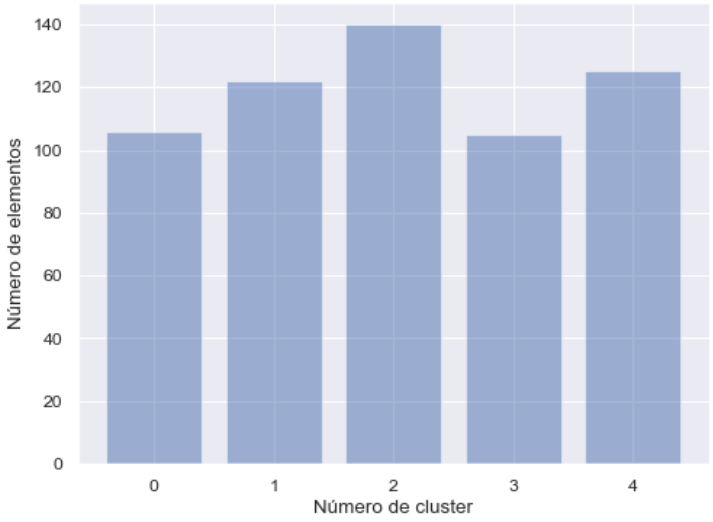

Figura 16: Caso 3 - Tamaño de los *clusters* -KMeans

Los grupos son semejantes en tamaño, la mayor diferencia es de menos de 40 objetos. Para discernir qué variables son las que definen cada grupo observamos la matriz de dispersión, representada en la Figura 17.

