Inteligencia de Negocio (2017-2018)

Grado en Ingeniería Informática

Universidad de Granada

Segmentación para Análisis Empresarial

UNIVERSIDAD DE GRANADA

Iván Rodríguez Millán

ivanrodmil@gmail.com

Índice

1 Introducción **10**

2 Caso de estudio 1: Accidentes de tráfico ocurridos en las comunidades autónomas de Madrid y Andalucía. **12**

2.1 Caso de Estudio: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid. 12

2.1.1 Resultados algoritmo K-means, caso de estudio de la comunidad de Madrid. 14

2.1.2 Resultados algoritmo MiniBatchKmeans, caso de estudio de la comunidad de Madrid. 19

2.1.3 Resultados algoritmo Birch, caso de estudio de la comunidad de Madrid. 22

2.1.4 Resultados algoritmo Mean Shift, caso de estudio de la comunidad de Madrid. 25

2.1.5 Resultados algoritmo Spectral, caso de estudio de la comunidad de Madrid. 27

2.1.6 Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid. 30

2.2 Caso de Estudio: Accidentes de tráfico ocurridos en la comunidad autónoma de Andalucía. 31

2.2.1 Resultados algoritmo K-means, caso de estudio de la comunidad de Andalucía. 32

2.2.2 Resultados algoritmo MiniBatchKmeans, caso de estudio de la comunidad de Andalucía. 35

2.2.3 Resultados algoritmo Birch, caso de estudio de la comunidad de Andalucía. 38

2.2.4 Resultados algoritmo Mean Shift, caso de estudio de la comunidad de Andalucía. 41

2.2.5 Resultados algoritmo Spectral, caso de estudio de la comunidad de Andalucía. 43

2.2.6 Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Andalucía. 45

2.3 Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid y Andalucía. 46
2.4 Modificación de los parámtros para algunos algoritmos referentes al caso de estudio de la comunidad de Madrid. 48
2.5 Modificación de los parámtros para algunos algoritmos referentes al caso de estudio de la comunidad de Andalucía. 50

3 Caso de estudio 2: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos y atropellos. 52

3.1 Caso de Estudio: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos. 52
3.1.1 Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 54
3.1.2 Resultados algoritmo MiniBatchKmeans, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 57
3.1.3 Resultados algoritmo Ward, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 60
3.1.4 Resultados algoritmo Mean Shift, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 62
3.1.5 Interpretación de la segmentación: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos 64
3.2 Caso de Estudio: Accidentes de tráfico en zonas urbanas y vías urbanas con atropellos. 65
3.2.1 Resultados algoritmo K-means, caso de estudio con atropellos en zonas y vías urbanas. 66
3.2.2 Resultados algoritmo DBSCAN, caso de estudio con atropellos en zonas y vías urbanas. 68
3.2.3 Resultados algoritmo Ward, caso de estudio con atropellos en zonas y vías urbanas. 72
3.2.4 Resultados algoritmo Mean Shift, caso de estudio con atropellos en zonas y vías urbanas. 74
3.2.5 Interpretación de la segmentación: Accidentes de tráfico en zonas urbanas y vías urbanas con atropellos 76

4 Caso de estudio 3: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave y con vurva fuerte. 77

4.1 Caso de Estudio: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave. 77

4.1.1 Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en trazado con curva suave. 79

4.1.2 Resultados algoritmo DBSCAN, caso de estudio de colisiones entre vehículos en trazado con curva suave. 81

4.1.3 Resultados algoritmo Spectral, caso de estudio de colisiones entre vehículos en trazado con curva suave. 83

4.1.4 Interpretación de la segmentación: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave. 86

4.2 Caso de Estudio: Accidentes de tráfico con colisiones entre vehículos en trazado con curva fuerte. 87

4.2.1 Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en trazado con curva fuerte. 88

4.2.2 Resultados algoritmo DBSCAN, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 90

4.2.3 Resultados algoritmo Spectral, caso de estudio de colisiones entre vehículos en zonas y vías urbanas. 93

4.2.4 Interpretación de la segmentación: Accidentes de tráfico con colisiones entre vehículos en trazado con curva fuerte. 96

4.3 Modificación de los parámtros para algunos algoritmos referentes al caso de estudio con colisiones entre vehículos en trazado con curva suave. . . . 98

4.4 Modificación de los parámtros para algunos algoritmos referentes al caso de estudio con colisiones entre vehículos en trazado con curva fuerte. . . . 99

5 Contenido adicional 101

Índice de figuras

2.1. Scatter Matrix usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid. 16

2.2. Heatmap usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid. 17
2.3. Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid. 20
2.4. Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid. 21
2.5. Scatter Matrix usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid. 23
2.6. Heatmap usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid. 24
2.7. Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 1 en la comunidad autónoma de Madrid. 26
2.8. Scatter Matrix usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid. 28
2.9. Heatmap usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid. 29
2.10. Scatter Matrix usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía. 33
2.11. Heatmap usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía. 34
2.12. Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía. 36
2.13. Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía. 37
2.14. Scatter Matrix usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía. 39
2.15. Heatmap usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía. 40
2.16. Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 1 en la comunidad autónoma de Andalucía. 42
2.17. Scatter Matrix usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía. 44
2.18. Heatmap usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía. 45

3.1. Scatter Matrix usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos. 55
3.2. Heatmap usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos. 56
3.3. Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos.. 58
3.4. Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos.. 59
3.5. Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos. 61
3.6. Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 2 con colisiones entre vehículos. 63
3.7. Scatter Matrix usando el algoritmo K-means, caso de estudio 2 con atropellos. 67
3.8. Heatmap usando el algoritmo K-means, caso de estudio 2 con atropellos. . 68
3.9. Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 2 con atropello. 70
3.10. Heatmap usando el algoritmo DBSCAN, caso de estudio 2 con atropellos. 71
3.11. Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con atropellos. 72
3.12. Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos. 73
3.13. Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 2 con atropellos. 75
4.1. Scatter Matrix usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave. 80
4.2. Heatmap usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave. 81
4.3. Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave. 82
4.4. Heatmap usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave. 83
4.5. Scatter Matrix usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave. 84

4.6. Heatmap usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave. 85
4.7. Scatter Matrix usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte. 89
4.8. Heatmap usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte. 90
4.9. Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte. 91
4.10. Heatmap usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte. 92
4.11. Scatter Matrix usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte. 94
4.12. Heatmap usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte. 95

Índice de tablas

2.1. Datos generales asociados a cada uno de los algoritmos, caso de estudio 1 en la comunidad autónoma de Madrid. 13
2.2. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid. 18
2.3. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid. 21
2.4. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid. 24
2.5. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid. 29
2.6. Datos generales asociados a cada uno de los algoritmos, caso de estudio 1 en la comunidad autónoma de Andalucía. 31

2.7. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía. 34
2.8. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía. 37
2.9. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía. 40
2.10. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía. 45
2.11. Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 1 en la comunidad autónoma de Madrid. . . . 48
2.12. Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 1 en la comunidad autónoma de Andalucía. . 50
3.1. Datos generales asociados a cada uno de los algoritmos, caso de estudio 2 con colisiones entre vehículos. 53
3.2. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos. . . 56
3.3. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos. 59
3.4. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos. 62
3.5. Datos generales asociados a cada uno de los algoritmos, caso de estudio 2 con atropellos. 65
3.6. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos. . . 68
3.7. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 2 con atropellos. 71
3.8. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Ward, caso de estudio 2 con atropellos. 74

4.1. Datos generales asociados a cada uno de los algoritmos, caso de estudio 3 en trazado con curva suave. 78
4.2. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave. . . . 81
4.3. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave. . . . 83
4.4. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave. 85
4.5. Datos generales asociados a cada uno de los algoritmos, caso de estudio 3 en trazado con curva fuerte. 87
4.6. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte. . . . 90
4.7. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte. . . . 92
4.8. Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte. . . . 95
4.9. Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 3 en trazado con curva suave. 98
4.10. Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 3 en trazado con curva fuerte. 100

1. Introducción

Esta práctica ha sido llevada a cabo para la asignatura de Inteligencia de Negocio de la universidad de Granada, asignatura de cuarto curso del Grado en Ingeniería Informática. Veremos el uso de algoritmos de aprendizaje no supervisado de agrupamiento para el análisis empresarial.
Así mismo, vamos a trabajar con el conjunto de datos de accidentes mortales de tráfico obtenidos de la página oficial de la DGT de todo el año 2013. [21]

Mediante las distintas variables que caracterizan a los accidentes se intentarán mostrar grupos de accidentes similares, así como relaciones entre los distintos tipos de accidentes para obtener información acerca de la gravedad del accidente.

Para la realización de la práctica se utilizará el lenguaje de programación Python en su versión 3.6.3 [5].Además nos ayudaremos de librerías como Matplotlib `https://matplotlib.org/`, Sklearn `http://scikit-learn.org/stable/`, Seaborn `https://seaborn.pydata.org/` y Scipy `https://www.scipy.org/`.

Para este estudio se han utilizado los siguientes algoritmos de agrupamiento (Clustering):

- K-means. [1]
- DBSCAN. [17]
- Spectral Clustering.[20]
- Mean Shift. [19]
- Ward. [18] (Aunque el algoritmo Ward solamente se utilizará en un solo caso de estudio apropiado para la extracción del dendograma junto con el heatmap.)
- MiniBatchKmean. [16]
- Birch. [7]

El conjunto de datos extraído de la DGT cuenta con un total de 90000 accidentes (instancias), pero como hemos comentado, cada accidente viene identificado por un conjunto

de sucesos; es ahí donde juegan un papel fundamental las variables del conjunto de datos, ya que nosotros cogeremos subconjuntos más pequeños, denominados casos de estudio, dependiendo de esas características de los accidentes. De hecho uno de los propósitos es ver diferencias y/o similitudes entre distintos tipos de accidentes, para así sacar conclusiones. Así se podrá tener un análisis bastante conciso de los casos de estudio que se escojan.

Principalmente en esta práctica se abordará el problema haciendo un estudio lo mas escrupuloso posible, reflejando con distintas gráficas y tablas con datos estadísticos la mayor cantidad de información posible.

Por último y una vez que se hayan mostrado todos los datos extraídos y las gráficas pintadas, se extraerán las conclusiones finales apropiadas para cada caso de estudio. En particular se usarán un total de 5 algoritmos de clustering, y a su vez sobre algunos de ellos se harán modificaciones para estudiar los distintos comportamientos de un mismo algoritmo con por ejemplo mismos datos de estudio pero distintos parámetros del algoritmo.

Por último comentar que se han obviado las gráficas y tablas referentes al algoritmo DBSCAN ya que nos aportaban poco debido al gran número de clusters y a los malos resultados que generaban.

2. Caso de estudio 1: Accidentes de tráfico ocurridos en las comunidades autónomas de Madrid y Andalucía.

En este primer punto se van a exponer dos casos de estudio distintos con el objetivo de poder obtener diferencias y comparar ambos análisis para llegar a una mejor comprensión del problema.

Un primer caso de estudio será dirigido a los accidentes de tráfico ocurridos en la comunidad autónoma de Madrid, y un segundo caso de estudio compuesto por los accidentes de tráfico ocurridos en la comunidad autónoma de Andalucía.

De entre ambos casos, podremos sacar datos como por ejemplo acerca de donde se producen más accidentes entre ambas comunidades, que tipos de accidentes son los más característicos de ambas comunidades, etc.

Este primer estudio me ha parecido curioso mostrarlo porque nos posibilita ya no solo a mostrar diferencias entre los distintos algoritmos de clustering que se han aplicado para el estudio, sino que también podemos observar cuáles son las características claves de los accidentes en cada comunidad y así poder compararlos.

Para el análisis de ambos casos de estudio, se cogerán dos algoritmos concretos que puedan ser claramente explotados, y se intentará explicar las similitudes y diferencias de los comportamientos de los mismos en dichos casos de estudio.

2.1. Caso de Estudio: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio de la comunidad autónoma de Madrid, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 14.114 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	32402.574027	4	0.844440	0.045392
MiniBatchKMeans	32584.427522	4	0.848114	0.027996
Birch	4661.577849	4	0.526679	0.303259
DBSCAN	2096.347820	13	0.410209	1.385544
MeanShift	86208.065169	56	0.961064	64.910662
Spectral	21712.489719	4	0.823545	89.585159

Tabla 2.1: Datos generales asociados a cada uno de los algoritmos, caso de estudio 1 en la comunidad autónoma de Madrid.

Para la realización de las tablas del tipo de la 2.1, se ha utilizado el siguiente código:

Listing 1: Sección de código en Python para la generación de la gráfica ScatterMatrix.

```
def DFValoresAlgoritmos(algoritmo, tiempo, nClusters,
CH, SC, DFTodosDatos):
df1 = pd.DataFrame({'Algoritmo':[algoritmo],
      'N.Clusters':[int(nClusters)],
      'Tiempo':[tiempo],
      'CH':[CH],
      'SH':[SC]})

return df1
```

En cada caso de estudio nos encontraremos con una tabla como la anterior, ya que nos permitirá valorar los algoritmos utilizados y los parámtreos que han sido pasados a dichos algoritmos.

En este caso, vemos como para el índice **Calinski-Harabasz** el mejor algoritmo es Mean Shift, recordemos que el índice CH se define como la razón entre la dispersión interior de los cluster y la dispersión entre clusters [2], y el objetivo claramente es maximizarlo. Básicamente viene a reflejar una de las máximas de los algoritmos de clustering, y es que un buen método de clustering debe **maximizar la similaridad intra-clusters** y **minimizar la similaridad inter-cluster** [2]. De esta forma podemos decir que Mean Shift es el que mejor comportamiento tiene teniendo en cuenta esta métrica. Ya que desde muy lejos le siguen algoritmos como K-means, MiniBatchKmeans y Spectral.

Para el índice **Silhouette** ocurre lo mismo que para el **Calinski-Harabasz**, teniendo Mean Shift la mejor métrica de los 6 utilizados. Recordemos que el índice **Silhouette** mide como de compactos y separados están los clusters, el intervalo de este índice está

entre [-1,1], en donde los valores cercanos a -1 hablan de una agrupación nada fiable y los valores cercanos a 1 hablan de una agrupación con una mayor confianza.

Como dato histórico en el año 1990, Kaufman y Rousseeuw sugirieron estimar el número óptimo de cluster K para el cual el índice **Silhouette** sea el mayor posible.

Así tendríamos que el algoritmo Mean Shift sería el que mejor se comporta, por lo tanto podríamos clonar el número de clusters que ha seguido dicho algoritmo en otros algoritmos, y así ver si se produce una mejora en sus métricas y gráficas.

Por último cabe decir que utilizando el tiempo en segundos como unidad de medida de lo que ha tardado cada algoritmo en ejecutarse, podemos afirmar que el algoritmo Spectral sería el que myor tiempo ha estado procesándose, seguido muy de cerca por el Mean Shift.

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio de la comunidad de Madrid, y al final un breve análisis.

2.1.1. Resultados algoritmo K-means, caso de estudio de la comunidad de Madrid.

El algoritmo **K-means** es uno de los más simples de todos los algoritmos de aprendizaje no supervisado que como es sabido soluciona problemas de clustering. La idea principal es definir k centroides, uno para cada cluster. Esos centroides deben ser colocados de foma astuta, ya que el alojamiento de los k centroides en diferentes posiciones nos dará, como es obvio, diferentes resultados. El paso siguiente es colocar cada dato con su centroide más cercano, en el siguiente punto necesitamos recalcular los nuevos centroides como baricentros de los clusters resultantes del paso anterior. Esto se repite hasta que no haya cambios. [12]
Para la extracción de esta gráfica 2.1se ha utilizado el siguiente fragmento de código 2:

Listing 2: Sección de código en Python para la generación de la gráfica ScatterMatrix.

```
1 def PintarScatterMatrix(DFclusterSinOutliersAux, scatter_dir,
2 nombreAlgoritmo, casoEstudio):
3
4 plt.figure()
5
6 variables = list(DFclusterSinOutliersAux)
```

```
variables.remove('cluster')
sns_plot = sns.pairplot(DFclusterSinOutliersAux, vars=variables,
hue="cluster", palette='Paired', plot_kws={"s": 25},
diag_kind="hist")

sns_plot.fig.subplots_adjust(wspace=.03, hspace=.03);

plt.savefig(scatter_dir + nombreAlgoritmo + casoEstudio)
plt.close()
```

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:

Para realizar el filtrado se utiliza el siguiente código:

Listing 3: Sección de código en Python para la generación de la gráfica ScatterMatrix.

```
minimoTama = 3
DFclusterSinOutliers = datasetConCluster[datasetConCluster.
groupby('cluster').cluster.transform(len) > minimoTama]
numeroClusterPostFiltrado =
len(set(DFclusterSinOutliers['cluster']))
```

De los 4 clusters hay 4 con más de 3 elementos. Del total de 14114 elementos, se seleccionan 14114.
La siguiente gráfica 2.1 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

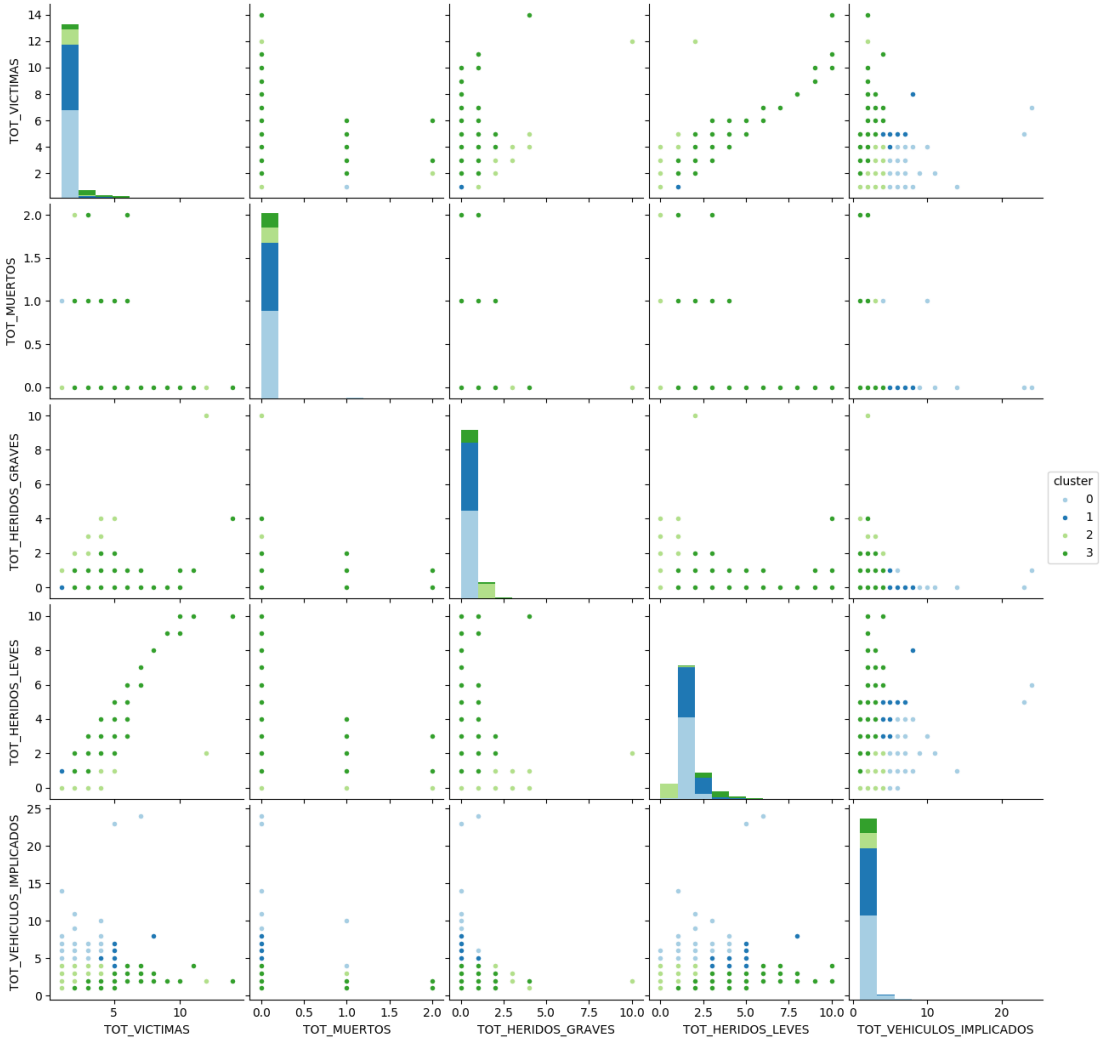

Figura 2.1: Scatter Matrix usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente gráfica (Heatmap) 2.2 representa a la tabla 2.2 pero con sus datos normalizados:

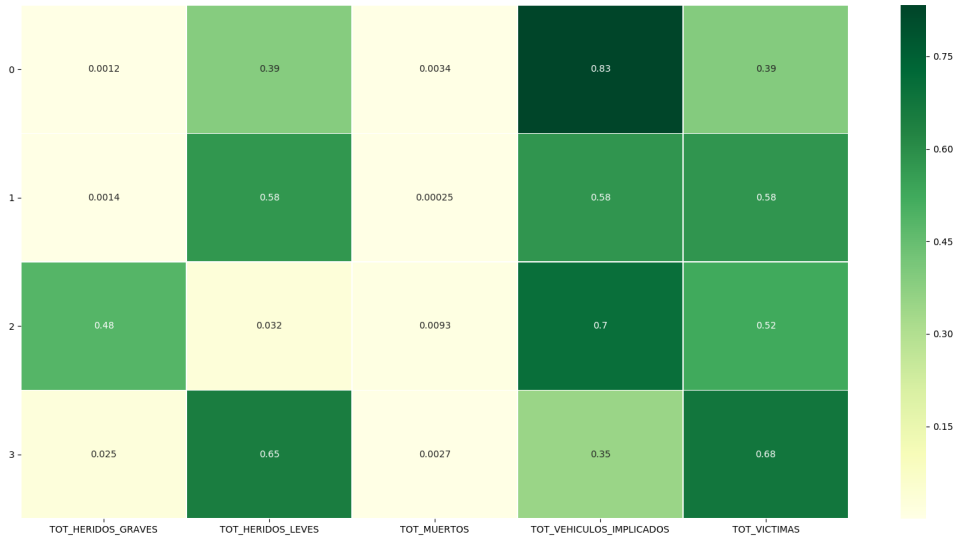

Figura 2.2: Heatmap usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid.

Para la extracción de esta gráfica 2.2se ha utilizado el siguiente fragmento de código 5:

Listing 4: Sección de código en Python para la generación de la gráfica Heatmap.

```
def PintarHeatmap(DFMediasNormal, heatmap_dir, nombreAlgoritmo,
casoEstudio,clusters_restantes):
plt.figure()


plt.subplots(figsize=(20, 10))

sns.heatmap(data=DFMediasNormal, annot=True, linewidths=0.5,
yticklabels=clusters_restantes, cmap='YlGn')

plt.xticks(rotation=0)
plt.yticks(rotation=0)
plt.savefig(heatmap_dir + nombreAlgoritmo + casoEstudio)
plt.close()
```

La siguiente tabla 2.2 está compuesta por los datos en media referentes al algoritmo K-means con todas las variabes que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.003300	1.056547	0.009300	2.265337	1.069147
1	0.003272	1.329869	0.000577	1.335065	1.333718
2	1.077519	0.071490	0.020672	1.564169	1.169681
3	0.113761	2.914679	0.011927	1.570642	3.040367

Tabla 2.2: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Madrid.

