SEGMENTACIÓN PARA ANÁLISIS EMPRESARIAL

Práctica2

UNIVERSIDAD DE GRANADA

Inteligencia de Negocio

Ismael Sánchez García

ÍNDICE

1. INTRODUCCIÓN 2
2. RESULTADOS OBTENIDOS 2
2.1. CASO DE ESTUDIO 1 3
RESULTADOS PARÁMETROS POR DEFECTO 4
INTERPRETACIÓN DE LA SEGMENTACIÓN 10
2.2. CASO DE ESTUDIO 2 11
RESULTADOS PARÁMETROS POR DEFECTO 12
INTERPRETACIÓN DE LA SEGMENTACIÓN 19
2.3. CASO DE ESTUDIO 3 19
RESULTADOS PARÁMETROS POR DEFECTO 19
INTERPRETACIÓN DE LA SEGMENTACIÓN 26
3. CONTENIDO ADICIONAL 27
4. BIBLIOGRAFÍA 27

1. INTRODUCCIÓN

- El problema que vamos a abordar consiste en la utilización de técnicas de aprendizaje no supervisado para el análisis empresarial.
- Una compañía aseguradora quiere comprender mejor las dinámicas en accidentes de tráfico en España. Para ello, a partir de diversas variables que caracterizan el accidente, se pretende encontrar grupos de accidentes similares y relaciones de causalidad que expliquen los tipos y gravedad de los accidentes.
- El conjunto de datos utilizado será los datos sobre accidentes de tráfico del año 2013 publicados por la Dirección General de Tráfico (DGT).
- El conjunto de datos contiene 32 variables y 89519 accidentes.
- Vamos a utilizar, a partir de todo el conjunto de datos, al menos 3 subconjuntos distintos como grupos de interés basados en las variables categóricas.
- Para cada caso de estudio vamos a utilizar 5 algoritmos distintos de agrupamiento, para los cuales obtendremos el tiempo de ejecución de cada uno y métricas de rendimiento, Silhouette y el índice Calinski-Harabaz.
    - **Silhoutte**[1]: se calcula utilizando la distancia media dentro del cluster (a) y la distancia media del cluster más cercano (b) para cada muestra. Es la distancia entre una muestra y el cluster más cercano del que no es parte esa muestra. El mejor valor es 1 y el peor valor es -1. Los valores cercanos a 0 indican clusters superpuestos. Los valores negativos generalmente indican que se ha asignado una muestra al cluster incorrecto, ya que un cluster diferente es más similar.
    - **Canlinski-Harabaz**[2,3]**:** se define como la relación entre la dispersión dentro del conglomerado y la dispersión entre conglomerados.

      Al igual que la mayoría de *los criterios de agrupación interna*, Calinski-Harabasz es un dispositivo heurístico. La forma correcta de usarlo es comparar las soluciones de agrupamiento obtenidas en los mismos datos, soluciones que difieren en el número de clusters o en el método de agrupamiento utilizado.

      No hay un valor de corte "aceptable". Simplemente compara los valores de CH a simple vista. Cuanto mayor sea el valor, "mejor" es la solución. Si en el gráfico de línea de los valores de CH aparece que una solución proporciona codo máximo o al menos abrupto, selecciónelo. Si, por el contrario, la línea es lisa, horizontal, ascendente o descendente, entonces no hay ninguna razón para preferir una solución para los demás.

      El criterio de CH es el más adecuado en el caso de que los clusters sean más o menos esféricos y compactos en el medio (como normalmente distribuidos, por ejemplo). Como otras condiciones son iguales, CH tiende a preferir las soluciones de cluster con agrupaciones que consisten aproximadamente en el mismo número de objetos.

2. RESULTADOS OBTENIDOS

- He elegido 3 casos de estudios en los que los accidentes, en principio, no tienen relación directa entre ellos, de modo que podamos entender mediante este estudio 3 casos bien diferenciados en los que la causalidad que los provoca pueda ser distinta, o quizás encontrar circunstancias similares entre ellos.

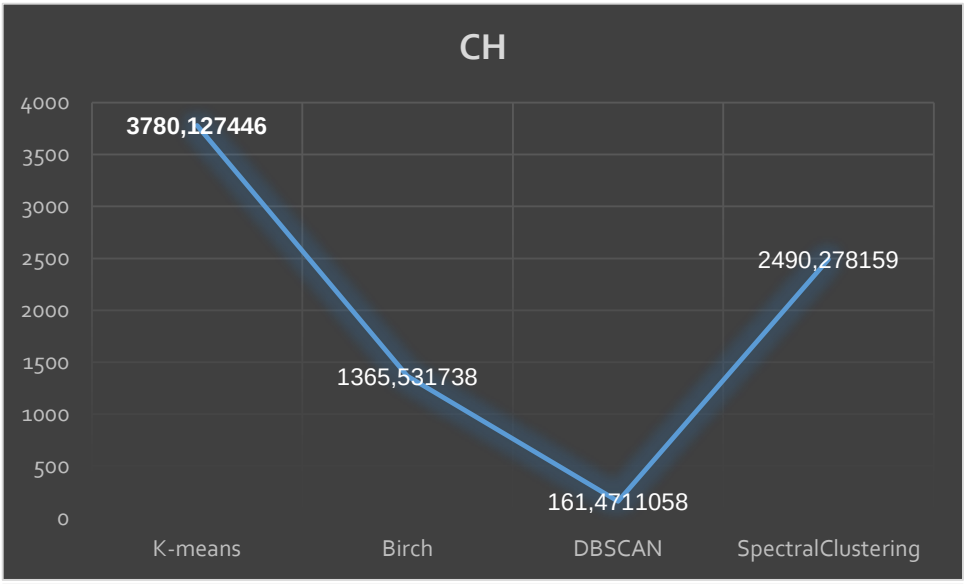
- El número de cluster para los algoritmos en los que hay que fijarlo, está fijado en 4 por defecto.
- He utilizado 3 algoritmos en los que se especifica el número de clusters a priori (k-means, SpectralClustering, Birch), 1 en los que el número de clusters lo determina por sí mismos (DBSCAN), y por último, he usado el jerárquico 'Ward', en el que utilizaremos un valor mínimo de elementos en un cluster una vez ejecutado, para así eliminar los outliders que caen en clusters muy pequeños.
- **K-means[4]:** Dado un conjunto de observaciones (x1, x2, ..., xn), donde cada observación es un vector real de d dimensiones, k-means construye una partición de las observaciones en k conjuntos (k ≤ n) a fin de minimizar la suma de los cuadrados dentro de cada grupo.
- **SpectralClustering[5]**: Las técnicas 'agrupamiento espectral' hacen uso del espectro(valores propios) de la [matriz [similitud]] de los datos para realizar reducción de dimensionalidad antes de la agrupación en un menor número de dimensiones. La matriz de similitud se proporciona como una entrada y consta de una evaluación cuantitativa de la similitud relativa de cada par de puntos en el conjunto de datos. Aplica la agrupación en una proyección al laplaciano normalizado. En la práctica, la Agrupación espectral es muy útil cuando la estructura de los conglomerados individuales es altamente no convexa o, en términos más generales, cuando una medida del centro y la dispersión del conglomerado no es una descripción adecuada del conglomerado completo. Por ejemplo, cuando los clústeres son círculos anidados en el plan 2D. Si la afinidad es la matriz de adyacencia de un gráfico, este método se puede usar para encontrar cortes de gráfico normalizados.
- **Birch[6]:** Es un algoritmo de aprendizaje de eficiencia lineal en la memoria que se ofrece como alternativa a MiniBatchKMeans. Construye una estructura de datos de árbol con los centroides de clúster que se leen de la hoja. Estos pueden ser los centroides del clúster final o se pueden proporcionar como entrada a otro algoritmo de agrupamiento como AgglomerativeClustering.
- **DBSCAN[7]:** Agrupamiento espacial basado en densidad de aplicaciones con ruido. Encuentra muestras de núcleos de alta densidad y expande clusters de ellas. Bueno para los datos que contienen grupos de densidad similar.
- **Ward[8]:** Recursivamente fusiona el par de clústeres que menos incrementa la varianza dentro del clúster.

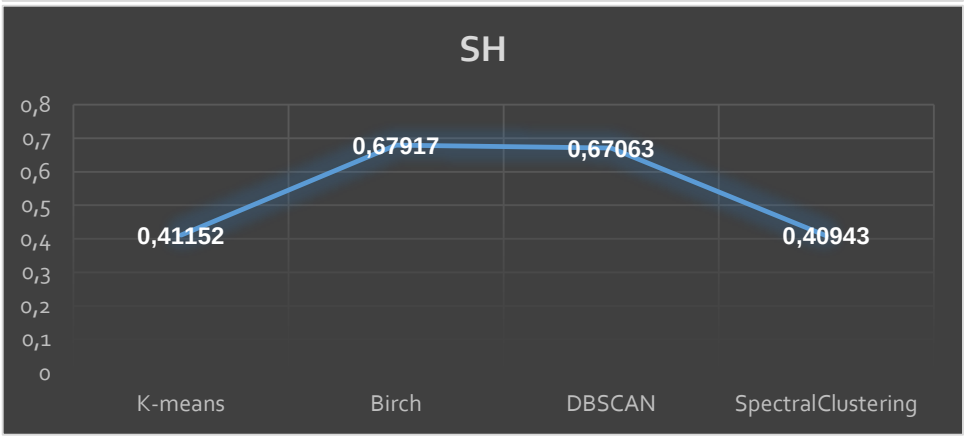
2.1. CASO DE ESTUDIO 1

- El primer caso de estudio será sobre accidentes donde ha habido colisión de vehículos por alcance, en concreto, colisión de vehículos por alcance, donde el tipo de vía es autovía o autopista.
- Es interesante conocer en qué circunstancias se producen este tipo de accidentes en mayor o menor medida y que factores son los que más afectan.
- Las variables utilizadas son: 'HORA', 'DIASEMANA', 'TOT_VICTIMAS'. Utilizamos estas variables para determinar si ciertos días de la semana o en ciertas horas se producen más este tipo de accidentes, además de conocer que influencia puede tener sobre el número de víctimas.
- Nº de casos: 3222
- Colisión accidentes: 169
- Colisión accidentes: 3334

RESULTADOS PARÁMETROS POR DEFECTO

Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	4	0,08	3780,127446	0,41152
Birch	4	0,07	1365,531738	0,67917
DBSCAN	3	0,24	161,4711058	0,67063
SpectralClustering	4	76,51	2490,278159	0,40943
Ward	4	0,38	0	0



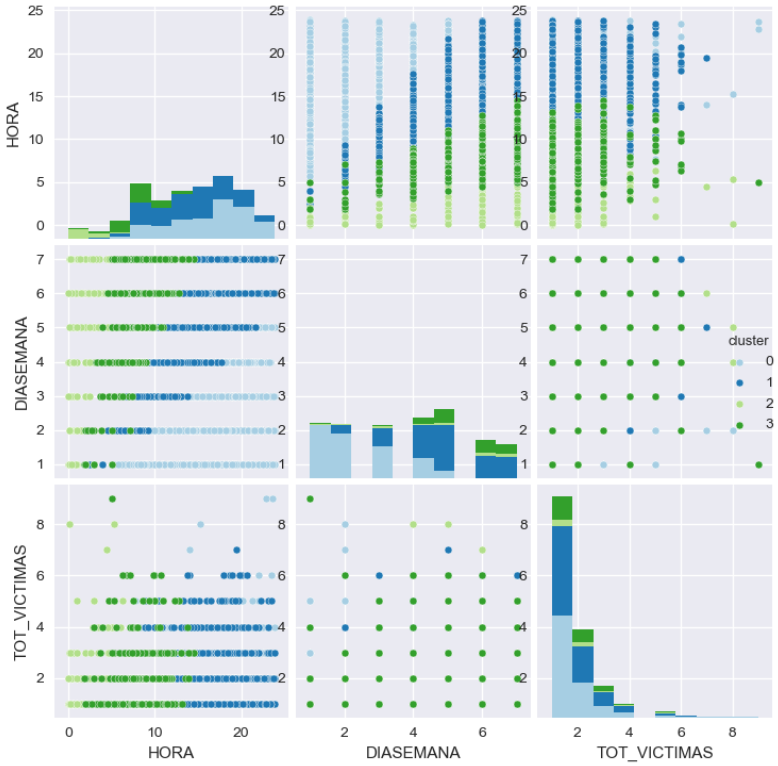


- El tiempo utilizado por Birch y k-means es claramente el menor. Igualmente, DBSCAN y Ward también tienen tiempos muy bajos, aunque debemos considerar que el número de ejemplos no es muy grande por lo que esa diferencia en grandes volúmenes de datos podría ser muy grande. El algoritmo SpectralClustering es claramente el más lento, que como acabo de decir, con el número de ejemplos que tenemos, es un tiempo asumible, pero que en grandes volúmenes de datos podría tardar tanto que no fuese un algoritmo funcional. Esta apreciación de los tiempos de computo, será igualmente válida para cualquier experimentación que haga durante la práctica.

