Práctica-2: Análisis Relacional mediante Segmentación.

UNIVERSIDAD DE GRANADA
E.T.S.I. INFORMÁTICA Y TELECOMUNICACIÓN

UNIVERSIDAD DE GRANADA

Departamento de Ciencias de la Computación e Inteligencia Artificial

Inteligencia de negocio (2019-2020)

Daniel Bolaños Martínez
danibolanos@correo.ugr.es
Grupo 2 - Jueves 09:30h

Índice

1. Introducción.

El objetivo de la práctica consiste en el estudio de técnicas de aprendizaje no supervisado para análisis relacional mediante segmentación. Aplicaremos distintos algoritmos de *clustering* sobre el conjunto de datos y realizaremos un estudio en profundidad a raíz de los resultados obtenidos.

El conjunto de datos del que disponemos corresponden a microdatos recogidos por el Instituto Nacional de Estadística (**INE**) en 2018 sobre la última encuesta de fecundidad. Disponemos de un conjunto de 14.556 respuestas de mujeres a una encuesta con 463 variables sobre datos personales, datos biográficos, hogar, vivienda, padres, relaciones de pareja, hijos, fecundidad, estudios, empleo y creencias.

Podemos estructurar las variables de las que disponemos de la siguiente forma:

- **Características Geográficas:** código de la comunidad autónoma donde reside.
- **Características Demográficas:** tenemos los datos relativos a edad, sexo, nacionalidad (de sí misma, pareja e hijos), etc.
- **Características Socioeconómicas:** tipo de unión de pareja (matrimonio, pareja de hecho registrada o pareja de hecho sin registrar), niveles de estudios completados, ocupación, relación de la actividad económica, religión y creencias, etc.
- **Fecundidad:** datos relativos a embarazo actual, historial de embarazos, fertilidad, uso de anticonceptivos y número deseado de hijos.

Elegiremos 3 muestras de la población del conjunto de datos y haremos el estudio utilizando los algoritmos de *clustering* **K-Means, Mean Shift, Agglomerative Clustering, Birch y DBSCAN**. Compararemos los resultados obtenidos para cada uno de ellos calculando tiempos de ejecución, número de clusters y valor de las métricas. Finalmente, mostraremos los gráficos de los dos que obtengan mejores resultados e interpretaremos la segmentación obtenida para cada caso.

El conjunto de datos ha sido tratado por el profesor para eliminar variables que no ofrecían información relevante o podrían aportar ruido al dataset (observaciones). Además, contamos con una plantilla que podemos utilizar como base para ejecutar los algoritmos.

2. Casos de Estudio.

2.1. Caso de estudio 1.

Para el primer caso, estudiaremos el conjunto de mujeres que actualmente no tienen un trabajo remunerado, es decir, o están desempleadas o trabajan como amas de casa y que tienen al menos 1 hijo biológico. Las variables que vamos a elegir se corresponden con:

- **EDADHIJO1**: Edad cuando tuvo el primer hijo biológico.
- **EDADIDEAL**: Edad ideal para tener al primer hijo.
- **ESTUDIOSA**: Valor ordinal que representa el nivel de estudios alcanzados. (**1**-menos de primaria, **2**-primaria, **3**-primera etapa secundaria, **4**-segunda etapa seundaria, **5**-educación postsecundaria no superior, **6**-formación profesional, **7 o más**-grado universitario/doctorado).
- **NHIJOS**: Número de hijos suyos o de su pareja.
- **EDAD**: Edad en años cumplidos.
- **TEMPRELA**: Número de años de la relación de pareja actual.

El objetivo de este caso de estudio consiste en distinguir grupos de mujeres de diferentes edades y relacionar su histórico de fecundidad con datos como su relación de pareja o grado de estudios obtenido, así como el deseo de las mismas de haber retrasado su primer embarazo o haber tenido más hijos. Después intentaremos sacar conclusiones acerca del perfil de mujer de cada grupo.

Para cada caso de estudio, he obtenido una muestra de la población total que rondará entre 400 y 8000 ejemplos. En este caso, estudiaremos un total de 2774. El conjunto seleccionado se ha obtenido a partir del siguiente código:

```
subset = datos.loc[(datos['TRABAJAACT']==6) & (datos['NHIJOBIO']>=1)]
usadas = ['EDADHIJO1', 'EDADIDEAL', 'ESTUDIOSA', 'NHIJOS', 'EDAD', 'TEMPRELA']
X = subset[usadas]
```

Hemos seleccionado como ya hemos indicado antes, las mujeres que no tienen actualmente un trabajo remunerado y con 1 o más hijos biológicos. Además, las 6 variables sobre las que se ha realizado el estudio, son numéricas y en el caso de *ESTUDIOSA* ordinal. El valor de los estudios sigue una distribución de orden por lo que podemos utilizarla sin problema.

A continuación, se mostrarán los resultados obtenidos por los algoritmos en este caso de estudio. Utilizaremos una tabla comparativa donde incluiremos el nombre de los algoritmos, los resultados obtenidos por las métricas utilizadas (Calinski-Harabaz y Silhouette) y los tiempos de ejecución de los mismos. Las métricas utilizadas se interpretarán de la siguiente forma:

- **Calinski-Harabaz**: se basa en el concepto de densidad y de como de bien están separados los clusters. Especifica la relación entre la dispersión entre los clusters y la dispersión dentro de los clusters. Nos indicará si estamos usando un buen número de clusters para un algoritmo en concreto. El número óptimo de clusters es la solución con el valor de índice Calinski-Harabasz más alto. [3]
- **Silhouette**: es una medida que indica como de similar es un objeto respecto a su propio grupo (cohesión) en comparación con otros grupos (separación). Toma valores entre -1 y +1, donde un alto valor indica que el objeto es bastante similar a su grupo y muy diferente a los de otros cluster. Si el valor es cercano a 1, la configuración de los cluster es apropiada, si no, habrá más o menos clusters de los necesarios. [2]

Los algoritmos elegidos han sido los siguientes: **K-Means, MeanShift, DBSCAN, Birch y Agglomerative Clustering** (clustering jerárquico). A continuación, se hará un estudio sobre varias modificaciones para los algoritmos descritos desarrollando más a fondo los dos que mejores resultados obtengan.

A la hora de elegir las mejores versiones de cada algoritmo, nos basaremos en el coeficiente de Calinski-Harabaz (**CH**) que como hemos explicado antes, tendrá mayor valor si estamos usando el número adecuado de clusters.

2.1.1. K-Means.

Ejecutaremos diferentes versiones de **K-Means** modificando el parámetro *n_clusters=X* donde X tomará diferentes valores entre 2 y 5, fijaremos como *random_state* una semilla con el valor de mi DNI y *n_init=5*. El código utilizado ha sido el siguiente:

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

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	1226.305547	0.271982	0.024908
n_clusters	3	1246.246928	0.284122	0.029773
n_clusters	4	1088.760379	0.243528	0.034425
n_clusters	5	1034.365773	0.239813	0.042675

Tabla 1: Tabla modificaciones K-Means para el Caso 1.

Podemos observar que la versión que obtiene mejores resultados para la métrica **CH** y **SH**, es la que obtenemos con 3 clusters. En esta versión, la proporción de cada cluster se ha asignado de la siguiente manera:

```
1:  1128 (40.66%)
2:   829 (29.88%)
0:   817 (29.45%)
```

Para este caso, se mostrarán 4 gráficas realizadas y se hará un estudio sobre los resultados obtenidos. Las gráficas obtenidas para este caso, serán **Scatter Matrix, Heatmap, KPlot y BoxPlot**.

Como podemos observar tanto en la Figura 1 como en la Figura 2 obtenemos tres grupos bastantes diferenciados en lo relativo a las variables (EDADHIJO1, ESTUDIOSA, EDAD y TEMPRELA). Tenemos bastante buena proporción entre clusters, siendo el 0 y 2 de tamaños similares y el cluster 1 un poco superior (contiene un 11 % más de las muestras), por lo que podemos decir que el tamaño de los clusters está balanceado.

Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster. **KPlot** muestra la densidad que toma cada variable para cada individuo en cada cluster, por lo que podemos estudiar la distribución con mayor precisión que la que nos aporta **Heatmap** al representar la media de los centroides de las distribuciones.

A su vez, con **BoxPlot** podremos observar la cantidad de outliers que tiene el caso estudiado, si los tuviese, para este algoritmo. Que en nuestro caso y como hemos visto en la proporción de cada cluster, será poco relevante.

Figura 1: Scatter Matrix para K-Means con 3 clusters.

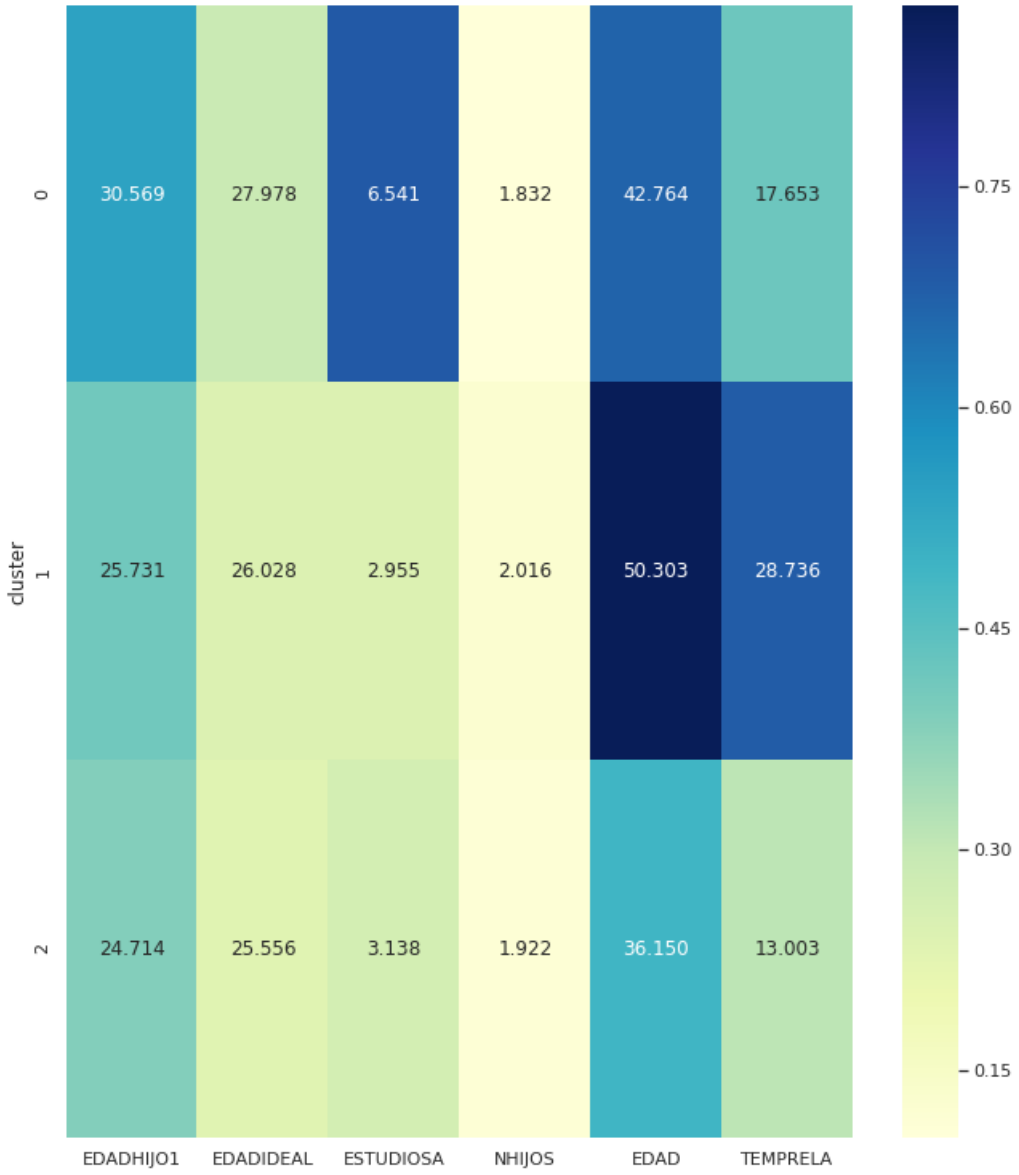

Figura 2: HeatMap para K-Means con 3 clusters.

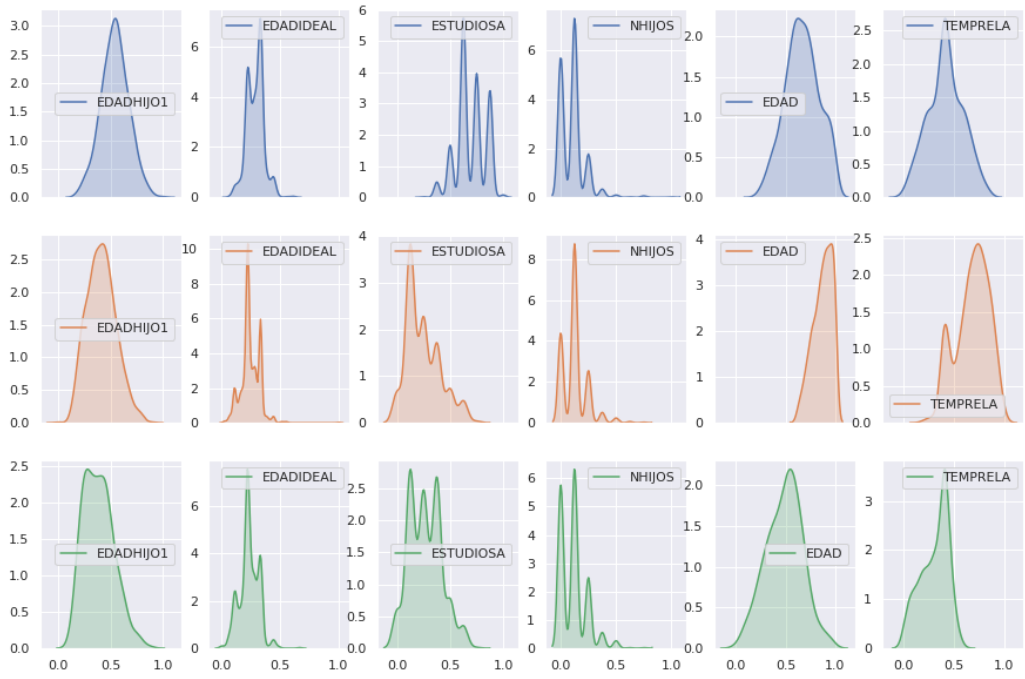

Figura 3: KPlot para K-Means con 3 clusters.

