TRABAJO FINAL

Especialización en Inteligencia de Datos Orientada a Big Data

ESTUDIO DE TÉCNICAS DE AGRUPAMIENTO EN PROCESOS DE DATOS A GRAN ESCALA

Su aplicación en la descripción de Casos de COVID-19

registrados en la República Argentina.

Autor: Patricia Elizabeth Prado
Director: Dr. Waldo Hasperué
Fecha: 30/08/2022

Contenido

1.Introducción.

En dominios donde el volumen de los datos crece exponencialmente y la enorme abundancia de estos desborda la capacidad humana para comprenderlos, existe una necesidad apremiante de desarrollar soluciones para aprovechar esta riqueza de datos utilizando métodos estadísticos.

El agrupamiento es la tarea descriptiva por excelencia, consiste en obtener grupos naturales a partir de los datos para luego poder describirlos de manera concisa. Ya sea para la comprensión o el resumen, el análisis de agrupamiento ha desempeñado durante mucho tiempo un papel importante en una amplia variedad de campos como psicología, ciencias sociales, biología, estadísticas, reconocimiento de patrones y recuperación de información. [1]

En la actualidad, en el marco de la pandemia del COVID-19, se dispone de información sobre millones de casos de contagios registrados en Argentina con sus características personales, geográficas y temporales, la cual es actualizada semanalmente. Resulta de interés segmentar los casos en grupos tales que las muestras de un grupo permitan extraer descripciones que resulten de utilidad para el sistema de salud, para este fin las técnicas de agrupamiento son una valiosa herramienta.

1.1. Motivación

En muchas situaciones, el método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizado en forma manual, mediante consultas efectuadas con lenguajes generalistas de consulta como SQL sobre la base de datos operacional o ejecutando operaciones de procesamiento analítico en línea [On-Line Analytical Processing, OLAP) sobre almacenes de datos. Estas soluciones admiten técnicas de análisis como pueden ser el resumen, la consolidación, la agregación y la posibilidad de ver la información desde distintas perspectivas. Sin embargo, no generan reglas, patrones, pautas, es decir conocimiento que pueda ser aplicado a otros datos [2].

Para el procesamiento de datos a gran escala se han desarrollado varios motores de procesamiento distribuido de datos como por ejemplo MapReduce [3] y librerías con funcionalidades de aprendizaje automático en estos motores. En particular, Apache Spark [4] es un sistema de computación en clúster tolerante a fallos y de propósito general que proporciona una interfaz de programación de aplicaciones en los lenguajes Java, Scala, Python y R, junto con un motor optimizado que admite grafos de ejecución general. Es eficiente en cálculos iterativos y, por lo tanto, es adecuado para el desarrollo de aplicaciones de aprendizaje automático a gran escala [5]

La biblioteca de aprendizaje automático distribuida de Spark, MLlib, proporciona implementaciones rápidas y escalables de algoritmos de aprendizaje estándar para entornos de aprendizaje comunes, incluida la clasificación, la regresión, el filtrado colaborativo, el agrupamiento y la reducción de la dimensionalidad. Escrito en Scala y utilizando bibliotecas de álgebra lineal nativas (basadas en C++) en cada nodo, MLlib incluye una interfaz de programación para aplicaciones en Java, Scala y Python, y se publica como parte del proyecto Spark bajo la licencia Apache 2.0. [5]

El presente estudio pretende analizar diferentes técnicas de agrupamiento en el entorno distribuido Spark, que permitan describir de forma novedosa el seguimiento de casos de COVID-19 en Argentina a partir de la aplicación de modelos de agrupamiento adecuados para el desarrollo de aplicaciones de aprendizaje automático a gran escala.

1.2. Objetivos

El objetivo de este trabajo es estudiar tres técnicas de agrupamiento en entornos que se benefician del paralelismo de datos, en particular, en el entorno distribuido Spark. Los algoritmos estudiados son:

- K-Means [6], uno de los algoritmos de clasificación no supervisada más utilizados, rápido y sencillo, agrupa observaciones en un número predefinido de grupos basándose en sus características.
- Bisecting K-Means [6], un algoritmo de agrupamiento jerárquico que utiliza un enfoque divisivo o "de arriba hacia abajo". Todas las observaciones comienzan en un grupo y las divisiones se realizan de forma recursiva a medida que uno desciende en la jerarquía.
- Gaussian mixture model [6], un modelo de mezcla gaussiana que representa una distribución compuesta mediante la cual las observaciones se extraen de una de las k sub-distribuciones gaussianas, cada una con su propia probabilidad. Este algoritmo utiliza el método de maximización de expectativas para inducir el modelo de máxima verosimilitud dado un conjunto de muestras.

Los experimentos de este trabajo se realizaron sobre el conjunto de datos “COVID-19. Casos registrados en la República Argentina”, el cual fue generado, guardado y publicado por la Dirección Nacional de Epidemiología y Análisis de Situación de Salud y descargado el 10 de agosto de 2021. Los ensayos realizados consistieron en la ejecución de las técnicas de agrupamiento con diferentes configuraciones de sus hiperparámetros, utilizando diferentes tamaños del conjunto de datos para evaluar escalabilidad y distintos grupos de características para determinar cómo influye la dimensionalidad. Se realizaron experimentos donde se ejecutaron los algoritmos en una única computadora, como así también en un clúster de tres nodos. En estos ensayos se midieron y compararon tiempos de ejecución.

Por otro lado, los grupos fueron evaluados con métodos de validación interna para determinar la pureza de estos, al mismo tiempo que se utilizaron técnicas de visualización y acumulación de evidencia para la comparación de los diferentes grupos de datos obtenidos por los diferentes modelos.

Cabe señalar que se utilizaron las implementaciones de los algoritmos optimizados para trabajar de manera paralela y eficiente aprovechando los nodos disponibles en el clúster. Estas implementaciones tienen un orden de complejidad lineal o logarítmico respecto al tamaño del conjunto de datos, permitiendo que escalen bien en grandes conjuntos de datos.

1.3. Trabajos relacionados

La base de datos PubMed incluye revistas de ciencias de la salud, entre ellas las de MEDLINE, pero también otros materiales de revistas científicas y de libros on-line. Se realizó una búsqueda con el tópico 'covid-19 and clustering', arrojando un total de 1253 documentos, publicados en 15 fuentes (diarios, libros, entre otras), durante el período 2020/2021.

De esa búsqueda se encontraron varios trabajos que utilizaron técnicas de agrupamiento para realizar diferentes estudios relacionados a las ciencias médicas, demostrando ser una herramienta muy útil en esta área. Es por ello por lo que resultó de interés estudiar las técnicas de agrupamiento en entornos que poseen grandes cantidades de datos y demanden mucho poder de cómputo.

Se seleccionaron las publicaciones que analizan conjuntos de datos con formatos similares al disponible para analizar en este trabajo. Algunos autores propusieron un enfoque de agrupamiento simple e inmediato para categorizar las regiones italianas que trabajaban en la prevalencia y tendencia de los casos positivos de SARS-CoV-2 en un período de tiempo determinado. Aplicando agrupamiento jerárquico y K-Means, identificaron tres grupos regionales ofreciendo una instantánea de la epidemia, que podría ser útil para delinear la jerarquía de necesidades a nivel subnacional [7].

En otras investigaciones se aplicaron el coeficiente de determinación y correlación de Pearson y el algoritmo K-Means. En la primera de ellas se tabularon los datos de regiones y distritos peruanos para observar la diferencia entre urbano y rural en infectados y fallecidos. El modelo generado por el algoritmo K-Means mostraba el comportamiento de la letalidad con picos bastante elevados en el intervalo mayo-agosto de 2020 [8].

En la segunda investigación se realizó un estudio transversal en 31 provincias de Irán utilizando el número diario de casos confirmados. Se calculó la Frecuencia Acumulada (CF, por sus siglas en inglés) y la CF Ajustada (ACF) de nuevos casos para cada provincia. Las características de las provincias como la densidad de población, el área, la distancia del epicentro original (provincia de Qom), la altitud del nivel del mar y el Índice de Desarrollo Humano (IDH) se utilizaron para investigar su correlación con los valores de ACF. Los resultados del análisis de agrupamiento de K-Means basados en ACF identificaron las provincias que tenían condiciones críticas y necesitaban atención [9].

Estos son solo algunos trabajos que muestran la utilidad de las técnicas de clustering en problemas médicos y de salud. Existen decenas de trabajos que no son mencionados en este trabajo ya que la descripción de los mismos no es relevante en esta investigación, solo resulta de interés mencionar al clustering como herramienta cotidiana en las ciencias de la salud, y más relacionado al estudio de una pandemia.

El resto del trabajo se organiza de la siguiente manera, en el capítulo 2 se detallan los tres algoritmos estudiados. En el capítulo 3 se detalla la base de datos y las herramientas utilizadas para su análisis. En el capítulo 4 se describen el pretratamiento realizado a la base de datos. En el capítulo 5 se menciona los experimentos llevados a cabo y los resultados obtenidos y finalmente en el capítulo 6 se describen las conclusiones del trabajo.

2. Conceptos preliminares

En este capítulo se describen las tres técnicas de clustering utilizadas en el presente trabajo como así también diferentes métricas utilizadas para la validación de los resultados obtenidos por los algoritmos de clustering. Estas técnicas de validación resultan útiles para llevar a cabo la comparación de los diferentes resultados obtenidos mostrando diferencias y similitudes de las técnicas de clustering estudiadas.

