Máster en Ingeniería de Sistemas y Control

Trabajo fin de Máster

Aplicación de técnicas no supervisadas para la caracterización de reseñas de libros y análisis de su impacto en ventas

Estudiante:
Antonio Janeiro Gallardo

Directoras:
Raquel Dormido Canto
Natividad Duro Carralero

Curso: 2021/2022
Convocatoria: Febrero 2022

Máster en Ingeniería de Sistemas y Control

Trabajo fin de Máster

Aplicación de técnicas no supervisadas para la caracterización de reseñas de libros y análisis de su impacto en ventas

Tipo A: Proyecto específico propuesto por un profesor

Estudiante:
Antonio Janeiro Gallardo

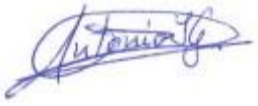
Directoras:
Raquel Dormido Canto
Natividad Duro Carralero

Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado: Antonio Janeiro Gallardo

Firma del alumno



Resumen

La implementación de reseñas dentro de las plataformas de comercio electrónico y servicios se ha vuelto hoy en día una estrategia muy utilizada por muchas empresas, permitiendo a los propios usuarios que sean los que valoren los productos y servicios que se ofrecen.

En el presente trabajo se describe cómo se van a caracterizar un gran conjunto de reseñas de libros obtenidas de la base de datos de Amazon y los metadatos referentes a los libros. Para ello se ha detallado el tratamiento que reciben dichas bases de datos para extraer las características más relevantes y combinarlas en un único repositorio.

Posteriormente se aplican diversas técnicas de *clustering* al conjunto de datos formado por esos libros, buscando que cada grupo contenga datos similares que sean diferentes del resto de grupos. Dado que algunos algoritmos de clasificación son sensibles a los parámetros iniciales se evalúa, mediante índices de validación, qué agrupaciones se ajustan mejor a los datos.

A cada uno de los grupos identificados se le aplican técnicas de procesamiento de lenguaje natural con la finalidad de identificar el sentimiento y los temas principales de cada uno de ellos.

Durante el desarrollo del trabajo se ha programado en R cada una de las partes para realizar las tareas planteadas, obteniendo los resultados que se exponen durante la memoria.

Palabras clave:

Bases de datos, reseñas, ETL, *clustering*, validación, estabilidad, criterio interno, minería de textos, lenguaje natural, sentimiento, tokenización, *topic modeling*.

Índice de Contenido

Índice de figuras

Índice de tablas

1. Introducción

En la actualidad la importancia y el impacto del *Word of mouth* (WOM), o lo que es lo mismo el boca a boca, es un concepto que ha recobrado importancia sobre todo para la gestión del marketing.

Tal como indica [1], el principal cambio en el mercado es el aumento de la oferta de canales de comunicación, mencionando especialmente el crecimiento de internet y las comunicaciones móviles en los últimos años, además de la creciente globalización. En las últimas décadas, internet ha facilitado a los consumidores el intercambio de información, cambiando la forma en que estos comparten sus opiniones. La típica comunicación boca a boca consiste en palabras habladas que se intercambian con un amigo o familiar en la comunicación cara a cara. En cambio, el boca a boca online consiste en la transmisión de opiniones y experiencias personales a través de la palabra escrita, siendo esto una ventaja para los consumidores al poder acceder a toda esta información desde casa.

El WOM no es un concepto nuevo, pero durante décadas se ha considerado un factor muy importante en el comportamiento de compra de los consumidores. Sin embargo, durante muchos años, las empresas lo han ignorado en gran medida. Actualmente el WOMM (*Word of Mouth Marketing*), o también llamado marketing boca a boca, se considera un factor decisivo en el éxito de una empresa, que como estrategia de marketing facilita las conversaciones y opiniones entre los consumidores. Dichas conversaciones y opiniones se traducen en lo que hoy en día llamamos reseñas, siendo posiblemente la referencia que más utilizan los clientes para buscar información sobre productos, servicios o empresas.

Existen numerosas herramientas en las que se utiliza el WOM, como es el caso de la compañía de ventas Amazon. Amazon posee actualmente una cantidad abrumadora de reseñas de productos de diversas categorías, que se han convertido para los consumidores en una pieza clave de los procesos de compra. Basándose en las reseñas de libros, estudios como [2] evidencian que el WOM del consumidor afecta al comportamiento de compra de otros clientes. También indica que las reseñas tienden a ser positivas y de mayor longitud en la página de

Amazon, teniendo un impacto positivo en las ventas, aunque por lo general las reseñas peor valoradas suelen tener un impacto más fuerte que las de mejor valoración.

Amazon recoge gran cantidad de reseñas, por lo que en [3] se expone la necesitad de la creación de una base de datos de gran tamaño de la cual extraer la mayor información posible. En ese estudio se buscó un método que representa las preferencias de los usuarios por la apariencia visual y un sistema que sea capaz de recomendar que ropa va mejor conjuntada y cual no. Numerosos son los estudios que se han basado en esos datos, como [4], que tiene el objetivo de estimar las funciones de clasificación de los usuarios basándose en sus anteriores opiniones.

Debido a la creciente cantidad de reseñas surge una nueva base de datos [5] más actualizada en la que se basará el proyecto actual, siendo de interés las reseñas y los metadatos de los productos de la categoría libros. Esta versión ofrece las siguientes características:

- Mayor cantidad de reseñas.
- Reseñas más actuales.
- *Metadata* más detallada de los productos y con más información.
- Mas categorías.

Ante la necesidad de clasificar las reseñas y entender las opiniones de los consumidores surgen otros estudios, como [6], donde se aplican algoritmos de *clustering* para agrupar por tema reseñas similares de un producto en concreto, como una marca de teléfono móvil.

El proyecto actual parte de un trabajo anterior [7] basado en la base de datos antigua donde se aplican técnicas de *clustering* y se desarrolla un modelo de predicción que relaciona ventas, precios, reseñas y votos útiles. La continuación de ese proyecto plantea el uso de la base de datos más actual y la extracción de información para formar un conjunto de datos basado en los libros y sus características. A este conjunto de datos se le aplicarán algoritmos de *clustering* y validación, y en base a estas agrupaciones se extraerá información relevante de sus reseñas mediante técnicas de minería de textos y procesamiento de lenguajes naturales.

1.1. Explicación del problema

Ante la necesidad de entender qué impacto pueden tener las opiniones de los consumidores en las ventas se necesita realizar un análisis profundo de las reseñas que estos escriben. La cantidad de reseñas que se encuentran en internet da lugar a un gran volumen de datos que está en constante crecimiento. En consecuencia, se busca como enfrentarse a esa gran cantidad de datos y extraer de ellos la información que interesa.

Otro requerimiento es poder identificar qué propiedades o características tienen esos datos y si existe relación entre ellos, por ello se busca cómo identificar estas relaciones mediante el uso de técnicas de *machine learning* que impliquen su agrupación. Para el agrupamiento se utilizan algoritmos que en algunos casos requieren de la elección de parámetros iniciales. Por lo tanto, es muy importante evaluar su calidad para saber si la elección es buena.

Otra dificultad es entender el texto no estructurado o semiestructurado escrito en lenguaje natural que componen el conjunto de reseñas. Para su comprensión, a pesar de ser un campo complejo, será necesarias la aplicación de técnicas de procesamiento del lenguaje natural.

1.2. Objetivos

En este proyecto se utilizan técnicas no supervisadas para la evaluación del impacto de las reseñas de libros en el rendimiento de las ventas y la caracterización de la utilidad de las reseñas de acuerdo con el análisis de su contenido y su contexto. El proyecto abordará el procesamiento y preparación de una base de datos, la aplicación de distintas técnicas de *clustering*, el análisis de su estabilidad y la realización de validación interna de los resultados. Adicionalmente, se realiza un análisis del sentimiento y de *topic modeling* en cada uno de los *clusters* identificados. Se combinarán diferentes técnicas de minería de textos y procesamiento de lenguajes naturales para extraer información relevante de las reseñas.

1.3. Organización de la memoria

Este documento está dividido en seis capítulos, siendo el primero la introducción que se acaba de ofrecer:

- En el segundo capítulo se analiza la metodología o plan de trabajo seguida durante el desarrollo del proyecto.
- En el tercer capítulo se expondrán los conceptos teóricos de las técnicas disponibles aplicadas a la resolución del problema planteado.
- En el cuarto capítulo se describe el desarrollo del proyecto comenzando por explicar el origen de los datos y su procesamiento, la aplicación de técnicas de aprendizaje no supervisado y la validación de resultados. Por último, se aplican de diferentes técnicas de minería de textos y procesamiento de lenguajes naturales para extraer información relevante de las reseñas.
- En el quinto capitulo se presentan las conclusiones alcanzadas durante la realización del proyecto en base a los resultados obtenidos.
- Finalmente, en el sexto capítulo se nombrarán posibles ideas de líneas futuras de trabajo relacionadas con este proyecto.

2. Metodología

A continuación, se expondrán los procedimientos que se han llevado a cabo para la realización de esta investigación y obtención de los objetivos propuestos.

La **primera fase** va enfocada a entender los objetivos del proyecto. Se hace una revisión bibliográfica y se recopila toda la información necesaria para abordar el proyecto. Dado que se trata de un trabajo de investigación la consulta bibliográfica se extiende a lo largo de todo el trabajo.

En la **segunda fase**, se realizó una investigación sobre las herramientas a utilizar, así como el aprendizaje autodidacta para obtener nociones básicas del propio lenguaje de programación, mediante cursos gratuitos online y manuales disponibles en internet.

Las tres siguientes fases van sujetas cada una de ellas a realizar una investigación del estado del arte de los métodos empleados en cada caso, ejecución de los experimentos mediante la programación de scripts en R, obtención y análisis de resultados.

Por lo tanto, en **la tercera fase** se realiza el análisis y procesado de las bases de datos de Amazon, incluyendo la base de datos de reseñas y los metadatos referentes a los libros que se combinarán para obtener un conjunto de datos de libros y reseñas. Se lleva a cabo el estudio de [7], además de la representación de los resultados. Obtener un conjunto de datos lo más limpio posible ha requerido bastante tiempo de trabajo, pero ha sido clave para la continuación de las siguientes fases.

En la **cuarta fase** se trabaja con los algoritmos de aprendizaje automático no supervisado empleados para la clasificación de los datos referentes a los libros, además del uso de algunos de los métodos de validación interna y de estabilidad que existen, su puesta en práctica, optimización de los algoritmos, repetición de los experimentos y comparación de los resultados obtenidos para elegir el mejor método de clasificación.

En la **quinta fase** se aplican diferentes técnicas de minería de textos a las agrupaciones obtenidas en la fase anterior. Se realizan tareas de preprocesamiento de texto y procesado de lenguaje natural.

La **sexta y última fase** es la elaboración de la propia memoria.

Cabe puntualizar que este trabajo no requiere de ningún costo de inversión, pero sí de poseer un equipo potente en el cual realizar las diferentes pruebas. Incluso tras cambiar a un equipo mejor durante la realización de este proyecto, al trabajar con bases de datos tan grandes y ejecutar algoritmos iterativos, los tiempos de espera son en ocasiones bastante elevados incluso provocando problemas por falta de memoria.

3. Estado del arte

El objetivo de este capítulo es realizar una revisión sobre las bases teóricas en las que se apoya este trabajo fin de máster. Por un lado, se ofrece una introducción al procesamiento de bases de datos, análisis de su contenido y almacenamiento, diferentes técnicas no supervisadas de *clustering*, su validación y el análisis mediante técnicas de procesado de lenguaje natural, del sentimiento y *topic modeling* de cada una de las agrupaciones identificadas.

3.1. Procesos ETL

La extracción, transformación y carga de datos o ETL (*extract, transform, load*) es el proceso de compilación de datos a partir de una o varias fuentes, su posterior organización y centralización en un único repositorio como una base de datos para facilitar el acceso y el análisis.

Antes de utilizar los datos para responder a los problemas planteados, lo más importante es prepararlos. Los datos suelen estar archivados en ficheros, y el uso de Excel o de editores de texto permite obtenerlos fácilmente. Sin embargo, los datos pueden encontrarse en distintas fuentes, como bases de datos, sitios web y diversos formatos de archivo. Ser capaz de importar datos de estas fuentes es crucial [8].

Existen diferentes tipos principales de datos, aunque los registrados en formato de texto son los más sencillos de extraer. Como algunos usuarios requieren almacenar los datos en un formato estructurado, se pueden utilizar archivos con extensión .tab o .csv para organizarlos en un número fijo de columnas. Durante muchos años, Excel ha tenido un papel destacado en el campo del procesamiento de datos utilizando los formatos .xls y .xlsx [8]. Hoy en día existe un número elevado de herramientas de análisis de datos que otorga un buen rendimiento y es fácil de usar, como por ejemplo el lenguaje R, siendo uno de los más utilizados en investigación científica. Por lo tanto, saber leer y manipular los datos de las bases de datos es otra habilidad crucial.

Sin embargo, no basta con recopilar datos, también hay que garantizar la calidad de los datos recopilados. Si la calidad de los datos utilizados es insuficiente, los resultados del análisis pueden ser engañosos debido a las muestras sesgadas o a los valores que faltan. Además, si los datos recogidos no están bien estructurados y formados, puede resultar difícil correlacionarlos e investigarlos. Por lo tanto, el preprocesamiento y la preparación de los datos es una tarea esencial que debe realizar antes del análisis de los datos. Algunos de los puntos más importantes que se deben tener en cuenta en la fase de preprocesamiento y preparación de datos son los siguientes [8]:

- Cambiar el nombre de la variable de datos
- Conversión de tipos de datos
- Trabajar con el formato de la fecha
- Añadir nuevos registros
- Filtrar y eliminación de datos
- Fusión de datos
- Ordenar datos
- Reformulación o remodelación de datos
- Detectar los datos que faltan
- Imputar los datos que falta

Durante la fase de preprocesamiento y dentro del texto no estructurado que se obtiene, a menudo nos interesa una información específica que puede incluir tanto cifras como letras, para ello se recurre al uso de expresiones regulares [9]. Una expresión regular es un patrón que define una cadena de texto con una particular estructura. Estas son muy útiles en programación en general, pero aquí se usarán para limpiar y preparar el texto, es decir para la búsqueda de patrones de caracteres y operaciones de sustitución.

3.2. Aprendizaje no supervisado

Los algoritmos de aprendizaje no supervisado basan su proceso de entrenamiento en un juego de datos no etiquetados. Con este método, podemos encontrar información nueva y no identificada previamente.

El aprendizaje no supervisado está dedicado a las tareas de agrupamiento, donde se busca estudiar la estructura de los datos, segmentando el conjunto de datos por atributos compartidos. En términos básicos, el objetivo de esta agrupación es encontrar diferentes grupos dentro de los datos proporcionados. Para ello, se usarán algoritmos de agrupamiento que encuentren una estructura en los datos, clasificándolos de manera que los elementos de un mismo grupo sean más similares entre sí que con los demás grupos (*clustering*).

El *clustering* es una tarea de aprendizaje automático no supervisado que divide los datos en clústeres o agrupaciones de elementos similares entre sí. Lo hace sin que se le diga de antemano qué aspecto deben tener los grupos. Como es posible que ni siquiera sepamos lo que buscamos, el *clustering* se utiliza para el descubrimiento de conocimientos más que para la predicción, proporcionando una visión de las agrupaciones naturales que se encuentran en los datos [10].

Existe una gran diversidad de algoritmos de *clustering* que permiten determinar las agrupaciones de los datos y encontrar la relación entre sus elementos. En el desarrollo de este proyecto se incluirán tres de los más representativos: K-means, DBSCAN y Jerárquico.

3.2.1. K-means

El algoritmo K-means [11] es quizás el método de agrupación más utilizado dentro del grupo de algoritmos de aprendizaje no supervisado, tratándose de un algoritmo iterativo. El agrupamiento se realiza minimizando la suma de las distancias entre cada objeto y el centro de su grupo. Inicialmente, se debe predefinir el número de grupos k, estableciéndose los centros de estos grupos de forma aleatoria. Este algoritmo consta esencialmente de dos fases. Como primer

paso, cada objeto de los datos se asigna al grupo con el centro más cercano, según la función de distancia euclídea (12):

$$dist(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (1)$$

En un segundo paso, los centros se recalculan como el valor medio de los puntos actualmente asignados a ese grupo y finalmente se repiten estos dos pasos hasta que los centros converjan.

Uno de los principales inconvenientes que ofrece este algoritmo es la necesidad de indicar previamente el número de grupos en los que se quieren clasificar los datos. Esta elección requiere un delicado equilibrio ya que, si se ajusta *k* a un valor muy grande, mejorará la homogeneidad de los *clusters*, pero al mismo tiempo se corre el riesgo de sobre ajustar los datos (*overfitting*).

Una aproximación inicial que ayudaría a elegir el número apropiado de grupos sería aplicar la técnica conocida como el método del codo. Con esto se intenta medir cómo cambia la homogeneidad o la heterogeneidad dentro de los clústeres para varios valores de k. Como se ilustra en la Figura 1, se espera que la homogeneidad dentro de los clústeres aumente a medida que se añaden clústeres adicionales, del mismo modo que la heterogeneidad también irá disminuyendo a medida que aumenta el número de grupos. El objetivo no es maximizar la homogeneidad o minimizar la heterogeneidad, sino observar en que punto de la gráfica se produce ese cambio brusco que nos dirá el número óptimo de *clusters* a seleccionar [10].

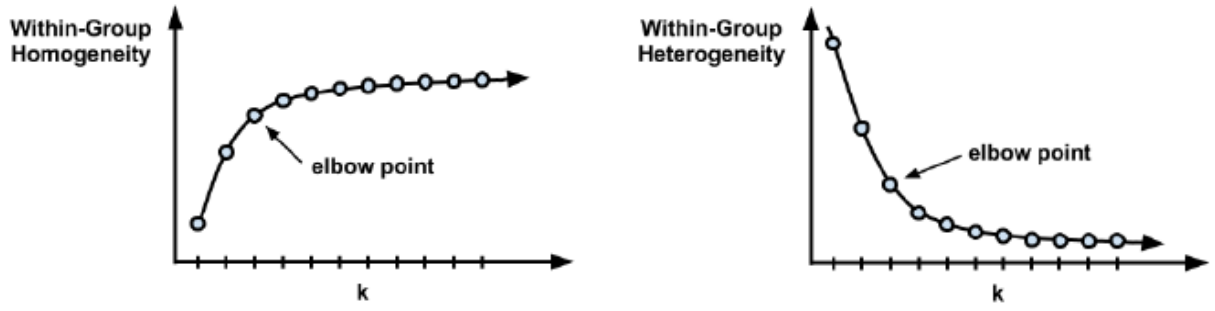


Figura 1. Método del codo para el valor óptimo de k. Fuente [10]

3.2.2. DBSCAN

DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) es un algoritmo de agrupamiento de datos basado en la densidad [12]. Este algoritmo permite identificar *clusters* determinando regiones con alta densidad de observaciones, separadas por regiones de baja densidad.