- Atendiendo a las métricas de rendimiento podemos observar como k-means con los parámetros por defecto obtiene el mejor resultado en Calinski-Harabaz (cuanto

mayor, mejor), y un rendimiento bastante pobre en Silhouette (Cuanto más cercano a 1, mejor), lo que nos indica que seguramente tenga clusters superpuestos. Por otro lado, es un rendimiento similar a SpectralClustering que obtiene peores resultados en Calinski-Harabaz. DBSCAN y BIRCH obtienen el mejor resultado en Silohuette (tendrán clusters mejor diferenciados), aunque DBSCAN el peor en Calinski-Harabaz, mientras que BIRCH solo ligeramente mejor.

- Dado que k-means puede ser el que tenga unos resultados más equilibrados entre ambas métricas con los parámetros por defecto ,vamos a analizar sus clusters mediante un scatter matrix y así entender algo mejor el problema.



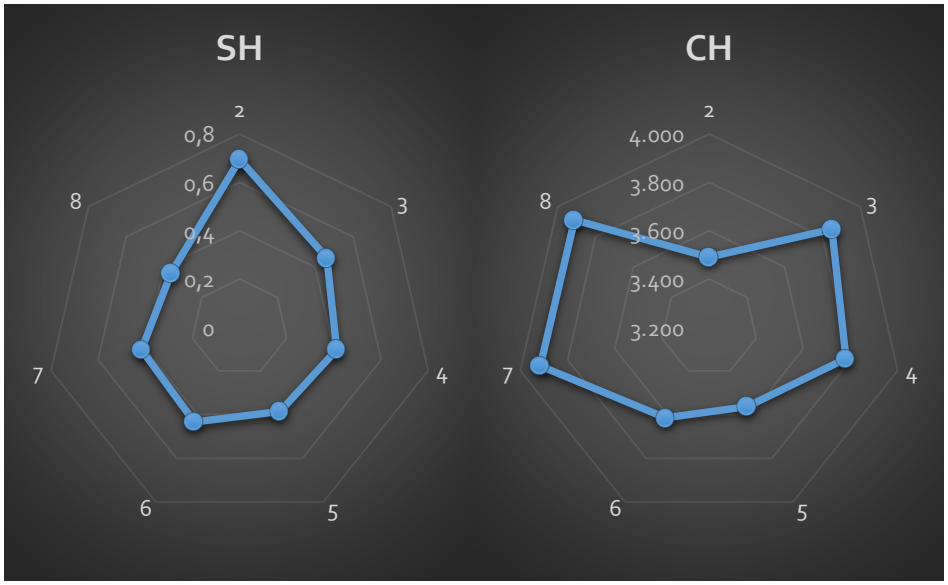
0. Podemos observar como el cluster (0) tiene más presencia en los primeros días de la semana y conforme avanza la semana va disminuyendo el rango de horas en el que está presente, que va desde las 5 hasta las 24 horas el lunes hasta el viernes que prácticamente tiene una sola hora a las 24. En cuanto al número de víctimas, la mayor parte son entre 0-4, excepto algún caso aislado con hasta más de 8 víctimas, que podríamos pensar que son outlaiers.
1. El cluster (1) está presente entre las horas centrales y finales del día, teniendo más presencia en las horas finales. También se concentra más en los días de mitad de la semana en adelante, sobre todo en el fin de semana. Las víctimas se concentran de forma similar que el cluster (0).
2. El cluster (2) está presente entre las 0-4 horas de la noche y más presente conforme avanza la semana, siendo un cluster con un número pequeño de ejemplos. El número de víctimas está entre 0-2 casi en su totalidad.
3. El cluster (3) está presente entre las primeras horas del día y las 15 horas, teniendo más presencia al igual que el cluster (1) conforme avanza la semana. El número de víctimas está entre 0-6, siendo la mayor parte entre 0-2.

- De forma general podemos observar como los cluster no tienen unas separaciones claras. Por ejemplo los clusters (0 y 1) están muy mezclados en cualquier variable que

observemos, por lo que quizás 4 no sea el mejor número de clusters o no se haya realizado una buena separación de los datos. A continuación, realizaremos distintas ejecuciones de los algoritmos, modificando sus parámetros.

K-MEANS

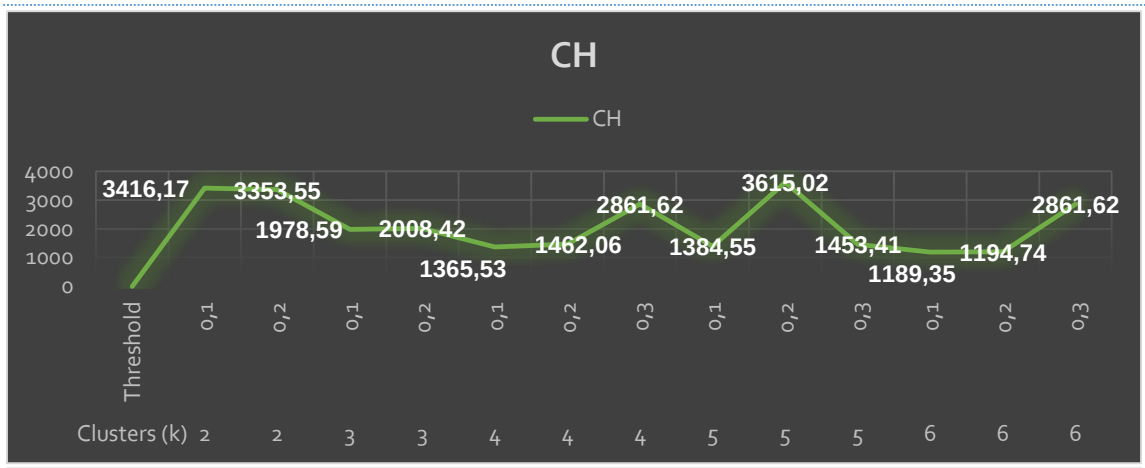
Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	2	0,02	3.491	0,6946
K-means	3	0,04	3850,26517	0,46111
K-means	4	0,08	3780,12745	0,41152
K-means	5	0,05	3562,99843	0,38502
K-means	6	0,06	3615,01685	0,43333
K-means	7	0,08	3915,18845	0,41577
K-means	8	0,12	3911,83302	0,3615

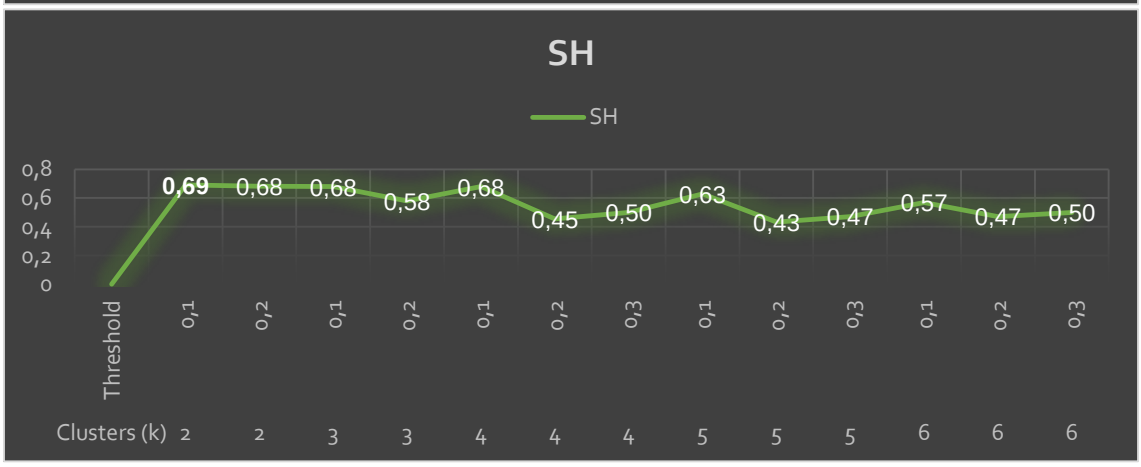


- Parece que el número de cluster que obtienen un mejor rendimiento teniendo en cuenta ambas métricas, puede ser con 6 clusters, que es el más equilibrado entre la dispersión de los datos y la superposición de ejemplos. Otra posibilidad sería con 2 clusters, que es el que mayor valor tiene en SH, lo que indica que hay menos superposición entre elementos y además el valor de CH no es mucho menor que el resto. Con la opción de 2 clusters obtendríamos un modelo más sencillo y por tanto más facilmente entendible.

BIRCH

Algoritmo	Clusters (k)	Threshold	Tiempo (segundos)	CH	SH
BIRCH	2	0,1	0,07	3416,17	0,69
BIRCH	2	0,2	0,07	3353,55	0,68
BIRCH	3	0,1	0,07	1978,59	0,68
BIRCH	3	0,2	0,07	2008,42	0,58
BIRCH	4	0,1	0,07	1365,53	0,68
BIRCH	4	0,2	0,06	1462,06	0,45
BIRCH	4	0,3	0,06	2861,62	0,50
BIRCH	5	0,1	0,06	1384,55	0,63
BIRCH	5	0,2	0,06	3615,02	0,43
BIRCH	5	0,3	0,05	1453,41	0,47
BIRCH	6	0,1	0,05	1189,35	0,57
BIRCH	6	0,2	0,05	1194,74	0,47
BIRCH	6	0,3	0,1	2861,62	0,50

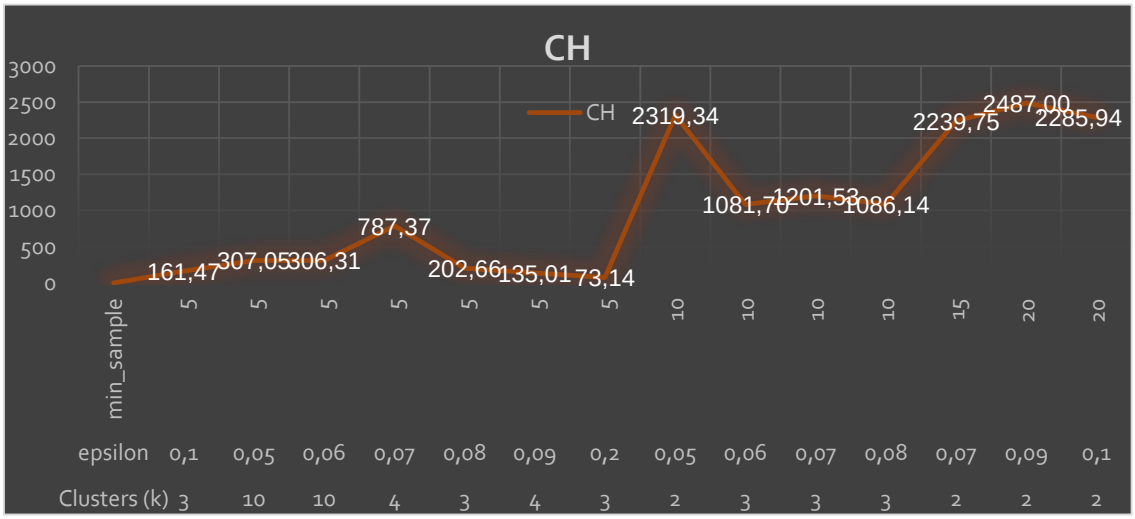


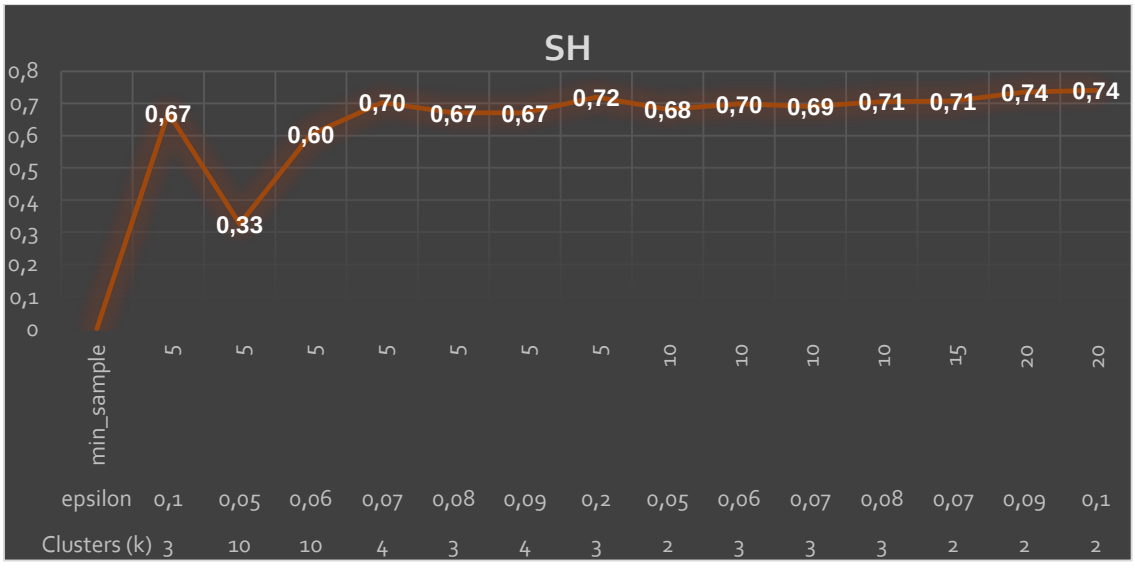


- Con Birch hemos obtenido peores resultados que con k-mean, aunque con 2 clusters y threshold=0.1 es un resultado muy similar al de 2 clusters de k-means.
- No sería preferible a k-means ni siquiera por tiempo de computo, ya que tarda ligeramente más.

