Facultad de Ciencias Sociales y Jurídicas de Elche
Estadística Empresarial
2022/2023

Programación lineal entera mixta y métodos heurísticos, para la selección de características en clustering

Francisco Molina Ferrández

Tutor:
Mercedes Landete Ruiz

Índice

1. Introducción a los problemas de clasificación (Análisis Cluster)

¿Sería posible determinar en que empresas es más rentable invertir?

¿Es posible agrupar a los clientes de una empresa, según sus gustos, para ofrecerles productos afines a sus preferencias?

¿Se pueden clasificar las vacas de un rebaño según sus facultades de producción y aptitudes para la explotación bovina (obtención de leche y carne, edad, enfermedades...) o las explotaciones ganaderas según su participación en funciones no productivas

¿Se pueden catalogar los diferentes vinos producidos en España atendiendo a sus características químicas y ópticas?

¿Es posible clasificar las estrellas del universo en función de su radiación?

Se trata fundamentalmente de resolver el siguiente problema: Dado un conjunto de objetos (U) diferenciados por la información de (V) características se plantea el reto de clasificarlos de manera que los objetos pertenecientes a un grupo (al que llamaremos cluster, y siempre con respecto a la información disponible) sean tan similares entre sí como sea posible, siendo los distintos grupos entre ellos tan desiguales como sea posible.

Dentro de los métodos de Análisis Multivariante, el Análisis Cluster, también conocido como Análisis de Conglomerados, es uno de los más recientes y tiene como objetivo la clasificación de individuos en grupos distintos, de forma que los perfiles de los objetos en un mismo grupo sean muy similares entre sí (homogeneidad, cohesión interna del grupo) y los de los objetos de grupos diferentes sean distintos (separación, aislamiento externo del grupo). Por comodidad, utilizaremos el término "cluster" para referirnos a estos grupos o conglomerados y "clustering" para referirnos a las respectivas agrupaciones.

Pertenece al igual que otras tipologías y que el Análisis Discriminante al conjunto de técnicas que tiene por objetivo la clasificación de los individuos. La diferencia fundamental entre el Análisis Cluster y el Discriminante reside en que en el Análisis Cluster los grupos son desconocidos a priori y son precisamente los que queremos determinar; mientras que en el Análisis Discriminante, los grupos son conocidos y se pretende describir (si existen) las diferencias significativas entre ellos, de forma que nos pueden ayudar a clasificar o asignar los individuos en los grupos dados.

Tiene una extraordinaria importancia en la investigación científica, en cualquier rama del saber, siendo la clasificación uno de los objetivos fundamentales de la ciencia. Sin embargo, junto con los beneficios del Análisis Cluster existen algunos inconvenientes. El Análisis Cluster es una técnica descriptiva y no inferencial. No tiene bases estadísticas sobre las que deducir inferencias estadísticas para una población a partir de una muestra, es un método basado en criterios geométricos y se utiliza fundamentalmente como una técnica exploratoria, descriptiva pero no explicativa. Las soluciones no son únicas, en la medida en que la pertenencia al cluster para cualquier número de soluciones depende de varios elementos del procedimiento elegido. Por otra parte, la solución depende totalmente de las variables utilizadas. La adición o destrucción de variables relevantes puede tener un impacto sustancial sobre la solución resultante.

Debido a esto último, es condición primordial en este tipo de estudio realizar una buena elección de las variables iniciales, así como también elegir una medida de homogeneidad o similaridad adecuada para la situación que se esté analizando. No existe una única medida de homogeneidad, ni tampoco es único el método de agrupar observaciones en distintos clusters. Es tarea del analista decidir qué medida y qué método son más adecuados según los datos de partida y los objetivos a conseguir con la agrupación.

Así pues, el objetivo es obtener clasificaciones (a las que también nos referiremos como clusterings), teniendo dicho análisis un marcado carácter exploratorio.

Para conseguir este objetivo, una vez establecidas las variables y los objetos a clasificar, el siguiente paso consiste en establecer una medida de proximidad o de distancia entre ellos que cuantifique el grado de similitud entre cada par de objetos. Este paso, junto con la elección de distancia a definir entre dos clusters o entre un objeto y un cluster, serán los que más impacto tendrán sobre la solución final.

El proceso completo puede estructurarse de acuerdo con el siguiente esquema:

- Partimos de un conjunto de n objetos de los que se dispone de una información cifrada por un conjunto de m características (una matriz de datos n × m).
- Establecemos la forma de medir la distancia entre objetos, para comprobar la semejanza entre sí.
- Escogemos un algoritmo de clasificación para determinar la estructura de agrupación de los individuos.
- Especificamos esa estructura mediante diagramas arbóreos, dendogramas u otros gráficos.

Los algoritmos para la agrupación se basan en estadística, matemática y ciencias de la computación. Se deben tener los siguientes criterios en cuenta antes de plantear el análisisc cluster:

1. Objetivo del agrupamiento (criterios).
2. La justificación de perseguir ese objetivo (axiomática).
3. Las restricciones a considerar (elección del tipo de agrupamiento).
4. Dificultad en llevar a cabo la agrupación (la cuestión de la complejidad).
5. Cómo debe hacerse la agrupación (diseño de algoritmos).
6. Significación del cluster obtenido (interpretación).

2. Ejemplos reales de clustering

2.1. Spotify

Spotify utiliza el análisis de clusters como parte de su sistema de recomendación de música. A través del análisis de clusters, Spotify agrupa a sus usuarios y canciones en segmentos con características similares, lo que les permite ofrecer recomendaciones personalizadas a cada usuario.

El análisis de clusters en Spotify implica la aplicación de algoritmos de agrupamiento para identificar patrones y similitudes entre los usuarios y las canciones. A partir de los datos recopilados de los usuarios, como las canciones que han escuchado, las listas de reproducción que han creado y los artistas que siguen, Spotify puede construir perfiles de usuario.

Estos perfiles de usuario se utilizan para agrupar a los usuarios en segmentos o clusters, donde cada cluster representa un grupo de usuarios con preferencias musicales similares. El análisis de clusters también se aplica a las canciones para agruparlas en función de características como el género, el tempo, la energía y otros atributos musicales.

Una vez que se han creado los clusters de usuarios y canciones, Spotify utiliza técnicas de filtrado colaborativo y análisis de contenido para generar recomendaciones personalizadas. Por ejemplo, si un usuario pertenece a un determinado cluster y ha escuchado canciones similares a las de otros usuarios del mismo cluster, Spotify puede recomendarle canciones que hayan sido populares entre esos usuarios.

Además, Spotify también utiliza información de clusters para crear listas de reproducción personalizadas, como las populares 'Descubrimiento semanal' y 'Daily Mix'. Estas listas de reproducción se generan en función de las preferencias y similitudes musicales de cada usuario, utilizando tanto el análisis de clusters como otros algoritmos de recomendación.

En conclusión, Spotify utiliza el análisis de clusters para agrupar a sus usuarios y canciones en segmentos con características similares, lo que les permite ofrecer recomendaciones personalizadas y crear experiencias musicales adaptadas a cada usuario.

2.2. TOYOTA

Toyota utiliza modelos lineales en su producción y planificación de la cadena de suministro para optimizar la eficiencia y reducir los costes.

Un ejemplo concreto del uso de modelos lineales por parte de Toyota es en la planificación de la demanda y la programación de la producción. Utilizan modelos lineales para predecir la demanda de diferentes modelos de vehículos en función de factores como las tendencias del mercado, las preferencias del cliente y las condiciones económicas. Estos modelos les ayudan a determinar la cantidad óptima de vehículos a producir y distribuir en diferentes regiones, lo que les permite evitar el exceso de inventario o la escasez de productos.

Toyota también utiliza modelos lineales en la optimización de la cadena de suministro. Estos modelos les permiten planificar el flujo de materiales y componentes desde los proveedores hasta las plantas de fabricación, minimizando los costes de transporte y almacenamiento. Los modelos lineales también ayudan a tomar decisiones sobre la localización de almacenes y la asignación de recursos de manera eficiente.

Mediante el uso de modelos lineales, Toyota puede obtener soluciones que les permiten maximizar la eficiencia de la producción, optimizar la cadena de suministro y tomar decisiones basadas en datos para mejorar su rendimiento operativo.

Es importante tener en cuenta que aunque se sabe que Toyota utiliza modelos lineales en sus operaciones, los detalles específicos y las soluciones exactas que obtienen de estos modelos no están disponibles públicamente debido a la naturaleza competitiva de la industria.

2.3. FedEx

FedEx ha aplicado el análisis de clusters para mejorar la eficiencia en la gestión de su cadena de suministro y en la distribución de paquetes.

A través del análisis de clusters, FedEx puede agrupar destinos y rutas de entrega en función de características similares, como la densidad de población, la demanda de envíos y las condiciones geográficas. Esto les permite optimizar la asignación de recursos, como vehículos de transporte y personal de entrega, al asignarlos a grupos de destinos con características similares. Además, el análisis de clusters también les ayuda a identificar patrones de demanda y a ajustar sus rutas y servicios en consecuencia.

Algunas de las soluciones y beneficios que FedEx ha obtenido del uso del análisis de clusters incluyen:

Optimización de rutas: El análisis de clusters les permite identificar patrones en la demanda de envíos y agrupar destinos que tienen características similares. Esto les ayuda a optimizar las rutas de entrega al asignar de manera eficiente los recursos disponibles y reducir los tiempos de viaje y los costos operativos.

Personalización de servicios: Mediante la segmentación de destinos y clientes en grupos basados en características comunes, FedEx puede adaptar sus servicios de entrega para satisfacer las necesidades específicas de cada grupo. Esto incluye la personalización de los horarios de entrega, los servicios adicionales y las soluciones logísticas.

Mejora en la gestión de inventario: Al comprender los patrones de demanda y agrupar destinos en función de características similares, FedEx puede optimizar la gestión de inventario y la distribución de productos en diferentes ubicaciones. Esto les permite reducir los costes de almacenamiento y mejorar la eficiencia en la entrega.

Es importante tener en cuenta que la información detallada y las soluciones específicas que FedEx ha obtenido del análisis de clusters no están disponibles públicamente debido a la naturaleza competitiva de la industria. Sin embargo, se sabe que FedEx utiliza el análisis de clusters como una herramienta clave en su gestión logística para mejorar la eficiencia y la calidad de sus servicios de entrega.

2.4. Verizon

Una empresa de telecomunicaciones conocida que utiliza el análisis de clusters en sus operaciones es Verizon Communications Inc. Verizon ha aplicado el análisis de clusters para segmentar a sus clientes en grupos con características similares, lo que les permite personalizar sus ofertas y servicios.

A través del análisis de clusters, Verizon puede agrupar a sus clientes en segmentos basados en factores como el comportamiento de uso, las preferencias de servicios y las necesidades de comunicación.

Al realizar esta segmentación, Verizon puede adaptar sus ofertas, planes y promociones a cada grupo de manera más efectiva, brindando servicios más relevantes y satisfactorios a sus clientes.