La característica más interesante del método DBSCAN es que es robusto a los valores atípicos (*outliers*). Además, no requiere que se diga de antemano el número de clústeres, a diferencia de K-Means, donde tenemos que especificar este dato.

El algoritmo DBSCAN requiere de dos parámetros:

- *Eps (Epsilon)*: especifica el radio que define la región en torno a un punto dado *p.*
- *MinPts (Minimum points):* es el número mínimo de puntos dentro de la región *Eps* que se requieren para definir un clúster.

Empleando ambos parámetros se pueden clasificar los datos en dos tipos de puntos: puntos dentro del propio clúster *(core points)* y puntos en el borde del clúster *(border points)*. Lo cual permite definir niveles de conectividad entre ellos:

- Directamente alcanzable (*direct density reachable*): un punto *p* es directamente alcanzable desde otro punto *q* si *p* forma parte del rango de vecindad de *q,* siendo *q* un *core point*.
- Alcanzable (*density reachable*): un punto *p* es alcanzable desde otro punto *q* si existe una secuencia de *core points* que van desde *q* hasta *p.*
- Densamente conectadas (*density conected*): dos puntos *p* y *q* están densamente conectados si existe otro punto *core point o,* tal que *p* y *q* son densamente alcanzables desde *o*.

El algoritmo funciona de la siguiente manera: para formar un *cluster* se comienza con un punto arbitrario *p* y se comprueba que todos los puntos densamente alcanzables desde *p* formen

parte del radio de vecindad *Eps* y contenga los suficientes puntos *MinPts*. Si *p* es un *core point*, se forma un clúster en función de *Eps* y *MinPts*. Si *p* es un *border point,* no hay puntos densamente alcanzables desde *p* y se pasa el siguiente punto de la base de datos. Los puntos que no pertenezcan a ningún clúster serán tratados como *outliers* o ruido.

A la hora de estimar los parámetros del algoritmo, el valor de *MinPts* suele ser el más fácil de ajustar. Según Ester et al. [12], se puede ajustar el valor de *MinPts* para el uso de datos de dos dimensiones manteniendo su valor por defecto *MinPts = 4*. Sin embargo, Sander et al. [13] sugieren establecer este valor como el doble de la dimensión del conjunto de datos, es decir *minPts = 2 * dim*. Para los conjuntos de datos que tienen mucho ruido, que son muy grandes o de alta dimensionalidad, puede mejorar los resultados aumentar el valor de *MinPts*.

Una vez conocido *MinPts*, se puede elegir el valor *Eps* mediante el uso del gráfico k-distancia. La idea es calcular las distancias de cada punto a sus *k* vecinos más cercanos siendo *k = minPts* [12]. De nuevo Sander et al. [13] sugiere usar un valor de *k = 2 * dim - 1*.

La representación de este gráfico se puede utilizar para ayudar a encontrar valores de parámetros adecuados para DBSCAN. Un buen valor para *Eps* sería donde este gráfico muestre una fuerte curvatura. Si se elige un valor de *Eps* demasiado pequeño, una gran parte de los datos no serán agrupados, mientras que para un valor de *Eps* muy alto, las agrupaciones se fusionan y la mayoría de los objetos estarán en el mismo grupo.

3.2.3. Algoritmo Jerárquico

El algoritmo jerárquico [14] es un método de agrupación de datos en el que los elementos quedan anidados en jerarquías con forma de árbol, de tal forma que mediante la matriz de distancias se muestra la relación de proximidad que existe entre ellos. Los métodos jerárquicos se clasifican en aglomerativos y divisivos.

- Los métodos aglomerativos comienzan el análisis con tantos grupos como elementos haya. A partir de esto se van formando grupos, cada vez mayores, hasta englobar a todos

los casos tratados en un mismo clúster. Generalmente estos algoritmos tienen una complejidad $O(n^3)$ lo cual los hace muy lentos para grandes bases de datos.

- Los métodos divisivos, constituyen el proceso inverso. Comienzan con un conjunto inicial que engloba a todos los elementos y mediante divisiones sucesivas, se van formando grupos cada vez más pequeños. Al final del proceso se tienen tantas agrupaciones como casos.

Los datos se agruparán en función de la distancia entre ellos, siendo de utilidad la distancia euclídea, máxima, manhattan y minkowski. Además de esto se debe definir como se mide la distancia entre clústeres por lo que se nombrarán algunas de las estrategias que serán empleadas a la hora de unir los clústeres:

- Enlace simple (*single linkage*): se usa la distancia entre los dos puntos más próximos que pertenecen a dos clústeres diferentes.
- Enlace completo (*complete linkage*): se usa la distancia entre los dos puntos más alejados de clústeres diferentes.
- Enlace promedio (*average linkage*): usa la proximidad promedio entre todos los pares de elementos que pertenecen a clústeres diferentes.
- Método centroid (*centroid method*): combina los clústeres con la mínima distancia entre los centroides de dos clústeres diferentes.
- Método Ward (*Ward's method*): usa como medida de proximidad la suma de los errores al cuadrado y tiene como objetivo minimizar la varianza total dentro del grupo de tal forma que se van uniendo los grupos con menor suma de errores al cuadrado.

3.3. Índices de validación

Una de las cuestiones más importantes en el análisis de *clustering* es la evaluación de los resultados obtenidos para encontrar la agrupación que mejor se ajusta a los datos.

Como ya se ha mencionado anteriormente el objetivo de los métodos de *clustering* es descubrir grupos significativos presentes en un conjunto de datos. En general, se deben buscar *clusters* cuyos miembros estén próximos entre sí, es decir, que tengan un alto grado de similitud y estén bien separados. Un problema al que nos enfrentamos al aplicar técnicas de *clustering* es decidir el número óptimo de *clusters* que se ajuste al conjunto de datos. Además, si a los parámetros del algoritmo de *clustering* se les asigna un valor inadecuado, el método de agrupación puede dar lugar a un esquema de partición que no es óptimo para el grupo de datos dado, lo que conduce a decisiones erróneas. Por lo tanto, se deben aplicar procedimientos de evaluación de los resultados de los algoritmos de *clustering* conocidos como técnicas de validación [15]. En términos generales, existen tres enfoques para investigar la validación de los *clusters*:

- El primero se basa en el criterio externo. Esto implica que se evalúan y comparan los resultados de un algoritmo de *clustering* basándose en resultados preestablecidos externamente.

- El segundo enfoque se basa en el criterio interno. Se evalúa que tan buena es la estructura de *clustering* basándose en la información interna de los datos (por ejemplo, la matriz de proximidad), sin necesidad de información ajena al propio algoritmo y su resultado.

- El tercer enfoque se basa en el criterio relativo. Aquí la idea básica es la evaluación de una estructura de *clustering* comparada con otras agrupaciones resultantes del mismo algoritmo, pero con diferentes valores de los parámetros.

Además de las tres medidas de validación mencionadas, se propone analizar la estabilidad de las agrupaciones mediante las medidas de estabilidad, que son una versión especial de las medidas internas, con las que se evalúan la consistencia de un resultado de *clustering* comparándolo con las agrupaciones obtenidas después de omitir cada vez una columna distinta del conjunto de datos.

En el presente trabajo se plantea escoger el mejor algoritmo de *clustering*, además del número óptimo de *clusters* sin tener ningún tipo de información adicional, por lo que las métricas de validación interna y estabilidad son la mejor elección para analizar en este caso.

3.3.1. Criterio interno

Existen diversas métricas de validación interna que se pueden usar para escoger el mejor algoritmo de *clustering* o el número óptimo de *clusters*. Para la validación interna se seleccionan medidas que reflejan la compacidad, la conectividad y la separación de las particiones de los *clusters*. El índice de Dunn [16] y el índice de Silhouette [17] son ejemplos de combinaciones no lineales de la compacidad y la separación, y que junto con la conectividad [18] comprenden las tres de las medidas internas que se mencionan a continuación. Dichos índices están disponibles en el paquete *clValid* [19] de R de validación de resultado de *clustering*.

Además de los índices mencionados anteriormente, otra medida que se analizará para la validación interna es el diagrama jerárquico que produce un algoritmo jerárquico mediante la matriz cofenética.

3.3.1.1. Conectividad

Se corresponde a la medida en que los elementos están situados dentro de un mismo clúster con respecto a sus vecinos más cercanos.

Sea N el número total de observaciones (filas) de un conjunto de datos y M el número total de columnas. Se define $nn_{i(j)}$ como el j vecino más cercano de la observación i, donde $x_{i,nni(j)}$ será cero si i y j están en el mismo cluster y $1/j$ en caso contrario. Entonces, para una determinada partición de cluster $C = \{C1, \ldots, CK\}$ de las N observaciones en K *clusters* disjuntos, la conectividad se define como:

$$Conn(\mathcal{C}) = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,nn_{i(j)}} \quad (2)$$

donde *L* es un parámetro que representa el número de vecinos más cercanos a utilizar. La conectividad tiene un valor entre 0 e infinito y debe minimizarse [19].

3.3.1.2. Coeficiente de Silhouette

El coeficiente de Silhouette es la media del valor de Silhouette de cada observación. El valor de Silhouette mide el grado de confianza en la asignación de la agrupación de una determinada observación, donde las observaciones bien agrupadas obtendrán valores cercanos a 1 y las observaciones mal agrupadas valores cercanos a -1. Para una observación *i*, se define como:

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)} \quad (3)$$

donde a_i es la distancia media entre *i* y todas las demás observaciones del mismo clúster, y b_i es la distancia media entre *i* y las observaciones en el "clúster vecino más cercano". Por lo tanto, el resultado del valor de la anchura de la Silhouette se encuentra en el intervalo [-1, 1] y debe ser maximizada.

3.3.1.3. Índice de Dunn

El índice de Dunn es la relación entre la menor distancia entre observaciones que no están en el mismo clúster y la mayor distancia intraclúster. Se calcula como:

$$D(\mathcal{C}) = \frac{\min\limits_{C_k, C_l \in \mathcal{C}, C_k \neq C_l} \left(\min\limits_{i \in C_k, j \in C_l} dist(i,j) \right)}{\max\limits_{C_m \in \mathcal{C}} diam(C_m)} \quad (4)$$

donde *diam(C_m)* es la distancia máxima entre las observaciones del *cluster* C_m. El índice de Dunn tiene un valor entre cero e infinito y debe ser maximizado.

3.3.1.4. Agrupación formada por un algoritmo jerárquico

Una forma que se puede utilizar para la definición de las agrupaciones es la distancia cofenética. El elemento $P_c(i, j)$ de la matriz cofenética representa el nivel de proximidad en el que los dos vectores x_i y x_j se encuentran por primera vez en el mismo *cluster*. El índice estadístico que mide el grado de similitud entre las matrices P_c y P (matriz de proximidad) se denomina *Coeficiente de Correlación Cofenética* [15] y se define como:

$$CPCC = \frac{(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}c_{ij} - \mu_p\mu_c}{\sqrt{\left[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} d_{ij}^2 - \mu_p^2\right]\left[(1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} c_{ij}^2 - \mu_C^2\right]}} \qquad (5)$$

$-1 \leq CPCC \leq 1$

Donde $M = N - (N - 1)/2$ y N es el número de puntos de un conjunto de datos. Además, μp y μc son las medias de las matrices P y Pc respectivamente, y vienen dadas por:

$$\mu_p = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P(i,j),\ \mu_C = (1/M)\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} P_c(i,j) \qquad (6)$$

Los valores del CPCC están comprendidos entre -1 y 1. Por lo tanto, cuanto más se acerque el índice CPCC a 1, mejor será la concordancia entre la matriz cofenética y la matriz de proximidad.

3.3.2. Medidas de estabilidad

Las medidas de estabilidad comparan los resultados del *clustering* del conjunto de datos completo basándose en la eliminación de cada columna del conjunto de cada vez. Estas medidas funcionan especialmente bien si los datos están altamente correlacionados. Las medidas incluidas son la *Average proportion of non-overlap* (APN), *Average distance* (AD), *Average*

distance between means (ADM) y *Figure of merit* (FOM). En todos los casos, el promedio se toma sobre todas las columnas eliminadas y todas las medidas deben minimizarse. Estas medidas también están disponibles en el paquete clValid [19] de R de validación de resultado de *clustering*.

3.3.2.1. Average proportion of non-overlap (APN)

El APN mide la proporción media de observaciones que no se agruparon en el mismo *cluster* al realizar la agrupación con en el conjunto de datos completo y la agrupación basada en los datos de los que se elimina una columna de cada vez. Sea $C^{i,0}$, el *cluster* que contiene la observación *i* usando el agrupamiento original (basado en todos los datos disponibles) y $C^{i,l}$, el *cluster* que contiene la observación *i* donde el agrupamiento se basa en el conjunto de datos con la columna *l* eliminada. Entonces ajustando K al número total de *clusters*, la medida APN se define como:

$$APN(K) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{\ell=1}^{M}\left(1 - \frac{n(C^{i,\ell} \cap C^{i,0})}{n(C^{i,0})}\right) \tag{7}$$

El valor APN está en el intervalo [0, 1], donde los valores cercanos a cero corresponden a resultados de agrupamiento altamente consistentes [19].

3.3.2.2. Average distance (AD)

La medida AD calcula la distancia media entre las observaciones que hay en el mismo *cluster* al realizar las agrupaciones en función del conjunto de datos completo, y en función de los datos con una sola columna eliminada. Se define como:

$$AD(K) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{\ell=1}^{M}\frac{1}{n(C^{i,0})n(C^{i,\ell})}\left[\sum_{i\in C^{i,0}, j\in C^{i,\ell}} dist(i,j)\right] \tag{8}$$

El AD tiene un valor entre cero e ∞, y se prefiere obtener valores más bajos [19].

3.3.2.3. Average distance between means (ADM)

La medida ADM calcula la distancia media entre los centros de los *clusters* de las observaciones ubicadas en el mismo *cluster*, mediante la agrupación en función del conjunto de datos completo y en función de los datos con una sola columna eliminada. Se define como:

$$ADM(K) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{\ell=1}^{M} dist(\bar{x}_{C^{i,\ell}}, \bar{x}_{C^{i,0}}) \quad (9)$$

donde $\bar{x}C^{i,0}$ es la media de las observaciones en el grupo que contiene la observación *i*, cuando el agrupamiento se basa en los datos completos, y $\bar{x}C^{i,l}$ se define de forma similar. Actualmente, ADM solo usa la distancia euclidiana. También tendrá un valor de entre cero e ∞, y nuevamente se prefieren obtener valores más pequeños [19].

3.3.2.4. Figure of merit (FOM)

El FOM mide la varianza media intra-cluster de las observaciones en la columna eliminada, donde la agrupación se basa en las muestras restantes (no eliminadas). Esto estima el error medio utilizando predicciones basadas en las medias del *cluster*. Para una columna *l* omitida en particular, el FOM es:

$$FOM(\ell, K) = \sqrt{\frac{1}{N}\sum_{k=1}^{K}\sum_{i\in C_k(\ell)} dist(x_{i,\ell}, \bar{x}_{C_k(\ell)})} \quad (10)$$

donde x_{il} es el valor de la i-ésima observación en la l-ésima columna en el cluster $C_k(l)$, y $\bar{x}C_{k(l)}$ es la media del *cluster* $C_k(l)$. Actualmente, la única distancia disponible para FOM es la euclidiana. El FOM se multiplica por un factor de ajuste $\sqrt{\frac{N}{N-K}}$, para paliar la tendencia a la disminución a medida que aumenta el número de *clusters*. La puntuación final se promedia sobre todas las

columnas eliminadas y tiene un valor entre cero e ∞, donde los valores más bajos equivalen a un mejor rendimiento [19].

3.4. Minería de textos

La minería de textos es una rama específica de la minería de datos que se refiere al proceso de extracción de información procesable o ideas útiles a partir de grandes cantidades de lenguaje no estructurado o semiestructurado, tal como correos electrónicos, redes sociales, reseñas de clientes, artículos de revistas, etc. Para alcanzar este objetivo se combinan diferentes técnicas de minería de textos [20] que comprenden el preprocesamiento del texto y la aplicación de técnicas de procesamiento de lenguaje natural incluyendo el análisis de modelado de temas (*topic modeling*).

3.4.1. Preprocesamiento del texto

A menudo los documentos de texto contienen datos textuales no estructurados, por lo que antes de la extracción de palabras de los documentos, es necesario aplicar una secuencia de tareas de preprocesamiento para convertirlos en una estructura de datos adecuada. El paso de preprocesamiento generalmente consiste en diversos puntos que a continuación, se describen brevemente.

3.4.1.1. Limpieza de texto y filtrado

La limpieza de texto agrupa tareas básicas como la expansión de las contracciones formadas en inglés y la eliminación de caracteres especiales y símbolos, o palabras que componen una mezcla de ambas.

El filtrado en documentos se realiza generalmente para eliminar algunas de las palabras que no son relevantes. Un filtrado común es la eliminación de las llamadas *stop words*, o palabras

vacías que carecen de significado por sí solas. Las palabras vacías aparecen con frecuencia en el texto sin tener mucha información de contenido (por ejemplo, artículos, preposiciones, conjunciones, etc.). Del mismo modo, las palabras que aparecen con bastante frecuencia en el texto pueden ser poco útiles para agrupar o relacionar información para distinguir los diferentes documentos. Por lo contrario, las palabras que aparecen muy raramente en los textos posiblemente tampoco tengan una relevancia significativa y pueden ser eliminadas.

3.4.1.2. Tokenización

La tokenización es la tarea encargada de dividir un texto en piezas más pequeñas o tokens. Un token es una unidad de texto que puede estar compuesto por palabras individuales, oraciones, párrafos, n-grams, etc. La tokenización también se conoce como segmentación de texto o análisis léxico.

En este paso también se eliminan signos de puntuación del texto, espacios en blanco innecesarios y se convierten todas las palabras a letra minúscula para facilitar su comparación.

