ANÁLISIS RELACIONAL MEDIANTE SEGMENTACIÓN

Ignacio Garach Vélez
igarachv@correo.ugr.es IN Viernes 12:30

11 de diciembre de 2022

Índice

1. Introducción

En esta práctica vamos a explorar las posibilidades del paradigma del aprendizaje no supervisado. Vamos a llevar a cabo un análisis de un conjunto de datos de incendios en Toronto mediante segmentación con técnicas de clustering. La idea será llevar a cabo un breve preprocesado para preparar los datos para aplicar los distintos algoritmos, básicamente eliminar los valores nulos y normalizar para evitar dar importancia mayor a algunas variables. Realizaremos informes sobre 3 casos de estudio fijados a priori normalmente a través de las variables categóricas que no son útiles en los algoritmos de clustering pero sí para fijar estos casos, uno de los casos se fijará en torno a un rango de una variable numérica. Aplicaremos los algoritmos y realizaremos un análisis de los casos seleccionados y sus complementarios para ver como varían los clusters y además una comparación entre los distintos algoritmos mediante métricas como Silhouette o Calinski-Harabasz. Se intentará utilizar variedad de algoritmos para comparar las distintas técnicas estudiadas en teoría, clustering ya sea jerárquico, particional o incremental.

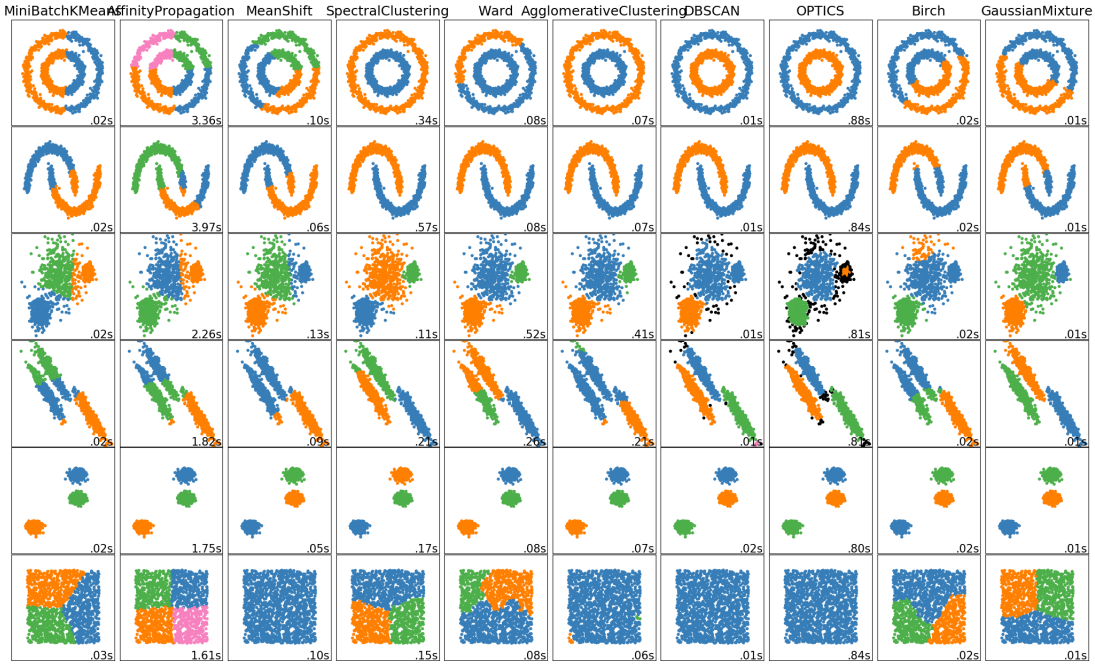

Figura 1: Ejemplo de ejecuciones en datasets sintéticos

Para el escalado se ha optado por usar el objeto MinMaxScaler que lleva los datos al intervalo $[0, 1]$ manteniendo de forma perfecta los ratios de distancia entre datos. Se probó también la estandarización y los resultados no diferían mucho, como no es el objeto de estudio principal, se ha optado por simplificar. En cuanto a la limpieza de datos, sólo había nulos en una columna categórica y se han imputado por la media aunque igualmente se ha optado por no fijar casos de estudios en torno a esa variable, de modo que es irrelevante.

El trabajo se realizará empleando bibliotecas y paquetes de Python, principalmente numpy, pandas, scikit-learn, matplotlib y seaborn.

1.1. Algoritmos considerados

A continuación se realiza una breve descripción de los algoritmos utilizados:

- K-Means: Este algoritmo es iterativo, al comenzar se asignan unos centroides, tantos como número de clusters se indiquen y las instancias se van moviendo entre ellos hasta estabilizarse, en cada iteración se recalculan los centroides y se reasigna cada instancia al cluster con centroide mas cercano en ese momento. Se generan clusters convexos siempre por su naturaleza.
- MeanShift: Alternativa similar a k-Means pero que esta vez decide a priori un radio y cada vez reasigna los centroides a regiones con mayor densidad de instancias. De nuevo, los clusters son convexos y el algoritmo es particional como en el caso anterior.
- Birch: En este caso, estamos ante un algoritmo incremental. Va agrupando los objetos en orden de llegada y trata de mantener las características de los clusters para agrupar las instancias que van llegando. Se genera una especie de árbol que va decidiendo donde colocar cada dato, si no puede absorberse en ninguno cluster se forma uno nuevo.
- AgglomerativeClustering: De nuevo cambiamos de paradigma, en este caso al clustering jerárquico, se genera una jerarquía en la que para cada nivel se obtendrían clusters distintos. Se utiliza la distancia intercluster para fusionar clusters que esten muy cercanos.
- DBSCAN: Sus siglas significan Agrupamiento Espacial Basado en Densidad de Aplicaciones con Ruido, es bastante descriptivo, a partir de un punto busca en su entorno otros puntos similares y los une al cluster, hasta que no pueda alcanzar más puntos dado el tamaño del entorno. Si aparecen puntos no agrupables se agrupan en un cluster específico para ellos.

1.2. Métricas utilizadas

Aunque es difícil considerar medidas objetivas para valorar la calidad del agrupamiento, es razonable considerar que 2 características de un buen agrupamiento son la minimización de la similaridad inter-cluster y la maximización de la similaridad intra-cluster. Es díficil definir cuando algo tiene suficiente nivel de similaridad y en general se suele necesitar el uso de umbrales que implican cierta subjetividad. Principalmente se utilizarán estas 2 extendidas métricas basadas en distancias que miden cohesión y dispersión, además es importante dependiendo del ámbito de aplicación considerar el tiempo de ejecución del algoritmo:

- Coeficiente Silhouette: Compara la smilaridad de los objetos de un cluster comparandolo con otros clusters. Es costoso de calcular en datasets grandes porque para cada dato se calcula la media de distancias al resto de objetos de su grupo y la media de distancias al resto de objetos del resto de clusters. Finalmente se le resta el primer término al segundo y se divide entre el máximo de ellos. Finalmente se calcula la media de este valor en todos los puntos. El agrupamiento es mejor cuanto más se acerque a 1, si un dato tiene coeficiente 0 es que esta en el borde de 2 grupos.
- Índice Calinski-Harabarz: Calcula la razón entre la dispersión intra-clusters y la dispersión interclusters. Cuanto mayor es el valor, mejor es el agrupamiento calculado.
- Tiempo de ejecución del algoritmo.

2. Caso de estudio 1: Tiempo de llegada

En este primer caso, vamos a analizar los clusters que se generan si tomamos por un lado los datos en los casos de llegada más tardía de los bomberos, para que el análisis sea significativo en este dataset en el que los tiempos de llegada son tan rápidos se ha estudiado por percentiles la variable Arrival Time y se ha determinado la separación a partir del percentil 80, es decir los tiempos mayores a 360. Al hacer la separación han resultado 2 conjuntos de 2226 y 8988 incendios respectivamente. De este modo se tratará de encontrar patrones de estos casos para contribuir con medidas destinadas a evitarlos.