DBSCAN

Algoritmo	Clusters (k)	epsilon	min_sample	Tiempo (segundos)	CH	SH
DBSCAN	3	0,1	5	0,08	161,471106	0,67063
DBSCAN	10	0,05	5	0,06	307,054825	0,3254
DBSCAN	10	0,06	5	0,06	306,309252	0,60353
DBSCAN	4	0,07	5	0,06	787,366055	0,70269
DBSCAN	3	0,08	5	0,07	202,657347	0,67063
DBSCAN	4	0,09	5	0,07	135,01305	0,67063
DBSCAN	3	0,2	5	0,06	73,1389403	0,7194
DBSCAN	2	0,05	10	0,05	2319,34161	0,68211
DBSCAN	3	0,06	10	0,09	1081,70043	0,69839
DBSCAN	3	0,07	10	0,06	1201,52821	0,69047
DBSCAN	3	0,08	10	0,07	1086,13647	0,70542
DBSCAN	2	0,07	15	0,06	2239,74907	0,70716
DBSCAN	2	0,09	20	0,08	2487,00224	0,7356
DBSCAN	2	0,1	20	0,86	2285,9417	0,74073

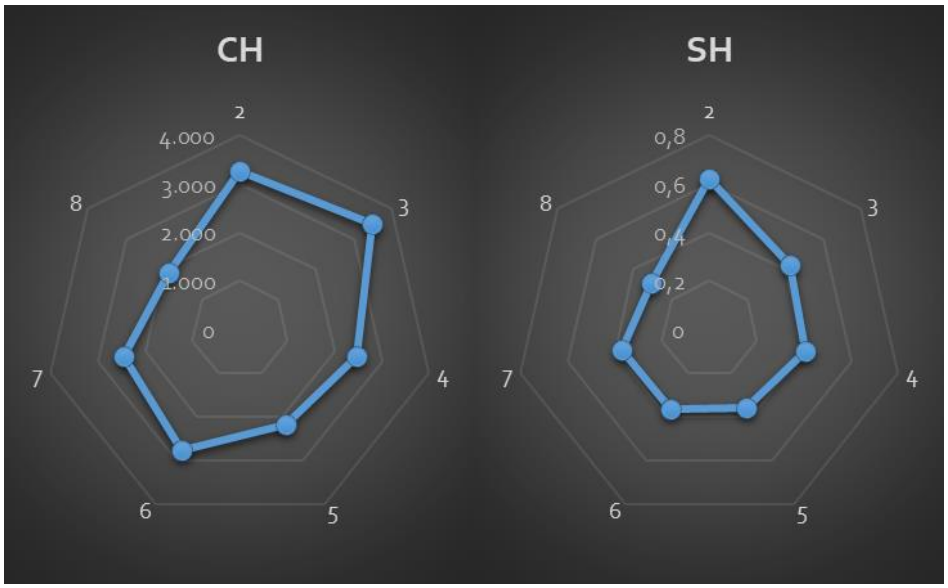




- Para este algoritmo he realizado distintas pruebas cambiando 2 de los parámetros que nos permite el algoritmo (épsilon y min_sample). Con el parámetro **épsilon** establecemos la distancia máxima entre 2 puntos para que se considere que está dentro del vecindario. Con el parámetro **min_sample** establecemos la cantidad de muestras necesarias en el vecindario para que ese punto se considere un centroide (incluido ese mismo punto).
- En la tabla hay un resumen de los resultados obtenidos, donde podemos observar que quizás los mejores parámetros para nuestro caso de estudio podría ser Épsilon=0.09 y min_sample=20, ya que en silohutte obtiene un rendimiento bastante aceptable (0.74073), el que más cerca de uno está, que nos indica que los clusters están bastante bien diferenciados y no apenas están superpuestos, mientras que en Calinski-Harabaz está entre los 3 primeros obteniendo un valor muy similar a los otros dos. Es un valor por debajo de los obtenidos en k-means, lo que nos indica que los datos estarán más dispersos.

SPECTRAL-CLUSTERING

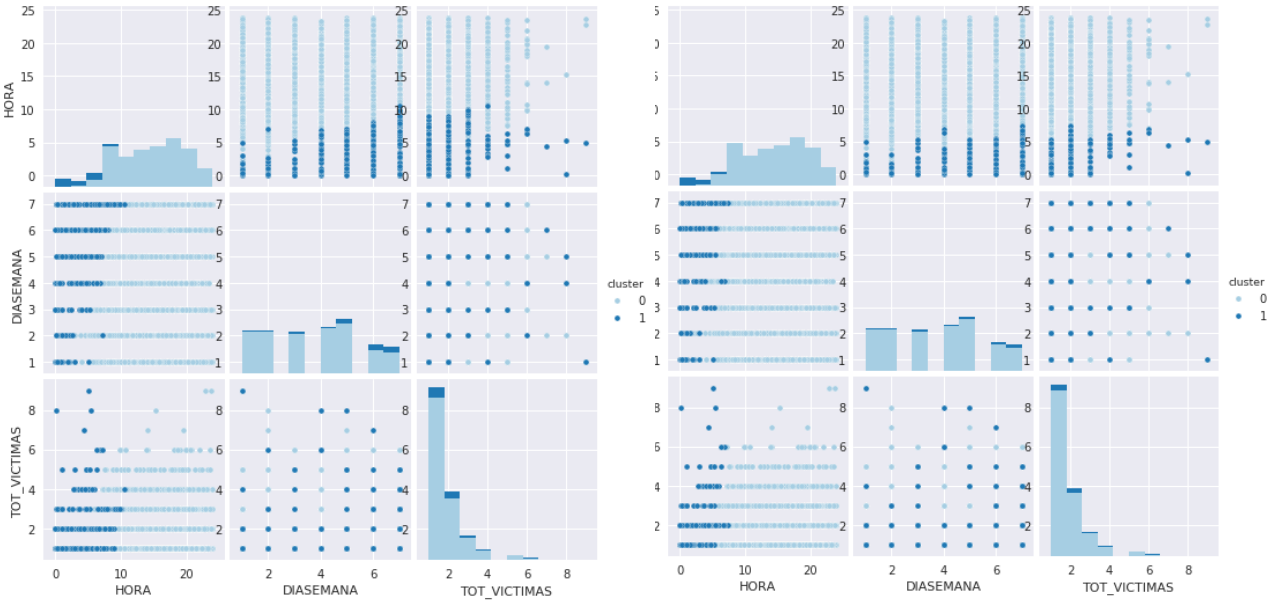
Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
SpectralClustering	2	1,12	3.255	0,61984
SpectralClustering	3	1,21	3478,52	0,42387
SpectralClustering	4	1,05	2471,52298	0,40671
SpectralClustering	5	1,09	2184,22685	0,3565
SpectralClustering	6	1,1	2766,54509	0,36531
SpectralClustering	7	1,19	2454,78479	0,37264
SpectralClustering	8	1,25	1872,99111	0,30968



- Este algoritmo tiene bastante posibles parámetros distintos, pero tras realizar algunas pruebas he podido comprobar que el rendimiento apenas cambia excepto cuando se cambia el número de clusters.
- El mejor caso es con 2 cluster, que obtiene un rendimiento en CH algo superior a la mayoría de algoritmos, exceptuando k-means, y un valor muy bajo en Silohuette, lo que nos sugiere que los datos están poco dispersos en los cluster pero que habrá bastante solapamiento entre ellos.

WARD

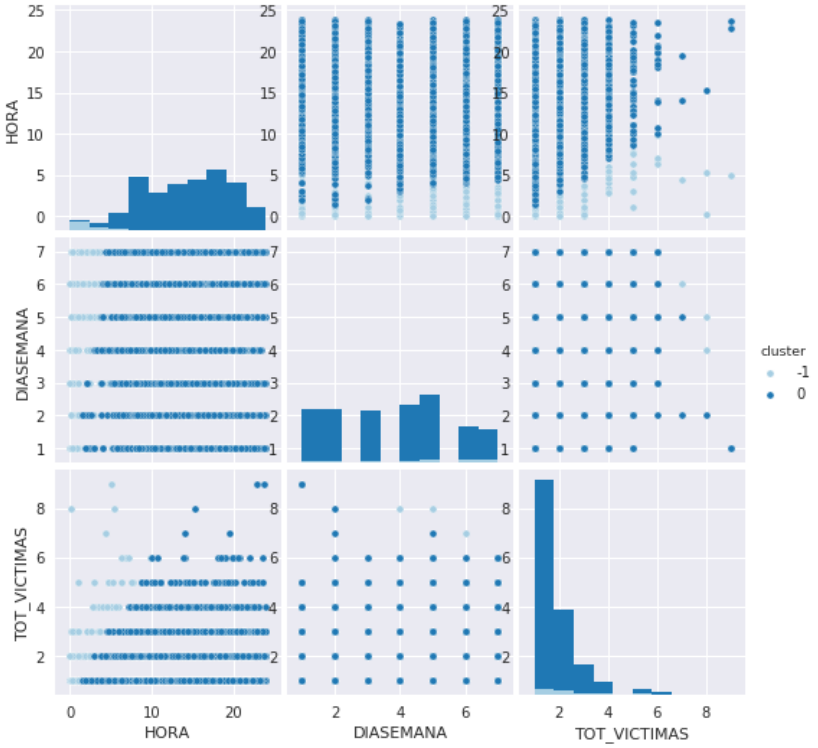
- Con este algoritmo he realizado 2 cambios:
    - **Número de cluster**: lo he establecido en 2, ya que es el número de cluster que mayor rendimiento ha dado en el resto de algoritmos.
    - El siguiente parámetro modificado es ***connectivity*** que hace que los elementos de un mismo cluster tengan que estar conectados con los vecinos más cercanos, lo que hace que estén mejor agrupados. En este caso hemos utilizado 10 vecinos más cercanos.
- Vamos a mostrar la diferencia entre no aplicar *connectivity y aplicarlo.*



- Podemos observar como los elementos de ambos clusters están más agrupados al usar conectiviti en la imagen de la derecha.