3.4.2. Procesamiento de lenguaje natural

El procesamiento del leguaje natural (o NLP en sus siglas en inglés), es una rama de la inteligencia artificial y de la lingüística aplicada que permite investigar la manera de comunicar las computadoras con las personas mediante el uso de lenguajes naturales.

En términos general, con las tareas NLP se pretende dividir el lenguaje en piezas más cortas para intentar entender las relaciones entre ellas y si juntas crean algún significado. A menudo se utilizan diferentes recursos como el análisis del sentimiento, frecuencia de palabras y documentos (tf-idf) y el modelado de tópicos.

3.4.2.1. Análisis del sentimiento

Cuando un lector lee un texto utiliza su comprensión e intención emocional de las palabras para identificar si una sección del texto es positiva o negativa, o tal vez si está caracterizada por alguna otra emoción más matizada como la sorpresa o el asco. El análisis de sentimientos se utiliza para extraer de forma automática información sobre la connotación negativa o positiva del lenguaje de un documento.

Una forma de analizar el sentimiento de un texto es considerar el texto como una combinación de sus palabras de forma individual, y el contenido del sentimiento del texto completo como la suma del contenido del sentimiento de sus palabras individuales.

En el caso del análisis del sentimiento, se suele asignar una puntuación a cada uno de los textos o palabras en una escala continua o discreta para marcar el grado de sentimiento. Existe una gran variedad de métodos y diccionarios para evaluar la opinión y la emoción en un texto. Por ejemplo, en el paquete “tidytext” de R, los tres léxicos de uso general son [20]:

- El léxico NRC, clasifica las palabras de forma binaria ("sí"/"no") en categorías de positivo, negativo, ira, anticipación, asco, miedo, alegría, tristeza, sorpresa y confianza.
- El léxico Bing clasifica las palabras de forma binaria en categorías positivas y negativas.
- El léxico AFINN asigna a las palabras una puntuación que oscila entre -5 y 5, en la que las puntuaciones negativas indican un sentimiento negativo y las positivas un sentimiento positivo.

3.4.2.2. Análisis TF-IDF

Una cuestión central en la minería de textos y el procesamiento del lenguaje natural es cómo cuantificar de qué trata un documento. Una forma de hacerlo sería analizando la frecuencia de palabras y documentos. Una medida que expresa con qué frecuencia aparece una palabra en un documento es el *term frequency (tf)*. Existen palabras que pueden no parecer tan importantes en las reseñas como son las *stop words* mencionadas anteriormente, pero puede que estas

tengan más importancia en una reseña que en otra, por lo que eliminarlas no sería una buena idea. Otra medida para observar es el *inverse document frequency (idf)* la cual disminuye el peso de las palabras más comúnmente usadas y aumenta el de las menos frecuentes. Este término se define como [20]:

$$idf(term) = ln\left(\frac{n_{documents}}{n_{documents\ containing\ term}}\right) \quad (11)$$

Estos dos términos se combinan para calcular el *tf-idf* (la multiplicación de las dos) y se puede definir como la medida que expresa la importancia de una palabra para un documento en una colección de documentos, en este caso, reseñas. Este valor aumenta proporcionalmente al número de veces que la palabra aparece en una reseña, pero es compensada con la frecuencia de la palabra en la agrupación identificada de reseñas.

3.4.2.3. Modelado de temas

El modelado de temas o *topic modeling*, es un método de clasificación no supervisada que automáticamente encuentra los temas o grupos generales de palabras relacionadas dentro de un conjunto de documentos de texto.

Uno de los métodos más populares en el campo del modelado de temas es LDA (*Latent Dirichlet Allocation*) [21]. Este método trata cada documento como una mezcla de temas, y cada tema como una mezcla de palabras. Esto permite que los documentos se "superpongan" en términos de contenido, en lugar de estar separados en grupos discretos, de una manera que refleja el uso típico del lenguaje natural.

La Figura 2 muestra el modelo gráfico de representación del método LDA, donde se denota *M* como el número de documentos y *N* como el número de palabras por documento, siendo *w* una palabra específica y *z* el tema. El parámetro α representa la densidad de documentos y temas, mientras que β representa la densidad de palabras. Cuanto mayor sea el valor de α, de más temas se compondrán los documentos. Por otra parte, a mayor β, los temas se componen

de mayor cantidad de palabras en el corpus. Θ representa la distribución de temas del documento.

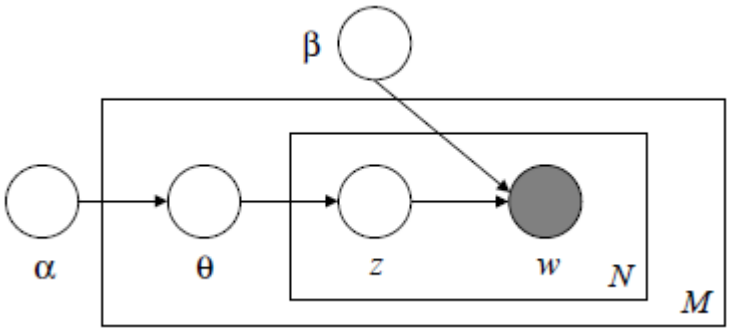


Figura 2. Representación del modelo gráfico LDA. Fuente [21]

3.5. R para minería y ciencia de datos

R es un entorno de software libre y lenguaje de programación, distribuido bajo licencia GNU (*GNU is Not Unix*, en inglés) con un enfoque estadístico y gráfico, con la posibilidad de ser ejecutado en una amplia variedad de plataformas como UNIX, Windows y MacOS [22]. Dispone de una gran variedad de documentación, además de diferentes bibliotecas o paquetes gracias a la gran comunidad activa que posee por formar parte de un proyecto colaborativo y abierto.

R es uno de los lenguajes de programación más utilizados en investigación científica, minería de datos, análisis estadístico, matemáticas financieras...etc. Siendo además muy popular en los campos de aprendizaje automático (*machine learning*). Para tareas de análisis y cálculo intensivas, el lenguaje R puede interactuar bien con otros lenguajes llamando en tiempo de ejecución a código escrito en C, C++ o Fortran. Adicionalmente, con R es posible ejecutar únicamente las líneas de código que se deseen en el orden que interese.

En cuanto a los entornos de trabajo, existen diferentes opciones, aunque la más conocida y utilizada hoy en día, además de tratarse de software libre, es RStudio. RStudio es un entorno de desarrollo integrado (IDE) para R que incluye una consola, un editor que resalta la sintaxis y

admite la ejecución directa del código, así como herramientas para el trazado, el historial, la depuración y la gestión del espacio de trabajo. También cabe mencionar otro entorno de desarrollo integrado como es IntelliJ IDEA, principalmente usado para el desarrollo de programas informáticos el cual permite el trabajo en gran cantidad de lenguajes mediante el uso de *plugins*. Esta herramienta no es gratuita.

Se nombran ambos entornos de desarrollo ya que han sido utilizados durante este proyecto, decantándose especialmente por el segundo debido a su aspecto visual y algunas de sus opciones que permiten organizar mejor el trabajo.

Existe una gran cantidad de paquetes desarrollados por la comunidad que se pueden descargar y permiten añadir y mejorar las funcionalidades base en R. A continuación, se nombrarán los que han sido utilizados durante este proyecto.

Un paquete que se usa a menudo y está diseñado especialmente para la ciencia de datos conteniendo funciones para la visualización, manipulación, limpieza e importación de datos es "tidyverse". Otros paquetes que se han empleado como herramientas de visualización, tanto en formato gráfico como en tablas son "gridExtra", "flextable" y "zoom".

Específicamente, en la parte de ETL para la lectura y manejo de los archivos de las bases de datos se emplearon paquetes como: "data.table", "rjson" y "reader".

Durante la fase de *clustering*, además de las funciones que ya incluye el propio R, se emplean otros algoritmos de clasificación, visualización de agrupaciones y medidas de validación, por lo que tienen especial mención los paquetes "dbscan", "factoextra" y "clValid".

Finalizando con la fase de minería de textos, tanto como para realizar tareas de limpieza de texto, tokenizar y aplicar el modelado de temas se ha requerido el uso de los paquetes: "tidytext", "textclean" y "topicmodels".

4. Desarrollo y resultados del trabajo

En este capítulo se explica de forma detallada cual es el origen de los datos, los pasos generales que se han dado para su procesamiento y visualización, las diferentes tareas llevadas a cabo en el proceso de *clustering*, pruebas realizadas, análisis, comparaciones y finalmente la aplicación de diferentes técnicas de minería de textos y obtención de resultados en cada caso.

4.1. Extracción, transformación y carga de datos

Como se ha mencionado con anterioridad el conjunto de datos utilizado en este proyecto proviene de la base de datos de reseñas de productos proporcionados por Amazon [5]. Este conjunto de datos, que recoge reseñas entre 1996 y 2018, es una versión actualizada del conjunto de datos de reseñas de Amazon publicado en 2014 y que es ampliamente utilizado por la comunidad investigadora llevando a cabo estudios como [23] que tiene como objetivo generar justificaciones convincentes y diversas a partir de reseñas y consejos extraídos de la base de datos, identificando segmentos de reseñas que pueden usarse como justificación y construyendo a partir de ellos un conjunto de datos de justificación personalizada.

En concreto, nuestro proyecto se basará en la categoría libros. Esta base de datos incluye reseñas (calificaciones, texto, votos útiles), metadatos de productos (descripciones, información de la categoría, precio, marca y características de imagen) y enlaces (gráficos de también visto/también comprado).

La base de datos de Amazon consta de un total de 51,311,621 reseñas de libros. Para este trabajo se han usado los datos de la revisión “5-core”, es decir el subconjunto de los datos en los que todos los usuarios y artículos tienen al menos 5 opiniones (en total 27,164,983 reseñas). Esta base de datos se trata de un subconjunto más pequeño de datos para uso experimental.

Además, se hace uso de los metadatos referentes a los libros, los cuales contienen descripciones, precio, ranking, información de la marca y enlaces de compra. Los metadatos constan de 2,935,525 productos.

Tal como muestra la Figura 3, el objetivo es combinar estos dos conjuntos de datos en otro que contenga los datos resumidos de cada libro, tanto para las reseñas como para el conjunto de metadatos.

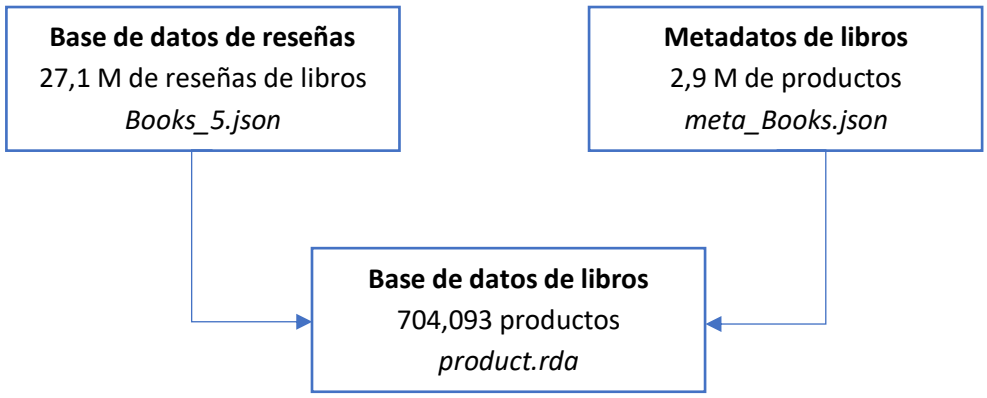


Figura 3. Metodología ETL

Por lo tanto, el preprocesamiento y la preparación de datos es una tarea indispensable que se debe realizar antes del análisis de datos. Durante el proceso ETL se han llevado a cabo diversos pasos que se expondrán en los siguientes apartados. A modo resumido se puede mencionar los siguientes puntos como los más importantes: importación y lectura de los datos de los archivos que forman el conjunto de la base de datos, limpieza del formato de los datos, siendo esta parte la que inicialmente ha sido más laboriosa y duradera al intentar conseguir un conjunto de datos limpio y organizado, y por último exploración y análisis de los datos obtenidos. Cabe señalar que, al tratarse de archivos de tan gran tamaño, el tiempo empleado en cálculos ha sido bastante elevado, así como los constantes problemas por falta de memoria.

4.1.1. Lectura y comprensión de la base de datos de reseñas

Los datos obtenidos de Amazon vienen en formato *json*, es decir se componen de una reseña por línea. Como primer paso, se cargarán los datos en el entorno de trabajo para su lectura y comprensión de su estructura y dimensiones. Cada línea tiene la forma mostrada en la Figura 4.

"{\"overall\": 1.0, \"vote\": \"6\", \"verified\": false, \"reviewTime\": \"11 15, 2005\", \"reviewerID\": \"A1E7G56OX03JKO\", \"asin\": \"0002005263\", \"style\": {\"Format:\": \" Mass Market Paperback\"}, \"reviewerName\": \"Ranch Girl\", \"reviewText\": \"If Tony Hillerman wrote this novel (novella actually), I sure can't tell. I have been a Chee/Leaphorn fan for many years, but this novel is not in the same league with Hillerman's other work. It is reminiscent of a 'B' movie. Instead of building suspense, the plot is laid out all too plainly, the characters are unreal, and the Chee/Bernie romance doesn't ring true. I am amazed that it was published.\", \"summary\": \"Who wrote this? Surely not Hillerman\", \"unixReviewTime\": 1132012800}"

Figura 4. Estructura de una línea del archivo en formato .json

- *asin*: identificación del producto
- *reviewerName*: nombre del revisor
- *vote*: votos útiles de la reseña
- *style*: información de los metadatos del producto, por ejemplo: "Format" es "Hardcover" (tapa dura)
- *reviewText*: texto de la reseña
- *overall*: calificación del producto
- *summary*: resumen de la reseña
- *unixReviewTime*: fecha de la reseña (formato unix)
- *reviewTime*: fecha de la reseña
- *image*: imágenes que los usuarios publican después de haber recibido el producto

Una vez cargadas las listas, se transformarán en tablas de datos para su posterior uso. Se muestra como ejemplo, en la Figura 5, la información de una reseña transformado a un objeto en R.

```
$overall
[1] 1

$vote
[1] "6"

$verified
[1] FALSE

$reviewTime
[1] "11 15, 2005"

$reviewerID
[1] "A1E7G56OX03JKO"

$asin
[1] "0002005263"

$style
$style$`Format:`
[1] " Mass Market Paperback"

$reviewerName
[1] "Ranch Girl"

$reviewText
[1] "If Tony Hillerman wrote this novel (novella actually), I sure can't tell. I have been a Chee/Leaphorn fan
for many years, but this novel is not in the same league with Hillerman's other work. It is reminiscent of a 'B'
movie. Instead of building suspense, the plot is laid out all too plainly, the characters are unreal, and the
Chee/Bernie romance doesn't ring true.  I am amazed that it was published."

$summary
[1] "Who wrote this?  Surely not Hillerman"

$unixReviewTime
[1] 1132012800
```

Figura 5. Reseña transformada a tabla de datos en R

En el trabajo desarrollado y dado el gran tamaño del archivo de datos y la cantidad de tiempo necesaria para su procesado, ha sido necesario leer los datos dividiéndolos en cuatro partes, optimizando el código para evitar en la medida de lo posible problemas de memoria.

Con respecto a la base de datos anterior (la del 2014) se ha eliminado el campo “image” que producía dos líneas por reseña dando lugar a datos duplicados, se modifican los cálculos asociados al antiguo campo “helpful” ya que Amazon cambió el método de votación, antes había

dos métricas para este campo (votos y votos útiles) y ahora sólo una “vote” (votos útiles) el cual se limpia de caracteres y se convierte a valor numérico.

Además de los puntos anteriores, el script empleado para el proceso ETL también realiza las siguientes tareas. La carga de las listas de datos, eliminando las filas duplicadas que son producidas por algunos campos. La conversión de las listas a tablas de datos. Transformación de algunos campos como el formato de fechas. Reorganización de las columnas de las tablas. Eliminación de objetos específicos que ya no se usan en los cálculos y están ocupando memoria. Finalmente, la unión de las partes anteriormente fraccionadas en un solo archivo.

4.1.1.1. Información y visualización de los patrones que siguen los datos

El análisis de las reseñas puede proporcionar información sobre que patrones siguen los datos. Desde una perspectiva temporal, el número de reseñas diarias aumenta considerablemente a partir del año 2012 alcanzando un gran volumen en los últimos años como se muestra en la Figura 6.

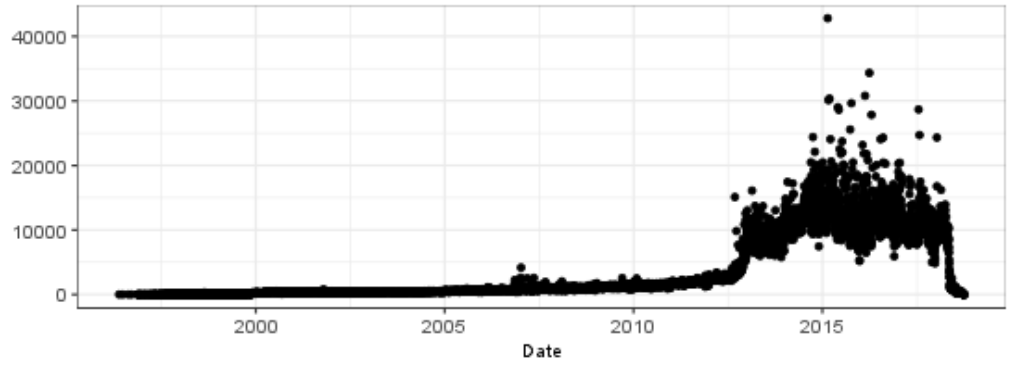


Figura 6. Evolución diaria del número de reseñas

En cuanto al número de votos de los productos, están distribuidos principalmente entre los valores 4 y mayormente 5, siendo esta última la puntuación de una gran cantidad de los productos (Figura 7).