Diferenciaremos 3 grupos de mujeres en las que se caracterizan por las siguientes características:

- **Cluster 0**: Mujeres sobre los 40 años que tuvieron su primer hijo biológico a una edad mayor que la media (30 años). Se caracterizan por tener un mayor nivel de estudios y una relación de pareja duradera.
- **Cluster 1**: Mujeres cuya edad ronda los 50 años o más, con un nivel de estudios bajo y una relación de pareja muy larga. Se caracterizan por tener más hijos que la media.
- **Cluster 2**: Mujeres con 36 años de media, con un nivel medio de estudios y una relación de pareja más corta que la media.

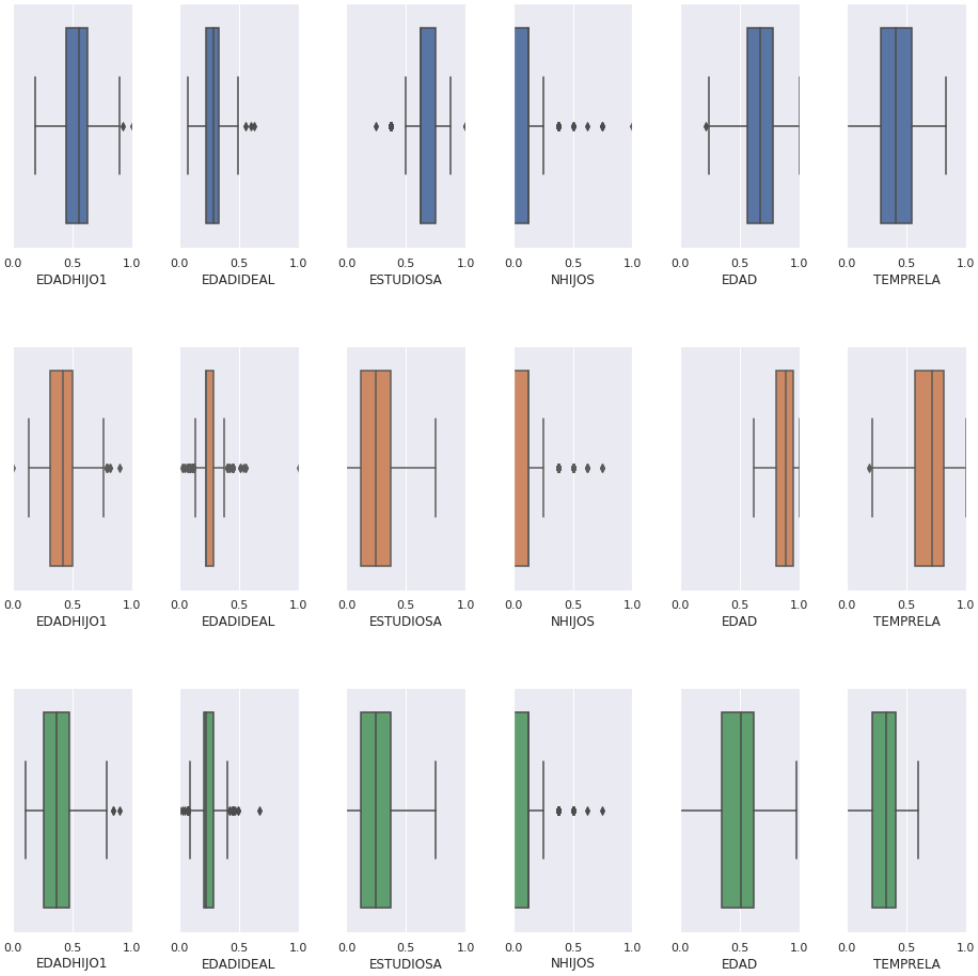

Figura 4: BoxPlot para K-Means con 3 clusters.

2.1.2. Agglomerative Clustering.

Ejecutaremos diferentes versiones de **Agglomerative Clustering** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5. El código utilizado ha sido el siguiente:

```
AgglomerativeClustering(n_clusters=X, linkage="ward")
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	878.579458	0.214783	0.187131
n_clusters	3	1010.128017	0.243308	0.200412
n_clusters	4	903.403235	0.200147	0.184900
n_clusters	5	855.145852	0.192042	0.192742

Tabla 2: Tabla modificaciones Agglomerative Clustering para el Caso 1.

Podemos observar que la versión que obtiene mejores resultados es la que obtenemos con 3 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   1193 (43.01%)
1:    911 (32.84%)
2:    670 (24.15%)
```

Aunque no es uno de los dos mejores algoritmos para este caso, se mostrarán 2 **Dendogramas** de diferentes tipos (uno simple y otro con Heatmap).

En la Figura 5, se representa en forma de Dendogramas, la forma en la que se agruparán todas las instancias hasta formar un único cluster teniendo en cuenta la distancia entre cada cluster.

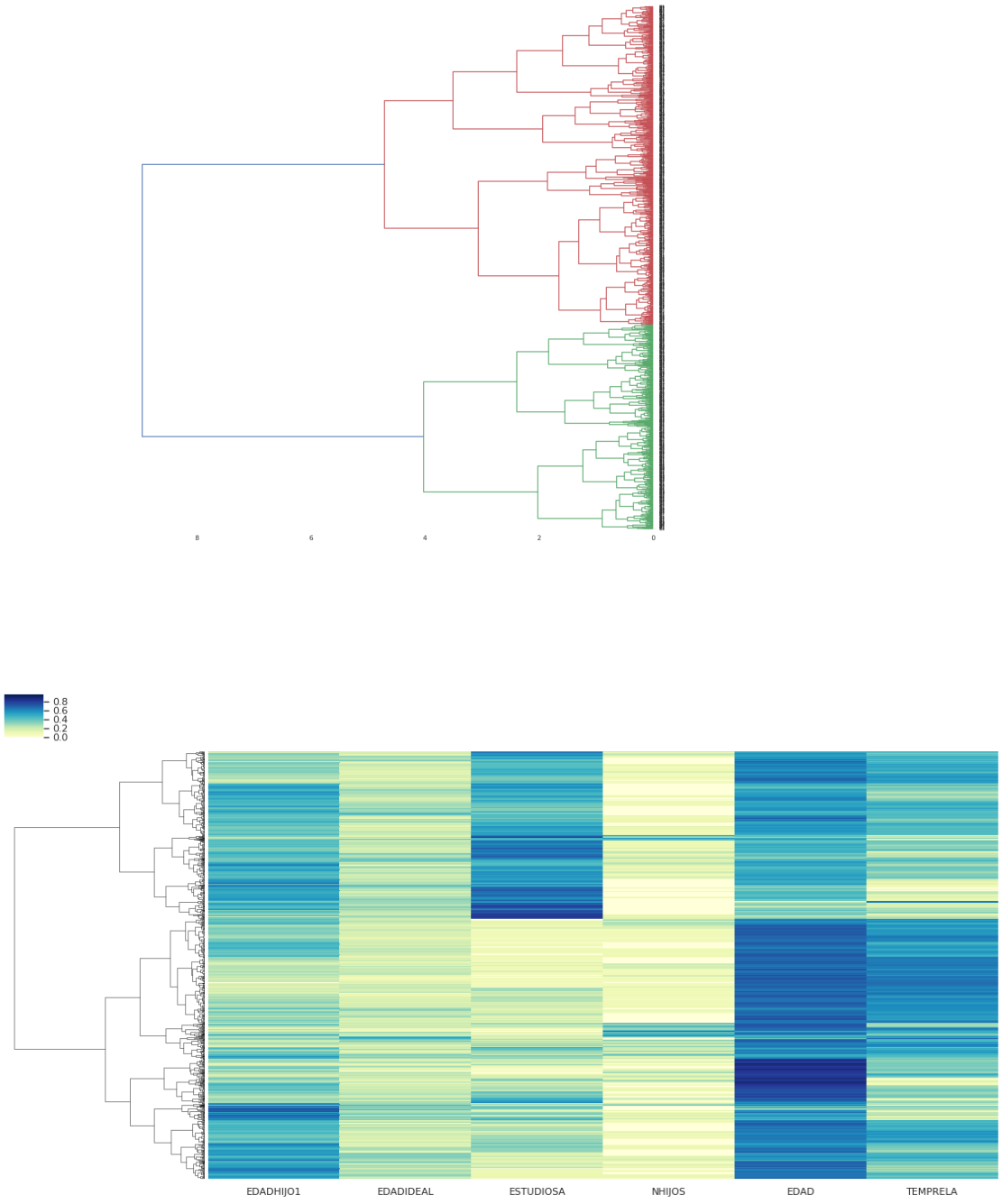

Figura 5: Dendogramas para Agglomerative Clustering con 3 clusters.

2.1.3. Birch.

Ejecutaremos diferentes versiones de **Birch** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5 y se ha fijado *threshold=0.25* y *branching_factor=25*. El código utilizado ha sido el siguiente:

```
Birch(branching_factor=25, n_clusters=X, threshold=0.25, compute_labels=
    True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	1026.215122	0.246898	0.065029
n_clusters	3	939.022449	0.223858	0.073361
n_clusters	4	636.709939	0.221875	0.070389
n_clusters	5	609.660934	0.164468	0.065896

Tabla 3: Tabla modificaciones Birch para el Caso 1.

Podemos observar que la versión que obtiene mejores resultados es la que obtenemos con 2 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
1:   1414 (50.97%)
0:   1360 (49.03%)
```

Para este caso, se mostrarán 4 gráficas y se hará un estudio sobre los resultados. Las gráficas obtenidas para este caso, serán **Scatter Matrix, Heatmap, KPlot y BoxPlot**.

Como podemos observar en las Figuras 6 y 7 obtenemos dos grupos diferenciados en lo relativo a las variables estudiadas (ESTUDIOSA, EDAD, TEMPRELA). Tenemos bastante buena proporción entre clusters, representando cada uno casi el 50 % de la muestra.

Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster.

Figura 6: Scatter Matrix para Birch con 2 clusters.

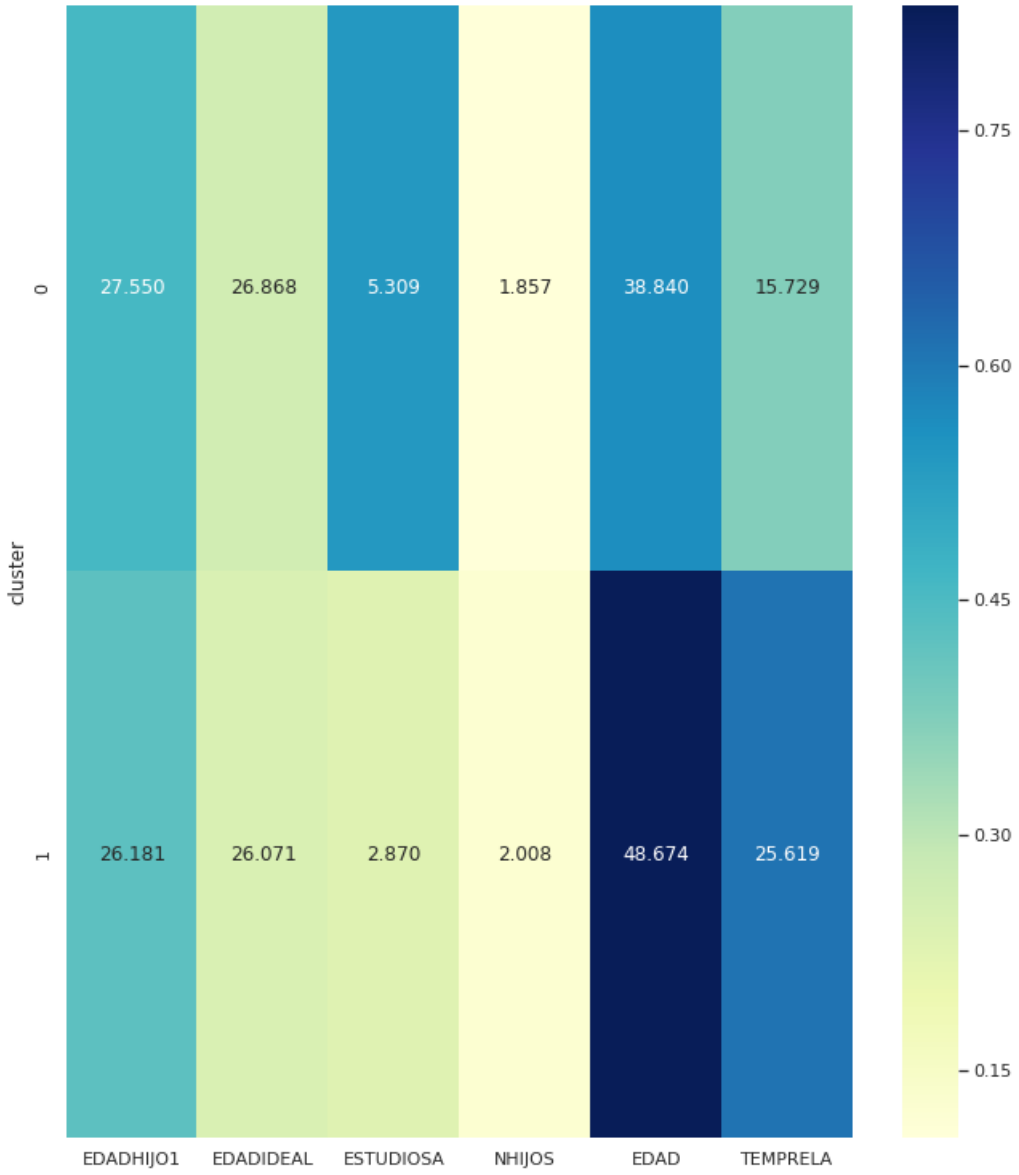

Figura 7: HeatMap para Birch con 2 clusters.