2.1. Resultados

Observamos que los resultados obtenidos en este caso de estudio son los siguientes en los distintos algoritmos:

	Algoritmo	Tiempo	Calinski Harabarz	Silhouette
Caso 1	kMeans	0.01	4526.707	0.88939
	MeanShift	1.67	1576.331	0.93577
	Agglomerative	0.09	4474.794	0.88865
	BIRCH	0.03	2341.273	0.95538
	DBSCAN	0.07	1925.762	0.89461
Complementario	kMeans	0.07	17581.506	0.49296
	MeanShift	28.63	1197.317	0.73675
	Agglomerative	2.50	15072.943	0.42879
	BIRCH	0.12	5414.296	0.55060
	DBSCAN	1.28	2305.262	0.70854

En un primer análisis en eficiencia se obtiene que los algoritmos son bastante rápidos en general, en el caso con menos datos, no llegan al segundo, salvo MeanShift que es muy lento con respecto al resto, evidentemente en el caso con mas datos, el tiempo en este algoritmo se dispara a los 28 segundos, algo considerable si se quisiera realizar este tipo de análisis en conjuntos de datos más masivos, aunque en este caso es bastante asumible. El aumento en MeanShift y Aglomerativo nos hace sospechar que la complejidad es superior a la del resto, aunque haría falta llevar a cabo un análisis de la complejidad en profundidad. En cuanto a los resultados del caso de estudio específico, el índice de Calinski Harabarz es superior en el algoritmo de k medias y en el clustering aglomerativo ascendiendo a 4526 y 4474 respectivamente, mas del doble que en el resto indicando que se alcanza un mayor grado de cohesión en los cluster con respecto a la separación con el resto. Sin embargo el coeficiente Silhouette es ligeramente inferior en estos dos casos, indicando que hay más datos dudosos, en el sentido de que estan más cercanos a fronteras de 2 clusters distintos. Por ello se considera que a partir de las métricas objetivas el mejor algoritmo sobre estos datos es k medias. En este caso el número de clusters obtenidos es 3 (ver análisis paramétrico). En el caso complementario, ocurre algo similar, aunque la disminución de Silhouette es mucho mayor con respecto a los demás algoritmos indicando esa debilidad en algunos datos que hemos definido anteriormente, aunque en este caso dado que el otro índice es muy superior respecto al resto, seguimos optando por k medias como el mejor algoritmo seguido del clustering aglomerativo. En este caso el número de clusters obtenidos es 5 (ver análisis paramétrico).

2.2. Análisis paramétrico

En esta sección llevaremos a cabo un análisis de los parámetros de los algoritmos kMeans y Agglomerative Clustering para este caso de estudio y valoraremos la elección óptima para ellos:

2.2.1. kMeans

Para este algoritmo el parámetro fundamental es el número de clusters a calcular que es clave a la hora de valorar los resultados y dotar de significado a los clusteres generados en función de las

variables, hemos considerado un rango de valores de 2 a 9 y calculado las métricas estudiadas. A partir de ellas se han preparado unas gráficas para poder decidir el mejor parámetro:

Número de clusters	CH Index	Silhouette
2	4179.5	0.95524
3	4526.7	0.88939
4	3899.4	0.89105
5	3698.5	0.89299
6	3783.7	0.89835
7	3698.1	0.89573
8	4251.5	0.90255

En el primer caso, se observa que para el grupo de llegada tardía, los mejores valores de las métricas los obtienen para 2 y 3 clusteres, pero hemos considerado utilizar 3, debido al mayor ratio de diferencias y a aprovechar la variabilidad, ya que como veremos proximamente, este algoritmo tiende a obtener mejores métricas únicamente con 2 clusters.

Número de clusters	CH Index	Silhouette
2	6272.1	0.50878
3	12974.1	0.54989
4	16588.0	0.51913
5	17581.5	0.49296
6	17763.1	0.47399
7	17567.7	0.46468
8	16907.1	0.45483

Para el caso complementario sí se observan buenos resultados para un valor más alto del parámetro, resultando en que para unos coeficientes Silhouette e índices de Calinski-Harabarz muy similares el mejor en conjunto para ambos es con 5 clusters, siendo determinante Silhouette en este caso porque al ser mayor el número de clusters es más importante que haya una separación clara y estable entre ellos.