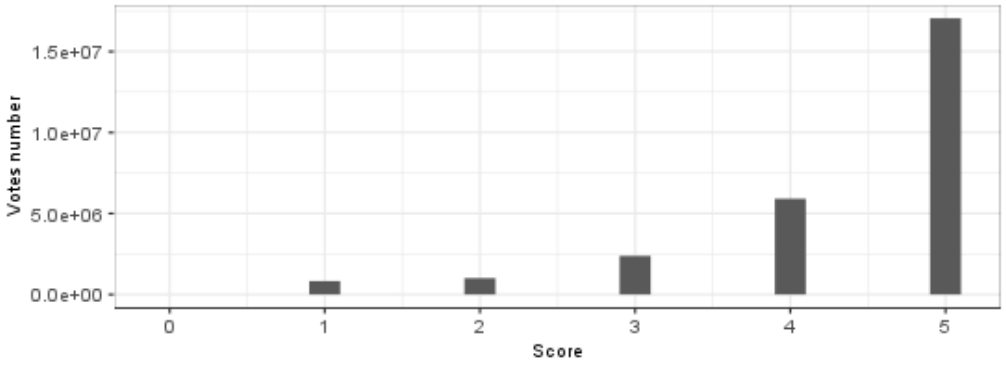


Figura 7. Distribución de la calificación de los productos

El número de votos útiles presenta una distribución muy sesgada, con un número reducido de reseñas con gran cantidad de votos como se observa en la Figura 8, también se aprecia que existe una gran cantidad de reseñas que no han recibido ningún voto como útil.

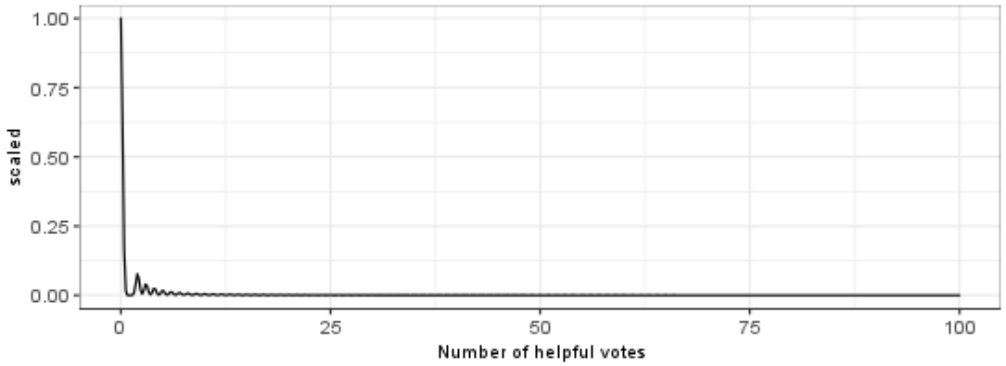


Figura 8. Función de densidad del número de votos útiles

Se procede a eliminar todas las reseñas que no han recibido ningún voto, para poder analizar como estarán distribuidos los votos. La Figura 9, muestra que en esta distribución habría una gran cantidad de reseñas que habrían recibido al menos un voto y habrá menor cantidad de reseñas que obtienen más votaciones.

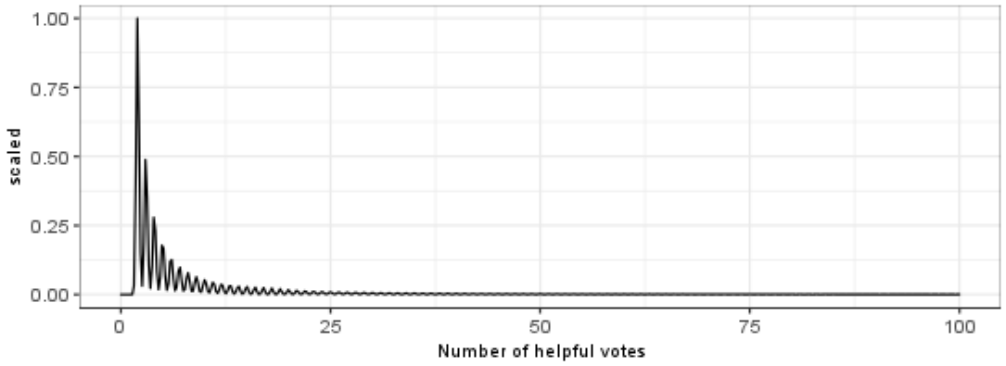


Figura 9. Función de densidad del número de votos útiles de reseñas votadas

La distribución del número de reseñas por libro también está muy sesgada, como era de esperar. Sólo unos pocos libros tienen un número muy elevado de reseñas (Figura 10).

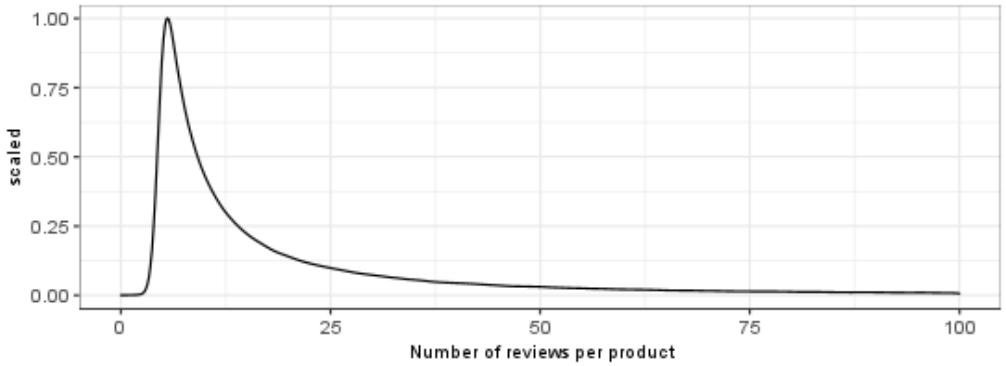


Figura 10. Función de densidad del número de reseñas por producto

Del mismo modo, el número de reseñas por revisor sigue una distribución similar, lo que significa que un número muy bajo de revisores tiene una gran cantidad de las reseñas (Figura 11).

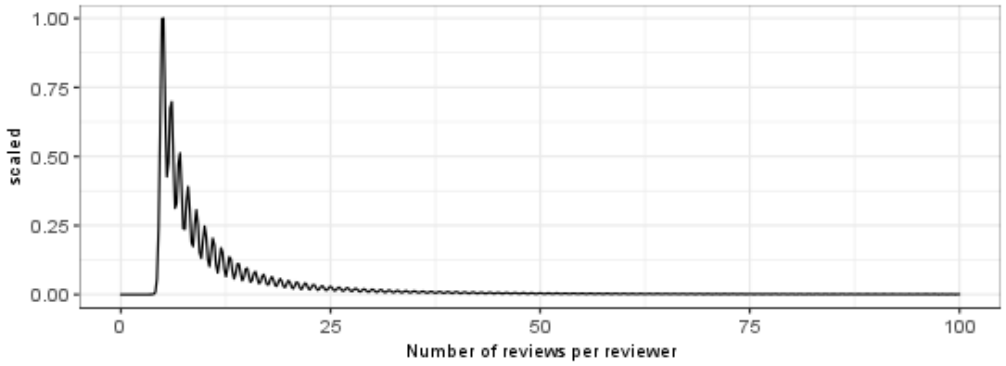


Figura 11. Función de densidad del número de reseñas por revisor

4.1.2. Lectura de los metadatos de libros

Para acceder a la información de este archivo se usará un procedimiento similar al mencionado anteriormente, ya que se trata también de un archivo *json*, formado por listas que deberán ser transformadas a tablas de datos. Esta tabla de datos está compuesta de los siguientes campos:

- *asin*: identificación del producto
- *title*: nombre del producto
- *feature*: características del producto en formato de viñetas
- *description*: descripción del producto
- *price*: precio en dólares estadounidenses (en el momento de la compra)
- *imageURL*: url de la imagen del producto
- *imageURLHighRes*: url de la imagen del producto en alta resolución
- *related*: productos relacionados (también comprados, también vistos, comprados juntos, comprados después de verlos)
- *salesRank*: información sobre el rango de ventas
- *brand*: nombre de la marca

- *categories*: lista de categorías a las que pertenece el producto
- *tech1*: la primera tabla de detalles técnicos del producto
- *tech2*: la segunda tabla de detalles técnicos del producto
- *similar*: tabla de productos similares

Se obtiene como resultante un archivo formado por listas anidadas, el cual necesita ser depurado. Del mismo modo que se ha realizado con el conjunto de datos de reseñas, se transforma en tablas de datos para uso posterior.

Se realizan algunas modificaciones como la eliminación de campos que evitan vincular las tablas en la lista anidada que producen sublistas de diferente longitud, como por ejemplo "details". Se trasforma la lista en un marco de datos para evitar inconsistencias (los marcos de datos son listas de vectores de igual longitud que no tienen que ser del mismo tipo). Se elimina de la base de datos todos aquellos productos que contienen en su precio el valor cero y se sustituye por NA (valor no disponible), ya que se considera un campo de los datos de valor ausente y para los análisis posteriores se desea tener en cuenta este valor.

A la tarea de limpieza de datos se le añade la eliminación de algunos caracteres como las comas y el símbolo del dólar del campo precio. Además, se requiere el uso de expresiones regulares para extraer el valor del ranking de los libros. Las expresiones regulares nos proporcionan una sintaxis para acceder sistemáticamente a patrones en el texto, es decir se usan para la detección, extracción y sustitución de patrones. Por ejemplo, es de esperar que el campo "rank" contenga solo números enteros, sin embargo, se muestra como una mezcla de símbolos, palabras y números que hay que limpiar y convertir, como muestra la siguiente Tabla 1.

Datos originales "rank"	Resultado
#1,057,594 Paid in Kindle Store (	1057594
#1,225,957 in Health & Household (	1225957
# 643,626 in Books (	643626

Tabla 1. Aplicando expresiones regulares al campo "rank"

Se utilizarán las características disponibles que más interesan, en este caso: *asin, rank y price.* Por último, del conjunto de datos que se obtiene, se seleccionan solo las filas que son únicas para evitar datos duplicados.

4.1.3. Fusión de ambas bases de datos

Se combinan ambos conjuntos de datos (reseñas y metadatos de libros) en otro que mantenga solo la información relevante, partiendo de la variable clave "asin" perteneciente a ambos conjuntos y siendo el patrón a través del cual se relacionan ambas bases de datos. El número "asin" de los libros es el mismo que el código estándar IBSN (International Book Serial Number) utilizado para identificar los libros en las bases de datos de las bibliotecas.

La base de datos resultante tras la fusión y filtrado contiene un total de 704,093 libros y está caracterizada por los siguientes campos:

- *asin*: número de identificación estándar de Amazon
- *salesRank*: ranking de ventas de Amazon del libro
- *price*: precio del libro en USD
- *reviews_n*: número de reseñas recibidas
- *score_mean*: puntuación media otorgada al libro
- *votes_sum*: suma total de votos útiles de reseñas de cada libro
- *votes_mean*: número medio de votos por reseña
- *votes_pearson*: medida de la dispersión del número de votos por revisión (desviación estándar)

4.2. Algoritmos de clustering

En este apartado se trabajará con el conjunto de datos “product.rda”, al que se aplicarán los algoritmos de *clustering* mencionados con anterioridad. Lo que se busca es encontrar una relación entre los siguientes elementos: *rank, price, reviews_n, score_mean y votes_mean*.

Inicialmente se realiza una exploración y preparación de los datos. En los datos numéricos, se encuentran algunos valores definidos como NA (not available). La función kmeans incluida en R no funciona con NAs, por lo que se eliminan todas aquellas muestras a las que les falten algún dato para evitar incongruencias en los resultados o posibles alteraciones y mantener sólo observaciones completas, lo que reduce el conjunto a 505,536 libros. De esta cantidad de datos nos vamos a quedar con un conjunto aleatorio formado por 35,000 libros, el motivo recae de nuevo en la cantidad de tiempo empleado ejecutando algunos algoritmos como el jerárquico, cálculos de los índices de validación y estabilidad, además de problemas por falta de memoria, sobre todo durante los cálculos de la correlación entre la distancia cofenética y la distancia original.

Adicionalmente, para entrenar el modelo con los datos, una práctica común empleada antes de cualquier análisis que utilice cálculos de distancia es normalizar o estandarizar con puntuación z (z-score) las características para que cada una utilice el mismo rango [10]. De este modo, se puede evitar el problema de que algunos rasgos lleguen a dominar únicamente porque tienen un rango de valores mayor que los demás.

El proceso de estandarización puntuaciones z, reescala los rasgos para que tengan una media de cero y una desviación estándar de uno. Esta transformación cambia la interpretación de los datos de una manera que puede ser útil en este caso.

Aplicando K-means

Con este algoritmo se tiene como objetivo minimizar las distancias dentro de cada *cluster* y maximizar las distancias entre *clusters*. Previamente se pretende determinar el número apropiado de clústeres que se usarán en el análisis.

En [7] se justifica la elección del número de grupos *k* usando la suma de errores al cuadrado SSE (*sum of squared errors*), es decir la relación entre la suma de cuadrados entre clúster y la suma total de cuadrados. En el trabajo desarrollado se utiliza la suma de los cuadrados de las distancias de cada elemento de datos con su centroide correspondiente, WCSS (*Within Cluster Sum of Squares*) y se denota de la siguiente manera:

$$WCSS = \sum_{i \in n} (X_i - Y_i)^2 \tag{12}$$

donde Y_i es el centroide del elemento X_i y n el total de datos en la muestra. Adicionalmente, se han aumentado el número de iteraciones del algoritmo y la cantidad de conjuntos aleatorios de centros que se eligen inicialmente, obteniendo mejores resultados.

En las Figuras 12 y 13 se observa que independientemente del método utilizado se obtiene el mismo resultado. A partir de 7 *clusters* la reducción en la suma total de diferencias internas parece estabilizarse, indicando que *k = 7* es una buena opción para iniciar el análisis. En la Figura 14 se representa la agrupación obtenida mediante K-means con 7 *clusters*.

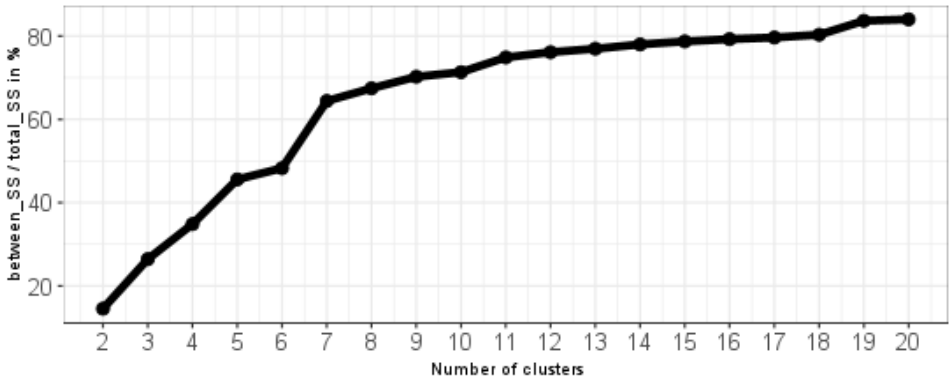


Figura 12. Método del codo usando SSE

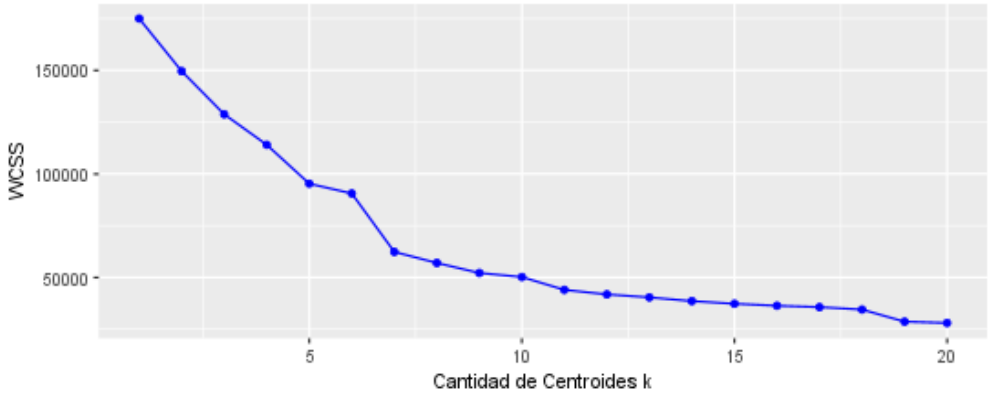


Figura 13. Método del codo usando WCSS

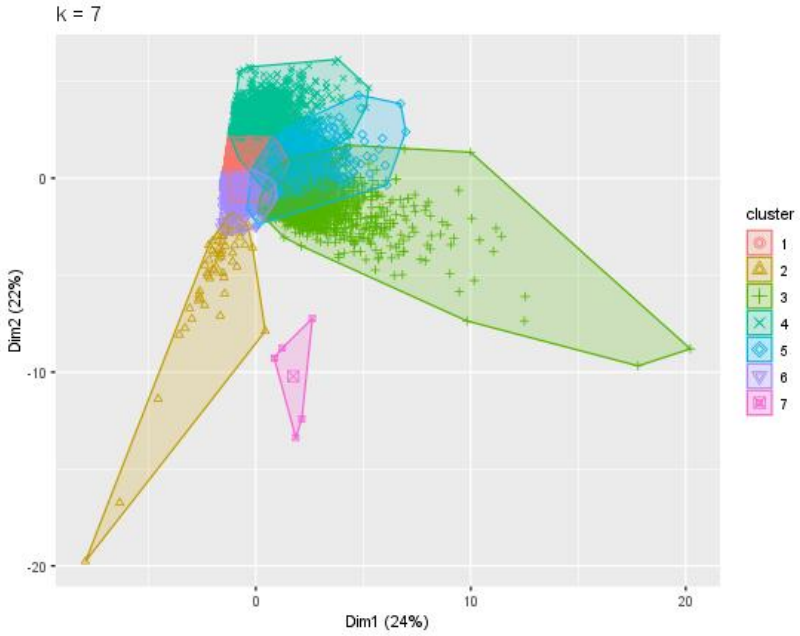


Figura 14. Agrupación k-means para k=7

Aplicando DBSCAN

Inicialmente se busca determinar los valores de *Eps* y *MinPts* [13]. El conjunto de datos posee una dimensión de 5, por lo que se ajusta el valor de *MinPts* al doble de la dimensión del conjunto, es decir *MinPts = 10*. Una vez conocido *MinPts*, se puede elegir el valor *Eps* mediante el uso del gráfico k-distancia, siendo *k = 9*. En la Figura 15 se aprecia un "codo" claramente visible y parece más apropiado establecer el corte en torno al valor *Eps = 1.25* de la curva.