INTERPRETACIÓN DE LA SEGMENTACIÓN

- Dado que el algoritmo que mejores resultados ha dado ha sido DBSCAN con los parámetros eps=0.1, min_samples=20, vamos a mostrar su scatter matrix y así llegar a una conclusión sobre el caso de estudio analizado:



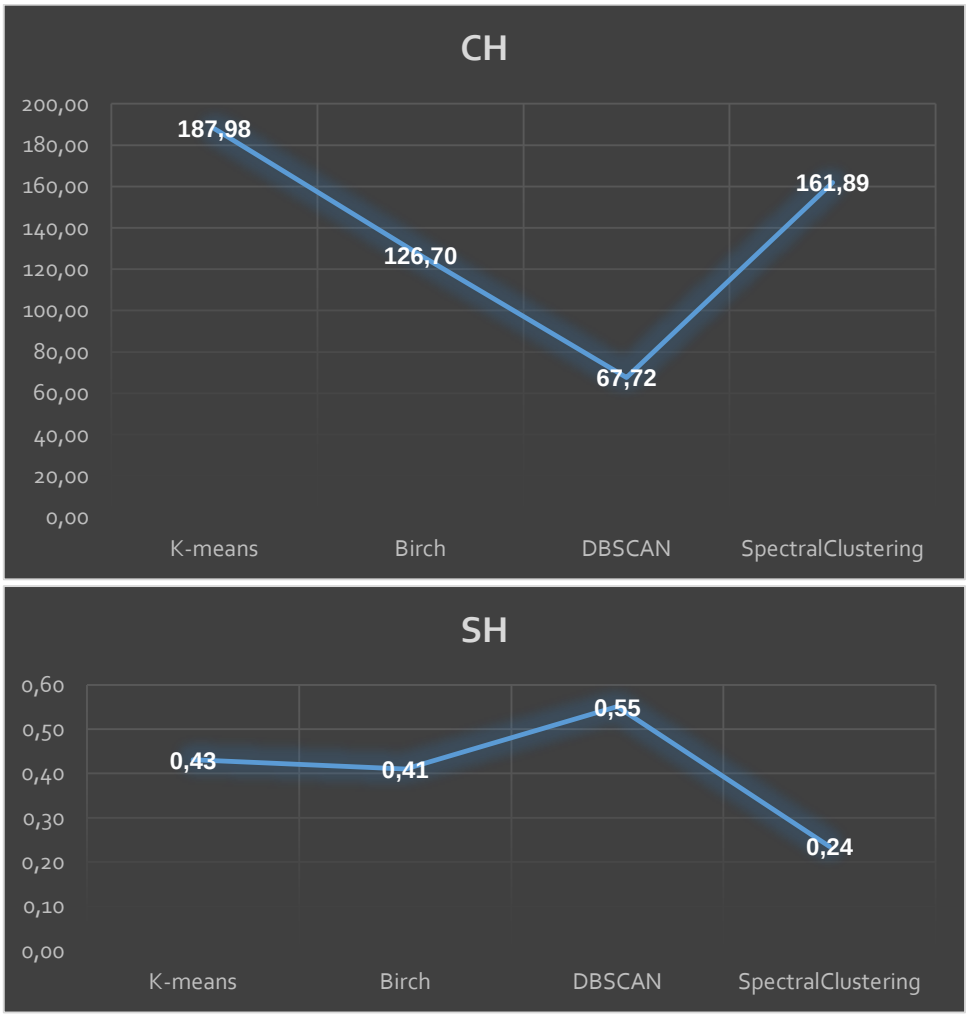
- Se puede observar que hay un cluster mayoritario (0) y una minoritario (-1), y que existe cierto solapamiento entre ellos, de ahí que el valor de Silhoutte fuera menor de 1.
- Si lo comparamos con los scatter obtenidos del algoritmo ward, vemos que son muy similares, de modo que podemos apoyarnos tambien en ellos para explicar el caso de estudio.
- Vemos como según avanza la semana, se producen más accidentes pertenecientes al cluster -1 y menos de los pertenecientes al cluster 0.
- Todos los accidentes del cluster -1 ocurren entre las 0-10 horas. El día 1 entre las 0-5 horas y va aumentando las horas hasta llegar al día 7, que se producen entre las 0-10 horas.
- Por último, en cuanto al número total de víctimas, la distribución es similar entre ambos cluster, siendo el número de vícitimas más habitual entre 0-4.

2.2. CASO DE ESTUDIO 2

- El segundo caso de estudio será sobre accidentes en los que ha habido atropello, concretamente en la provincia de Córdoba. Es un tipo de accidente que puede ser de especial importancia debido a que en este tipo de accidentes la/s persona/s atropelladas son un elemento muy vulnerable, por lo que sería interesante conocer que circunstancias son las que más afectan.
- Las variables utilizadas son: 'MES', 'DIASEMANA','HORA', 'TOT_MUERTOS'.
- Nº de casos: 169

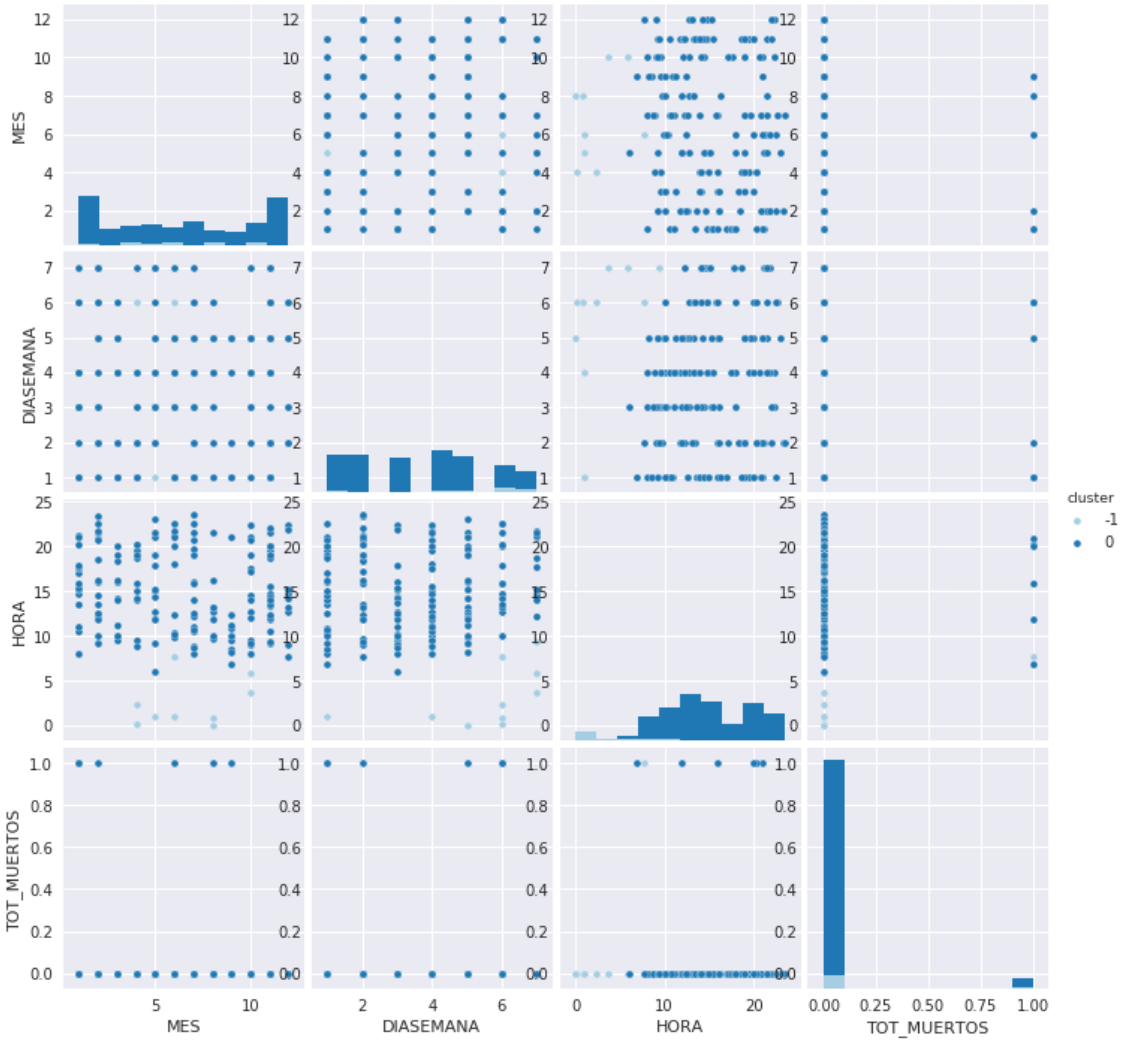
RESULTADOS PARÁMETROS POR DEFECTO

Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	4	0,05	187,98	0,43
Birch	4	0,01	126,70	0,41
DBSCAN	2	0,02	67,72	0,55
SpectralClustering	4	0,14	161,89	0,24
Ward	4	0,02	0,00	0,00



- Atendiendo a las métricas de rendimiento podemos observar como k-means con los parámetros por defecto obtiene el mejor resultado en CH, y el segundo mejor en SH.(es el más equilibrado). Por otro lado, SpectralClustering obtiene peores resultados en Calinski-Harabaz y muy malos en Silhoutte. DBSCAN es el que mejor rendimjiento da en SH, aunque un valor muy bajo en CH, por lo que podemos deducir que aunque los datos de sus clusters están menos superpuestos, están muy dispersos. BIRCH obtienen resultados muy bajos en ambas métricas.

- En este caso vamos a utilizar el scatter matrix de DBSCAN, para analizar sus clusters y así entender algo mejor el problema. Es uno de los que mejores resultados ha obtenido con los parámetros por defecto (ninguno en realidad tiene bueno resultados en general), y al usar 2 cluster puede ser un modelo más sencillo de entender.

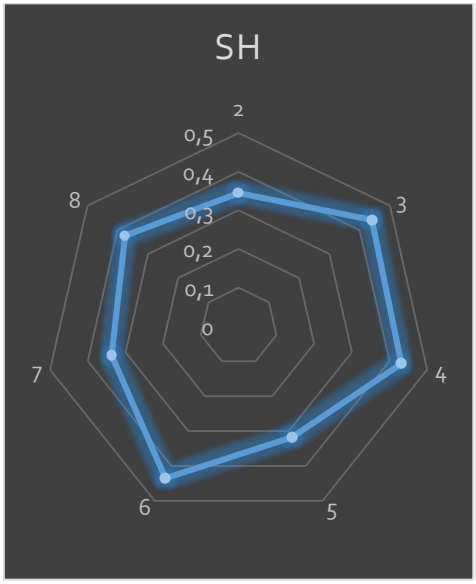


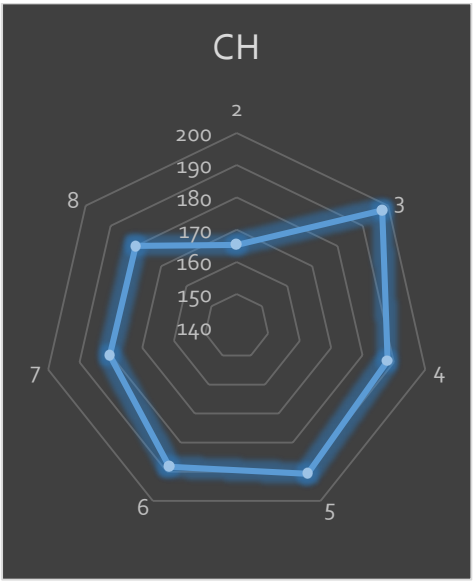
- El cluster (0) es muy predominante con respecto al cluster (-1).
- Cluster (0):
  - El cluster (0) aparece a partir de las 5 horas hasta las 24 horas.
  - Aparece en todos los días de la semana de forma más o menos regular, aunque algo reducido en el día 6 y 7, que además coincide con que son los días donde más presencia hay del cluster (-1).
  - La presencia del cluster (0) en los distintos meses del año es bastante regular, excepto en los meses 12 y 1 que aumentan.
  - En la mayor parte (90% aprox.) de los atropellos no hay víctimas mortales.
- Cluster (-1):
  - El cluster (-1) solamente aparece en las primeras horas del día (0-5 horas), los días 1, 4, 5, 6 y 7, sobre todo en el fín de semana.
  - Los meses en los que más aparece el cluster (-1) son los meses 2,6,8 y 10, aunque también ligeramente en el mes 2.
  - El total de muertos para los atropellos del cluster (-1) es casi siempre 0, excepto algún ejemplo aislado.

- A continuación, realizaremos distintas ejecuciones de los algoritmos, modificando sus parámetros para encontrar una mejor separación de los datos en clusters.