2.1. K-Means

K-Means [1] es uno de los algoritmos más utilizados para agrupar datos en un número predefinido de agrupaciones. Es rápido, simple y paralelizable. Es necesario especificar de antemano cuántos grupos se buscarán definiendo el parámetro k. El algoritmo asignará cada dato al centroide más cercano generando k agrupaciones, luego el centroide de cada agrupación se actualizará en función de los datos asignados al grupo iterando hasta que se asignen los mismos datos a cada grupo en rondas consecutivas. Esta búsqueda local, llamada iteración de Lloyd se repite hasta que los centroides se han estabilizado y permanecen iguales para siempre. En este momento cada muestra del conjunto de datos queda asignado exclusivamente a un único grupo.

Para asignar un dato al centroide más cercano, se necesita una medida de proximidad que cuantifica la noción de "más cercano" para los datos específicos considerados. La distancia euclidiana se utiliza a menudo para muestras de datos en el espacio euclídeo, mientras que la similitud del coseno es más apropiada para los documentos, ambas medidas pueden configurarse en la librería MLlib de Spark. [10].

Por lo general, las medidas de similitud utilizadas para K-Means son relativamente simples, ya que el algoritmo calcula repetidamente la similitud de cada muestra con cada uno de los centroides.

2.1.1. Selección de centroides

La selección de centroides iniciales aleatoria que se realiza en el algoritmo K-Means estándar puede acarrear algunos problemas, como encontrar un mínimo local. Por este motivo es aconsejable utilizar otras alternativas para la selección inicial de centroide como K-Means++ o su versión paralelizable K-Means|| [10].

K-Means++, selecciona solo el primer centroide uniformemente al azar. Cada centroide posterior se selecciona con una probabilidad proporcional a su contribución al error general dadas las selecciones anteriores. El algoritmo de inicialización depende críticamente de la selección de los i−1 centros anteriores ya que son las elecciones anteriores las que determinan qué puntos están ausentes en la selección.

Este hecho se ve exacerbado en el escenario de datos masivos. En primer lugar, a medida que el conjunto de datos crece, también lo hace el número de grupos en los que se desea particionar los datos.

La versión paralelizable K-Means|| [10] en lugar de seleccionar un solo dato en cada pasada como el algoritmo K-Means++, selecciona O (k) datos en cada iteración y repite el proceso para aproximadamente O (log n) iteraciones. Al final del algoritmo se queda

con O (k log n) datos que forman una solución que está dentro de un factor constante alejado del óptimo. Luego agrupa estos O (k log n) datos en k centros iniciales asignando repetidamente cada punto a su centro más cercano y calculando los centros dada la asignación de muestras. Este algoritmo de inicialización es bastante simple y se presta a implementaciones paralelas fáciles.

2.1.2. El algoritmo

Sea X = $\{x_1, \ldots, x_n\}$ un conjunto de puntos en el espacio Euclídeo n-dimensional y sea k un entero positivo que especifique el número de agrupaciones. $\| x_i - \mathbf{x}_j \|$ denota la distancia euclidiana entre x_i y x_j. Para un punto x y un subconjunto Y ⊆ X de puntos, la distancia es definida como d (x, Y) = min y ∈ Y ||x − y||. Para un subconjunto Y ⊆ X de puntos, sea su centroide dado por

$$\text{centroide (Y)} = \frac{1}{|Y|}\sum_{y \in Y} y$$

Se define C= $\{c_1, c_2, \ldots, c_k\}$ como un conjunto de puntos Y ⊆ X. El costo de Y con respecto a C se define como

$$\phi_Y(C) = \sum_{y \in Y} d^2(y, C) = \sum_{y \in Y} \min_{i=1,\ldots,k} \|y - c_i\|^2$$

El objetivo de la agrupación de K-Means es elegir un conjunto C de k centros para minimizar $\phi_X(C)$. Definimos $\phi_X(C)$ como el costo de la agrupación óptima de K-Means; encontrar $\phi_X(C)$ tiene complejidad NP. Un conjunto C de centros es una aproximación α a K-Means si $\phi_X(C) \leq \alpha\phi_X(C)$. Teniendo en cuenta que los centros definen automáticamente una agrupación de X de la siguiente manera: el clúster i^{th} es el conjunto de todos los puntos en X que están más cerca de c_i que cualquier otro c_j , j ≠ i.

Arthur y Vassilvitskii modificaron el paso de inicialización de manera cuidadosa y obtuvieron un algoritmo de inicialización aleatorio llamado K-Means++. La idea principal en su algoritmo es elegir los centros uno por uno de manera controlada, donde el conjunto actual de centros elegidos sesgará estocásticamente la elección del siguiente centro. El inconveniente central de la inicialización K-Means++ desde el punto de vista de la escalabilidad es su naturaleza secuencial inherente: la elección del siguiente centro depende del conjunto actual de centros.

Algoritmo 1

K-Means++
1 C ← seleccionamos un centroide aleatoriamente de X
2 para cada punto del conjunto de datos |C| < k
3 seleccionamos x ∈ X con probabilidad $\frac{d^2(x,C)}{\phi_X(C)}$
4 C ← C ∪ {x} agregamos un nuevo centroide
5 fin mientras

Las ventajas demostrables de K-Means++ sobre la inicialización aleatoria se encuentran precisamente en la selección no uniforme constantemente actualizada. El algoritmo K-Means|| busca lo mejor de ambos mundos: un algoritmo que funcione en un

pequeño número de iteraciones, seleccione más de un punto en cada iteración, pero de manera no uniforme y tenga garantías de aproximación demostrables.

Si bien el algoritmo K-Means|| se inspira en gran medida en K-Means++, utiliza un factor de sobremuestreo $\ell = \Omega(k)$, que es diferente a K-Means++; intuitivamente, ℓ debe ser considerado como $\Theta(k)$. Spark por defecto utiliza un valor ℓ de 2k. El algoritmo K-Means|| elige un centro inicial (por ejemplo, uniformemente al azar) y calcula ψ, el costo inicial de la agrupación en clústeres después de esta selección. Luego procede en log ψ iteraciones, donde en cada iteración, dado el conjunto actual C de centros, selecciona cada x con probabilidad $\ell d^2(x, C) / \phi_X(C)$. Los puntos seleccionados se agregan a C, la cantidad $\phi_X(C)$ se actualiza y la iteración continúa. Como en el algoritmo 2, el número esperado de puntos elegidos en cada iteración es ℓ y al final, el número esperado de puntos en C es $\ell \log \psi$, que suele ser mayor que k. Para reducir el número de centros, el paso 7 asigna pesos a los puntos en C y el paso 8 agrupa estos puntos ponderados para obtener k centros. Como el tamaño de C es significativamente menor que el tamaño de entrada, el reagrupamiento para obtener k centros se puede hacer rápidamente con K-Means++ asignando todos los centroides a una sola máquina.

Algoritmo 2

K-Means|| (k, ℓ) inicialización
1: C ← seleccionamos un centroide al azar desde X
2: $\psi \leftarrow \phi_X(C)$ calculamos el costo inicial
3: Por cada O (log ψ) iteraciones
4: C' ← seleccionamos cada punto x ∈ X en forma independiente
con probabilidad $p_x = \frac{\ell d^2(x,C)}{\phi_X(C)}$
5: C ← C ∪ C' añadimos los nuevos centroides
6: fin
7: por cada x ∈ C, sea w_x el número de puntos en X más cercano a x que cualquier otro punto en C
8: Reagrupar los puntos ponderados en k agrupaciones

La selección de nuevos centroides según la implementación de la librería MLlib de Spark se realiza verificando que rand.nextDouble < 2.0 * c * k / sumCost, donde rand.nextDouble es un número aleatorio decimal entre 0.0 y 1.0, c es la distancia euclídea de dicho punto a su centroide más cercano, k es el número de grupos y sumCost es el costo total de clusterización definido como $\phi_X(C)$ [11] [12].

El algoritmo es muy simple y se presta a una implementación paralela natural (en log ψ iteraciones), si se utiliza la inicialización K-Means++ en el paso 8, K-Means|| tendrá una aproximación O (log k).

2.1.3. Problemas adicionales en la agrupación K-Means

2.1.3.1. Formas naturales de los datos

El algoritmo K-Means no funciona correctamente cuando los grupos a encontrar no son grupos convexos, es decir grupos en los cuales se puede unir cada par de puntos con una línea recta, y dicha línea sigue dentro de los límites del grupo.

El algoritmo K-Means también tiene problemas para llevar a cabo su tarea cuando existen agrupaciones de distinto tamaño o densidad ya que, en un intento de minimizar la suma de cuadrados dentro de cada grupo, K- Means da mayor prioridad a los grupos grandes sin considerar que unos pocos ejemplos queden muy alejados de sus centroides.

