El Mercado Automovilístico: Un caso para el Análisis Estadístico Multivariante

José Jaén Delgado

1 Introducción

En esta última parte del proyecto el foco de atención pasa a la aplicación de métodos estadísticos de Análisis de Conglomerados o *Cluster Analysis*. Salvo que se especifique lo contrario, la autoría de las diferentes aplicaciones no predifinidas en **MATLAB** de las técnicas estadísticas que emplearemos se atribuye a Baíllo & Grané (2007).

2 Descripción de los Datos

Table 1: Datos de automóviles de ***UCI ML Repository***

Variables	Tipo	Transformación	Indicador
`distancia_ejes`	Cuantitativa	$\log(X_i)$	-
`largo`	Cuantitativa	$\log(X_i)$	-
`ancho`	Cuantitativa	$\log(X_i)$	-
`altura`	Cuantitativa	$\sqrt{X_i}$	-
`peso`	Cuantitativa	$\log(X_i)$	-
`motor`	Cuantitativa	$\log(X_i)$	-
`caballos`	Cuantitativa	$\log(X_i)$	-
`max_revoluciones`	Cuantitativa	$\log(X_i)$	-
`precio`	Cuantitativa	$\log(X_i)$	-
`combustible`	Binaria	-	Diésel / Gasolina
`rueda_motriz`	Multiestado	-	Delantera / Trasera / 4x4

Aplicamos ***transformaciones no lineales*** para aumentar la simetría de las variables, identificar fácilmente las relaciones entre éstas, atenuar los efectos de los datos atípicos e incrementar la variabilidad explicada de la matriz de datos por PCA o MDS.

3 Clasificación Jerárquica

En esta sección aplicamos el método de Clasificación Jerárquica como primera técnica del Análisis de Conglomerados. Partimos de un escenario en el cual suponemos que en nuestro conjunto de datos no existe un etiquetado específico que determine el grupo al que pertenecen las observaciones. Consecuentemente, nos encontramos en un contexto de **clasificación no supervisada**, y por tanto emplearemos algoritmos que identifiquen y asignen automáticamente una observación dada a un *cluster* en el cual se encuentren otros individuos con características similares.

Obtendremos distintos conglomerados de manera sucesiva en clases de nivel superior que representaremos mediante un dendrograma. La base para lograr este objetivo es una matriz de distancias $\mathbf{D}^*$ que cumpla la propiedad **ultramétrica**:

$$\mathbf{D}^* \text{ es ultramétrica si } \begin{cases} \delta_{ij} = \delta_{ji} & \forall i, j \\ \delta_{ii} = 0 & \forall i \\ \delta_{ij} \leq \max\{\delta_{ik}, \delta_{kj}\} & \forall\, i, j, k \end{cases}$$

Emplearemos el siguiente algoritmo para obtener representaciones jerárquicas:

Algorithm 1: Algoritmo de tipo aglomerativo

Input: Matriz de distancias $\mathbf{D}^{(2)}$

Output: Matriz de distancias ultramétrica $\mathbf{D}^*$

1. Inicializar partición:

$$\varepsilon = \{1\} + \{2\} + \ldots + \{n\}$$

2. Crear un nuevo conglomerado:

$$\{i\} \cup \ \{j\} = \{i, j\} \quad \text{donde } \delta_{ij} = \min\{\delta_{kl}\}$$

3. Definir distancia de 2) al resto de elementos:

$$\delta'_{k,(ij)} = f(\delta_{ik}, \delta_{jk}) \quad k \neq i, j$$

$$\varepsilon = \{1\} + \cdots + \{i, j\} + \cdots + \{n\}$$

while $\varepsilon \neq \{1, 2, \ldots, n\}$ **do**

Pasos 2) y 3)

end

Con dicho algoritmo aunaremos iterativamente los automóviles y se recalcularán las distancias de los conglomerados del segundo paso al resto procurando cumplir la propiedad ultramétrica hasta obtener un único *cluster*. Nótese cómo la matriz que sirve de input es la Distancia de Gower personalizada que ***propusimos*** anteriormente, ya que nuestra base de datos presenta variables cuantitativas y cualitativas. Asimismo, es necesario señalar que el recálculo de distancias del algoritmo puede realizarse de diferentes modos, ya que mientras la función $f(\delta_{ik}, \delta_{jk})$ se defina de tal forma que no se comprometa la propiedad ultramétrica los resultados de los algoritmos de tipo aglomerativo son correctos. En concreto contrastaremos tres métodos diferentes de definir la función:

- Método del Mínimo: $f(\delta_{ik}, \delta_{jk}) = \min\{\delta_{ik}, \delta_{jk}\} \quad k \neq i, j$
- Método del Máximo: $f(\delta_{ik}, \delta_{jk}) = \max\{\delta_{ik}, \delta_{jk}\} \quad k \neq i, j$
- Método *Unweigthed Pair Group Method using Arithmetic Averages* (UPGMA): $\delta'(E_k, E_i \cup E_j) = \frac{n_i}{n_i+n_j}\delta E_i, E_k + \frac{n_J}{n_I+n_J}\delta E_j, E_k \quad k \neq i, j$

La representación gráfica de dichos métodos se aprecia a continuación:

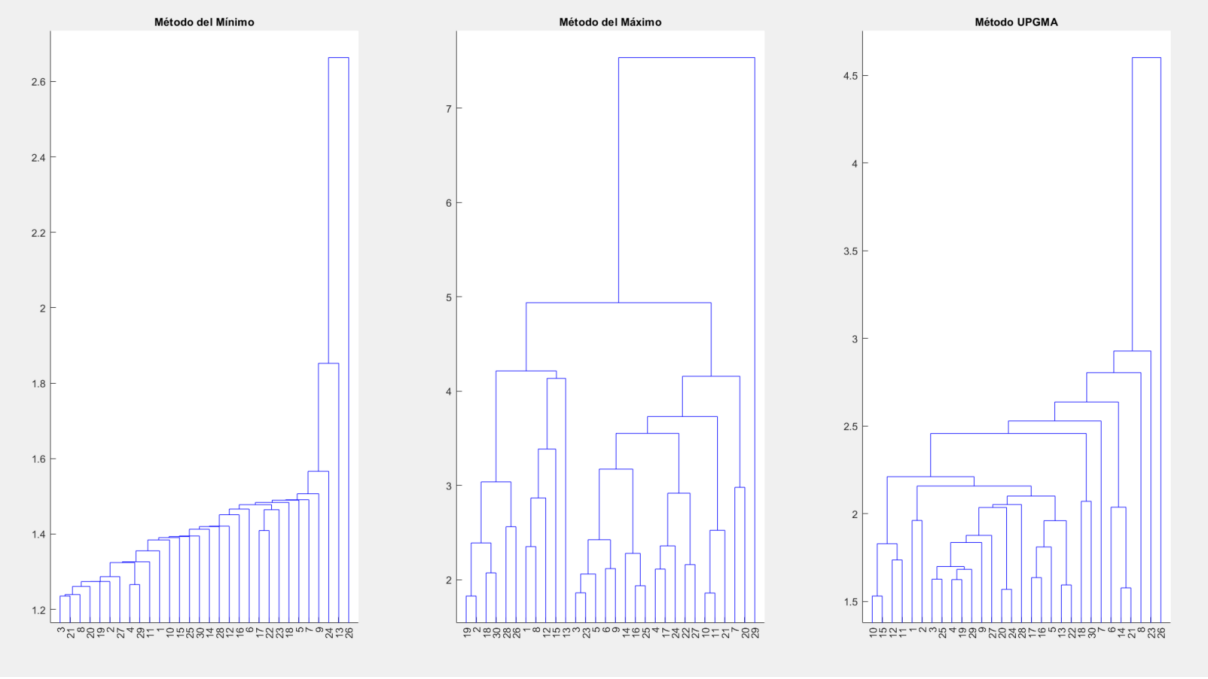

Figure 1: Comparación Clasificación Jerárquica

El primero de todos tiende a contraer los grupos, el segundo a dilatar el espacio de las clases y el último representa un punto más bien intermedio, dado que utiliza medias ponderadas según el número de elementos que hay en cada *cluster*.