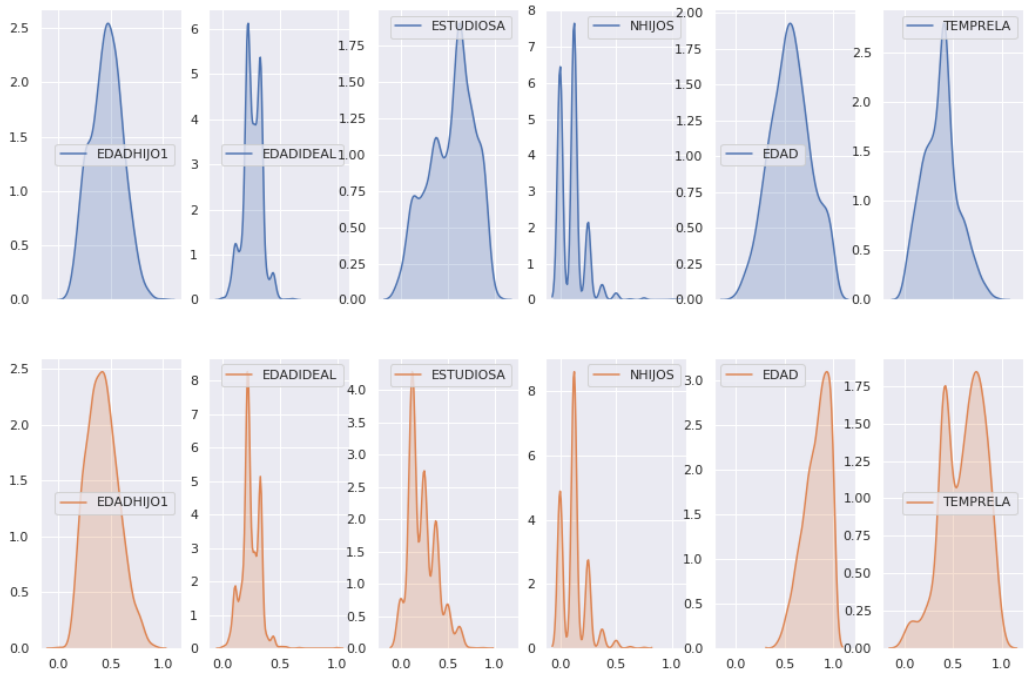

Figura 8: KPlot para Birch con 2 clusters.

Podemos observar que la distribución de NHIJOS no es muy representativa, ya que toma muchos valores. Aun así, las distribuciones de ambos clusters son muy similares. Diferenciaremos 2 grupos de mujeres en las que se caracterizan por las siguientes características:

- **Cluster 0**: Mujeres de menos de 40 años con alto nivel de estudios y que tuvo a su primer hijo relativamente hace poco tiempo, con una relación de pareja de 15 o más años.
- **Cluster 1**: Mujeres mayores de 40 años con un nivel medio-bajo de estudios y que tuvo su primer hijo a una edad bastante temprana (sobre los 20 años) y una relación de pareja muy larga.

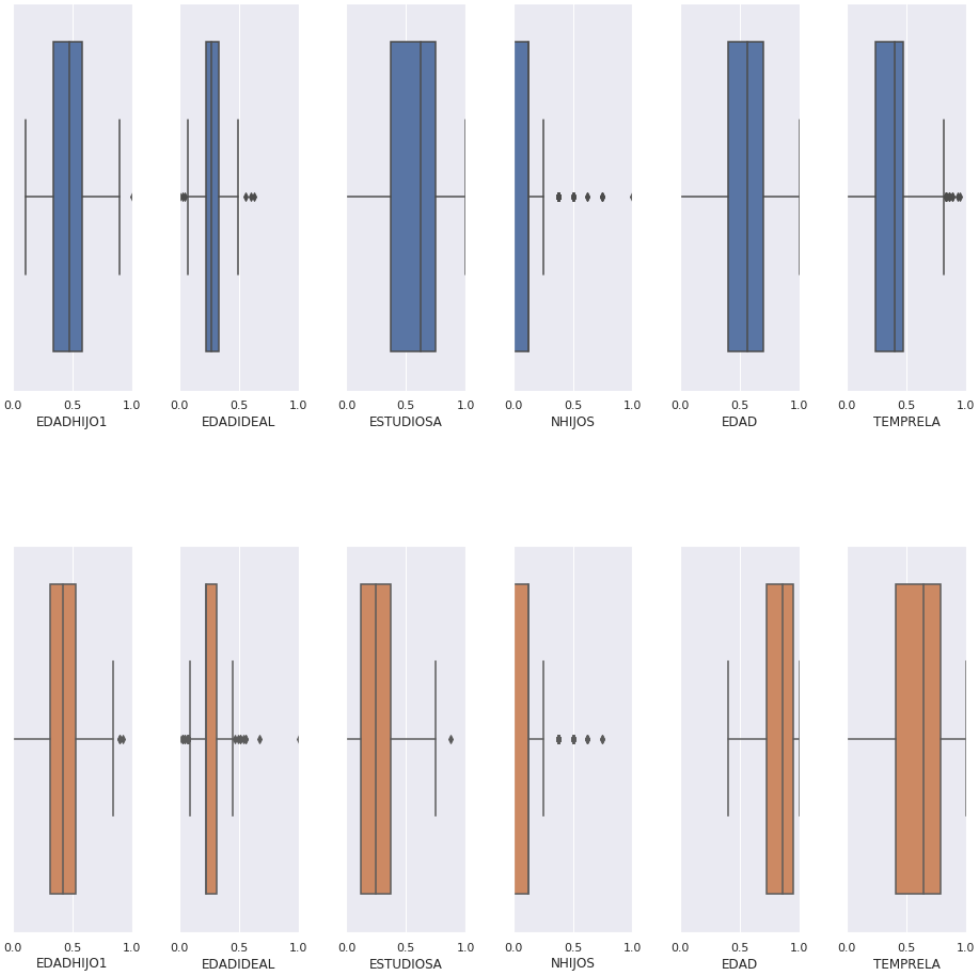

Figura 9: BoxPlot para Birch con 2 clusters.

2.1.4. MeanShift.

Ejecutaremos diferentes versiones de **MeanShift** modificando el parámetro *quantile* y fijando los parámetros *n_samples=400* y *bin_seeding=True*. El código utilizado ha sido el siguiente:

```
MeanShift(bandwidth=estimate_bandwidth(X_normal, quantile=X, n_samples
    =400), bin_seeding=True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
quantile=0.96	2	165.131049	0.176898	1.041183
qunatile=0.95	3	103.731611	0.146798	0.955801
quantile=0.90	5	97.728955	0.063901	1.870991

Tabla 4: Tabla modificaciones MeanShift para el Caso 1.

Podemos observar que la versión que obtiene mejores resultados es la que obtenemos con 2 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   2655 (95.71%)
1:    119 ( 4.29%)
```

2.1.5. DBSCAN.

Ejecutaremos diferentes versiones de **DBSCAN** modificando el parámetro *eps*. El código utilizado ha sido el siguiente:

```
DBSCAN(eps=X)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
eps=0.25	2	19.083031	0.302892	0.086445
eps=0.14	4	22.669268	-0.118313	0.048391
eps=0.15	5	19.414028	-0.121727	0.055155

Tabla 5: Tabla modificaciones DBSCAN para el Caso 1.

Podemos observar que la versión que obtiene mejores resultados es la que obtenemos con 4 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   2185 (78.77%)
-1:   577 (20.80%)
2:      7 ( 0.25%)
1:      5 ( 0.18%)
```

Mostraremos la tabla comparativa que contiene las mejores versiones de cada algoritmo ejecutado. Previamente y conociendo estos resultados, se ha hecho un estudio más profundo sobre aquellos dos algoritmos que han obtenido un mayor valor del coeficiente de Silhouette. En este caso, **K-Means** y **Birch**.

Name	**Nº Clusters**	**CH**	**SH**	**Time**	**Clusters**
K-Means	3	1246.246928	0.284122	0.029773	1: 1128 (40.66 %) 2: 829 (29.88 %) 0: 817 (29.45 %)
DBSCAN	4	22.669268	-0.118313	0.048391	0: 2185 (78.77 %) -1: 577 (20.80 %) 2: 7 (0.25 %) 1: 5 (0.18 %)
Birch	2	1026.215122	0.246898	0.065029	1: 1414 (50.97 %) 0: 1360 (49.03 %)
AggCluster	3	1010.128017	0.243308	0.200412	0: 1193 (43.01 %) 1: 911 (32.84 %) 2: 670 (24.15 %)
MeanShift	2	165.131049	0.176898	1.041183	0: 2655 (95.71 %) 1: 119 (4.29 %)

Tabla 6: Tabla comparativa general para el Caso 1.

2.1.6. Interpretación de la segmentación.

Tomando los valores para las métricas **CH** y **SH** el que ofrece una mejor agrupación de los individuos es el algoritmo **K-Means**.

El estudio realizado sobre este algoritmo, divide la población en mujeres en tres rangos de edad. Podemos ver como en la población sobre la que se realizó el estudio, las mujeres sin trabajo remunerado con al menos 1 hijo biológico, se distribuyen entre los 30 y los 50 años.

Las mujeres más mayores (50 o más) que representan el 29.45 % de la muestra estudiada, tienen un bajo nivel de estudios y unas relaciones de pareja muy largas. Además, no se arrepienten de haber tenido a su primer hijo a una edad tan temprana. Estas características son tradicionalmente mayoritarias entre las mujeres mayores de 50 años. Probablemente no tengan un trabajo remunerado porque se dediquen a las tareas del hogar.

Las mujeres que se encuentran en un rango medio de la edad (sobre los 40 años) que representan un 40.66 % de la muestra, tienen un alto nivel de estudios (formación profesional o universitario), tuvieron su hijo bastante tarde (sobre los 30 años) y se arrepienten de no haberlo tenido más jóvenes. Un nivel de estudios alto, la mayoría de veces, retrasa la idea de tener un hijo, hasta que se obtenga una siuación estable.

Finalmente, las mujeres más jóvenes en este rango (sobre los 35 años) que representan el 29.88 % de la muestra, se caracterizan por tener un nivel bajo de estudios (secundaria) y haber tenido sus hijos a una edad joven (sobre los 24 años). En este caso, el grupo se caracteriza por madres que tuvieron a su hijo a una edad temprana y por ello, no pudieron progresar en sus estudios.

2.2. Caso de estudio 2.

Para el segundo caso, estudiaremos al conjunto de mujeres menores de 30 años. Las variables que vamos a elegir se corresponden con:

- **EDAD**: Edad en años cumplidos.
- **EDADIDEAL**: Edad ideal para tenerel primer hijo.
- **TEMPRELA**: Número de años de la relación de pareja actual.
- **SATISRELAC**: Grado de satisfacción que le proporciona la pareja.
- **NHIJOSDESEO**: Número de hijos que le gustaría o le hubiera gustado tener.
- **NINTENHIJOS**: Número de hijos que tiene intención de tener en los próximos 3 años.

El objetivo de este caso de estudio consiste en distinguir grupos de mujeres jóvenes menores o iguales a 30 años y realizar un estudio sobre su relación de pareja y su intención de tener hijos en un futuro próximo. Después intentaremos sacar conclusiones acerca del perfil de mujer de cada grupo.

Para cada caso de estudio, he obtenido una muestra de la población total que rondará entre 400 y 8000 ejemplos. En este caso, estudiaremos un total de 3460. El conjunto seleccionado se ha obtenido a partir del siguiente código:

```
subset = datos.loc[datos['EDAD']<=30]
usadas = ['EDAD', 'EDADIDEAL', 'TEMPRELA', 'SATISRELAC', 'NHIJOSDESEO', '
    NINTENHIJOS']
X = subset[usadas]
```

Hemos seleccionado como ya hemos indicado antes, las mujeres de edad igual o inferior a 30 años. Además, las 6 variables sobre las que se ha realizado el estudio son todas numéricas.

A continuación, se mostrarán los resultados obtenidos por los algoritmos en este caso de estudio. Los algoritmos elegidos han sido los mismos que para el caso anterior: **K-Means, MeanShift, DBSCAN, Birch y Agglomerative Clustering**. Haremos un estudio sobre varias modificaciones para los algoritmos descritos desarrollando más a fondo los dos que mejores resultados obtengan.

2.2.1. K-Means.

Obtenemos diferentes resultados de las ejecuciones para las distintas versiones de **K-Means** donde modificamos el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5, fijaremos como *random_state* una semilla con valor de mi DNI y *n_init=5*. El código utilizado ha sido el siguiente:

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

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	**Nº Clusters**	**CH**	**SH**	**Time**
n_clusters	2	3550.829642	0.475700	0.015393
n_clusters	3	3675.741404	0.458313	0.023925
n_clusters	4	4130.191401	0.447899	0.025948
n_clusters	5	3758.492633	0.416641	0.043916