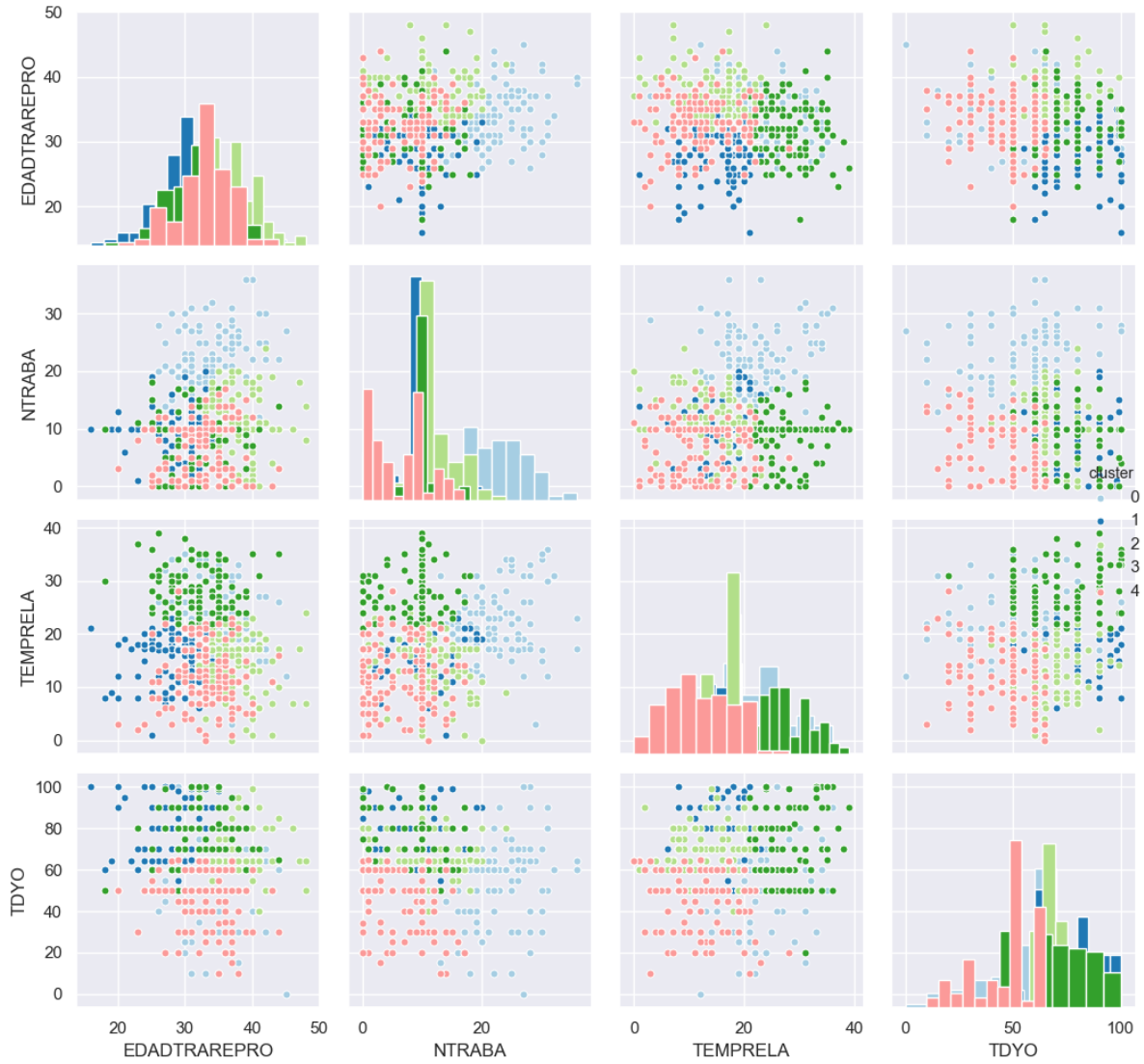

Figura 17: Caso 3 - Matriz de dispersión - KMeans

Los grupos son más difíciles de distinguir que en otras ocasiones, puede que otro número de *clusters* se adapte mejor a este conjunto. Aún así, podemos ver que la variable NTRABA nos ayuda a separar el *cluster* 0, EDADTRAREPRO y TDYO los grupos 3 y 4, TEMPRELA y EDADTRAREPRO los grupos 1 y 2. Para aclararnos con qué características realmente determinan cada grupo realizaremos la Tabla 13, apoyándonos en el diagrama de cajas de la Figura 18.