Para obtener los valores promedios de todas las variables sobre cada cluster en las tablas del tipo a la tabla 2.2, así como sus desviaciones típicas se utiliza el siguiente código:

Listing 5: Sección de código en Python para la generación de la gráfica Heatmap.

```
1  def calcularMediaStd(cluster):
2  vars = list(cluster)
3  vars.remove('cluster')
4  return dict(np.mean(cluster[vars],axis=0)),
5  dict(np.std(cluster[vars],axis=0))
6
7  def DFClusterConMedias(dataFrame):
8
9  listaClusters = list(set(dataFrame['cluster']))
10
11 DFMedia = pd.DataFrame()
12 DFStd = pd.DataFrame()
13
14 for cluster_n in listaClusters:
15 cluster_i = dataFrame[dataFrame['cluster'] == cluster_n]
16 DicMedia, DicStd = calcularMediaStd(cluster=cluster_i)
17 auxDFMedia = pd.DataFrame(DicMedia,index=[str(cluster_n)])
18 auxDFStd = pd.DataFrame(DicStd,index=[str(cluster_n)])
19 DFMedia = pd.concat([DFMedia, auxDFMedia])
20 DFStd = pd.concat([DFStd, auxDFStd])
21
22 return DFMedia, DFStd
```

2.1.2. Resultados algoritmo MiniBatchKmeans, caso de estudio de la comunidad de Madrid.

El algoritmo **MiniBatchK-means** es una propuesta alternativa al K-means para clustering de datasets masivos. Una de las ventajas con respecto a K-means es la reducción en coste computacional por no usar todo el dataset en cada iteración. Aunque también produce una pérdida en la calidad del clustering. En nuestro análisis se tiene en cuenta una comparación entre ambos algoritmos. **??**

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 14114 elementos, se seleccionan 14114.
La siguiente gráfica 2.3 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

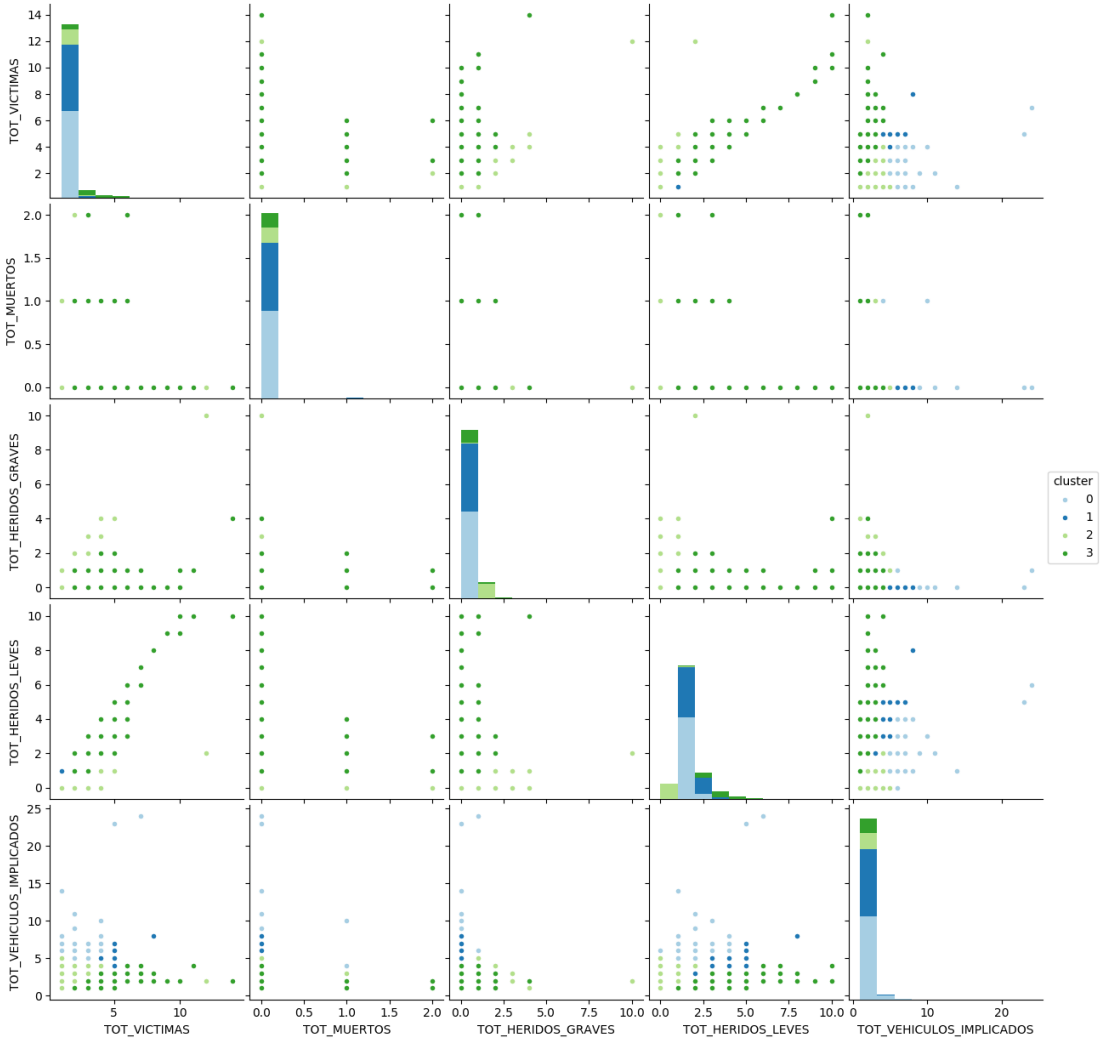

Figura 2.3: Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente gráfica (Heatmap) 2.4 representa los datos de la tabla 2.3 pero con sus datos normalizados:

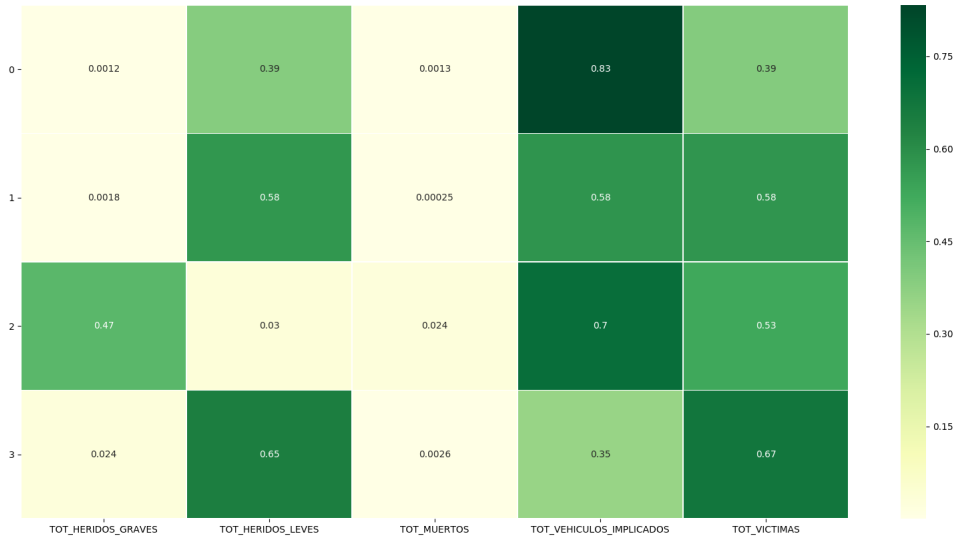

Figura 2.4: Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente tabla 2.3 está compuesta por los datos en media referentes al algoritmo MiniBatchKmeans con todas las variabes que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.003168	1.062764	0.003470	2.272028	1.069403
1	0.004062	1.315667	0.000580	1.326692	1.320309
2	1.041736	0.065943	0.052588	1.547579	1.160267
3	0.110912	2.941860	0.011628	1.606440	3.064401

Tabla 2.3: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Madrid.

2.1.3. Resultados algoritmo Birch, caso de estudio de la comunidad de Madrid.

Birch es un algoritmo de aprendizaje no supervisado usado para clustering jerárquico sobre datasets grandes. Una de sus ventajas es la capacidad para agrupar incremental y dinámicamente los clusters. En la mayoría de los casos Birch solo necesita un único escaneo de los datos. [8]

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 14114 elementos, se seleccionan 14114.
La siguiente gráfica 2.5 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

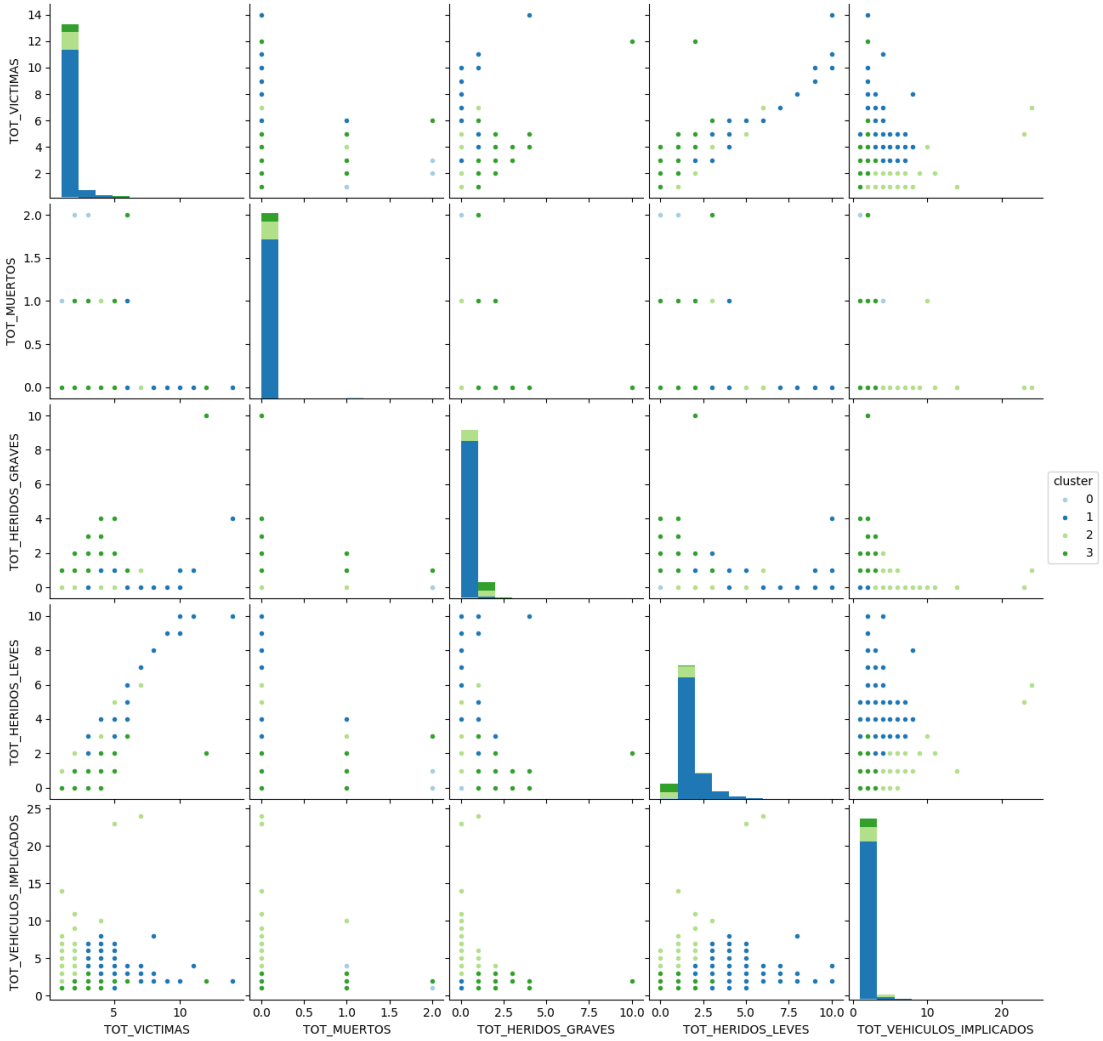

Figura 2.5: Scatter Matrix usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente gráfica (Heatmap) **??** representa los datos de la tabla 2.4 pero con sus datos normalizados:

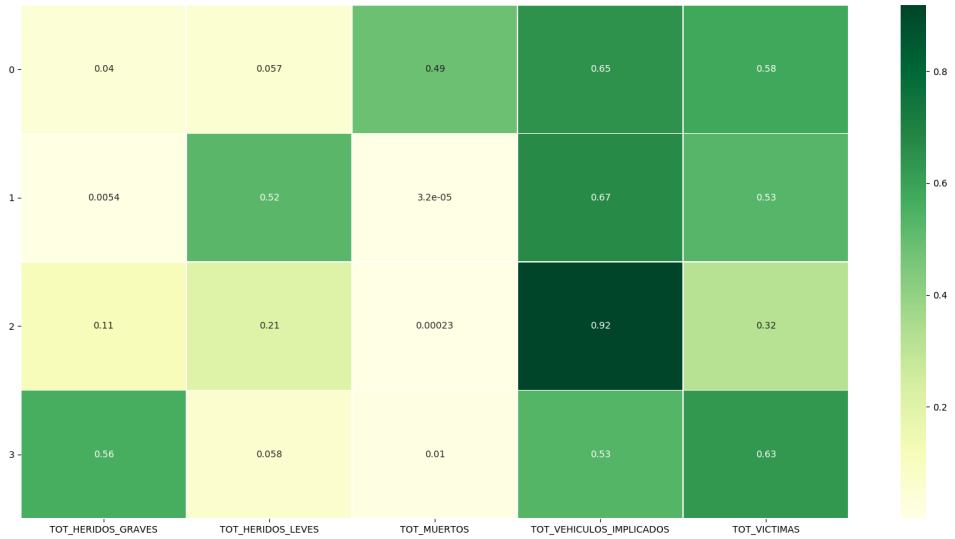

Figura 2.6: Heatmap usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente tabla 2.4 está compuesta por los datos en media referentes al algoritmo Birch con todas las variabes que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.089744	0.128205	1.102564	1.461538	1.320513
1	0.013945	1.347721	0.000083	1.723998	1.361750
2	0.367625	0.673222	0.000756	3.003782	1.041604
3	1.128936	0.116942	0.020990	1.070465	1.266867

Tabla 2.4: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Madrid.

2.1.4. Resultados algoritmo Mean Shift, caso de estudio de la comunidad de Madrid.

El algoritmo **Mean Shift** es una técnica de clustering no paramétrica que no requiere conocimiento del número de clusters. Dado N puntos de datos, en un espacio d-dimensional, el núcleo de densidad multivariado estima obtener con K (x). [6]

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 56 clusters hay 25 con más de 3 elementos. Del total de 14114 elementos, se seleccionan 14062.
La siguiente gráfica 2.7 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

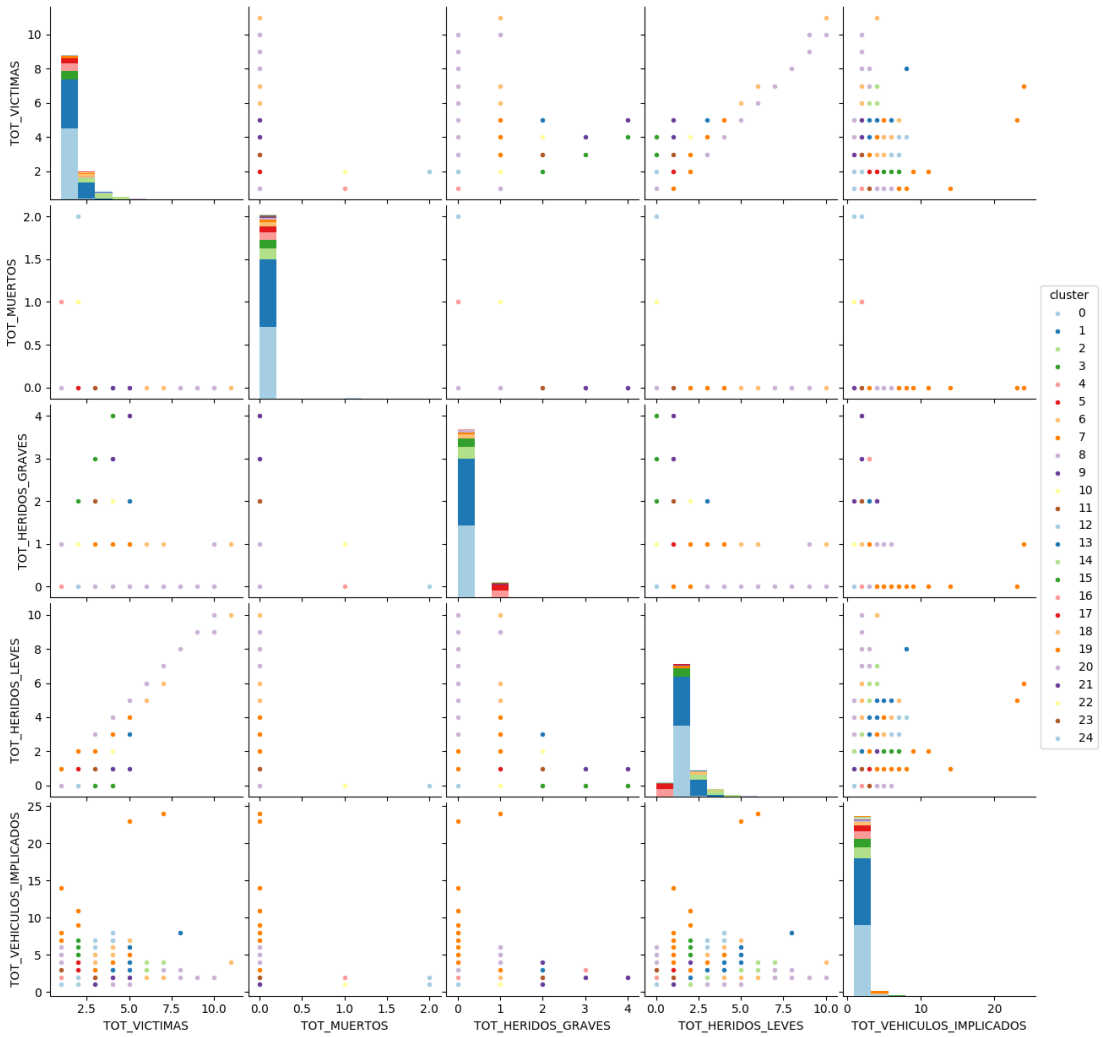

Figura 2.7: Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 1 en la comunidad autónoma de Madrid.

En este algoritmo no se ha considerado la extracción de un heatmap y de la tabla de datos de las medias debido a la gran cantidad de clusters que propone el propio algoritmo. Por ello y dado que no nos sería de mucha utilidad, se ha decidido obviar dicha gráfica y tabla para este algoritmo.

2.1.5. Resultados algoritmo Spectral, caso de estudio de la comunidad de Madrid.

El algoritmo **Spectral** es una técnica que hace uso del espectro de la matriz de similaridad de los datos para realizar la reducción dimensional antes antes del clustering en menos dimensiones. La matriz de similaridad es proporcionada como una entrada. Se suele utilizar en segmentación de imágenes. [10]

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 14114 elementos, se seleccionan 14114.
La siguiente gráfica 2.8 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

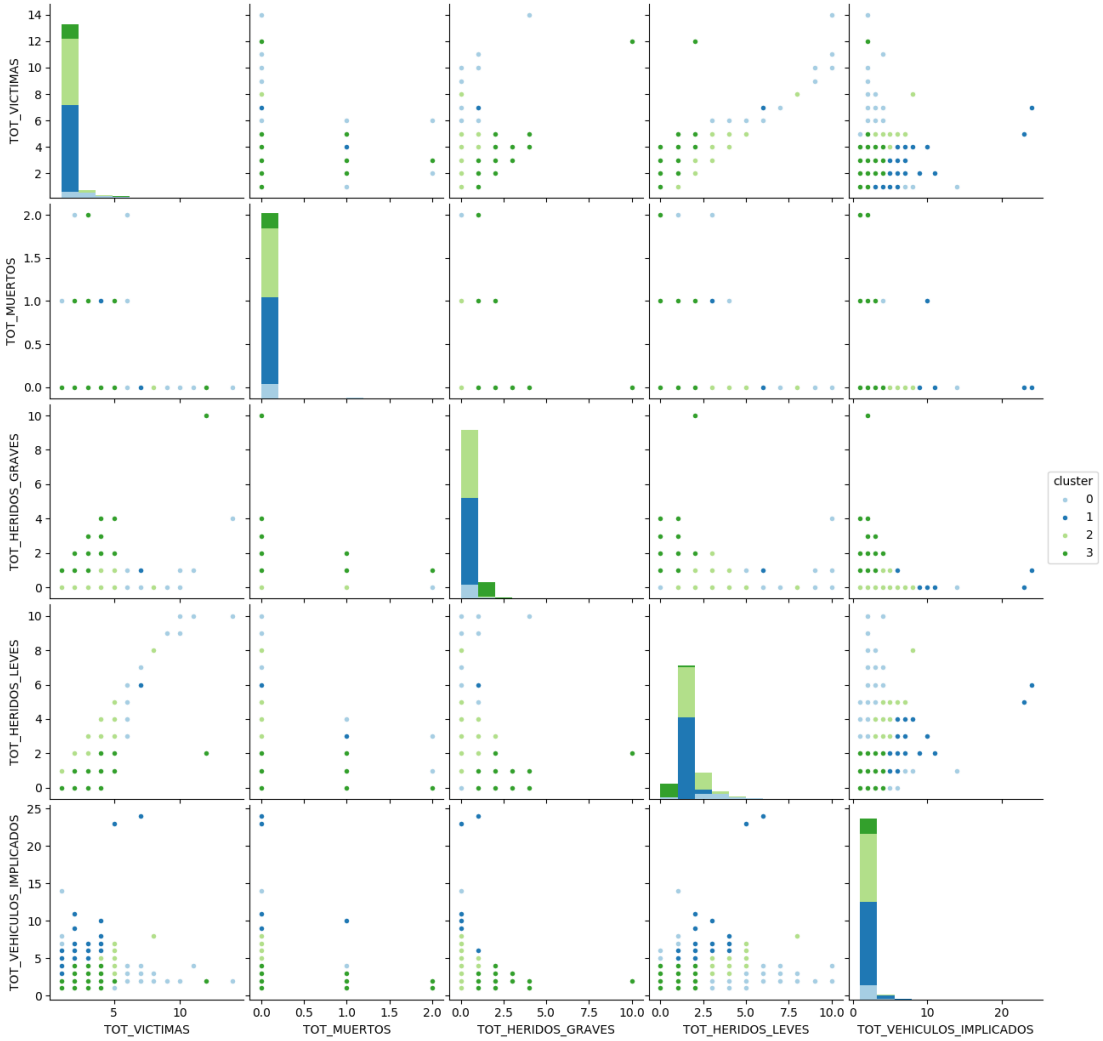

Figura 2.8: Scatter Matrix usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente gráfica (Heatmap) 2.9 representa los datos de la tabla 2.5 pero con sus datos normalizados:

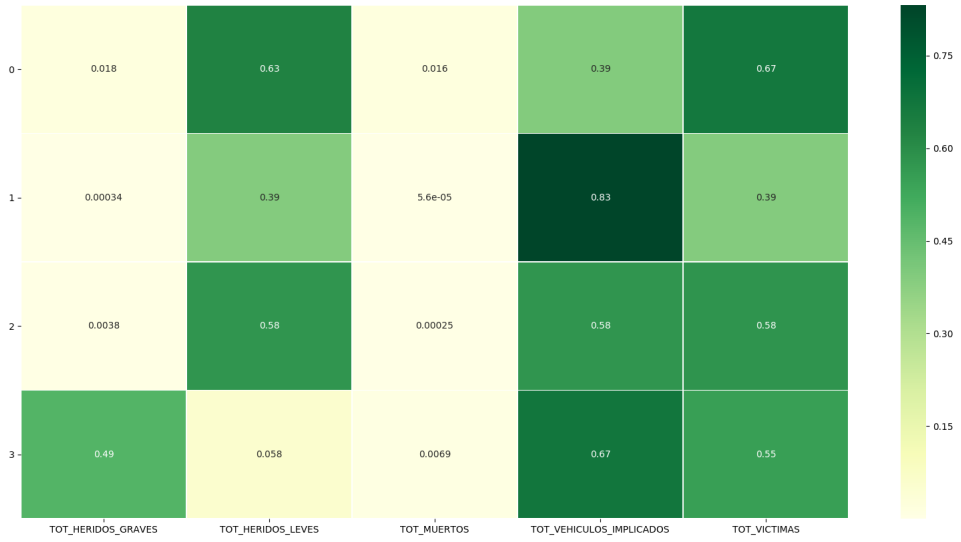

Figura 2.9: Heatmap usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid.

La siguiente tabla 2.5 está compuesta por los datos en media referentes al algoritmo Spectral con todas las variabes que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.008806	1.334418	0.000574	1.339587	1.343798
1	0.000911	1.066535	0.000152	2.262494	1.067598
2	1.083333	0.129252	0.015306	1.498299	1.227891
3	0.077807	2.714412	0.070734	1.692308	2.862953

Tabla 2.5: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Madrid.

2.1.6. Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid.

Para terminar con el estudio de este pequeño caso de uso y antes de dar paso al verdadero objetivo de esta sección, que es el de comparar dos casos de estudios distintos y visualizar como se comportan los distintos algoritmos, se darán unas pequeñas interpretaciones de los resultados mostrados anteriormente.

En primer lugar podemos observar el comportamiento del algoritmo K-means en la figura 2.1, en donde se muestra mediante un scatter matrix los valores que permiten diferenciar los distintos clusters, en este caso para 4 clusters. Vemos como el cluster 0 representa aquellos accidentes donde hay entre 5 y 10 vehículos implicados (aunque podemos ver algunos datos sueltos por 25 vehículos implicados) y un número bajo de heridos leves (entre 0 y 5). Este mismo cluster apenas cuenta con accidentes en donde ha habido muertos. Al contrario que los clusters 2 y 3 que cuentan con los accidentes donde menos vehículos implicados ha habido, pero más muertos se han producido. Si miramos la figura 2.1 vemos como efectivamente de media, los clusters 2 y 3 son los que contienen a los accidentes donde más muerto se han producido.

Comparando el algoritmo K-mean con los resultados del algoritmo Birch, que tiene el mismo número de clusters, nos encontramos con que el cluster 0 contiene a los accidentes de tráfico en donde más muertos ocurren de media, y sin embargo no contiene a los accidentes en donde más vehículos en promedio hay implicados.
Con respecto al cluster 3 (para el algoritmo Birch), también parece abarcar algunos accidentes donde ocurren muertos y pocos vehículos implicados, aunque puede parecer que podríamos estar en un caso en donde los datos estén bastante dispersos.