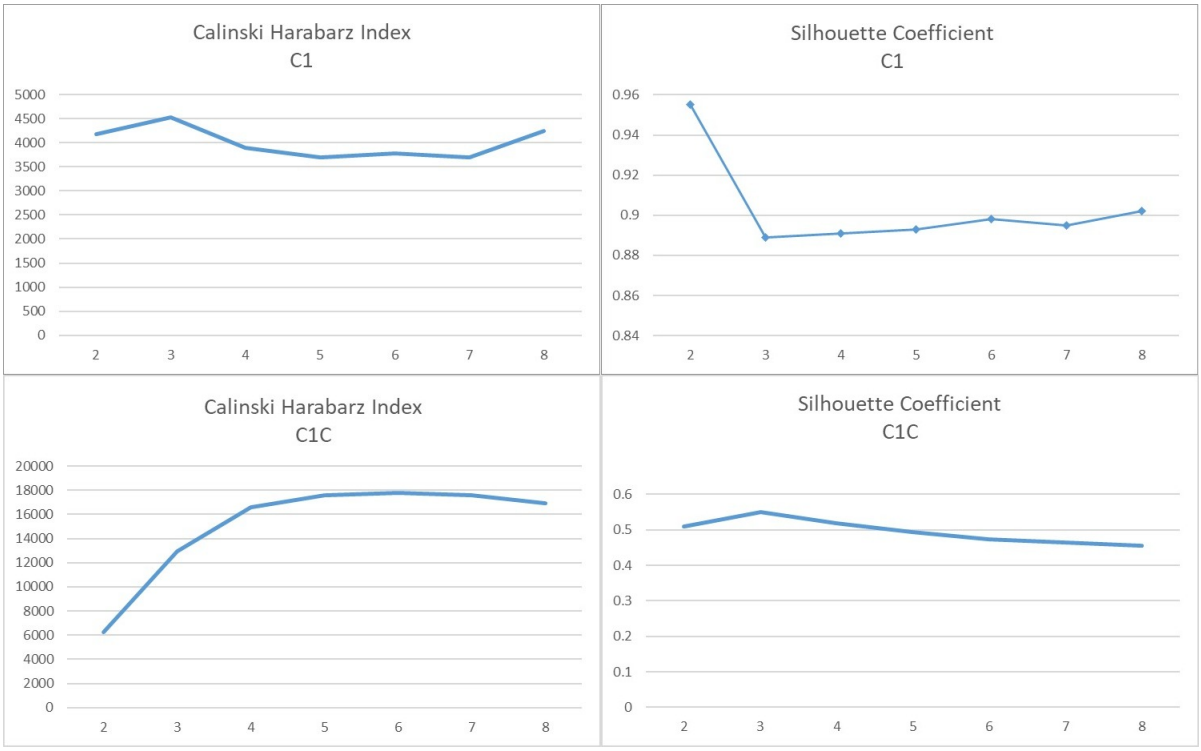

Figura 2: Análisis de parámetros en kMeans

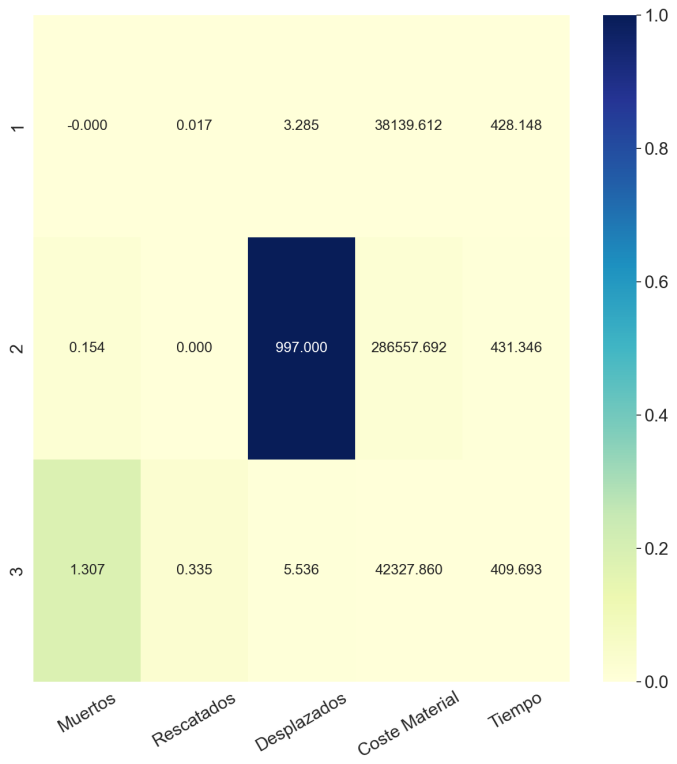

Figura 3: Heatmap de los centroides en el caso 1

Podemos construir estos mapas de calor para los algoritmos basados en centroides como KMeans, en ellos el valore que tiene cada cluster en cada variable indica el centroide que tiene, cuanto mas azul, más determinantes ha resultado ser tal variable en la hora del agrupamiento.

Son bastante indicativos, por ejemplo en el caso de llegada tardía se pueden observar 3 situaciones muy claras, la del cluster 1, que registra los incendios de baja gravedad con valores bajos en todas las variables, en segundo lugar aquellos que dada una llegada tardía, se ha debido de desplazar por precaución a un gran número de individuos aunque luego el número de fallecidos es bajo, y finalmente el tercer grupo determinado por un mayor número de fallecidos que son resultado probable de una tardanza muy alta en el socorro. Dada la separación realizada en el caso de estudio, hay mucha variedad en la variable del tiempo y los clusters mayormente se han formado en torno a rangos de dicha variable, salvo uno de ellos con un número de desplazados muy alto

2.2.2. Agglomerative Clustering

En cuanto al clustering aglomerativo, scikit-learn solo permite ajustar el método de fusión y el número de clusters. En los métodos de fusión se ofrecen los siguientes:

1. ‘ward’: Minimiza la varianza de los clusters unidos.
2. ‘average’: Usa la media de las distancias entre cada observación de los 2 conjuntos.
3. ‘complete’ o ‘maximum’: Utiliza la máxima distancia de entre todas las observaciones de los 2 conjuntos.
4. ’single’: Utiliza la mínima distancia entre todas las observaciones de los 2 conjuntos

Hemos probado con todos y se ha observado que los resultados son sustancialmente mejores con ’ward’ por lo que se fijará, es sensato ya que lo que se busca en 2 clusters a unir es minimizar la varianza como estadístico que mide la variación entre los ejemplos.

Posteriormente vamos a analizar del mismo modo que en kMeans el número de clusters que arrojan mejores métricas en Agglomerative Clustering:

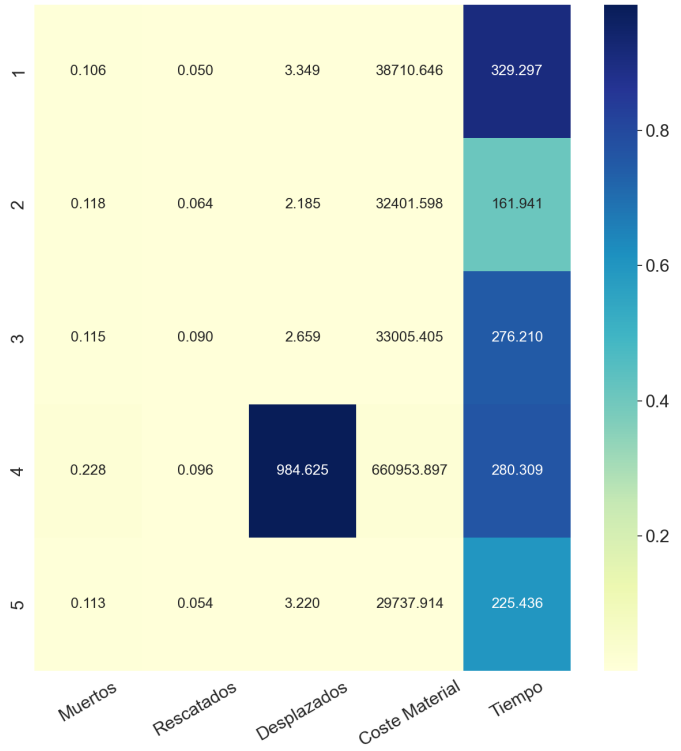

Figura 4: Heatmap de los centroides en el complementario del caso 1

Número de clusters	CH Index	Silhouette
2	4179.51	0.95524
3	4474.7	0.88865
4	3863.1	0.88269
5	3543.9	0.88564
6	3589.87	0.88859
7	3957.88	0.89032
8	4324.6	0.89692

Ocurre algo similar al análisis realizado en kmeans, con mayor valores de las métricas, sobre todo para el índice de Calinski que nos hace reforzar nustra elección anterior de 3 clusters con la misma interpretación antes dada.

Número de clusters	CH Index	Silhouette
2	5994.16	0.50182
3	12443.7	0.54254
4	13490.4	0.44633
5	14131.6	0.42644
6	15072.9	0.42879
7	15098.6	0.42342
8	14659.7	0.42149

Finalmente mostramos el gráfico que indica como se han ido realizando las fusiones en el clustering aglomerativo, un dendrograma donde se muestran las jerarquías:

2.3. Interpretación de la segmentación

Finalmente, en conclusión podemos determinar que los algoritmos tienden a llevar a cabo agrupaciones distintas en el caso tardío y su complementario, los criterios son variados, aunque en el caso de llegadas tempranas, tiende a distinguir en rangos de la variable tiempo y en las tardías ocurran estos 3 claros grupos diferenciados que hemos mencionado antes, el del cluster 1, que registra los incendios