2.1.3.2. Agrupaciones vacías

Uno de los problemas con el algoritmo básico de K-Means es que se pueden obtener agrupaciones vacías si no se asignan puntos a una agrupación durante el paso de asignación. Si esto sucede, se puede elegir el punto más alejado de cualquier centroide actual o elegir el centroide de reemplazo del grupo que tiene la suma del error al cuadrado (SSE) más alta dividiendo el clúster. Ambos enfoques reducirán la SSE general del clúster. [1]

2.1.3.3. Valores atípicos

Cuando se utiliza el criterio de error al cuadrado, los valores atípicos pueden influir indebidamente en los grupos que se encuentran. En particular, cuando hay valores atípicos presentes, los centroides de grupo resultantes pueden no ser tan representativos como lo serían de otro modo y, por lo tanto, la suma del error cuadrático (SSE) también será más alta. Si se utilizan enfoques que eliminen los valores atípicos antes de la agrupación, se evitarán resultados indeseados. Alternativamente, los valores atípicos también se pueden identificar en un paso posterior al procesamiento. Por ejemplo, se puede hacer un seguimiento de la suma del error cuadrático aportada por cada punto y eliminar esos puntos con contribuciones inusualmente altas, especialmente en ejecuciones múltiples. Además, es posible eliminar grupos pequeños, ya que con frecuencia representan grupos de valores atípicos. [1]

2.1.3.4. Reducción de la SSE con post procesamiento

Una forma obvia de reducir la suma del error cuadrático es encontrar más grupos, es decir, utilizar un k más grande. Sin embargo, en muchos casos, se busca mejorar la suma del error cuadrático, pero sin aumentar el número de clústeres.

Es posible cambiar la suma del error cuadrático total realizando varias operaciones en los clústeres, como dividirlos o fusionarlos. Un enfoque comúnmente utilizado es utilizar fases alternativas de división y fusión de clústeres. Durante una fase de división, los grupos se dividen, mientras que, durante una fase de fusión, los grupos se combinan.

De esta manera, a menudo es posible escapar de los mínimos de la suma del error cuadrático locales y aun así producir una solución de agrupación con el número deseado de agrupaciones.[1]

2.1.4. Paralelismo de datos y resultados

Spark utiliza el concepto de paralelismo de datos o paralelismo de resultados al realizar la agrupación de K-Means. En este trabajo se intentan encontrar agrupaciones para describir casos de Covid, estudiando 16.486.507 observaciones con los atributos de información del caso. Si se busca ejecutar múltiples iteraciones de K-Means en un sistema local, por ejemplo, para k = 5, la cantidad de métricas de distancia que se deben calcular es 5 x 16.486.507 = 82.432.535 métricas. Es necesario calcular 82.432.535 de estas métricas, por ejemplo, 20 veces antes de que se cumpla un criterio de convergencia, es decir, 1.648.650.700 de distancias euclidianas, demandando mucho poder de cálculo y tiempo de procesamiento [13].

El paralelismo de datos crea paralelismo desde el principio al dividir el conjunto de datos en particiones más pequeñas. Por ejemplo, si se considera:

D = Número de registros $\{X_1, X_2, \ldots\ldots, Xn\}$
k = Número de agrupaciones
P = Número de procesadores $\{P_1, P_2, \ldots, P_m\}$
C = Centroides iniciales $\{C_1, C_2, \ldots, C_k\}$

1. Los datos D se dividen en procesadores P. Cada procesador trabaja en un conjunto de registros determinado por la configuración Spark. Los valores del centroide inicial, C, se comparten entre cada uno de estos procesadores.
2. Cada procesador tiene información de centroide. Los procesadores calculan la distancia de sus registros a estos centroides y forman grupos locales asignando puntos de datos a su centroide más cercano.
3. Una vez que se realiza el paso 2, un proceso maestro almacena la suma y el recuento de registros para cada uno de estos grupos en los procesadores P para referencia futura.
4. Una vez que se completa una iteración, se intercambia la información del procesador y un proceso maestro calcula los centroides actualizados y los comparte entre los procesadores P nuevamente, es decir, se asigna un punto a k agrupaciones, un proceso maestro actualiza los centroides y vuelve a compartir la información nuevamente con los procesadores.
5. Este proceso continúa iterando hasta que se alcanza la convergencia. Una vez que se cumple un criterio de convergencia, el proceso maestro recopila grupos locales y los combina en uno global.

Si se dividen los 16.486.507 casos por ejemplo entre 12 procesadores, cada uno de los cuales tendrá 16.486.507/12 aproximadamente 1.373.876 registros. El procesamiento distribuido entra en escena para reducir el volumen de datos, pero asegurando un resultado completo. [13]

El paralelismo de resultados se basa en clústeres específicos. En este caso

1. Los datos D se dividen en procesadores P y luego se clasifican dentro de cada procesador. Cada procesador trabaja en un conjunto de registros determinado por la configuración de Spark.
2. Los valores de centroide iniciales, C, se inicializan, dividen y comparten en cada uno de estos procesadores, es decir, a diferencia del paralelismo de datos donde todos los valores de centroide se comparten en todos los procesadores, ahora se pasa un valor de centroide a un procesador.
3. Cada procesador tiene un centroide de información. Se calcula la distancia de los puntos a los centroides. Para puntos de datos en un procesador que son extremadamente bajos o altos: si están más cerca del centroide del procesador, se asignan a ese grupo de lo contrario, si están más cerca del centroide que pertenece a un procesador diferente, se moverán al nuevo procesador
4. El proceso se repite hasta alcanzar la convergencia devolviendo todos los clústeres locales del procesador P. [13]

2.2. Bisecting K-Means

El algoritmo Bisecting K-Means es un algoritmo de agrupamiento jerárquico divisivo, de arriba hacia abajo. Es una extensión directa del algoritmo básico de K-Means que puede ejecutarse en paralelo sin problemas, ya que tras decidir qué grupo será el siguiente en dividirse en dos, los dos nuevos grupos pueden dividirse utilizando un algoritmo de agrupamiento escalable como K-Means|| [11]

El algoritmo Bisecting K-Means, requiere la configuración inicial del parámetro "k" que indica el número de grupos que se desea obtener y el tamaño mínimo de un grupo para permitirle que pueda seguir dividiéndose en nuevos grupos.

Algoritmo 3

Bisecting K-Means|| (k, ℓ) inicialización
1: Comienza con el conjunto de datos completo formando un solo grupo
Repetir
2: Si supera el mínimo establecido por ℓ divide el grupo en 2 nuevos grupos
3: Selecciona un grupo de acuerdo con algún tipo de medida de evaluación (el grupo más grande o el de mayor distancia intragrupo)
Hasta encontrar k grupos

Bisecting K-Means es menos susceptible a los problemas de inicialización, porque realiza varias "bisecciones" de prueba y toma la que tiene la más baja suma de errores al cuadrado (SSE), y porque solo hay dos centroides en cada paso.

El algoritmo Bisecting K-Means suele producir resultados algo mejores que el algoritmo K-Means estándar, al menos en algunos campos como el de la agrupación de documentos [11]. Sin embargo, este algoritmo tiene un orden de complejidad de $O(n)$, superior al orden de complejidad del algoritmo K-Means que sería $O(\log n)$, lo que hace que la ejecución del algoritmo K-Means sea más rápida que la del algoritmo divisivo

Bisecting K-Means, y además escale peor según se agreguen ejemplos al conjunto de datos.

2.3. Mezclas Gaussianas

Una distribución gaussiana es una distribución de probabilidad, y se define por una media y una desviación estándar.

En el modelo de Mezclas Gaussianas, existen tantas distribuciones gaussianas como grupos, es decir, "k" distribuciones gaussianas, y además un peso o una probabilidad a priori de pertenencia a cada grupo, que vendrá dada por la proporción de elementos que han sido asignados a cada gaussiana.

K-Means se puede considerar un caso particular de Mezclas Gaussianas donde todas las variables tienen la misma covarianza. En K-Means, cada grupo quedaba definido por un centroide, que básicamente era la media entre todos los ejemplos que pertenecían al grupo. En el modelo de Mezclas Gaussianas, cada grupo quedará definido por una gaussiana con una media y una matriz de covarianzas y también por un peso. La suma de todos los pesos deberá ser igual a 1 y representa las probabilidades a priori de que un ejemplo del conjunto de datos pertenezca a un determinado grupo. Con este modelo se obtiene una mayor flexibilidad a la hora de definir la forma de los grupos, que ya no estará limitada a una forma esférica, y también a la hora de tratar con grupos de diferente tamaño.

El modelo de Mezclas Gaussianas es un algoritmo iterativo que basa su funcionamiento en el algoritmo de EM (Maximización de expectativas) para ir adaptándose poco a poco a los datos. La expectativa es cómo determinar la clase de una instancia desconocida. La maximización es como determinar las probabilidades de valor de atributo de las instancias de entrenamiento, con la pequeña diferencia de que en el algoritmo EM las instancias se asignan a clases probabilísticamente en lugar de categóricamente. El algoritmo EM, en primer lugar, estima la agrupación a la que pertenece cada instancia dados los parámetros de distribución y luego estima los parámetros de las instancias clasificadas.

El modelo parte de la premisa de que todas las variables o dimensiones de nuestro conjunto de datos siguen una distribución normal o gaussiana, lo cual no siempre se cumple.

El algoritmo de mezclas gaussianas es mucho más costoso computacionalmente que K-Means y el tiempo de ejecución dado un conjunto de datos de un tamaño específico es superior. Además, cuanto más se aumenta el tamaño de dicho conjunto de datos, el tiempo de ejecución del algoritmo de mezclas gaussianas se incrementa en la misma medida, ya que tiene un orden de complejidad lineal O (n), mientras que K-Means tiene un orden de complejidad logarítmico O (log n).