Algunas de las soluciones y beneficios que Verizon ha obtenido del uso del análisis de clusters incluyen:

Personalización de servicios: Al segmentar a sus clientes en grupos con características similares, Verizon puede personalizar sus servicios y ofertas según las necesidades y preferencias específicas de cada grupo. Esto incluye la adaptación de planes de datos, paquetes de servicios y promociones especiales para maximizar la satisfacción del cliente.

Mejora de la retención de clientes: Mediante el análisis de clusters, Verizon puede identificar grupos de clientes que tienen un mayor riesgo de abandono o churn. Al comprender mejor las características y los comportamientos de estos grupos, Verizon puede implementar estrategias de retención específicas para reducir el churn y fomentar la lealtad del cliente.

Desarrollo de estrategias de marketing más efectivas: El análisis de clusters permite a Verizon comprender mejor las necesidades y preferencias de diferentes segmentos de clientes. Esto les ayuda a desarrollar estrategias de marketing más efectivas al dirigir mensajes específicos a cada grupo, maximizando el impacto y la relevancia de sus campañas.

Optimización de recursos y capacidades: Al agrupar a los clientes en segmentos con características similares, Verizon puede optimizar la asignación de recursos y capacidades para satisfacer las demandas de cada grupo. Esto incluye la gestión eficiente de la infraestructura de red, la capacidad de ancho de banda y los servicios adicionales.

Es importante tener en cuenta que la información detallada y las soluciones específicas que Verizon ha obtenido del análisis de clusters no están disponibles públicamente debido a la naturaleza competitiva de la industria. Sin embargo, se sabe que Verizon utiliza el análisis de clusters como una herramienta clave en su gestión de clientes y estrategias de marketing para mejorar la personalización y la calidad de sus servicios de telecomunicaciones.

2.5. Migros

Migros es una empresa minorista suiza que ofrece una tarjeta de fidelización llamada "Migros Cumulus". A través de esta tarjeta, Migros recopila datos sobre las compras de los clientes y utiliza técnicas de análisis de clusters para segmentar a sus clientes y ofrecerles ofertas y beneficios personalizados.

La información específica sobre cómo Migros utiliza el análisis de clusters en relación con su tarjeta Money Club no está disponible públicamente debido a la naturaleza comercial de dicha información y a las políticas de privacidad de la empresa. Sin embargo, en general, las empresas minoristas utilizan el análisis de clusters en programas de fidelización de clientes para lograr los siguientes objetivos:

1. Segmentación de clientes: El análisis de clusters permite a las empresas minoristas agrupar a sus clientes en segmentos con características similares. Esto ayuda a comprender mejor los patrones de comportamiento de compra, las preferencias de productos y las necesidades individuales de los clientes. A través de esta segmentación, las empresas pueden adaptar sus estrategias de marketing y ofrecer promociones y ofertas específicas a cada segmento.

2. Personalización de ofertas: Utilizando el análisis de clusters, las empresas minoristas pueden personalizar las ofertas y los beneficios para cada segmento de clientes. Esto implica ofrecer descuentos, cupones u otros incentivos que sean relevantes y atractivos para cada grupo de clientes, lo que puede mejorar la satisfacción y la lealtad de los clientes.

3. Mejora de la retención de clientes: El análisis de clusters puede ayudar a identificar a los clientes que corren un mayor riesgo de abandonar la empresa o cambiar a la competencia. Al comprender los patrones de comportamiento y las características de estos clientes, las empresas pueden implementar estrategias de retención específicas, como programas de recompensas, comunicaciones personalizadas y ofertas exclusivas, para mantener a estos clientes satisfechos y comprometidos.

Los ejecutivos de Migros afirman que despues de que se creases los distintos segmentos, empezaron a llevar a cabo las diferentes estrategias de marketing. Se enfocaron principalmente en segmentos que incluían a personas que consumen alimentos saludables, gourmets, comen comida basura y familias con niños. Migros contacta con esos segmentos directamente y ofrece precios especiales para los productos que compran con frecuencia. Además, informan a sus clientes sobre los productos que podrían ser de su interés.

2.6. Identificador de fake news

Las fake news, no son algo nuevo, pero cada vez se dan con más frecuencia debido principalmente a las redes sociales. Un ejemplo de noticias falsas conocido por todo el mundo, es el que se dio durante la campaña por la presidencia de Estados Unidos en 2016.

Esto llamó la atención de dos estudiantes de la universidad de California, quienes mediante un análisis cluster del contenido de las noticias, consiguieron identificar las noticias falsas.

El algoritmo funciona analizando el cuerpo de las noticias, para agrupar noticias en diferentes grupos o clusters según sus características. Esto puede incluir el análisis de similitudes en términos de contenido, estilo de redacción, patrones de difusión o comportamientos en redes sociales, entre otros factores relevantes.

Algunas aplicaciones del análisis de clusters en la detección de fake news son:

1. Análisis de contenido: El análisis de clusters se puede aplicar al contenido de las noticias para identificar patrones comunes entre noticias falsas. Esto implica analizar características como el uso de información engañosa, titulares sensacionalistas, ausencia de fuentes confiables o falta de verificación de datos. Al agrupar noticias similares en clusters, es posible identificar ciertos temas o estilos de redacción que son característicos de las noticias falsas. Por ejemplo, si muchas noticias falsas comparten palabras clave o frases específicas, se pueden utilizar algoritmos de agrupamiento para identificar estos patrones y clasificar las noticias en función de su similitud. Esto facilita la detección de contenido sospechoso o potencialmente falso.

2. Análisis de difusión: El análisis de clusters también se puede utilizar para analizar la difusión de noticias falsas en redes sociales u otras plataformas. Mediante el seguimiento de las interacciones en línea, como compartidos, retuits o comentarios, se pueden identificar patrones de difusión. Al agrupar noticias falsas que han sido ampliamente compartidas por cuentas similares o que se han difundido en comunidades específicas, es posible identificar redes de difusión de noticias falsas. Esto proporciona información valiosa para identificar cuentas sospechosas o fuentes de noticias no confiables.

3. Identificación de fuentes sospechosas: El análisis de clusters puede ayudar a identificar fuentes de noticias falsas al encontrar patrones en los sitios web o fuentes que las difunden. Esto implica examinar características comunes en las páginas de inicio, el diseño, los nombres de dominio o los perfiles de los sitios web asociados con la difusión de fake news. Al agrupar sitios web sospechosos en clusters, es posible identificar patrones o características similares que son indicativos de noticias falsas. Esto puede incluir sitios web con diseños poco profesionales, fuentes de noticias desconocidas o contenidos sensacionalistas.

Es importante tener en cuenta que la detección de fake news es un campo de investigación en constante desarrollo y que ninguna técnica es infalible. El análisis de clusters es solo una de las muchas herramientas que se pueden utilizar para este propósito. Combinar diferentes enfoques y técnicas, como el análisis de contenido, el procesamiento de lenguaje natural y la verificación de hechos, puede mejorar la precisión y confiabilidad en la identificación de noticias falsas.

2.7. Mejorar atención hospitalaria

El análisis de clusters puede aplicarse en el contexto de la atención hospitalaria para mejorar diversos aspectos del servicio.

- Segmentación de pacientes: El análisis de clusters puede ayudar a identificar grupos de pacientes con características similares, como diagnósticos, historias clínicas, patrones de comportamiento o necesidades de atención específicas. Esto permite personalizar el enfoque de atención para cada segmento de pacientes, lo que puede mejorar la calidad y la eficiencia de la atención médica.

- Asignación de recursos: Mediante el análisis de clusters, se pueden identificar patrones de utilización de recursos hospitalarios, como camas, personal médico y equipos. Esto ayuda a optimizar la asignación de recursos al identificar las áreas que requieren una mayor atención o los momentos de mayor demanda. Al comprender las necesidades y los patrones de utilización de los pacientes en cada grupo, los hospitales pueden mejorar la planificación y la gestión de recursos.

- Detección de riesgos y complicaciones: El análisis de clusters puede ayudar a identificar patrones de riesgo y complicaciones en los pacientes. Al agrupar a los pacientes según factores de riesgo comunes o características clínicas similares, se pueden identificar grupos de pacientes con mayor probabilidad de desarrollar ciertas complicaciones. Esto permite implementar medidas preventivas y de gestión de riesgos específicas para cada grupo, lo que puede mejorar la seguridad del paciente y los resultados de atención.

- Mejora de la comunicación y coordinación: El análisis de clusters puede ayudar a identificar patrones de comunicación y coordinación efectiva entre los diferentes profesionales de la salud y departamentos hospitalarios. Al identificar grupos de pacientes que requieren una colaboración estrecha entre equipos médicos, se puede facilitar la comunicación y la transferencia de información relevante. Esto promueve una atención más integrada y mejora la experiencia del paciente.

- Evaluación de la calidad de la atención: El análisis de clusters puede utilizarse para evaluar la calidad de la atención hospitalaria al identificar patrones de resultados clínicos, satisfacción del paciente o eficiencia en la entrega de servicios. Al agrupar a los pacientes según estos indicadores, los hospitales pueden identificar áreas de mejora y tomar medidas para optimizar la calidad de la atención en cada grupo.

Hemos encontrado el ejemplo de una investigación que pretendía apoyar la gestión estratégica de las cirujias en un hospital público, a partir de la agrupación de pacientes a los que se les diagnosticaron tumores malignos.

Para resolver esta problema, se determinó un conjunto de datos con características sociodemográficas y clínicas y se utilizo el algoritmo de las k-medias y la distancia euclídea.

Los clústeres obtenidos permitieron demostrar la presencia de diagnósticos asociados al cáncer, agrupando la población por edad, género, estado de salud, zona de residencia, etnia, grupo sanguineo y tipo de atención.

Las conclusiones que se extrajeron de este estudio, teniendo en cuenta que no se trata de una solución universal y que para extraer conclusiones más precisas sería necesario incluir un mauor conjunto de datos en el estudio son: Los principales tipos de cáncer en varones eran tumores del estómago, colon y la laringe. Entre las mujeres, las principales fueron los tumores de colon, estómago y cerebro. Aunque el grupo más afectado fue el de más de 60 años también se observó la presencia significativa de tumores a partir de 40 años.

3. K-medias

3.1. Introducción

La agrupación k-medias [1] (Forgy,1965 ; MacQueen, 1967) es un tipo de apendizaje no supervisado que se usa cuando se tienen datos sin clasificar. El Objetivo de este algoritmo en encontrar grupos en los datos, los grupos en los datos se representan con la variable k. Este algoritmo trabaja de forma iterativa asignando cada dato a uno de los k grupos o clusters, basandose en sus características. k-medias es uno de los algoritmos más simples y más utilizados que resuelven el problema de clustering.

3.2. Concepto

La idea principal comienza por definir k centroides, uno para cada cluster. Estos centroides se seleccionan inicialmente de forma aleatoria, aunque hay que tener cuidado ya que distintas posiciones pueden llevar a un resultado final diferente. El siguiente paso es seleccionar cada uno de los daos de entrada y asignarlos al centroide mas cercano. Cuando ya no queden datos sin asociar se habrá completado el agrupamiento inicial. En este punto hay qye recalcular k nuevos centroides, según la media de cada grupo resultante del paso anterior. Después debe comenzar una nueva etapa de asignación de los datos más cercanos a estos nuevos centroides. Como resultado de este proceso iteratico los centroides se van desplanzado poco a poco, hasta que llega un momento en que dejan de moverse y alcanzan su posición final.