Tabla 7: Tabla modificaciones K-Means para el Caso 2.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 4 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
1:  1145 (33.09%)
0:   938 (27.11%)
2:   703 (20.32%)
3:   674 (19.48%)
```

Para este caso, se mostrarán 4 gráficas realizadas y se hará un estudio sobre los resultados obtenidos. Las gráficas obtenidas para este caso, serán **Scatter Matrix, Heatmap, KPlot y BoxPlot**.

Como podemos observar en las Figuras 10 y 11 obtenemos cuatro grupos diferenciados en lo relativo a las variables estudiadas (EDAD, TEMPRELA, SATISRELAC). Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster.

Figura 10: Scatter Matrix para K-Means con 4 clusters.

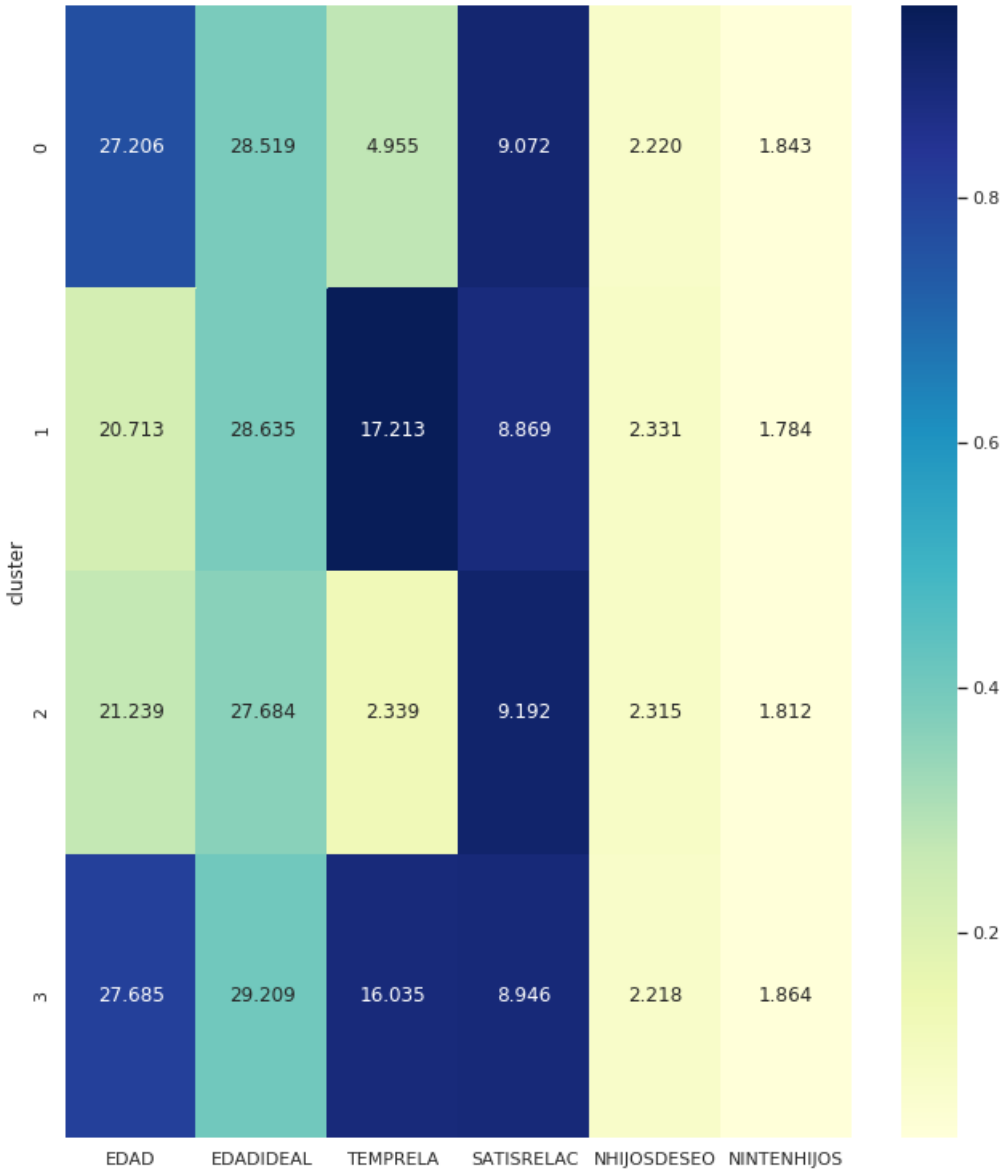

Figura 11: HeatMap para K-Means con 4 clusters.

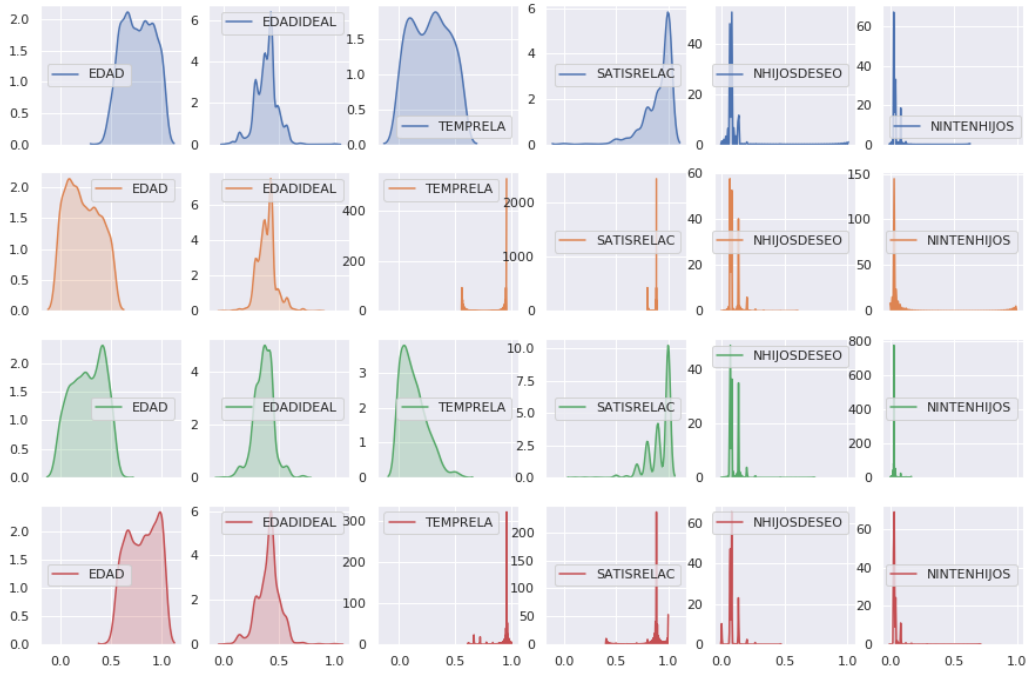

Figura 12: KPlot para K-Means con 4 clusters.

Diferenciaremos 4 grupos de mujeres en las que se caracterizan por las siguientes características:

- **Cluster 0**: Mujeres de 27 años de media con intención de tener 1 o 2 hijos en los próximos 3 años. Con una relación de pareja de unos 5 años.
- **Cluster 1**: Mujeres de 20 años de media con intención de tener hijos pero a los 28. (En la variable TEMPRELA podemos ver en la Figura 12, que existen outliers que aumentan la media y la hacen menos relevante).
- **Cluster 2**: Mujeres de 21 años muy satisfechas con su relación en pareja (corta duración) y que piensan tener 1 o 2 hijos en un periodo de tiempo de entre 3 y 6 años.
- **Cluster 3**: Mujeres rozando los 30 años con una relación de pareja larga que pretenden tener 1 o 2 hijos en un periodo de 2 o 3 años.

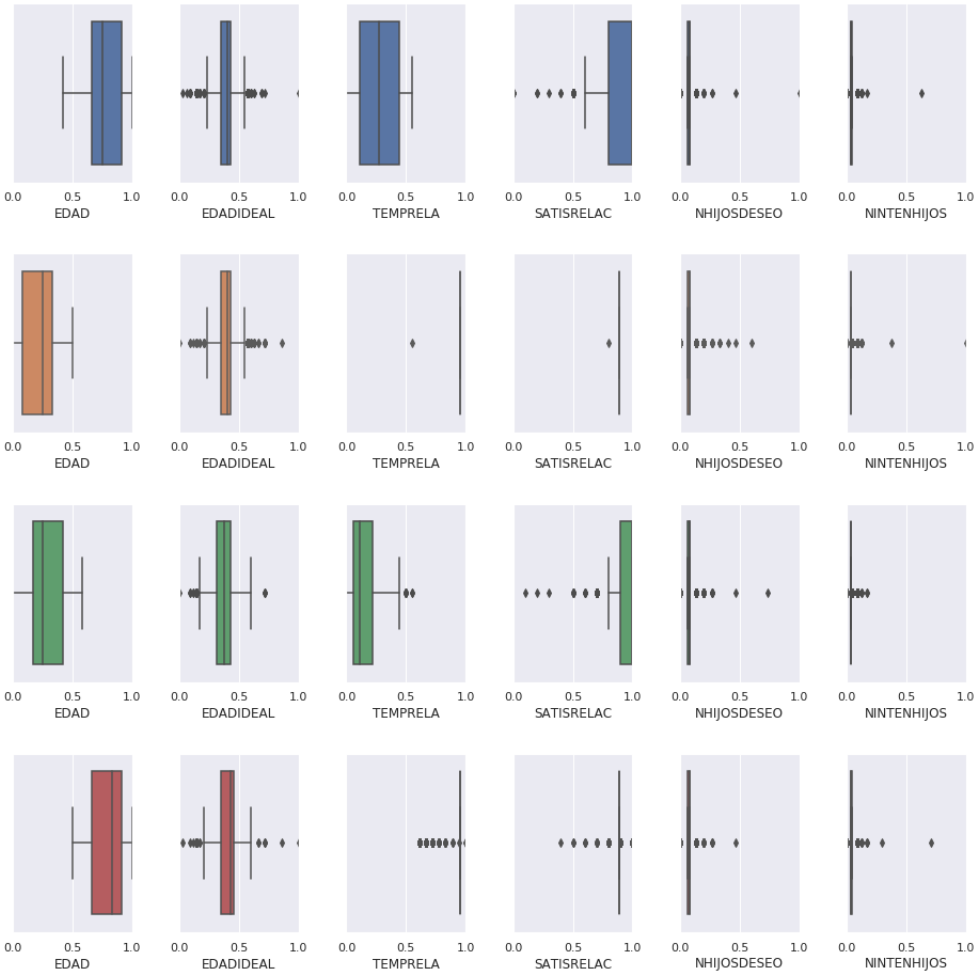

Figura 13: BoxPlot para K-Means con 4 clusters.

2.2.2. Agglomerative Clustering.

Ejecutaremos diferentes versiones de **Agglomerative Clustering** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5. El código utilizado ha sido el siguiente:

```
AgglomerativeClustering(n_clusters=X, linkage="ward")
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	3492.878158	0.473488	0.321365
n_clusters	3	3118.002135	0.430591	0.332533
n_clusters	4	3483.076157	0.390232	0.321060
n_clusters	5	3237.563071	0.372632	0.329160