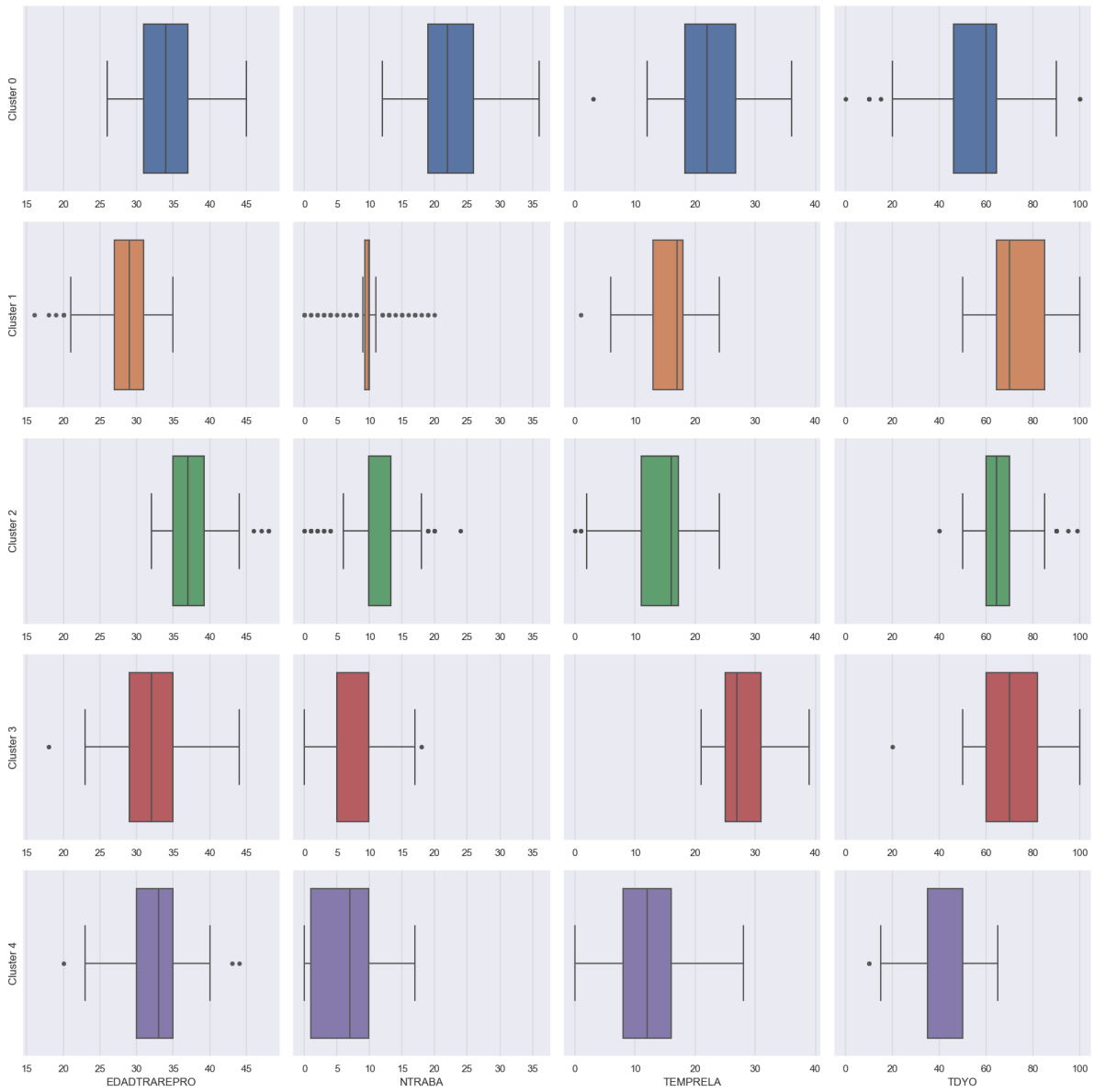

Figura 18: Caso 3 - Diagrama de cajas - KMeans

Tabla 13: Caso 3 - Características de los grupos - KMeans

Cluster	EDADTRAREPRO	NTRABA	TEMPRELA	TDYO
0	31-37	19-26	18-27	45-63
1	27-31	9-10	13-18	62-87
2	35-39	10-14	11-17	60-70
3	27-35	5-10	25-32	60-82
4	30-35	1-10	7-16	37-50

El *cluster* 0 agrupa a las mujeres que llevan más tiempo en su puesto de trabajo actual, en torno a 20 años. Este grupo probablemente contenga a las mujeres de mayor edad. Las relaciones de las entrevistadas son estables, llevan entre 20 y 30 años con la misma pareja y el reparto de tareas es equitativo, facilitando la convivencia. Además, las mujeres de este grupo comenzaron con los tratamientos de reproducción durante sus 30 años.

Es en el *cluster* 1 donde se encuentran las mujeres que realizan mayor porcentaje de las tareas del hogar. Está formado por mujeres con relaciones que han durado unos 15 años y empezaron a trabajar en el puesto actual hace 10. Estas mujeres comenzaron el tratamiento de fecundidad cuando tenían aproximadamente 29 años, siendo el grupo en el que más jóvenes empezaron este tipo de tratamientos.

Las mujeres del *cluster* 2 son las que más tarde comenzaron con un tratamiento de reproducción asistida. Es en este hecho y en el reparto de las tareas, en lo que más se diferencian del *cluster* 1, pues en los años trabajando y en los de la relación tienen valores similares.

En *cluster* 3 está formado por mujeres que llevan muchos años con sus parejas, entre 25 y 30 años, pero relativamente poco tiempo en el trabajo actual, entre 5 y 10 años. Además, es un grupo en el que el reparto de tareas es poco equitativo. Esto se puede deber a que anteriormente la mujer se ocupara de las tareas del hogar (por no tener trabajo) y al empezar con su trabajo no dejara de hacerlo. El rango de edad en el que las entrevistadas de este grupo se sometieron a tratamientos de reproducción es similar al del *cluster* 4, entre 30 y 35 años. Posiblemente tras haber intentado un embarazo natural y que este no se produjera.

Por último, el *cluster* 4 recoge a las mujeres que llevan trabajando en el puesto actual menos de diez años y su relación de pareja ha durado menos de 15 en muchos casos. Sospechamos que las mujeres de este grupo son más jóvenes que las de los otros grupos. La diferencia principal con el resto de grupos es que en este caso el porcentaje de tareas domésticas realizadas por la entrevistada no supera la mitad de las tareas.

4.1.1. Análisis de parámetros

Veamos, dentro de KMeans, cómo de buena o mala fue la elección de parámetros. Para ello ejecutamos el *script* `caso3-kmeans.py` en el que varía el número de *clusters* y calcula los coeficientes Calinski-Harabasz y Silhouette para medir el desempeño.