2.2. Caso de Estudio: Accidentes de tráfico ocurridos en la comunidad autónoma de Andalucía.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio de la comunidad autónoma de Andalucía, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH), la métrica Silhouette (SC) y el tiempo que ha tardado el algoritmo en ejecutarse en segundos. Para este caso de estudio se ha contado con un total de 13.944 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	25849.419373	4	0.803007	0.039327
MiniBatchKMeans	22824.948638	4	0.771541	0.031887
Birch	3890.444772	4	0.629435	0.385256
DBSCAN	1098.641581	16	0.293370	1.924708
MeanShift	25988.558672	55	0.924912	64.856352
Spectral	22577.839251	4	0.773327	115.230545

Tabla 2.6: Datos generales asociados a cada uno de los algoritmos, caso de estudio 1 en la comunidad autónoma de Andalucía.

En este caso y al contrario de lo ocurrido para la comunidad autónoma de Madrid, las diferencias en la métrica Calinski-Harabasz (CH) son ínfimas, lo cuál nos índica que los algoritmos K-means, MiniBatchKmeans, Mean Shift y Spectral maximizan la similaridad intra-cluster y minimizan la similaridad inter-cluster, por tanto cumplimos uno de los objetivos con la mayoría de los algoritmos.

Para la métrica Silhouette (SH) ocurre exactamente lo mismo que para el caso de estudio de la comunidad de Madrid, en donde hay un algoritmo vencedor por encima del resto, Mean Shift, con bastante diferencia sobre los demás. También como ocurría para el anterior caso de estudio, el claro perdedor sería DBSCAN que con sus 13 clusters no mejora en este caso de estudio los resultados de la tabla 2.1

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio de la comunidad de Andalucía, y al final un breve

análisis.

2.2.1. Resultados algoritmo K-means, caso de estudio de la comunidad de Andalucía.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 13944 elementos, se seleccionan 13944
La siguiente gráfica 2.10 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

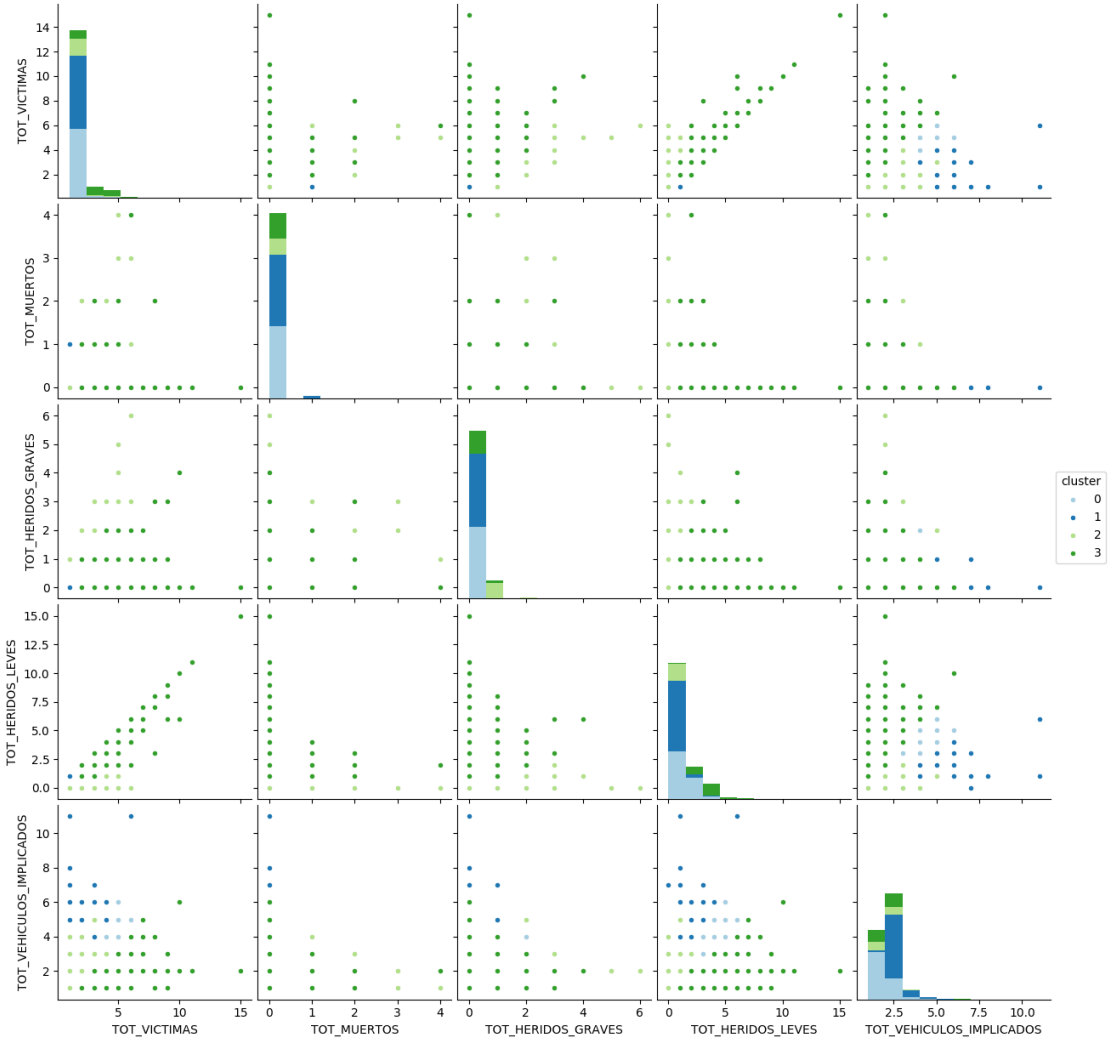

Figura 2.10: Scatter Matrix usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente gráfica (Heatmap) 2.11 representa los datos de la tabla 2.7 pero con sus datos normalizados:

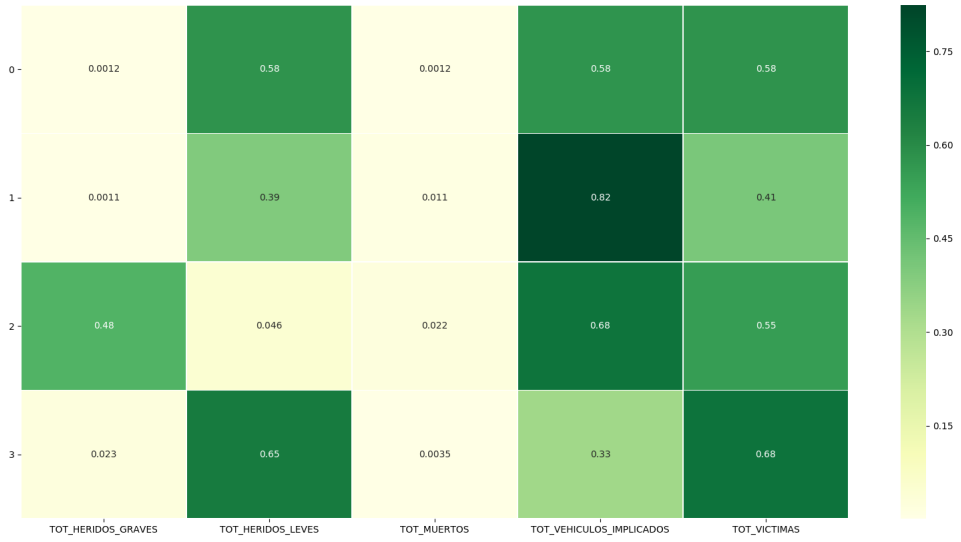

Figura 2.11: Heatmap usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente tabla 2.7 está compuesta por los datos en media referentes al algoritmo K-means con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.002803	1.403214	0.002803	1.398729	1.408819
1	1.095536	0.104933	0.050117	1.533281	1.250587
2	0.002945	1.022823	0.027793	2.135468	1.053562
3	0.109989	3.103613	0.016472	1.589798	3.230074

Tabla 2.7: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 1 en la comunidad autónoma de Andalucía.

2.2.2. Resultados algoritmo MiniBatchKmeans, caso de estudio de la comunidad de Andalucía.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 13944 elementos, se seleccionan 13944.
La siguiente gráfica 2.12 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

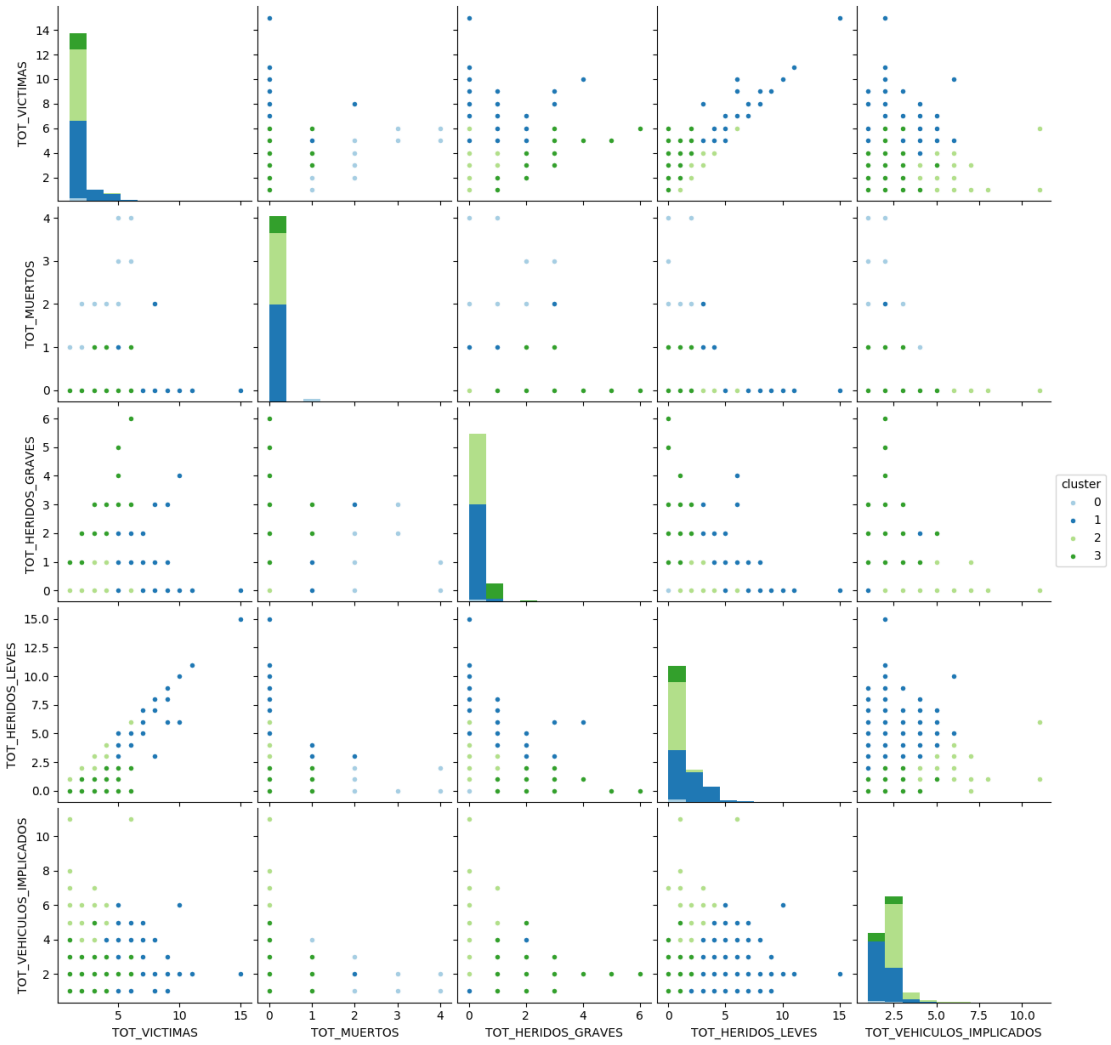

Figura 2.12: Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente gráfica (Heatmap) 2.13 representa los datos de la tabla 2.8 pero con sus datos normalizados:

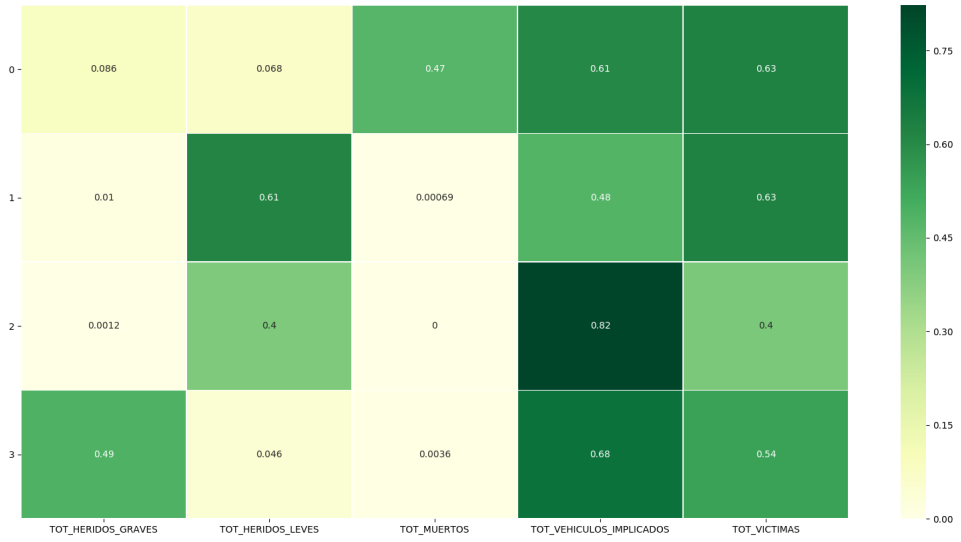

Figura 2.13: Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente tabla 2.8 está compuesta por los datos en media referentes al algoritmo MiniBatchKmeans con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.109989	3.103613	0.016472	1.589798	3.230074
1	0.003002	1.042589	0.008068	2.155910	1.053659
2	0.002803	1.403214	0.002803	1.398729	1.408819
3	1.013768	0.097101	0.124638	1.499275	1.235507

Tabla 2.8: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 1 en la comunidad autónoma de Andalucía.

2.2.3. Resultados algoritmo Birch, caso de estudio de la comunidad de Andalucía.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 13944 elementos, se seleccionan 13944.
La siguiente gráfica 2.14 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

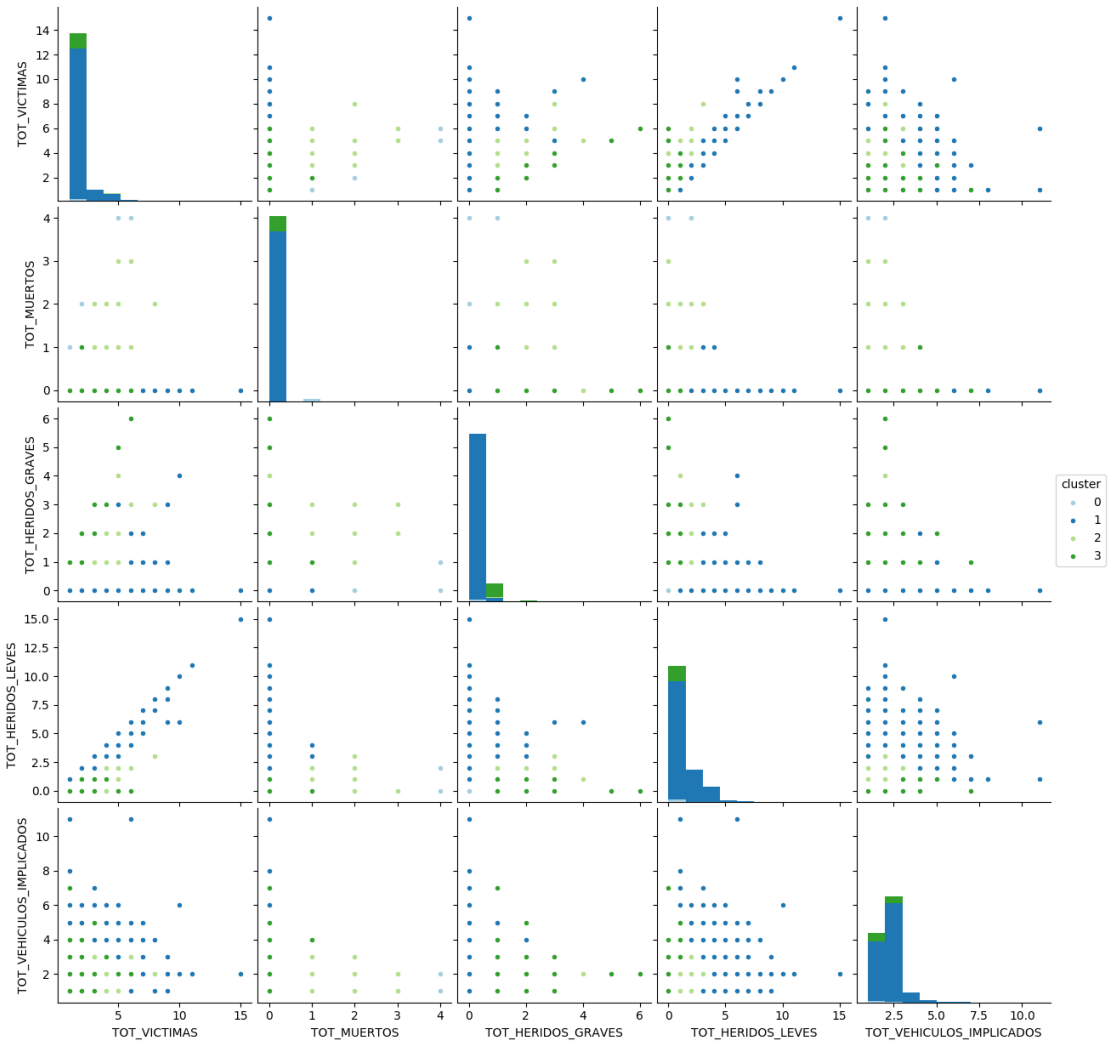

Figura 2.14: Scatter Matrix usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente gráfica (Heatmap) 2.15 representa los datos de la tabla 2.9 pero con sus datos normalizados:

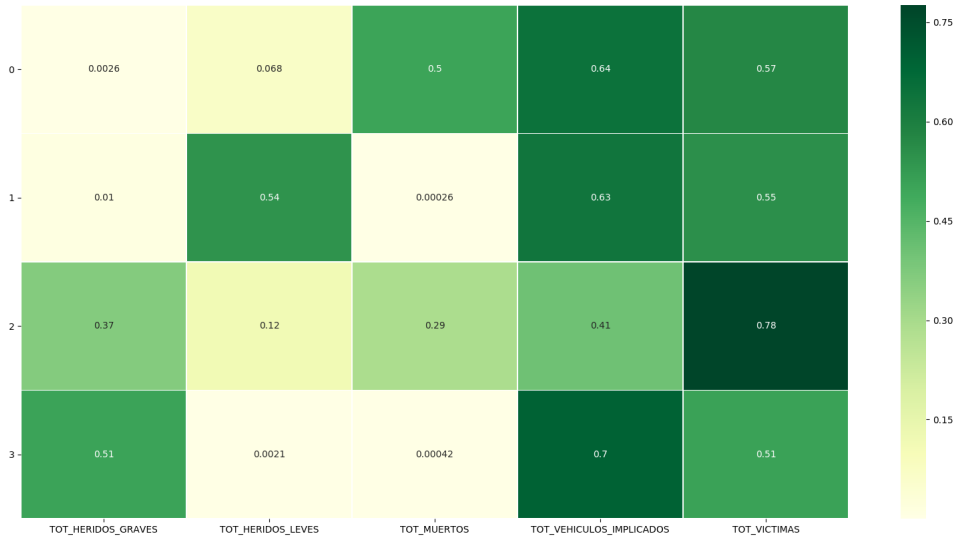

Figura 2.15: Heatmap usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente tabla 2.9 está compuesta por los datos en media referentes al algoritmo Birch con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.005780	0.150289	1.104046	1.416185	1.260116
1	0.028320	1.506188	0.000714	1.749326	1.535221
2	1.500000	0.480000	1.200000	1.660000	3.180000
3	1.079821	0.004484	0.000897	1.486996	1.085202

Tabla 2.9: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Birch, caso de estudio 1 en la comunidad autónoma de Andalucía.

2.2.4. Resultados algoritmo Mean Shift, caso de estudio de la comunidad de Andalucía.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 16 clusters hay 16 con más de 3 elementos. Del total de 13944 elementos, se seleccionan 13944
La siguiente gráfica 2.16 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

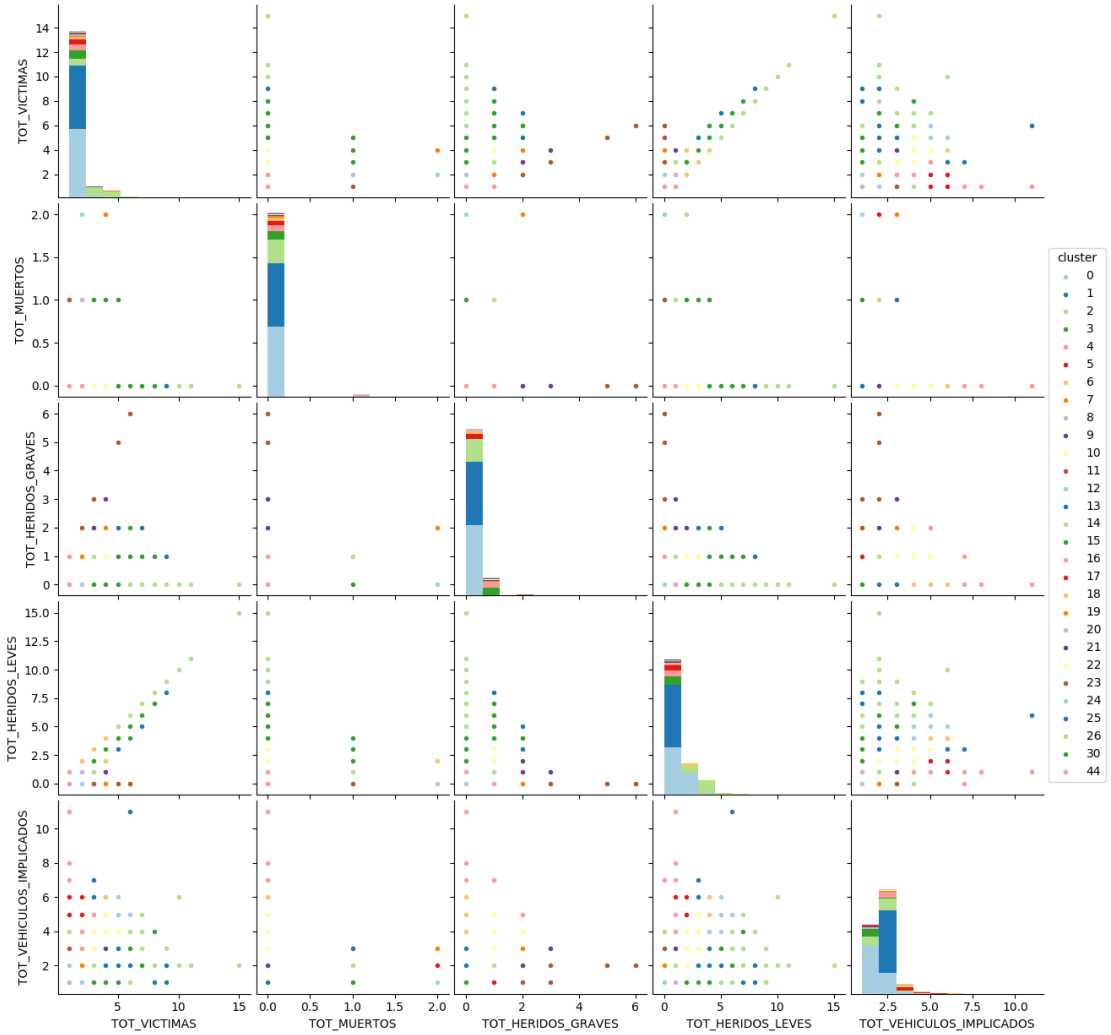

Figura 2.16: Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 1 en la comunidad autónoma de Andalucía.

En este algoritmo no se ha considerado la extracción de un heatmap y de la tabla de datos de las medias debido a la gran cantidad de clusters que propone el propio algoritmo. Por ello y dado que no nos sería de mucha utilidad, se ha decidido obviar dicha gráfica y tabla para este algoritmo.