K-MEANS

Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	2	0,03	165	0,34487
K-means	3	0,07	197,697324	0,44072
K-means	4	0,03	187,984034	0,431
K-means	5	0,08	190,71951	0,31821
K-means	6	0,06	188,289861	0,43562
K-means	7	0,08	180,331191	0,33616
K-means	8	0,08	180,007732	0,37557

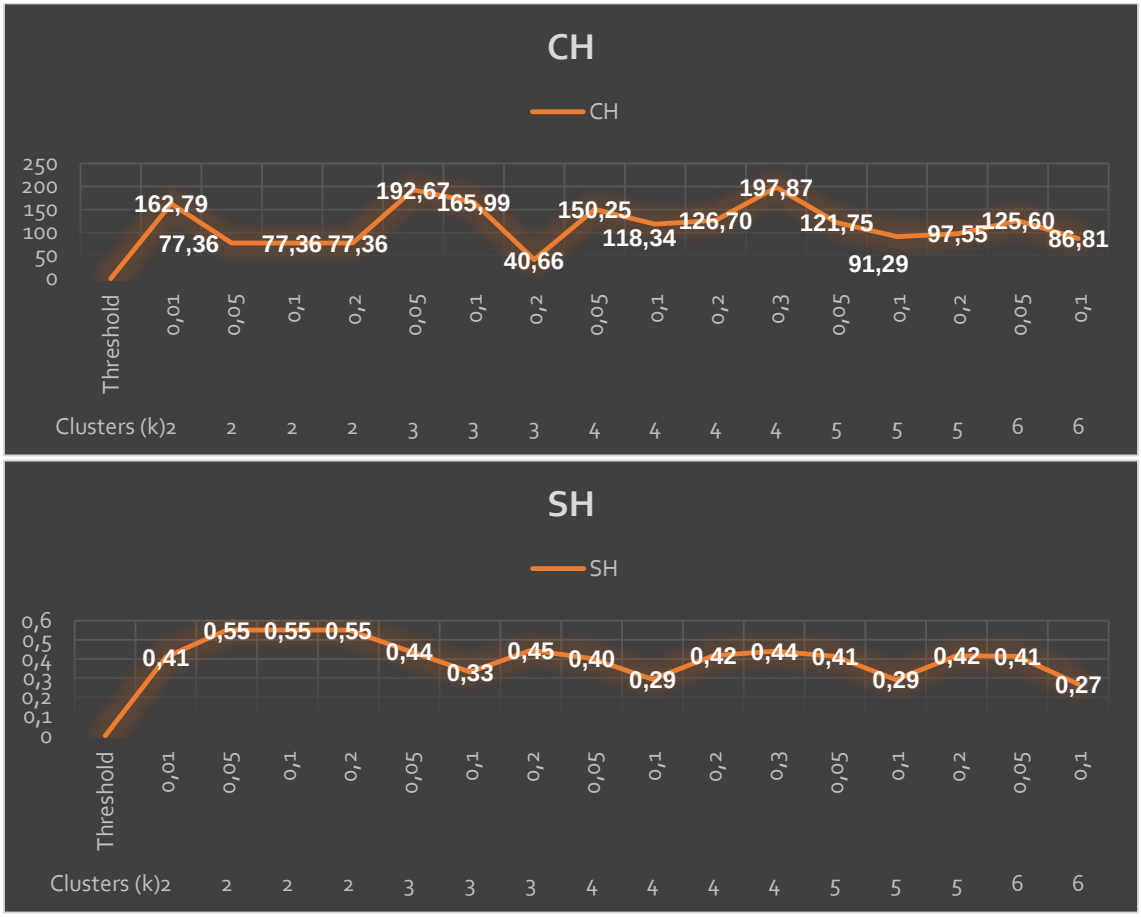




- Parece que el número de cluster que obtienen un mejor rendimiento teniendo en cuenta ambas métricas, es con 3 clusters, que es el más equilibrado entre la dispersión de los datos y la superposición de ejemplos.

BIRCH

Algoritmo	Clusters (k)	Threshold	Tiempo (segundos)	CH	SH
BIRCH	2	0,01	0,04	162,79	0,41
BIRCH	2	0,05	0,03	77,36	0,55
BIRCH	2	0,1	0,01	77,36	0,55
BIRCH	2	0,2	0,01	77,36	0,55
BIRCH	3	0,05	0,01	192,67	0,44
BIRCH	3	0,1	0,01	165,99	0,33
BIRCH	3	0,2	0,03	40,66	0,45
BIRCH	4	0,05	0,01	150,25	0,40
BIRCH	4	0,1	0,01	118,34	0,29
BIRCH	4	0,2	0,01	126,70	0,42
BIRCH	4	0,3	0,03	197,87	0,44
BIRCH	5	0,05	0,01	121,75	0,41
BIRCH	5	0,1	0,01	91,29	0,29
BIRCH	5	0,2	0,02	97,55	0,42
BIRCH	6	0,05	0,01	125,60	0,41
BIRCH	6	0,1	0,01	86,81	0,27



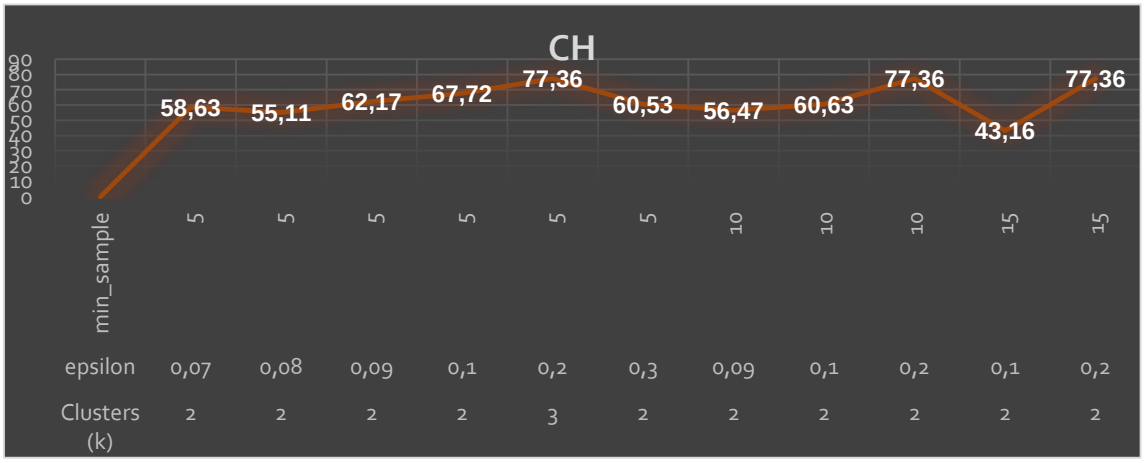
- Con Birch he realizado bastante pruebas, ya que iba modificando las métricas de rendimiento de forma sustancial con los cambios. En el resumen presentado en la

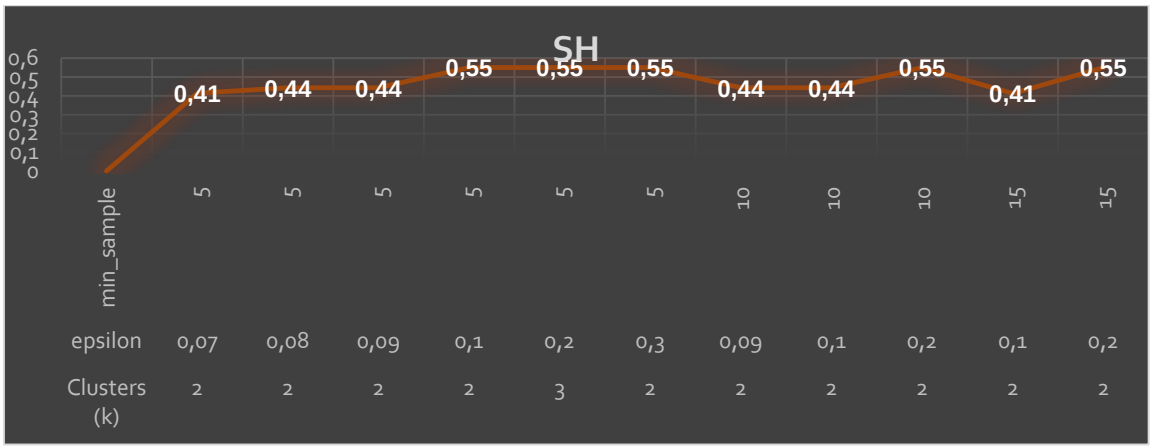
tabla podríamos considerar varios casos como el mejor, ya que unos tiene mejores valores en CH y otros en SH.

- Como mejor valor para SH (con 0.55), tenemos los casos con 2 clusters y 0.05,0.1 y 0.2 threshold, que obtiene 77.36 en CH. Esos casos tendrán menor superposición que los otros, aunque la dispersión de los datos será mayor.
- El caso que mejor rendimiento obtiene en CH es con cluter=4 y threshold=0.3 y que empeora ligeramente en SH respecto a los mejores en SH. Este úlitmo caso parece ser más equilibrado.

DBSCAN

Algoritmo	Clusters (k)	epsilon	min_sample	Tiempo (segundos)	CH	SH
DBSCAN	2	0,07	5	0,001	58,63	0,41
DBSCAN	2	0,08	5	0,001	55,11	0,44
DBSCAN	2	0,09	5	0,001	62,17	0,44
DBSCAN	2	0,1	5	0,001	67,72	0,55
DBSCAN	3	0,2	5	0,001	77,36	0,55
DBSCAN	2	0,3	5	0,001	60,53	0,55
DBSCAN	2	0,09	10	0,001	56,47	0,44
DBSCAN	2	0,1	10	0,001	60,63	0,44
DBSCAN	2	0,2	10	0,001	77,36	0,55
DBSCAN	2	0,1	15	0,001	43,16	0,41
DBSCAN	2	0,2	15	0,001	77,36	0,55

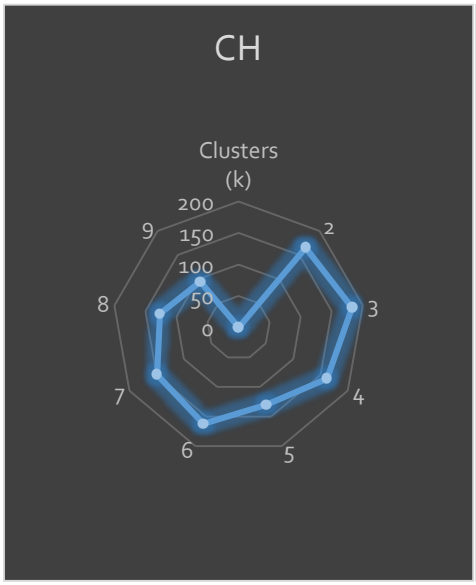


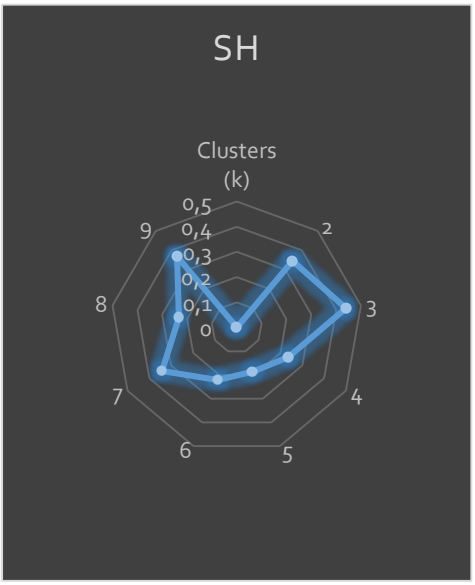


- Al igual que en el caso de estudio anterior, para este algoritmo he realizado distintas pruebas cambiando 2 de los parámetros que nos permite el algoritmo (épsilon y min_sample).
- En este algoritmo podemos observar como hay unos valores para las métricas de rendimiento que se repiten en varios casos, y que además coinciden con alguno de los casos del algoritmo BIRCH. Quizás la división en 2 clústeres sea la mejor opción

SPECTRAL-CLUSTERING

Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
SpectralClustering	2	0,13	165	0,34487
SpectralClustering	3	0,13	182,666357	0,44
SpectralClustering	4	0,16	161,891331	0,23539
SpectralClustering	5	0,14	130,150543	0,18604
SpectralClustering	6	0,21	162,359127	0,22065
SpectralClustering	7	0,21	149,403453	0,34277
SpectralClustering	8	0,59	126,645684	0,23273
SpectralClustering	9	0,92	94,6963007	0,36757