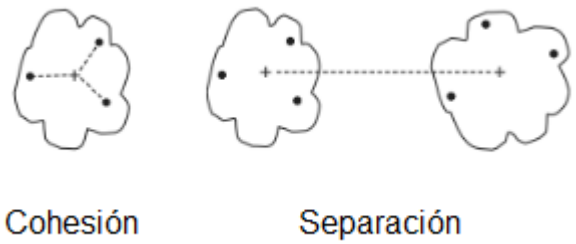
El algoritmo de mezclas gaussianas, al igual que sucedía con K-Means tiene problemas para tratar con grupos no convexos y no escala bien cuando se aumenta el número de dimensiones, por lo tanto, si se utilizara un conjunto de datos de gran tamaño con un elevado número de variables, es posible que usar mezclas gaussianas no sea la mejor opción. [11]

2.4. Métodos de validación interna

2.4.1. Índice Silhouette

El método de índice o coeficiente Silhouette [1] es una medida para la validación de la coherencia dentro de los agrupamientos que ayuda a determinar si los grupos son compactos y están bien espaciados combinando los criterios de cohesión y separación.

Gráfico 2. 1 Vista basada en prototipos de la cohesión y separación de grupos.



El criterio de cohesión indica que cada miembro de un grupo debe estar lo más cerca posible de los otros miembros del mismo grupo, mientras que el criterio de separación infiere que los grupos deben estar ampliamente separados entre sí.

El proceso para calcular el índice consta de los 3 pasos siguientes:

1. Para cada objeto i^{th}, calcular su distancia media a todos los demás objetos en su clúster y llamarlo a_i.
2. Para cada objeto i^{th} de cualquier clúster que no contenga el objeto a_i, calcular la distancia media del objeto a todos los objetos del clúster dado. Encontrar el valor mínimo de este tipo con respecto a todos los grupos y llamarlo b_i.
3. Para cada objeto i^{th} calcular el índice Silhouette como:

$$s_i = (b_i - a_i) / \max (a_i, b_i).$$

El valor del índice Silhouette puede variar entre −1 y 1. Un valor cercano a 1 significa que los puntos en un grupo están cerca de los otros puntos en el mismo grupo y lejos de los puntos de los otros grupos. Se espera que el índice Silhouette sea positivo (ai < bi), y que a_i esté lo más cerca posible de 0, ya que el coeficiente asume su valor máximo de 1 cuando $a_i = 0$.

Un valor 0 indica que la distancia entre los agrupamientos no es significativa. Un valor negativo no es deseable porque corresponde a un caso en el que a_i, la distancia media a los puntos del clúster es mayor que b_i, la mínima distancia media a puntos de otro clúster. Por último, un valor -1 indica que los agrupamientos están asignados de forma incorrecta.

Es posible calcular el índice Silhouette promedio de un clúster simplemente tomando el promedio de los índices de Silhouette de los puntos pertenecientes al clúster. Una medida global de la bondad de un agrupamiento se puede obtener calculando el índice Silhouette promedio de todos los puntos.

2.4.2. Medición de la validez del clúster a través de la correlación

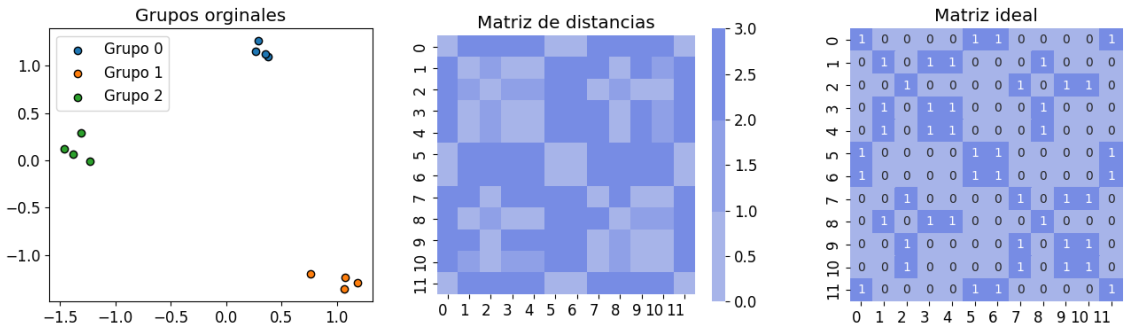
La matriz de similitud real se construye creando una matriz que tiene una fila y una columna para cada punto de datos y asignando la distancia entre estos a la entrada correspondiente al par de puntos. La matriz de similitud ideal se construye creando una matriz que tiene una fila y una columna para cada punto de datos al igual que una matriz de similitud real, asignando un 1 a una entrada si el par de puntos asociado pertenece al mismo clúster y 0 si el par pertenece a grupos diferentes.

Si se dispone de la matriz de similitud para un conjunto de datos y las etiquetas de clúster de un análisis de clúster del conjunto de datos, entonces es posible evaluar la "bondad" de la agrupación observando la correlación entre la matriz de similitud y una versión ideal de la matriz de similitud basada en las etiquetas de clúster. Más específicamente, un clúster ideal es uno cuyos puntos tienen una similitud de 1 con todos los puntos del clúster, y una similitud de 0 a todos los puntos de otros grupos. Así, si se ordenan las filas y columnas de la matriz de similitud para que todos los objetos que pertenecen a la misma clase queden juntos, entonces una matriz de similitud ideal tiene una estructura diagonal de bloques. En otras palabras, la similitud es distinta de cero, es decir, 1, dentro de los bloques de la matriz de similitud cuyas entradas representan la similitud en el mismo clúster, y 0 en otros lugares.

La alta correlación entre las matrices de similitud ideal y real indica que los puntos que pertenecen al mismo clúster están cerca uno del otro, mientras que la baja correlación indica lo contrario. Como las matrices de similitud real e ideal son simétricas, la correlación se calcula sólo entre n(n−1) /2 entradas por debajo o por encima de la diagonal de las matrices. En consecuencia, esto no es una buena medida para muchos grupos basados en la densidad o la contigüidad, porque no son globulares y pueden estar estrechamente entrelazados con otros agrupamientos.

El gráfico 2.2 muestra la matriz de similitud y la matriz de similitud ideal correspondiente a un ejemplo de 12 puntos agrupados en 3 clústeres encontrados por K-Means.

Gráfico 2. 2 Matriz de similitud de distancias y matriz de similitud ideal

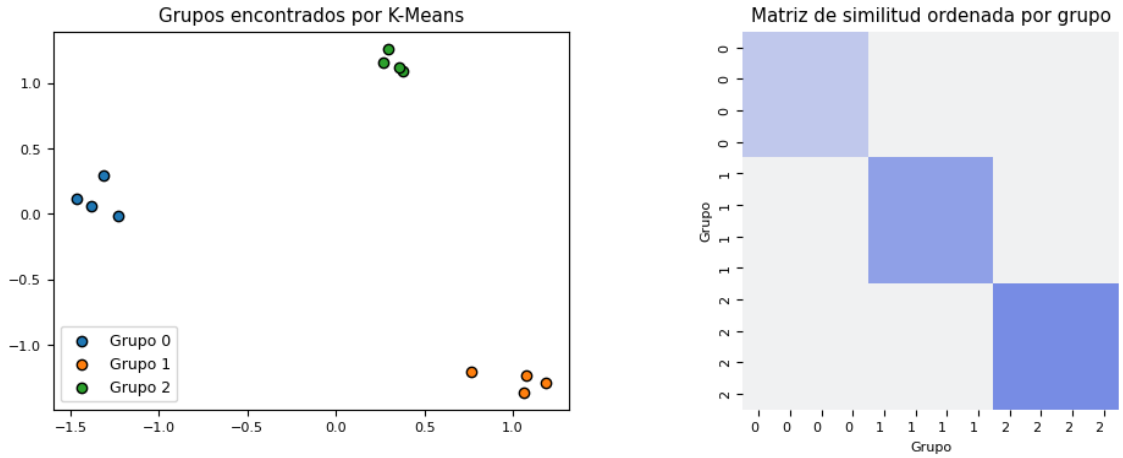


2.4.3. Visualización de agrupamientos por su matriz de similitud

La técnica anterior sugiere un enfoque más general y cualitativo para juzgar un conjunto de agrupamientos. Ordenar la matriz de similitud con respecto a las etiquetas de clúster y luego trazar la matriz. En teoría, si tenemos agrupamientos bien separados, entonces la matriz de similitud debería ser aproximadamente una diagonal de bloques. De

lo contrario, los patrones que se muestran en la matriz de similitud pueden revelar las relaciones entre los clústeres.

Gráfico 2. 3 Visualización de agrupamientos por su matriz de similitud



Este enfoque puede parecer irremediablemente costoso para grandes conjuntos de datos, ya que el cálculo de la matriz de proximidad toma el tiempo O (m^2), donde m es el número de objetos, pero con el muestreo, este método aún se puede utilizar.