3.3. Algoritmo de K-medias

Partiendo del conjunto de entrenamiento $x^{(1)}, ..., x^{(m)}$ queremos agrupar los datos en clusters. Se tienen los vectores de variables para cada dato $x^{(i)} \in R$ pero no las caegorías $y^{(i)}$ como es propio de los problemas de aprendizaje no supervisado. Nuestro objetivo es predecir k centroides y una categoría $c^{(i)}$ para cada dato. El algoritmo k-medias actúa de la siguiente forma:

1. Inicializar de forma aleatoria K centroides $u_{(1)}, u_{(2)}, ..., u_{(k)} \in R^{(n)}$
2. Asignar cada dato al grupo con el centroide más cercano (distancia euclidia mínima). Siendo $c^{(i)}$ el índice del centroide al qe el dato $x^{(i)}$ está asignado.

$$c^{(1)} := argmin||x^{(i)} - u_{(k)}||^2$$

3. Cuando todos los datos han sido asignados, recalcular la posición k de los centroides.

$$u_{(k)} := \text{media de los puntos asignados al cluster } k$$

4. El algoritmo itera entre los puntos 2 y 3 hasta que se cumpla alguno de los criterios de parada (los datos no cambian de cluster, se minimiza la suma de las distancias o se alcanza el número máximo de iteraciones).

Finalmente, este algoritmo como muchos otros busca minimizar una función objetivo, que en este caso se trata de:

$$J(c^{(1)}, ..., c^{(m)}, u_i, ..., u_k) = \frac{1}{m} \sum_{i=1}^{m} ||x^{(i)} - u_{(c)}(i)||^2$$

También conocida como función de distorsión, mide la distancia al cuadrado entre cada dato y su centroide asignado. El algoritmo de k-medias minimiza esta función de dos formas:

- Cuando se asigna cada objeto al centroide más cercano.
- En el paso de mover los centroides, la posición de los centroides se actualiza tomando como nuevo centroide la posición del promedio de los objetos que pertencen a este grupo.

Está demostrado que k-medias siempre converge a un resultado (Selim e Ismail, 1984). Este resultado puede ser un óptimo local, que no siempre es la mejor opción, por lo que realizar más de una ejecución del algortimo puede proporcionar una mejor alternativa.

3.4. Inicialización aleatoria

La inicialización de los centroides en k-medias no es algo trivial, una mala elección de centroides puede llevar a que el algoritmo se quede .ªtascado.en un óptimo local, impidiendo así que se alcance una mejor solución.

Para evitar esto lo ideal es que los centroides iniciales estén suficientemente alejados unos de otros. Una solución para este problema con bastante buenos resultados es realizar un inicio multiple aleatorio que consiste en elegir los centroides iniciales entre los puntos de la muestra de entrenamiento e iniciar y ejecutar k-medias varias veces para quedarse con la solución que minimice la función de distorsión.

Si el número de clusters es relativamente pequeño (entre 2 y 10), el realizar un inicio multiple aleatorio ayudará mucho a reduchir la posibilidad de acabar con soluciones subóptimas. Sin embargo, para problemas en los que el número k de clusters es muy elevado, es más probable que el inicio aleatorio proporcione una solución aceptable a la primera, por lo que realizar multiples inicios solo mejorará la solución muy ligramente.

3.5. Elegir el número de clusters K

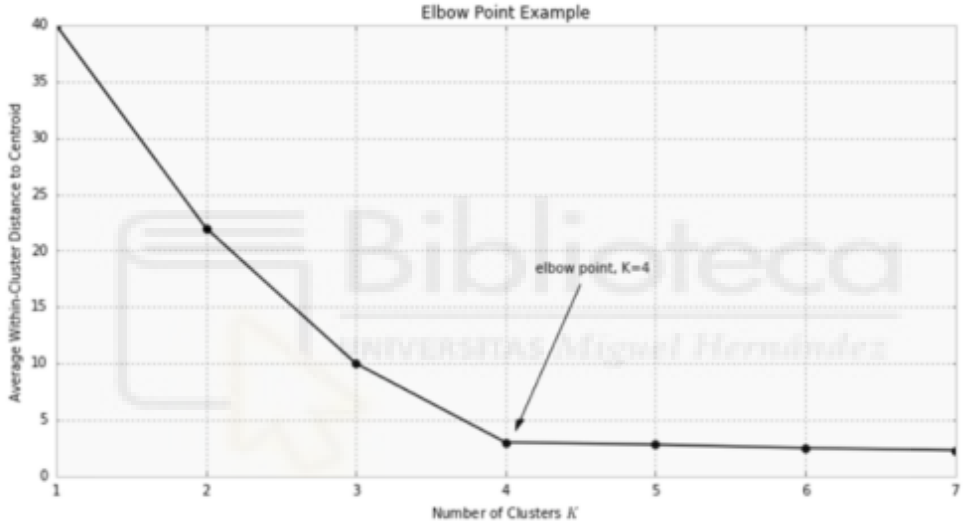
El algoritmo de k-medias no es capaz de determinar el número de clusters por su cuenta por lo que el parámetro K deberá ser fijado antes de ejecutar k-medias.

No siempre es buena idea resolver este problema de forma automática. Para elegir K se necesita ejecutar el algoritmo de k-medias para una rango de valores K y comparar los resultados. En general, no hay forma de determinar el mejor K de forma exacta, pero si que se puede conseguir una estimación bastante acertada utilizando las siguientes técnicas.

Si nuestros datos contienen 3 o menos variables sería muy fácil representarlos gráficamente y observar la estructura que forman para detectar posibles agrupamientos y determinar así un valor K estimado. Desafortunadamente este no suele ser el caso y se deberán usar otros métodos para determinar K.

Una de las medidas más comunmente utilizadas para comparar resultados entre distintos valores de K es la distancia media entre los datos y los centroides de los clusters. Como al aumentar el número de clusters se reduce la distancia a los datos, incrementar K siempre conllevará una reducción de la distancia media a los centroides, hasta el punto de llegar a cero cuando K es igual al número de datos.

El método Elbow o método del codo utiliza esta distancia representada frente a K para determinar el número de clusters ideal para el problema en cuestión



Si aplicamos k-medias con distintos valores de K mientras a su vez calculamos la distancia media de los puntos a cada centroide, acabaremos con una gráfica similar a la anterior. Si tenermos suerte, se puede comprobar que en la gráfica llega un punto en el que un aumento de K no conlleva una reducción significativa de la distancia, es punto se llama çodo.[ei]ndica el número de clusters K que se debe tomar para el algortimo de k-medias.

No obstante el método del codo no se utiliza mucho debido a que lo más común es que no salga una gráfica como la del ejemplo, sino que la distancia se vaya reduciendo de forma continua y que el punto de inflexión para obtener K no esté tan claro.

Otra forma de calcular el número de clusters es mediante el analisis de la varianza, como nuestro objetivo es agregar objetos con las mismas características en el mismo cluster y los objetos diferentes ubicarlos en clusters diferentes, utilizararemos los criterios de cohesión y separación:

- Cohesión: Cada objeto del mismo clúster debe ser lo más cercano posible al resto de objetos que engloba el clúster.
- Separación: Los clúster tienen que estar lo más separados posible unos de otros. Para medir la distancia entre clústers, podemos utilizar la distancia entre centroides.

Suma al cuadrado de los errores dentro del clúster (SSW)

Esta medida se usa especialmente para determinar la cohesión de los clústeres que se han generado.

$SSW = \sum_{i=1}^{k} \sum_{x \in p_i} d^2(x, p_i)$

Sea K el número de clústeres, x un punto del clúster p_i y c_i el centroide del clúster p_i.

Suma al cuadrado de los errores entre clústers (SSB)

Esta medida de sepación se utiliza para determinar la distancia entre clústers (separación).

$SSB = \sum_{i=1}^{k} n_i d^2(\mu, c_i)$

Sea k el número de clústeres, n_j el número de instancias en el clúster i, c_i el centroide del clúster i y μ es la media de todo el data set.

Con esto calculamos $F = SSB/SSW$, buscando obtener el menos valor de SSW y mayor valor de SSB.

3.6. Selección de características en el método de las K-medias

Una de las limitaciones del algoritmo de las k-medias es que no nos indica que características de nuestra base de datos utilizar para obtener el resultado óptimo, por tanto, para resolver esta limitación tendremos que repetir el experimento tantas veces como variables queramos seleccionar. Es decir, en el ejemplo que presentamos como un caso sencillo de uso, en el que tenemos cuatro variables tendremos que repetir el algoritmo: Si queremos seleccionar las cuatro variables, tan solo hay que realizar una vez el algoritmo. Si queremos seleccionar tres de las cuatro variables, habrá que repetir el algortimo cuatro veces. Si queremos seleccionar dos varibales, repetiremos el experimento seis veces. Si queremos seleccionar una única varible, repetiremos el experimento cuatro veces.

3.7. Ventajas de K-medias

El algoritmo de K-meas es uno de los métodos de clustering más sencillos y más utilizados por númerosas razones:

- Muy sencillo de implementar y ejecutar
- Funciona bien con grandes cantidades de datos y requiere tiempos de computacion mucho más reducidos que otros métodos de clustering
- Genera clusters más concentrados que otros métodos.
- Resultados fáciles de interpretar
- Ideal solución para hacer un pre-clustering reduciendo el espacio para poder aplicar otros algoritmos de clustering

3.8. Limitaciones y desventajas de K-medias

- Dificultades para predecir el valor de K
- Solo funciona con datos numéricos, las variables categóricas deben ser modificadas
- Baja capacidad de evitar óptimos locales
- Muy sensibe al casos aislados y ruido
- No funciona bien con clusters no globulares
- Diferentes particiones iniciales pueden resultar en distintos clusters obtenidos. Es necesario ejecutarlo varias veces para comprobar resultados
- Es difícil comprobar la calidad de los clusters
- Solo se pueden visualizar los clusters en espacios de hasta tres dimensiones

4. Un modelo de programación lineal para la construcción de p clusters y la selección de q características.

La formulación del problema es:

$$f(z,x,w) = min \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{r} d_{ijk}w_{ijk} \tag{1}$$

$$s.t. \sum_{j=1}^{m} w_{ijk} = qx_{ik} \qquad \forall i \in U,\ \forall k \in V, \tag{2}$$

$$\sum_{k=1}^{r} x_{ik} = 1 \qquad \forall i \in V, \tag{3}$$

$$x_{ik} \leq y_k \qquad \forall i \in U, \forall k \in R, \tag{4}$$

$$\sum_{k=1}^{r} y_k = p, \tag{5}$$

$$\sum_{k=1}^{r} w_{ijk} \leq z_j \qquad \forall i \in U,\ \forall j \in V, \tag{6}$$

$$\sum_{j=1}^{m} z_j = q, \tag{7}$$

$$w_{ijk} \in \{0,1\} \qquad \forall i \in U,\ \forall j \in V, \tag{8}$$

$$x_{ij} \in \{0,1\} \qquad \forall i \in U,\ \forall j \in V, \tag{9}$$

$$z_j \in \{0,1\} \qquad \forall j \in V, \tag{10}$$

$$y_k \in \{0,1\} \qquad \forall k \in R, \tag{11}$$

Conjuntos y parámetros:

- $U = \{1, \ldots, n\}$ Objetos
- $V = \{1, \ldots, m\}$ Características
- $R = \{1, \ldots, r\}$ Centros de cluster
- q Parámetro definido por el usuario, número de características seleccionadas, para realizar el análisis.
- p Parámetro definido por el usuario, número de clústeres seleccionados.
- $d_{ijk}, i \in U, j \in V, k \in R$, distancia entre el objeto i y el centro del cluster k medida por la característica j

Variables de decisión del modelo:

- $z_j, j = 1, ..., m$, representa si la característica j se elije o no.
  $z_j = 1$ si $j \in Q$. La características j es relevante.
  $z_j = 0$ si $j \notin Q$. La características j NO es relevante.
- $x_{ik}, i \in U, k \in R$, variable de asignación global del objeto i al centro del cluster k.
  $x_{ik} = 1$ si $i \in k$. Objeto i se asigna al centro del cluster k.
  $x_{ik} = 0$ si $i \notin k$. Objeto i NO se asigna al centro del cluster k.
- $w_{ijk}, i \in U, j \in V, k \in R$, Variable de asignación del objeto i al centro del cluster k siendo la característica j relevante.
  $w_{ijk} = 1$ si el objeto i está asignado al centro del cluster k y la variable j es relevante.