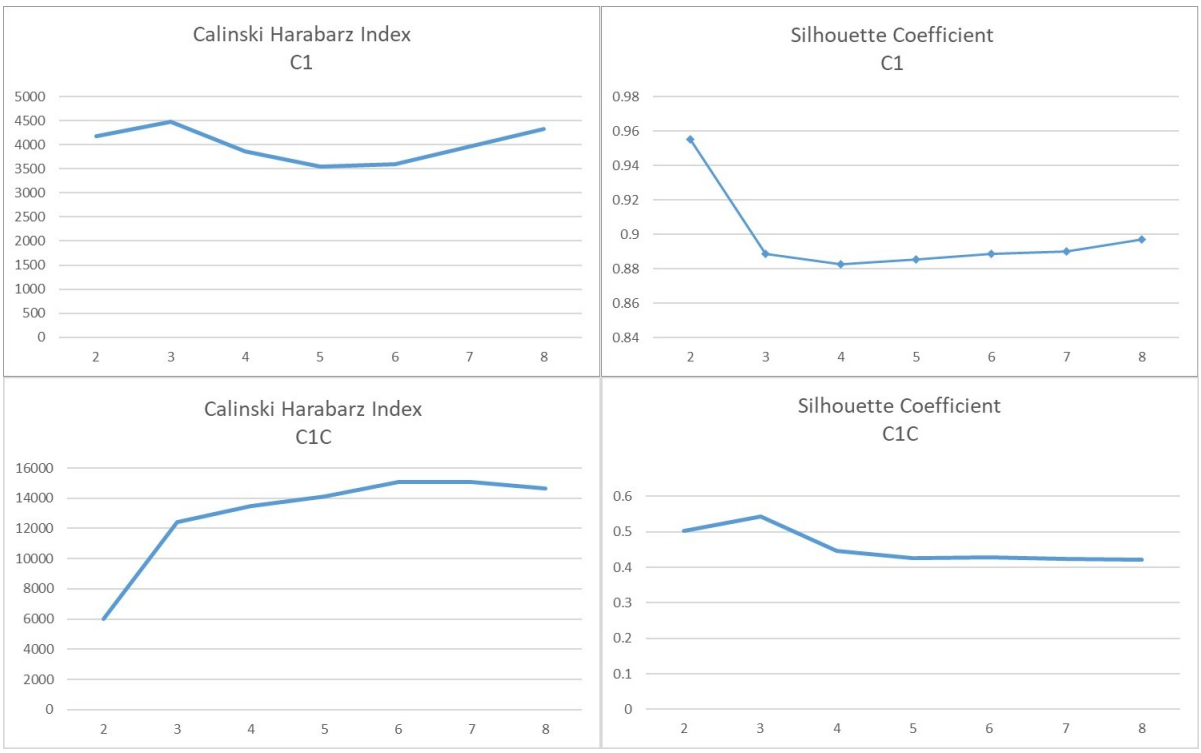

Figura 5: Análisis de parámetros en Agglomerative Clustering

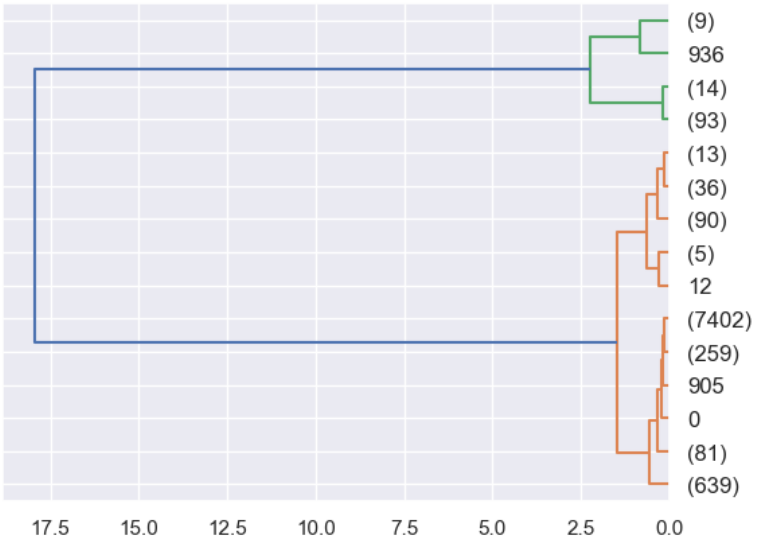

Figura 6: Dendrograma de Agglomerative Clustering en el caso 1

de baja gravedad , en segundo lugar aquellos en que se ha debido de desplazar por precaución a un gran número de individuos aunque luego el número de fallecidos es bajo, y finalmente el tercer grupo determinado por un mayor número de fallecidos.

3. Caso de estudio 2: Presencia de alarma de incendios

En este segundo caso, vamos a analizar los clusters que se generan si tomamos por un lado los datos en los casos que no existía presencia de alarma de incendios, de los que sí. Al hacer la separación han resultado 2 conjuntos de 2569 y 8645 incendios respectivamente. De este modo se tratará de encontrar patrones de estos casos para comprobar como de influyente es la presencia de una alarma en los resultados de la actuación de los equipos de bomberos.

3.1. Resultados

Observamos que los resultados obtenidos en este caso de estudio son los siguientes en los distintos algoritmos:

	Algoritmo	Tiempo	Calinski Harabarz	Silhouette
Caso 2	kMeans	0.02	3699.485	0.88618
	MeanShift	2.72	1999.335	0.87283
	Agglomerative	0.14	3698.354	0.88723
	BIRCH	0.03	1413.593	0.73491
	DBSCAN	0.12	1172.381	0.55019
Complementario	kMeans	0.01	57022.543	0.97294
	MeanShift	9.78	17977.605	0.97307
	Agglomerative	2.16	56105.259	0.97258
	BIRCH	0.12	21270.086	0.97230
	DBSCAN	0.96	28304.643	0.96475

En primer lugar remarcamos que los tiempos de ejecución respetan lo discutido en el anterior caso, además como la partición realizada es similar, los tiempos también lo son. En cuanto a métricas lo obtenido es idéntico al apartado anterior pero hay menos duda en el sentido de que ambas son muy superiores al resto en KMeans y Agglomerative, indicando un alto grado de cohesión, y separación clara entre clusters. También podría indicar que no hemos logrado ajustar bien los parámetros del resto de algoritmos. En el caso complementario, con más datos, el Silhouette del resto de algoritmos también es competitivo, aunque Calinski no lo sea indicando que podría tomarse el resultado de BIRCH o DBSCAN si el número de clusters arrojado tiene una mayor utilidad en la interpretación.

3.2. Análisis paramétrico

En esta sección llevaremos a cabo un análisis de los parámetros de los algoritmos kMeans y Agglomerative Clustering para este caso de estudio y valoraremos la elección óptima para ellos:

3.2.1. kMeans

Para este algoritmo el parámetro fundamental es el número de clusters a calcular que es clave a la hora de valorar los resultados y dotar de significado a los clusteres generados en función de las variables, hemos considerado un rango de valores de 2 a 9 y calculado las métricas estudiadas. A partir de ellas se han preparado unas gráficas para poder decidir el mejor parámetro:

Número de clusters	CH Index	Silhouette
2	3699.5	0.88618
3	3149.3	0.45803
4	3162.1	0.51542
5	2957.4	0.47616
6	2823.9	0.49351
7	2899.9	0.49278
8	2965.7	0.49570

Número de clusters	CH Index	Silhouette
2	57022.5	0.97294
3	45579.9	0.85444
4	39190.2	0.86307
5	37315.9	0.87042
6	35178.4	0.86484
7	34781.4	0.87427
8	37727.5	0.86599