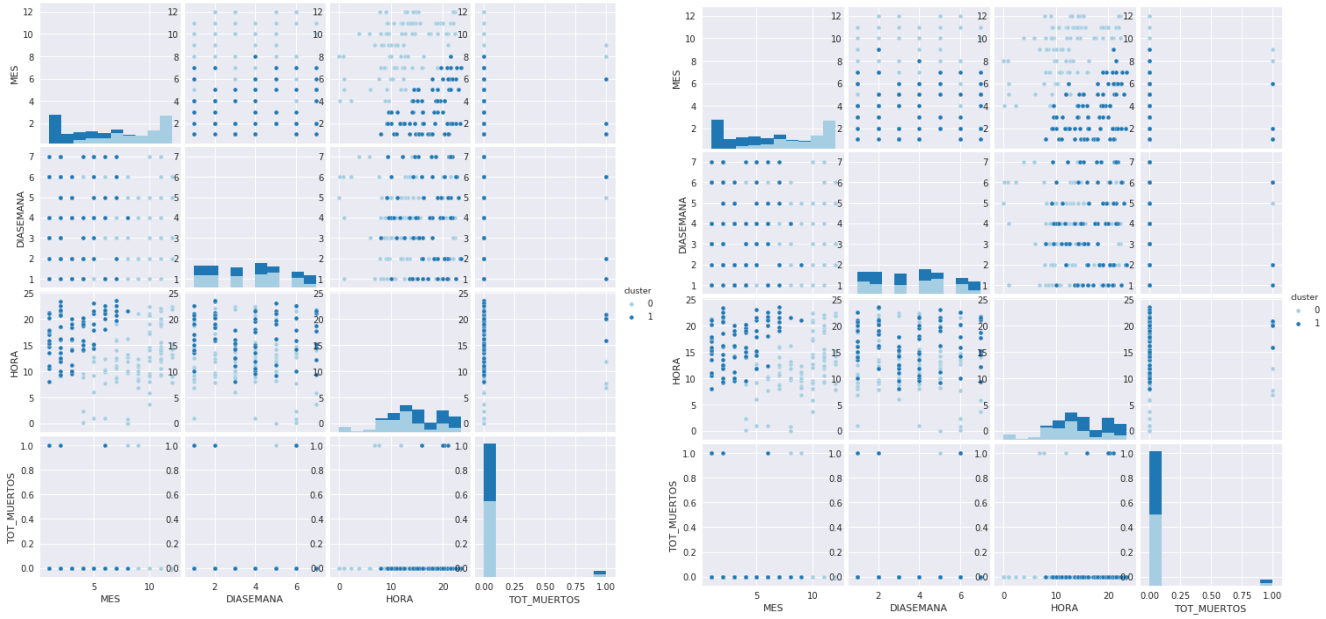




- Como en el caso de estudio anterior, he realizado pruebas cambiado el número de clusters.
- El mejor caso es con 3 clusters,que obtiene un rendimiento por debajo de los otros algoritmos, por lo que no lo tendremos en cuenta para la interpretación final.

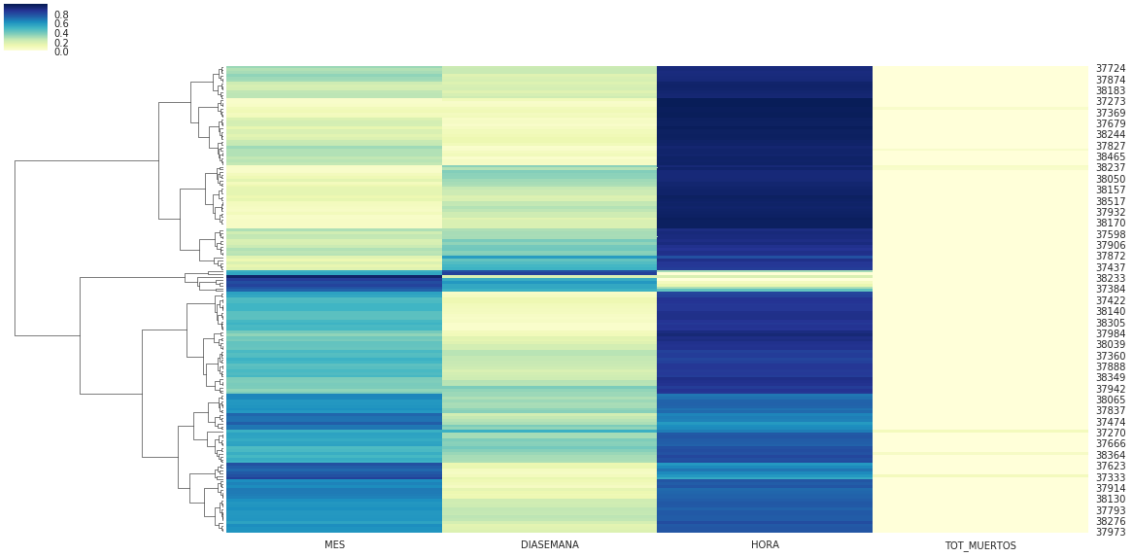
WARD

- Con este algoritmo he realizado 2 cambios:
    - **Número de cluster**: lo he establecido en 2, ya que es el número de cluster que mayor rendimiento ha dado en el resto de algoritmos.
    - El siguiente parámetro modificado es ***connectivity*** que hace que los elementos de un mismo cluster tengan que estar conectados con los vecinos más cercanos. En este caso hemos utilizado 10 vecinos más cercanos.



- En este caso podemos observar que no ha surtido mucho efecto *connectivity*, ya que practicamente son iguales.

- Para este caso de estudio con pocos ejemplos, podemos mostrar un dendograma junto al heatmaps para entender mejor el caso de estudio.



INTERPRETACIÓN DE LA SEGMENTACIÓN

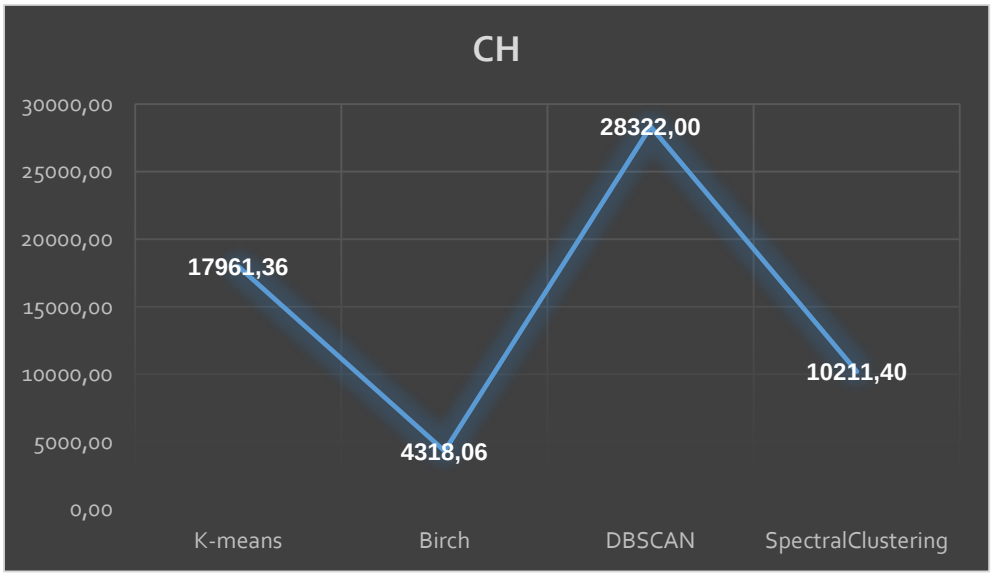
- Para dar una interpretación final a la segmentación de este caso de estudio vamos a apoyarnos en los scatter matrix obtenidos en Ward y en el dendograma junto al heatmaps.
- Ahora vemos como los 2 cluster estan bastante equilibrados en cuanto al número de ejemplos que tiene cada uno.
- Observando tanto el scatter matrix como el dendograma, observamos que los clusters están divididos entre la primera mitad del año (cluster (1)) y la segunda mitad del año (cluster (0)).
- En cuanto a los días de la semana, vemos como estan equilibrados excepto los últimos días de la semana, que como se observa en el dendograma, están más presentes en el cluster (1).
- En las horas si que se observa una tendencia, que en las primeras horas del día está más presente el cluster (0) y que conforme avanza el día, está más presente el cluster (1).
- Por último, el número total de muertos, se observa en el dendograma como prácticamente es uniforme, excepto algunos casos que vemos con un color más oscuro.

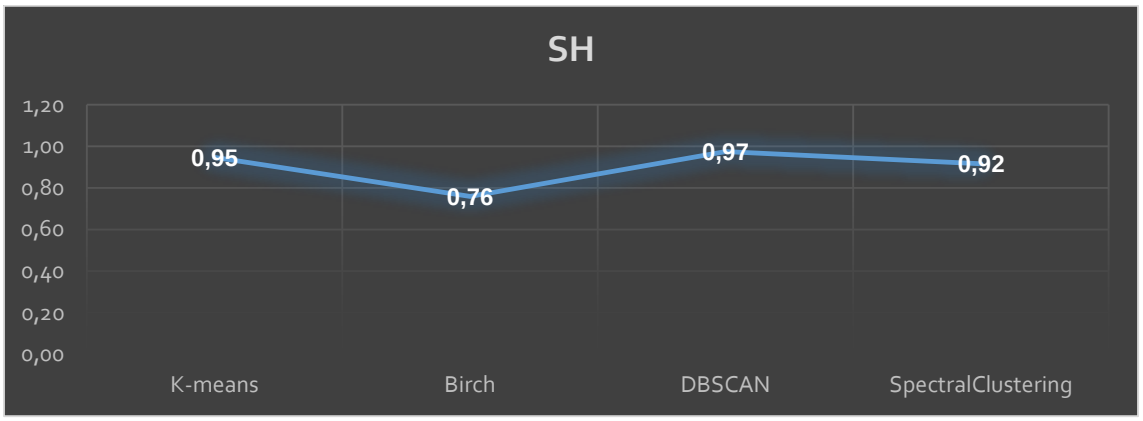
2.3. CASO DE ESTUDIO 3

- El tercer caso de estudio será sobre "Vuelco en la calzada". He elegido este caso de estudio porque me parece interesante poder conocer que aspecto influyen en este tipo de accidentes, que quizás, no sean de los más comunes, pero sí que pueden ser accidentes bastante graves.
- Las variables utilizadas son: 'TOT_VICTIMAS', 'TIPO_VIA', 'TOT_HERIDOS_GRAVES', 'TOT_HERIDOS_LEVES', 'TRAZADO_NO_INTERSEC'
- Nº de casos: 3334