2.2.5. Resultados algoritmo Spectral, caso de estudio de la comunidad de Andalucía.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:

De los 4 clusters hay 4 con más de 3 elementos. Del total de 13944 elementos, se seleccionan 13944.
La siguiente gráfica 2.17 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

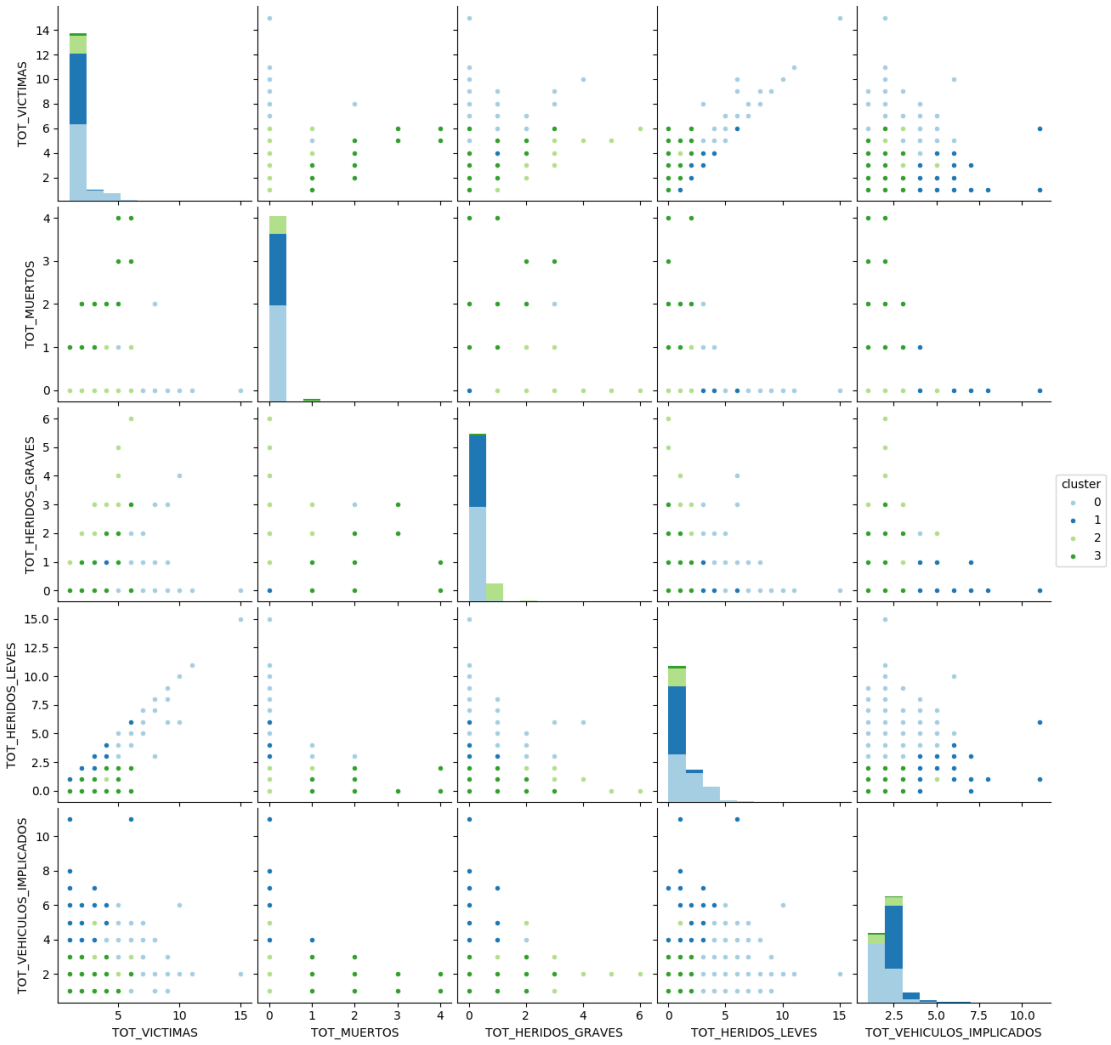

Figura 2.17: Scatter Matrix usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente gráfica (Heatmap) 2.18 representa los datos de la tabla 2.10 pero con sus datos normalizados:

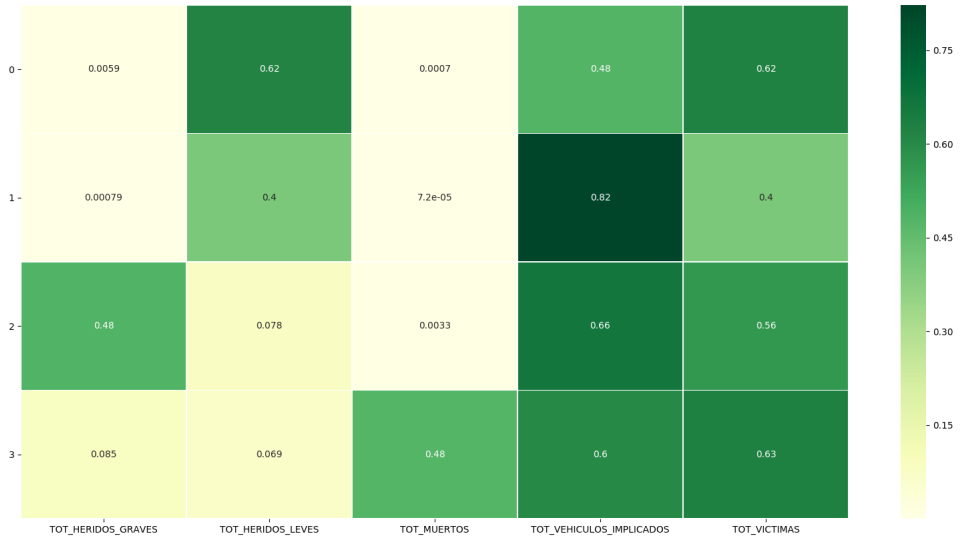

Figura 2.18: Heatmap usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía.

La siguiente tabla 2.10 está compuesta por los datos en media referentes al algoritmo Spectral con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.002073	1.058790	0.000188	2.164877	1.061051
1	0.017751	1.853339	0.002113	1.443646	1.873204
2	0.202899	0.164251	1.135266	1.434783	1.502415
3	1.094595	0.175676	0.007508	1.505255	1.277778

Tabla 2.10: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 1 en la comunidad autónoma de Andalucía.

2.2.6. Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Andalucía.

Para terminar con el estudio de este pequeño caso de uso y antes de dar paso al verdadero objetivo de esta sección, que es el de comparar dos casos de estudios distintos y visualizar

como se comportan los distintos algoritmos, se darán unas pequeñas interpretaciones de los resultados mostrados anteriormente.

Para el algoritmo K-means:

Se puede apreciar en la gráfica 2.17 como el cluster 3 se refiere a los accidentes donde hay un alto número de vícitimas y de heridos leves. Este cluster está muy relacionado con altos valores de ambas variables.
Si queremos ver que cluster contiene la mayor parte de los datos en donde ocurren muertos, debemos de acudir al cluster 2, el cluál contiene accidentes donde previsiblemente más muertos ocurren y menos vehículos hay implicados.

Para el algoritmo Birch:

Vemos de nuevo como para el algoritmo K-means que en el cluster 1, hay una fuerte relación entre las víctimas de un accidente y los heridos leves, ya que este cluster parece tener aquellos accidentes donde hay implicadas un número alto de víctimas e igual de alto con los heridos leves.

Como principal diferencia con respecto al algoritmo K-means, podemos observar en la gráfica 2.14 como el cluster 1 abarca aquellos accidentes en donde no se producen muertos pero si hay un alto número de víctimas, y enfocándolo con el cluster similar en el algoritmo K-means que sería el 3, ocurre en la gráfica 2.17 que si abarca muertes a la vez que un número alto de víctimas.

2.3. Interpretación de la segmentación: Accidentes de tráfico ocurridos en la comunidad autónoma de Madrid y Andalucía.

Por último y para cerrar este caso de estudio, se habló al principio del mismo que se intentaría ver las diferencias entre los dos casos de distintas comunidades autónomas, o las similitudes que hubiese.

En este caso, se encuentran bastantes mas características similares que diferentes. Por ejemplo comenzando con lo más evidente, ocurren prácticamente el mismo número de

accidentes de tráfico, con 14.114 ocurridos en Madrid por los 13.944 que ocurrieron en Andalucía.

En segundo lugar, se da el caso de que en ambos lados se aprecia como los clusters que recogen los datos donde mas muertes han habido, a su vez son datos con pocos vehículos implicados, algo que al principio del estudio podría parecer contrario a la lógica.

Como similaridad notable vemos como el algoritmo K-means en el caso de estudio referente a la comunidad autónoma de Madrid 2.1, nos indica que los clusters que recogen los accidentes con más muertes ocurridas no son los que más víctimas tienen. Al igual ocurre con el mismo algoritmo en el caso de estudio de la comunidad autónoma de Andalucía 2.10. Repitiéndose prácticamente el patrón.

Visualizando la gráfica heatmap del algoritmo K-mean 2.13 para el caso de estudio de la comunidad de Andalucía, los datos reflejados en cuanto a heridos graves, vemos como en Andalucía el mismo cluster que agrupa a los datos donde más heridos graves ocurren también agrupa a los accidentes donde más muertos ocurren en promedio.
Sin embargo, para el caso de estudio de la comunidad de Madrid, visualizando la gráfica heatmap del algoritmo K-means 2.2, se puede observar como el cluster 2 que es el que en promedio contiene los accidentes con más heridos graves no contiene a los accidentes más mortales, como digo haciendo uso de los valores medios calculados.

Y por último decir que los datos en promedio son prácticamente idénticos, habiendo una diferencia ínfima o indetectable en algunos casos.

2.4. Modificación de los parámtros para algunos algoritmos referentes al caso de estudio de la comunidad de Madrid.

En esta sub-sección nos encargaremos de realizar un estudio con mayor detalle de algunos de los algoritmos detallados anteriormente , de tal forma que podamos entender mejor los parámetros pasados a los mismos.

En este caso vamos a realizar este análisis sobre los algoritmos K-means, DBSCAN y Birch. Estos dos últimos serán estudiados debido a los malos resultados de la sub-sección anterior 2.1, y para ello intentaremos modificarles sus parámetros para optar a tener mejores resultados.

Para el análisis nos han salido los siguientes resultados referentes a los datos asociados a cada algoritmo utilizado para este caso de estudio de la comunidad autónoma de Madrid, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 14.114 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	38730.264822	6	0.890841	0.063403
Birch	3792.641606	8	0.436158	0.319835
DBSCAN	150.902713	3	0.610556	3.213082

Tabla 2.11: Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 1 en la comunidad autónoma de Madrid.

En la tabla se aprecian diferencias con respecto al índice Calinski-Harabasz (CH) en los 3 algoritmos, para el algoritmo K-means se nota una cierta mejoría ya que pasamos de 32.402 a 38.730, luego parece ser que nos beneficia la modificación, que ha sido pasar de 4 a 6 cluster de entrada. Para los algoritmos Birch y DBSCAN sin embargo se sufre una involución, ya que pasamos a tener peores índices CH, teniendo al DBSCAN como gran damnificado de las modificaciones. La modificación realizada para el algoritmo Birch ha

sido pasar de un k (número de clusters de entrada) de 4 a 8, y podemos decir con total certeza que no ha funcionado.
Después para el algoritmo DBSCAN hemos modificado el parámetro eps, que no es más que la distancia que hay entre 2 muestras para considerarlas en el mismo vecindario, dejando el parámetro min_samples a 10.

En lo referente a la métrica Silhouette (SC) observamos como el algoritmo K-means de nuevo mejora su resultado anterior de la tabla 2.1. Y en cuanto al algoritmo Birch y DBSCAN ocurre que para el primero(Birch) disminuye aún más su anterior resultado, que ya era pésimo, y para el segundo (DBSCAN) ocurre que mejora sustancialmente, lo que nos puede dar una idea para posteriores análisis variar algún parámetro más como por ejemplo el min_samples. Como ya hemos citado anteriormente 2.1 , en algunos documentos científicos se habla que para encontrar el número correcto de K para los algoritmos que lo requieren como entrada, se debe hacer mediante la maximización del índice Silhouette, luego para el algoritmo K-means al haberlo incrementado el índice podemos decir qué, bajo esta sospecha, el nuevo número de K funciona correctamente para dicho algoritmo. No así para el algoritmo Birch, que como hemos comentado sufre una disminución en este índice.

Por último con respecto a los tiempos necesitados para ejecutar cada algoritmo se ve una subida en las tres modificaciones con respecto a la tabla 2.1, hasta tal punto que en el caso del DBSCAN lo triplica prácticamente con la nueva modificación.

Para terminar con este análisis podemos decir como conclusión final que con respecto al algoritmo K-means nos ha servido realizar la modificación ya que hemos experimentado mejoras en todos los aspectos. Para el algoritmo DBSCAN es algo contradictorio, ya que en unos casos se mejora y en otros se empeora. Y para el Birch, sentimos profunda decepción puesto que no se han obtenido para ninguna métrica mejoras, lo cuál nos indica que para este dataset este no es el camino.

2.5. Modificación de los parámtros para algunos algoritmos referentes al caso de estudio de la comunidad de Andalucía.

En esta sub-sección nos encargaremos de nuevo de realizar un estudio con mayor detalle de algunos de los algoritmos detallados anteriormente , de tal forma que podamos entender mejor los parámetros pasados a los mismos.

En este caso vamos a realizar este análisis sobre los algoritmos K-means, DBSCAN y Birch. Estos dos últimos serán estudiados debido a los malos resultados de la sub-sección anterior 2.6, y para ello intentaremos modificarles sus parámetros para optar a tener mejores resultados.

Para el análisis nos han salido los siguientes resultados referentes a los datos asociados a cada algoritmo utilizado para este caso de estudio de la comunidad autónoma de Madrid, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 13.944 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	51939.661901	9	0.885453	0.100871
Birch	5642.095450	3	0.647677	0.313237
DBSCAN	5489.784198	3	0.607339	1.956589

Tabla 2.12: Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 1 en la comunidad autónoma de Andalucía.

Para esta modificación se han considerado las siguientes modificaciones:

Algoritmo K-means hemos pasado de tener un K (parámetro de clusters de entrada) igual a 4 a 9.

Algoritmo Birch hemos pasado de tener un k igual a 4 a 3.

Algoritmo DBSCAN hemos pasado de un eps con 0.1 a otro igual a 0.2, recordemos que el eps es la distancia que hay entre 2 muestras para considerarlas en el mismo vecindario, y hemos pasado de un min_samples a 10 a otro con un valor de 1000. El parámetro min_samples es el tamaño de un vecindario para que un punto se considere como punto central.

Una vez descritos los cambios realizados para los tres algoritmos, podemos asegurar en primer lugar que el algoritmo K-means mejora notablemente con respecto a los datos de la tabla 2.6, tanto para el CH, el SC y el tiempo empleado para ejecutarlo. Luego es un claro indicativo de que las modificaciones en este caso han ido como esperábamos.

En segundo lugar, observando los resultados del algoritmo Birch podemos decir que también han ido muy bien las modificaciones, ya que mejoramos los índices CH y SC, así como el tiempo empleado para ejecutarlo. Luego como comentábamos para el algoritmo K-means, en este caso han ido correctamente las modificaciones, consiguiendo mejorar todas las estadísticas.

Y por último, nos quedaría el algoritmo DBSCAN que tan malos resultados está obteniendo a lo largo de estos dos casos de estudio. En este caso y con las métricas sobre el papel podemos decir que en este caso se ha experimentado una mejora muy buena para la métrica CH, ya que conseguimos pasar de un índice de CH bajísimo en la tabla 2.6 a un índice a la par que el del algoritmo Birch. Lo cuál nos indica que se maximiza la similaridad intra-cluster y se minimiza inter-custer bastante mejor de lo que lo hacía en anteriores estudios. Lo mismo ocurre para el parámetro SC, ya que duplicamos su valor de manera estratosférica. En cuanto al tiempo empleado para su ejecución, podemos considerarlo prácticamente el mismo.

Para finalizar, y a modo de conclusión, podemos asegurar que con estas modificaciones los 3 algoritmos mejoran increíblemente, y es que los resultados de la tabla 2.12 son con mucha distancia mejores que los resultados anteriormente mostrados en la tabla 2.6

3. Caso de estudio 2: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos y atropellos.

En este segundo punto se van a exponer dos casos de estudio similares con el objetivo de poder obtener diferencias y comparar ambos análisis para llegar a una mejor comprensión del problema.

Un primer caso de estudio será dirigido a los accidentes de tráfico ocurridos con colisiones entre vehículos en zonas urbanas y vías urbanas. Y un segundo caso de estudio en donde será todo igual salvo que esta vez se referirá a los atropellos y no a las colisiones entre vehículos.

Este segundo estudio creo que puede ser bastante útil, ya que podremos discernir entre dos tipos de accidentes bastante comunes como son las colisiones entre vehículos y los atropellos.

Por último decir que este estudio está mas centrado en el algoritmo Ward (Aglomerativo), de tal forma que se ha intentado buscar un caso de estudio en el cuál se pudiera obtener un dendograma bastante comprensible y una gráfica con un Dendograma junto a un Heatmap lo más pequeña posible, ya que nos era imposible en otros casos de estudio.

3.1. Caso de Estudio: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 30.705 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (m)
K-means	91597.534879	4	0.869222	0.096982
MiniBatchKMeans	91587.442389	4	0.868027	0.052714
Birch	17138.753102	4	0.758159	0.723934
DBSCAN	5187.538445	12	0.590084	10.303498
Ward	0.000000	20	0.000000	40.367393
MeanShift	59297.938433	29	0.917540	567.177918
Spectral	35955.028158	4	0.831953	5095.648292

Tabla 3.1: Datos generales asociados a cada uno de los algoritmos, caso de estudio 2 con colisiones entre vehículos.

En este caso, vemos como para el índice **Calinski-Harabasz** el mejor algoritmo es K-means (con 4 clusters), aunque prácticamente empatado con el algoritmo MiniBAtchK-means, esto es bastante normal, ya que el algoritmo MiniBatchKmeans es una modificación del algoritmo K-means, por lo tanto los resultados serán parecidos casi siempre.
Dado que tienen las mejores métricas CH, podemos decir que son los mejores algoritmos para este caso de estudio de todos los usados en cuanto a maximizar la similaridad intra-cluster y minimizar la similaridad inter-cluster. Estos algoritmos se encuentran por delante del algoritmo Mean Shift por bastante.

Para el índice **Silhouette** sin embargo, tenemos al algoritmo Mean Shift por muy poco por delante de K-means y MiniBatchKmean, luego podemos aclarar que se está usando para Mean Shift un número de clusters bastante certero.

Por último cabe decir que utilizando el tiempo en segundos como unidad de medida de lo que ha tardado cada algoritmo en ejecutarse, podemos afirmar que el algoritmo Spectral sería el que myor tiempo ha estado procesándose, además con mucha diferencia sobre el resto, cosa que juega en contra, y es que este algoritmo hace uso del espectro de la matriz de los datos para realizar la reducción dimensional, antes de la agrupación en un menor número de dimensiones, y este trabajo es bueno para un número pequeño de clusters, pero no es recomendado para un número alto. Además el Spectral es interesante para trabajar con imágenes.

En relación al peor algoritmo para este caso de estudio sería el DBSCAN, ya que tiene

unos índices muy por debajo de la media de los demás.

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio con colisiones entre vehículos en zonas y vías urbanas , y al final un breve análisis.

3.1.1. Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
e los 4 clusters hay 4 con más de 3 elementos. Del total de 30705 elementos, se seleccionan 30.705.
La siguiente gráfica 3.1 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

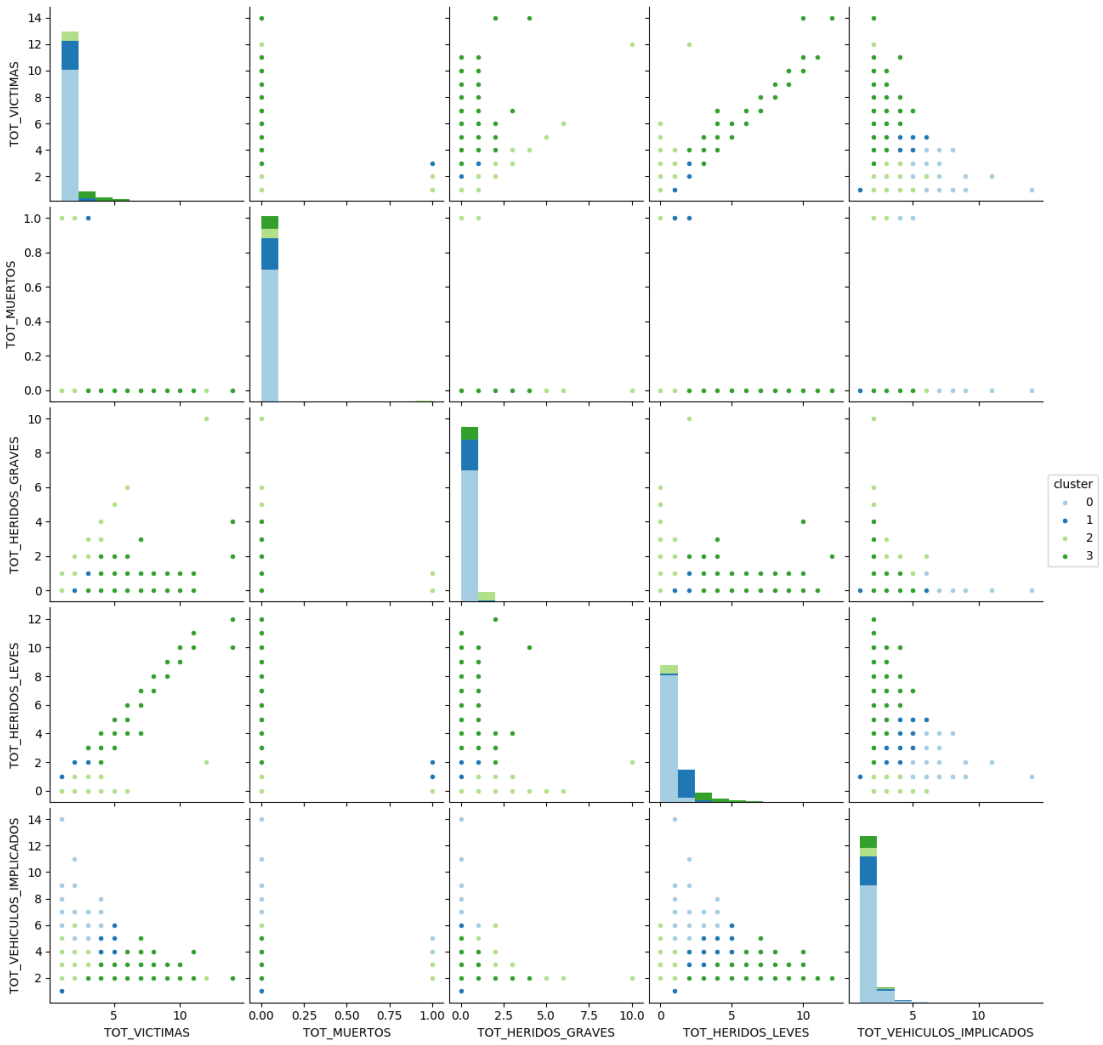

Figura 3.1: Scatter Matrix usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos.

La siguiente gráfica (Heatmap) 3.2 representa a la tabla 3.2 pero con sus datos normalizados:

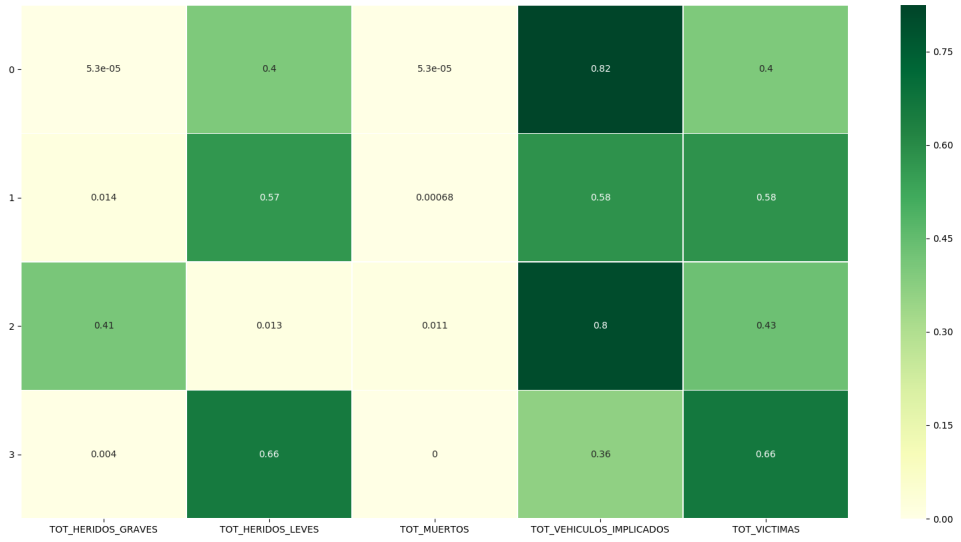

Figura 3.2: Heatmap usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos.

La siguiente tabla 3.2 está compuesta por los datos en media referentes al algoritmo K-means con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.000138	1.038513	0.000138	2.139821	1.038788
1	0.050238	2.051380	0.002474	2.104472	2.104091
2	1.071663	0.033531	0.028928	2.095989	1.134122
3	0.022854	3.790578	0.000000	2.101213	3.813433

Tabla 3.2: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos.