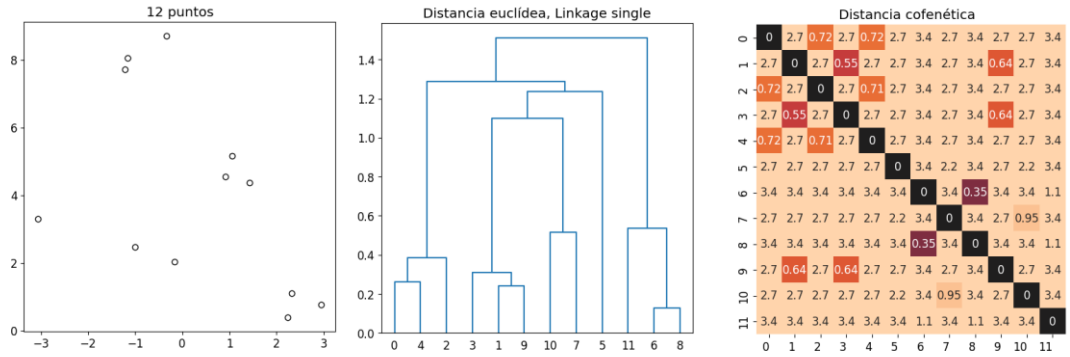
2.4.4. Evaluación no supervisada de la agrupación jerárquica

Los enfoques anteriores para la evaluación de clústeres están destinados a agrupaciones particionales. Para la evaluación de las agrupaciones jerárquicas puede utilizarse la correlación cofenética. La distancia cofenética entre dos objetos es la proximidad a la que una técnica de agrupamiento jerárquico aglomerativo coloca los objetos en el mismo clúster por primera vez. Por ejemplo, si en algún momento del proceso de agrupación jerárquica aglomerativa, la distancia más pequeña entre los dos clústeres que se fusionan es 0.1, entonces todos los puntos de un clúster tienen una distancia cofenética de 0.1 con respecto a los puntos del otro clúster. En una matriz de distancia cofenética, las entradas son las distancias cofenéticas entre cada par de objetos. La distancia cofenética es diferente para cada agrupación jerárquica de un conjunto de puntos.

El coeficiente de correlación cofenética (CPCC) es la correlación entre las entradas de esta matriz y la matriz de disimilitud original y es una medida estándar de qué tan bien se ajusta una agrupación jerárquica (de un tipo particular) a los datos. Uno de los usos más comunes de esta medida es evaluar qué tipo de agrupación jerárquica es mejor para un tipo particular de datos.

El gráfico 2.3 muestra el dendrograma y cálculo de la matriz de distancias cofenéticas para el agrupamiento de un solo enlace de los 12 puntos de la imagen de la izquierda.

Gráfico 2. 4 Matriz de distancia cofenética



2.4.5. Determinación del número correcto de clústeres

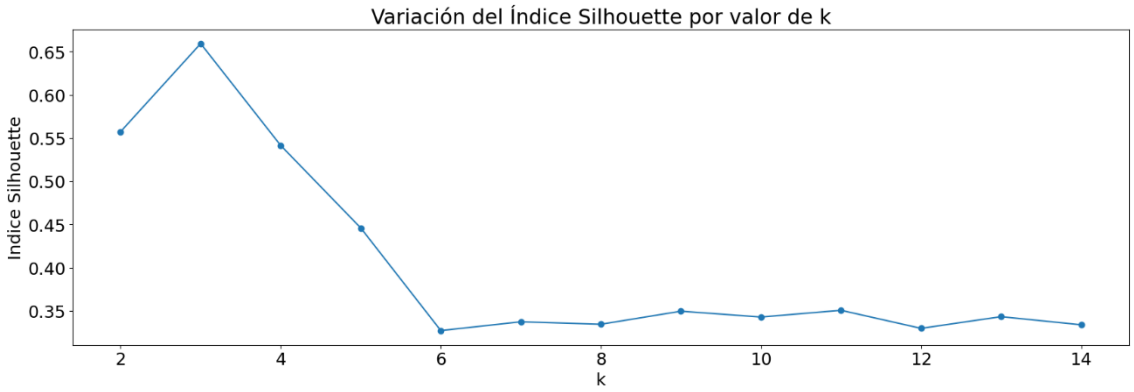
Es posible tratar de encontrar el número natural de grupos en un conjunto de datos buscando el número de grupos en los que hay una rodilla, pico o caída en la gráfica de la medida de evaluación cuando se traza contra el número de grupos. Por supuesto, tal enfoque no siempre funciona bien ya que los grupos pueden estar considerablemente entrelazados, superpuestos o formados en clústeres anidados. Si bien se necesita precaución, esta técnica puede proporcionar información sobre el número de clústeres en los datos.

Una forma de determinar si un conjunto de datos tiene clústeres es intentar agruparlo, sin embargo, casi todos los algoritmos de agrupación en clústeres encontrarán clústeres en los datos. Para abordar este problema, se podrían evaluar los clústeres resultantes y solo afirmar que un conjunto de datos tiene clústeres si al menos algunos de los clústeres son de buena calidad. Sin embargo, este enfoque no aborda el hecho de que los clústeres en los datos pueden ser de un tipo diferente a los buscados por nuestro algoritmo de agrupación.

Para manejar este problema adicional, podríamos usar múltiples algoritmos y evaluar nuevamente la calidad de los clústeres resultantes. Si los clústeres son uniformemente pobres, entonces esto puede indicar que no hay clústeres en los datos. Alternativamente, y este es el enfoque de las medidas de tendencia de agrupamiento, podemos intentar evaluar si un conjunto de datos tiene clústeres sin agrupamiento. El enfoque más común, especialmente para los datos en el espacio euclídeo, ha sido utilizar pruebas estadísticas para la aleatoriedad espacial. Desafortunadamente, elegir el modelo correcto, estimar los parámetros y evaluar la significación estadística de la hipótesis de que los datos no son aleatorios puede ser bastante desafiante. [1]

El gráfico 2.5 muestra los resultados del índice Silhouette para k entre 2 y 15, visualizando que el número ideal de clusters estaría entre 2 y 4

Gráfico 2. 5 Determinación del número correcto de k según el Índice Silhouette



3. Marco experimental

3.1. Origen de los datos

El conjunto de datos "COVID-19. Casos registrados en la República Argentina" utilizado en este trabajo, fue generado, guardado y publicado por la Dirección Nacional de Epidemiología y Análisis de Situación de Salud.

Los datos utilizados para los ensayos fueron descargados el día 10/08/2021 desde [14], en formato .zip y descomprimidos obteniendo el archivo Covid19Casos.csv de 3.578.152KB.

El conjunto de datos descargado incluye la información del día hasta la hora de corte que en ese momento era a las 17:45 hs.

Para abrir el archivo y explorar los datos se utilizó Python y Spark desde Google Colab.

3.1.1. Descripción del conjunto de datos

El archivo correspondiente al 10/08/2021 contiene 16.492.245 filas y 25 columnas que se detallan en la tabla 3.1

Tabla 3. 1 Conjunto de Datos covid-19-casos-registrados-en-la-republica-argentina

Característica	Tipo	Descripción	Valor Mínimo	Valor Máximo	Datos faltantes
id evento caso	Número entero	Número de caso	566790	18452513	0
sexo	Texto	Sexo	F / M / NR		0
edad	Número entero	Edad	-15	1945	5674
edad años meses	Texto	Edad indicada en meses o años	Años / Meses		0
residencia país nombre	Texto	País de residencia	74 países		Sin especificar 787154
residencia departamento nombre	Texto	Departamento de residencia	446 departamentos		Sin especificar 1642648
carga provincia nombre	Texto	Provincia de establecimiento de carga	24 provincias		0
fecha inicio síntomas	Texto	Provincia de establecimiento de carga	01/01/2020	10/08/2021	10731720 None
fecha apertura	Fecha	Fecha de inicio de síntomas	31/01/2020	10/08/2021	3
sepi apertura	Número entero	Semana Epidemiológica de fecha de apertura	1	53	0
fecha internación	Fecha	Fecha de internación	01/01/2020	10/08/2021	16096798 None
cuidado intensivo	Texto	Indicación si estuvo en cuidado intensivo	SI / NO		

fecha cui intensivo	Fecha	Fecha de ingreso a cuidado intensivo en caso de corresponder	21/11/2018	10/08/2021	16421111 None
fallecido	Texto	Indicación de fallecido	SI / NO		0
fecha de fallecimiento	Fecha	Fecha de fallecimiento en el caso de corresponder	09/02/2020	10/08/2021	16366228 None
asistencia respiratoria mecánica	Texto	Indicación si requirió asistencia respiratoria mecánica	SI / NO		0
carga provincia id	Número entero	Código de Provincia de carga	2	94	0
origen financiamiento	Texto	Origen de financiamiento	Privado / Público		0
clasificación	Texto	Clasificación manual del registro	Detalle en 2.1		0
clasificación resumen	Texto	Clasificación del caso	Sospechoso / Sin clasificar / Descartado / Confirmado		0
residencia provincia id	Numero entero	Código de provincia de residencia	2	99	0
fecha diagnostico	Fecha	Fecha de diagnóstico	29/01/2001	10/08/2021	983011
residencia departamento id	Numero entero	Código de departamento de residencia	0	882	0
última actualización	Fecha	Última actualización	10/08/2021		0

De una primera examinación de la base de datos se destacan las siguientes particularidades:

- El 99,94% de los casos corresponden a personas con edades válidas. Se considera como válida la edad entre 0 y 121 años. El 0,06% restante se distribuye entre 5674 casos sin valor y 19 casos con edades con valor negativo o superior a 121 años.
- En su distribución por género el 51,85% de los casos registrados ocurrieron en mujeres, el 47,34% en hombres y el 0,79% restante se ingresó como sexo no registrado.
- El 99,76 % de los casos corresponden a personas mayores a un año, mientras que 38721 casos registrados corresponden a menores de un año.
- Se observan casos registrados correspondientes a personas con residencia en 74 países diferentes, siendo 95,20% residentes argentinos, 4,77% con residencia sin especificar y 0.02 % residentes de otros países.
- En años anteriores a 2019 se registraron 17 casos, en 2019 se registraron 25 casos, en 2020 se registraron 4314320 (26%) casos y hasta agosto de 2021 se registraron 11194872 casos (67,87%).
- De los 16492245 casos, 126017 (0.76%) casos corresponden a fallecidos, 70767 (0.42%) de los casos necesitaron cuidado intensivo y 33143 (0,20%) casos necesitaron asistencia respiratoria mecánica.
- El origen de financiamiento del 75,39% de los casos se clasificó como público y el 24,60% como privado.