Tabla 8: Tabla modificaciones Agglomerative Clustering para el Caso 2.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 2 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   1798 (51.97%)
1:   1662 (48.03%)
```

Para este caso, se mostrarán 6 gráficas realizadas. Las gráficas obtenidas para este caso, serán dos **Dendogramas** de diferentes tipos (uno simple y otro con Heatmap) y las gráficas **HeatMap**, **Scatter Matrix, KPlot y BoxPlot**.

Como podemos observar en las Figuras 14 y 15, obtenemos dos grupos diferenciados en lo relativo a las variables estudiadas (EDAD, TEMPRELA, SATISRELAC). Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster.

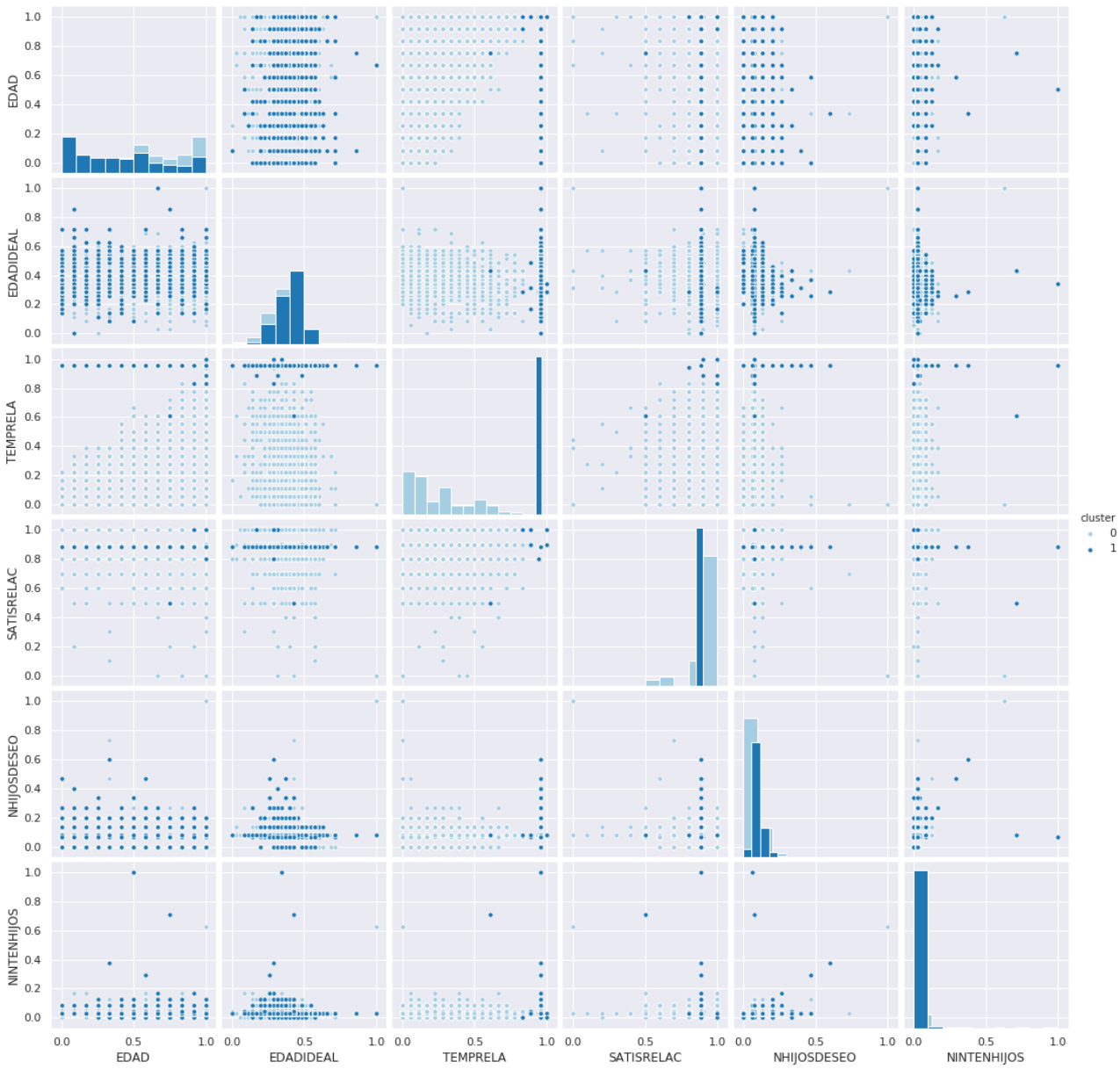

Figura 14: Scatter Matrix para Agglomerative Clustering con 2 clusters.

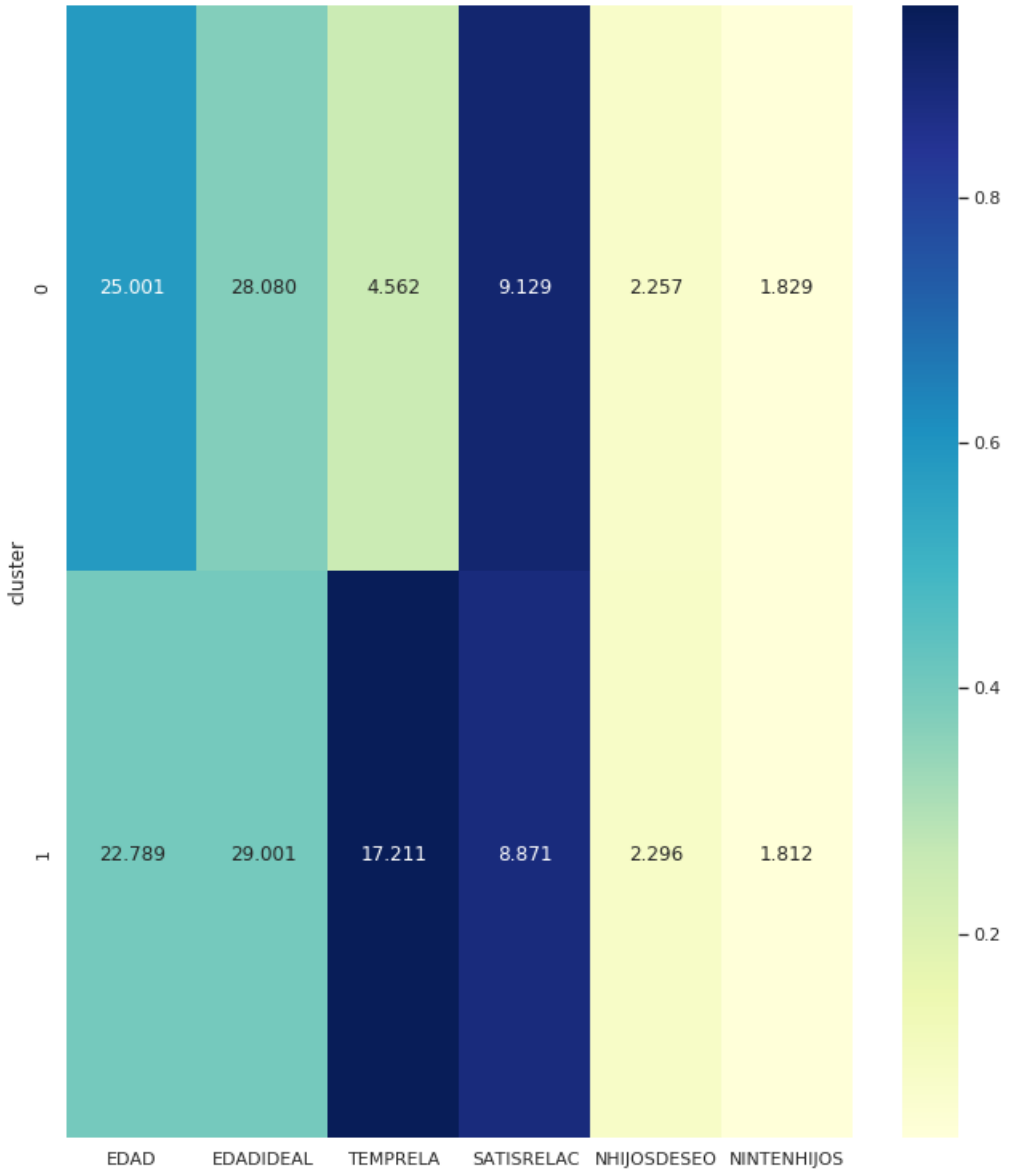

Figura 15: HeatMap para Agglomerative Clustering con 2 clusters.

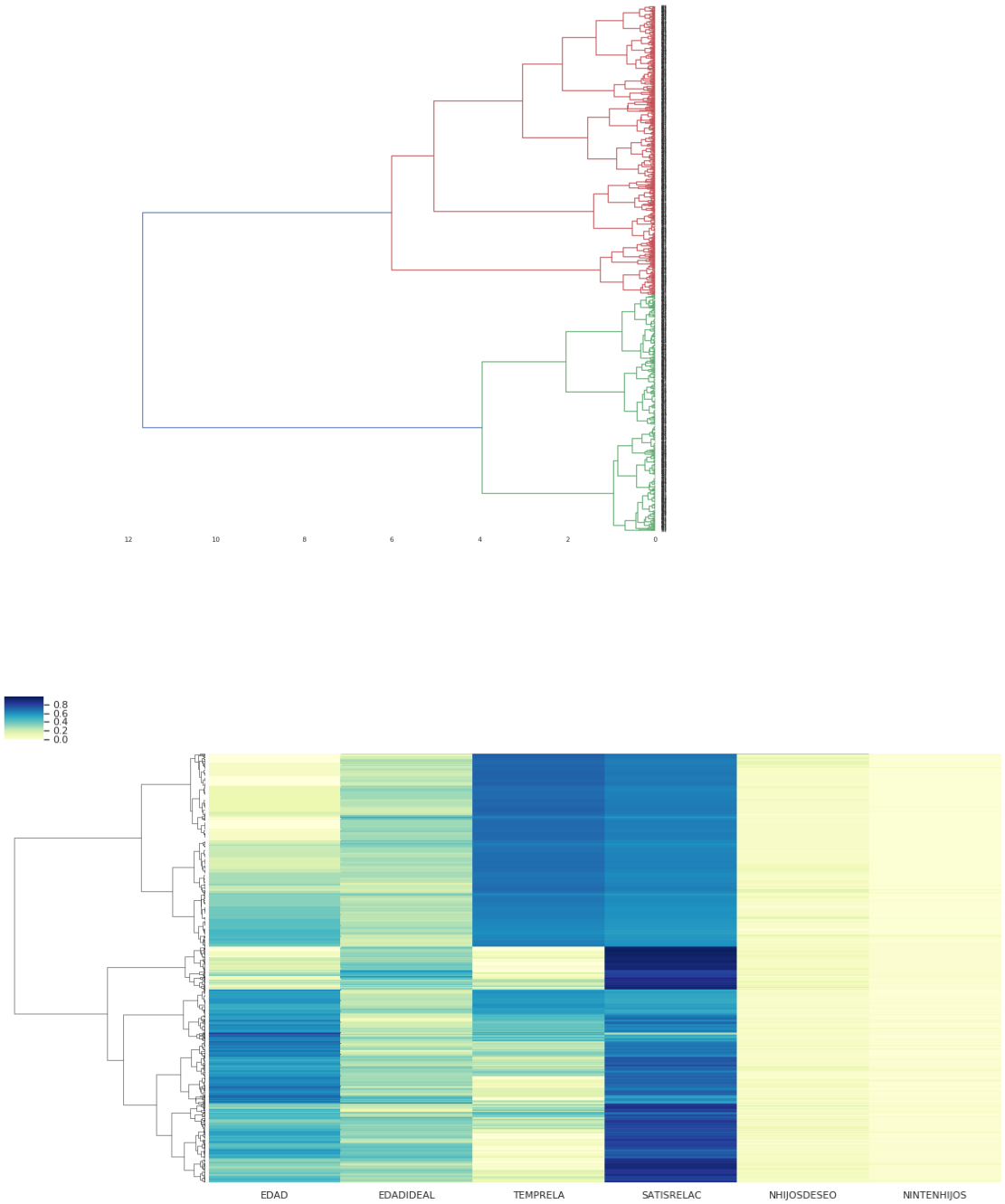

Figura 16: Dendogramas para Agglomerative Clustering con 2 clusters.

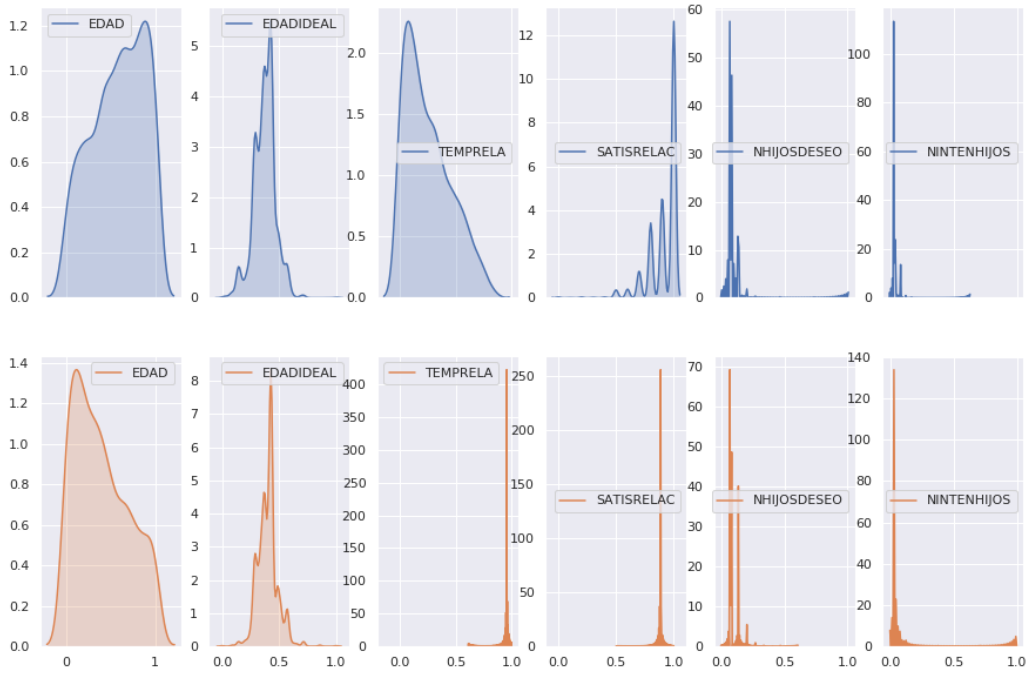

Figura 17: KPlot para Agglomerative Clustering con 2 clusters.

Diferenciaremos 2 grupos de mujeres en las que se caracterizan por las siguientes características:

- **Cluster 0**: Mujeres de 25 años de media con intención de tener 1 o 2 hijos en los próximos 3 años. Con una relación de pareja corta de unos 5 años.
- **Cluster 1**: Mujeres de 22 años de media con intención de tener hijos pero a los 29 y con relaciones muy largas de media (observando la distribución de la Figura 17).

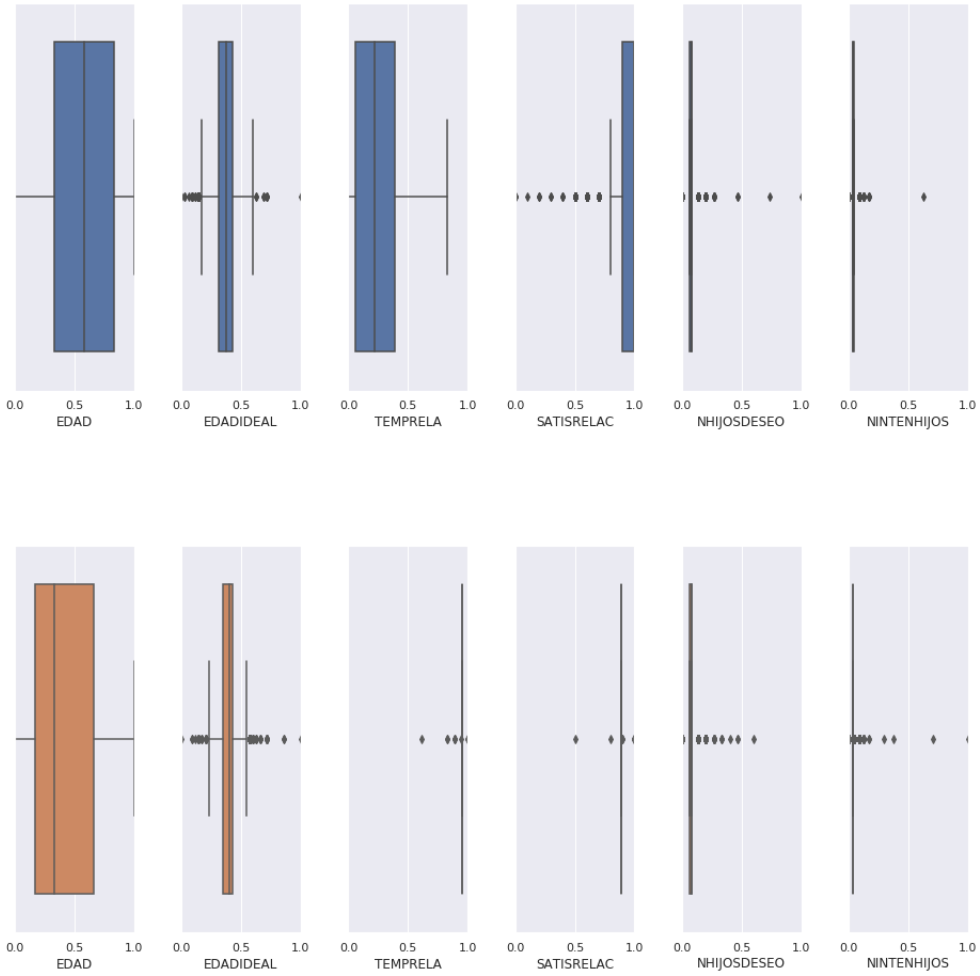

Figura 18: BoxPlot para Agglomerative Clustering con 2 clusters.

2.2.3. Birch.

Ejecutaremos diferentes versiones de **Birch** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5 y se ha fijado *threshold=0.25* y *branching_factor=25*. El código utilizado ha sido el siguiente:

```
Birch(branching_factor=25, n_clusters=X, threshold=0.25, compute_labels=
    True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	11.363978	0.628718	0.052299
n_clusters	3	1364.151719	0.417737	0.053521
n_clusters	4	966.117237	0.360215	0.052725
n_clusters	5	1914.520323	0.427266	0.052128

Tabla 9: Tabla modificaciones Birch para el Caso 2.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 5 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
1:  1259 (36.39%)
4:  1234 (35.66%)
2:   919 (26.56%)
0:    47 ( 1.36%)
3:     1 ( 0.03%)
```

2.2.4. MeanShift.

Ejecutaremos diferentes versiones de **MeanShift** modificando el parámetro *quantile* y fijamos los parámetros *n_samples=400* y *bin_seeding=True*. El código utilizado ha sido el siguiente:

```
MeanShift(bandwidth=estimate_bandwidth(X_normal, quantile=X, n_samples
    =400), bin_seeding=True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
quantile=0.90	5	897.129058	0.462479	0.383211

Tabla 10: Tabla modificaciones MeanShift para el Caso 2.