3.1.2. Resultados algoritmo MiniBatchKmeans, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 30.705 elementos, se seleccionan 30.705.
La siguiente gráfica 3.3 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

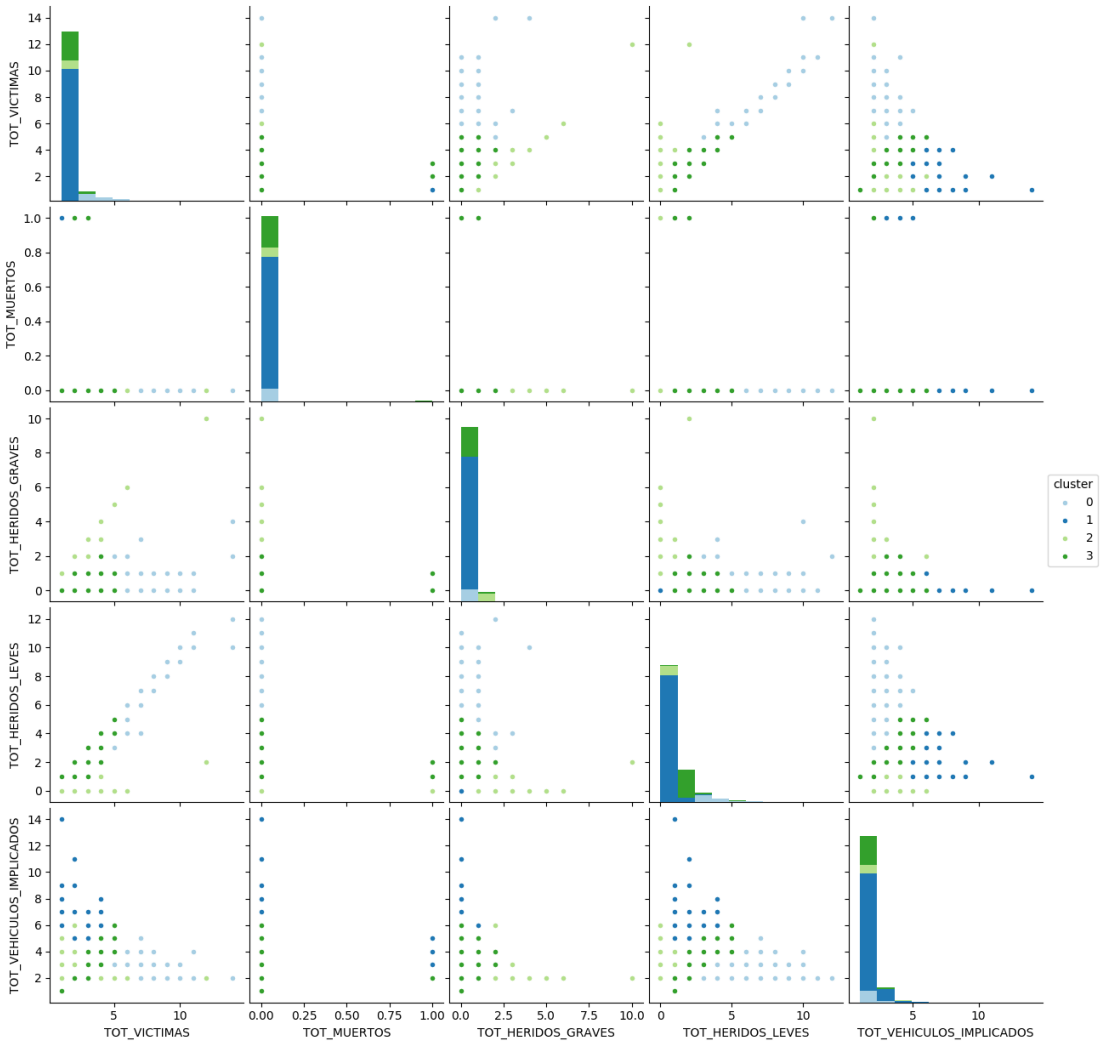

Figura 3.3: Scatter Matrix usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos..

La siguiente gráfica (Heatmap) 3.4 representa los datos de la tabla 3.3 pero con sus datos normalizados:

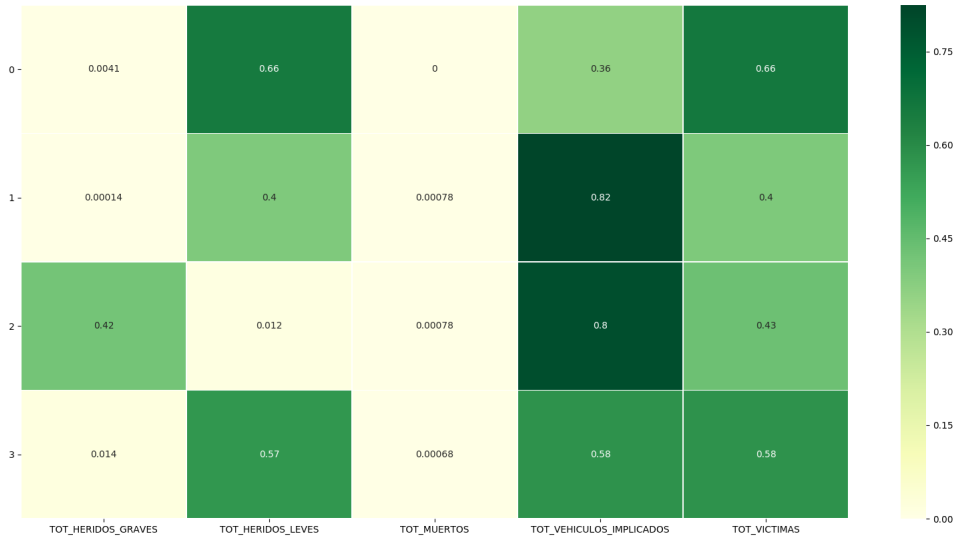

Figura 3.4: Heatmap usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos..

La siguiente tabla 3.3 está compuesta por los datos en media referentes al algoritmo MiniBatchKmeans con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.023776	3.790210	0.000000	2.101632	3.813986
1	0.000366	1.036554	0.002015	2.140122	1.038935
2	1.101695	0.031186	0.002034	2.090169	1.134915
3	0.049867	2.051199	0.002474	2.104301	2.103540

Tabla 3.3: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo MiniBatchKmeans, caso de estudio 2 con colisiones entre vehículos.

3.1.3. Resultados algoritmo Ward, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

El método **Ward** es un procedimiento jerárquico en el cual, en cada etapa, se unen los dos clusters para los cuales se tenga el menor incremento en el valor total de la suma de los cuadrados de las diferencias, dentro de cada cluster, de cada individuo al centroide del cluster. [15]

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 20 clusters hay 20 con más de 3 elementos. Del total de 30.705 elementos, se seleccionan 30.705.
Para la extracción de esta gráfica 3.5se ha utilizado el siguiente fragmento de código 6:

Listing 6: Sección de código en Python para la generación de la gráfica Dendograma junto al Heatmap.

```
1  def PintarHeatmapConDendograma(DFMediasNormal, dendograma_dir,
2  nombreAlgoritmo, casoEstudio, clusters_restantes):
3  plt.figure()
4
5  linkage_array = hierarchy.ward(DFMediasNormal)
6
7
8  hierarchy.dendrogram(linkage_array, orientation='left')
9
10 sns.clustermap(DFMediasNormal, method='ward',
11 col_cluster=False, figsize=(20, 10),
12 yticklabels=clusters_restantes, linewidths=0.5, cmap='YlGn')
13
14 plt.savefig(dendograma_dir + nombreAlgoritmo + casoEstudio)
15 plt.close()
```

La siguiente gráfica 3.5 está compuesta por un Dendograma [9] y un Heatmap [13], de tal forma que podemos ver de forma bastante sencilla los clusters más vinculados y por que variables son más cercanos unos de otros:

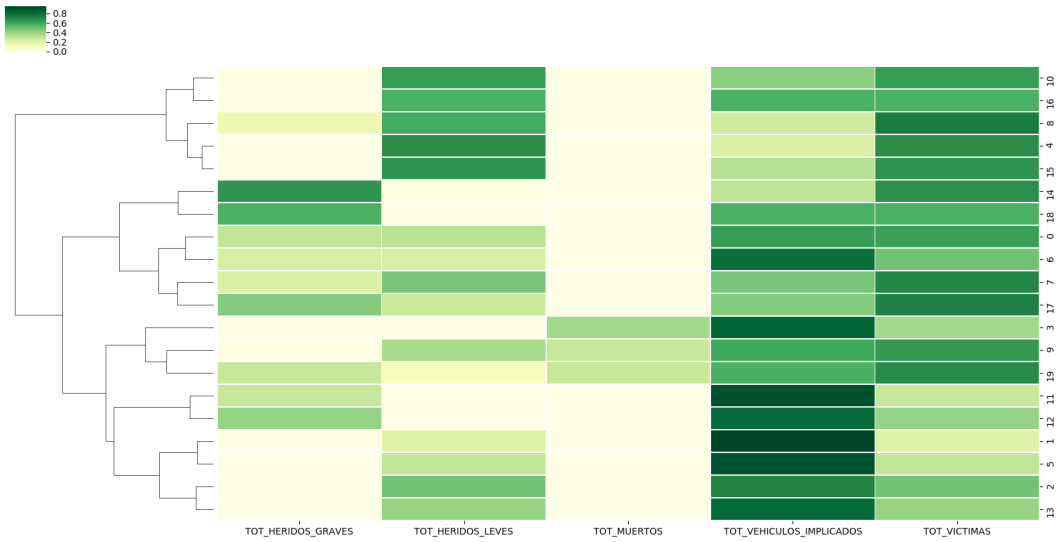

Figura 3.5: Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos.

La siguiente tabla 3.4 está compuesta por los datos en media referentes al algoritmo Ward con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	1.032864	1.117371	0.0	2.178404	2.150235
1	0.000000	1.012346	0.0	4.539095	1.012346
2	0.000000	2.162088	0.0	3.193681	2.162088
3	0.000000	0.000000	1.0	2.186047	1.000000
4	0.000000	5.884735	0.0	2.040498	5.884735
5	0.000000	1.025751	0.0	3.052217	1.025751
6	1.000000	1.000000	0.0	3.235294	2.000000
7	1.020408	2.102041	0.0	2.102041	3.122449
8	1.228571	4.771429	0.0	2.142857	6.000000
9	0.000000	1.250000	1.0	2.083333	2.250000
10	0.000000	3.200000	0.0	2.160440	3.200000
11	1.013514	0.000000	0.0	3.202703	1.013514
12	1.003956	0.000000	0.0	2.006329	1.003956
13	0.000000	1.006210	0.0	2.012419	1.006210
14	4.187500	0.125000	0.0	2.000000	4.312500
15	0.000000	4.082005	0.0	2.041002	4.082005
16	0.000000	2.063395	0.0	2.063395	2.063395
17	2.125000	1.312500	0.0	2.125000	3.437500
18	2.013514	0.000000	0.0	2.013514	2.013514
19	1.000000	0.400000	1.0	2.000000	2.400000

Tabla 3.4: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos.

3.1.4. Resultados algoritmo Mean Shift, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 29 clusters hay 21 con más de 3 elementos. Del total de 30.705 elementos, se seleccionan 30.687.
La siguiente gráfica 3.6 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

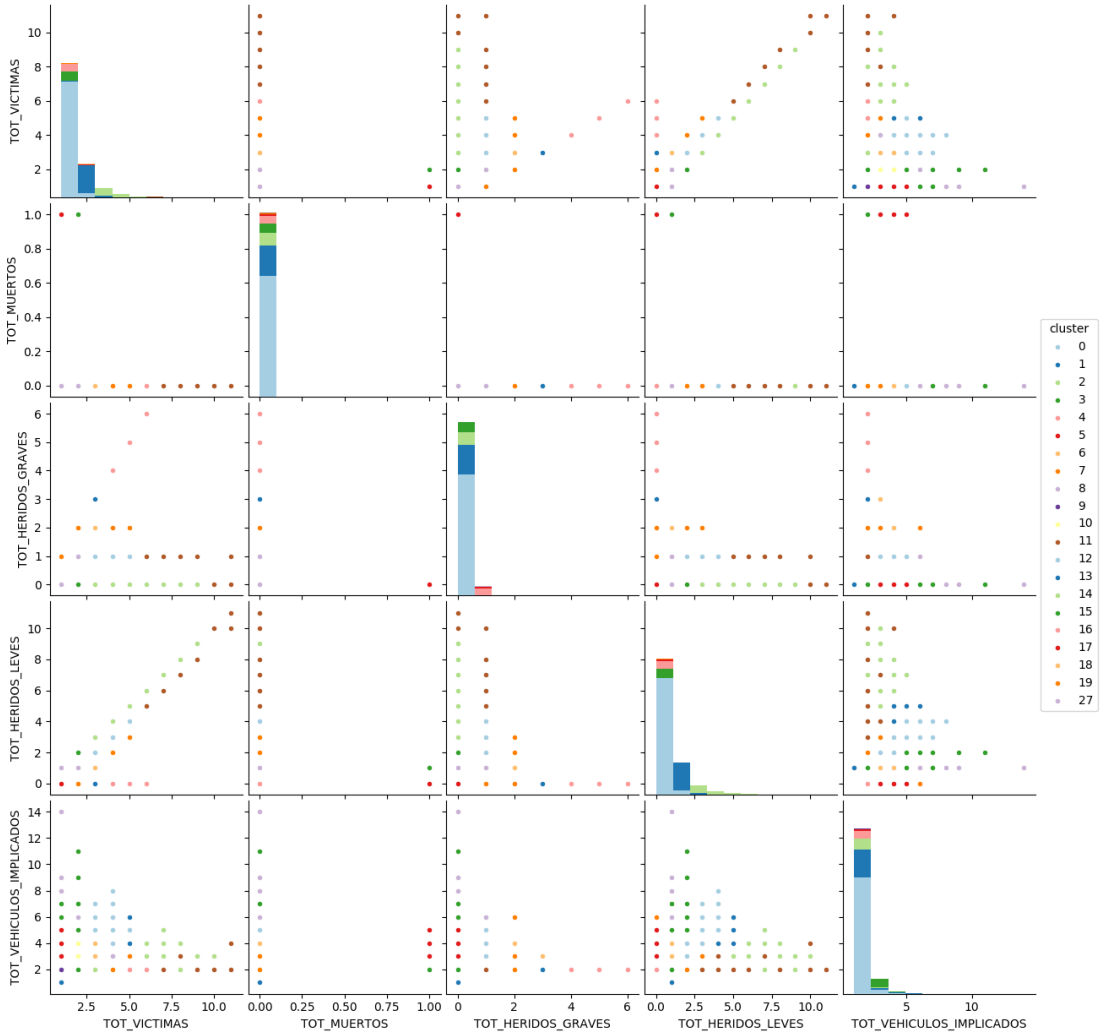

Figura 3.6: Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 2 con colisiones entre vehículos.

En este algoritmo no se ha considerado la extracción de un heatmap y de la tabla de datos de las medias debido a la gran cantidad de clusters que propone el propio algoritmo. Por ello y dado que no nos sería de mucha utilidad, se ha decidido obviar dicha gráfica y tabla para este algoritmo.

3.1.5. Interpretación de la segmentación: Accidentes de tráfico en zonas urbanas y vías urbanas con colisión entre vehículos .

Como interpretación final de los resultados de este caso de estudio, se intentará analizar los resultados del algoritmo Ward, en concreto el Dendograma junto con el Heatmap, ya que es bastante explícito.

Para la interpretación tenemos la gráfica 3.5 que nos servirá de apoyo para explicar un poco como están distribuidos los datos en los diferentes clusters. En la gráfica 3.5 se aprecia en el eje Y por la izquierda el dendograma, y por la derecha los clusters, de tal forma que podemos ver como se van agrupando los clusters y que clusters están más cercanos a otros. Por otro lado en el eje X tenemos las variables de siempre.

Así podemos ver algo bastante importante, que es como quedarían distribuidos los clusters si cortáramos el dendograma, por ejemplo si dijéramos de tener un total de 4 clusters, los clusters formados actualmente estarían divididos en los siguientes mega-clusters : [14]

- Cluster 1: 10, 16, 8, 4 y 15.
- Cluster 2: 14, 18, 0, 6, 7 y 17.
- Cluster 3: 3, 9 y 19.
- Cluster 4: 1, 5, 2 y 13.

Podemos ver por ejemplo como los clusters 10 y 16 son prácticamente el mismo, teniendo en la tabla de medias, como mayor diferencia el número medio de víctimas.

Lo mismo ocurre con los clusters 0 y 6, en donde la mayor diferencia reside en el promedio de vehículos implicados.

Sin embargo si nos vamos a comparar los clusters 2 y 3, vemos que en el dendograma se encuentran muy alejados, y no es casualidad, ya que por ejemplo en promedio el cluster 3 contiene datos con 1 muerto mientras el cluster 2 en promedio contiene aquellos datos con 0 muertos. Este hecho, es bastante identificativo a la hora de mostrar notables diferencias, pero no queda todo ahí ya que si miramos el número medio de víctimas, el cluster 2 duplica al 3.

Igual ocurre con los clusters 8 y 9, en donde residen entre ambos grandes diferencias, como por ejemplo el número en promedio de heridos leves en los accidentes.

3.2. Caso de Estudio: Accidentes de tráfico en zonas urbanas y vías urbanas con atropellos.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 10.125 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (m)
K-means	54271.478949	4	0.906234	0.039048
MiniBatchKMeans	54279.862211	4	0.906879	0.025384
Birch	26577.324076	4	0.827940	0.304586
Ward	0.000000	20	0.000000	2.646923
DBSCAN	100399.673812	12	0.980360	1.314043
MeanShift	332870.212489	22	0.985180	26.502399
Spectral	53426.777692	4	0.907196	30.402875

Tabla 3.5: Datos generales asociados a cada uno de los algoritmos, caso de estudio 2 con atropellos.

En este caso, vemos como para el índice **Calinski-Harabasz** el mejor algoritmo es Mean Shift, con muchísima diferencia. Como dato curioso mencionar que el algoritmo DBSCAN en este caso de estudio se encuentra en un gran tercer puesto por detrás del algoritmo Birch.

Para el índice **Silhouette** tenemos a los algoritmos Mean Shift y DBSCAN muy pegados, con diferencias ínfimas, mientras que los demás también se encuentran cercanos a ellos.

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio con atropellos en zonas y vías urbanas , y al final un breve análisis.

3.2.1. Resultados algoritmo K-means, caso de estudio con atropellos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 10.125 elementos, se seleccionan 10.125.
La siguiente gráfica 3.7 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

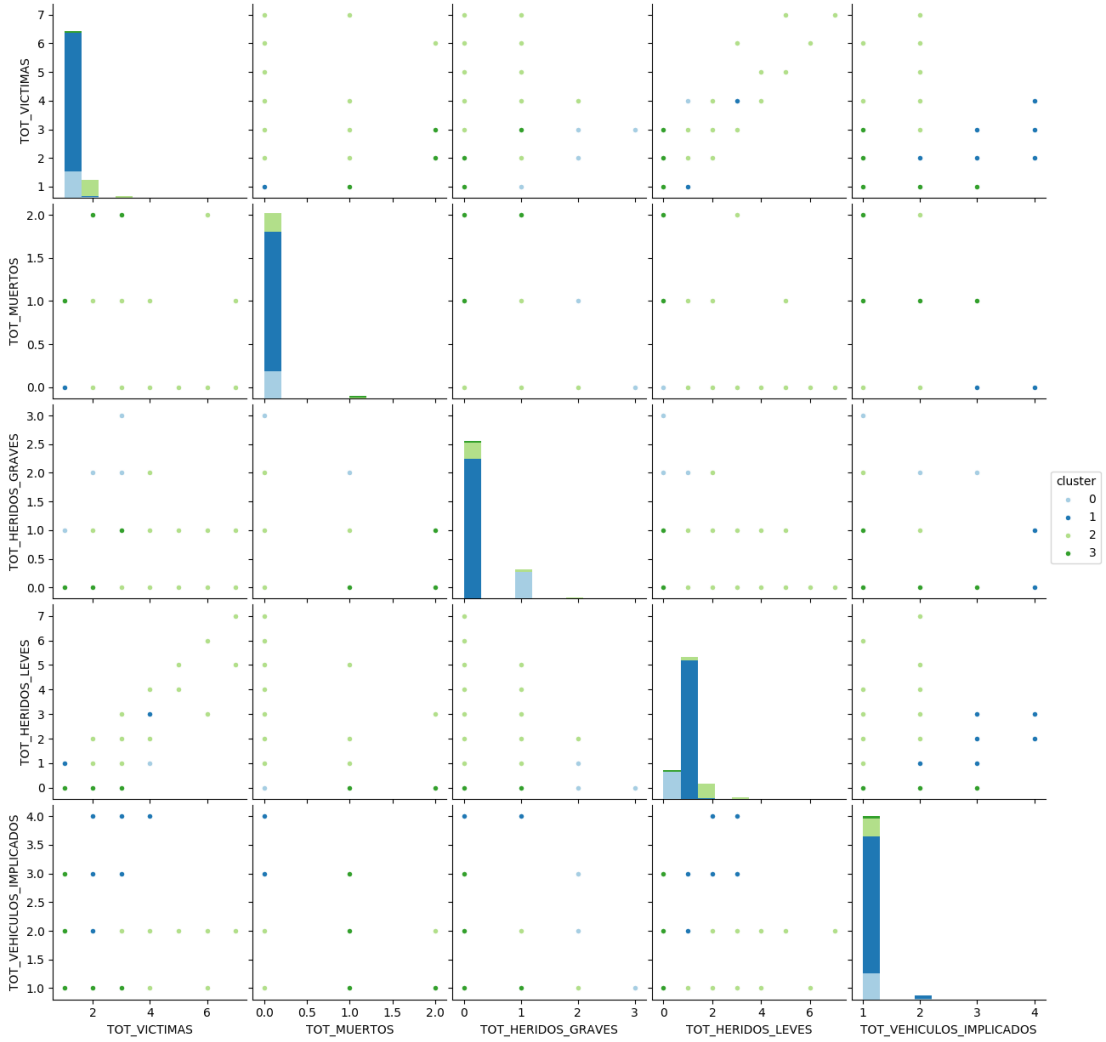

Figura 3.7: Scatter Matrix usando el algoritmo K-means, caso de estudio 2 con atropellos.

La siguiente gráfica (Heatmap) 3.8 representa a la tabla 3.6 pero con sus datos normalizados:

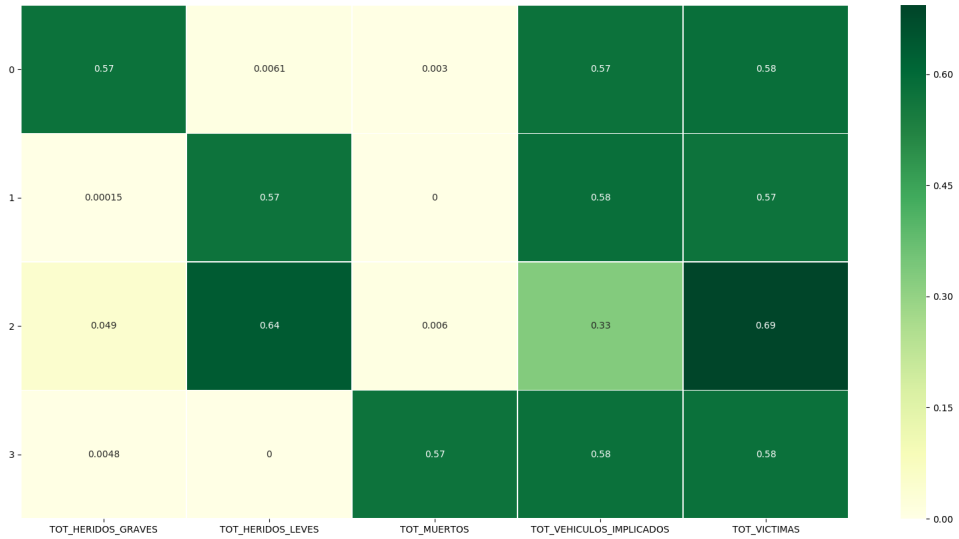

Figura 3.8: Heatmap usando el algoritmo K-means, caso de estudio 2 con atropellos.

La siguiente tabla 3.6 está compuesta por los datos en media referentes al algoritmo K-means con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.001328	1.005977	0.000000	1.026165	1.007305
1	1.030780	0.005472	0.005472	1.025308	1.041724
2	0.150591	1.978346	0.018701	1.020669	2.147638
3	0.008475	0.000000	1.016949	1.025424	1.025424

Tabla 3.6: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 2 con colisiones entre vehículos.