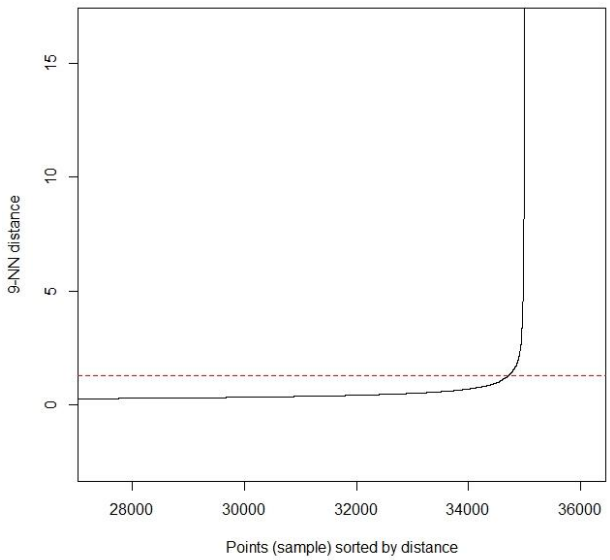


Figura 15. Gráfico k-Nearest Neighbor Distance

Al ejecutar el algoritmo se obtiene como resultado que la agrupación contiene 1 clúster y 179 puntos de ruido, como muestra la Figura 16.

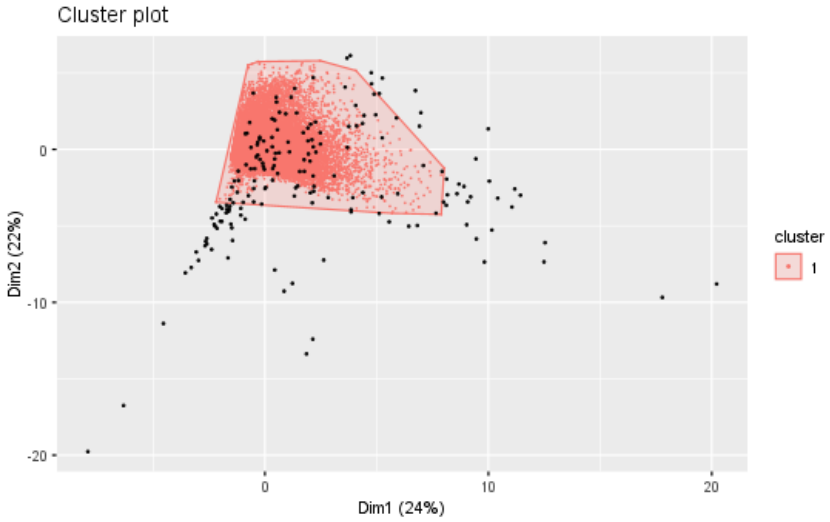


Figura 16. Agrupación DBSCAN

Este algoritmo no ofrece resultados óptimos a la hora de clasificar los datos en grupos bien diferenciados. Se obtienen generalmente agrupaciones similares al variar ligeramente los valores asignados a *Eps* y *MinPts*, produciendo como resultado un gran clúster que engloba a la mayoría de las muestras y algún cluster muy pequeño o ninguno que no parecen arrojar mucha información. Incluso eliminando los valores de ruido se obtiene resultados similares de cada vez.

Aplicando el algoritmo jerárquico

En este paso, se analiza el algoritmo jerárquico de tipo aglomerativo. Para realizar una primera aproximación en la definición de las agrupaciones se calculará el coeficiente de correlación cofenética. Con esto, se buscan valores próximos a 1, lo que indicaría que el resultado del *clustering* es aceptable. Por lo tanto, eligiendo diferentes métodos de *linkage* y diferentes distancias se obtienen los valores que se muestran en la Tabla 2.

Distancia/linkage	Average	Complete	Single	Ward.D	Centroid
Euclidean	**0.91918108**	0.82901276	0.78010115	0.21900264	0.89694041
Maximum	**0.9235064**	**0.9123261**	0.77082664	0.28224466	0.90200545
Manhattan	0.88514416	0.7604292	0.74785182	0.23524441	0.85234819
Minkowski	**0.91918108**	0.82901276	0.78010115	0.21900264	0.89694041

Tabla 2. Valores del coeficiente cofenético

Como resultado se aprecia que los valores más próximos a uno son los resultantes de aplicar el método *average* y en particular, la distancia *máximum*.

A continuación, se representan los resultados mediante el dendrograma (Figura 17). No existe una norma fija para establecer cuántos grupos pueden considerarse, pero el dendrograma puede servir de ayuda visual para determinar dicho número. Dependiendo del coeficiente de

proximidad usado el aspecto visual variará, por lo que se traza una división (línea roja) que corta el árbol y genera el número de *clusters* que se considera más apropiado.

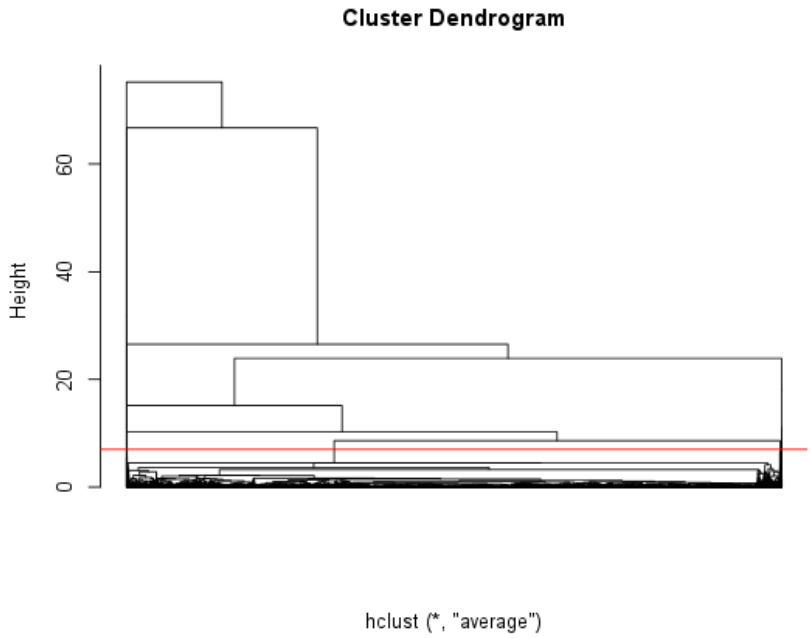


Figura 17. Método "average" y distancia "maximum"

A partir de la Figura 17 y de la información mostrada en la Tabla 3 se determina que hay varias agrupaciones de un solo elemento o muy pequeñas, es decir hay *clusters* con un único elemento muy distantes del resto de grupos.

cluster	nº muestras
1	34824
2	30
3	108
4	5
5	15
6	7
7	2
8	3
9	1
10	1
11	2
12	1
13	1

Tabla 3. Elementos por cluster con el método jerárquico

Se ha observado un comportamiento similar en las clasificaciones que muestran los dendrogramas al utilizar las distancias *euclidian* y *minkowski*. Adicionalmente se repite el mismo comportamiento aplicando el método de linkage *complete.*

Una cuestión importante es la detección de valores atípicos en los datos (ver Figura 18). Los valores atípicos son aquellas observaciones cuyos valores son muy diferentes a otros conjuntos de observaciones, es decir que son numéricamente distantes al resto de datos. En consecuencia, se procede a la eliminación de dichos valores que conforman grupos de 3 o menos componentes y a la repetición de los experimentos.

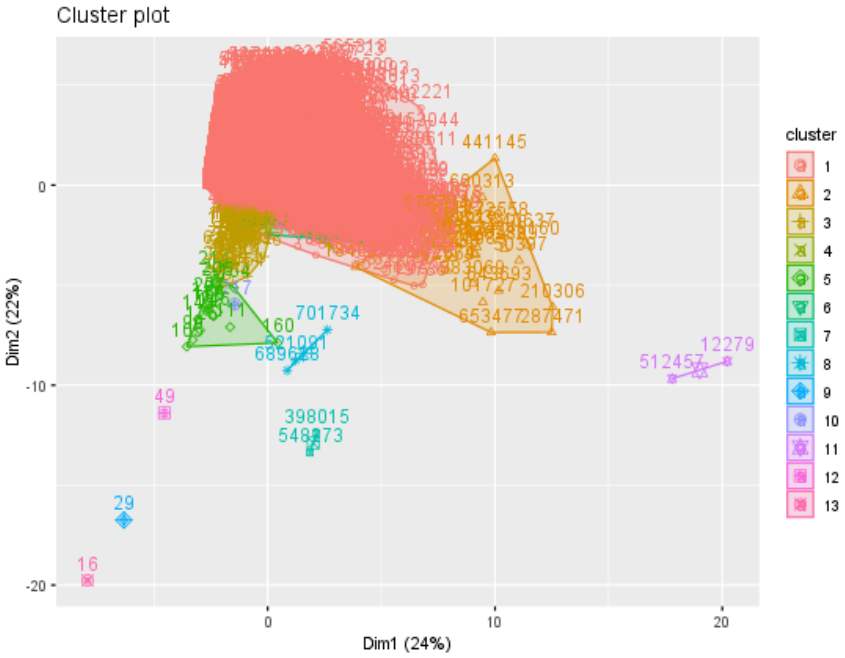


Figura 18. Detección de valores atípicos en el clustering jerárquico

4.3. Métodos de validación

Una vez realizado el agrupamiento hay que comprobar la calidad de los grupos. Para ello se aplicarán los índices de validación interna y de estabilidad que están disponibles en el paquete *clValid* [19] de R.

Validación interna

Se especifican los métodos de agrupación empleados, en este caso k-means y jerárquico y el número de *clusters* de los que se quiere obtener los valores, eligiendo entre 2 y 8. Como resultado se muestra en la Figura 19 el valor de los índices para ambos algoritmos en función de la cantidad de clústeres especificados.

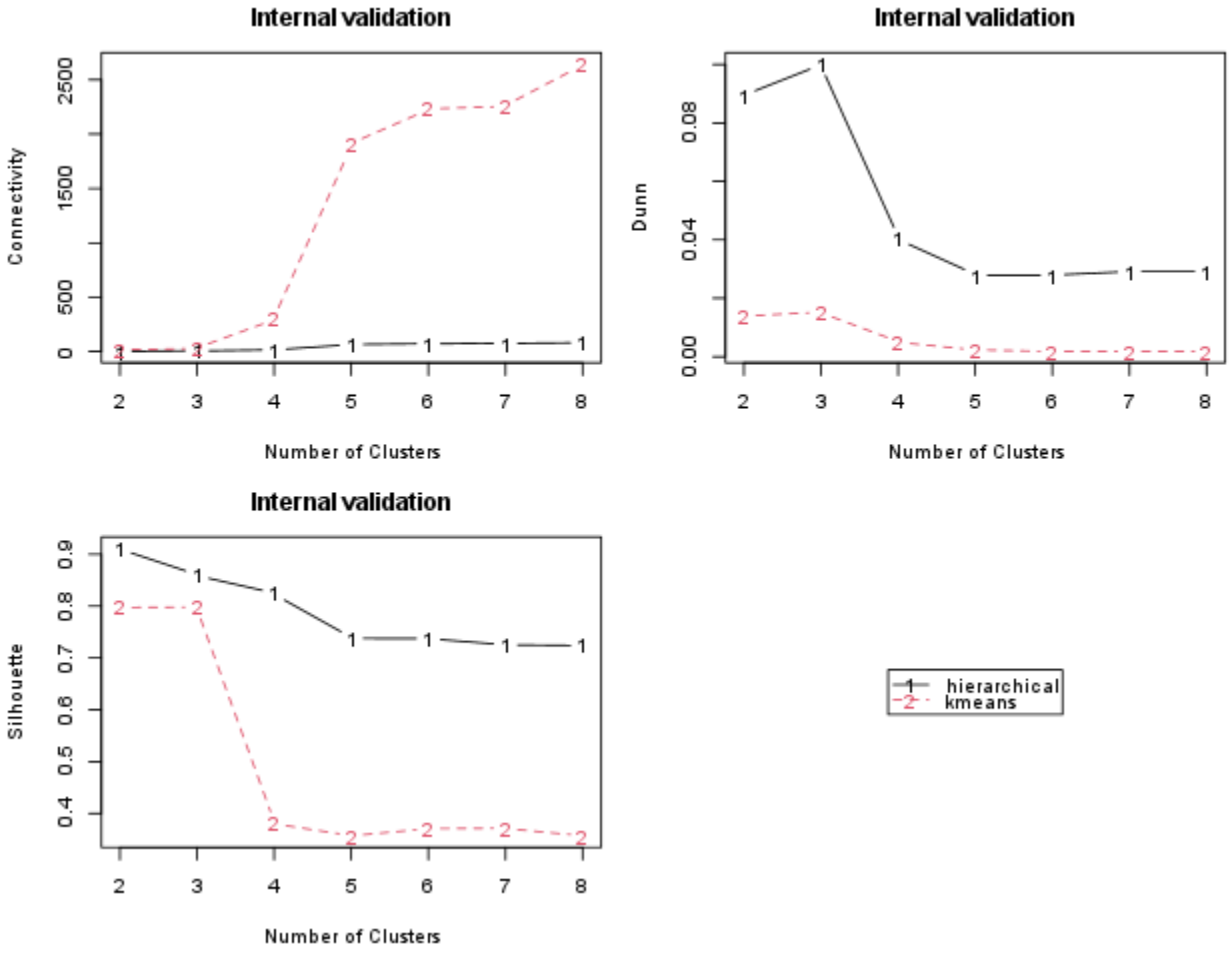


Figura 19. Valores de los índices de validación interna, Connectivity, Dunn y Silhouette

Como es de esperar los valores de los índices varían en función del número de *clusters* especificados. Recordemos que la conectividad debe ser minimizada, mientras que tanto el índice de Dunn como el de Silhouette deben ser maximizados. Se observa que el algoritmo k-means muestra un empeoramiento extremo en cuanto a la conectividad al llegar a 5 *clusters* y en cuanto

a Silhouette al llegar a 4, sin embargo, en Dunn esta variación no es tan grande, aunque empeora también a partir de 4 *clusters*. Por lo contrario, el algoritmo jerárquico muestra estabilidad en cuanto a la conectividad y un empeoramiento al llegar a 4 y 5 *clusters* según el índice Dunn y Sihouette respectivamente. La Tabla 4 presenta el método de agrupación y el número de *clusters* correspondientes a la puntuación óptima de cada índice. La agrupación jerárquica con dos y tres *clusters* es la que mejor funciona en cada caso.

Índice	Puntuación	Algoritmo	Nº clusters
Connectivity	2.8619	jerárquico	2
Dunn	0.1001	jerárquico	3
Silhouette	0.9110	jerárquico	2

Tabla 4. Mejor puntuación sobre los índices internos

Validación de la estabilidad

Las medidas de estabilidad incluyen el APN, AD, ADM y el FOM. Estas medidas deben minimizarse en cada uno de los casos. El cálculo de la validación de la estabilidad requiere mucho más tiempo que el de la validación interna, ya que hay que volver a hacer la agrupación para cada uno de los conjuntos de datos con una sola columna eliminada. La Figura 20 muestra la representación gráfica de estas medidas para los algoritmos k-means y jerárquico con un número de *clusters* seleccionado entre 2 y 8.

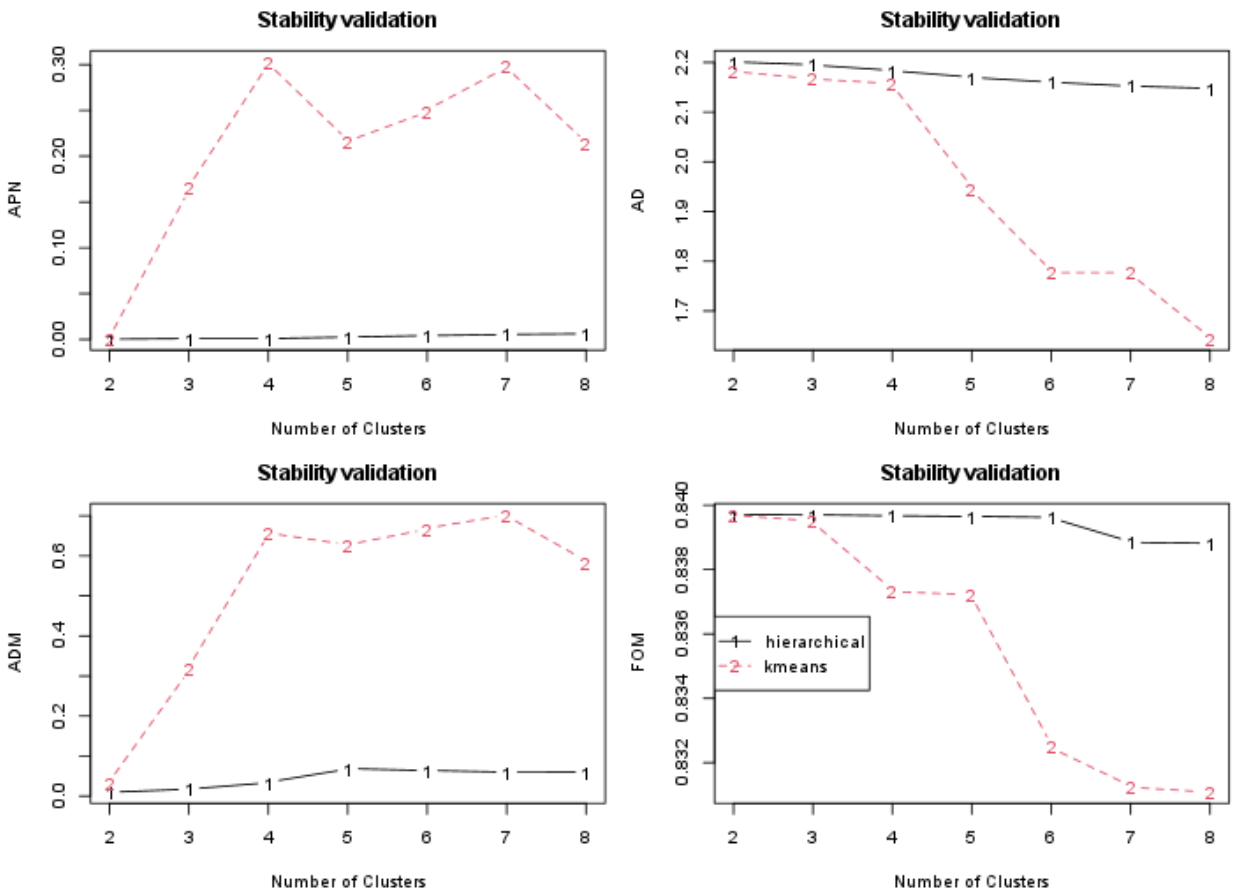


Figura 20. Valores de las medidas de estabilidad, APN, AD, ADM y FOM

Resulta interesante que en cualquiera de los casos el algoritmo jerárquico no muestre apenas oscilaciones en los valores de estabilidad. Sin embargo, k-means fluctúa más presentando mejores valores de estabilidad con el FOM y AD, y muchos peores mediante con el APN y ADM. La Tabla 5 presenta el método de agrupación y el número de *clusters* correspondientes a la puntuación óptima de cada medida.