Observamos en la Tabla 14 los resultados conseguidos según el valor de la variable `n_clusters`.

Tabla 14: Resultados cambio de parámetros KMeans

Número de *clusters*	Tiempo (*s*)	Calinski-Harabasz	Silhouette
2	0.042	171.607	0.23063
3	0.024	167.222	0.20796
4	0.036	152.306	0.18711
5	0.061	141.784	0.18321
6	0.059	136.510	0.19751
7	0.079	132.802	0.20224
8	0.108	129.802	0.20587
9	0.127	124.104	0.19464
10	0.121	121.246	0.20349
11	0.143	114.839	0.19463
12	0.189	113.748	0.20430
13	0.193	111.573	0.21155
14	0.192	109.976	0.21027
15	0.253	108.101	0.20806

Como en ocasiones anteriores, el aumento en el número de *clusters* provoca un incremento en el tiempo de ejecución. Para facilitar la comparación de ambos coeficientes observamos, en la Figura 19 las variaciones en los mismos.

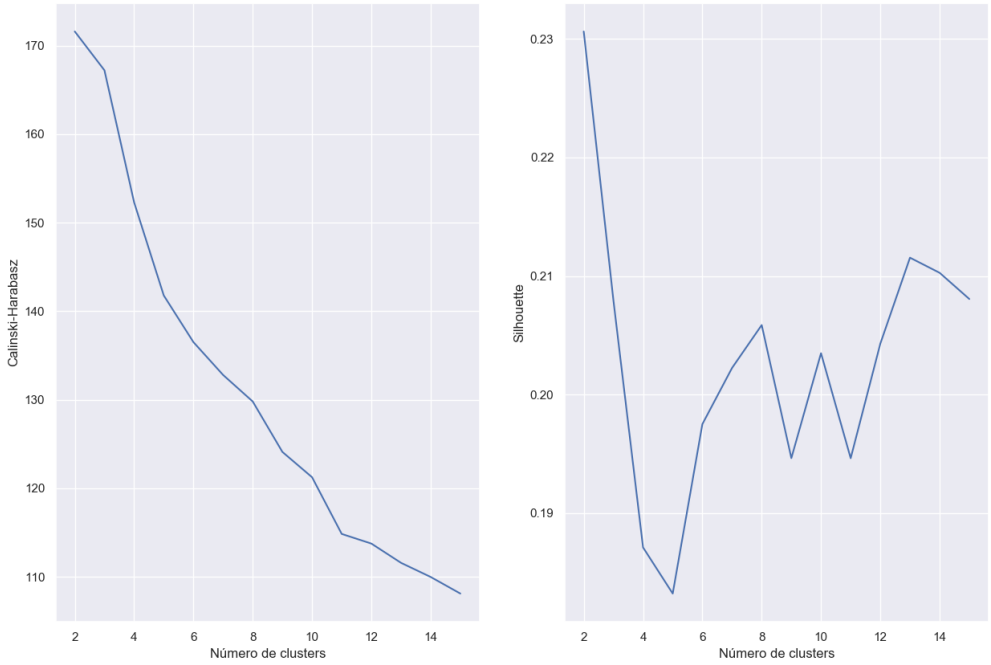

Figura 19: Caso 3 - Comparación de coeficientes según el número de *clusters* - KMeans

El coeficiente Calinski-Harabasz decrece a medida que aumenta el número de clusters. Como los *clusters* son, posiblemente, cercanos, las diferencias entre ellos son pequeños y este coeficiente se ve afectado. El coeficiente Silhouette, a pesar de alcanzar el máximo también para dos clusters, alcanza máximos locales en 8, 10 y 13 clusters, probablemente por conseguir grupos en los que los elementos estén muy cerca de su centro.

4.2. Análisis Ward

El análisis a realizar a partir de un algoritmo jerárquico es diferente a los realizados para los algoritmos anteriores. En este tipo de algoritmos no tiene tanto interés los grupos que se generan sino el cómo se generan, cuáles son las variables por las que se empieza agrupando y separando a los objetos del conjunto.

El algoritmo jerárquico elegido, Ward, es de tipo aglomerativo. Al principio cada objeto forma su propio *cluster* y se va iterando uniendo los *clusters* más cercanos. Podemos usar dendrogramas, que representan estas uniones, para saber cómo ha sido el proceso de agrupar los objetos y qué variables han tenido más peso para ello.