3.2.2. Resultados algoritmo DBSCAN, caso de estudio con atropellos en zonas y vías urbanas.

El algoritmo **DBSCAN** también conocido como **agrupamiento espacial basado en densidad de aplicaciones con ruido** (Density-based spatial clustering of applications with noise) es un algoritmo de agrupamiento basado en densidad, ya que encuentra un

número de clusters comenzando por la distribución de densidad de los correspondientes puntos. Es uno de los algoritmos de clustering más usados. [11]

En segundo lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 12 clusters hay 12 con más de 3 elementos. Del total de 10.125 elementos, se seleccionan 10.125.
La siguiente gráfica 3.9 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

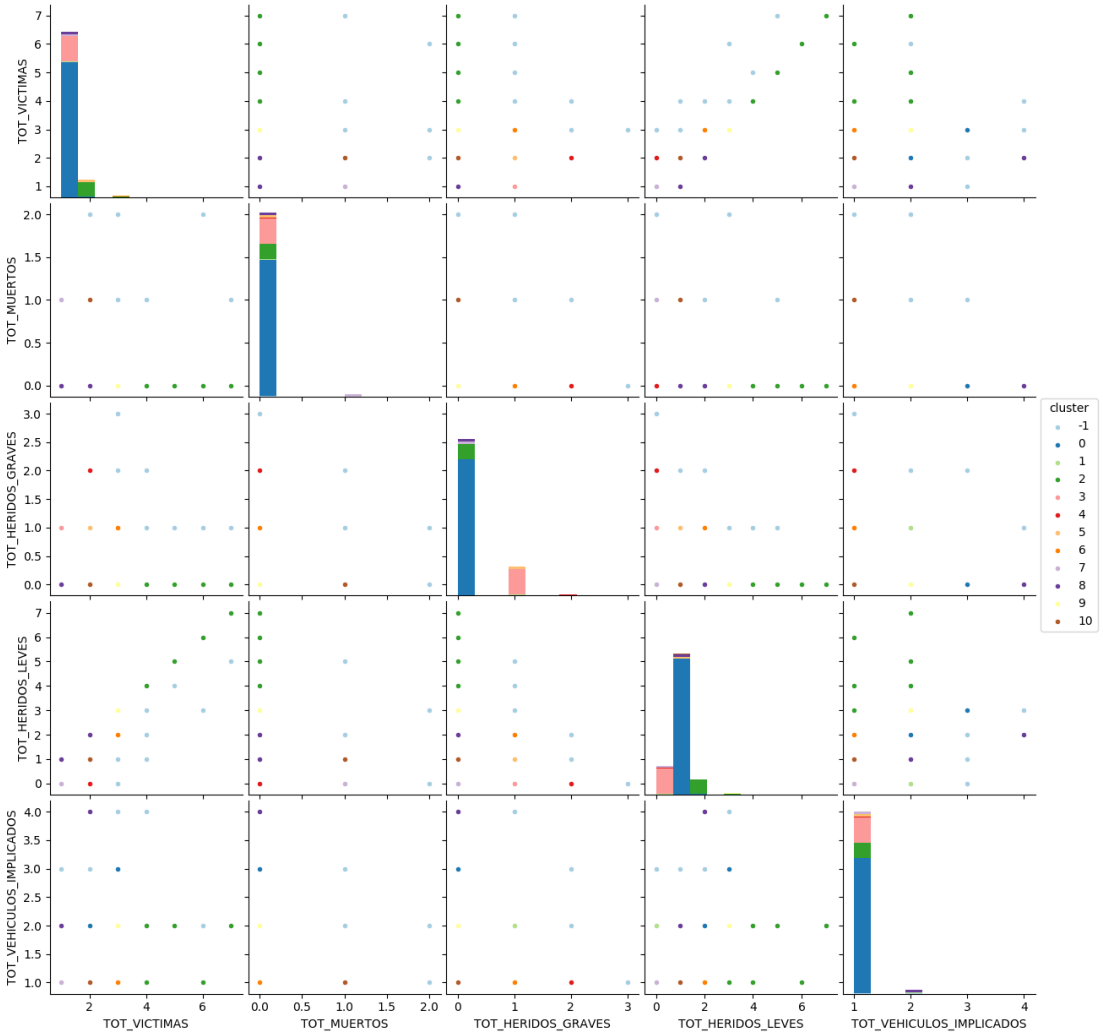

Figura 3.9: Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 2 con atropello.

La siguiente gráfica (Heatmap) 3.10 representa los datos de la tabla 3.7 pero con sus datos normalizados:

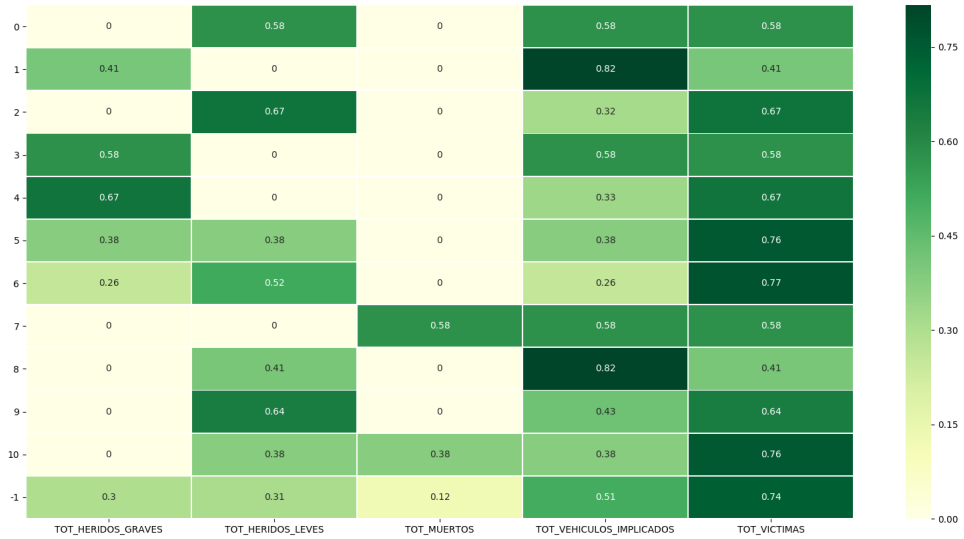

Figura 3.10: Heatmap usando el algoritmo DBSCAN, caso de estudio 2 con atropellos.

La siguiente tabla 3.7 está compuesta por los datos en media referentes al algoritmo DBSCAN con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.000000	1.005282	0.000000	1.005282	1.005282
1	1.000000	0.000000	0.000000	2.000000	1.000000
2	0.000000	2.117717	0.000000	1.005945	2.117717
3	1.000000	0.000000	0.000000	1.000000	1.000000
4	2.000000	0.000000	0.000000	1.000000	2.000000
5	1.000000	1.000000	0.000000	1.000000	2.000000
6	1.000000	2.000000	0.000000	1.000000	3.000000
7	0.000000	0.000000	1.000000	1.000000	1.000000
8	0.000000	1.007692	0.000000	2.015385	1.007692
9	0.000000	3.000000	0.000000	2.000000	3.000000
10	0.000000	1.000000	1.000000	1.000000	2.000000
-1	1.085106	1.127660	0.446809	1.829787	2.659574

Tabla 3.7: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 2 con atropellos.

3.2.3. Resultados algoritmo Ward, caso de estudio con atropellos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 20 clusters hay 17 con más de 3 elementos. Del total de 10.125 elementos, se seleccionan 10.119.
La siguiente gráfica 3.11 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

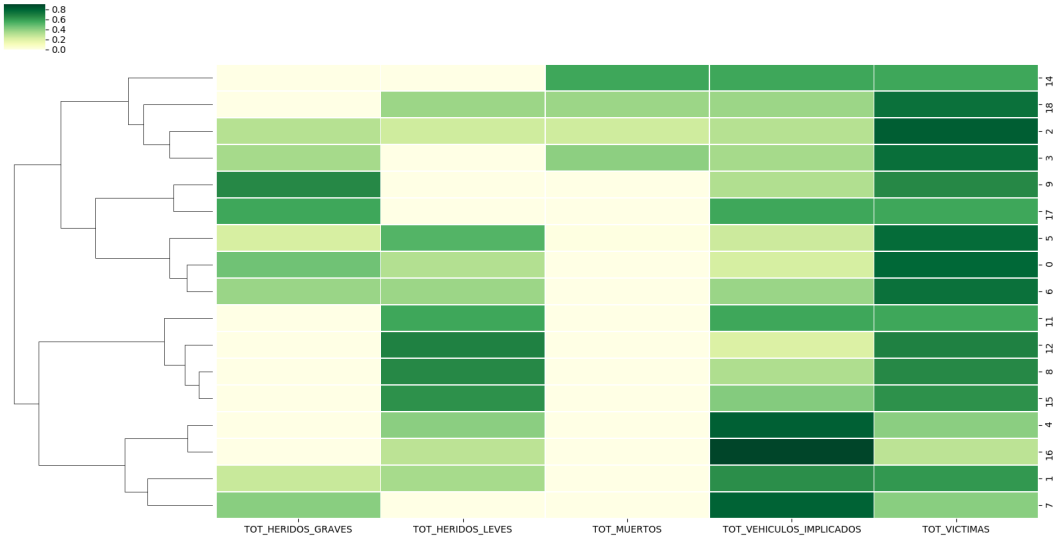

Figura 3.11: Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con atropellos.

Para la extracción de esta gráfica 3.12se ha utilizado el siguiente fragmento de código 7:

Listing 7: Sección de código en Python para la generación de la gráfica Dendograma solo.

```
def PintarDendograma(DFMediasNormal, dendograma_dir,
nombreAlgoritmo, casoEstudio):

linkage_array = hierarchy.ward(DFMediasNormal)
plt.figure()
plt.clf()
hierarchy.dendrogram(linkage_array, orientation='left')

plt.savefig(dendograma_dir + nombreAlgoritmo +
"DendogramaSolo" + casoEstudio)
plt.close()
```

La siguiente gráfica 3.12 detalla de forma más ampliada el Dendograma separado:

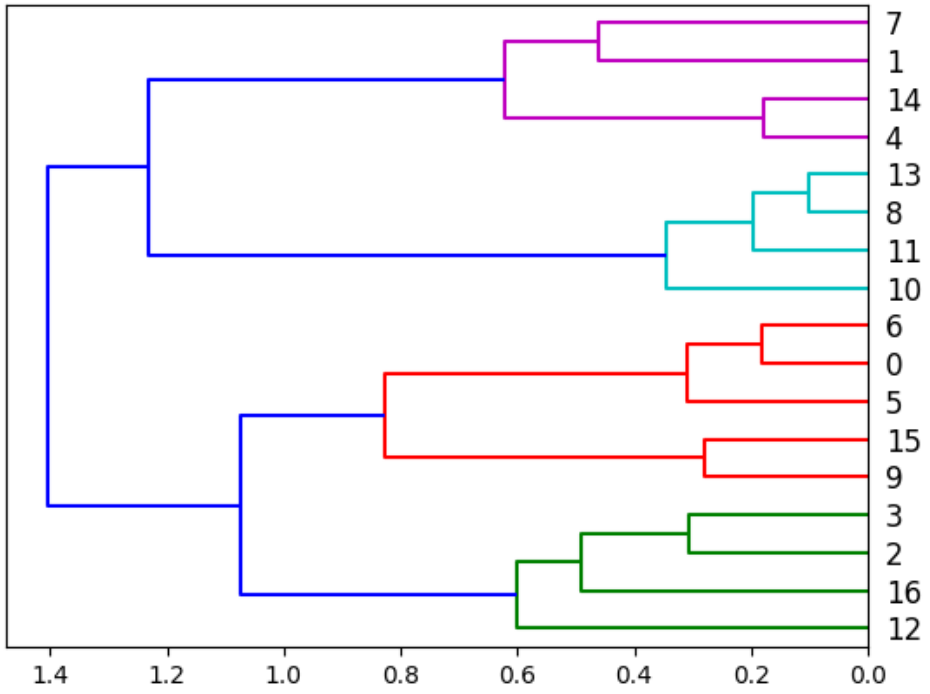

Figura 3.12: Dendograma junto a Heatmap usando el algoritmo Ward, caso de estudio 2 con colisiones entre vehículos.

Hay que aclarar que el número de cada cluster en la gráfica referente al dendograma en solitario 3.12 no se corresponde con los de la tabla o los de la gráfica 3.11, ya que la función que nos permite pintar la gráfica con el Dendograma en solitario no nos permite indicarle el nombre de cada cluster, al contrario que para la gráfica con el Dendograma y el Heatmap juntos, como se puede apreciar en 6. Para solucionar los problemas de un mal entendimiento de esta última gráfica, se recomienda visualizar la imagen al revés, de tal forma que lo que empieza por arriba, se supone que es lo que habría abajo y viceversa. Así se puede comparar con la imagen 3.11. Se lamenta el problema ocasionado.

La siguiente tabla 3.8 está compuesta por los datos en media referentes al algoritmo Ward con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	2.000000	1.375000	0.000000	1.000000	3.375000
1	1.000000	1.300000	0.000000	2.400000	2.300000
2	1.250000	1.000000	1.000000	1.250000	3.250000
3	1.000000	0.000000	1.142857	1.000000	2.142857
4	0.000000	1.022727	0.000000	2.030303	1.022727
5	1.000000	2.277778	0.055556	1.111111	3.333333
6	1.007937	1.000000	0.000000	1.007937	2.007937
7	1.060606	0.000000	0.000000	2.060606	1.060606
8	0.000000	2.007823	0.000000	1.003911	2.007823
9	2.028571	0.000000	0.000000	1.000000	2.028571
11	0.000000	1.005282	0.000000	1.005282	1.005282
12	0.000000	3.256757	0.000000	1.027027	3.256757
14	0.000000	0.000000	1.000000	1.000000	1.000000
15	0.000000	3.000000	0.000000	2.000000	3.000000
16	0.000000	1.000000	0.000000	3.000000	1.000000
17	1.000000	0.000000	0.000000	1.000000	1.000000
18	0.000000	1.000000	1.000000	1.000000	2.000000

Tabla 3.8: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Ward, caso de estudio 2 con atropellos.

3.2.4. Resultados algoritmo Mean Shift, caso de estudio con atropellos en zonas y vías urbanas.

Tras realizar la eliminación de los outliers, se ha producido las siguientes gráficas, filtrado realizado a los clusters con menos de 3 elementos:
De los 22 clusters hay 15 con más de 3 elementos. Del total de 10.125 elementos, se seleccionan 10.117.
La siguiente gráfica 3.13 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

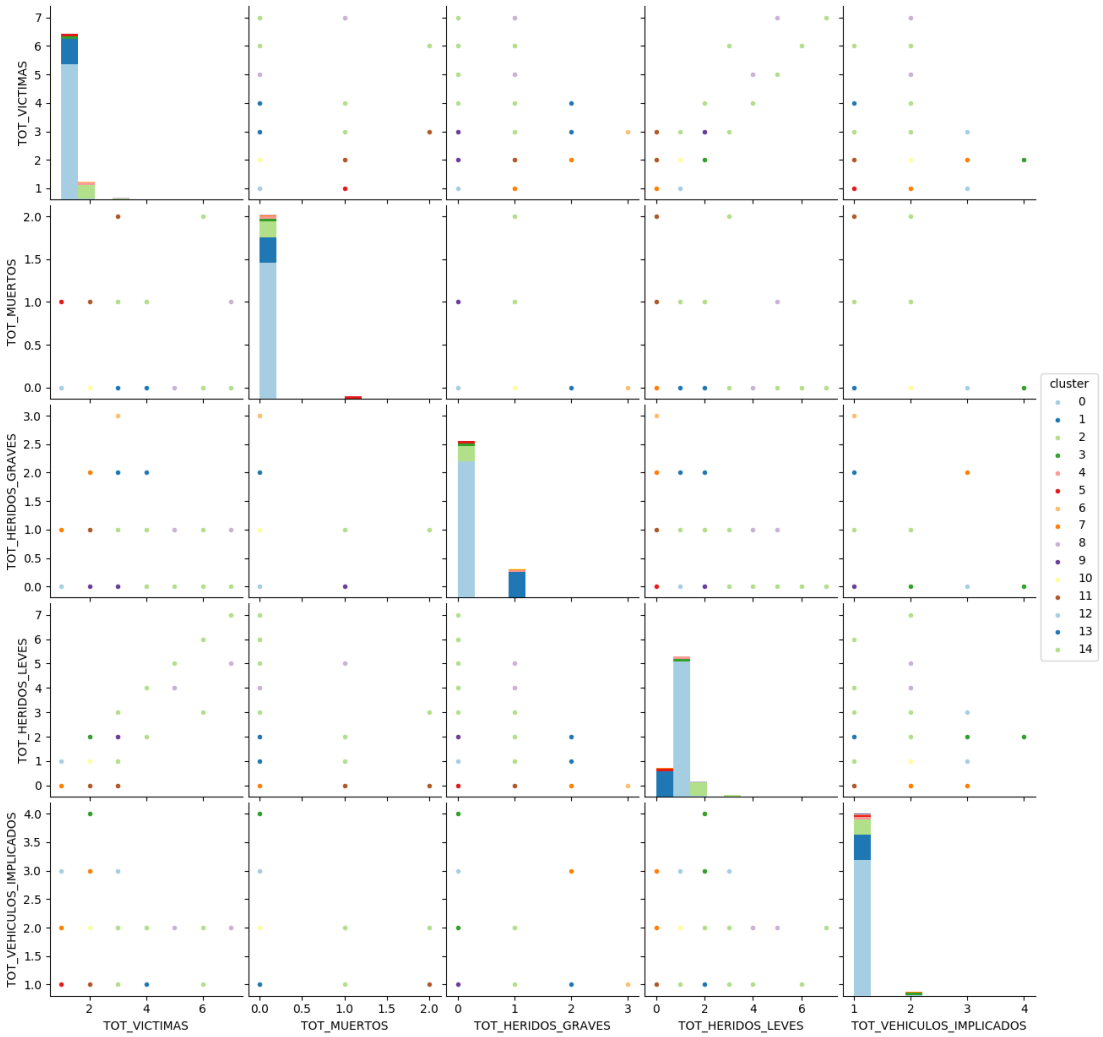

Figura 3.13: Scatter Matrix usando el algoritmo Mean Shift, caso de estudio 2 con atropellos.

En este algoritmo no se ha considerado la extracción de un heatmap y de la tabla de datos de las medias debido a la gran cantidad de clusters que propone el propio algoritmo. Por ello y dado que no nos sería de mucha utilidad, se ha decidido obviar dicha gráfica y tabla para este algoritmo.

3.2.5. Interpretación de la segmentación: Accidentes de tráfico en zonas urbanas y vías urbanas con atropellos .

Como interpretación final de los resultados de este caso de estudio, se intentará analizar los resultados del algoritmo Ward, en concreto el Dendograma junto con el Heatmap, ya que es bastante explícito.

Para este caso de estudio de nuevo intentaremos fijarnos mayormente en el algoritmo Ward jerárquico, y poder ver como agrupa a los distintos datos. Para este caso contamos de nuevo con una gráfica Dendograma junto a un Heatmap, y aparte con un Dendograma en solitario, de tal forma que podemos comparar ambas gráficas y hablar sobre ellas.

En primer lugar comentaremos como se relacionan los clusters, ya que en este caso vemos dos tipos de clsuters muy marcados, por un lado, tenemos los clusters que agrupan a los accidentes donde más vehículos implicados hay y menos muertes se producen (clusters 7, 1, 16 y 4 principalmente), y por otro lado tenemos a los clusters con accidentes donde se producen más muertes y menos vehículos implicados hay (14, 18, 2, 3 principalmente). También a su vez vemos como hay una relación entre el total de víctimas y el total de vehículos implicados sobre el dendograma, ya que los clusters con menos víctima producidas y más vehículos implicados están más relacionadas, y por otro lado tienen otra relación diferente los clusters con más víctimas producidas y menos vehículos implicados.

Después si queremos comprobar la salida a priori que nos daría el tener por ejemplo 4 clusters podemos fijarnos en la gráfica 3.12, de tal forma que si cortamos la figura de forma vertical nos saldría lo siguiente: [14]

- Cluster 1: Compuesto por los datos de los clusters 7, 1, 14 y 4.
- Cluster 2: Compuesto por los datos de los clusters 13, 8, 11 y 10.
- Cluster 3: Compuesto por los datos de los clusters 6, 0, 5, 15 y 9.
- Cluster 4: Compuesto por los datos de los clusters 3, 2, 16 y 12.

Fijándonos ya de una manera más independiente vemos como los clusters 4 y 16 son prácticamente el mismo, ya que reunen a datos con similares características, accidentes con un valor alto de vehículos implicados, un número de víctimas medio, número bajo

de muertos, un número medio-bajo de heridos leves y un valor bajo de heridos graves.

No se puede decir lo mismo con los clusters 2 y 7, que están en lados opuestos del dendograma, y es que mientras que el cluster 2 reúne a datos en donde el número de muertos es medio, el cluster 7 reúne a datos donde es prácticamente 0 el número de muertos. También hay entre ambos grandes diferencias en los heridos leves como ya se ha comentado anteriormente.

4. Caso de estudio 3: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave y con vurva fuerte.

En este tercer y último punto de estudio se van a exponer tres casos de estudio parecidos con el objetivo de poder visualizar mediante gráficas o tablas con métricas las diferencias y similaridades entre los dos casos. En el primer caso tendremos los accidentes con colisiones entre vehículos en trazado en trazado con curva suave y en el segundo caso tendremos los accidentes con colisiones entre vehículos en trazado con curva fuerte.

He intentado poner énfasis en este estudio porque me parece curioso ver como están distribuidos los datos en estos tres casos y hacer un análisis de lo que podría ser algo habitual por ejemplo para intentar evitar los accidentes más mortales de entre los tres.

Al final de la sección también podremos encontrar un análisis de algunos de los algoritmos con modificaciones en los parámetros de entrada con respecto a estos análisis principales.

En el siguiente apartado se mostrarán los resultados y el análisis final con respecto al caso de estudio con colisiones entre vehículos en trazado con curva suave.

4.1. Caso de Estudio: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio, datos como el número de clusters que se han utilizado, la métrica

Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 2.542 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	3511.140887	4	0.719879	0.018582
MiniBatchKMeans	3432.975045	4	0.718814	0.017018
Birch	1158.484252	4	0.579408	0.052481
DBSCAN	1401.561701	16	0.689134	0.064199
MeanShift	1684.801559	22	0.660505	0.659362
Spectral	3087.418304	4	0.720713	0.985690

Tabla 4.1: Datos generales asociados a cada uno de los algoritmos, caso de estudio 3 en trazado con curva suave.

Con respecto a las métricas concentradas en la tabla 4.1 podemos argumentar lo siguiente:

El algoritmo K-means tiene el mejor comportamiento atendiendo al conjunto de las métricas, ya que por un lado tiene el mejor índice CH y por otro lado tiene el segundo mejor índice SC por detrás del algoritmo Spectral. Lo cuál nos indica que el número de clusters (en este caso 4) elegido para este algoritmo funciona no de manera excepcional, pero si para colocarse en una gran posición con respecto a sus competidores. En cuanto a tiempo, no es el peor de todos.

Podemos también decir que el algoritmo Spectral con los 4 clusters como parámetro de entrada se comporta de forma correcta. Si que es verdad que se ve penalizado a la hora de comparar los tiempos de computación requeridos para cada algoritmo, pues tenemos en el Spectral un algoritmo costoso en demasía.

Para el comportamiento, tendríamos al Birch, ya que tanto en las métricas CH como SC tiene los peores valores de los 6 algoritmos comparados.

Y por último cabe decir que el algoritmo MiniBatchKmeans calca prácticamente los

resultados en lo que a métricas se refiere del algoritmo K-means, esto se debe, como ya se explicó anteriormente, a que el algoritmo MiniBatchKmeans es una variante del K-means, por lo que no debería suscitar dudas de coportamientos anómalos.

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio con con colisiones entre vehículos en trazado con curva suave, y al final un breve análisis.

4.1.1. Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en trazado con curva suave.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 2.542 elementos, se seleccionan 2.542.
La siguiente gráfica 4.1 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

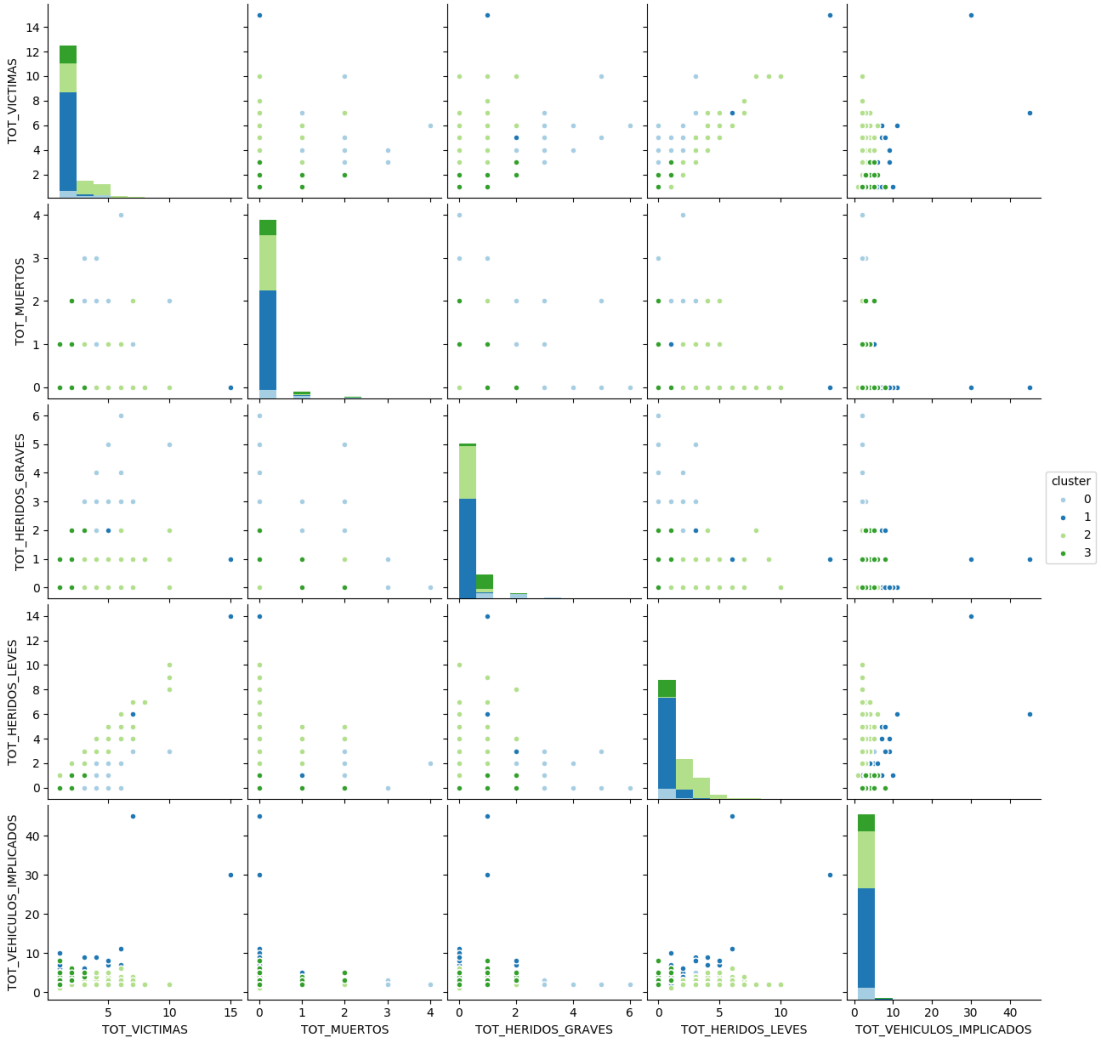

Figura 4.1: Scatter Matrix usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave.