Al objeto de elegir el método óptimo utilizaremos como métrica la Correlación Cofenética, un criterio de proximidad que determina el grado de alteración sufrido por la matriz de distancias original respecto a la nueva matriz ultramétrica. Mientras más cercana sea a uno menor es la alteración sufrida por $\mathbf{D}^{(2)}$ para convertirse en $\mathbf{D}^{*}$.

Una vez calculadas las Correlaciones Cofenéticas obtenemos que el Método UPGMA es el óptimo dado que es el que perturba menos nuestra matriz de distancias original (0.7899), seguida de cerca por el Método del Mínimo (0.7464), mientras que el Método del Máximo altera sustancialmente $\mathbf{D}^{(2)}$ con una Correlación Cofenética de 0.5166. A raíz de estos resultados es fácilmente perceptible que la matriz de distancias que hemos propuesto no cumplía la propiedad ultramétrica dado que ninguna Correlación Cofenética es igual a uno.

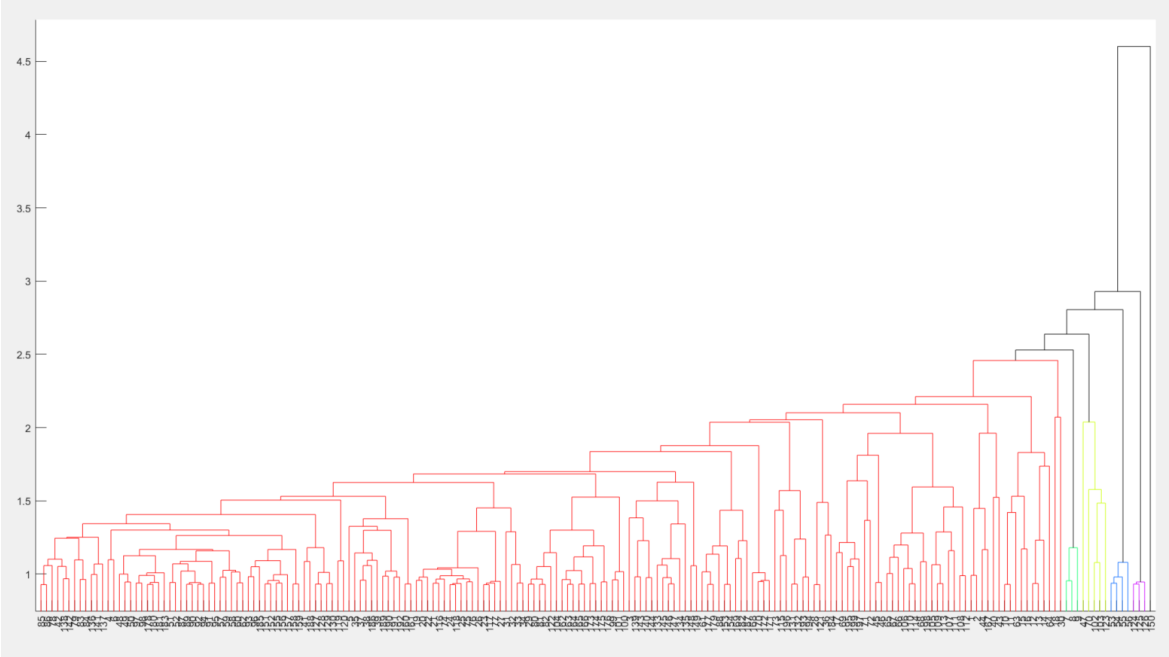

Figure 2: Método UPGMA

La interpretación de la clasificación jerárquica puede convertirse en una tarea intricada incluso para el caso de expertos en la materia de estudio, en buena medida explicado por la ausencia de una etiqueta que indique el grupo al que pertenecen las observaciones. No obstante, que no exista un agrupamiento previo y aún así se lo-

gre clasificar individuos es una evidente ventaja respecto de ténicas como la regresión logística o modelos probit entre otros, ya que precisan de un *label* o identificador de grupos para poder emplearse.