Solo hemos podido conseguir una versión con 5 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
1:  1795 (51.88%)
0:  1658 (47.92%)
3:     5 ( 0.14%)
4:     1 ( 0.03%)
2:     1 ( 0.03%)
```

Esta versión de **MeanShift** tiene un valor alto del coeficiente Silhouette (**SH**), mirando la tabla comparativa del caso 2, es el segundo con mejor valor, lo que nos podría indicar que sería un buen algoritmo para estudiar a fondo. Sin embargo, si observamos su **Scatter Matrix** en la Figura 19, nos daremos cuenta de que este valor alto de **SH** se debe a que varios clusters están formados por 1 o un conjunto pequeño de individuos (outliers), lo que hace que el coeficiente Silhouette aumente.

Por tanto, lo descartaremos del estudio en profundidad y elegiremos el siguiente con mejor valor de **SH**.

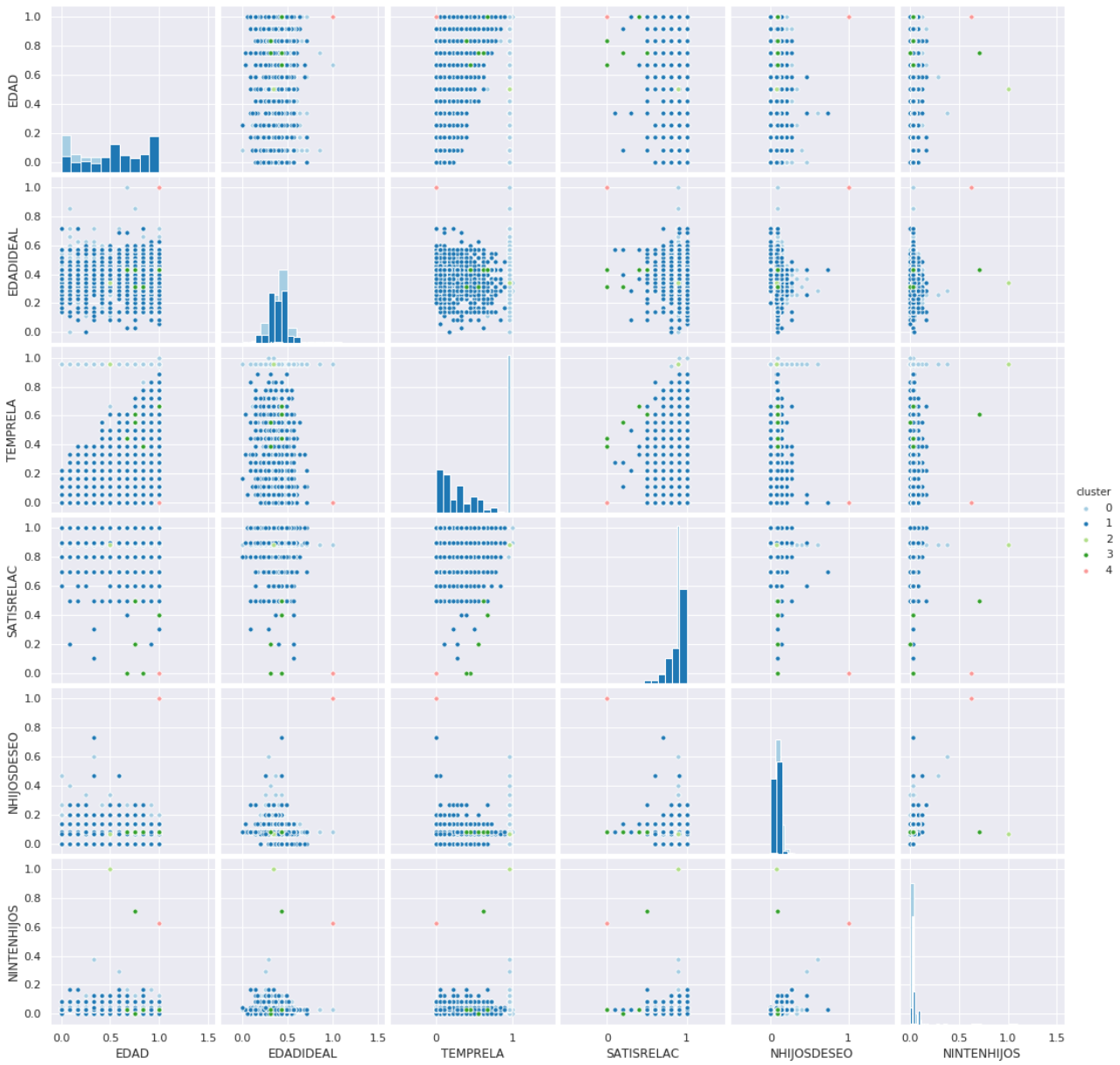

Figura 19: Scatter Matrix para MeanShift con 5 clusters.

2.2.5. DBSCAN.

Ejecutaremos diferentes versiones de **DBSCAN** modificando el parámetro *eps*. El código utilizado ha sido el siguiente:

```
DBSCAN(eps=X)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
eps=0.2	2	21.652302	0.302082	0.093933
eps=0.12	3	1768.746342	0.393759	0.068426
eps=0.11	4	1177.754069	0.122921	0.098747

Tabla 11: Tabla modificaciones DBSCAN para el Caso 2.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 3 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
1:   1675 (48.41%)
0:   1642 (47.46%)
-1:   143 ( 4.13%)
```

Mostraremos la tabla comparativa que contiene las mejores versiones de cada algoritmo ejecutado. Previamente y conociendo estos resultados, se ha hecho un estudio más profundo sobre aquellos dos algoritmos que han obtenido un mayor equilibrio en ambas métricas destacando sobre el resto. En este caso, **K-Means** y **Agglomerative Clustering**.

Name	**Nº Clusters**	**CH**	**SH**	**Time**	**Clusters**
K-Means	4	4130.191401	0.447899	0.025948	1: 1145 (33.09 %) 0: 938 (27.11 %) 2: 703 (20.32 %) 3: 674 (19.48 %)
DBSCAN	3	1768.746342	0.393759	0.078585	1: 1675 (48.41 %) 0: 1642 (47.46 %) -1: 143 (4.13 %)
Birch	5	1914.520323	0.427266	0.052132	1: 1259 (36.39 %) 4: 1234 (35.66 %) 2: 919 (26.56 %) 0: 47 (1.36 %) 3: 1 (0.03 %)
AggCluster	2	3492.878158	0.473488	0.321365	0: 1798 (51.97 %) 1: 1662 (48.03 %)
MeanShift	5	897.129058	0.462479	0.380385	1: 1795 (51.88 %) 0: 1658 (47.92 %) 3: 5 (0.14 %) 4: 1 (0.03 %) 2: 1 (0.03 %)

Tabla 12: Tabla comparativa general para el Caso 2.

2.2.6. Interpretación de la segmentación.

Tomando los valores para las métricas **CH** y **SH** el que ofrece una mejor agrupación de los individuos es el algoritmo **Agglomerative Clustering**.

El estudio realizado sobre este algoritmo, divide la población en mujeres jóvenes de dos tipos:

Las mujeres del primer tipo que representan un 51.97 % de la muestra estudiada, se caracterizan por tener una edad más cercana a los 30 años y una relación de pareja de 5 años de media y pretende tener uno o dos hijos dentro de 3 años. Las mujeres del segundo tipo, son más jóvenes (sobre los 23 años) y se caracterizan por tener unas

relaciones más largas y preferir ser madres a una edad más tardía.

Los resultados obtenidos son bastante lógicos, las mujeres más cercanas a los 30 años y con relaciones de pareja de menos tiempo (conocidas a los 24 o 25) tienen intención de tener en poco tiempo un total de 1 o 2 hijos.

Por el contrario, el grupo que representa al 48.03 % de la muestra, está formado por las mujeres más jóvenes que llevan más tiempo con su pareja (probablemente conocida en la adolescencia) son más estrictas a la hora de puntuar su relación de pareja y tienden a alargar la edad a la que quieren ser madres, ya que son bastante jóvenes.

En cuanto al número de hijos deseados, podemos concluir que las jóvenes actualmente tienen la intención de tener una media de dos hijos, sea cual sea su rango de edad, por lo que es una característica común en esta muestra.

2.3. Caso de estudio 3.

Para el tercer caso, estudiaremos al conjunto de mujeres casadas menores de 45 años. Las variables que vamos a elegir se corresponden con:

- **EDAD**: Edad en años cumplidos.
- **NHIJOS**: Número de hijos suyos o de su pareja.
- **INGREHOG_INTER**: Ingresos mensuales netos del hogar en intervalos. (**1**-menos de 500, **2**-500 a 1000, **3**-1000 a 1500, **4**-1500 a 2000, **5**-2000 a 2500, **6**-2500 a 3000, **7**-3000 a 5000).
- **ABUELOS**: Número de días a la semana de uso de abuelos.
- **TEMPRELA**: Número de años de la relación de pareja actual.
- **PCUIDADOHIJOS**: Gasto mensual en estos cuidados para sus hijos.

El objetivo de este caso de estudio consiste en distinguir grupos de mujeres casadas de edad menor o igual a 45 años y realizar un estudio sobre el número de hijos, los ingresos mensuales netos y el dinero que invierte al cuidado de los niños. Después intentaremos sacar conclusiones acerca del perfil de mujer de cada grupo.

Para cada caso de estudio, he obtenido una muestra de la población total que rondará entre 400 y 8000 ejemplos. En este caso, estudiaremos un total de 3958. El conjunto seleccionado se ha obtenido a partir del siguiente código:

```
subset = datos.loc[(datos['EC']==2) & (datos['EDAD']<=45)]
usadas = ['EDAD','NHIJOS','INGREHOG_INTER','ABUELOS','TEMPRELA','
    PCUIDADOHIJOS']
X = subset[usadas]
```

Hemos seleccionado como ya hemos indicado antes, las mujeres casadas de edad igual o inferior a 45 años. Además, las 6 variables sobre las que se ha realizado el estudio son todas numéricas.

A continuación, se mostrarán los resultados obtenidos por los algoritmos en este caso de estudio. Los algoritmos elegidos han sido los mismos que para el caso anterior: **K-Means, MeanShift, DBSCAN, Birch y Agglomerative Clustering**. Realizaremos de nuevo un estudio sobre diferentes modificaciones y estudiaremos a fondo los dos algoritmos que mejores resultados obtengan.

2.3.1. K-Means.

Ejecutaremos diferentes versiones de **K-Means** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5, fijaremos como *random_state* una semilla con valor de mi DNI y *n_init=5*. El código utilizado ha sido el siguiente:

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

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	1204.516264	0.226095	0.020867
n_clusters	3	1451.077451	0.267710	0.030554
n_clusters	4	1439.619508	0.258578	0.043986
n_clusters	5	1340.513019	0.237735	0.065477

Tabla 13: Tabla modificaciones K-Means para el Caso 3.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 3 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   1643 (41.51%)
1:   1623 (41.01%)
2:    692 (17.48%)
```

Para este caso, se mostrarán 4 gráficas realizadas y se hará un estudio sobre los resultados obtenidos. Las gráficas obtenidas para este caso, serán **Scatter Matrix, Heatmap, KPlot y BoxPlot**.

Como podemos observar en las Figuras 20 y 21 obtenemos tres grupos diferenciados en lo relativo a las variables estudiadas (INGREHOG_INTER, ABUELOS, PCUIDADOHIJOS). Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster.

Figura 20: Scatter Matrix para K-Means con 3 clusters.

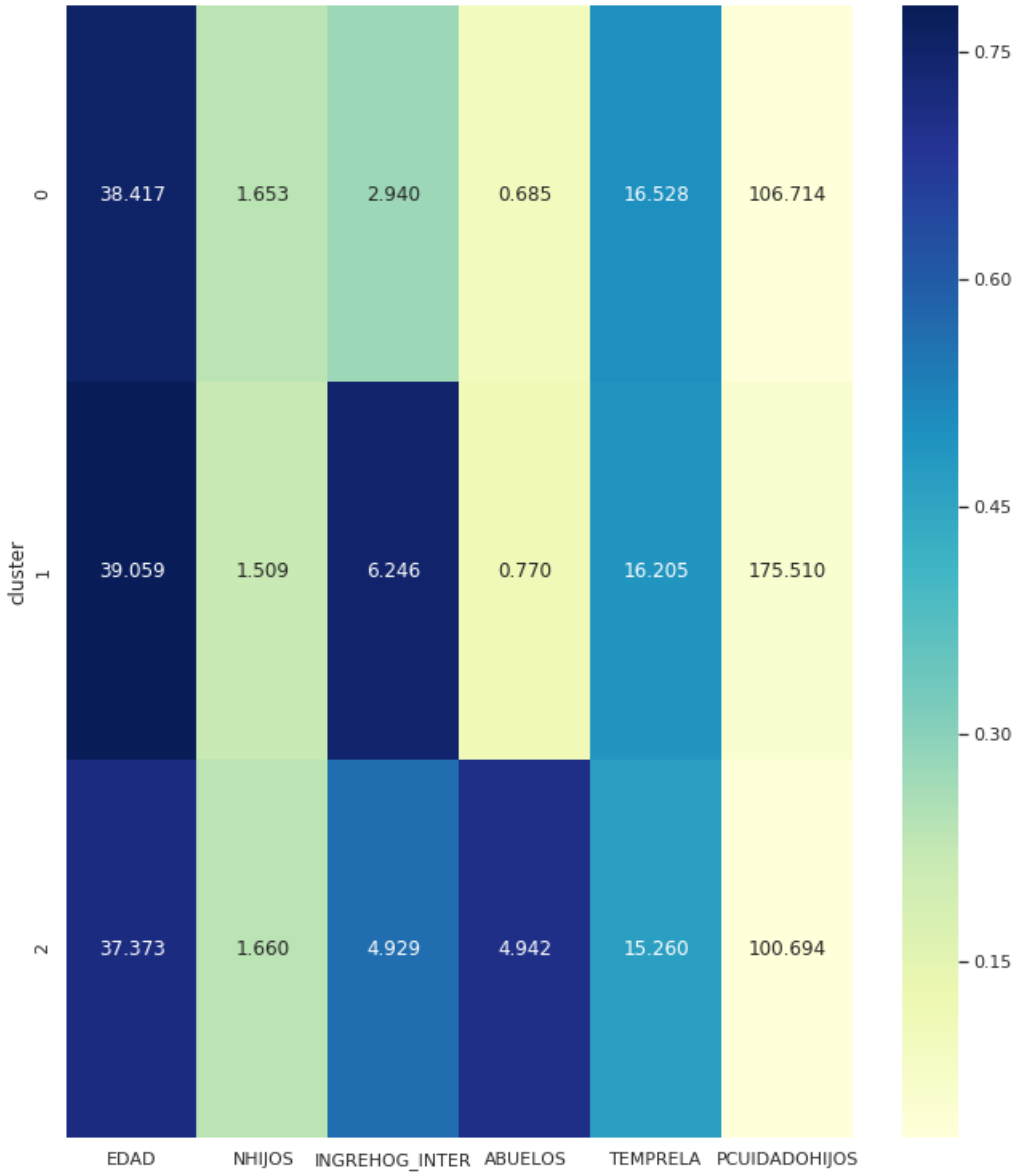

Figura 21: HeatMap para K-Means con 3 clusters.