En la Figura 20 observamos un dendrograma truncado (a 10 hojas) que nos permite distinguir varios niveles, uno de cada color. Los valores en el eje de abscisas indica la distancia a la que se encuentran los enlaces. Los números en el eje de ordenadas indican el número de elementos en el *cluster* correspondiente.

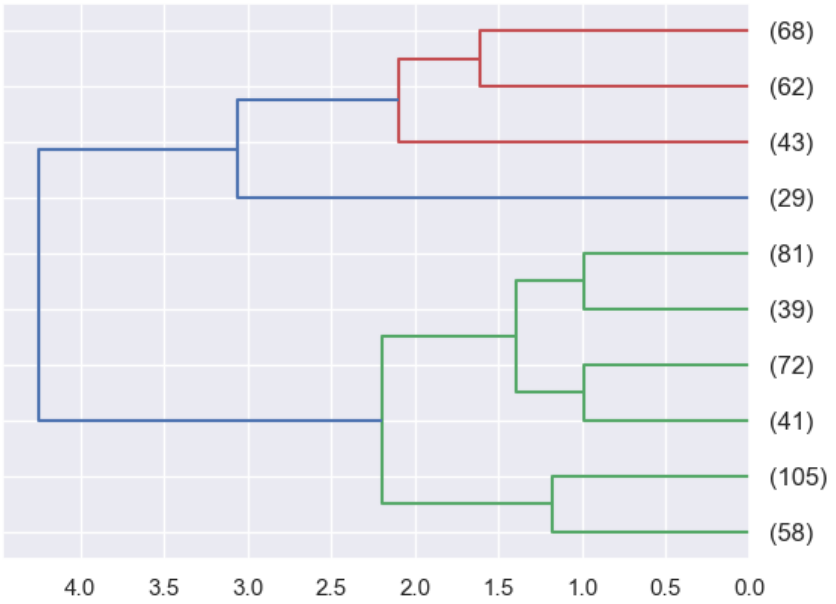

Figura 20: Caso 3 - Dendrograma - Ward

Comenzamos interpretando el dendrograma de arriba a abajo. La primera hoja contiene 68 elementos, no la juntamos con la siguiente hoja seguidamente, pues están a una distancia mayor de 1.5. Por un lado, las primeras hojas en agruparse serán las de 81 y 39 elementos y por otro las de 72 y 42, ambas a una distancia de 1.0. En la parte baja del dendrograma vemos representado en verde un grupo mayor, con uniones más cercanas que en el grupo formado por la rama azul superior del diagrama. Esto no nos ayuda a comprender nuestros datos, por ello, utilizamos un dendrograma combinado con un mapa de calor para tratar de obtener información específica del problema. En la Figura 21 se encuentra este diagrama combinado para este caso de estudio.