- El 58,89% de los casos se clasificaron como “descartado”, el 30,56% de los casos se clasificaron como “confirmado”, el 10.53% de los casos se clasificaron como “sospechoso” y 469 casos no fueron clasificados

3.1.2. Detalles del atributo Clasificación

La página de referencia que se accede desde [15], permite descargar el reporte en formato PDF correspondiente al conjunto de datos que se actualiza diariamente. Se utilizó el reporte diario correspondiente al 10/08/2021 para validar la exactitud de las consultas realizadas con Pyspark desde Google Colab, considerando que las diferencias encontradas podrían corresponder a la hora de corte que utiliza el reporte y la de subida del archivo que contiene el conjunto de datos.

Agrupando los casos en los que el atributo clasificación contiene la palabra Activo se obtuvieron un total de 241759 casos, el reporte PDF muestra 241799.

La tabla 3.2 muestra la cantidad de casos clasificados como Activos correspondiente a cada clasificación.

Tabla 3. 2 Cantidad de casos clasificados como activos

Caso confirmado por criterio clínico-epidemiológico – Activo	4350
Caso confirmado por laboratorio – Activo Internado	154653
Caso confirmado por laboratorio – Activo	75934
Caso confirmado por criterio clínico – epidemiológico - Activo internado	6822

La tabla 3.3 muestra los resultados obtenidos al agrupar el atributo clasificación si contiene la palabra “Fallecido”. Se obtiene un total de 108203 casos contra los 108165 obtenidos del reporte PDF.

Tabla 3. 3 Cantidad de casos clasificados como fallecidos

Caso confirmado por criterio clínico-epidemiológico - Fallecido	2268
Caso confirmado por laboratorio - Fallecido	105935

La tabla 3.4 muestra los resultados obtenidos al agrupar el atributo clasificación si contiene la palabra “No activo”. Estos casos se clasifican como recuperados y la cantidad de casos obtenida 4691523 coincide con el valor del reporte PDF.

Tabla 3. 4 Cantidad de casos clasificados como recuperados

Caso confirmado por criterio clínico-epidemiológico - No activo (por tiempo de evolución)	430686
Caso confirmado por laboratorio - No Activo por criterio de laboratorio	115235
Caso confirmado por laboratorio - No activo (por tiempo de evolución)	4145602

El total de casos confirmados se calculó sumando los casos activos, recuperados y fallecidos:

$$241759 + 108203 + 4691523 = 5041485$$

obteniendo la cantidad total de 5041485 casos siendo 5041487 la cantidad total informada por el reporte PDF.

En la tabla 3.5 se muestran agrupadas por clasificación 9713027 casos con distintas clasificaciones.

Tabla 3. 5 Otras clasificaciones

Caso Descartado	9677747
Caso Invalidado Epidemiológicamente	25019
Otro diagnóstico	9792
Sin clasificar	469

La tabla 3.6 muestra las cantidades de casos agrupadas por clasificación de los 1202970 casos negativos no concluyentes.

Tabla 3. 6 Casos negativos con resultados no conclusivos

Caso con resultado negativo-no conclusivo - No activo	1091019
Caso con resultado negativo-no conclusivo – Activo	97541
Caso con resultado negativo-no conclusivo - Activo internado	13860
Caso con resultado negativo-no conclusivo – Fallecido	550

La tabla 3.7 muestra las cantidades de casos agrupadas por clasificación de los 534763 casos sospechosos.

Tabla 3. 7 Casos sospechosos = 534763

Caso sospechoso - No Activo - Sin muestra	241463
Caso sospechoso - No Activo - Con muestra sin resultado concluyente	232218
Caso sospechoso - Activo - Con muestra sin resultado concluyente	13425
Caso sospechoso - Activo Internado - Con muestra sin resultado concluyente	13106
Caso sospechoso - Activo - Sin muestra	8791
Caso sospechoso - No Activo - Muestra no apta	4786
Caso sospechoso - Fallecido - Sin muestra	2155
Caso sospechoso - Fallecido - Con muestra sin resultado concluyente	1166
Caso sospechoso - Activo Internado - Muestra no apta	476
Caso sospechoso - Activo - Muestra no apta	75
Caso sospechoso - Fallecido - Muestra no apta	53
Caso sospechoso - Activo Internado - Sin muestra	17049

3.2. Herramientas utilizadas para la exploración de datos y ensayos

Para la exploración de datos, creación y entrenamiento de modelos y visualizaciones gráficas se utilizaron las herramientas que se describen a continuación.

3.2.1. Google Colab

"Colaboratory" es un entorno interactivo que permite programar y ejecutar Python desde el navegador con las ventajas de no requerir configuración, proporcionar acceso gratuito a GPUs y permitir compartir contenido fácilmente.

Los cuadernos de Colab son cuadernos de Jupyter alojados en Colab. Jupyter Notebook es una aplicación web original para crear y compartir documentos que ofrece una experiencia simple y optimizada.

Estos cuadernos se almacenan en la cuenta de Google Drive del usuario conectado, permitiendo compartirlos fácilmente con otros usuarios con la finalidad de comentarlos o incluso editarlos.

Entre las facilidades para el área de ciencia de datos se destaca que permite aprovechar toda la potencia de las bibliotecas más populares de Python para analizar y visualizar datos.

Los cuadernos de Colab ejecutan código en los servidores en la nube de Google, lo que permite aprovechar la potencia del hardware de Google, incluidas las GPU y TPU, independientemente de la potencia del equipo desde donde se ejecuta el navegador. [16]

3.2.2. PySpark

PySpark es una interfaz para Apache Spark en Python. Permite escribir aplicaciones utilizando las API de Python y proporciona el shell de PySpark para analizar de forma interactiva sus datos en un entorno distribuido. PySpark es compatible con la mayoría de las funciones de Spark, como Spark SQL, DataFrame, Streaming, MLlib (Machine Learning) y Spark Core.

- La API de pandas en Spark permite escalar la carga de trabajo de pandas.
- La función de transmisión en Apache Spark permite potentes aplicaciones interactivas y analíticas tanto en transmisión como en datos históricos, al tiempo que hereda la facilidad de uso y las características de tolerancia a fallas de Spark.
- Spark SQL es el módulo de Apache Spark para el procesamiento de datos estructurados. Proporciona una abstracción de programación llamada DataFrame y también puede actuar como un motor de consulta SQL distribuido. Utilizable en Java, Scala, Python y R. Brinda acceso uniforme a los datos permitiendo acceder a una variedad de orígenes de datos incluidos Hive, Avro, Parquet, ORC, JSON y JDBC y unir datos de estas fuentes.

3.2.2.1. MLlib

MLlib [17] es la biblioteca de aprendizaje automático (ML) de Spark. Su objetivo es hacer que el aprendizaje automático práctico sea escalable y fácil. A un alto nivel, dispone de las herramientas que se enumeran a continuación:

- Algoritmos de ML: están incluidos los algoritmos de aprendizaje automático más comunes como clasificación, regresión, agrupamiento y filtrado colaborativo
- Caracterización: extracción de características, transformación, reducción de dimensionalidad y selección
- Pipelines: herramientas para construir, evaluar y ajustar ML Pipelines
- Persistencia: guardar y cargar algoritmos, modelos y Pipelines
- Utilidades: álgebra lineal, estadística, manejo de datos, etc.

A partir de Spark 2.0, las API basadas en RDD en el paquete Spark MLlib entraron en modo de mantenimiento. La API de aprendizaje automático principal para Spark ahora es la API basada en DataFrame en el paquete spark.ml. Este cambio se debe a que los DataFrames proporcionan una API más fácil de usar que los RDD. Los muchos beneficios de DataFrames incluyen Spark Datasources, consultas SQL/DataFrame, optimizaciones de Tungsten y Catalyst, y API uniformes en todos los lenguajes. Otro motivo es que la API basada en DataFrame para MLlib proporciona una API uniforme en los algoritmos de ML y en varios lenguajes. Por último, los DataFrames facilitan las canalizaciones prácticas de ML, en particular las transformaciones de funciones.

3.2.3. Formato Parquet

Para el almacenamiento de datos intermedios y resultados se crearon archivos en formato parquet.

El formato parquet [18] es un tipo de formato clasificado como orientado a columnas (column oriented file format) donde cada columna se almacena de forma independiente. Este formato reduce el tamaño total de los datos y optimiza el rendimiento mientras procesa los datos porque funciona en subconjuntos de columnas requeridas en lugar de la totalidad de los datos [19].