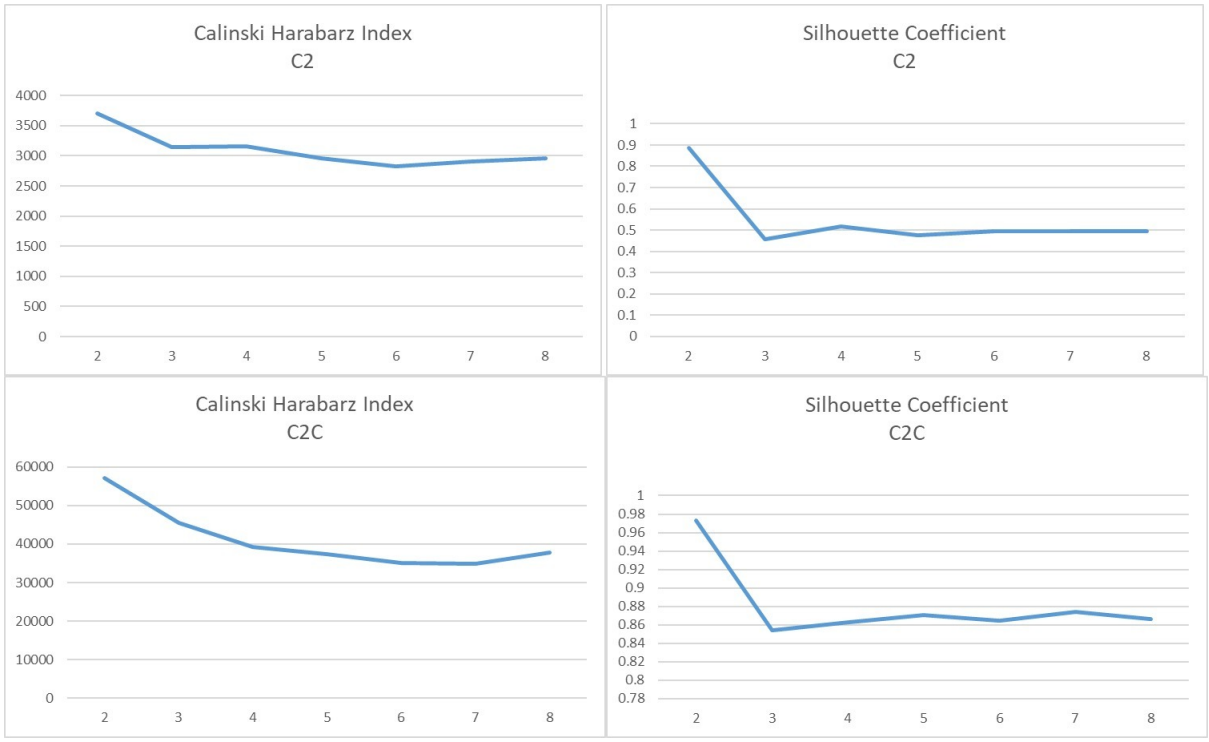

Figura 7: Análisis de parámetros en kMeans

Se hace evidente que en ambos conjuntos de datos generados en esta separación, los clusters generados con mejor calidad dada por las métricas objetivas son 2, consigun un resultado muy claro en ambas sobre todo el resto de rango paramétrico. En el mapa de calor se aprecia que la variable tiempo ha tenido cierto peso en la separación aunque no ha sido determinante, no parece que una diferencia en promedio de 20 segundos en el tiempo de llegada vaya a generar una gran diferencia, lo que si ha resultado determinante han sido los casos del cluster mayoritario que ha sido formado debido a la gran cantidad de desplazados en estos incendios que probablemente hayan resultado mas graves por que el aviso ha sido retardado por la falta de alarma. En el caso de los incendios con alarma activa en el momento del mismo, el resultado es muy similar, aunque la diferencia radica en el tamaño de los clusters, en este segundo caso, el caso con menor transcendencia es mucho más frecuente.

3.2.2. Agglomerative Clustering

Posteriormente vamos a analizar del mismo modo que en kMeans el número de clusters que arrojan mejores métricas en Agglomerative Clustering:

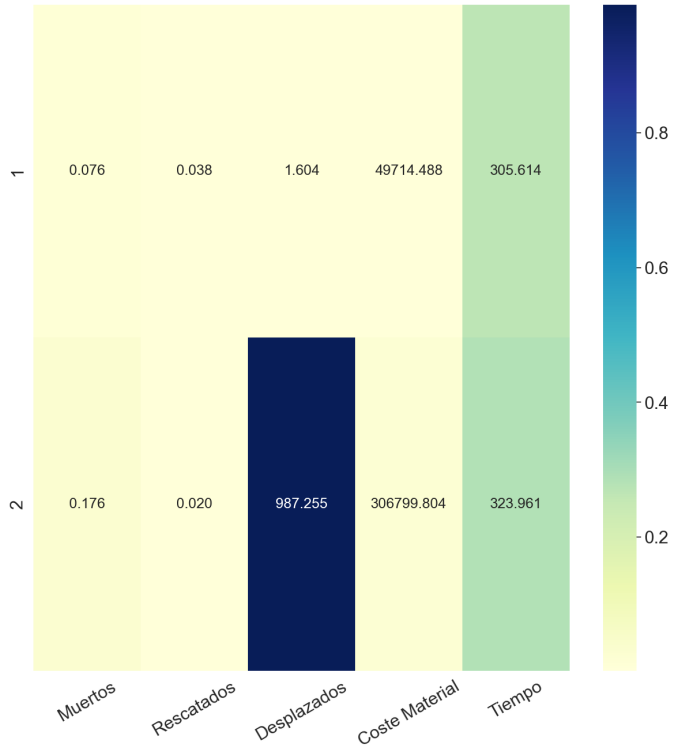

Figura 8: Heatmap de los centroides en el caso 2

Número de clusters	CH Index	Silhouette
2	3698.35	0.88723
3	2971.5	0.44021
4	2988.33	0.51063
5	2788.63	0.43886
6	2764	0.44624
7	2710.9	0.45674
8	2783.52	0.47015

Número de clusters	CH Index	Silhouette
2	56105.3	0.97258
3	44410.5	0.85467
4	36045.8	0.85499
5	32980.6	0.84338
6	31669.9	0.86101
7	31553.5	0.86218
8	33324.5	0.86378

Tanto las tablas como las gráficas obtenidas son prácticamente idénticas por lo que cabe preguntarse tras repetirse esta tendencia como en el caso 1, si el comportamiento de estos 2 algoritmos tiene alguna similaridad siempre a pesar de tratarse de paradigmas distintos, por un lado clustering particional y convexo y por otro aglomerativo y jerárquico.

Las tablas lo dejasn claro, el número de clusters ha de ser 2 aunque se observa que la calidad del cluster en el segundo caso es superior en cuanto a estabilidad y separación con el resto de clusters, tiene un coeficiente medio de Silhouette bastante mas alto en todos los tamaños, incluso en aquellos que funcionan mal para el índice Calinski.

De nuevo, añadimos una visualización de la agrupación jerarquica de Agglomerative Clustering:

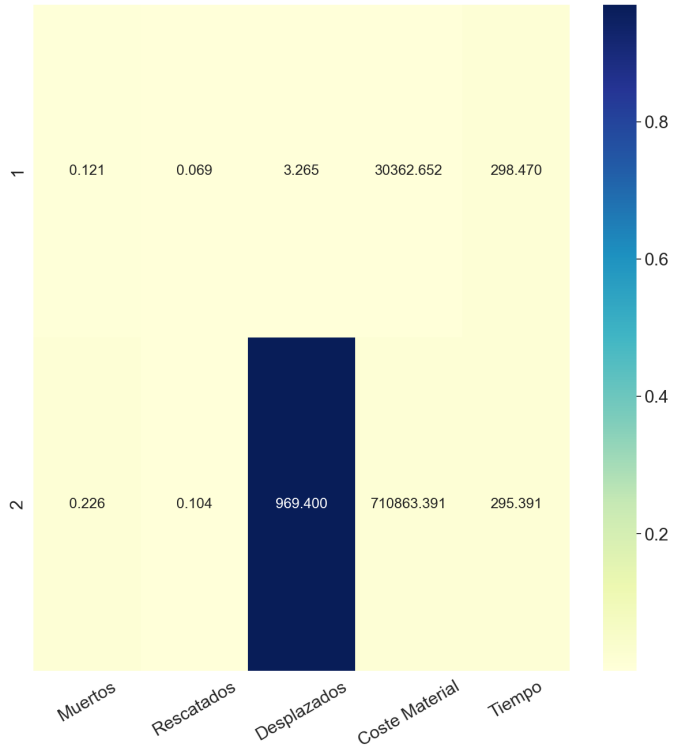

Figura 9: Heatmap de los centroides en el complementario del caso 2

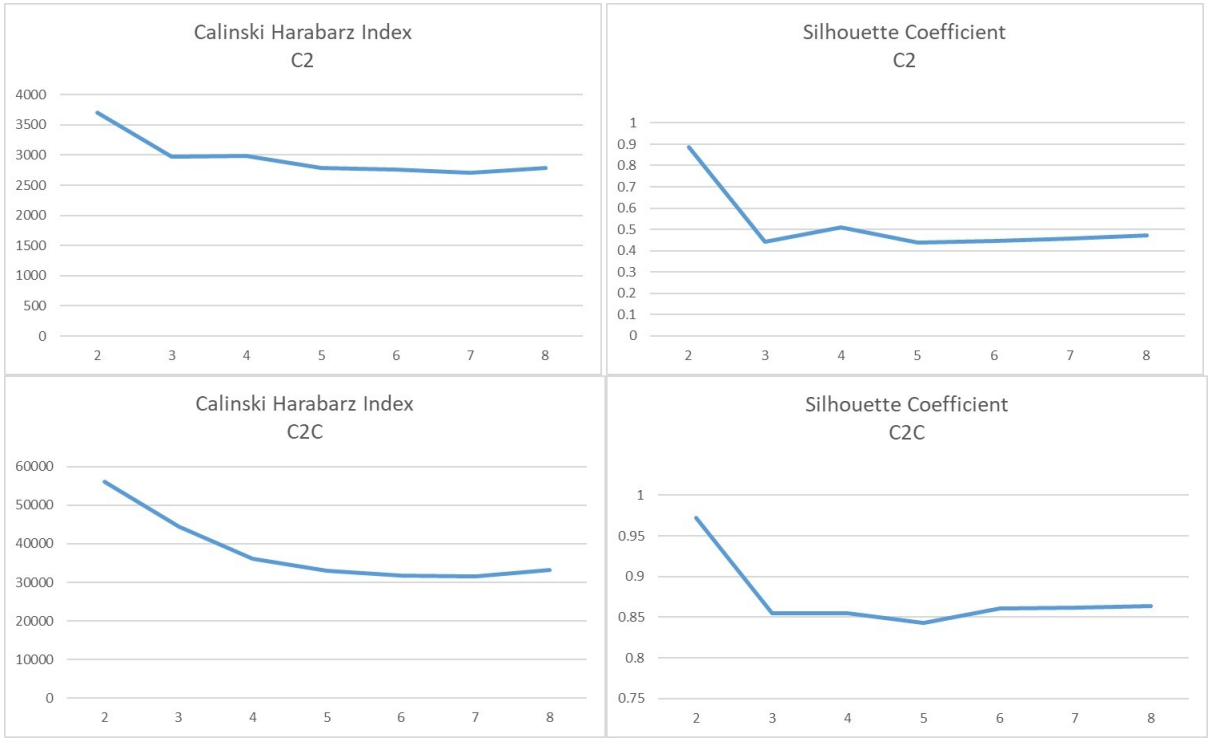

Figura 10: Análisis de parámetros en Agglomerative Clustering

3.3. Interpretación de la segmentación

Para finalizar comentamos que aunque no se han encontrado grandes patrones aparte del señalado por los desplazados en ambos clusters, en este caso es fundamental el tamaño de los mismos, ya que en el caso de la presencia de alarmas se hace patente como esto afecta en que el grupo mayoritario sea aquel con menor número de víctimas y desplazados en promedio.

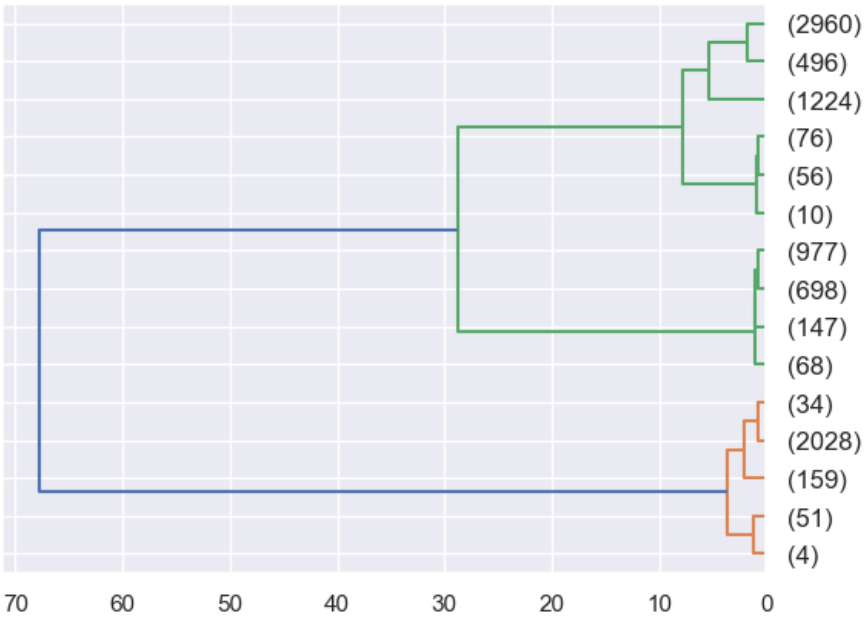

Figura 11: Dendrograma de Agglomerative Clustering en el caso 2

4. Caso de estudio 3: Fines de semana

Finalmente hemos decidido considerar un atributo temporal para el estudio, vamos a considerar por un lado los incendios que se producen en fines de semana y por otro durante las jornadas intersemanales, para analizar las particularidades de ambos casos. Trás hacer la separación se obtienen 3306 y 7908 en cada grupo respectivamente. Queremos observar que patrones son evitables sobre todo en el día a día entre semana, donde ocurren un alto porcentaje de los incendios estudiados.

4.1. Resultados

Observamos que los resultados obtenidos en este caso de estudio son los siguientes en los distintos algoritmos:

	Algoritmo	Tiempo	Calinski Harabarz	Silhouette
Caso 3	kMeans	0.02	2656.300	0.86970
	MeanShift	4.03	818.370	0.83744
	Agglomerative	0.25	2655.632	0.87125
	BIRCH	0.04	1042.027	0.83749
	DBSCAN	0.20	1298.160	0.82537
Complementario	kMeans	0.01	62763.863	0.97376
	MeanShift	8.75	18118.217	0.96680
	Agglomerative	1.88	62414.515	0.97420
	BIRCH	0.10	20221.100	0.97310
	DBSCAN	0.89	30825.851	0.96649

Observando la tabla, podemos llegar a las misma conclusión que en los casos anteriores, los algoritmos aglomerativo y de las k medias son los que mejores métricas obtienen tanto en los clusteres generados entre semana como en los fines de semana. En este caso ocurre como en el caso 2, los valores son muy altos tanto de Silhouette como de Calinski-Harabarz, en especial ocurre en el grupo más numeroso intersemanal, donde el grado de separación entre los clusteres se presupone más alto debido a la diversidad de casos y numerosidad de casos sin mayor trascendencia posterior, bajos valores de muertos, desplazados, gastos, etc...

Entre semana, el clustering aglomerativo ha sido el mejor algoritmo mientras que los fines de semana ha resultado ser kMeans aunque por poca ventaja. Esta superioridad del algoritmo de las k

medias en todos los casos seleccionados nos hace ver que la restricción de la formación de sólo clusteres convexos en este algoritmo no parece una desventaja, al menos en problemas como este donde además se han normalizado las variables para no dotar de mayor importancia a unas sobre otras. En este conjunto es posible plantearse el uso de DBSCAN pues obtiene métricas decentes aunque no cerca de las 2 dominantes

4.2. Análisis paramétrico

En esta sección llevaremos a cabo un análisis de los parámetros de los algoritmos kMeans y Agglomerative Clustering para este caso de estudio y valoraremos la elección óptima para ellos:

4.2.1. kMeans

Para este algoritmo el parámetro fundamental es el número de clusters a calcular que es clave a la hora de valorar los resultados y dotar de significado a los clusteres generados en función de las variables, hemos considerado un rango de valores de 2 a 9 y calculado las métricas estudiadas. A partir de ellas se han preparado unas gráficas para poder decidir el mejor parámetro:

Número de clusters	CH Index	Silhouette
2	2656.3	0.86970
3	3407.2	0.47409
4	3128.0	0.50540
5	3286.4	0.45891
6	3186.5	0.45307
7	3230.6	0.46515
8	3222.9	0.46410

En este caso se observa que pese a que Silhouette otorga muy buena puntuación a 2 clusters, la otra métrica le da mayor valoración a 3 y para facilitar la interpretación y tratar de obtener agrupamientos útiles (medida subjetiva de la calidad de la segmentación) se escogeran 3.