- $y_k, k \in R$, variable que representa si el cluster k se usa: $y_k = 1$ si el cluster k se usa, $y_k = 0$ en caso contrario.

Función Objetivo:

$\mathrm{f}(z, x, w) = min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{r} d_{ijk} w_{ijk}$

La función objetivo busca minimizar la suma de las distancias de los objetos a los centroides y cada distancia se calcula como la suma de las distancias de las diferentes características al centro del cluster. Esta función representa la suma ponderada de las distancias $\mathrm{d}_{ijk} w_{ijk}$, donde:

-La variable d_{ijk}, representa la distancia entre el objeto i y el centro del cluster k medida por la característica j. Esta variable se utiliza para modelar la relación entre los objetos, los centros de clústeres y las características.
-La variable w_{ijk}, es una variable de asignación del objeto i al centro del clúster k, teniendo en cuenta la relevancia de la característica j. Si $\mathrm{w}_{ijk} = 1$, significa que el objeto i está asignado al centro del clúster k y la característica j es relevante. Si $\mathrm{w}_{ijk} = 0$, significa que el objeto i no está asignado al centro del clúster k o la característica j no es relevante.

Restricciones:

- $\sum_{j=1}^{m} w_{ijk} = q x_{ik} \qquad \forall i \in U, \forall k \in V,$

  Establece que ninguna asignación local i, k que use la variable j es factible, a menos que exista la asignacón global i, k. Además el número total de asignaciones locales i, k, es exactamente q.

  De forma más sencilla, podemos decir que con esta restricción, aseguramos que cada objeto i, solo puede estar asígnado a un cluster k, mediante la variable j, cuando el cluster haya sido seleccionado y se seleccionarán tantas variables como el usuario haya expecificado, con el parámetro q.

- $\sum_{k=1}^{r} x_{ik} = 1 \qquad \forall i \in V,$

  Establece que cada unidad i, se asigna a un único cluster k. Es decir, un objeto solo se puede asignar a un cluster.

- $x_{ik} \leq y_k \qquad \forall i \in U, \forall k \in R,$

  Establece que cada unidad i se asigna a un cluster k, siempre que el cluster k haya sido elegido.

- $\sum_{k=1}^{r} y_k = p$

  Esta restricción establece que exactamente p clusters deben ser utilizados.

- $\sum_{k=1}^{r} w_{ijk} \leq z_j \qquad \forall i \in U, \ \forall j \in V,$

  Establece que ninguna asignación local i, k que use la variable j es factible, a menos que la variable j haya sido seleccionada. Es decir, si la variable z no se selecciona en el modelo, no podemos realizar ninguna asignación con esa variable.

- $\sum_{j=1}^{m} z_j = q$

  La suma de variables j elegidas, tiene que ser igual al parámetro q, que indica el usuario como el número de características seleccionadas.

Ciudad/Gasto	Luz	Agua	Basura	Otros
Alicante	3	8	5	2
Valencia	2	10	7	3
Madrid	60	26	25	22

4.1. Ejemplo del módelo en un caso sencillo

En nuestro ejempo, queremos agrupar las ciudadades por el gasto en luz, agua, basura y otros.

Variables:

- $U = \{1, 2, 3\}$ Ciudades
- $V = \{1, 2, 3, 4\}$ Gastos
- $R = \{1, 2\}$ Centros de cluster
- $q = 4$ Para realizar el ejemplo, vamos a elegir 4 características.
- $p = 2$ Para realizar el ejemplo, vamos a seleccionar 2 clusters.
- $d_{ijk}, i \in U, j \in V, k \in R$, distancia entre cada ciudad i y el centro del cluster k medida por los gastos j

Código en R para resolver nuestro problema de ejemplo:

```
#Insertamos los datos de nuestra tabla.


datos <- matrix(c(3,8,5,2,
                  2,10,7,3,
                  60,26,25,22), nrow=3, byrow = T)
colnames(datos) <- c("Luz","Agua","Basura","Otros")
rownames(datos) <- c("Sevilla","Valencia","Zaragoza")
datos

##          Luz Agua Basura Otros
## Sevilla    3    8      5     2
## Valencia   2   10      7     3
## Zaragoza  60   26     25    22

num_caracteristicas_a_seleccionar <- 4
num_clusters_a_crear <- 2
num_ciudades <-nrow(datos)
num_ciudades

## [1] 3

num_caracteristicas <-ncol(datos)
num_caracteristicas

## [1] 4

num_clusters_iniciales <-nrow(datos)
num_clusters_iniciales

## [1] 3

ijk<-num_caracteristicas*num_ciudades*num_clusters_iniciales
i_2<-num_ciudades^2
```

```r
#Vamos a definir el número de columna donde comienza y terminan nuestras variables de
#decisión del módelo para crear la matriz de restricciones que resolverá el problema.

col0_W <- 0
total_cols_W <- num_ciudades*num_caracteristicas*num_clusters_iniciales
col0_X <- col0_W + total_cols_W
total_cols_X <- num_ciudades*num_clusters_iniciales
col0_Z <- col0_X + total_cols_X
total_cols_Z <- num_caracteristicas
col0_Y <- col0_Z + total_cols_Z
total_cols_Y <- num_clusters_iniciales

total_columnas <- total_cols_W+total_cols_X+total_cols_Z+total_cols_Y
total_columnas

## [1] 52

total_filas <- total_cols_X + num_ciudades + total_cols_X + 1 +
               (num_ciudades*num_caracteristicas) + 1
total_filas

## [1] 35
```

```r
#Creamos la matriz de distancias.

distancia<- c()
for(ciudad in 1:num_ciudades)
  for(caracteristica in 1:num_caracteristicas)
    for(ciudad2 in 1:num_ciudades)
      distancia <- c(distancia, (datos[ciudad,caracteristica]
      - datos[ciudad2,caracteristica])^2)

distancia <- c(distancia, replicate(total_columnas
               - num_ciudades*num_caracteristicas*num_ciudades, 0))
distancia

##  [1]    0    1 3249    0    4  324    0    4  400    0    1  400    1    0 3364
## [16]    4    0  256    4    0  324    1    0  361 3249 3364    0  324  256    0
## [31]  400  324    0  400  361    0    0    0    0    0    0    0    0    0    0
## [46]    0    0    0    0    0    0    0
```

```r
#Definimos la dimensión de la matriz de restricciones y creamos los vectores
#de direciones y rhs.


restricciones <- matrix(0, nrow = total_filas ,ncol=total_columnas)
dir<-c()
b<-c()
```

```r
#Completamos la matriz de restricciones y los vectores de direcciones y rhs
#utilizando los valores que hemos almacenado en los pasos anteriores.

fila <- 0
for (ciudad in 1:num_ciudades) {
```

```
  for (cluster in 1:num_clusters_iniciales) {
    fila <- fila+1

    for(caracteristica in 1:num_caracteristicas){
      restricciones[fila, (ciudad - 1) * num_ciudades * num_caracteristicas + cluster
                    + (caracteristica - 1) * num_ciudades] = 1
    }

    restricciones[fila, col0_X + fila] = -num_caracteristicas_a_seleccionar

    dir<-c(dir,'=')
    b<-c(b,0)
  }
}


for(ciudad in 1:num_ciudades){
     fila <- fila+1
     for(cluster in 1:num_clusters_iniciales){
        restricciones[fila, col0_X+cluster+(ciudad-1)*num_ciudades] = 1
       }

     dir<-c(dir,'=')
     b<-c(b,1)
}


contador<-0
for (ciudad in 1:num_ciudades) {
  for (cluster in 1:num_clusters_iniciales) {
    fila <- fila+1
    contador <-contador+1
    restricciones[fila, col0_X + contador] = 1

    for(k in 1:num_clusters_iniciales){
      restricciones[fila, col0_Y + cluster]=-1
    }

    dir<-c(dir,'<=')
    b<-c(b,0)
  }
}


for(contador in 1:1){
  fila <- fila + 1
  for(cluster in 1:num_clusters_iniciales){
      restricciones[fila, col0_Y + cluster] = 1
  }

  dir<-c(dir,'=')
  b<-c(b,num_clusters_a_crear)
}


for (ciudad in 1:num_ciudades) {
```

```
  for(caracteristica in 1:num_caracteristicas){

    fila <- fila+1
    for (cluster in 1:num_clusters_iniciales) {
      restricciones[fila, (ciudad - 1) * num_ciudades * num_caracteristicas + cluster
                    + (caracteristica - 1) * num_ciudades] = 1
    }

    restricciones[fila, col0_Z + caracteristica] = -1

    dir<-c(dir,'<=')
    b<-c(b,0)
  }
}

for(contador in 1:1){
  fila <- fila + 1
  for(caracteristica in 1:num_caracteristicas){
    restricciones[fila, col0_Z + caracteristica] = 1
  }

  dir<-c(dir,'=')
  b<-c(b,num_caracteristicas_a_seleccionar)

}
```

```
#Así quedarían la matriz de restricciones y los vectores de direcciones y rhs

#restricciones
dir

##  [1] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "<=" "<=" "<="
## [16] "<=" "<=" "<=" "<=" "<=" "<=" "="  "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [31] "<=" "<=" "<=" "<=" "="

b

##  [1] 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 4
```

- Solución, seleccionando cuatro características y agrupando en dos clústers:

```
#solopt <- lp('min',distancia,restricciones,dir,b,all.bin = TRUE)
```

```
> solopt <- lp('min',distancia,restricciones,dir,b,all.bin = TRUE)
> solopt
Success: the objective function is 10
```

```
#solopt\$solution
#vector_solucion<-c(solopt\$solution)
vector_solucion<-c(1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,1,1,0,0,1,0,
vector_solucion

##  [1] 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0
## [39] 0 1 0 0 0 0 1 1 1 1 1 1 0 1
```

```
#solopt\$objective
#matriz_distancias<-c(solopt\$objective)
matriz_distancia<-c(0,1,3249,0,4,324,0,4,400,0,1,400,1,0,3364,4,0,256,4,0,324,1,0,361,3249,3364,0,324
matriz_distancia

##  [1]    0    1 3249    0    4  324    0    4  400    0    1  400    1    0 3364
## [16]    4    0  256    4    0  324    1    0  361 3249 3364    0  324  256    0
## [31]  400  324    0  400  361    0    0    0    0    0    0    0    0    0    0
## [46]    0    0    0    0    0    0    0
```

¿Cómo interpretamos esta solución que nos devuelve R?