La siguiente gráfica (Heatmap) 4.2 representa a la tabla 4.2 pero con sus datos normalizados:

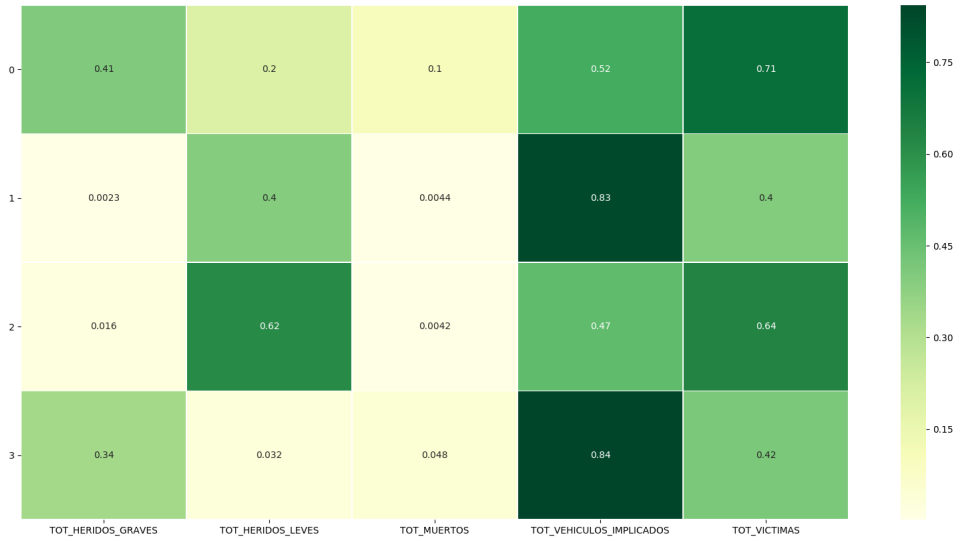

Figura 4.2: Heatmap usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave.

La siguiente tabla 4.2 está compuesta por los datos en media referentes al algoritmo K-means con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	1.640244	0.823171	0.420732	2.115854	2.884146
1	0.006584	1.119239	0.012436	2.334309	1.138259
2	0.071336	2.822309	0.019455	2.140078	2.913100
3	0.933333	0.087500	0.133333	2.341667	1.154167

Tabla 4.2: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 3 en trazado con curva suave.

4.1.2. Resultados algoritmo DBSCAN, caso de estudio de colisiones entre vehículos en trazado con curva suave.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:

De los 16 clusters hay 16 con más de 3 elementos. Del total de 2.542 elementos, se seleccionan 2.542.

La siguiente gráfica 4.3 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

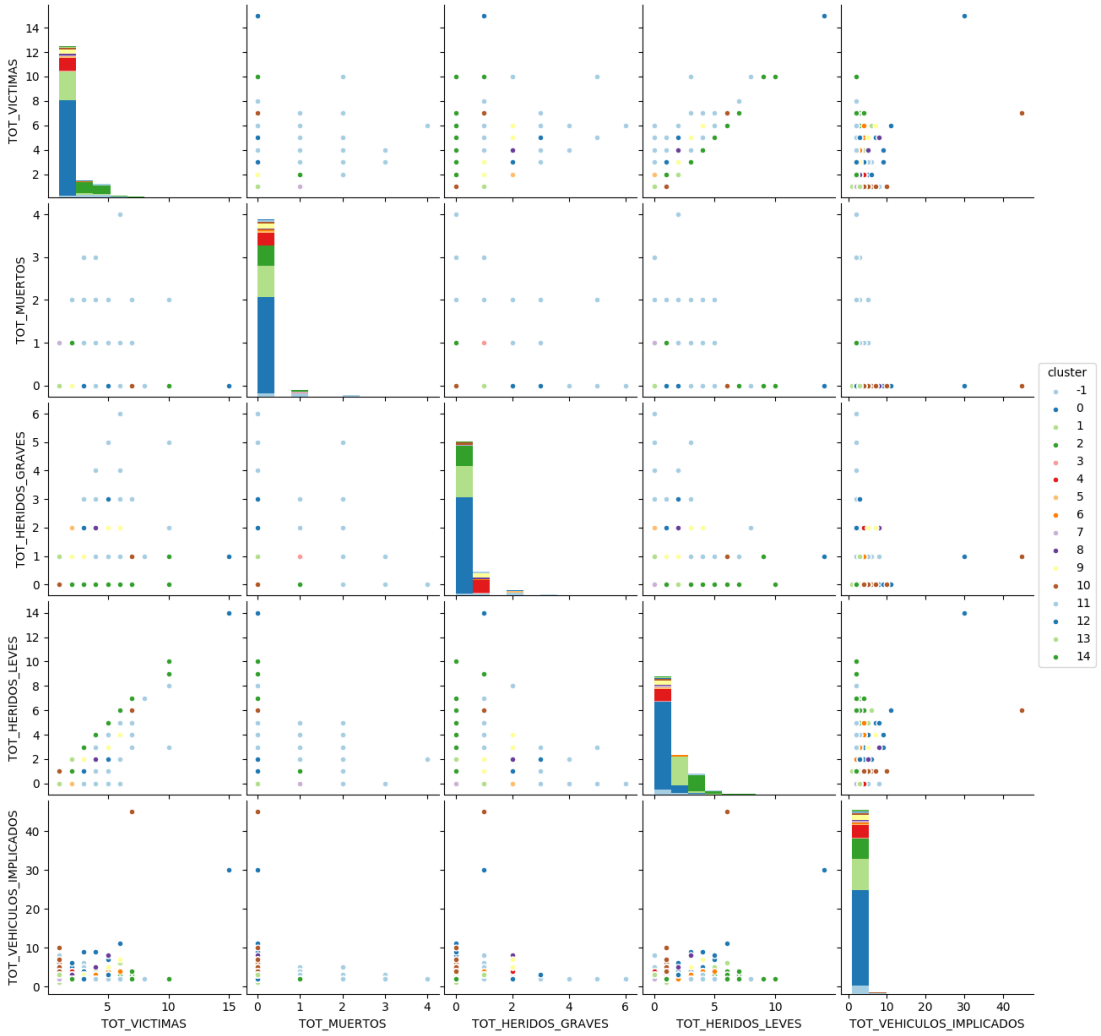

Figura 4.3: Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave.

La siguiente gráfica (Heatmap) 4.4 representa los datos de la tabla 4.3 pero con sus datos normalizados:

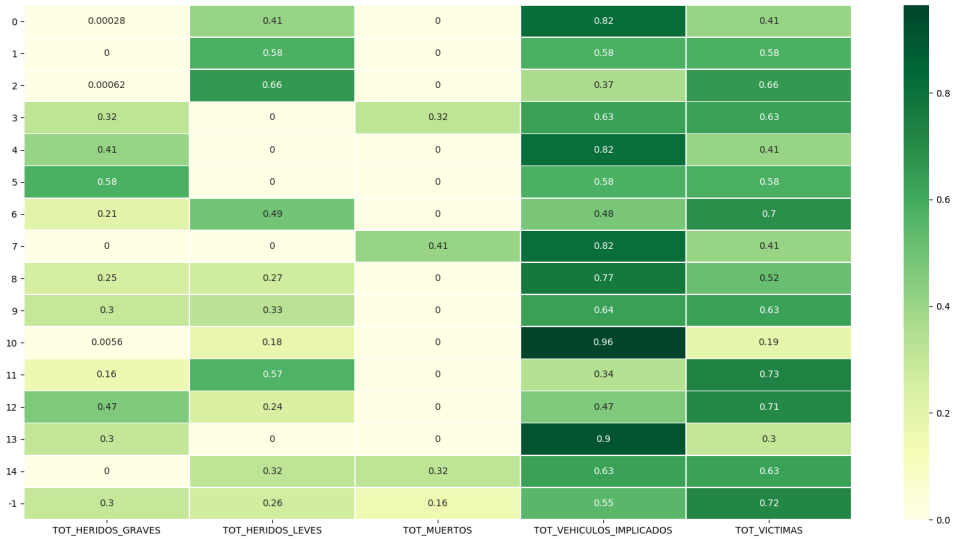

Figura 4.4: Heatmap usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave.

La siguiente tabla 4.3 está compuesta por los datos en media referentes al algoritmo DBSCAN con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.023776	3.790210	0.000000	2.101632	3.813986
1	0.000366	1.036554	0.002015	2.140122	1.038935
2	1.101695	0.031186	0.002034	2.090169	1.134915
3	0.049867	2.051199	0.002474	2.104301	2.103540

Tabla 4.3: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva suave.

4.1.3. Resultados algoritmo Spectral, caso de estudio de colisiones entre vehículos en trazado con curva suave.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:

De los 4 clusters hay 4 con más de 3 elementos. Del total de 2.542 elementos, se seleccionan 2.542.

La siguiente gráfica 4.5 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

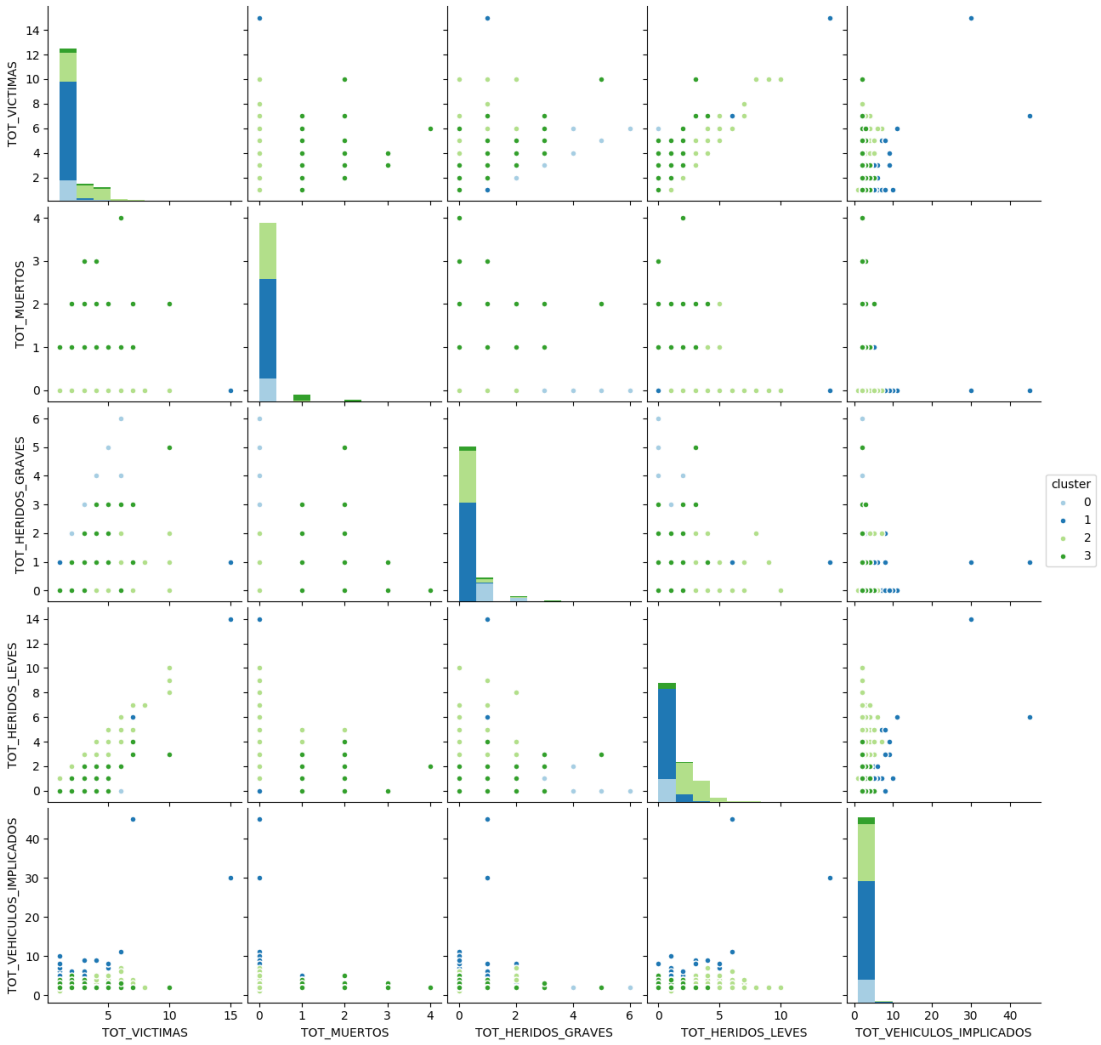

Figura 4.5: Scatter Matrix usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave.

La siguiente gráfica (Heatmap) 4.6 representa los datos de la tabla 4.4 pero con sus datos normalizados:

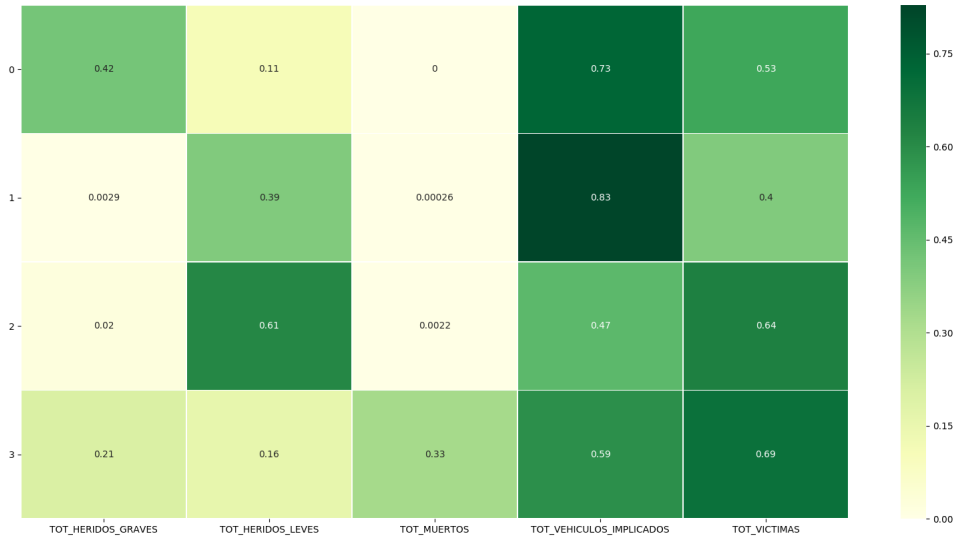

Figura 4.6: Heatmap usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave.

La siguiente tabla 4.4 está compuesta por los datos en media referentes al algoritmo Spectral con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	1.257143	0.326984	0.000000	2.190476	1.584127
1	0.008124	1.116691	0.000739	2.342688	1.125554
2	0.093023	2.824289	0.010336	2.149871	2.927649
3	0.787879	0.616162	1.252525	2.262626	2.656566

Tabla 4.4: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 3 en trazado con curva suave.

4.1.4. Interpretación de la segmentación: Accidentes de tráfico con colisiones entre vehículos en trazado con curva suave.

Para el análisis de el caso de estudio de los accidentes de tráfico con colisiones entre vehículos en trazado con curva suave se hará uso de las distintas gráficas y tablas mostradas anteriormente, de tal manera que podremos crear una base sólida de todo aquello que argumentemos.

En primer lugar y observando la gráfica 4.1 del algoritmo K-means, podemos observar como el cluster 0 agrupa a los accidentes donde más muertos se producen y donde hay pocos vehículos implicados. De hecho agrupa a a datos con un valor alto en cuanto a muertos se refiere el accidente.
Por otro lado tenemos el cluster 1 de la gráfica 4.1 también para el algoritmo K-means, que parece agrupar a los accidentes donde hay un número alto de vehículos implicados pero no se producen muertos. Algo que en principio si nos fijásemos en la variable víctimas sería bastante lógico, ya que este cluster agrupa también aquellos accidentes donde hay un número alto de víctimas y de vehículos implicados. Y digo parece lógico, porque podríamos pensar que si hubiera un número alto de vehículos implicados en un accidente el número de víctimas subiría.
En otro orden de cosas, si nos fijamos para el algoritmo K-means en la gráfica 4.3, podemos ver como por ejemplo el cluster 2 acoge aquellos accidentes en donde el número de heridos leves en promedio es alto con respecto a los demás clusters.

Comparando los resultados del K-means con los del algoritmo Spectral, vemos en la gráfica 4.5 el cluster 3 sería el análogo al cluster 0 en el caso del K-means. Ya que contiene por ejemplo a los accidentes que mas muertos provocan y en donde menos vehículos implicados hay.
También para el algoritmo Spectral tendríamos al cluster 1 como el análogo al cluster 1 en el algoritmo K-means, ya que contiene a los accidentes donde hay un alto número de vehículos implicados y no se producen muertos.
Sin embargo en el algoritmo Spectral, el cluster 0 contiene a los accidentes con un número de heridos graves alto y pocos vehículos implicados. Esto si intentamos verlo de forma análoga para el algoritmo K-means lo agruparía el cluster 0 también, pero como hemos comentado son diferentes, ya que el cluster 0 del algoritmo K-means 4.1 contenía a datos

con accidentes en donde se producían muchas muertes, y esto para el algoritmo no se da por parte del cluster 0. De hecho para el caso del algoritmo Spectral podríamos afirmar que el cluster 0 agrupa a su vez accidentes en donde no se producen muertes.
En el siguiente apartado se mostrarán los resultados y el análisis final con respecto al caso de estudio con colisiones entre vehículos en trazado con curva fuerte.

4.2. Caso de Estudio: Accidentes de tráfico con colisiones entre vehículos en trazado con curva fuerte.

En la siguiente tabla se muestran datos asociados a cada algoritmo utilizado para este caso de estudio, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 1.143 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	1674.925168	4	0.703378	0.011308
MiniBatchKMeans	1672.991880	4	0.703378	0.021997
Birch	553.091634	4	0.565622	0.022549
DBSCAN	706.660193	11	0.816862	0.016068
MeanShift	1006.650146	20	0.669160	0.119152
Spectral	1459.547373	4	0.706005	0.361586

Tabla 4.5: Datos generales asociados a cada uno de los algoritmos, caso de estudio 3 en trazado con curva fuerte.

Con respecto a las métricas concentradas en la tabla 4.5 podemos argumentar lo siguiente:

El algoritmo que mejores métricas saca para este caso de estudio vuelve a ser K-means, seguido de MiniBAtchKmeans, en este caso el CH es algo mejor en K-means, luego parece ser que volviendo ala explicación anterior, el K-mean es el que mejor maximiza la similaridad intra-cluster y minimiza la similaridad inter-cluster.

Para el índice SC sin embargo sería el algoritmo DBSCAN, de hecho, supera por bas-

tante al resto de algoritmos, y esto es un claro indicativo de que esos 11 clusters pueden estar cercanos al número óptimo de clusters con los que representar el problema mediante agrupamiento.

Para este caso, tendríamos el algoritmo Birch de nuevo como el peor de todos, ya que la métrica CH está muy por debajo de las demás, y la métrica SC igual. Y es que parece ser que esos 4 clusters de entrada al algoritmo que se le pasa como parámetro no es ni mucho menos el más idóneo.

Como de costumbre en nuestro análisis, el Spectral es el algoritmo más costoso en cuanto a tiempo de ejecución se refiere, y es que como venimos contando es algo costoso.

En las siguientes subsecciones se muestran gráficas y tablas asociadas a cada algoritmo usado para el caso de estudio con con colisiones entre vehículos en trazado con curva fuerte, y al final un breve análisis.

4.2.1. Resultados algoritmo K-means, caso de estudio de colisiones entre vehículos en trazado con curva fuerte.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 1.143 elementos, se seleccionan 1.143.
La siguiente gráfica 4.7 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

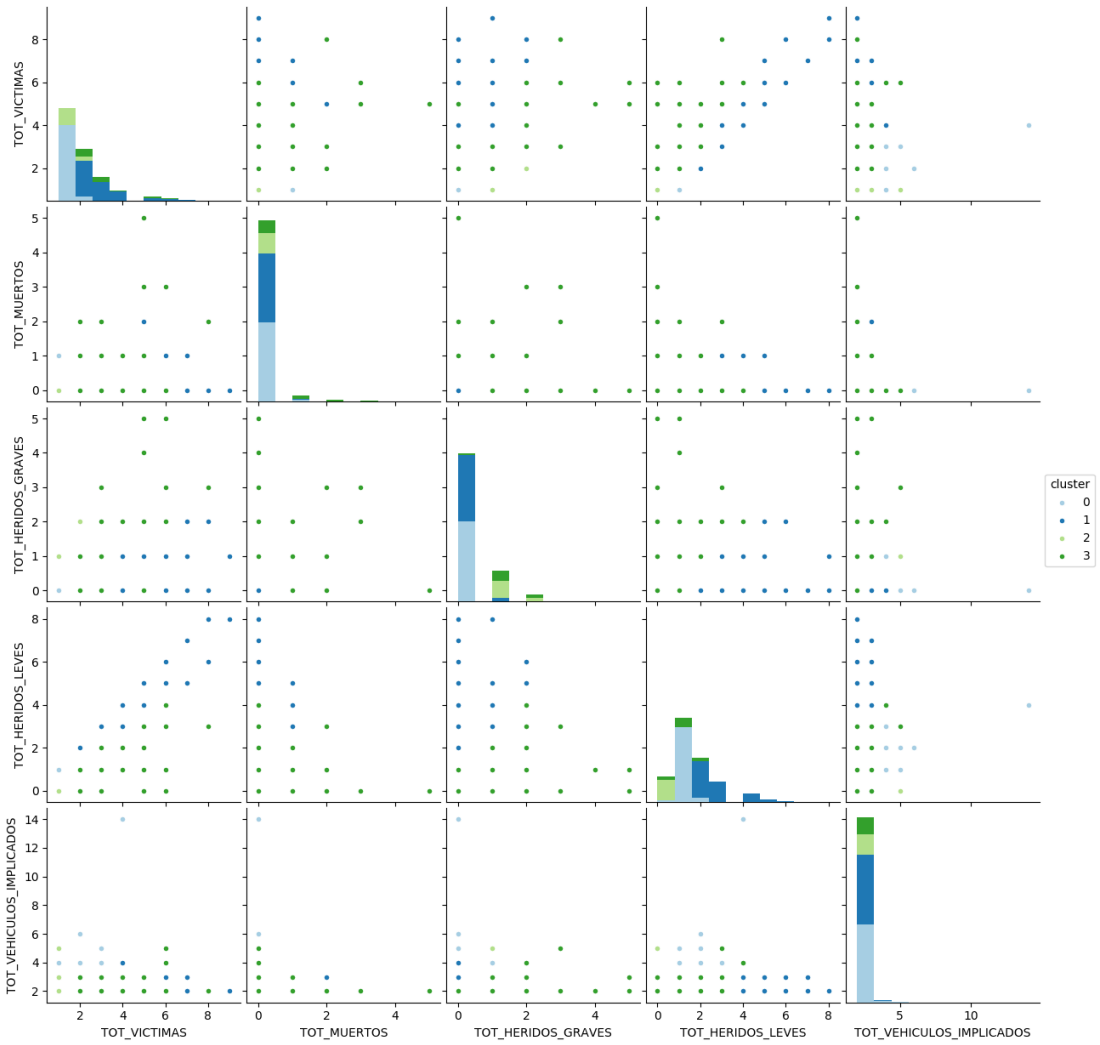

Figura 4.7: Scatter Matrix usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte.