En un intento de explicar los resultados del Método UPGMA, en primer lugar cabe señalar cómo se interpreta el dendrograma: las ramas que se juntan antes (representadas por un mismo color) son más similares entre sí que el resto. Apreciamos cómo existe una distribución desigual entre *clusters* de la misma manera que nuestra base de datos presenta un pronunciado desequilibrio: 90% de los automóviles usan gasolina frente al 10% restante de diésel. Similarmente, el 58.7% de los coches utilizan tracción delantera, un 37% tracción trasera y sólo un 3.7% son todoterreno. Recordamos que los gráficos de configuración MDS que ***representamos*** en el anterior proyecto mostraban grupos claramente dominantes sobre otros (tendencias parecidas de distintos métodos).

Así, echando un vistazo detallado al dendrograma, los automóviles que pertenecen al conglomerado de color morado son de tracción trasera, su precio es el más alto de todos los grupos (en promedio 10.45 contra una media de 9.35) y también presentan los mayores valores para las variables `caballos` y `motor`. Es presumible que los coches más modernos, potentes y de estilo deportivo pertenezcan a este *cluster* según Zuto (2020). Por su parte los coches del grupo amarillo consumen gasolina, son de tracción trasera y su precio es mayor que la media, pero inferior a los anteriores. Dado que el peso de éstos es superior a los del grupo morado, es lógico concluir que aún situándose en una gama de calidad similar a éstos, los coches del conglomerado amarillo son modernos pero no deportivos.

El grupo mayoritario (rojo) incluye una verdadera amalgama de vehículos: todos los diésel y los todoterreno, si bien un análisis de medidas de centralidad revela que sus características se sitúan más bien en la media. Pese a representar el automóvil medio, acapara categorías que intuitivamente habrían de estar separadas en un principio. No debería de sorprendernos este resultado, ya que en tareas de clasificación es muy frecuente que grupos infrarrepresentados (como el caso de los diésel y todoterreno) no sean asignados con mucha precisión a su grupo correspondiente.

Y es que pese a haber realizado varias operaciones conducentes a aumentar la simetría de las variables, aproximar las unidades de medida y empleado una distancia de Gower un tanto más robusta, ni el preprocesamiento de los datos ni nuestro clasificador están cerca de ser calificados de sofisticados. Aún así hemos conseguido no perturbar considerablemente la matriz de distancias original e interpretar algunos *clusters*.

4 Clasificación No Jerárquica

En esta última sección aplicamos el algoritmo de k-means o k-medias, a veces referido como algoritmo Lloyd-Forgy, ya que fue propuesto por Stuart Lloyd en 1957 para tareas de modulación por impulsos codificados (no lo publicó fuera de la compañía Bell Laboratories hasta 1982), y Forgy (1965) por su parte compartió en *Biometrics* un procedimiento prácticamente idéntico.

Una explicación intuitiva del funcionamiento del algoritmo es la siguiente: en nuestro conjunto de datos elegimos k candidatos a representar un conglomerado (centroides), y asignamos las restantes observaciones a los *clusters* según estén más cerca de éstos. La complejidad computacional es generalmente lineal en relación al número de observaciones, el número de *clusters* y el número de dimensiones. La principal dificultad radica en que usualmente ni sabemos el número de centroides a elegir ni a cuál pertenece cada individuo de nuestra base de datos.

Por ello, es frecuente inicializar aleatoriamente los centroides, identificar las observaciones más cercanas a los mismos, actualizar los centroides en la siguiente iteración y proseguir sucesivamente hasta que los centroides no se muevan. La convergencia está asegurada ya que la distancia al cuadrado media de un determinado individuo y su centroide más cercano no puede más que disminuir por cada iteración, lo que no implica que sí converja a un óptimo global (Géron, 2019).