1. La distancia mínima entre cada ciudad i y el centro del cluster k medida por los gastos j es 10.
2. La función 'solopt$solution' nos devuelve un vector de 1 y 0 donde:
   - En las primeras $i*j*k$ (36) posiciones nos indica los puntos que minimizan nuestra matriz de distancia.
   - En las siguientes $i*i$ (9) posiciones nos indica a que centro del cluster se asigna cada ciudad.
   - En las siguientes k (4) posiciones nos indica las características elegidas
   - En las últimas k posiciones nos indica los cluster elegidos para minimizar el problema.
3. Por último, la función 'solopt$objective' nos devuelve la matriz de distancias.

Con estas indicaciones podriamos llegar a la conclusión de que para minimizar el problema, se eligen los clústers 1 'Sevilla' y 3 'Zaragoza', Sevilla se asigna al clúster 1, Valencia se asigna al clúster 1 y Zaragoza se asigna al clúster 3, como comentamos se han elegido las cuatro características (luz, agua, basura y otros).
Para facilitar la interpretación he creado el siguiente código que nos retorna de fomar mucho más visible e interpretable la solución:

```
#Cluster_asignado <- solopt\$solution[(ijk+1):(ijk+i_2)]

Cluster_asignado <- c(1,0,0,1,0,0,0,0,1)
objetos<-rownames(datos)
Objetos <- rep(objetos, times = rep(cluster, length(objetos)))
Cluster <- seq_along(rownames(datos))
Asignacion <- cbind(Objetos, Cluster, Cluster_asignado )
Asignacion

##       Objetos    Cluster Cluster_asignado
##  [1,] "Sevilla"  "1"     "1"
##  [2,] "Sevilla"  "2"     "0"
##  [3,] "Sevilla"  "3"     "0"
##  [4,] "Valencia" "1"     "1"
##  [5,] "Valencia" "2"     "0"
##  [6,] "Valencia" "3"     "0"
##  [7,] "Zaragoza" "1"     "0"
##  [8,] "Zaragoza" "2"     "0"
##  [9,] "Zaragoza" "3"     "1"
```

```
#caracteristicas_elegida <- solopt\$solution[(ijk+i_2+1):(ijk+i_2+num_caracteristicas)]
caracteristicas_elegida<-c(1,1,1,1)
Caracteristica<-colnames(datos)
Caracteristicas_elegidas <- cbind(Caracteristica, caracteristicas_elegida)
Caracteristicas_elegidas
```

```
##      Caracteristica caracteristicas_elegida
## [1,] "Luz"          "1"
## [2,] "Agua"         "1"
## [3,] "Basura"       "1"
## [4,] "Otros"        "1"
```

```
#cluster_elegidos <- solopt\$solution[(ijk+i_2+num_caracteristicas+1):(length(vector_solucion))]
cluster_elegidos<-c(1,0,1)
Cluster <- seq_along(rownames(datos))
Cluster_elegidos <- cbind(Cluster, cluster_elegidos)
Cluster_elegidos
```

```
##      Cluster cluster_elegidos
## [1,]       1                1
## [2,]       2                0
## [3,]       3                1
```

- Solución, seleccionando tres características y agrupando en dos clústers:

  Se eligen los clústers 2 'Valencia' y 3 'Zaragoza', Sevilla se asigna al clúster 2, Valencia se asigna al clúster 2 y Zaragoza se asigna al clúster 3, las características seleccionadas han sido luz, agua y otros.

```
#Cluster_asignado <- solopt\$solution[(ijk+1):(ijk+i_2)]

Cluster_asignado <- c(0,1,0,0,1,0,0,0,1)
objetos<-rownames(datos)
Objetos <- rep(objetos, times = rep(cluster, length(objetos)))
Cluster <- seq_along(rownames(datos))
Asignacion <- cbind(Objetos, Cluster, Cluster_asignado )
Asignacion
```

```
##       Objetos    Cluster Cluster_asignado
##  [1,] "Sevilla"  "1"     "0"
##  [2,] "Sevilla"  "2"     "1"
##  [3,] "Sevilla"  "3"     "0"
##  [4,] "Valencia" "1"     "0"
##  [5,] "Valencia" "2"     "1"
##  [6,] "Valencia" "3"     "0"
##  [7,] "Zaragoza" "1"     "0"
##  [8,] "Zaragoza" "2"     "0"
##  [9,] "Zaragoza" "3"     "1"
```

```
#caracteristicas_elegida <- solopt\$solution[(ijk+i_2+1):(ijk+i_2+num_caracteristicas)]
caracteristicas_elegida<-c(1,1,0,1)
Caracteristica<-colnames(datos)
Caracteristicas_elegidas <- cbind(Caracteristica, caracteristicas_elegida)
Caracteristicas_elegidas
```

```
##      Caracteristica caracteristicas_elegida
## [1,] "Luz"          "1"
## [2,] "Agua"         "1"
## [3,] "Basura"       "0"
## [4,] "Otros"        "1"
```

```
#cluster_elegidos <- solopt\$solution[(ijk+i_2+num_caracteristicas+1):(length(vector_solucion))]
cluster_elegidos<-c(0,1,1)
Cluster <- seq_along(rownames(datos))
Cluster_elegidos <- cbind(Cluster, cluster_elegidos)
Cluster_elegidos

##      Cluster cluster_elegidos
## [1,]       1                0
## [2,]       2                1
## [3,]       3                1
```

- Solución, seleccionando dos características y agrupando en dos clústers:

  Se eligen los clústers 1 'Sevilla' y 3 'Zaragoza', Sevilla se asigna al clúster 1, Valencia se asigna al clúster 1 y Zaragoza se asigna al clúster 3, las características seleccionadas han sido luz y otros.

```
#Cluster_asignado <- solopt\$solution[(ijk+1):(ijk+i_2)]

Cluster_asignado <- c(1,0,0,1,0,0,0,0,1)
objetos<-rownames(datos)
Objetos <- rep(objetos, times = rep(cluster, length(objetos)))
Cluster <- seq_along(rownames(datos))
Asignacion <- cbind(Objetos, Cluster, Cluster_asignado )
Asignacion

##       Objetos    Cluster Cluster_asignado
##  [1,] "Sevilla"  "1"     "1"
##  [2,] "Sevilla"  "2"     "0"
##  [3,] "Sevilla"  "3"     "0"
##  [4,] "Valencia" "1"     "1"
##  [5,] "Valencia" "2"     "0"
##  [6,] "Valencia" "3"     "0"
##  [7,] "Zaragoza" "1"     "0"
##  [8,] "Zaragoza" "2"     "0"
##  [9,] "Zaragoza" "3"     "1"
```

```
#caracteristicas_elegida <- solopt\$solution[(ijk+i_2+1):(ijk+i_2+num_caracteristicas)]
caracteristicas_elegida<-c(1,0,0,1)
Caracteristica<-colnames(datos)
Caracteristicas_elegidas <- cbind(Caracteristica, caracteristicas_elegida)
Caracteristicas_elegidas

##      Caracteristica caracteristicas_elegida
## [1,] "Luz"          "1"
## [2,] "Agua"         "0"
## [3,] "Basura"       "0"
## [4,] "Otros"        "1"
```

```
#cluster_elegidos <- solopt\$solution[(ijk+i_2+num_caracteristicas+1):(length(vector_solucion))]
cluster_elegidos<-c(1,0,1)
Cluster <- seq_along(rownames(datos))
Cluster_elegidos <- cbind(Cluster, cluster_elegidos)
Cluster_elegidos

##      Cluster cluster_elegidos
## [1,]       1                1
## [2,]       2                0
## [3,]       3                1
```

- Solución, seleccionando una característica y agrupando en dos clústers:

  Se eligen los clústers 1 'Sevilla' y 3 'Zaragoza', Sevilla se asigna al clúster 1, Valencia se asigna al clúster 1 y Zaragoza se asigna al clúster 3, la característica seleccionada ha sido otros.

```
#Cluster_asignado <- solopt\$solution[(ijk+1):(ijk+i_2)]

Cluster_asignado <- c(1,0,0,1,0,0,0,0,1)
objetos<-rownames(datos)
Objetos <- rep(objetos, times = rep(cluster, length(objetos)))
Cluster <- seq_along(rownames(datos))
Asignacion <- cbind(Objetos, Cluster, Cluster_asignado )
Asignacion

##       Objetos    Cluster Cluster_asignado
##  [1,] "Sevilla"  "1"     "1"
##  [2,] "Sevilla"  "2"     "0"
##  [3,] "Sevilla"  "3"     "0"
##  [4,] "Valencia" "1"     "1"
##  [5,] "Valencia" "2"     "0"
##  [6,] "Valencia" "3"     "0"
##  [7,] "Zaragoza" "1"     "0"
##  [8,] "Zaragoza" "2"     "0"
##  [9,] "Zaragoza" "3"     "1"
```

```
#caracteristicas_elegida <- solopt\$solution[(ijk+i_2+1):(ijk+i_2+num_caracteristicas)]
caracteristicas_elegida<-c(0,0,0,1)
Caracteristica<-colnames(datos)
Caracteristicas_elegidas <- cbind(Caracteristica, caracteristicas_elegida)
Caracteristicas_elegidas

##      Caracteristica caracteristicas_elegida
## [1,] "Luz"          "0"
## [2,] "Agua"         "0"
## [3,] "Basura"       "0"
## [4,] "Otros"        "1"
```

```
#cluster_elegidos <- solopt\$solution[(ijk+i_2+num_caracteristicas+1):(length(vector_solucion))]
cluster_elegidos<-c(1,0,1)
Cluster <- seq_along(rownames(datos))
Cluster_elegidos <- cbind(Cluster, cluster_elegidos)
Cluster_elegidos
```

```
##      Cluster cluster_elegidos
## [1,]       1                1
## [2,]       2                0
## [3,]       3                1
```

4.2. Ejemplo del módelo en un caso de uso real

Antes de introducirnos en el caso de uso real es importante hacer una pequeña introducción sobre el concepto **analítica web**, ya que nuestro ejemplo se va a utilizar en esta rama de análisis y actualmente mi puesto de trabajo es analista web en un grupo editorial a nivel nacional.