Es un formato de datos autodescriptivo que integra el esquema o la estructura dentro de los datos en sí. Es decir, propiedades o metadatos de los datos como el tipo, si es un número entero, un real o una cadena de texto, el número de valores, el tipo de compresión, etc. están incluidas en el propio archivo junto con los datos como tal. De esta forma, cualquier programa que se utilice para leer los datos, puede acceder a estos metadatos, para, por ejemplo, determinar sin ambigüedades, qué tipo de datos se espera leer en una columna determinada.

3.2.4. QlikView

QlikView [20] es una plataforma de descubrimiento de datos que permite desarrollar e implementar aplicaciones de análisis guiado de alto valor. Es intuitivo y proporciona una capacidad de análisis ultra rápido, en memoria, mediante la integración y presentación dinámica de los datos, procedentes de múltiples fuentes.

Es el primer producto para el análisis detallado de datos de diversas fuentes que desarrolló la empresa Qlik y fue lanzado al mercado en 1993. Este software fue diseñado como una herramienta de escritorio, pero luego de diferentes versiones ofrece una solución web basada en servidor.

QlikView se enfoca en la creación de análisis guiados. Para desarrollar un flujo de datos ofrece conectores de datos predeterminados y personalizados; una vez que las aplicaciones son creadas por desarrolladores de documentos, los analistas podrán explorar, buscar y analizar los datos con libertad, es decir, no se limita a rutas de exploración predefinidas.

En QlikView, los datos se comprimen, procesan y almacenan en la memoria RAM del servidor, dando disponibilidad inmediata de estos a múltiples usuarios para que los exploren. Cuando el volumen de datos es muy grande para ser guardada en memoria, se conectará a la fuente de datos directamente. QlikView presenta un entorno de desarrollo de scripts que requiere ciertas habilidades de programación para llevar a cabo la carga y transformación de los datos.

Qlik ofrece una versión QlikView de uso gratuito para uso personal. QlikView Personal Edition es una solución completa de QlikView Desktop, que no requiere una clave de licencia, en su lugar, el documento de QlikView se guarda con la clave del usuario que vincula ese archivo a la computadora donde se crea, es decir que no pueden usarse en otra computadora, ni con otro usuario [21]

Los gráficos 3.1.1, 3.1.2 y 3.1.3 muestran la exploración inicial utilizando herramientas de visualización de los datos como QlikView.

Gráfico 3.1. 1 Fallecidos por grupo etario y sexo

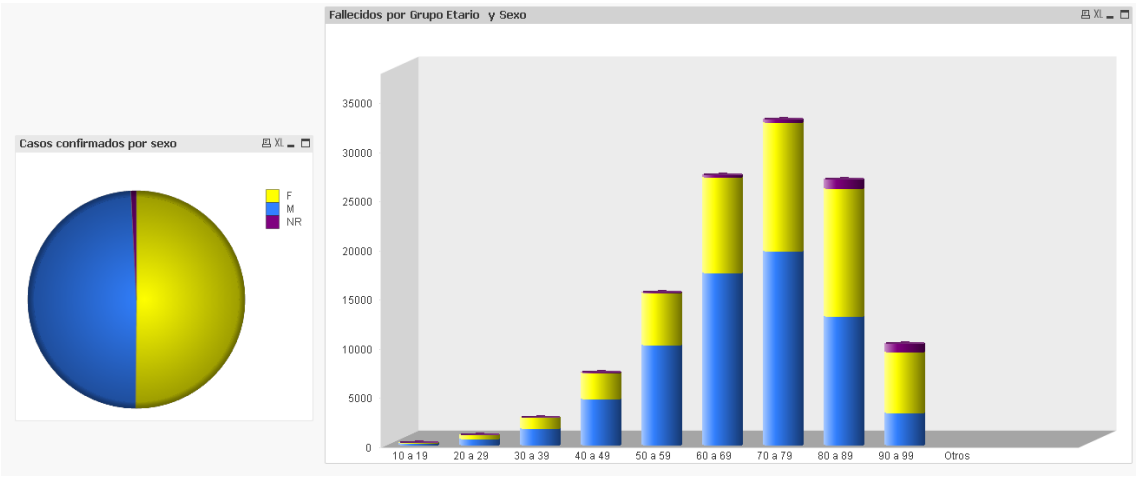


Gráfico 3.1.2 Casos confirmados y fallecidos al 10/08/2021

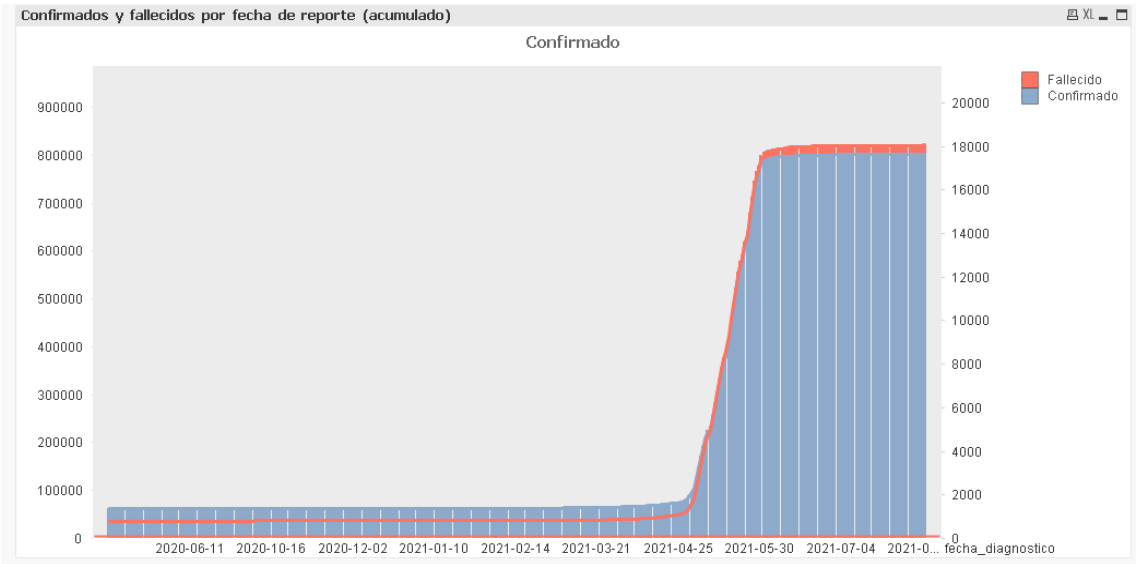
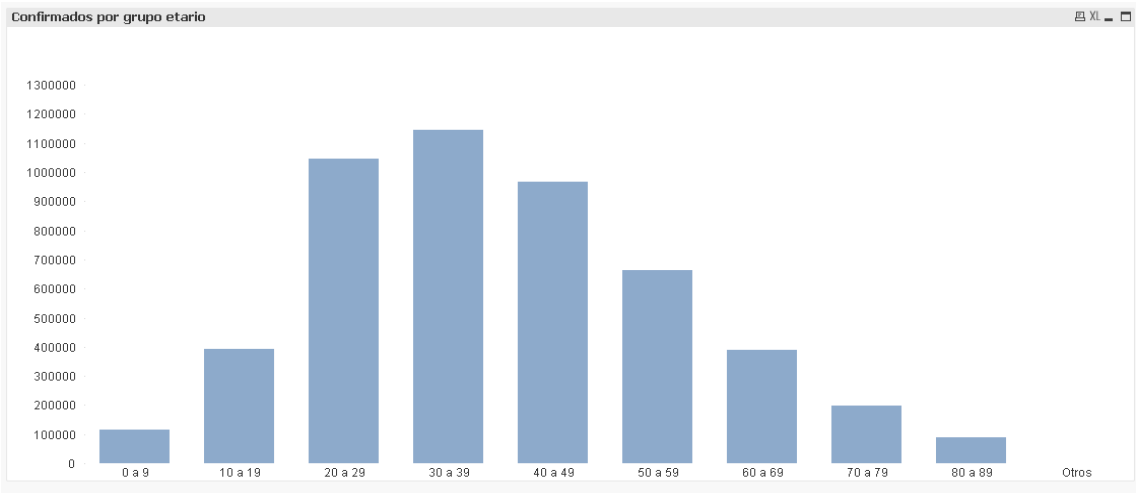


Gráfico 3.1. 31 Casos confirmados agrupados por grupo etario al 10/08/2021



4. Preprocesamiento de datos.

En este capítulo se describen las técnicas utilizadas para la integración, limpieza y transformación de datos que garanticen que los mismos estén íntegros, completos y consistentes. Además, se presentan los conjuntos de datos a utilizar en los ensayos luego del preprocesamiento y su correspondiente análisis correlacional.

4.1. Selección de atributos

La selección de características, variables o atributos tiene cuatro objetivos principales. En primer lugar, permite reducir el tamaño de los datos, al eliminar características o atributos de todos los ejemplos que puedan ser irrelevantes o redundantes. En segundo lugar, una buena selección de características puede mejorar la calidad del modelo, al permitir al método de minería de datos centrarse en las características relevantes. En tercer lugar, una buena selección de características permite expresar el modelo resultante en función de menos variables; esto es especialmente importante cuando se desean modelos comprensibles como árboles de decisión, regresión lineal, entre otros. En cuarto lugar, se puede requerir una reducción de dimensionalidad a dos o tres características exclusivamente, con el propósito de representar los datos visualmente. Existen otras razones para eliminar atributos, por ejemplo, cuando existen muchos datos erróneos o faltantes en un atributo, y es preferible deshacerse de él.