La plataforma computacional **MATLAB** emplea un método más sofisticado, concretamente el k-means++ propuesto por Arthur & Vassilvitskii (2007) en el que se tiende a seleccionar centroides distantes unos de otros, evitando más eficazmente soluciones subóptimas. La siguiente figura recoge el pseudo-código de los conceptos presentados:

Algorithm 2: Algoritmo de k-medias

Input: Número de clusters K, matriz de datos $\mathbf{X}$, tolerancia $\varepsilon > 0$

Output: Conjunto de K clusters

1. Inicializar centroides aleatoriamente: $\{\mu_1, \mu_2, \ldots, \mu_k\}$

while $\left||\mathbf{W}|^{(i+j)} - |\mathbf{W}|^{(i)}\right| > \varepsilon$ para $j \geq 1$ **do**

2. Por cada observación:

$$c^{(i)} = \arg\min_j \|x^{(i)} - \mu_j\|^2$$

$$\mu_j = \frac{\sum_{i=1}^{n} \mathbb{1}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{n} \mathbb{1}\{c^{(i)} = j\}}$$

end

Nótese cómo se minimiza la matriz de dispersión dentro de los grupos para que las observaciones de un mismo conglomerado se asemejen lo máximo posible, convergiendo el algoritmo cuando dicha matriz no disminuya sustancialmente. Asimismo, es imprescindible recalcar que la medida utilizada para definir la separación de los datos a los *clusters* es la Distancia Euclídea, con todo lo que ello conlleva: restricción a variables cuantitativas, asumir incorrelación entre *inputs*, sensibilidad a cambios de escala... Consecuentemente, proponemos el siguiente procedimiento para superar las limitaciones mencionadas: obtener unos ejes de representación ortogonales que permitan incluir variables binarias y multiestado. Esto se conseguirá mediante el **Multidimensional Scaling** o **MDS** a partir de nuestra propia matriz de distancias al cuadrado $\mathbf{D}^{(2)}$ que introducimos en la sección anterior.

Aplicamos la función `kmedias2()` de Baíllo & Grané (2007) sobre la configuración MDS de nuestros datos y obtenemos las siguientes salidas gráficas:

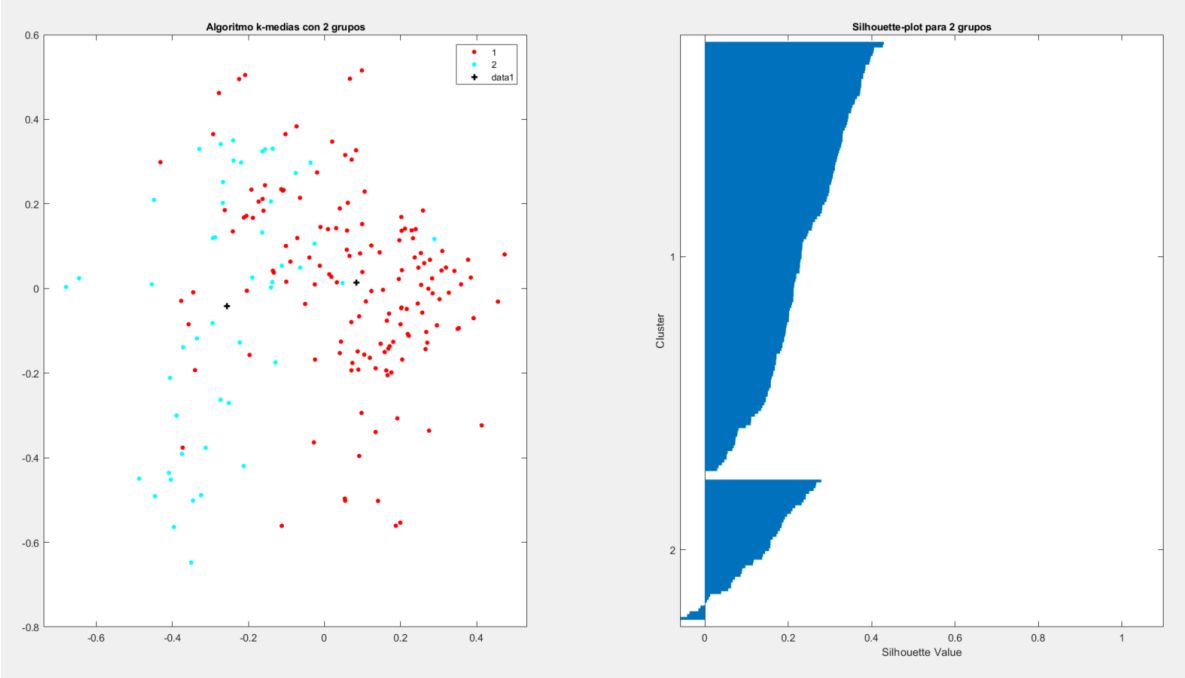

Figure 3: Algoritmo k-means (máximo 6 clusters)

El gráfico de la izquierda muestra la representación en dos dimensiones de la Clasificación del algoritmo k-means. A la derecha se aprecia el *Silhouette-plot* para evaluar la calidad del *clustering* a raíz de la siguiente métrica utilizada por Rousseeuw (1987):

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

La razón por la que no se ha empleado la técnica de PCA pese a procurar ejes incorrelados entre sí es fácilmente entendible a raíz de su propia naturaleza: sólo admite variables de tipo cuantitativo. Por ende, en caso de optar por PCA en vez de MDS estaríamos obviando la información contenida en los *inputs* de carácter cualitativo.

Volviendo a la métrica $s(i)$, considérese $a(i)$ como la distancia media de i al conglomerado que se asignó a dicha observación, y $b(i)$ como el mínimo de la distancia media de i al resto de *clusters*. Calculando la silueta media y considerando que en base a estas definiciones, $s(i) \in [-1, 1]$, el rendimiento de nuestro clasificador será alto si $\bar{s} \sim 1$ y todo lo contrario si se aproxima a -1. En nuestro caso la silueta media es 0.2134, lo que implica un resultado bastante mediocre, habida cuenta de que mientras más se acerque a cero la silueta media más incertidumbre rodea a nuestro clasificador.

El algoritmo de k-means detecta dos grupos en nuestro conjunto de datos y encuentra gran dificultad para identificar los componentes del segundo grupo (las únicas siluetas negativas se encuentran en dicho *cluster*), existiendo a su vez una gran vacilación por parte del clasificador en el resto de casos ($s(i) < 0.5 \ \forall i$).

Para interpretar el gráfico izquierdo, hará falta explicar el significado de las coordenadas principales Y_1, Y_2 a raíz de sus correlaciones con las variables originales (nos fijaremos en las relaciones mayores en valor absoluto). Y_1 presenta relaciones negativas con todas las variables menos las categóricas, por lo que lo tomaremos como un proxy inverso del tamaño (específicamente penaliza los coches con gasolina y de tracción trasera o 4x4). Y_2 exhibe correlaciones fuertes positivas únicamente para `precio` y `max_revoluciones`, de ahí que nos sirva como índice de potencia de los vehículos.

De esta manera, el primer grupo (color rojo), que se encuentra mayoritariamente en el lado derecho del eje de abscisas (valores positivos), agrupa a los automóviles de menor dimensión, con potencia más bien indefinida debido a la gran dispersión existente en el eje de ordenadas, si bien el centroide se identifica en la parte positiva de Y_2. Por ello, es esperable que coches de modesto tamaño pero con características técnicas heterogéneas se clasifiquen en el *cluster* rojo (encontraríamos deportivos y coches antiguos pequeños).

En cuanto al segundo conglomerado (color azul), lo identificamos en la parte negativa tanto de Y_1 como de Y_2 (sobre todo del primer eje), de ahí que esperemos coches más grandes pero que no optimicen tanto la tecnología de las revoluciones y sean además caros.