La analítica web es el proceso de recopilación, medición, análisis y reporte de datos relacionados con el comportamiento de los usuarios en un sitio web o aplicación. El objetivo principal de la analítica web es comprender cómo los usuarios interactúan con un sitio o una aplicación, con el fin de tomar decisiones informadas para mejorar su rendimiento y lograr los objetivos establecidos. Las labores de un analista web suelen incluir:

- Recopilación de datos: El analista web utiliza herramientas de análisis para recopilar datos sobre el tráfico del sitio web, como el número de visitantes, las páginas vistas, el tiempo de permanencia, las conversiones, entre otros. Esto implica configurar y gestionar herramientas como Google Analytics [1] [2] u otras soluciones de análisis web.
- Análisis de datos: El analista web analiza los datos recopilados para obtener información relevante sobre el comportamiento de los usuarios. Esto puede implicar la identificación de patrones de navegación, la segmentación de audiencias, el análisis de embudos de conversión y la evaluación del rendimiento de las campañas de marketing digital, entre otros aspectos.
- Generación de informes: El analista web crea informes y presenta los hallazgos de manera clara y comprensible para los diferentes stakeholders, como gerentes, equipos de marketing o desarrolladores web. Los informes suelen incluir métricas clave, análisis de tendencias y recomendaciones para mejorar el rendimiento del sitio web.
- Optimización del sitio web: El analista web utiliza los datos y los análisis para identificar oportunidades de mejora en el sitio web o la aplicación. Esto puede implicar la optimización de la experiencia del usuario, la identificación de páginas con bajo rendimiento, la prueba de elementos de diseño o la personalización del contenido para diferentes segmentos de usuarios.
- Evaluación de campañas: El analista web evalúa el rendimiento de las campañas de marketing digital, como anuncios pagados, campañas de correo electrónico o estrategias de SEO. Esto implica medir la efectividad de las campañas, identificar áreas de mejora y realizar ajustes para maximizar el retorno de la inversión.

Algunas de las ventajas que obtenemos al utilizar la analítica web son:

- Toma de decisiones informadas: La analítica web proporciona datos y conocimientos concretos sobre el comportamiento de los usuarios, lo que permite tomar decisiones basadas en evidencia en lugar de suposiciones o intuiciones.
- Optimización del rendimiento: Al comprender cómo los usuarios interactúan con un sitio web, se pueden identificar oportunidades de mejora y optimizar el rendimiento general. Esto puede conducir a un aumento de la conversión, la retención de usuarios y la satisfacción del cliente.
- Mejora del retorno de la inversión (ROI): Al evaluar y optimizar las campañas de marketing digital, se puede maximizar el retorno de la inversión al dirigir los recursos de manera más efectiva y centrarse en las estrategias que generan resultados positivos.

[1] Google Analytics es una plataforma que recoge datos de páginas web y aplicaciones para crear informes que proporcionan estadísticas sobre dichas webs.

- Personalización y segmentación: La analítica web permite segmentar a los usuarios en función de su comportamiento y preferencias, lo que facilita la personalización del contenido y la entrega de experiencias más relevantes y personalizadas.
- Identificación de oportunidades de crecimiento: Al analizar los datos, se pueden identificar nuevas oportunidades de crecimiento, como ident

Para mostrar como funciona el módelo en un caso real, hemos estraido desde Google Analytics datos [2] de las webs: https://www.elperiodico.com/es/, https://www.informacion.es/, https://www.lne.es/ y https://www.farodevigo.es/.
En la base de datos, incluimos la información de doscientos usuarios de la webs, tras examinar su comportamiento durante un mes, para analizar como podriamos agrupar a una muestra aleatoria de los letores en función de las dimensiones y métricas elegidas.
Dimensiones:

1. Id Usuario: Id aleatorio que se le asigna al navegador.
2. Periódico: Nombre de la web que ha visitado el usuario.
3. Dispositivo desde el que el usuarios accede a la web: ordenador, tablet o móvil.
4. Sexo: Género del navegador.
5. User Type: Tipo de usuario, diferenciamos entre anónimos, registrados y suscriptores.
6. Edad del navegador.

Métricas:

1. Páginas Vistas: Las páginas vistas son la cantidad total de veces que se ha visto una página en el sitio web.
2. Sesiones: Las sesiones generalmente se describen como el grupo de interacciones que ocurren en su sitio durante un período de tiempo determinado, generalmente de 30 minutos.
3. Duración de la sesión.
4. Rebotes: La tasa de rebote se calcula cuando alguien visita una página individual de tu web y no hace nada en la página antes de abandonarla.
5. Scroll: Mide cuanto se ha desplazado (profundidad) un usuario en la página web.

ID Usuario	Periódico	Dispositivo	Sexo	Tipo usuario	Edad	Páginas Vistas	Sesiones	Duración sesión	Rebotes	Scroll
user1	EP	tablet	female	suscriptor	50	843	73	533	150	25
user2	LNE	mobile	male	anonimo	40	600	203	654	124	75
user3	INF	desktop	female	registrado	30	305	152	469	86	50

El objetivo de este análisis es agrupar a los usuarios de la webs para ofrecerles contenidos afines a sus gustos en función del análisis de su comportamiento en anteriores visitas.

Finalmente para poder mostrar un ejemplo de uso real, con las limitaciones actuales de la versión de R, hemos tenido que reducir a doce el número de usuarios analizados, ya que cuando pasamos este umbral el tiempo de computo crece exponencialmente, no llegando a resolver el problema en tiempos de computo superiores a una hora.
Con estas limitaciones de tiempos de ejecución, hemos deccido crear dos clusters y seleccionar cuatro características de las aportadas. Llegando a la conclusión de que las carácteristicas relevantes para agrupar a los usuarios en clusters son: la edad, el número de páginas vistas, la duración de la sesión y los rebotes, dejando fuera del análisis las sesiones y el scroll. Lo que según mi experiencia tiene sentido ya que las sesiones y páginas vistas son variables con un alto grado de correlación y el scroll donde solo aportamos los valores (25, 50, 75, 100) no aporta demasiada información.

Los clusters quedan formados por:

[2] Datos anonimizados y que han sido tratados para no aportar datos reales sobre ninguna web del grupo editorial

- Cluster 1 (cluster liderado por el usuario 3): Usuarios 3 y 7.
- Cluster 2 (cluster liderado por el usuario 12): Resto de usuarios.

Además de las variables cuantitativas (edad, páginas vistas, duración de la sesión y rebotes) que son las elegidas por nuestro modelo para crear los cluster, en la siguiente tabla podemos ver que las principales diferencias entre los lideres de los cluster son el periódico al que acceden y el dispositivo de consumo estas características nos permiten de una forma sencilla dividir a los usuarios en clusters, además de hacer una descripción de estos muy simplificada.