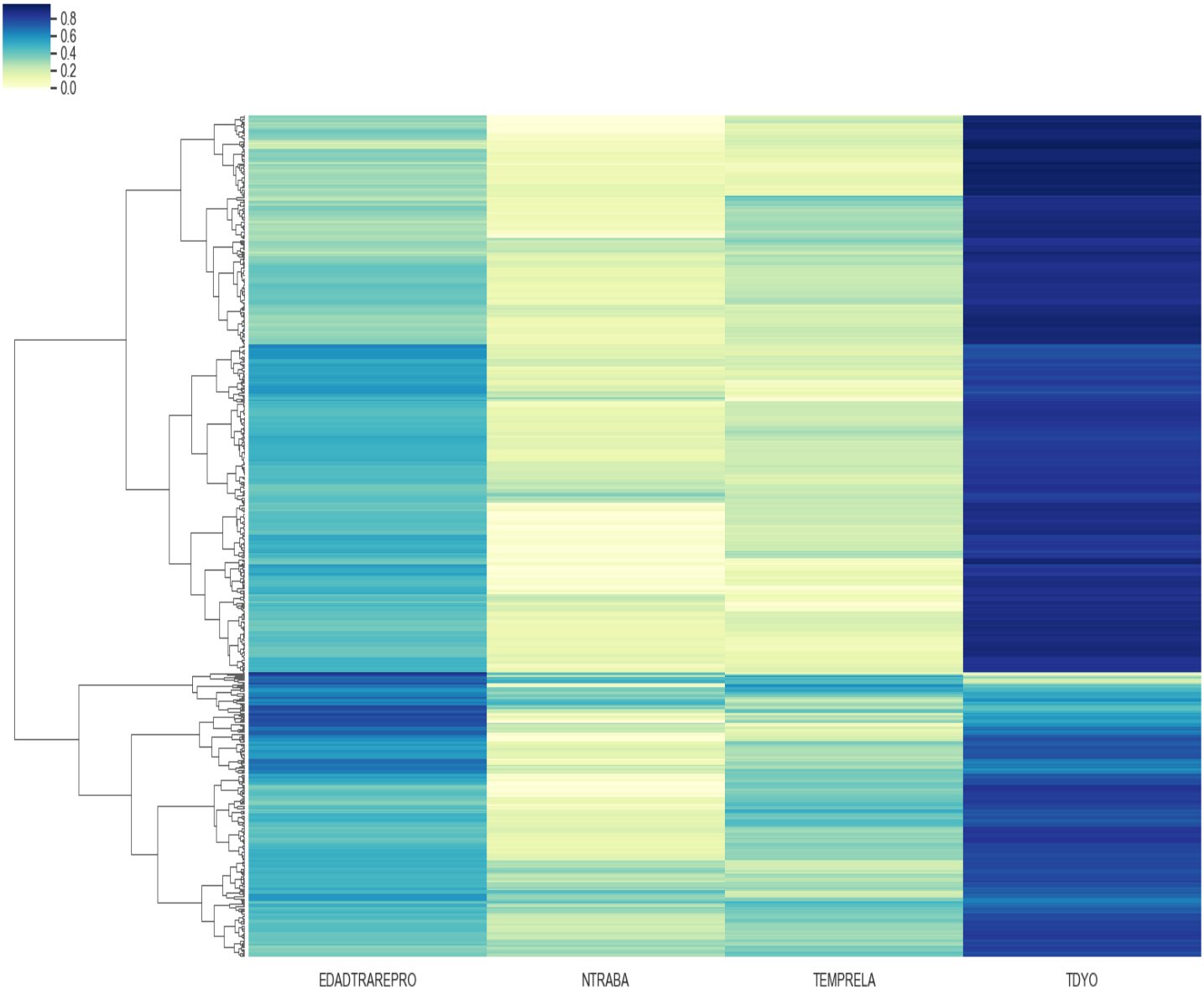

Figura 21: Caso 3 - Dendrograma combinado con un mapa de calor - Ward

Vamos estudiando los grupos formados de mayor a menor. El porcentaje de tareas domésticas realizadas por la entrevistada nos permite distinguir dos grandes grupos, la rama superior y la rama inferior del dendrograma. En la rama superior este porcentaje será superior al 70 %, mientras que en la rama inferior es menor que el 70 %.

En la rama superior se continúa dividiendo según la edad a la que se comenzó el tratamiento de reproducción, se distingue entre las que lo empezaron más jóvenes (entre estas se pasa al siguiente según si empezaron recientemente la relación y las que empezaron hace un tiempo "medio") y las que lo empezaron a una edad media (la siguiente ramificación es en función del número de años en el empleo actual).

La rama inferior se vuelve a ramificar según el porcentaje de tareas domésticas realizadas. En la rama en que este porcentaje es inferior se puede seguir descendiendo, según este valor o según el tiempo de la relación actual. En la rama en que este porcentaje es medio se divide según la edad a la que se comenzó el tratamiento de reproducción asistida y en los siguientes nodos, según los años que se lleve en el trabajo actual.

Concluimos que las variables más significativas para diferenciar los grupos dentro de las mujeres que se han sometido a tratamientos de reproducción asistida son: TDYO y EDADTRAREPRO. La primera variable sorprende por su poder de segmentación en este grupo. La segunda está directamente relacionada con este grupo, así que era esperable que nos ayudara a distinguir grupos.

5. Birch

Como ya se ha comentado, en los algoritmos jerárquicos no tiene sentido realizar un análisis de parámetros como en los anteriores casos, fijando el número de *clusters* por ejemplo. Por ello, para cubrir la parte de análisis de análisis de parámetros se utiliza el algoritmo Birch. Además, en los casos anteriores Birch obtuvo unos coeficientes similares a los de KMeans, mientras que en este caso sus coeficientes fueron inferiores. Se tratará de determinar qué parámetros son los que nos permiten alcanzar unos índices equiparables.