Medida	Puntuación	Algoritmo	Nº clusters
APN	0.0002	jerárquico	2
AD	1.6433	k-means	8
ADM	0.0100	jerárquico	2
FOM	0.8311	k-means	8

Tabla 5. Mejor puntuación sobre las medidas de estabilidad

4.4. Resultados del clustering

La evaluación de los resultados de la agrupación puede ser algo subjetiva. No siempre es sencillo determinar qué tipo de algoritmo clasifica mejor los datos ya que según como se interpreten los resultados obtenidos, se ajusten los parámetros...etc. se llegará a conclusiones diversas. En este caso los índices de validación interna y de estabilidad nos ayudan a modo orientativo a buscar un número óptimo de *clusters* a utilizar, aunque finalmente en base a la interpretación de los resultados obtenidos se especificará la elección de un algoritmo u otro y la cantidad de grupos clasificados.

Tras diversas pruebas se ha llegado a diferentes resultados. Inicialmente se ha descartado el uso del algoritmo de DBSCAN, ya que la clasificación en función de la densidad de los datos no ofrece los resultados deseados, obteniendo un único *cluster* de gran tamaño y otro de ruido, o también algún *cluster* secundario muy pequeño. Incluso tras la eliminación de estos valores de ruido y la repetición del proceso de clasificación se vuelve a obtener un solo *cluste*r y nuevamente otro grupo con tan solo valores de ruido.

Posteriormente, se eliminan los valores atípicos que se detectaron al aplicar el algoritmo jerárquico y se repiten las pruebas. En el proceso de validación se obtuvieron las mejores puntuaciones aplicadas al algoritmo jerárquico, principalmente para dos *clusters*. Esta agrupación se descarta ya que se engloba todo el conjunto en un gran grupo y adicionalmente otro muy pequeño, por lo que se están omitiendo otras agrupaciones las cuales se pueden diferenciar más por las características de sus elementos.

Según los criterios de validación, el algoritmo K-means obtiene generalmente peores puntuaciones. Se han hecho pruebas con distintos números de *clusters*, desde 3 hasta 8 pero se obtienen grupos en los que sus valores están muy promediados.

Finalmente se ha buscado analizar los grupos formados en función de los valores agrupados y sus características para detectar realmente indicadores de que posee un campo que lo hace único con respecto a los demás. En última instancia, el éxito o el fracaso del modelo depende de si las agrupaciones son útiles para su objetivo, por ello basándose en que el algoritmo Jerárquico

está generalmente mejor valorado por los índices que el K-means, se buscan clasificaciones que destaquen por su característica más saliente.

Tras recalcular los valores del coeficiente cofenético para el algoritmo jerárquico se obtiene el máximo valor para un linkage *average* y distancia máxima. En la Figura 21 se representa el dendrograma, donde la línea de corte divide al conjunto en 5 grupos.

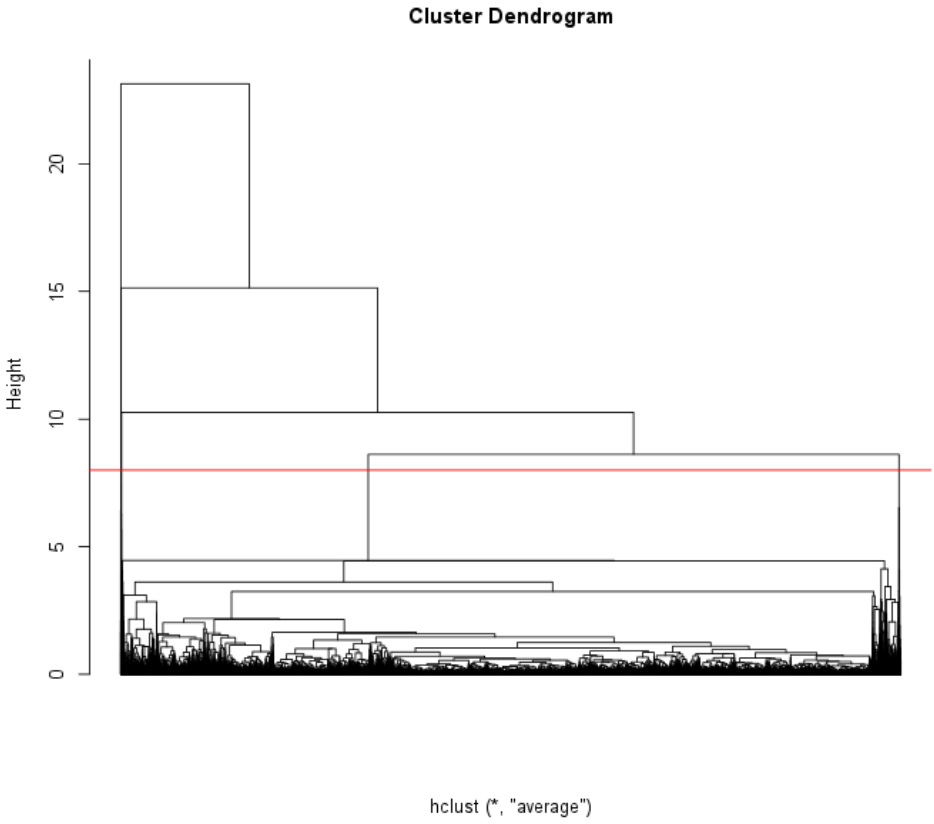


Figura 21. Clustering jerárquico con método ”average” y distancia "maximum"

Adicionalmente se muestra la representación de los datos en el plano, para una mayor apreciación visual de la distribución de las agrupaciones formadas (Figura 22).

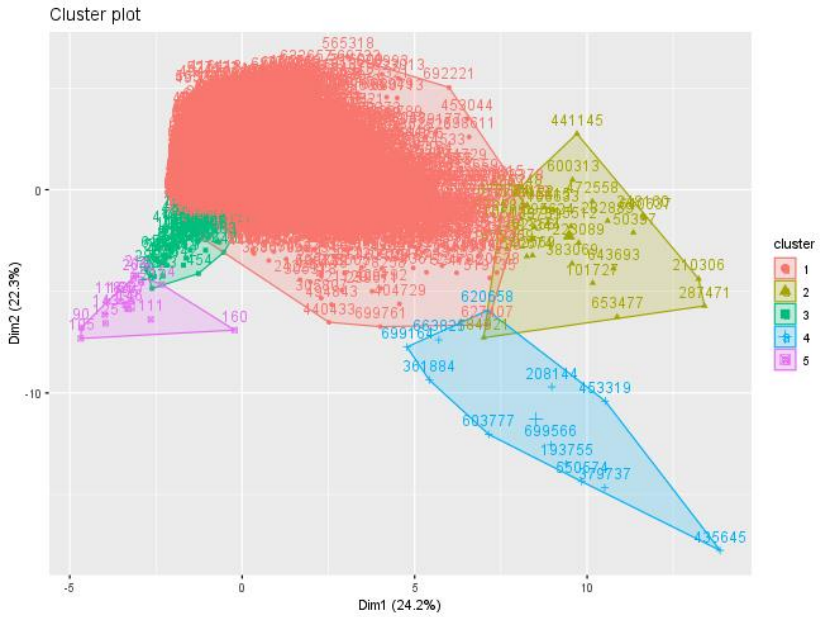


Figura 22. Agrupación clustering jerárquico

La Tabla 6 muestra el resultado de los valores medios de cada agrupación para su evaluación. A pesar de que se aprecia claramente que existe una agrupación que engloba la mayoría de los datos, se identifica en las otras que hay un valor característico por el cual se puede caracterizar la agrupación. La representación de los *clusters* de libros que se obtienen se muestra en la Figura 23.

cluster	rank	price	reviews_n	score_mean	votes_mean	n
3	1,581,225	14.63083	1,485.12037	4.335991	1.089009	108
5	1,781,158	16.54267	3,929.60000	4.395853	2.003611	15
1	2,318,717	15.44587	36.60484	4.374783	2.013746	34,824
2	2,647,750	26.91500	23.20000	3.057347	32.622723	30
4	6,774,734	815.07333	10.08333	4.388387	3.650213	12

Tabla 6. Valor promedio de todos los elementos en cada grupo de libros

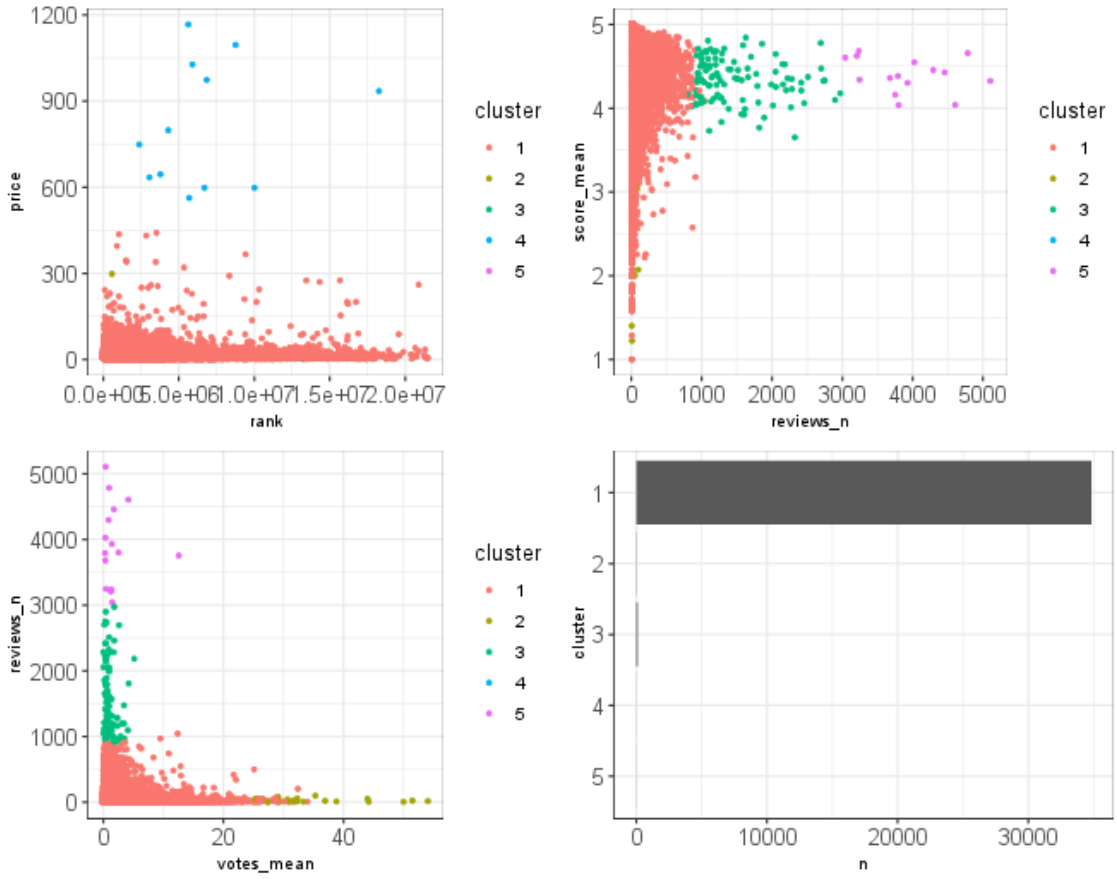


Figura 23. Visualización de grupos de libros

De la cantidad de libros por agrupamiento, destacaría el *cluster* 1, el cual engloba una gran cantidad de libros caracterizados por estar situados en un ranking intermedio, ser de precio medio bajo y tener buenas puntuaciones.

El *cluster* 3 podría decirse que engloba a los libros mejor considerados por estar situados en buenas posiciones del ranking, ser los más baratos y tener bastantes reseñas, aunque poco valoradas.

La agrupación que contiene la mayor cantidad de reseñas útiles y mejores puntaciones es el *cluster* 5, situado también en buena posición del ranking.

Los libros que por puntuación están peor valorados se agrupan en el *cluster* 2, estos productos son algo más caros que la media y sus reseñas tienen una gran cantidad de votos, además tienen una mala posición en el ranking. Podrían tratarse de libros con reseñas negativas.

Por últimos los libros peor posicionados en el ranking se encuentran en el *cluster* 4, además de ser los libros más caros y con menor cantidad de reseñas.

Evaluando las métricas: número de reseñas, puntuación, votos útiles y precio, se muestra en la Figura 24 la función de densidad de la característica más destacada de los cuatro clústeres más determinados por estas métricas comparados con la función de densidad de la misma variable combinada con el resto de *clusters*.

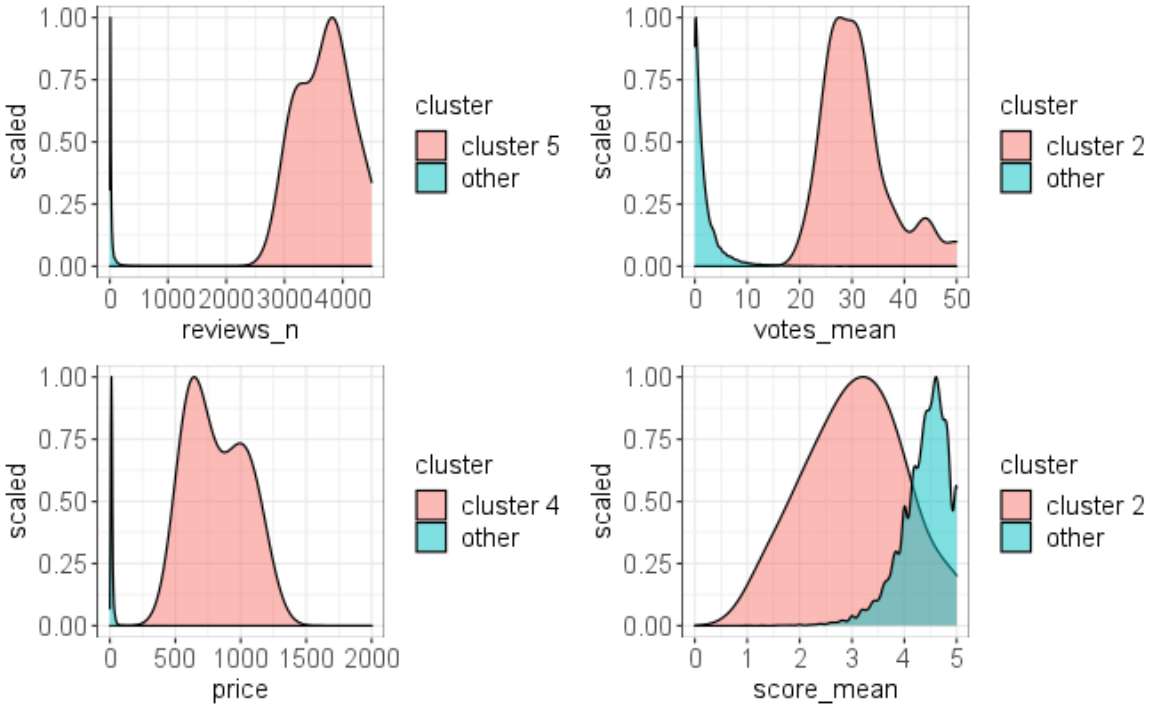


Figura 24. Función de densidad de los clusters según su variable más destacada

4.5. Preprocesamiento del texto

El objetivo de este apartado es extraer información útil e importante del texto, por medio de la identificación de patrones dentro de las agrupaciones de las reseñas. Para ello se combinan diferentes técnicas de minería de textos como el preprocesamiento, limpieza de textos y tokenización.

Inicialmente se seleccionan todas las reseñas pertenecientes a cada uno de los grupos de libros identificados en la fase de *clustering*. El primer paso en cualquier tarea de minería de textos es realizar algunas tareas de limpieza de datos. Los métodos seleccionados dependerán a menudo de la naturaleza de los datos. Dado que las reseñas con las que se trabaja están escritas en inglés se sustituye las contracciones por su forma larga, además se filtraran reseñas que no contengan texto alguno o datos perdidos.

4.5.1. Tokenización

Dentro del marco de texto que se obtiene, se debe dividir el texto en tokens individuales y transformarlo en una estructura de datos ordenada. En consecuencia, se divide cada fila del conjunto datos (reseña) para que cada token o palabra individual componga una fila del nuevo marco de datos, además se eliminan los signos de puntuación y se convierten todas las palabras a letra minúscula para facilitar su comparación.

Un primer análisis ha mostrado que las palabras más comunes utilizadas en las reseñas son las llamadas *stop words*, o palabras vacías que carecen de significado por sí solas. Estas suelen ser artículos, preposiciones, pronombres, conjunciones, etc. También aparecen en los resultados de forma muy frecuente las palabras *book*, *books y read*, siendo los libros y la lectura el tema principal de este trabajo no arrojan información relevante para el análisis. Por esta razón, durante la tarea de tokenización se eliminan todas estas palabras para poder visualizar cuales serían las más comunes usadas en cada uno de los *clusters*.

En la Figura 25 se representa las 10 palabras más comunes diferenciadas por agrupación.