Usuario	Periódico	Dispositivo	Sexo	Tipo usuario	Edad	Páginas Vistas	Sesiones	Duración sesión	Rebotes	Scroll
Usuario 3	EP	desktop	male	anonimo	60	589	69	2974	16	75
Usuario 12	VIG	mobile	male	anonimo	65	199	49	283	19	75

```
#Insertamos los datos, en este caso los vamos a introducir directamente desde
#un archivo excel, por lo que el usuario tendrá que cambiar la ruta y el nombre del archivo

#Como en Overleaf no podemos incluir los datos de forma auutomática, se van a incluir de forma
#manual, pero debería utilizarse el código comentado para crear nuestra matriz con los datos
#del excel

#library(lpSolve)
#library(readxl)

#ruta_archivo <- "C:/Users/fran.molina/Desktop/Desktop_OLD/Fran/TFG/datos_ga/usuarios_6.xlsx"
#datos_excel <- read_excel(ruta_archivo)
#matriz<- as.matrix(datos_excel)
#datos_trabajo <- matriz[, c(6, 7, 8, 9, 10, 11)]
#num_filas <- nrow(datos_trabajo)
#vector_id_usuario <- matriz[, 1]
#vector_cols <- colnames(datos_trabajo)
#datos_trabajo<- as.numeric(datos_trabajo)
#datos <- matrix(c(datos_trabajo), nrow=num_filas, byrow = T)
#colnames(datos) <- c(vector_cols)
#rownames(datos) <- c(vector_id_usuario)
#datos

datos <- matrix(c(50,40,60,50,50,30,
                  40,60,50,65,40,65,
                  843,240,589,977,600,348,
                  222,77,1421,176,185,199,
                  73,40,69,261,205,50,
                  45,37,309,63,50,49,
                  1269,290,2974,304,115,2753,
                  285,129,608,233,295,283,
                  18,17,16,16,19,17,
                  18,18,124,18,19,19,
                  25,75,75,25,50,75,
                  75,25,50,100,75,75), nrow=12, byrow = T)
colnames(datos) <- c("edad","paginas_vistas","sesiones","avg_duracion_sesion","rebotes"
                     ,"avg_scroll")
rownames(datos) <- c("user1","user2","user3","user4","user5","user6",
                     "user7","user8","user9","user10","user11","user12")
datos

##        edad paginas_vistas sesiones avg_duracion_sesion rebotes avg_scroll
```

```
## user1    50             40       60                  50       50         30
## user2    40             60       50                  65       40         65
## user3   843            240      589                 977      600        348
## user4   222             77     1421                 176      185        199
## user5    73             40       69                 261      205         50
## user6    45             37      309                  63       50         49
## user7  1269            290     2974                 304      115       2753
## user8   285            129      608                 233      295        283
## user9    18             17       16                  16       19         17
## user10   18             18      124                  18       19         19
## user11   25             75       75                  25       50         75
## user12   75             25       50                 100       75         75
```

```
num_caracteristicas_a_seleccionar <- 4
num_clusters_a_crear <- 2
num_ciudades <-nrow(datos)
num_ciudades
```

```
## [1] 12
```

```
num_caracteristicas <-ncol(datos)
num_caracteristicas
```

```
## [1] 6
```

```
num_clusters_iniciales <-nrow(datos)
num_clusters_iniciales
```

```
## [1] 12
```

```
ijk<-num_caracteristicas*num_ciudades*num_clusters_iniciales
ijk
```

```
## [1] 864
```

```
i_2<-num_ciudades^2
i_2
```

```
## [1] 144
```

```
#Vamos a definir el número de columna donde comienza y terminan nuestras variables de
#decisión del módelo para crear la matriz de restricciones que resolverá el problema.

col0_W <- 0
total_cols_W <- num_ciudades*num_caracteristicas*num_clusters_iniciales
col0_X <- col0_W + total_cols_W
total_cols_X <- num_ciudades*num_clusters_iniciales
col0_Z <- col0_X + total_cols_X
total_cols_Z <- num_caracteristicas
col0_Y <- col0_Z + total_cols_Z
total_cols_Y <- num_clusters_iniciales

total_columnas <- total_cols_W+total_cols_X+total_cols_Z+total_cols_Y
total_columnas
```

```
## [1] 1026

total_filas <- total_cols_X + num_ciudades + total_cols_X + 1 +
               (num_ciudades*num_caracteristicas) + 1
total_filas

## [1] 374
```

```
#Creamos la matriz de distancias.

distancia<- c()
for(ciudad in 1:num_ciudades)
  for(caracteristica in 1:num_caracteristicas)
    for(ciudad2 in 1:num_ciudades)
      distancia <- c(distancia, (datos[ciudad,caracteristica]
      - datos[ciudad2,caracteristica])^2)

distancia <- c(distancia, replicate(total_columnas
               - num_ciudades*num_caracteristicas*num_ciudades, 0))
distancia

##     [1]       0     100  628849   29584     529      25 1485961   55225    1024
##    [10]    1024     625     625       0     400   40000    1369       0       9
##    [19]   62500    7921     529     484    1225     225       0     100  279841
##    [28] 1852321      81   62001 8491396  300304    1936    4096     225     100
##    [37]       0     225  859329   15876   44521     169   64516   33489    1156
##    [46]    1024     625    2500       0     100  302500   18225   24025       0
##    [55]    4225   60025     961     961       0     625       0    1225  101124
##    [64]   28561     400     361 7414729   64009     169     121    2025    2025
##    [73]     100       0  644809   33124    1089      25 1510441   60025     484
##    [82]     484     225    1225     400       0   32400     289     400     529
##    [91]   52900    4761    1849    1764     225    1225     100       0  290521
##   [100] 1879641     361   67081 8549776  311364    1156    5476     625       0
##   [109]     225       0  831744   12321   38416       4   57121   28224    2401
##   [118]    2209    1600    1225     100       0  313600   21025   27225     100
##   [127]    5625   65025     441     441     100    1225    1225       0   80089
##   [136]   17956     225     256 7225344   47524    2304    2116     100     100
##   [145]  628849  644809       0  385641  592900  636804  181476  311364  680625
##   [154]  680625  669124  589824   40000   32400       0   26569   40000   41209
##   [163]    2500   12321   49729   49284   27225   46225  279841  290521       0
##   [172]  692224  270400   78400 5688225     361  328329  216225  264196  290521
##   [181]  859329  831744       0  641601  512656  835396  452929  553536  923521
##   [190]  919681  906304  769129  302500  313600       0  172225  156025  302500
##   [199]  235225   93025  337561  337561  302500  275625  101124   80089       0
##   [208]   22201   88804   89401 5784025    4225  109561  108241   74529   74529
##   [217]   29584   33124  385641       0   22201   31329 1096209    3969   41616
##   [226]   41616   38809   21609    1369     289   26569       0    1369    1600
##   [235]   45369    2704    3600    3481       4    2704 1852321 1879641  692224
##   [244]       0 1827904 1236544 2411809  660969 1974025 1682209 1811716 1879641
##   [253]   15876   12321  641601       0    7225   12769   16384    3249   25600
##   [262]   24964   22801    5776   18225   21025  172225       0     400   18225
##   [271]    4900   12100   27556   27556   18225   12100   28561   17956   22201
##   [280]       0   22201   22500 6522916    7056   33124   32400   15376   15376
##   [289]     529    1089  592900   22201       0     784 1430416   44944    3025
##   [298]    3025    2304       4       0     400   40000    1369       0       9
```

```
##   [307]   62500    7921     529     484    1225     225      81     361  270400
##   [316] 1827904       0   57600 8439025  290521    2809    3025      36     361
##   [325]   44521   38416  512656    7225       0   39204    1849     784   60025
##   [334]   59049   55696   25921   24025   27225  156025     400       0   24025
##   [343]    8100    8100   34596   34596   24025   16900     400     225   88804
##   [352]   22201       0       1 7306209   54289    1089     961     625     625
##   [361]      25      25  636804   31329     784       0 1498176   57600     729
##   [370]     729     400     900       9     529   41209    1600       9       0
##   [379]   64009    8464     400     361    1444     144   62001   67081   78400
##   [388] 1236544   57600       0 7102225   89401   85849   34225   54756   67081
##   [397]     169       4  835396   12769   39204       0   58081   28900    2209
##   [406]    2025    1444    1369       0     100  302500   18225   24025       0
##   [415]    4225   60025     961     961       0     625     361     256   89401
##   [424]   22500       1       0 7311616   54756    1024     900     676     676
##   [433] 1485961 1510441  181476 1096209 1430416 1498176       0  968256 1565001
##   [442] 1565001 1547536 1425636   62500   52900    2500   45369   62500   64009
##   [451]       0   25921   74529   73984   46225   70225 8491396 8549776 5688225
##   [460] 2411809 8439025 7102225       0 5597956 8749764 8122500 8404201 8549776
##   [469]   64516   57121  452929   16384    1849   58081       0    5041   82944
##   [478]   81796   77841   41616    4225    5625  235225    4900    8100    4225
##   [487]       0   32400    9216    9216    4225    1600 7414729 7225344 5784025
##   [496] 6522916 7306209 7311616       0 6100900 7485696 7474756 7171684 7171684
##   [505]   55225   60025  311364    3969   44944   57600  968256       0   71289
##   [514]   71289   67600   44100    7921    4761   12321    2704    7921    8464
##   [523]   25921       0   12544   12321    2916   10816  300304  311364     361
##   [532]  660969  290521   89401 5597956       0  350464  234256  284089  311364
##   [541]   33489   28224  553536    3249     784   28900    5041       0   47089
##   [550]   46225   43264   17689   60025   65025   93025   12100    8100   60025
##   [559]   32400       0   76176   76176   60025   48400   64009   47524    4225
##   [568]    7056   54289   54756 6100900       0   70756   69696   43264   43264
##   [577]    1024     484  680625   41616    3025     729 1565001   71289       0
##   [586]       0      49    3249     529    1849   49729    3600     529     400
##   [595]   74529   12544       0       1    3364      64    1936    1156  328329
##   [604] 1974025    2809   85849 8749764  350464       0   11664    3481    1156
##   [613]    1156    2401  923521   25600   60025    2209   82944   47089       0
##   [622]       4      81    7056     961     441  337561   27556   34596     961
##   [631]    9216   76176       0       0     961    3136     169    2304  109561
##   [640]   33124    1089    1024 7485696   70756       0       4    3364    3364
##   [649]    1024     484  680625   41616    3025     729 1565001   71289       0
##   [658]       0      49    3249     484    1764   49284    3481     484     361
##   [667]   73984   12321       1       0    3249      49    4096    5476  216225
##   [676] 1682209    3025   34225 8122500  234256   11664       0    2401    5476
##   [685]    1024    2209  919681   24964   59049    2025   81796   46225       4
##   [694]       0      49    6724     961     441  337561   27556   34596     961
##   [703]    9216   76176       0       0     961    3136     121    2116  108241
##   [712]   32400     961     900 7474756   69696       4       0    3136    3136
##   [721]     625     225  669124   38809    2304     400 1547536   67600      49
##   [730]      49       0    2500    1225     225   27225       4    1225    1444
##   [739]   46225    2916    3364    3249       0    2500     225     625  264196
##   [748] 1811716      36   54756 8404201  284089    3481    2401       0     625
##   [757]     625    1600  906304   22801   55696    1444   77841   43264      81
##   [766]      49       0    5625       0     100  302500   18225   24025       0
##   [775]    4225   60025     961     961       0     625    2025     100   74529
##   [784]   15376     625     676 7171684   43264    3364    3136       0       0
##   [793]     625    1225  589824   21609       4     900 1425636   44100    3249
##   [802]    3249    2500       0     225    1225   46225    2704     225     144
```

```
##  [811]   70225   10816      64      49    2500       0     100       0  290521
##  [820] 1879641     361   67081 8549776  311364    1156    5476     625       0
##  [829]    2500    1225  769129    5776   25921    1369   41616   17689    7056
##  [838]    6724    5625       0     625    1225  275625   12100   16900     625
##  [847]    1600   48400    3136    3136     625       0    2025     100   74529
##  [856]   15376     625     676 7171684   43264    3364    3136       0       0
##  [865]       0       0       0       0       0       0       0       0       0
##  [874]       0       0       0       0       0       0       0       0       0
##  [883]       0       0       0       0       0       0       0       0       0
##  [892]       0       0       0       0       0       0       0       0       0
##  [901]       0       0       0       0       0       0       0       0       0
##  [910]       0       0       0       0       0       0       0       0       0
##  [919]       0       0       0       0       0       0       0       0       0
##  [928]       0       0       0       0       0       0       0       0       0
##  [937]       0       0       0       0       0       0       0       0       0
##  [946]       0       0       0       0       0       0       0       0       0
##  [955]       0       0       0       0       0       0       0       0       0
##  [964]       0       0       0       0       0       0       0       0       0
##  [973]       0       0       0       0       0       0       0       0       0
##  [982]       0       0       0       0       0       0       0       0       0
##  [991]       0       0       0       0       0       0       0       0       0
## [1000]       0       0       0       0       0       0       0       0       0
## [1009]       0       0       0       0       0       0       0       0       0
## [1018]       0       0       0       0       0       0       0       0       0
```

```
#Definimos la dimensión de la matriz de restricciones y creamos los vectores
#de direciones y rhs.


restricciones <- matrix(0, nrow = total_filas ,ncol=total_columnas)
dir<-c()
b<-c()
```

```
#Completamos la matriz de restricciones y los vectores de direcciones y rhs
#utilizando los valores que hemos almacenado en los pasos anteriores.

fila <- 0
for (ciudad in 1:num_ciudades) {
  for (cluster in 1:num_clusters_iniciales) {
    fila <- fila+1

    for(caracteristica in 1:num_caracteristicas){
      restricciones[fila, (ciudad - 1) * num_ciudades * num_caracteristicas + cluster
                    + (caracteristica - 1) * num_ciudades] = 1
    }

    restricciones[fila, col0_X + fila] = -num_caracteristicas_a_seleccionar

    dir<-c(dir,'=')
    b<-c(b,0)
  }
}


for(ciudad in 1:num_ciudades){
```

```
     fila <- fila+1
     for(cluster in 1:num_clusters_iniciales){
        restricciones[fila, col0_X+cluster+(ciudad-1)*num_ciudades] = 1
       }

     dir<-c(dir,'=')
     b<-c(b,1)
}


contador<-0
for (ciudad in 1:num_ciudades) {
  for (cluster in 1:num_clusters_iniciales) {
    fila <- fila+1
    contador <-contador+1
    restricciones[fila, col0_X + contador] = 1

    for(k in 1:num_clusters_iniciales){
      restricciones[fila, col0_Y + cluster]=-1
    }

    dir<-c(dir,'<=')
    b<-c(b,0)
  }
}


for(contador in 1:1){
  fila <- fila + 1
  for(cluster in 1:num_clusters_iniciales){
      restricciones[fila, col0_Y + cluster] = 1
  }

  dir<-c(dir,'=')
  b<-c(b,num_clusters_a_crear)
}


for (ciudad in 1:num_ciudades) {
  for(caracteristica in 1:num_caracteristicas){

    fila <- fila+1
    for (cluster in 1:num_clusters_iniciales) {
      restricciones[fila, (ciudad - 1) * num_ciudades * num_caracteristicas + cluster
                    + (caracteristica - 1) * num_ciudades] = 1
    }

    restricciones[fila, col0_Z + caracteristica] = -1

    dir<-c(dir,'<=')
    b<-c(b,0)
  }
}

for(contador in 1:1){
  fila <- fila + 1
```

```
  for(caracteristica in 1:num_caracteristicas){
    restricciones[fila, col0_Z + caracteristica] = 1
  }

  dir<-c(dir,'=')
  b<-c(b,num_caracteristicas_a_seleccionar)

}
```

```
#Así quedarían la matriz de restricciones y los vectores de direcciones y rhs

#restricciones
dir
```

```
##   [1] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [16] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [31] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [46] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [61] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [76] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
##  [91] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
## [106] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
## [121] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
## [136] "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="  "="
## [151] "="  "="  "="  "="  "="  "="  "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [166] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [181] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [196] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [211] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [226] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [241] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [256] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [271] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [286] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [301] "="  "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [316] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [331] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [346] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<="
## [361] "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "<=" "="
```

```
b
```