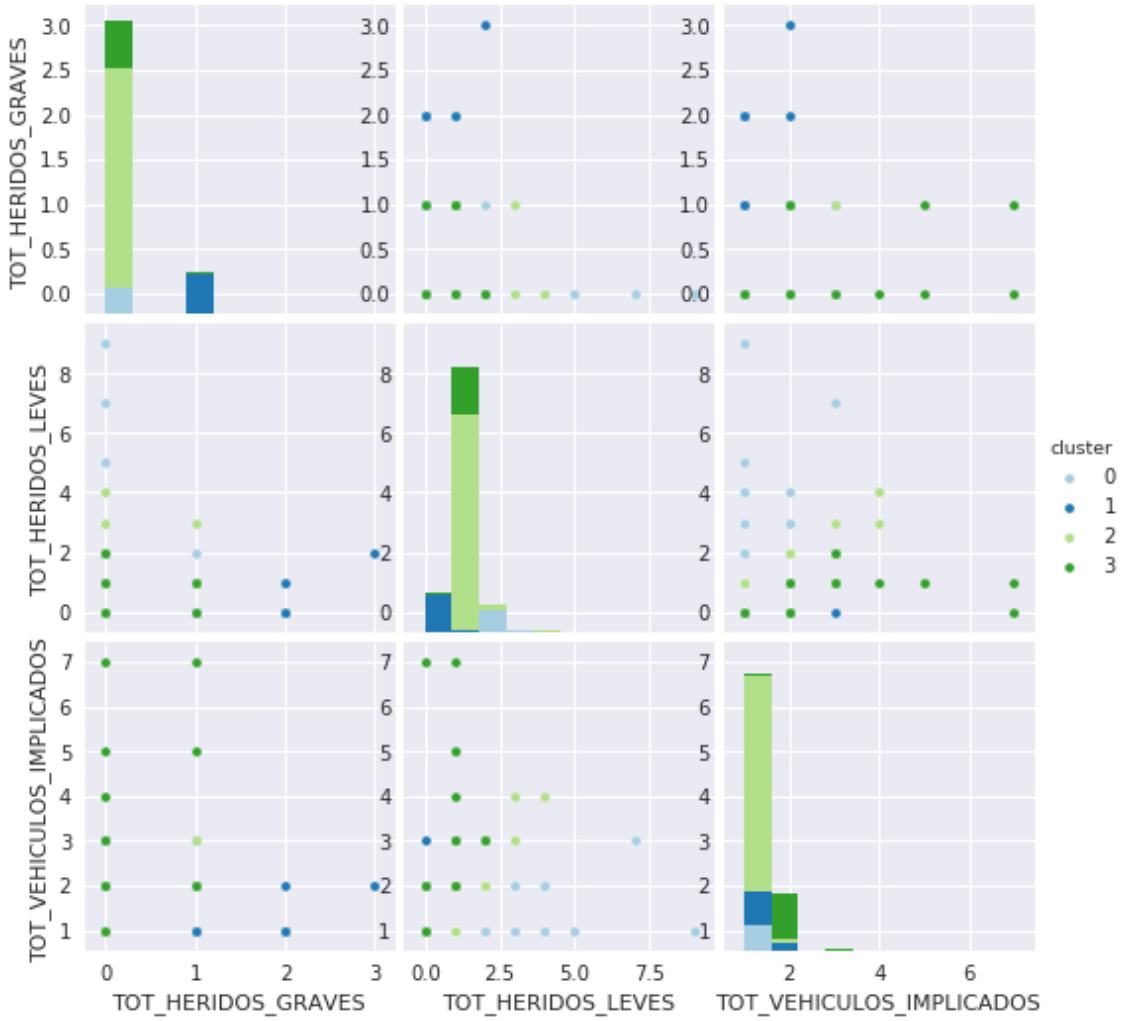
RESULTADOS PARÁMETROS POR DEFECTO

Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	4	0,04	17961,36	0,95
Birch	4	0,37	4318,06	0,76
DBSCAN	12	0,26	28322,00	0,97
SpectralClustering	4	43,52	10211,40	0,92
Ward	4	0,59	0,00	0,00





- Si observamos las métricas de rendimiento, DBSCAN con los parámetros por defecto obtiene el mejor resultado tanto en CH como en SH, seguido por K-means, SpectralClustering y en último lugar Birch. Sin embargo, los clusters de DBSCAN son elevados, siendo un modelo más simple K-means al tener menos clusters ,por lo que es más fácil de entender.
- Con respecto a los valores obtenidos en los dos casos de estudio anteriores, estos son muy óptimos y posiblemente se deba a que los ejemplos de este estudio son más fácilmente separables.
- En este caso vamos a utilizar el scatter matrix de K-means para analizar sus clusters, ya que es el que mejor resultado ha obtenido con los parámetros por defecto.



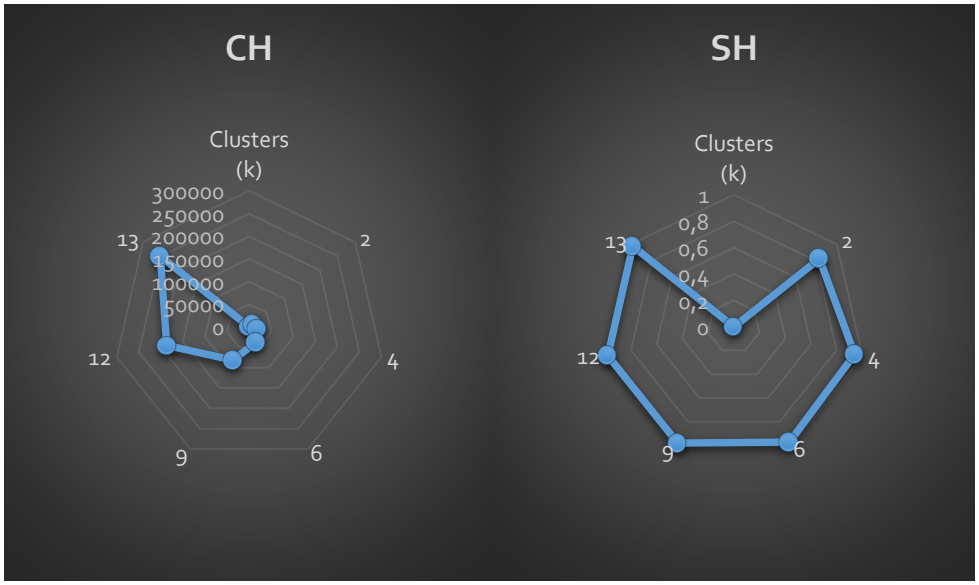
- El cluster (2) es el más predominante con respecto a los demás.
- La mayoría de los casos, la cantidad de vehículos implicados es 1 y en algunos casos 2. El número de casos en los que hay 3 o más vehículos implicados es despreciable.
- Con respecto a heridos leves, en la mayoría de los casos, el número total de heridos leves es 1, además, pertenece en su mayoría al cluster (2), aunque también hay casos del cluster (3).
    - La mayoría de los accidentes en los que no hay heridos leves pertenecen al cluster (1) y además coincide con los que hay un herido grave.
    - La mayor parte de los accidentes en los que hay 2 o más heridos leves, pertenecen al cluster (0).
- En los accidentes en los que hay al menos un herido grave, pertenecen al cluster (1).
- Se puede establecer una correspondencia entre un accidente y el cluster al que pertenece casi con total certeza:

Cluster	TOT_VEHÍCULOS_IMPLICADOS	TOT_HERIDOS_LEVES	TOT_HERIDOS_GRAVES
0	1	>=2	0
1	1	0	>=1
2	1	1	0
3	>=2	1	0

- A continuación, realizaremos distintas ejecuciones de los algoritmos, modificando sus parámetros para encontrar una mejor separación de los datos en clusters.

K-MEANS

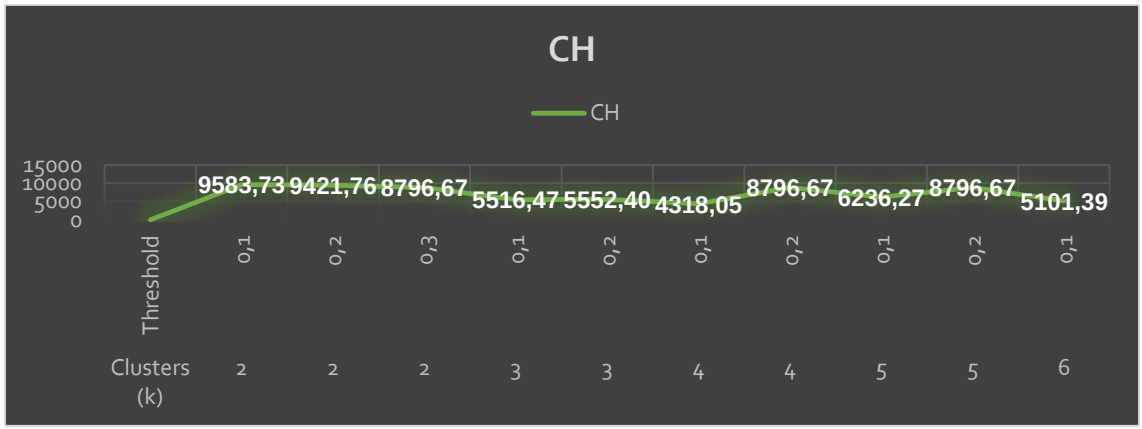
Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
K-means	2	0,03	9700,19	0,83
K-means	4	0,03	17961,36	0,94
K-means	6	0,06	37509,45	0,97
K-means	9	0,03	82052,76	0,98
K-means	12	0,03	186709,04	0,98
K-means	13	0,03	250948,09	0,98

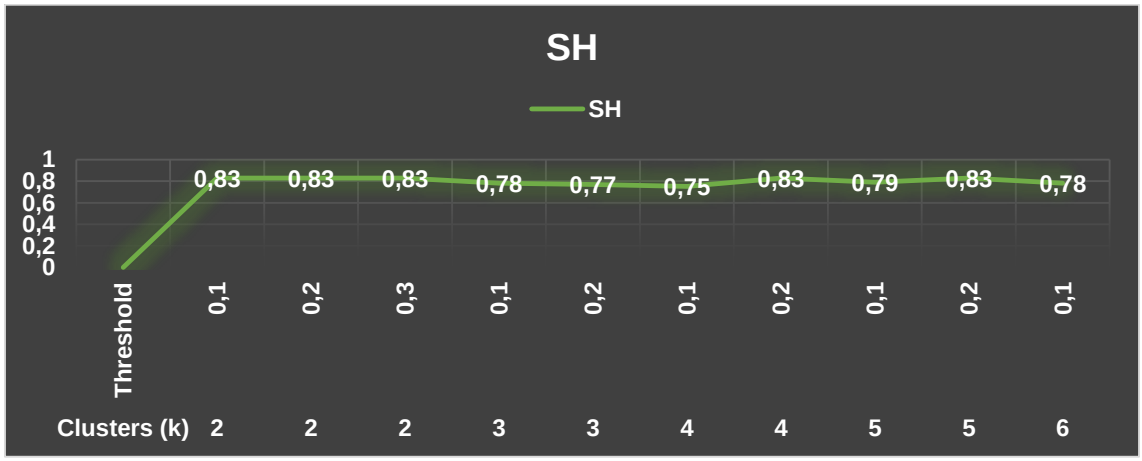


- Con la experimentación de K-means hemos obtenido unos resultados muy buenos tanto en SH como en CH, de hecho, si seguía aumentando el número de clusters CH seguía aumentando, aunque creaba clusters con menos de 3 ejemplos. Por tanto, he decidido dejar como mejor experimentación 13 clusters.

BIRCH

Algoritmo	Clusters (k)	Threshold	Tiempo (segundos)	CH	SH
BIRCH	2	0,1	0,41	9583,73	0,83
BIRCH	2	0,2	0,07	9421,76	0,83
BIRCH	2	0,3	0,21	8796,67	0,83
BIRCH	3	0,1	0,21	5516,47	0,78
BIRCH	3	0,2	0,18	5552,40	0,77
BIRCH	4	0,1	0,22	4318,05	0,75
BIRCH	4	0,2	0,21	8796,67	0,83
BIRCH	5	0,1	0,20	6236,27	0,79
BIRCH	5	0,2	0,19	8796,67	0,83
BIRCH	6	0,1	0,21	5101,39	0,78