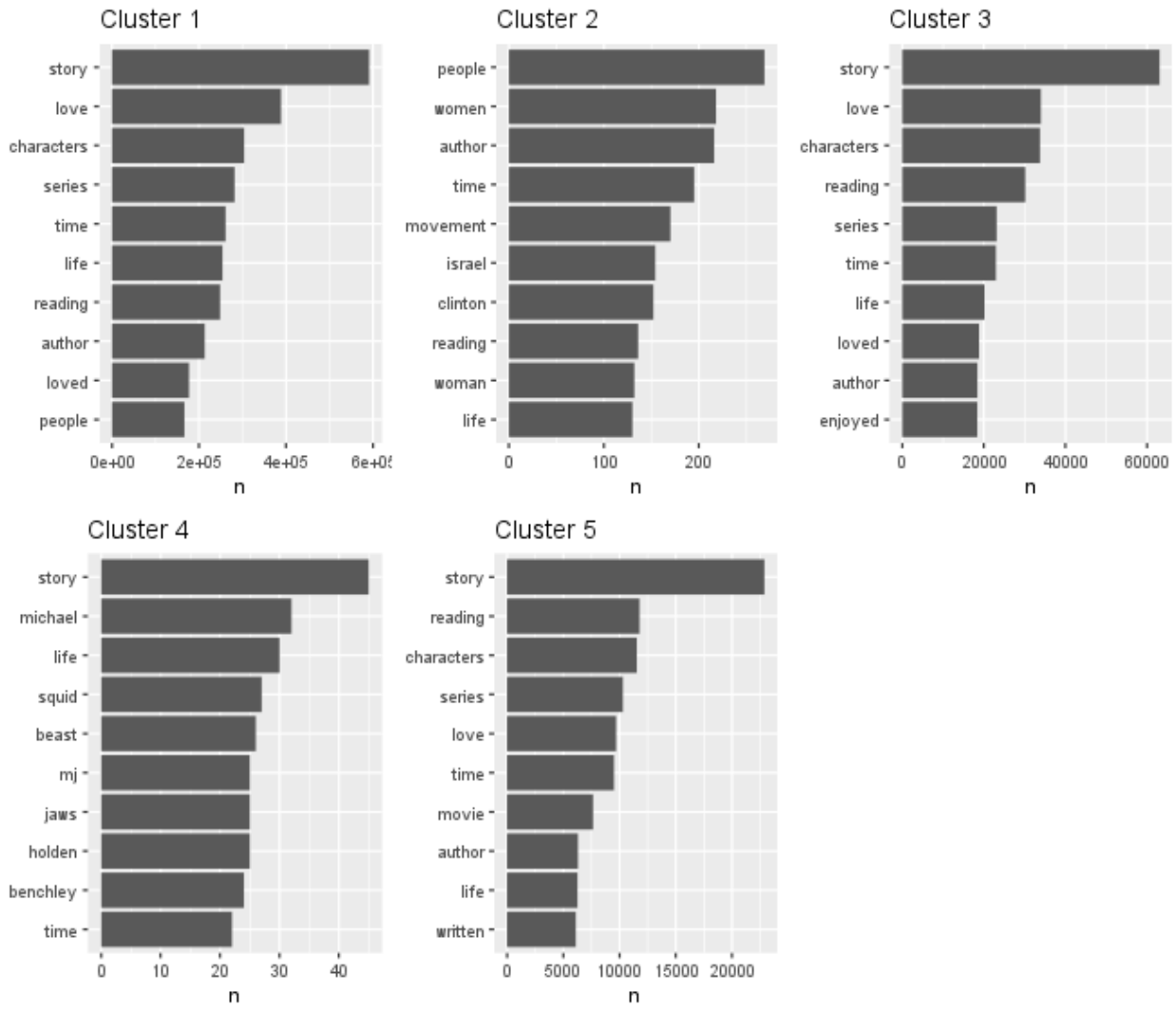


Figura 25. Las 10 palabras más comunes por cluster

A primera vista los *clusters* 1, 3 y 5 contienen palabras muy similares entre ellos, centrándose en los personajes e historia de los libros y mostrando un sentimiento positivo. Sin embargo, los *clusters* 2 y 4 muestran un grupo de palabras de temática diferente. El 2 podría ir enfocado a temas políticos, mientras que el 4 a algún tema o historia que no se logra identificar inicialmente.

4.6. Procesamiento del lenguaje natural

4.6.1. Análisis del sentimiento

Para este análisis se usará el léxico AFINN. Este asigna a las palabras una puntuación que oscila entre valores de -5 y 5, siendo la puntuación negativa la que indica un sentimiento negativo y la positiva la que indica sentimiento positivo. La puntuación total obtenida será añadida al conjunto de datos para cada una de las reseñas.

En la Figura 26 se muestran las palabras positivas y negativas más comunes de cada *cluster*. Se han seleccionado las palabras con la mayor contribución a las puntuaciones de sentimiento. La contribución es el producto de la palabra y la puntuación de sentimiento.

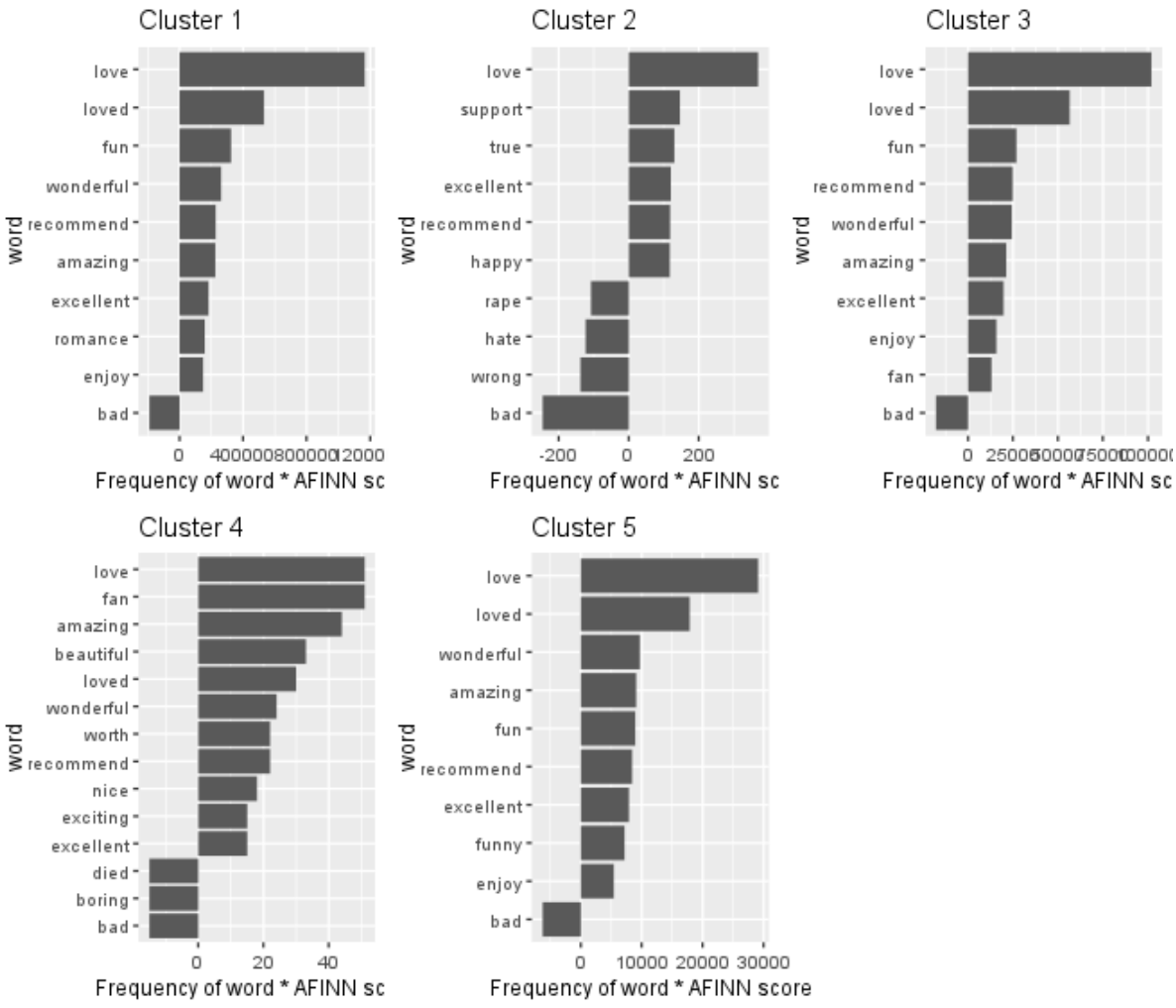


Figura 26. Análisis del sentimiento de cada cluster

Nuevamente los *clusters* 1, 3 y 5 muestran sentimientos similares, donde se valora muy positivamente el producto y se recomienda a otros lectores, entre ellos destaca el 1 del que se identifica una posible temática de romance.

Los clusters 2 y 4 contienen una mayor cantidad de opiniones negativas, en especial el 2 con palabras como el odio o la violación, mientras que el 4 se menciona la muerte y el aburrimiento. En todos los grupos aparece la palabra *bad*, siendo una palabra muy regular con connotación negativa.

4.6.2. Análisis TF-IDF

En este análisis se busca identificar la importancia de una palabra para un documento en una colección de documentos, es decir de reseñas. Con este análisis no se han obtenido buenos resultados, se observó que se le da mucha relevancia a palabras que no pertenecen al inglés, como por ejemplo el alemán e incluso español, u otras palabras que no se identifica su significado, como se muestra en la Figura 27. Además, este valor añadido incrementa la puntuación de algunas reseñas que pueden estar escritas en otros idiomas. Cabe destacar que, al usar este método, las reseñas más extensas siempre obtendrán una mejor puntuación.

Otro inconveniente en el cálculo del término tf-idf es que requiere de mucho tiempo, sobre todo para el grupo que contiene mayor cantidad de reseñas. A pesar de esto, los valores tf-idf serán añadidos al conjunto de datos como un campo extra para posteriores consultas.

Este análisis requiere interpretación adicional, por lo que parece más adecuando ampliarlo a pares de palabras consecutivas.

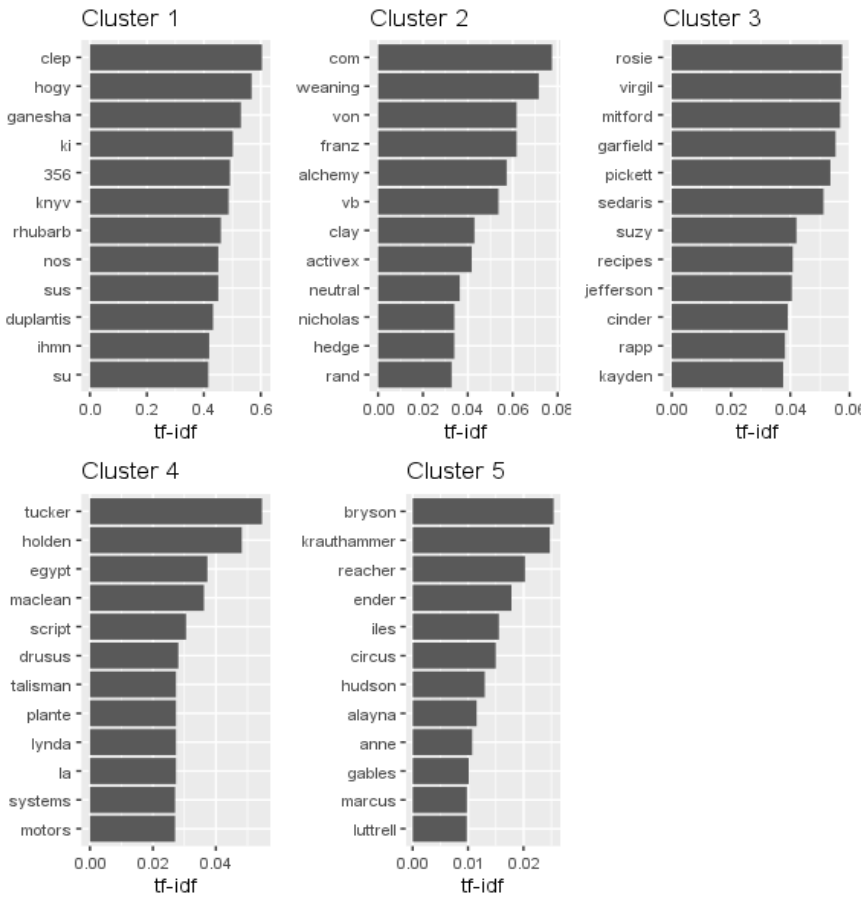


Figura 27. Palabras con mayor tf-idf por cluster

Hasta ahora se han considerado las palabras como unidades individuales y sus relaciones con los sentimientos o los documentos. Sin embargo, muchos análisis interesantes de texto se basan en las relaciones entre las propias palabras, ya sea examinando qué palabras tienden a seguir a otras, o que palabras tienden a coincidir dentro de los mismos documentos.

A continuación, se realiza el mismo análisis usando bigramas o lo que es lo mismo pares de palabras consecutivas. Como es de esperar, los bigramas más comunes y menos interesantes son los formados por combinación “of the” “in the”…etc., es decir los llamados *stopwords*. Para un primer vistazo se representan gráficamente cómo estarían las palabras conectadas entre sí, mostrando aquellas combinaciones de palabras más frecuentes, de lo cual se pueden destacar los siguientes detalles en la estructura de los textos.

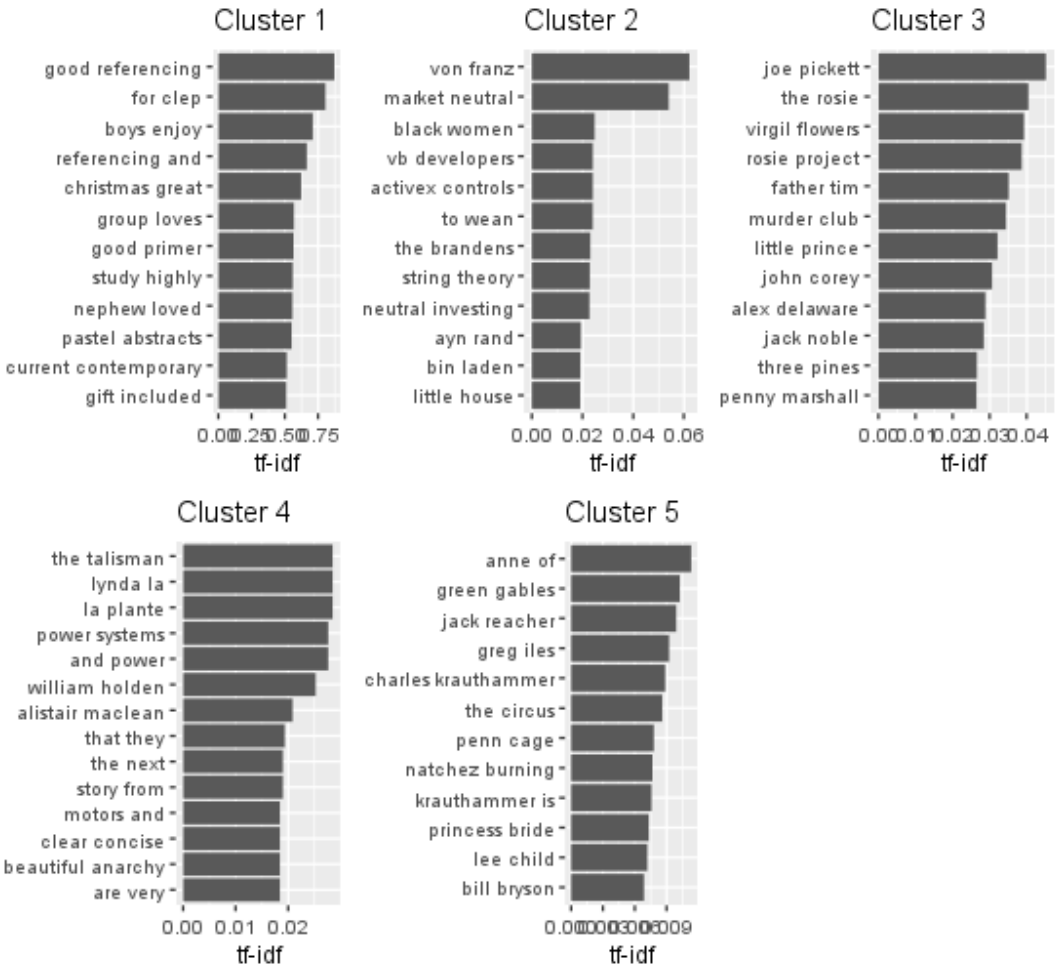


Figura 28. Bigramas con mayor tf-idf por cluster

En la Figura 28 se muestran las palabras más importantes de cada *cluster* medidas por su tf-idf. En el *cluster* 1, siendo el más amplio, destacan principalmente los buenos comentarios hacia los libros como buena referencia, de buen gusto o de haber disfrutado su lectura, incluso podría intuirse de alguna recomendación como regalo. En el resto de *clusters* se detectan individualmente temas diversos como nombres de autores, personajes, libros y alguna temática con la que pueden relacionarse.

4.6.3. Modelado de temas

En primer lugar, se debe convertir el texto en una matriz de términos del documento (DTM) que es una matriz dispersa que contiene sus términos y documentos como dimensiones, el método de separación de palabras es de un token por fila. Tras esto se procede a la creación de un modelo de tres temas para cada uno de los *clusters*, exceptuando el *cluster* número 1 que al ser tan grande se eligió un mayor número de temas. Además, se tuvo que de dividir el conjunto de datos en dos partes debido a problemas de memoria.