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [112] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [260] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [297] 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [334] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [371] 0 0 0 4
```

```r
#En este caso, solo vamos a mostrar la solución con el código que hemos elavorado para poder
#interpretarla de una forma sencilla:


#Cluster_asignado <- solopt\$solution[(ijk+1):(ijk+i_2)]

Cluster_asignado <- c(0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
objetos<-rownames(datos)
Objetos <- rep(objetos, times = rep(cluster, length(objetos)))
Cluster <- seq_along(rownames(datos))
Asignacion <- cbind(Objetos, Cluster, Cluster_asignado )
Asignacion
```

```
##        Objetos  Cluster Cluster_asignado
##   [1,] "user1"  "1"     "0"
##   [2,] "user1"  "2"     "0"
##   [3,] "user1"  "3"     "0"
##   [4,] "user1"  "4"     "0"
##   [5,] "user1"  "5"     "0"
##   [6,] "user1"  "6"     "0"
##   [7,] "user1"  "7"     "0"
##   [8,] "user1"  "8"     "0"
##   [9,] "user1"  "9"     "0"
##  [10,] "user1"  "10"    "0"
##  [11,] "user1"  "11"    "0"
##  [12,] "user1"  "12"    "1"
##  [13,] "user2"  "1"     "0"
##  [14,] "user2"  "2"     "0"
##  [15,] "user2"  "3"     "0"
##  [16,] "user2"  "4"     "0"
##  [17,] "user2"  "5"     "0"
##  [18,] "user2"  "6"     "0"
##  [19,] "user2"  "7"     "0"
##  [20,] "user2"  "8"     "0"
##  [21,] "user2"  "9"     "0"
##  [22,] "user2"  "10"    "0"
##  [23,] "user2"  "11"    "0"
##  [24,] "user2"  "12"    "1"
##  [25,] "user3"  "1"     "0"
##  [26,] "user3"  "2"     "0"
##  [27,] "user3"  "3"     "1"
##  [28,] "user3"  "4"     "0"
##  [29,] "user3"  "5"     "0"
##  [30,] "user3"  "6"     "0"
##  [31,] "user3"  "7"     "0"
##  [32,] "user3"  "8"     "0"
##  [33,] "user3"  "9"     "0"
##  [34,] "user3"  "10"    "0"
##  [35,] "user3"  "11"    "0"
##  [36,] "user3"  "12"    "0"
##  [37,] "user4"  "1"     "0"
##  [38,] "user4"  "2"     "0"
##  [39,] "user4"  "3"     "0"
##  [40,] "user4"  "4"     "0"
##  [41,] "user4"  "5"     "0"
##  [42,] "user4"  "6"     "0"
```

```
##  [43,] "user4"  "7"     "0"
##  [44,] "user4"  "8"     "0"
##  [45,] "user4"  "9"     "0"
##  [46,] "user4"  "10"    "0"
##  [47,] "user4"  "11"    "0"
##  [48,] "user4"  "12"    "1"
##  [49,] "user5"  "1"     "0"
##  [50,] "user5"  "2"     "0"
##  [51,] "user5"  "3"     "0"
##  [52,] "user5"  "4"     "0"
##  [53,] "user5"  "5"     "0"
##  [54,] "user5"  "6"     "0"
##  [55,] "user5"  "7"     "0"
##  [56,] "user5"  "8"     "0"
##  [57,] "user5"  "9"     "0"
##  [58,] "user5"  "10"    "0"
##  [59,] "user5"  "11"    "0"
##  [60,] "user5"  "12"    "1"
##  [61,] "user6"  "1"     "0"
##  [62,] "user6"  "2"     "0"
##  [63,] "user6"  "3"     "0"
##  [64,] "user6"  "4"     "0"
##  [65,] "user6"  "5"     "0"
##  [66,] "user6"  "6"     "0"
##  [67,] "user6"  "7"     "0"
##  [68,] "user6"  "8"     "0"
##  [69,] "user6"  "9"     "0"
##  [70,] "user6"  "10"    "0"
##  [71,] "user6"  "11"    "0"
##  [72,] "user6"  "12"    "1"
##  [73,] "user7"  "1"     "0"
##  [74,] "user7"  "2"     "0"
##  [75,] "user7"  "3"     "1"
##  [76,] "user7"  "4"     "0"
##  [77,] "user7"  "5"     "0"
##  [78,] "user7"  "6"     "0"
##  [79,] "user7"  "7"     "0"
##  [80,] "user7"  "8"     "0"
##  [81,] "user7"  "9"     "0"
##  [82,] "user7"  "10"    "0"
##  [83,] "user7"  "11"    "0"
##  [84,] "user7"  "12"    "0"
##  [85,] "user8"  "1"     "0"
##  [86,] "user8"  "2"     "0"
##  [87,] "user8"  "3"     "0"
##  [88,] "user8"  "4"     "0"
##  [89,] "user8"  "5"     "0"
##  [90,] "user8"  "6"     "0"
##  [91,] "user8"  "7"     "0"
##  [92,] "user8"  "8"     "0"
##  [93,] "user8"  "9"     "0"
##  [94,] "user8"  "10"    "0"
##  [95,] "user8"  "11"    "0"
##  [96,] "user8"  "12"    "1"
##  [97,] "user9"  "1"     "0"
##  [98,] "user9"  "2"     "0"
```

```
##  [99,] "user9"  "3"     "0"
## [100,] "user9"  "4"     "0"
## [101,] "user9"  "5"     "0"
## [102,] "user9"  "6"     "0"
## [103,] "user9"  "7"     "0"
## [104,] "user9"  "8"     "0"
## [105,] "user9"  "9"     "0"
## [106,] "user9"  "10"    "0"
## [107,] "user9"  "11"    "0"
## [108,] "user9"  "12"    "1"
## [109,] "user10" "1"     "0"
## [110,] "user10" "2"     "0"
## [111,] "user10" "3"     "0"
## [112,] "user10" "4"     "0"
## [113,] "user10" "5"     "0"
## [114,] "user10" "6"     "0"
## [115,] "user10" "7"     "0"
## [116,] "user10" "8"     "0"
## [117,] "user10" "9"     "0"
## [118,] "user10" "10"    "0"
## [119,] "user10" "11"    "0"
## [120,] "user10" "12"    "1"
## [121,] "user11" "1"     "0"
## [122,] "user11" "2"     "0"
## [123,] "user11" "3"     "0"
## [124,] "user11" "4"     "0"
## [125,] "user11" "5"     "0"
## [126,] "user11" "6"     "0"
## [127,] "user11" "7"     "0"
## [128,] "user11" "8"     "0"
## [129,] "user11" "9"     "0"
## [130,] "user11" "10"    "0"
## [131,] "user11" "11"    "0"
## [132,] "user11" "12"    "1"
## [133,] "user12" "1"     "0"
## [134,] "user12" "2"     "0"
## [135,] "user12" "3"     "0"
## [136,] "user12" "4"     "0"
## [137,] "user12" "5"     "0"
## [138,] "user12" "6"     "0"
## [139,] "user12" "7"     "0"
## [140,] "user12" "8"     "0"
## [141,] "user12" "9"     "0"
## [142,] "user12" "10"    "0"
## [143,] "user12" "11"    "0"
## [144,] "user12" "12"    "1"
```

```
#caracteristicas_elegida <- solopt\$solution[(ijk+i_2+1):(ijk+i_2+num_caracteristicas)]
caracteristicas_elegida<-c(1,1,0,1,1,0)
Caracteristica<-colnames(datos)
Caracteristicas_elegidas <- cbind(Caracteristica, caracteristicas_elegida)
Caracteristicas_elegidas

##      Caracteristica        caracteristicas_elegida
## [1,] "edad"                "1"
```

```
## [2,] "paginas_vistas"      "1"
## [3,] "sesiones"            "0"
## [4,] "avg_duracion_sesion" "1"
## [5,] "rebotes"             "1"
## [6,] "avg_scroll"          "0"
```

```
#cluster_elegidos <- solopt\$solution[(ijk+i_2+num_caracteristicas+1):(length(vector_solucion))]
cluster_elegidos<-c(0,0,1,0,0,0,0,0,0,0,0,1)
Cluster <- seq_along(rownames(datos))
Cluster_elegidos <- cbind(Cluster, cluster_elegidos)
Cluster_elegidos

##       Cluster cluster_elegidos
##  [1,]       1                0
##  [2,]       2                0
##  [3,]       3                1
##  [4,]       4                0
##  [5,]       5                0
##  [6,]       6                0
##  [7,]       7                0
##  [8,]       8                0
##  [9,]       9                0
## [10,]      10                0
## [11,]      11                0
## [12,]      12                1
```

Por último, para concluir con nuestro ejemplo de uso real, vamos a incluir a dos nuevos usuarios en la base de datos para incluirlos en los clusters creados midiendo la distancia con el centro del cluster.

Usuarios

ID Usuario	Periódico	Dispositivo	Sexo	Tipo usuario	Edad	Páginas Vistas	Sesiones	Duración sesión	Rebotes	Scroll
user13	INF	mobile	female	registrado	50	610	80	1750	15	25
user14	LNE	desktop	male	anonimo	60	174	53	500	20	100

Centroide de los cluster

Cluster	Periódico	Dispositivo	Sexo	Tipo usuario	Edad	Páginas Vistas	Sesiones	Duración sesión	Rebotes	Scroll
Cluster1	EP	desktop	male	anonimo	60	589	69	2974	16	75
Cluster2	VIG	mobile	male	anonimo	65	199	49	283	19	75

*Esta información la conocemos ya que según nuestro módelo el centroide de cada cluster se corresponde con uno de nuestros usuarios, en este caso los usuarios con id3 e id12

- Distancia entre el **user13** y el **cluster1**

$$(60-50)^2+(589-610)^2+(2{,}974-1{,}750)^2+(16-15)^2=1{,}498{,}718$$

- Distancia entre el **user13** y el **cluster2**

$$(65-50)^2+(199-610)^2+(283-1{,}750)^2+(19-15)^2=2{,}321{,}251$$

- Distancia entre el **user14** y el **cluster1**

$$(60-60)^2+(589-174)^2+(2974-500)^2+(16-20)^2=6{,}292{,}917$$

- Distancia entre el **user14** y el **cluster2**

$$(65-65)^2 + (199-174)^2 + (283-500)^2 + (19-20)^2 = 47{,}715$$

Con estas distancias entre los nuevos usuarios y los centros de los cluster, vemos que el **user13** se incluiría en el **cluster1** y el **user14** en el **cluster2**

Referencias

[1] https://rstudio-pubs-static.s3.amazonaws.com/399475_4eed578cf0154c23b54fe5b25e70e4d8.html

[2] https://support.google.com/analytic