5.1. Análisis de parámetros

Utilizaremos el *script* `caso3-birch.py` para probar diferentes combinaciones de los parámetros umbral y factor de ramificación utilizados en Birch. Vemos los resultados obtenidos en la Tabla 15.

Tabla 15: Resultados cambio de parámetros Birch

Umbral	Factor de ramificación	Tiempo (*s*)	Calinski-Harabasz	Silhouette	Nº de *clusters*
0.10	15	0.112	107.808	0.13658	5
0.15	15	0.075	96.142	0.14726	5
0.20	15	0.071	114.154	0.15479	5
0.25	15	0.041	84.632	0.15428	5
0.30	15	0.041	94.019	0.17670	5
0.10	20	0.062	111.892	0.16311	5
0.15	20	0.057	94.813	0.15713	5
0.20	20	0.051	107.255	0.13369	5
0.25	20	0.031	84.632	0.15428	5
0.30	20	0.030	94.019	0.17670	5
0.10	25	0.059	111.776	0.15187	5
0.15	25	0.056	90.949	0.12529	5
0.20	25	0.045	86.851	0.14934	5
0.25	25	0.030	84.632	0.15428	5
0.30	25	0.030	94.019	0.17670	5
0.10	30	0.060	110.359	0.12301	5
0.15	30	0.054	86.219	0.10800	5
0.20	30	0.042	102.221	0.13141	5
0.25	30	0.030	84.632	0.15428	5
0.30	30	0.031	94.019	0.17670	5
0.10	35	0.060	93.762	0.11711	5
0.15	35	0.054	104.590	0.14267	5
0.20	35	0.035	99.845	0.13213	5
0.25	35	0.031	84.632	0.15428	5
0.30	35	0.031	94.019	0.17670	5

Observamos que, como ocurrió en el caso de estudio 1, para un factor de ramificación fijo, el tiempo de ejecución disminuye. Sin embargo, si fijamos el valor del umbral, a medida que aumenta el factor de ramificación el tiempo solo varía al cambiar de 15 a 20, esto se debe a que no estamos usando estas ramas extra que conseguimos al aumentar el factor de ramificación.

Atendiendo al índice Calinski-Harabasz, notamos que en algunos casos para valores umbral menores consigue mayor valores, pero a veces también para umbrales intermedios. En cualquier caso no logra alcanzar al coeficiente Calinski-Harabasz de KMeans, 141.581. El coeficiente Silhouette tampoco llega al valor conseguido por KMeans, 0.18194, siendo el valor máximo 0.17670 (logrado en los casos con mayor umbral).

5.2. Interpretación de la segmentación

En primer lugar, tras analizar los grupos formados por KMeans notamos que pocas mujeres jóvenes empiezan un tratamiento de fecundidad.

Podemos distinguir las mujeres de este conjunto en dos grandes grupos según el porcentaje de tareas domésticas que realicen y a partir de ahí ir atendiendo al resto de variables para conseguir grupos más significativos.

Referencias

[1] *Clustering*. URL: http://scikit-learn.org/stable/modules/clustering.html (Último acceso 06-12-2019).

[2] *Creating annotated heatmaps — Matplotlib 3.1.1 documentation*. URL: https://matplotlib.org/gallery/images_contours_and_fields/image_annotated_heatmap.html#sphx-glr-gallery-images-contours-and-fields-image-annotated-heatmap-py (Último acceso 08-12-2019).

[3] *Example gallery — seaborn 0.9.0 documentation*. URL: https://seaborn.pydata.org/examples/index.html (Último acceso 08-12-2019).

[4] *INEbase / Demografía y población /Fenómenos demográficos /Encuesta de fecundidad / Enlaces relacionados*. URL: https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177006&menu=enlaces&idp=1254735573002 (Último acceso 06-12-2019).

[5] *Matplotlib Bar chart*. en. URL: https://pythonspot.com/matplotlib-bar-chart/ (Último acceso 08-12-2019).

[6] *The Lifecycle of a Plot — Matplotlib 3.1.1 documentation*. URL: https://matplotlib.org/3.1.1/tutorials/introductory/lifecycle.html (Último acceso 08-12-2019).