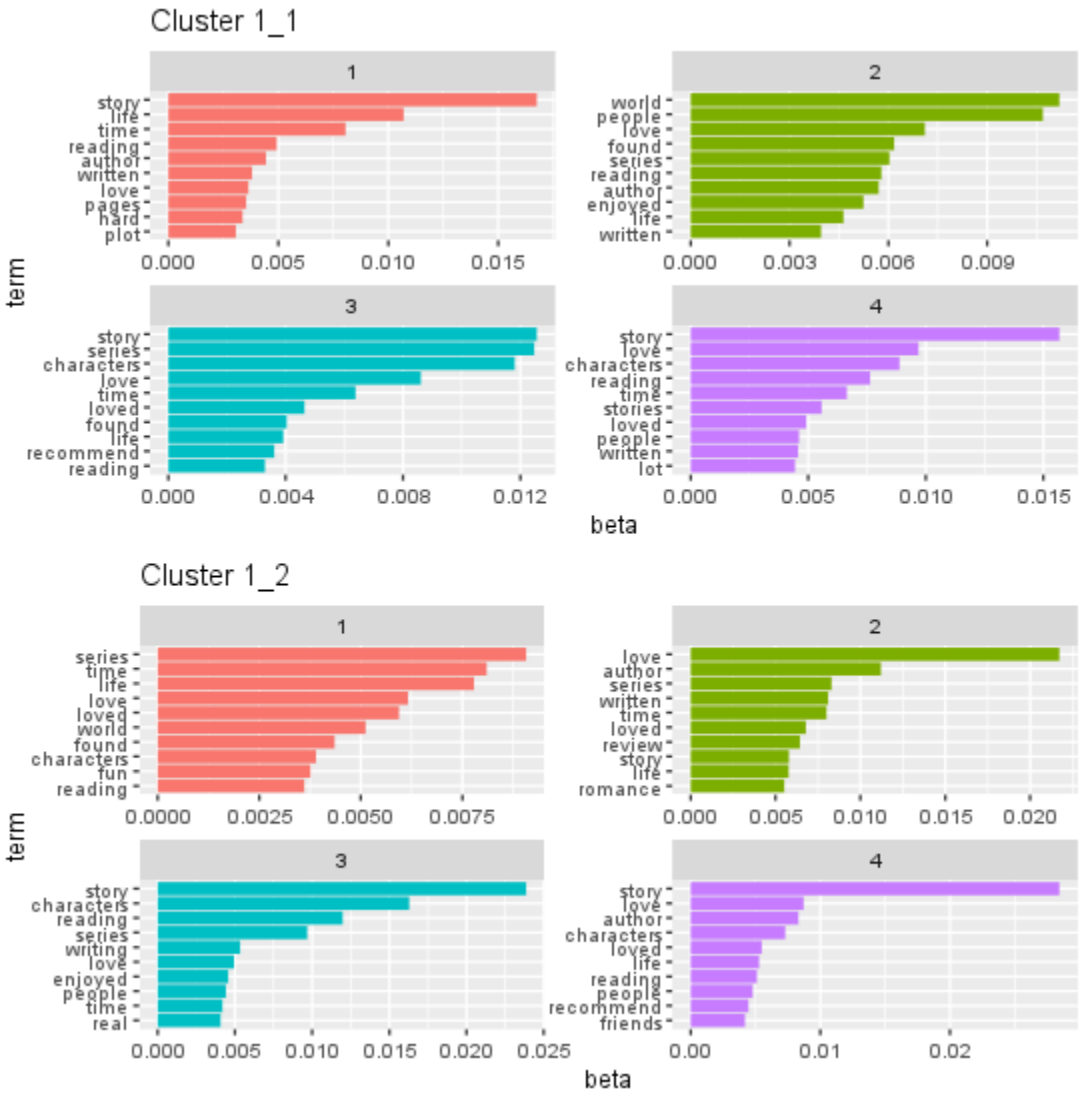


Figura 29. Topic modeling del cluster 1

Los resultados del primer *cluster* se muestran en la Figura 29. Como ya se había visto anteriormente, tanto entre las palabras más frecuentes como por su importancia según el valor tf-idf destacaban palabras positivas, los 8 temas que se aprecian son todos muy similares y generales, destacando la historia, al autor, a los personajes… Aunque parezca evidente aquí el tema principal son la lectura de libros.

En el *cluster* 2, mostrado en la Figura 30, y tal como era de esperar tras los análisis anteriores, el primer tema va enfocado a la política, nombrando a Clinton, Bush, Bin Laden y a la propia palabra política. El segundo tema está posiblemente enfocado al autor Warraq que siendo un autor de libros sobre el islam daría pie al contenido del tercer tema, el cual estaría más orientado a la religión conteniendo palabras como Israel, las mujeres, matrimonio, cristianos y *quiverfull*, que se trata de un movimiento reciente entre parejas cristianas conservadoras.

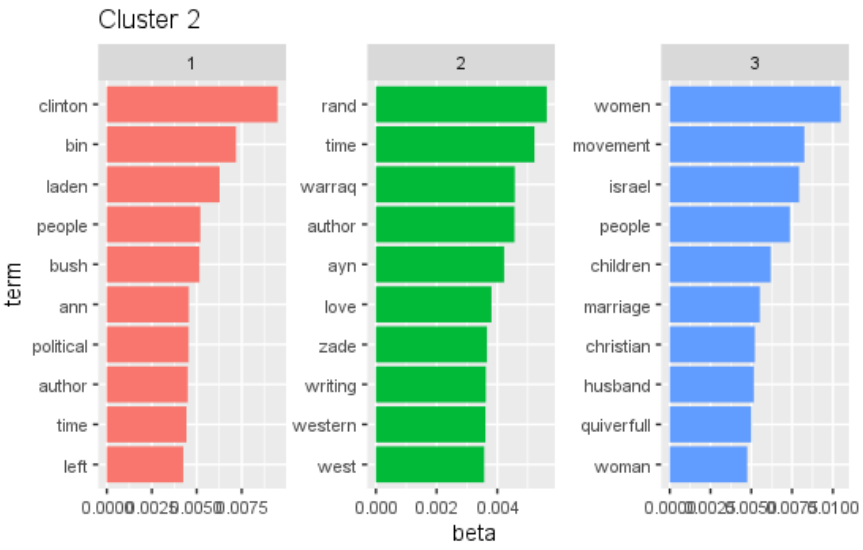


Figura 30. Topic modeling del cluster 2

En el tercer *cluster* mostrado en la Figura 31, el primer tema no aporta nada en concreto solo expresiones positivas, el segundo podría ser sobre la vida y la familia o alguna historia que sucede en un colegio y el tercero sobre otra temática, como la guerra en América y los dioses. Se podría plantear reducir el número de temas de este *cluster* a dos.

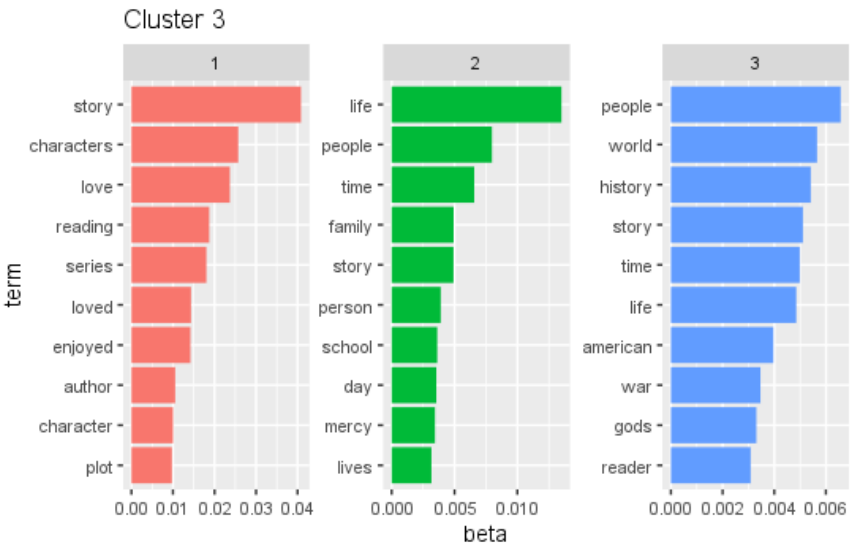


Figura 31. Topic modeling del cluster 3

El primer tema del *cluster* 4 (ver Figura 32) nombra a Willian MacLean pudiendo referirse a un político americano, otras palabras destacadas son holden y tucker, que podrían asociarse al sector automovilístico y alistair que según Wikipedia se trata de un luchador británico. El segundo tema trata claramente sobre el autor Benchley y alguna de sus obras como Jaws o Beast. Por último, el tercer tema es sobre el cantante Michael Jackson. Se podría valorar ampliar la cantidad de temas para este análisis ya que el primer tema es algo confuso.

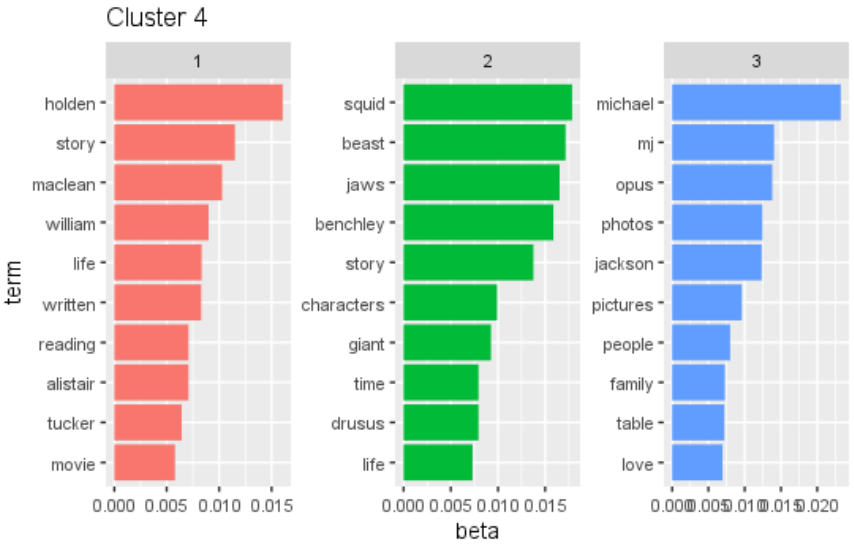


Figura 32. Topic modeling del cluster 4

Por último, el primer tema del *cluster* 5 (Figura 33) trata sobre la novela *Ender's Game* de la cual se ha hecho una película. El segundo tema no aporta ninguna información interesante por lo que se podría obviar. El tercer tema se enfoca probablemente en el escritor estadounidense Bryson.

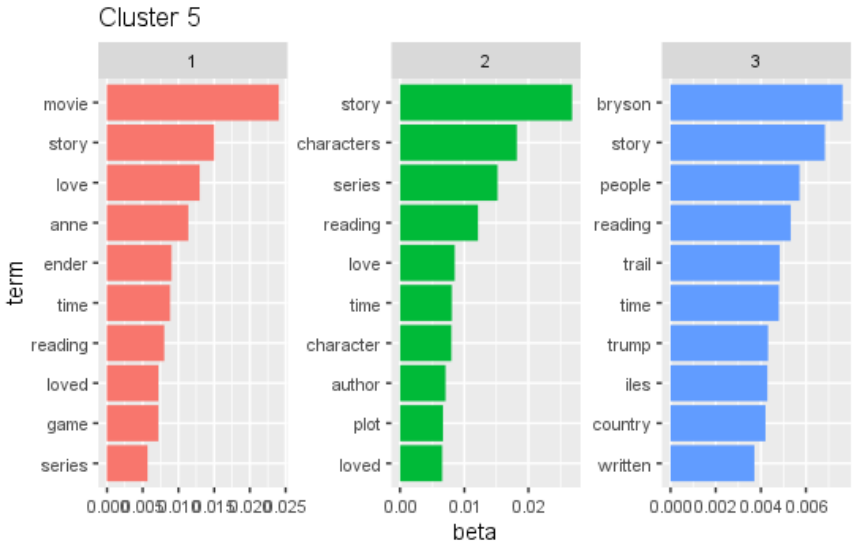


Figura 33. Topic modeling del cluster 5

5. Conclusiones

En el estudio realizado se muestran los pasos que se han seguido para conseguir los objetivos planteados.

Principalmente no sería posible lograr un resultado óptimo si no se realiza previamente una buena limpieza y preprocesado de los datos, por lo que esta etapa previa es de vital importancia.

En cuanto al *clustering,* la calidad de los resultados depende del algoritmo empleado, por eso es necesario conocer y aplicar diferentes tipos de algoritmos puesto que, en función de la forma de los datos, estos pueden ser más o menos apropiados, como es el caso del algoritmo de densidad que para nuestros datos no mostró ninguna eficiencia. Otro dato importante ha sido la elección de los parámetros del algoritmo y la detección de *outliers* ya que en función de ello se obtienen agrupaciones distintas de mayor o menor calidad. A pesar de que los índices de validación interna, estabilidad y de agrupaciones formadas por un algoritmo jerárquico ofrecen datos orientativos a la hora de elegir el número de grupos, la elección final puede variar en función del análisis o interpretación propia.

En cuanto a la minería de textos, el preprocesamiento de los textos también es una etapa importante para poder realizar los análisis deseados. Sin embargo, el procesamiento de lenguajes naturales es un campo amplio y complejo. De los resultados obtenidos se extrae que existen dos grupos (2 y 4) que tanto por análisis de sentimiento, por temas principales como la política, religión y sus elevados precios se sitúan en posiciones peores en el ranking, se intuye que podrían ser libros mal valorados. El grupo 3 y 5, presentan sentimientos positivos y el modelado de temas solo ofrece información general de los temas principales que se tratan en las reseñas, estos *clusters* agrupan a los libros por media mejor situado en el *ranking* y con mayor cantidad de reseñas. Un problema que se ha observado es en el grupo 1 que, al ser de gran tamaño, tras el análisis solo ofrece información general. Este grupo recoge el resto de los libros que han sido agrupados por tener características similares.

Finalmente se concluye que mediante el *clustering* se han logrado clasificar los datos por su característica más saliente, aunque el grupo de gran tamaño podría analizarse por separado para

extraer más información, es decir clasificarlo en subgrupos. Adicionalmente, los libros con mayor número de reseñas, buen precio y sentimiento positivo están mejor situados en el ranking, por el contrario, libros con precio altos, sentimientos más negativos y reseñas con muchos votos útiles se sitúan peor en el ranking.

Por lo que podría decirse que las reseñas influyen en las ventas, pero no sería el único factor influyente quedando la interpretación subjetiva por parte de quien analiza.

6. Trabajo futuro

Una posibilidad de trabajo futuro sería el realizar el estudio con unas características diferentes o con un subconjunto de las utilizadas. Además, se podría plantear y extender el trabajo realizando en este estudio para el caso de multi-productos o analizar multi-plaltaformas. Por ejemplo, se podría analizar el efecto de comentarios en páginas dedicadas al sector turístico: viajes, reservas de hotel, etc.

Por otro lado, el trabajo desarrollado se podría continuar o ampliar incorporando mejoras en el preprocesado lingüístico como añadir el *stemming* al procedimiento, que consiste en reducir las palabras a su raíz o buscar como determinar el número óptimo de *topics* para cada *cluster*.

Además, se podrían usar todos los datos extraídos para entrenar un modelo de aprendizaje supervisado.

Bibliografía

[1] Seeberger, B., Schwarting, U., & Meiners, N., «The Renaissance of Word-of-Mouth Marketing: A 'New' Standard in Twenty-First Century Marketing Management?,» *International Journal of Economic Sciences and Applied Research,* vol. 3(2), pp. 79-97, 2010.

[2] Chevalier, Judith A, and Dina Mayzlin, «The Effect of Word of Mouth on Sales: Online Book Reviews,» *Journal of Marketing Research 43.3,* pp. 345-54, 2006.

[3] McAuley, Julian, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel, «Image-Based Recommendations on Styles and Substitutes,» *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 43-52, 2015.

[4] He, Ruining, and Julian McAuley, «Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-class Collaborative Filtering,» *25th International World Wide Web Conference, WWW ,* pp. 507-517, 2016.

[5] J. Ni, «Amazon Review Data (2018),» [En línea]. Available: https://nijianmo.github.io/amazon/index.html.

[6] Fry, C., & Manna, S., «Can We Group Similar Amazon Reviews: A Case Study with Different Clustering Algorithms,» *IEEE Tenth International Conference on Semantic Computing (ICSC),* pp. 374-377, 2016.

[7] S. Bañales, «How good is an Amazon book review? Modeling the interaction between sales, prices, reviews and user's helpfulness votes,» TFM Master Big Data y Business Analytics, UNED, 2019.

[8] D. Chiu, «R for Data Science Cookbook : Over 100 Hands-on Recipes to Effectively Solve Real-world Data Problems Using the Most Popular R Packages and Techniques,» 1st ed. 2016.

[9] Kumar, A., & Paul, A., «Mastering text mining with R : Master text-taming techniques and build effective text-processing applications with R,» 2016 (1st ed.).

[10] B. Lantz, «Machine Learning with R,» 1st ed. 2013.

[11] J. Macqueen, «Some Methods for Classification and Analysis of MultiVariate Observations,» *In 5-th Berkeley Symposium on Mathematical Statistics and Probability,* pp. 281-297, 1967.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,» 1996.

[13] Sander, Jörg, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu, «Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications,» *Data Mining and Knowledge Discovery 2.2,* pp. 169-194, 1998.

[14] Patel Sakshi, Shivani Sihmar, and Aman Jatain, «A Study of Hierarchical Clustering Algorithms,» *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom),* pp. 537-541, 2015.

[15] Halkidi, M., Batistakis, Y., Vazirgiannis, M., «On Clustering Validation Techniques,» *Journal of Intelligent Information Systems, 17(2),* pp. 107-145, 2001.

[16] J. C. Dunn, «Well separated clusters and fuzzy partitions,» *Journal on Cybernetics,* pp. 4:95-104, 1974.

[17] P. J. Rousseeuw, «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,» *Journal of Computational and Applied Mathematics,* pp. 20:53-65, 1987.

[18] J. Handl, J. Knowles, and D. B. Kell, «Computational cluster validation in postgenomic data analysis,» *Bioinformatics,* pp. 21(15):3201-12, 2005.

[19] Brock, G., Pihur, V., Datta, S., & Datta, S., «clValid: An R Package for Cluster Validation,» *Journal of Statistical Software,* pp. 25(4), 1–22, 2008.

[20] J. &. R. D. Silge, «Text mining with R : A tidy approach,» 2017.

[21] Blei, D., Ng, A., & Jordan, M., «Latent Dirichlet allocation,» *Journal of Machine Learning Research, 3(4-5),* pp. 993-1022, 2003.

[22] R Core Team, «R: A Language and Environment for Statistical Computing,» 2021. [En línea]. Available: https://www.r-project.org/.

[23] Jianmo Ni, Jiacheng Li, Julian McAuley, «Justifying recommendations using distantly-labeled reviews and fined-grained aspects,» *Empirical Methods in Natural Language Processing (EMNLP),* 2019.

Glosario de términos

AD	Average Distance
ADM	Average Distance between Means
APN	Average Proportion of Non-overlap
CPCC	Coeficiente de Correlación Cofenética
CSV	Comma-separated Values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DTM	Document Term Matrix
Eps	Epsilon
ETL	Extract, Transform, Load
FOM	Figure of Merit
IDF	Inverse Document Frequency
JSON	JavaScript Object Notation
LDA	Latent Dirichlet Allocation
MinPts	Minimum Points
NA	Not Available
NLP	Natural Language Processing
SSE	Sum of Squared Errors
Tab	Tab separated data file
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
URL	Uniform Resource Locators
WCSS	Within Cluster Sum of Squares
WOM	Word of Mouth
WOMM	Word of Mouth Marketing
XLS	Microsoft Excel Spreadsheet
XLSX	Microsoft Excel Open XML Spreadsheet