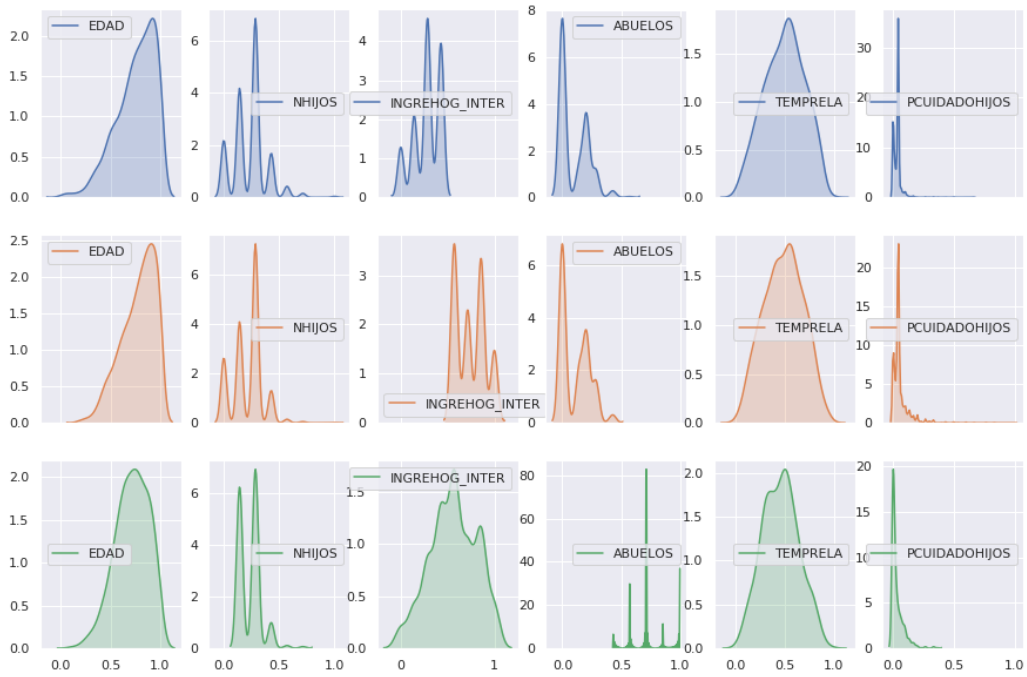

Figura 22: KPlot para K-Means con 3 clusters.

Diferenciaremos 3 grupos de mujeres en las que se caracterizan por las siguientes características:

- **Cluster 0**: Mujeres de 38 años de media con unos ingresos mensuales de entre 1000 a 1500 euros de media con una media de 2 hijos y que invierten una media de 106 euros mensuales en ellos.
- **Cluster 1**: Mujeres de 39 años de media con unos ingresos mensuales de entre 2500 a 3000 euros de media con una media de 2 hijos y que invierten una media de 175 euros mensuales en ellos.
- **Cluster 2**: Mujeres de 37 años de media con unos ingresos mensuales de entre 2000 a 2500 euros de media con una media de 2 hijos y que invierten una media de 100 euros mensuales en ellos y utilizan a los abuelos 5 días de media a la semana para el cuidado de sus hijos.

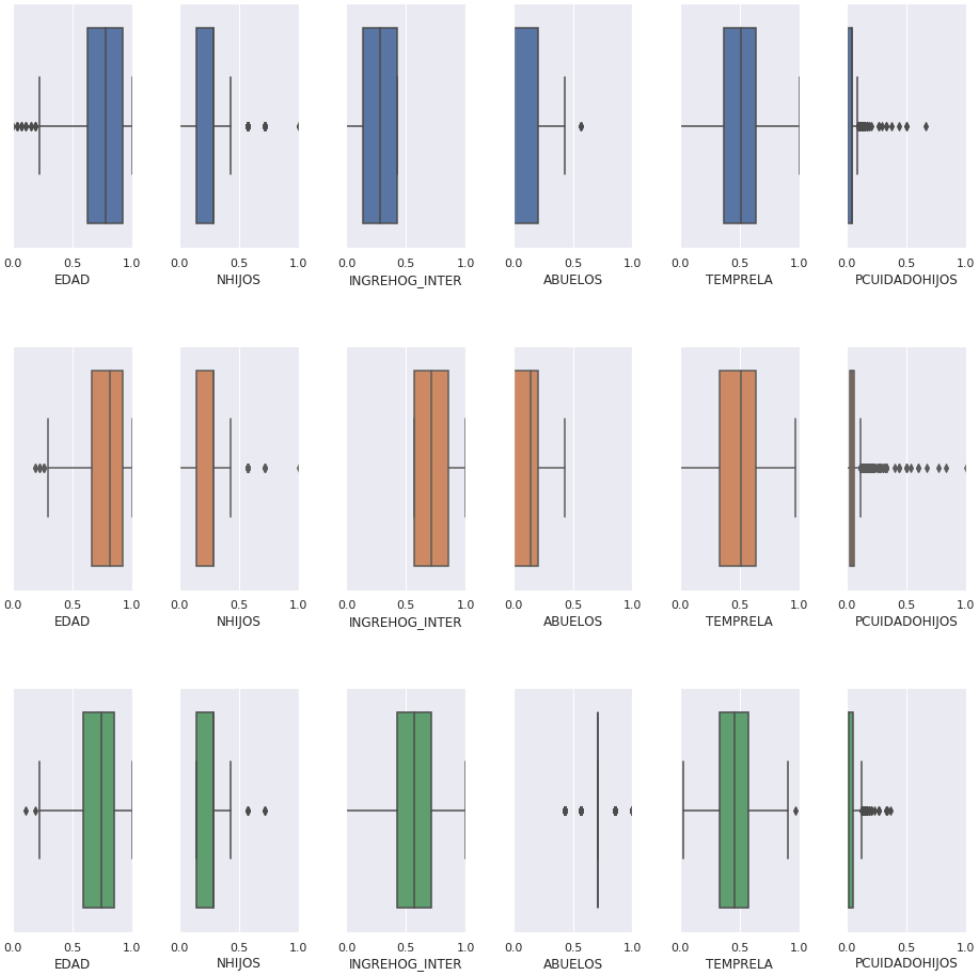

Figura 23: BoxPlot para K-Means con 3 clusters.

2.3.2. Agglomerative Clustering.

Ejecutaremos diferentes versiones de **Agglomerative Clustering** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5. El código utilizado ha sido el siguiente:

```
AgglomerativeClustering(n_clusters=X, linkage="ward")
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	1110.434676	0.310333	0.424029
n_clusters	3	1292.676163	0.238999	0.416693
n_clusters	4	1257.553275	0.216241	0.422984
n_clusters	5	1116.646083	0.185537	0.427551

Tabla 14: Tabla modificaciones Agglomerative Clustering para el Caso 3.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 3 clusters. Los clusters se han dividido de la siguiente manera:

```
0:   1965 (49.65%)
2:   1359 (34.34%)
1:    634 (16.02%)
```

Para este caso, se mostrarán 6 gráficas realizadas. Las gráficas obtenidas para este caso, serán dos **Dendogramas** de diferentes tipos (uno simple y otro con Heatmap) y las gráficas **HeatMap**, **Scatter Matrix, KPlot y BoxPlot**.

Como podemos observar en las Figuras 24 y obtenemos tres grupos diferenciados en lo relativo a las variables estudiadas (INGREHOG_INTER, ABUELOS, PCUIDADOHIJOS). Definiremos en rasgos generales y basándonos en los resultados de **HeatMap** y **KPlot** las características más relevantes de cada cluster.

Figura 24: Scatter Matrix para Agglomerative Clustering con 3 clusters.

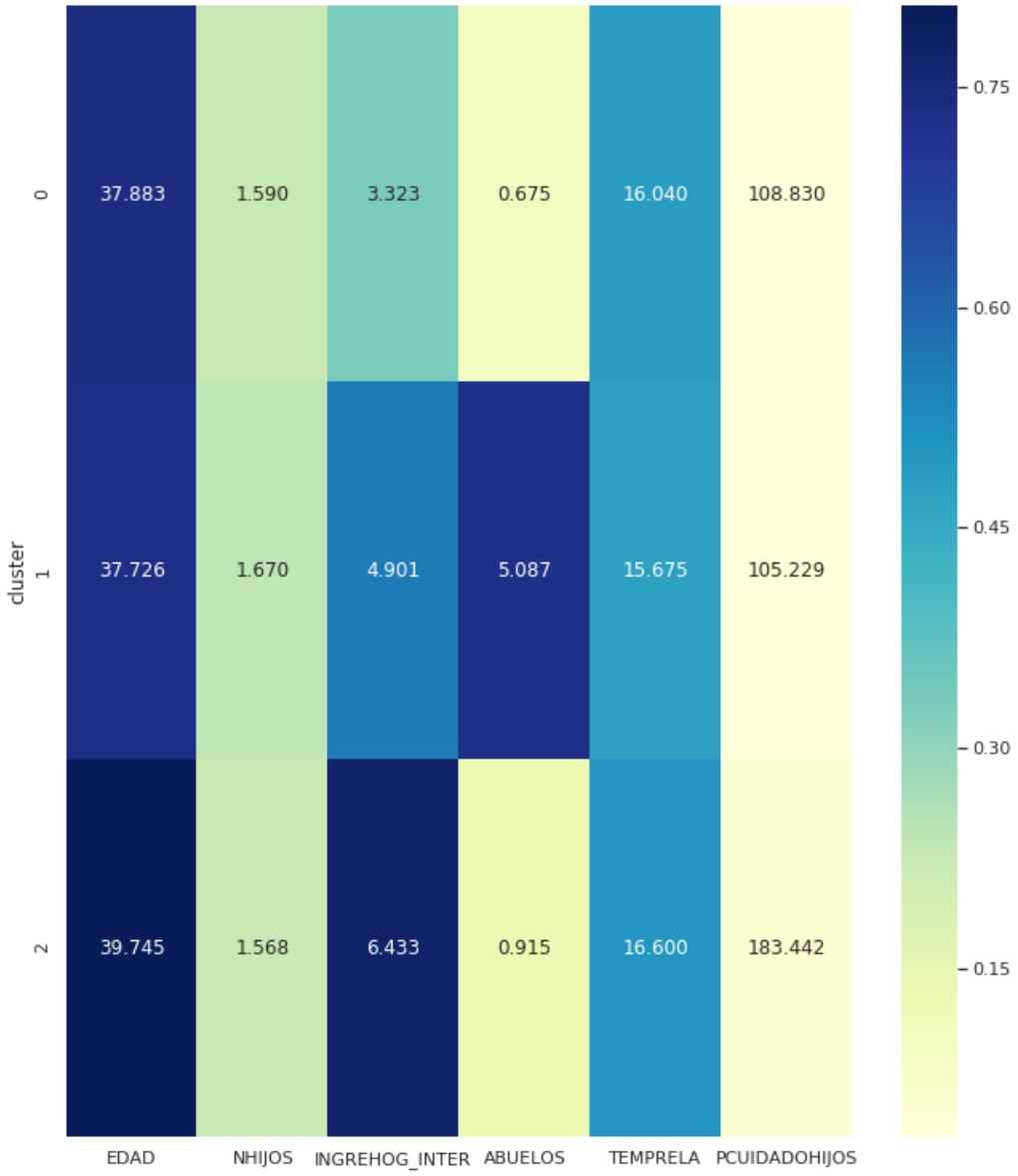

Figura 25: HeatMap para Agglomerative Clustering con 3 clusters.

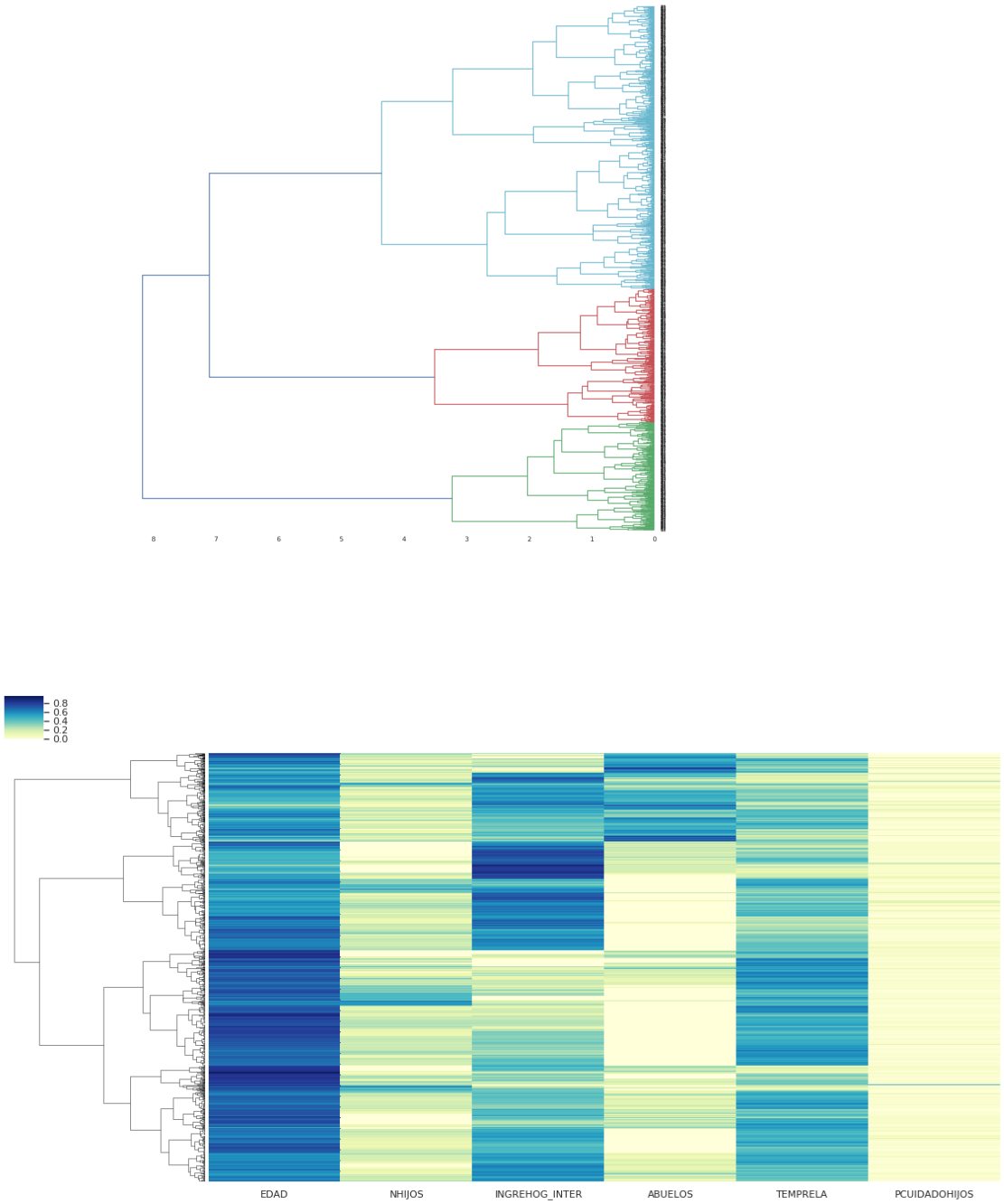

Figura 26: Dendogramas para Agglomerative Clustering con 3 clusters.

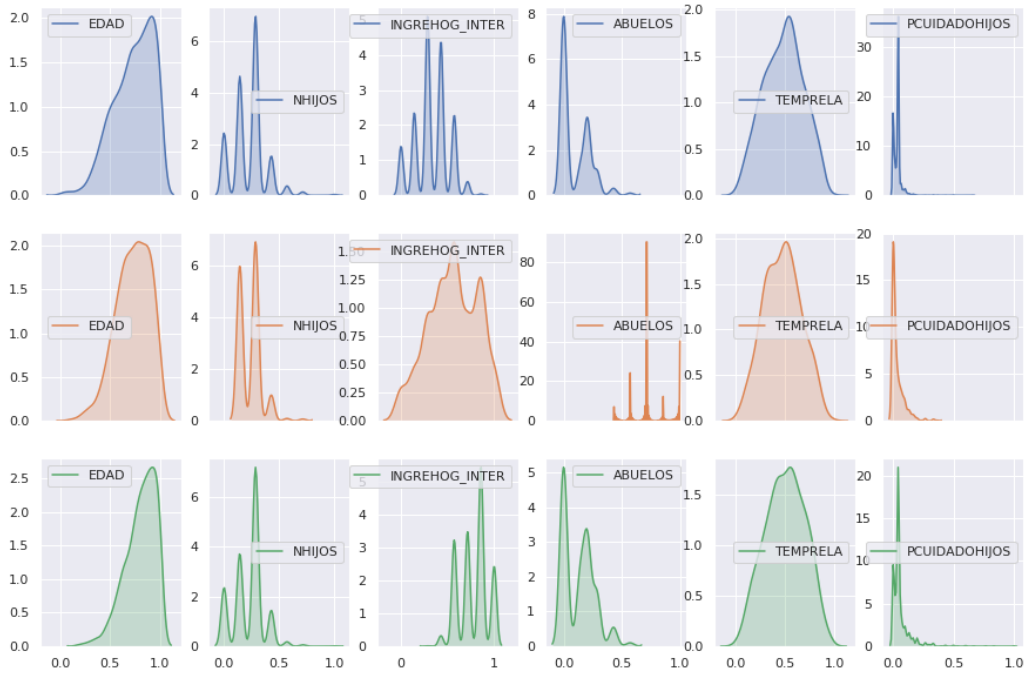

Figura 27: KPlot para Agglomerative Clustering con 3 clusters.

Diferenciaremos 3 grupos de mujeres en la que los grupos obtenidos cumplen unas características muy similares a las ya explicadas en el apartado del algoritmo **K-Means** salvo pequeñas variaciones y una alteración en el enumerado de los clusters.

- **Cluster 0**: Mujeres de 38 años de media con unos ingresos mensuales de entre 1000 a 1500 euros de media con una media de 2 hijos y que invierten una media de 108 euros mensuales en ellos.
- **Cluster 1**: Mujeres de 38 años de media con unos ingresos mensuales de entre 2000 a 2500 euros de media con una media de 2 hijos y que invierten una media de 105 euros mensuales en ellos y utilizan a los abuelos 5 días de media a la semana para el cuidado de sus hijos.
- **Cluster 2**: Mujeres de 40 años de media con unos ingresos mensuales de entre 2500 a 3000 euros de media con una media de 2 hijos y que invierten una media de 184 euros mensuales en ellos.

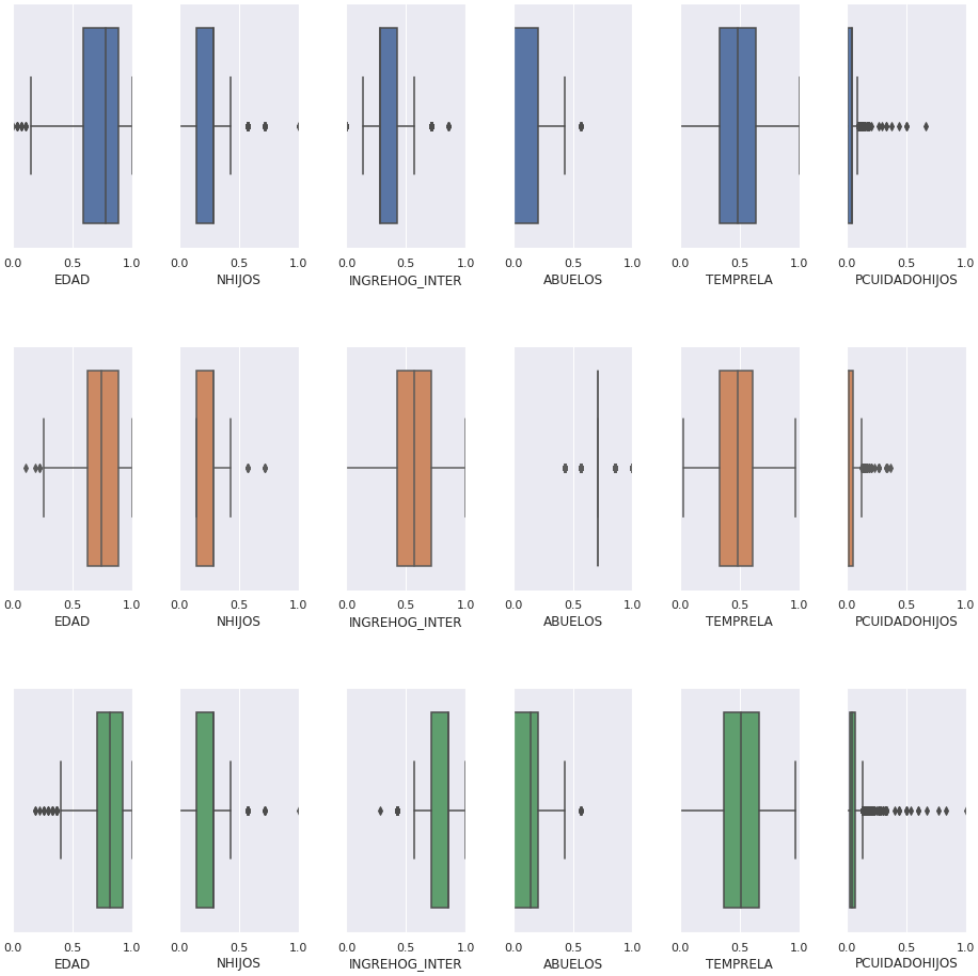

Figura 28: BoxPlot para Agglomerative Clustering con 2 clusters.

2.3.3. Birch.

Ejecutaremos diferentes versiones de **Birch** modificando el parámetro *n_clusters=X* donde *X* tomará diferentes valores entre 2 y 5 y se ha fijado *threshold=0.25* y *branching_factor=25*. El código utilizado ha sido el siguiente:

```
Birch(branching_factor=25, n_clusters=X, threshold=0.25, compute_labels=
    True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
n_clusters	2	1112.446954	0.314184	0.110045
n_clusters	3	1285.731331	0.249516	0.105628
n_clusters	4	1093.248074	0.202604	0.108606
n_clusters	5	909.194187	0.186180	0.111609

Tabla 15: Tabla modificaciones Birch para el Caso 3.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 3 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
2:  2121 (53.59%)
0:  1220 (30.82%)
1:   617 (15.59%)
```

2.3.4. MeanShift.

Ejecutaremos diferentes versiones de **MeanShift** modificando el parámetro *quantile* y fijamos los parámetros *n_samples=400* y *bin_seeding=True*. El código utilizado ha sido el siguiente:

```
MeanShift(bandwidth=estimate_bandwidth(X_normal, quantile=0.999,
    n_samples=X), bin_seeding=True)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
quantile=0.45	2	25.485019	0.295637	1.007742
quantile=0.40	3	577.361702	0.258427	0.695561
quantile=0.38	4	403.921265	0.208743	1.145767

Tabla 16: Tabla modificaciones MeanShift para el Caso 3.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 3 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   2890 (73.02%)
1:   1049 (26.50%)
2:     19 ( 0.48%)
```

2.3.5. DBSCAN.

Ejecutaremos diferentes versiones de **DBSCAN** modificando el parámetro *eps*. El código utilizado ha sido el siguiente:

```
DBSCAN(eps=X)
```

A continuación, se mostrará la tabla compararíva para los valores obtenidos:

Mod	Nº Clusters	CH	SH	Time
eps=0.25	2	27.931281	0.294172	0.171702
eps=0.20	3	32.590419	0.197307	0.126914
eps=0.18	5	35.578137	0.167760	0.120676

Tabla 17: Tabla modificaciones DBSCAN para el Caso 3.

Podemos observar que la versión que obtiene mejores resultados para **CH** es la que obtenemos con 5 clusters en la que la proporción de cada cluster se ha hecho de la siguiente manera:

```
0:   3752 (94.80%)
-1:   186 ( 4.70%)
3:      9 ( 0.23%)
1:      6 ( 0.15%)
2:      5 ( 0.13%)
```

Mostraremos la tabla comparativa que contiene las mejores versiones de cada algoritmo ejecutado. Previamente y conociendo estos resultados, se ha hecho un estudio más profundo sobre aquellos dos algoritmos que han obtenido un mayor valor del coeficiente de Silhouette y **CH**. En este caso, **K-Means** y **Agglomerative Clustering**.

Name	**Nº Clusters**	**CH**	**SH**	**Time**	**Clusters**
K-Means	3	1451.077451	0.267710	0.030554	0: 1643 (41.51 %) 1: 1623 (41.01 %) 2: 692 (17.48 %)
DBSCAN	5	35.578137	0.167760	0.120676	0: 3752 (94.80 %) -1: 186 (4.70 %) 3: 9 (0.23 %) 1: 6 (0.15 %) 2: 5 (0.13 %)
Birch	3	1285.731331	0.249516	0.105628	2: 2121 (53.59 %) 0: 1220 (30.82 %) 1: 617 (15.59 %)
AggCluster	3	1292.676163	0.238999	0.416693	0: 1965 (49.65 %) 2: 1359 (34.34 %) 1: 634 (16.02 %)
MeanShift	3	577.361702	0.258427	0.695561	0: 2890 (73.02 %) 1: 1049 (26.50 %) 2: 19 (0.48 %)

Tabla 18: Tabla comparativa general para el Caso 3.

2.3.6. Interpretación de la segmentación.

Tomando los valores para las métricas **CH** y **SH** el que ofrece una mejor agrupación de los individuos es el algoritmo **K-Means**.

El estudio realizado sobre este algoritmo, divide la población en mujeres casadas de tres tipos, los cuales tienen en común que la edad media de su cluster es entre 37 y 39 años, por lo que podemos concluir que la mayoría de mujeres casadas menores de 45 años están en ese rango de edad, por lo que no será una característica muy representativa a la hora de agrupar los datos. Las tres divisiones en este caso, vendrán determinadas por los ingresos mensuales en el hogar, el tiempo que dejan a los hijos con sus abuelos y el dinero invertido en el cuidado de los hijos.

Las mujeres del primer tipo que representan un 41.51 % de la muestra estudiada, se caracterizan por tener unos ingresos mensuales en el hogar bajos (entre 1000 y 1500

euros de media). Esto, sumado a que tienen de media unos 2 hijos, ocasiona que el dinero invertido en los mismos sea escaso.

Las mujeres del segundo tipo que representan un 41.01 % de la muestra, tienen mayor ingreso mensual, por encima de la media (entre 2500 y 3000 euros de media) por lo que pueden invertir bastante más en el cuidado de sus hijos y no es necesario dejarlos con los abuelos, ya que se podrán pagar actividades que puedan ocupar su tiempo de ocio.

Las mujeres del tercer tipo que representan un 17.48 % de la muestra, tienen un ingreso medio (entre 2000 y 2500 euros mensuales), pero además se caracteriza por dejar a sus hijos 5 días de media con los abuelos, a los cuales usualmente no se les suele remunerar por sus cuidados, por lo que el dinero invertido en los hijos será menor.

El número medio de hijos también es similar en los tres casos (entre 1 y 2 hijos de media), siendo un poco inferior (viendo la Figura 22) en el caso 2, por lo que se pueden invertir más dinero en el cuidado de los hijos.

3. Contenido Adicional.

A continuación, se muestra el código utilizado para crear las gráficas **KPlot** y **BoxPlot** utilizadas para visualizar los resultados de los algoritmos.

```
def KPlot(X, name, k, usadas, path):
  print("\nGenerando kplot...")
  n_var = len(usadas)
  fig, axes = plt.subplots(k, n_var, sharex='col', figsize=(15,10))
  fig.subplots_adjust(wspace=0.2)
  colors = sns.color_palette(palette=None, n_colors=k, desat=None)

  for i in range(k):
    dat_filt = X.loc[X['cluster']==i]
    for j in range(n_var):
      sns.kdeplot(dat_filt[usadas[j]], shade=True, color=colors[i], ax=
          axes[i,j])

  plt.savefig(path+"kdeplot"+name+".png")
  plt.clf()
```

```
def BoxPlot(X, name, k, usadas, path):
  print("\nGenerando boxplot...")
  n_var = len(usadas)
  fig, axes = plt.subplots(k, n_var, sharey=True, figsize=(16, 16))
  fig.subplots_adjust(wspace=0.4, hspace=0.4)
  colors = sns.color_palette(palette=None, n_colors=k, desat=None)
  rango = []

  for i in range(n_var):
    rango.append([X[usadas[i]].min(), X[usadas[i]].max()])

  for i in range(k):
    dat_filt = X.loc[X['cluster']==i]
    for j in range(n_var):
      ax = sns.boxplot(dat_filt[usadas[j]], color=colors[i], ax=axes[i, j
          ])
      ax.set_xlim(rango[j][0], rango[j][1])

  plt.savefig(path+"boxplot"+name+".png")
  plt.clf()
```

Referencias

[1] Diapositivas de clase.

[2] https://en.wikipedia.org/wiki/Silhouette_(clustering)

[3] https://es.mathworks.com/help/stats/clustering.evaluation.calinskiharabaszevaluation-class.html

[4] https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

[5] https://scikit-learn.org/stable/modules/clustering.html