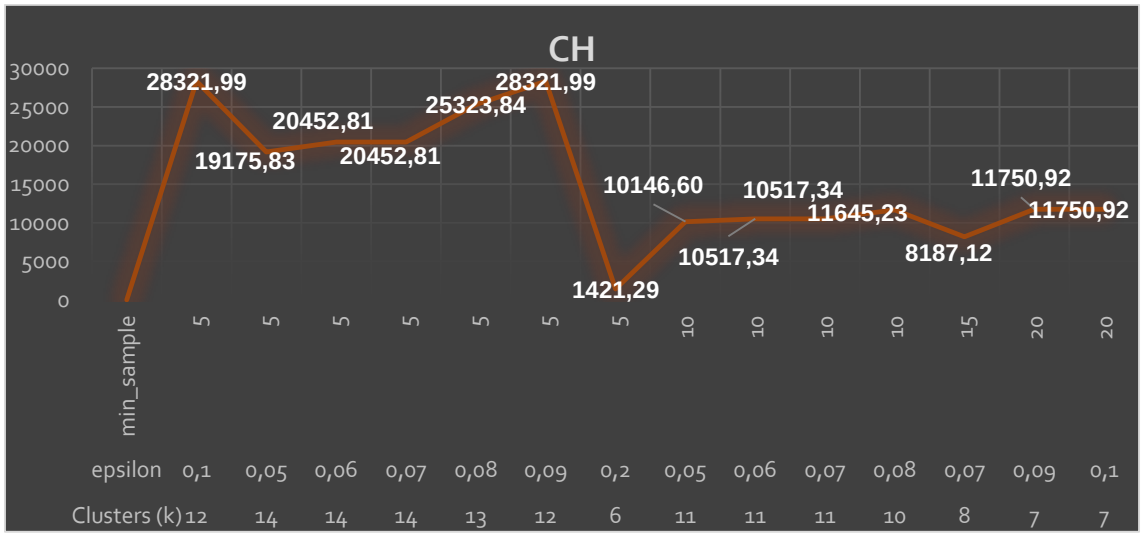


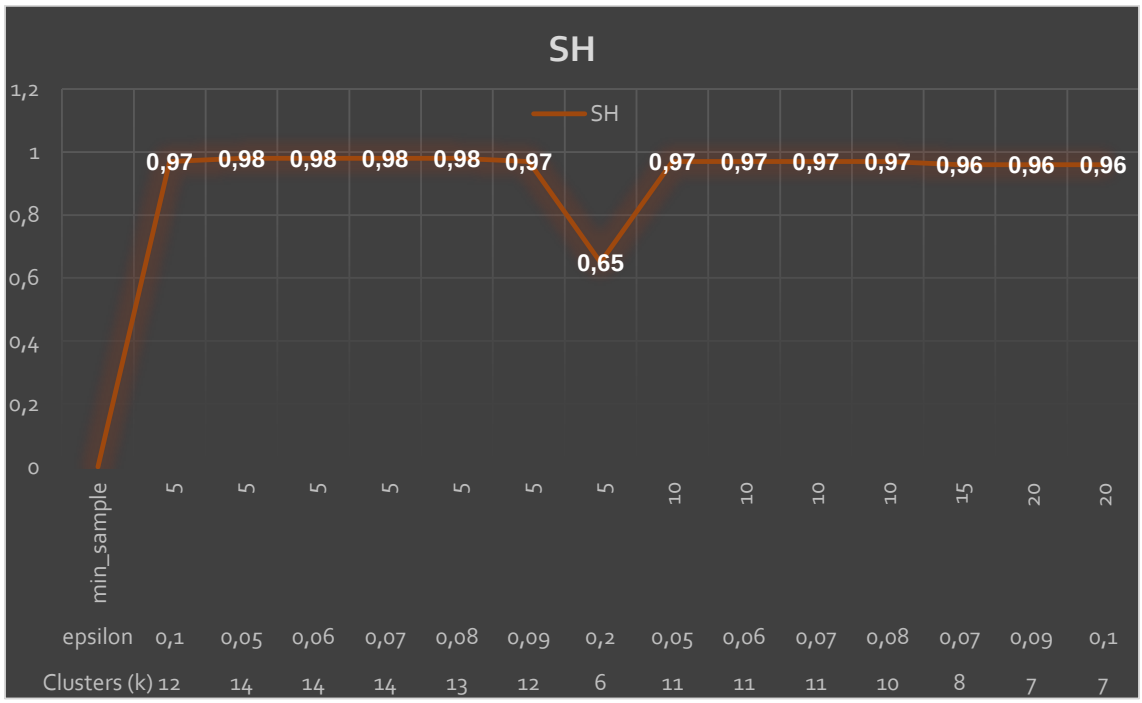


- Con Birch hemos obtenido unos resultados que aun no siendo malos están bastante por debajo en rendimiento a los obtenidos con K-means, por lo que no los tendremos en cuenta para el estudio del caso.

DBSCAN

Algoritmo	Clusters (k)	epsilon	min_sample	Tiempo (segundos)	CH	SH
DBSCAN	12	0,1	5	0,20	28321,99	0,97
DBSCAN	14	0,05	5	0,18	19175,83	0,98
DBSCAN	14	0,06	5	0,22	20452,81	0,98
DBSCAN	14	0,07	5	0,22	20452,81	0,98
DBSCAN	13	0,08	5	0,27	25323,84	0,98
DBSCAN	12	0,09	5	0,24	28321,99	0,97
DBSCAN	6	0,2	5	0,27	1421,29	0,65
DBSCAN	11	0,05	10	0,19	10146,60	0,97
DBSCAN	11	0,06	10	0,19	10517,34	0,97
DBSCAN	11	0,07	10	0,19	10517,34	0,97
DBSCAN	10	0,08	10	0,19	11645,23	0,97
DBSCAN	8	0,07	15	0,20	8187,12	0,96
DBSCAN	7	0,09	20	0,27	11750,92	0,96
DBSCAN	7	0,1	20	0,22	11750,92	0,96

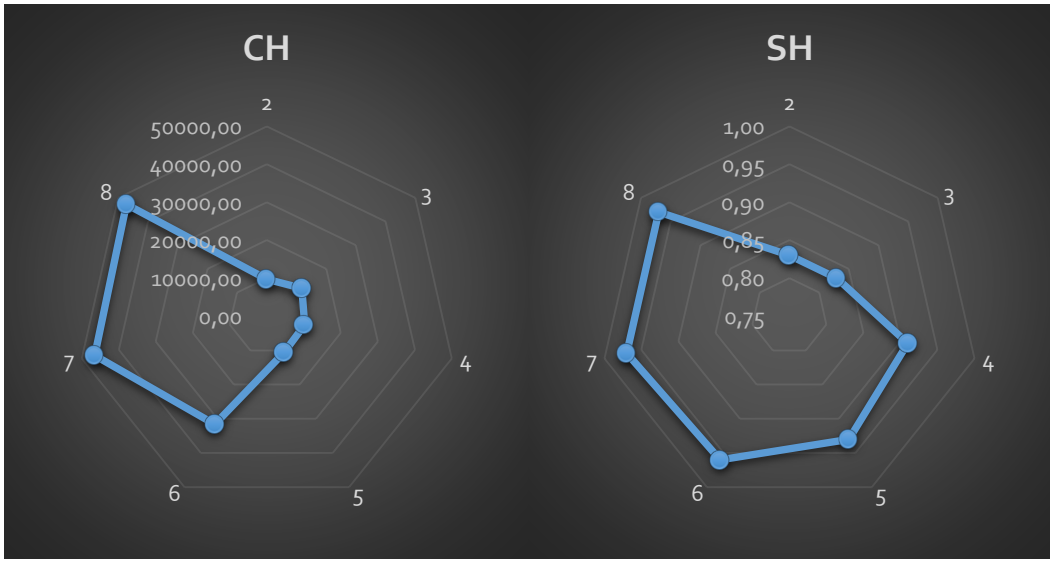




- Con DBSCAN hemos obtenidos muy buenos resultados, al igual que con k-means.
- Con la métrica de rendimiento SH obtenemos exactamente el mismo resultado (0,98), que nos indica que prácticamente no hay solapamiento entre clusters.
- Los resultados en la métrica CH están algo por debajo de los obtenidos en k-means.
- El número de clusters que nos proporciona DBSCAN en estos mejores casos es 12 o 13 que coincide con los mejores casos de k-means.

SPECTRAL-CLUSTERING

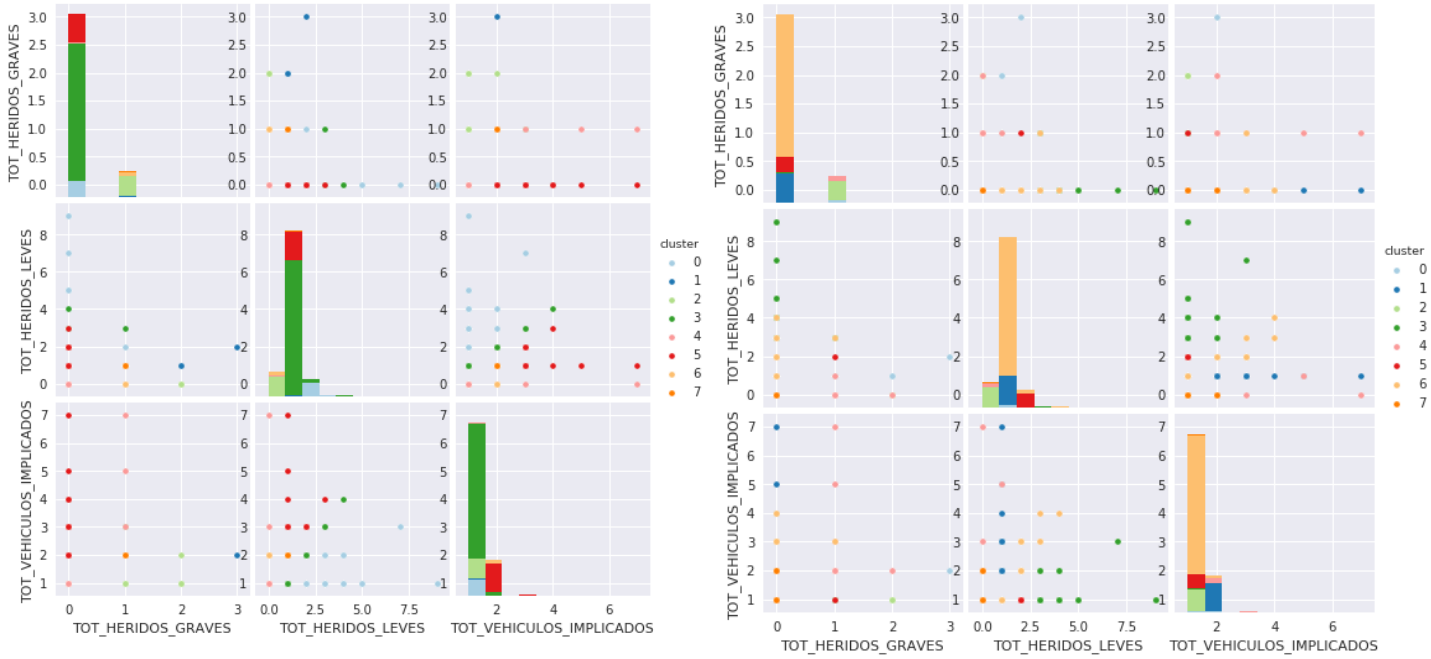
Algoritmo	Clusters (k)	Tiempo (segundos)	CH	SH
SpectralClustering	2	33,01	9700,19	0,83
SpectralClustering	3	40,93	11814,42	0,83
SpectralClustering	4	40,66	10211,39	0,91
SpectralClustering	5	40,05	10572,17	0,93
SpectralClustering	6	39,65	31580,31	0,96
SpectralClustering	7	13,03	46499,52	0,97
SpectralClustering	8	109,22	47275,52	0,97



- Con 8 clusters obtiene resultados casi tan buenos como k-means y DBSCAN.
- Al ser un modelo más simple y tener menos clusters, se puede tener en cuenta.

WARD

- Al igual que los otros dos casos de estudio, voy a obtener un modelo con el número de clusters igual a los mejores resultados obtenidos. En este caso, haré un modelo con 8 clusters y no con 12 o 13 ya que nos permitirá entender el problema de forma más sencilla.
- También usaremos el parámetro *connectivity* como en los otros dos casos de estudio y mostraremos ambos scatter matrix para comparar.



- Como en los otros dos casos observamos que al utilizar el parámetro *connectivity* conseguimos que los elementos de cada cluster estén más agrupados.

INTERPRETACIÓN DE LA SEGMENTACIÓN

Cluster	TOT_VEHÍCULOS_IMPLICADOS	TOT_HERIDOS_LEVES	TOT_HERIDOS_GRAVES
0	1 ó 2	1	1
1	2	1	0
2	1	0	1
3	1	3	0
4	2 ó 3	0	1
5	1	2	0
6	1	1	0
7	1	0	0

- Hay un cluster mayoritario frente al resto (cluster 6), que son los "vuelcos en la calzada" en el que el total de vehículos implicados es 1, hay 1 herido leve y ningún herido grave.
- El siguiente cluster de mayor tamaño es el (1) que difiere del cluster (6) en que el total de vehículos implicados es 2.
- El cluster que sigue en tamaño es el cluster (2) en el que el total de vehículos implicados es 1, no hay heridos leves y hay 1 grave.
- El cluster (4) es igual al cluster (2) excepto en que el total de vehículos implicados es 2 ó 3.
- El cluster (5) se diferencia del cluster (6) en que el total de heridos leves es 2.
- Los clusters (0,3 y 7) tienen una cantidad de elementos muy baja. El cluster (7) son los "vuelcos en la calzada" con un vehículo implicado en los que no hay heridos. En el cluster (0) hay 1 ó 2 vehículos implicados, con 1 herido leve y 1 herido grave. Con respecto al cluster (3), hay 1 vehículo implicado y es el que mayor número de heridos leves tiene, siendo estos 3 heridos y ningún herido grave.

3. CONTENIDO ADICIONAL

4. BIBLIOGRAFÍA

1. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html
2. http://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabaz_score.html
3. https://stats.stackexchange.com/questions/52838/what-is-an-acceptable-value-of-the-calinski-harabasz-ch-criterion
4. https://es.wikipedia.org/wiki/K-means
5. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
6. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html
7. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html