Número de clusters	CH Index	Silhouette
2	62763.9	0.97376
3	48004.3	0.85679
4	41918.5	0.86516
5	38469.3	0.87112
6	36413.1	0.85702
7	37410.9	0.85571
8	39721.4	0.87731

En este caso suplementario, las métricas no nos dan esa opción, son muy superiores para 2 clusters, veamos que podemos examinar en los mapas de calor basados en centroides.

En los fines de semana ocurre una segmentación clara en 3 grupos, aunque podrían unirse en 2 tal como indicaban las métricas. Por un lado se tienen los casos con mas pronta llegada de los bomberos que probablemente por ello tienen asociados bajos valores en las variables de daños humanos y económicos, por otro obtenemos un grupo con un gran número de desplazados, aunque minoritario, también esta asociado a un tiempo de llegada mayor, finalmente ek grupo que dijimos podría asociarse al primero tiene un tiempo de llegada mayor pero el incendio es menos grave y esta bien representado por los bajos valores del primer cluster. Por otro lado entre semana, el comportamiento si que queda claramente asociado al número de desplazados, dada la repetición, parece tratarse de una variable muy determinante a la hora de crear agrupaciones, es posible a que esto se deba a la alta tendencia a que el número sea bajo, y cuando es alto varias veces parezca significativo en el modo de actuar de los algoritmos.

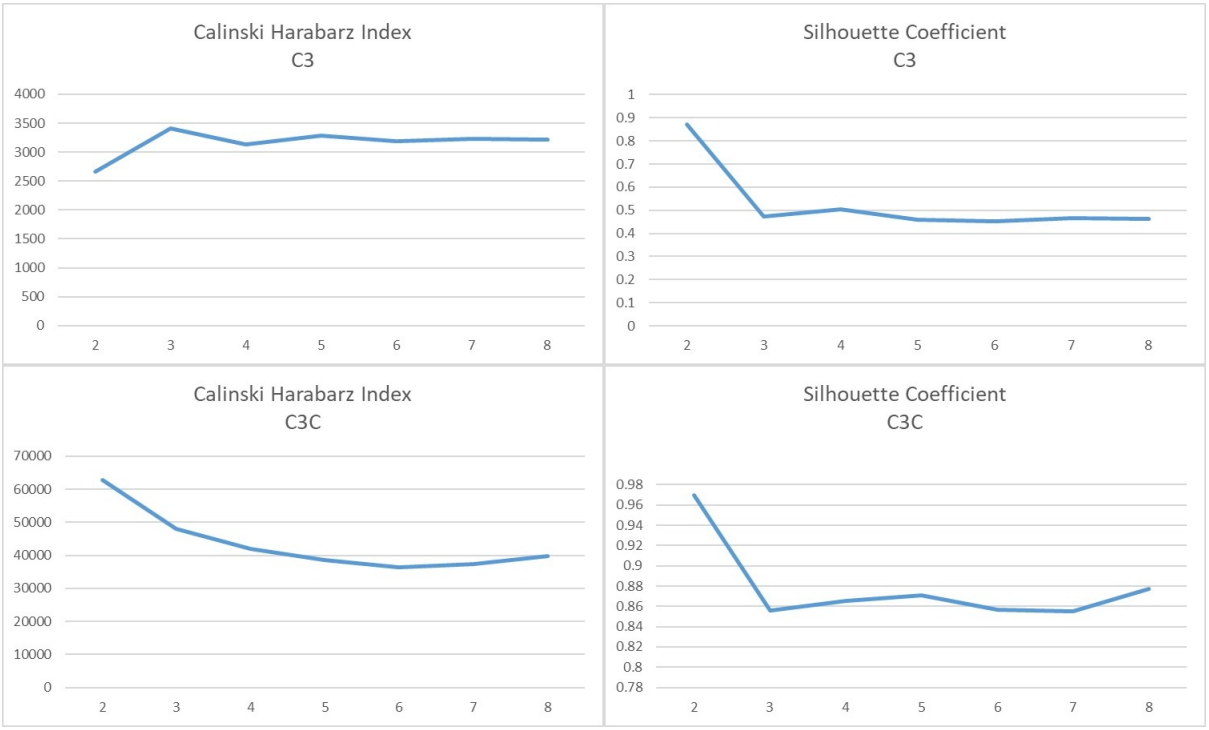

Figura 12: Análisis de parámetros en kMeans

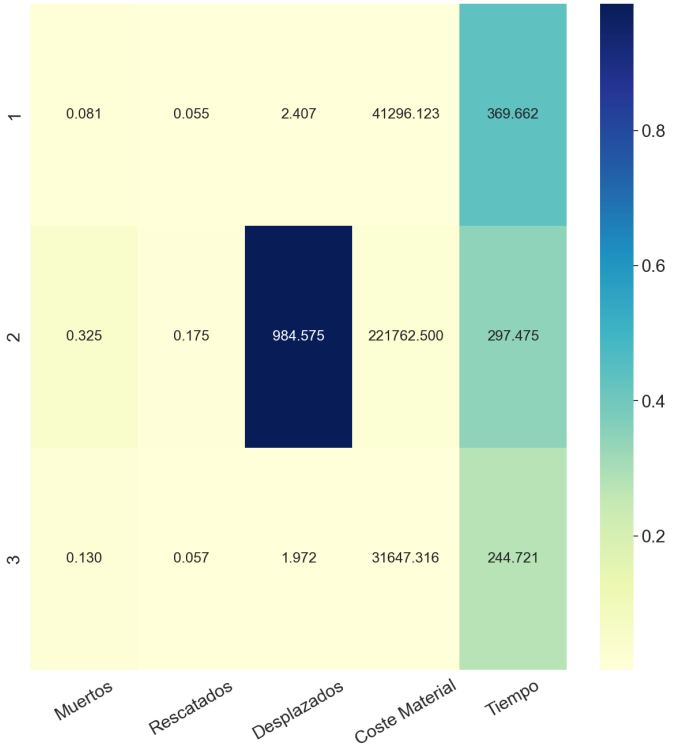

Figura 13: Heatmap de los centroides en el caso 3

4.2.2. Agglomerative Clustering

Posteriormente vamos a analizar del mismo modo que en kMeans el número de clusters que arrojan mejores métricas en Agglomerative Clustering:

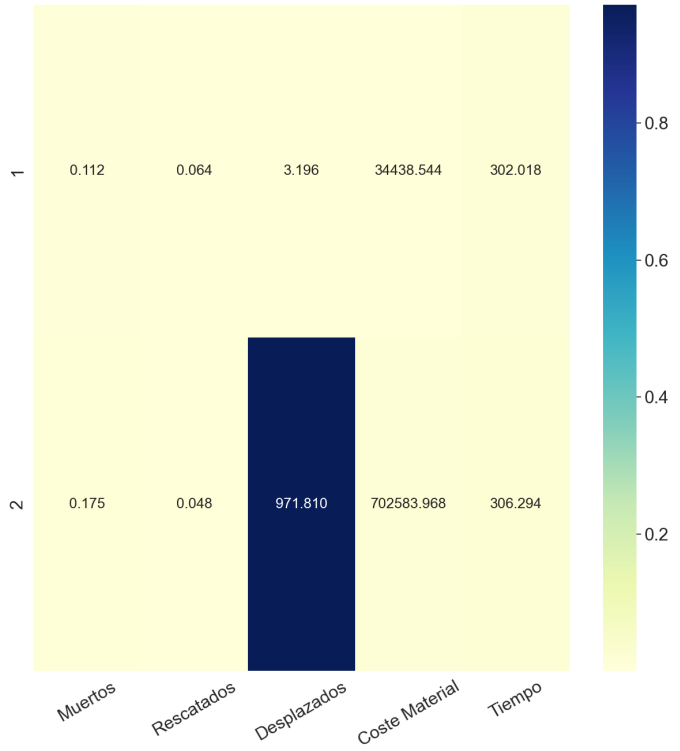

Figura 14: Heatmap de los centroides en el complementario del caso 3

Número de clusters	CH Index	Silhouette
2	2655.63	0.87125
3	3101.74	0.43564
4	2926.61	0.38083
5	3190.84	0.45088
6	2950.35	0.44922
7	2866.27	0.45771
8	2844.4	0.38046

Número de clusters	CH Index	Silhouette
2	62414.5	0.9742
3	47413.1	0.85384
4	41779.8	0.86548
5	37842.7	0.85544
6	36157.7	0.8556
7	37098.7	0.85421
8	40832.3	0.85601

En ambos casos el clustering aglomerativo otorga una mayor puntuación al tamaño 2 reafirmando lo ocurrido en el caso anterior y quizá indicando que nos hemos equivocado en la toma de 3 clusters de kmeans, cuando 2 parecen explicar la falta de variabilidad en los datos, distinguiendo sólo un cluster minoritario con los incendios graves o grandes del resto.

Añadimos como anteriormente el dendrograma correspondiente al agrupamiento en el caso complementario pues muestra un comportamiento algo diferente a los otros 2 anteriormente comentados.

4.3. Interpretación de la segmentación

Para finalizar este caso de estudio nos hemos dado cuenta que probablemente no ha sido demasiado interesante, ni por si mismo ni en la comparación, no se pueden extraer conclusiones claras salvo quizá

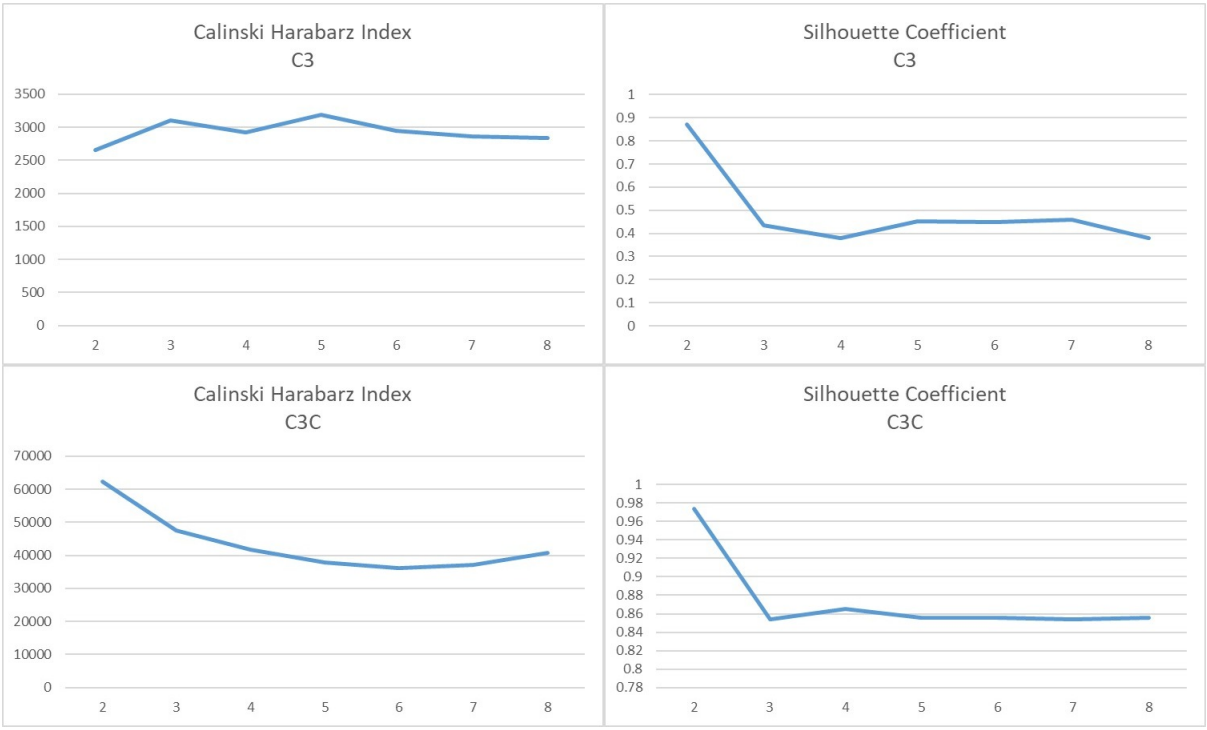

Figura 15: Análisis de parámetros en Agglomerative Clustering

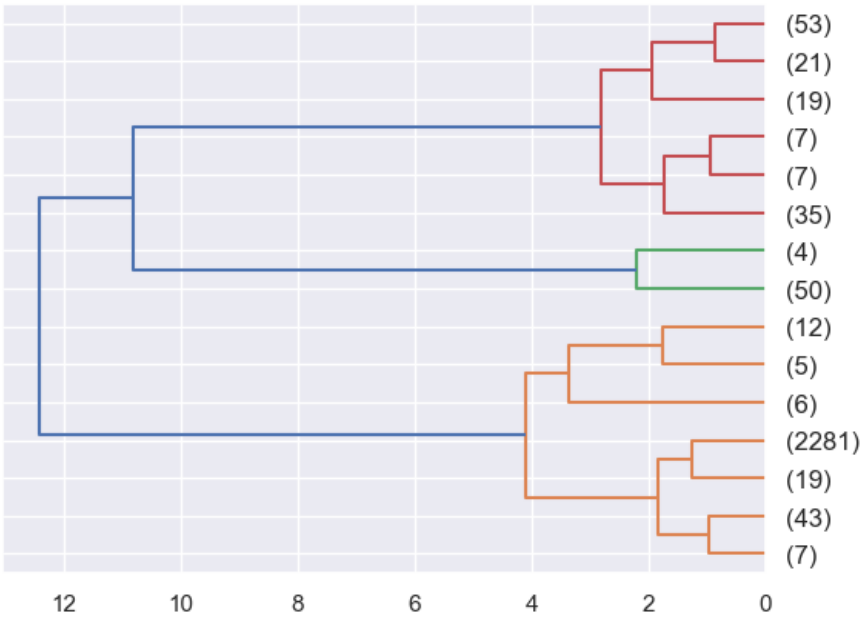

Figura 16: Dendrograma de Agglomerative Clustering en el caso 3

el mayor número de incendios graves (en el sentido de que provocan desplazados) en fines de semana. Por ello podemos considerarlo un estudio fallido, no hemos aportado nada con el clustering que no pudieramos haber conseguido mucho más facilmente con un análisis de estadísticos descriptivos de los datos. Con esto queremos concluir que no siempre es necesaria la realización de análisis profundos, muchas veces la simple estadística es la respuesta en la búsqueda de patrones y variaciones significativas

5. Bibliografía

- Código proporcionado por los profesores de la asignatura para las gráficas.
- http://scikit-learn.org/stable/modules/clustering.html
- https://stackabuse.com/hierarchical-clustering-with-python-and-scikit-learn/
- https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html
- https://joernhees.de/blog/2015/08/26/scipy-hierarchical-clustering-and-dendrogram-tutorial/