La siguiente gráfica (Heatmap) 4.8 representa a la tabla 4.6 pero con sus datos normalizados:

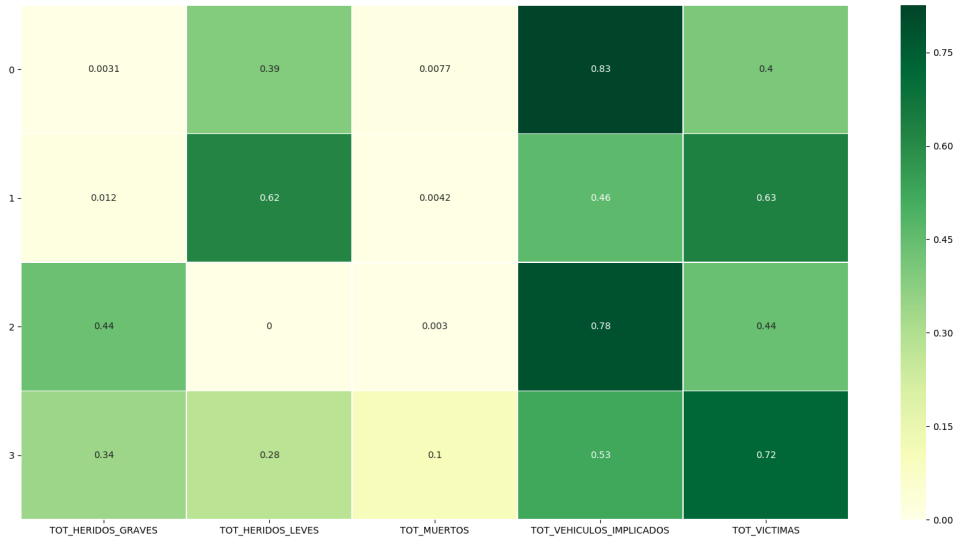

Figura 4.8: Heatmap usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte.

La siguiente tabla 4.6 está compuesta por los datos en media referentes al algoritmo K-means con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.008163	1.044898	0.020408	2.193878	1.073469
1	0.054632	2.779097	0.019002	2.078385	2.852732
2	1.157480	0.000000	0.007874	2.078740	1.165354
3	1.409524	1.152381	0.419048	2.171429	2.980952

Tabla 4.6: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo K-means, caso de estudio 3 en trazado con curva fuerte.

4.2.2. Resultados algoritmo DBSCAN, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:

De los 11 clusters hay 11 con más de 3 elementos. Del total de 1.143 elementos, se seleccionan 1.143.

La siguiente gráfica 4.9 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

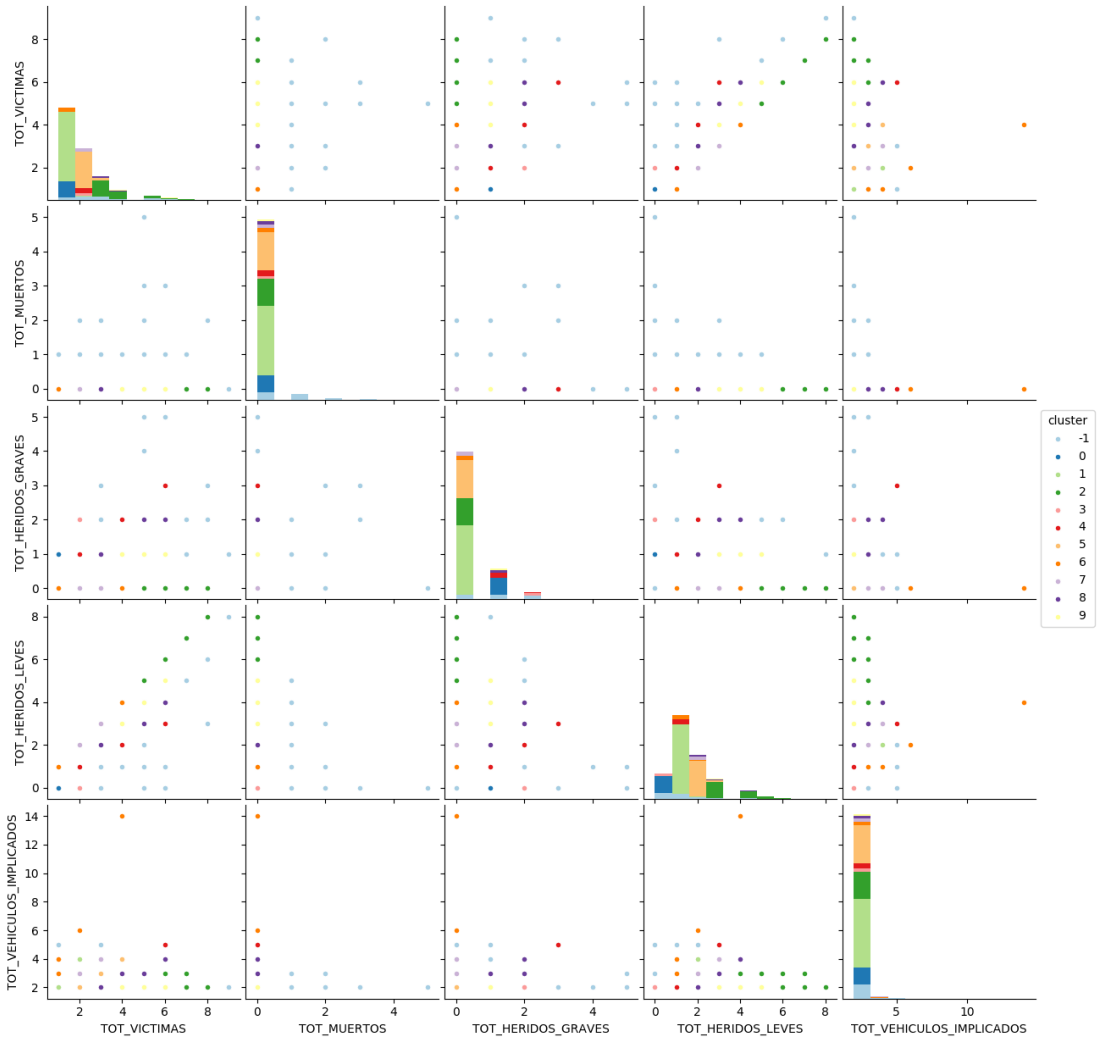

Figura 4.9: Scatter Matrix usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte.

La siguiente gráfica (Heatmap) 4.10 representa los datos de la tabla 4.7 pero con sus datos normalizados:

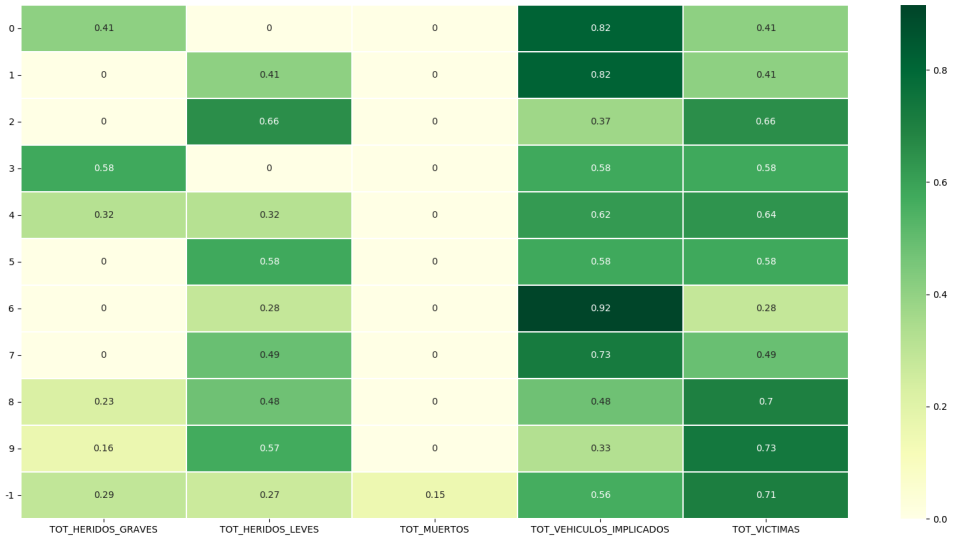

Figura 4.10: Heatmap usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte.

La siguiente tabla 4.7 está compuesta por los datos en media referentes al algoritmo DBSCAN con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	1.000000	0.000000	0.000000	2.000000	1.000000
1	0.000000	1.004717	0.000000	2.009434	1.004717
2	0.000000	3.642424	0.000000	2.078788	3.642424
3	2.000000	0.000000	0.000000	2.000000	2.000000
4	1.090909	1.090909	0.000000	2.121212	2.181818
5	0.000000	2.056277	0.000000	2.056277	2.056277
6	0.000000	1.142857	0.000000	3.678571	1.142857
7	0.000000	2.045455	0.000000	3.045455	2.045455
8	1.105263	2.315789	0.000000	2.315789	3.421053
9	1.000000	3.500000	0.000000	2.000000	4.500000
-1	1.293478	1.184783	0.684783	2.500000	3.163043

Tabla 4.7: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo DBSCAN, caso de estudio 3 en trazado con curva fuerte.

4.2.3. Resultados algoritmo Spectral, caso de estudio de colisiones entre vehículos en zonas y vías urbanas.

En primer lugar comentar que se ha realizado la eliminación de los outliers, es decir, la eliminación de aquellos clusters con pocos datos. Se ha realizado mediante un filtrado a los clusters con menos de 3 elementos:
De los 4 clusters hay 4 con más de 3 elementos. Del total de 1.143 elementos, se seleccionan 1.143.
La siguiente gráfica 4.11 representa como están distribuidos los diferentes clusters sobre las diferentes variables estudiadas:

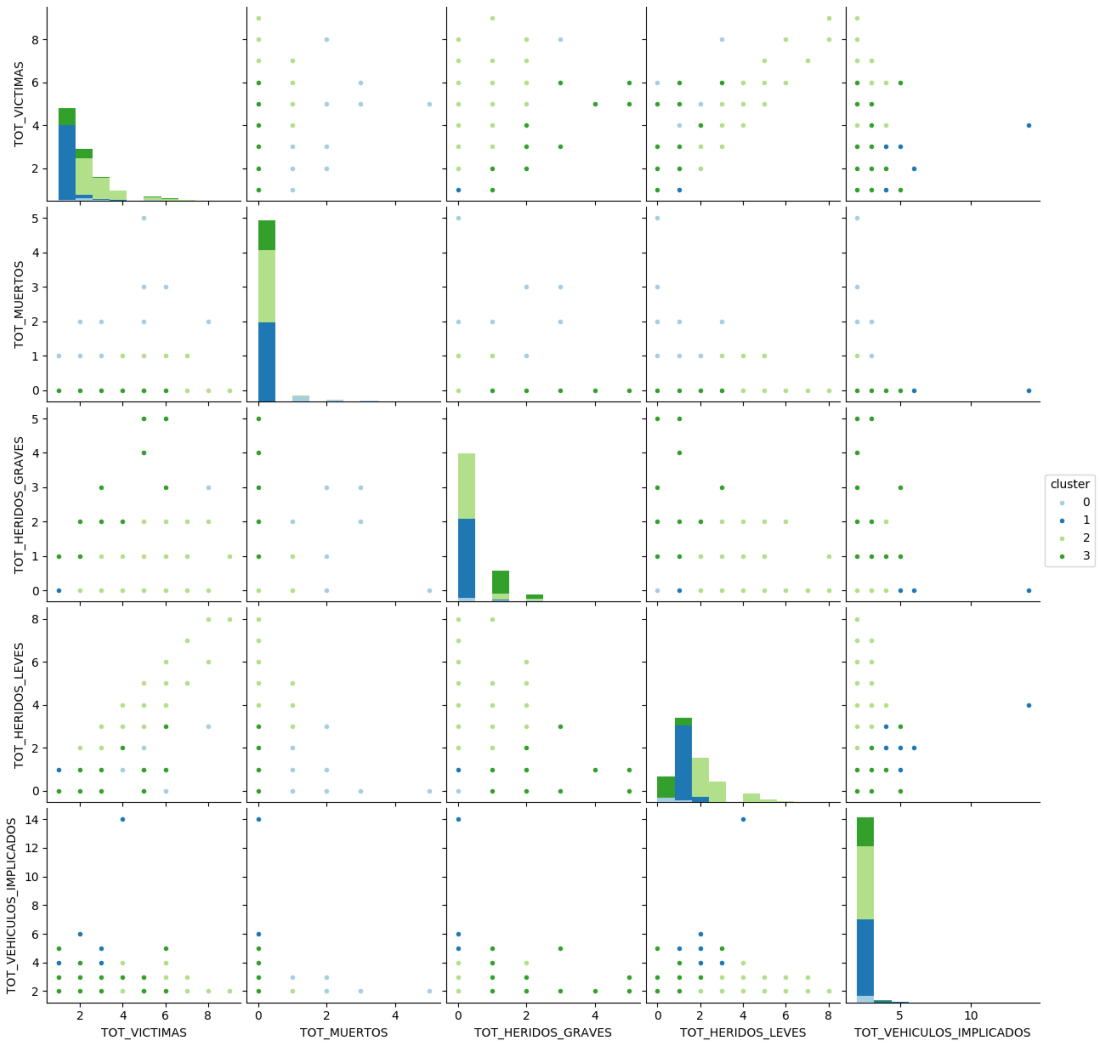

Figura 4.11: Scatter Matrix usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte.

La siguiente gráfica (Heatmap) 4.12 representa los datos de la tabla 4.8 pero con sus datos normalizados:

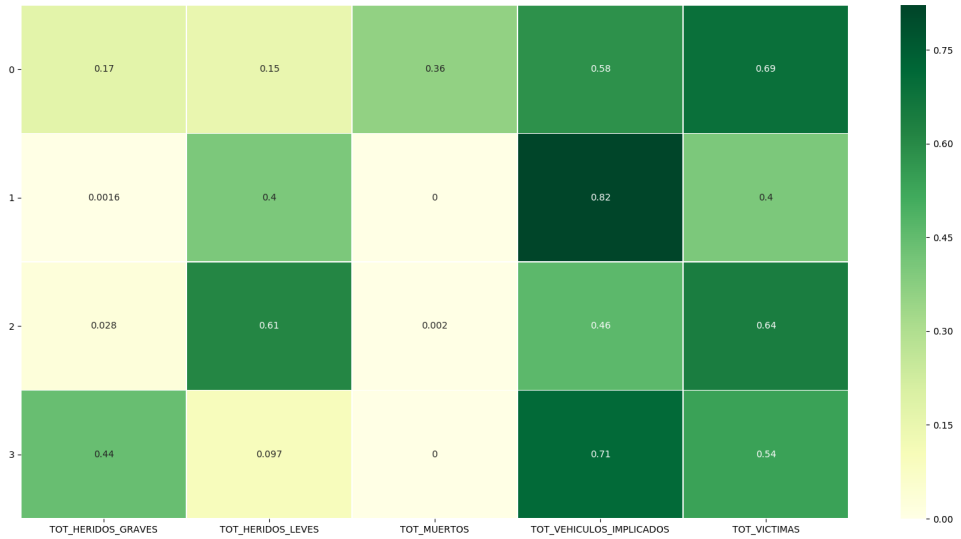

Figura 4.12: Heatmap usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte.

La siguiente tabla 4.8 está compuesta por los datos en media referentes al algoritmo Spectral con todas las variables que se han tenido en cuenta para el análisis:

CLUSTER	HERIDOS_GRAVES_MED	HERIDOS_LEVES_MED	MUERTOS_MED	VEHICULOS_IMPLICADOS_MED	VICTIMAS_MED
0	0.666667	0.595238	1.404762	2.261905	2.666667
1	0.004184	1.064854	0.000000	2.177824	1.069038
2	0.124153	2.747178	0.009029	2.083521	2.880361
3	1.316667	0.288889	0.000000	2.127778	1.605556

Tabla 4.8: Media de los datos totales de cada variable asociados a cada cluster usando el algoritmo Spectral, caso de estudio 3 en trazado con curva fuerte.

4.2.4. Interpretación de la segmentación: Accidentes de tráfico con colisiones entre vehículos en trazado con curva fuerte.

Para el análisis de el caso de estudio de los accidentes de tráfico con colisiones entre vehículos en trazado con curva fuerte se hará uso de las distintas gráficas y tablas mostradas anteriormente, de tal manera que podremos crear una base sólida de todo aquello que argumentemos.

Para el algoritmo DBSCAN se puede apreciar como se distribuyen los clusters de la siguiente forma:

El cluster -1, mediante la gráfica 4.9 se puede ver como agrupa a aquellos accidentes donde se producen muertos y donde hay pocos vehículos implicados. También se puede ver que en dicho cluster se agrupan los accidentes donde se producen un número alto de heridos graves y un número bajo de heridos leves.

También podemos observar, que el cluster 4 que aparece con cuentagotas en la gráfica 4.9 agrupa a los accidentes en donde hay un número medio de heridos graves y número medio de vehículos implicados. Pero como se comentaba, parece ser que los accidentes de este tipo no son muy comunes en los trazados con curva fuerte.
Si fijamos nuestros intereses en la gráfica 4.10 podemos ver en términos promedios como los clusters que no son el -1, no agrupan accidentes en donde se producen muertos, recordemos en términos promedios.
Para ver, sin embargo, de forma promedia el clusters que agrupa aquellos accidentes donde más vehículos implicados hay, debemos irnos al cluster 6. Sin embargo no es el que recoge los accidentes donde más muertos se producen, dato bastante relevante, ya que nos hace ver que no son los accidentes que más muertos producen, por muchos vehículos implicados que haya.

Para el algoritmo Spectral, podemos argumentar lo siguiente:
En primer lugar la diferencia con respecto al algoritmo DBSCAN es que agrupa en 4 clusters diferentes.
En segundo lugar, observando la gráfica 4.12, y a diferencia del algoritmo DBSCAN, parece que el Spectral los accidentes agrupados por el cluster 1, que es el que concentra

en términos promedios los accidentes donde más vehículos implicados hay, también concentra a los accidentes donde más heridos graves hay, al contrario de lo que ocurría en el algoritmo DBSCAN, según se puede observar en la gráfica 4.10.

Por útlimo cabe comentar que si se observa la gráfica 4.11 del algoritmo Spectral, ocurre como en el caso del algoritmo DBSCAN previamente comentado, y es que el cluster que agrupa a aquellos accidentes donde más muertos se producen, en este caso el 0, no es el que concentra a los accidentes donde más vehículos hay implicados.

4.3. Modificación de los parámtros para algunos algoritmos referentes al caso de estudio con colisiones entre vehículos en trazado con curva suave.

En esta sub-sección nos encargaremos de realizar un estudio con mayor detalle de algunos de los algoritmos detallados anteriormente , de tal forma que podamos entender mejor los parámetros pasados a los mismos.

En este caso vamos a realizar este análisis sobre los algoritmos K-means, DBSCAN y Spectral.

Para el análisis nos han salido los siguientes resultados referentes a los datos asociados a cada algoritmo utilizado para este caso de estudio con colisiones entre vehículos en trazado con curva suave, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado con un total de 2.542 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	3546.777282	6	0.750620	0.026536
DBSCAN	681.872497	4	0.420032	0.095258
Spectral	3355.232123	6	0.745046	1.966724

Tabla 4.9: Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 3 en trazado con curva suave.

En la tabla 4.9 se aprecian las siguientes diferencias con respecto a la tabla 4.1, el primero de ellos es que en lo que al índice CH se refiere los algoritmos K-means y Spectral mejoran con las nuevas modificaciones, que para el caso del algoritmo K-means es el cambio del parámetro K de 4 a 6, y para el caso del algoritmo Spectral es el cambio del parámetro K también de 4 a 6. Estos cambios no solo han afectado positivamente al índice CH, si no que también ha hecho crecer a la métrica SC, lo que nos indica que es más acertada esta elección con estos parámetros que la elección primera que se hizo.

En contraposición a estos buenos resultados, tenemos el algoritmo DBSCAN, que ha sufrido el siguiente cambio, que no es más que pasar de tener una variable min_sample igual a 10 a otra nueva igual a 100, y con un eps igual a 0.2 por el que había anteriormente que era de 0.1.
Como decíamos, este algoritmo no termina de mejorar las métricas existentes en la tabla 4.1, de hecho lo que ocurre es que empeoran los datos con la nueva disposición del algoritmo.

Por último, y en forma de conclusión para esta pequeña modificación de los distintos algoritmos, podemos decir que los algoritmos K-means y Spectral si nos acogemos a las métricas expuestas en la tabla 4.9 han mejorado con respecto a propuestas anteriores, de hecho lo único que empeoraría sería el tiempo necesario para la ejecución de cada algoritmo.
Y también cabe comentar que para el algoritmo DBSCAN los resultados no son nada buenos para la nueva disposición, por lo tanto deberíamos probar algo nuevo, y examinarlo con atención.

4.4. Modificación de los parámtros para algunos algoritmos referentes al caso de estudio con colisiones entre vehículos en trazado con curva fuerte.

En esta sub-sección nos encargaremos de nuevo de realizar un estudio con mayor detalle de algunos de los algoritmos detallados anteriormente , de tal forma que podamos entender mejor los parámetros pasados a los mismos.

En este caso vamos a realizar este análisis sobre los algoritmos K-means, DBSCAN y Spectral.

Para el análisis nos han salido los siguientes resultados referentes a los datos asociados a cada algoritmo utilizado para este caso de estudio con colisiones entre vehículos en trazado con curva fuerte, datos como el número de clusters que se han utilizado, la métrica Calinski-Harabasz (CH) [4], la métrica Silhouette (SC) [3] y el tiempo que ha tardado el algoritmo en ejecutarse en segundos.Para este caso de estudio se ha contado

con un total de 1.143 instancias.

Algoritmo	CH	N.Clusters	SC	Tiempo (s)
K-means	2198.220100	9	0.807287	0.021003
DBSCAN	108.532491	2	0.338955	0.028306
Spectral	1852.292881	9	0.774868	0.334506

Tabla 4.10: Datos generales asociados a cada uno de los algoritmos con sus parámetros modificados, caso de estudio 3 en trazado con curva fuerte.

Para esta modificación se han sacado las siguientes conclusiones:

En primer lugar podemos decir que el algoritmo K-means disminuye su métrica CH con respecto a anteriores disposiciones del mismo. Sin embargo, para la métrica SC ocurre exactamente lo contrario, y es que aumenta sustancialmente con respecto por ejemplo de la tabla 4.5. La modificación realizada para este algoritmo ha sido la de variar el parámetro K de 4 a 9.

Con respecto al algoritmo DBSCAN seguimos bajando los valores tanto de la métrica CH como de la métrica SC, lo cuál nos indica que no es el camino a seguir. La modificación realizada en este caso ha sido la de aumentar el min_sample de 10 a 1000 y aumentar el eps de 0.1 a 0.5. Dichos valores se encuentran explicados junto a su definición en la sección 2.4.

Y por último, para el algoritmo Spectral se puede apreciar que se produce una disminución en la métrica CH con respecto a disposiciones pasadas como las de las tablas 4.9 y 4.5. Sin embargo con respecto a la métrica SC se produce lo que con el algoritmo K-means, y es que se produce un aumento con respecto a disposiciones anteriores como la tabla 4.5

5. Contenido adicional

Referencias

[1] http://scikit-learn.org/stable/modules/clustering.html#k-means, consultado el 1 3de Diciembre de 2017.

[2] http://datamining.rutgers.edu/publication/internalmeasures.pdf, consultado el 11 de Diciembre de 2017.

[3] http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html, consultado el 11 de Diciembre de 2017.

[4] https://eva.fing.edu.uy/file.php/514/ARCHIVO/2011/TrabajosFinales2011/informe_final_introini_lena.pdf, consultado el 11 de Diciembre de 2017.

[5] https://www.python.org/downloads/release/python-363/, consultado el 11 de Diciembre de 2017.

[6] http://.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/TUZEL1/MeanShift.pdf, consultado el 12 de Diciembre de 2017.

[7] http://scikit-learn.org/stable/modules/clustering.html#birch, consultado el 12 de Diciembre de 2017.

[8] https://en.wikipedia.org/wiki/BIRCH, consultado el 12 de Diciembre de 2017.

[9] https://en.wikipedia.org/wiki/Dendrogram, consultado el 12 de Diciembre de 2017.

[10] https://en.wikipedia.org/wiki/Spectral_clustering, consultado el 12 de Diciembre de 2017.

[11] https://es.wikipedia.org/wiki/DBSCAN, consultado el 12 de Diciembre de 2017.

[12] https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html, consultado el 12 de Diciembre de 2017.

[13] https://netwalkersuite.org/tutorials/doxorubicin/clustering-heatmap-analysis, consultado el 12 de Diciembre de 2017.

[14] http://wpd.ugr.es/~bioestad/guia-spss/practica-8/, consultado el 12 de Diciembre de 2017.

[15] http://www.ugr.es/~gallardo/pdf/cluster-3.pdf, consultado el 12 de Diciembre de 2017.

[16] http://scikit-learn.org/stable/modules/clustering.html, consultado el 13 de Diciembre de 2017.

[17] http://scikit-learn.org/stable/modules/clustering.html#dbscan, consultado el 13 de Diciembre de 2017.

[18] http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering, consultado el 13 de Diciembre de 2017.

[19] http://scikit-learn.org/stable/modules/clustering.html#mean-shift, consultado el 13 de Diciembre de 2017.

[20] http://scikit-learn.org/stable/modules/clustering.html#spectral-clustering, consultado el 13 de Diciembre de 2017.

[21] https://sedeapl.dgt.gob.es/WEB_IEST_CONSULTA/subcategoria.faces, consultado el 8 de Diciembre de 2017.