Aportamos una tabla con medidas de centralidad por grupos en la siguiente página.

Table 2: Grupos Algoritmo k-means

	Cluster Rojo			Cluster Azul		
Variables	Media	Mediana	Moda	Media	Mediana	Moda
`dist_ejes`	4.5653	4.5633	-	4.6719	4.6812	-
`largo`	5.1319	5.1399	-	5.2355	5.2407	-
`ancho`	3.3862	3.3869	-	3.4292	3.4348	-
`altura`	7.2805	7.2835	-	7.4868	7.4967	-
`peso`	7.7508	7.7456	-	8.0584	8.0488	-
`motor`	4.7153	4.6821	-	5.0568	5.0239	-
`caballos`	4.5011	4.4543	-	4.8200	4.8122	-
`max_revo`	8.5459	8.5564	-	8.5058	8.5172	-
`precio`	9.1977	9.0992	-	9.8240	9.7351	-
`combust`	-	-	1	-	-	1
`rueda`	-	-	2	-	-	1

Vemos cómo el *cluster* rojo o grupo 1 presenta valores medios y medianos menores que el *cluster* azul o grupo 2 en todas las variables cuantitativas a excepción de `max_revoluciones`, ya que como adelantamos los coches del segundo conglomerado no las optimizan tanto. Los coches con dimensión más reducida (valores inferiores de altura, anchura, largo, peso y distancia entre ejes), así como menor precio y caballos se encuentran en el primer grupo.

En cuanto a las variables cualitativas, en ambos grupos dominan los vehículos con gasolina (no sorprendente debido al gran desequilibrio de los datos) pero existen tendencias diferentes en cuanto a la tracción del coche: los automóviles más pequeños tienden a concentrar su fuerza en las ruedas traseras, mientras que los más grandes y caros lo hacen en las delanteras.

5 Conclusión & Limitaciones

Hemos empleado dos métodos para realizar *clustering* no supervisado: clasificación jerárquica y no jerárquica. El primero crea conglomerados sucesivamente en clases de nivel superior mientras que el segundo parte de centroides y la distancia de las observaciones a éstos.

La calidad de los clasificadores difiere bastante: el Método UPGMA no ha perturbado demasiado la matriz original de distancias, mientras que la métrica de rendimiento del algoritmo k-means (silueta media) no ha sido abrumadora.

No obstante, hemos sido capaces de identificar tendencias y algunos patrones para el caso de la clasificación jerárquica e interpretar la conformación de grupos en el algoritmo k-means. Pese a que la Correlación Cofenética es alta en el caso de la Clasificación Jerárquica, intentar comprender sus resultados ha resultado ligeramente más difícil y no tan claro como el segundo caso.

Como limitación encontramos el gran desequilibrio existente en nuestra base de datos, así como ausencia de una posible mejor inicialización de los centroides en el caso de k-means. Al primer caso podría responderse con la aplicación de los algoritmos de remuestreo **SMOTE**, al segundo con recomendaciones de expertos que permitan cambiar de perspectiva: introducción de a prioris informativos para superar las limitaciones de la perspectiva frecuentista.

6 Apéndice

El código de MATLAB puede encontrarse en el siguiente repositorio de ***GitHub***.

Las transformaciones se ovtienen a partir del algoritmo del primer proyecto elaborado en ***Python***.

Las funciones no modificadas y que no son propias de MATLAB se atribuyen a Baíllo & Grané (2007).

7 Referencias

Arturh, D., & Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding. *Conference: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*

Baíllo, A. & Grané, A. (2007).*100 Problemas Resueltos de Estadística Multivariante.* Delta Publicaciones

Forgy, E.W. (1965). Cluster Analysis of Multivariate Data: Efficiency vs Interpretability of Classifications. *Biometrics*

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* O'Reilly

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*

Zuto (2020). *The Car Size Evolution.* Zuto Car Finance