Algunos atributos son fáciles de eliminar, por ejemplo, si un atributo es constante, es decir, tiene el mismo valor para todas las instancias es claramente eliminable.

Existen dos reglas generales para eliminar características, en atributos nominales, la primera de ellas es la eliminación de partes de claves candidatas. La regla general es eliminar cualquier atributo que pueda ser clave primaria de la tabla o que sea clave candidata o incluso parte de clave candidata, parcial o totalmente. Una manera sencilla de saber si un atributo nominal es demasiado específico y debe ser eliminado es ver si tiene casi tantos valores como ejemplos. Si no se elimina este tipo de atributos puede ser especialmente problemático para tareas de clasificación o de regresión.

La segunda regla es la eliminación de atributos dependientes. En la teoría de la normalización de bases de datos, cuando existen dependencias funcionales entre atributos se intenta normalizar en varias tablas. El conjunto de datos descargado tiene atributos desnormalizados como

carga_provincia_nombre,	carga_provincia_id,
residencia_provincia_nombre,	residencia_departamento_id,
residencia_departamento_nombre,	residencia_provincia_

entre otros, que resultan redundantes. Una buena opción es mantener sólo el atributo descriptivo eliminando el id, en este trabajo se incluyó para algunos ensayos el atributo carga_provincia_nombre. No eliminar los atributos dependientes es especialmente crítico para las tareas de agrupamiento.

En general, detectados los atributos irrelevantes y redundantes, es posible utilizar técnicas más sofisticadas para seguir reduciendo la dimensionalidad, en particular para los atributos numéricos. Existen dos tipos generales de métodos para seleccionar características.

• Métodos de filtro o métodos previos: se filtran los atributos irrelevantes antes de cualquier proceso de minería de datos y, en cierto modo, independiente de él. Las técnicas, son fundamentalmente estadísticas, entre ellas medidas de información, distancia, dependencia o inconsistencia. El criterio para establecer el subconjunto de características "óptimo" se basa en medidas de calidad previa que se calculan a partir de los datos mismos.
• Métodos basados en modelo o métodos de envolvente (wrapper): la bondad de la selección de atributos se evalúa respecto a la calidad de un modelo de minería de datos o estadístico extraído a partir de los datos utilizando algún buen método de validación. Lógicamente, este tipo de técnicas requieren mucho más tiempo que las otras, ya que para evaluar hay que entrenar un modelo.

Sean de filtro o basados en modelo, ambos métodos pueden ser iterativos, es decir, se van eliminando atributos y se va observando el resultado sobre la medida de calidad previa o la medida de calidad del modelo. Se van recuperando o eliminando más atributos de una manera iterativa, hasta que se obtiene una combinación que maximiza la calidad. Existen muchas maneras de realizar este procedimiento, por ejemplo, empezando con un atributo y elegir el que dé mayor calidad de la selección con atributos, después añadir el atributo que dé mayor calidad de selección con dos atributos, y así hasta que no se mejore la calidad o se llegue al número deseado de atributos (estrategia forward). La manera inversa, comenzar con todos e ir eliminando o maneras mixtas también se pueden utilizar. Otra manera, más simple pero más costosa, consiste en realizar selecciones aleatorias de atributos, hasta que se encuentra una selección satisfactoria (estrategia backward). El objetivo en los dos casos es el mismo, obtener un subconjunto de atributos representativo que contenga la mayor parte de la información que existía en el conjunto inicial y que no exista la menor correlación entre los atributos seleccionados. [2]

Los atributos del conjunto de datos a analizar se pueden dividir en tres categorías más amplias. Información del caso (Como clave principal id_evento_caso, sexo, edad, edad_años_meses, clasificación, clasificación resumen), información temporal (fecha de inicio de síntomas, fecha de apertura, semana de apertura, fecha de internación, fecha de fallecimiento, fecha de diagnóstico, etc.) y datos geográficos (residencia_pais_nombre, residencia_provincia_nombre, carga_provincia_nombre, etc.).

En este trabajo se seleccionaron como atributos relevantes a las características de información del caso descartando los atributos id_evento_caso y clasificación. Esta elección se realiza considerando relevantes aquellos atributos que casi no tienen valores faltantes teniendo en cuenta que se trata de un conjunto de datos públicos generados, guardados y almacenados por organismos del gobierno de la República Argentina que abordan un tema de actualidad. El atributo id_evento_caso se descartó para el entrenamiento de los modelos de agrupamiento, pero se utiliza para identificar los casos y el atributo clasificación se eliminó por ser demasiado específico y estar incluido en clasificación_resumen. Para las pruebas de dimensionalidad se agregó además la provincia de carga, por la misma razón que las anteriores, pero también para analizar si influye incluirlas o no en la formación de grupos.

Es importante señalar que una característica que no contiene ninguna información relevante para el objetivo a analizar podría provocar que la agrupación sea menos evidente. La ocurrencia de varias de estas características irrelevantes produce muchos términos aleatorios en las distancias, ocultando así la información útil proporcionada por las otras características. Por lo tanto, a tales características no informativas se les debe dar un peso cero en el análisis, lo que equivale a eliminarlas [22].

4.2. Limpieza y preparación de datos.

La preparación de datos tiene como objetivo la eliminación del mayor número posible de datos erróneos o inconsistentes e irrelevantes y tratar de presentar los datos de la manera más apropiada para la minería de datos. Pero además de la irrelevancia, existen otros problemas que afectan a la calidad de los datos. Uno de esto problemas es la presencia de valores que no se ajustan al comportamiento general de los datos (outliers). Estos datos anómalos pueden representar errores en los datos o pueden ser valores correctos que son simplemente diferentes a los demás. La limpieza de datos puede en muchos casos, detectar y solucionar estos problemas. Los datos faltantes pueden ser originados muchas veces al integrar fuentes diferentes, para los cuales no existen soluciones fáciles [2].

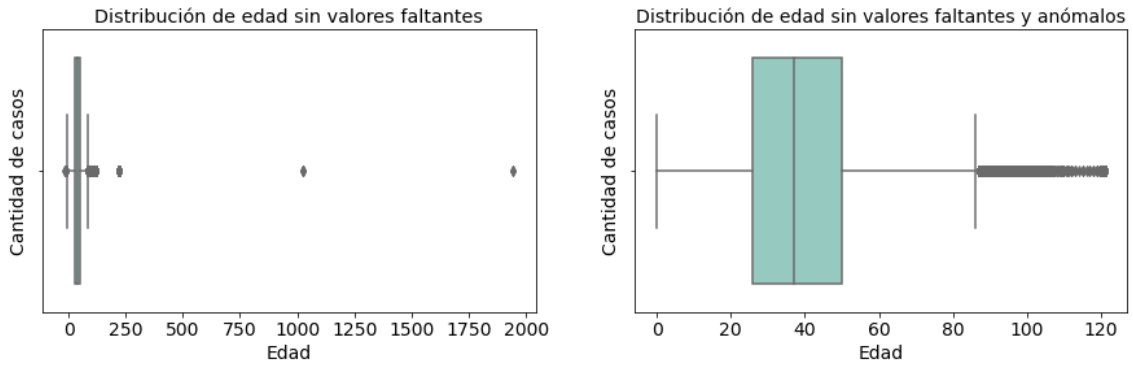
Los atributos en consideración con excepción de la edad son atributos nominales sin valores faltantes que fueron convertidos a variables numéricas.

La variable edad tiene 5674 casos con valores faltantes que fueron eliminados. También se descartaron casos con valores anómalos de edad como los menores a 0 y mayores a 121.

Para estudiar visualmente la frecuencia de la variable edad se utilizó un diagrama de caja. El diagrama de caja o boxplot es un tipo de gráfico que muestra un resumen de una gran cantidad de datos en 5 medidas descriptivas, además de intuir su morfología y simetría. Este tipo de gráficos permite identificar valores atípicos y comparar distribuciones, además de conocer de una forma cómoda y rápida como el 50% de los valores se distribuyen.

El gráfico 4.1 muestra la distribución del atributo edad con valores anómalos en el boxplot de la izquierda y la distribución de edad luego de eliminar los valores anómalos en el boxplot de la derecha. La eliminación de los 5674 valores faltantes fue realizada antes de graficar ambos diagramas de caja.

Gráfico 4. 1 Distribuciones del atributo edad



La caja va del primer cuartil al tercer cuartil, es decir, del 25% de los datos al 75% de los datos. Es decir, la caja muestra el rango intercuartil, que contiene el 50% de los datos. La mediana se representa con una línea vertical y muestra el segundo cuartil, el valor tal que la mitad de los datos estén por debajo y la mitad por encima. Los bigotes o líneas acabadas en un segmento muestran el resto de los datos hasta los valores más extremos. El valor de edad válida máxima igual a 121 se decidió teniendo en cuenta la existencia de 616 casos cargados con dicha edad y que es un valor alto pero probable. Los 40 casos con edad 221, los 3 casos con edad 1024 y los 2 casos con edad 1945 se consideraron valores anómalos y fueron descartados.

El atributo “edad_años_meses” se eliminó analizando que para el valor “meses” la edad puede tomarse como 0. Los 38721 casos correspondientes a menores de un año se incluyeron con edad 0, perdiendo el detalle de meses que resulta irrelevante teniendo en cuenta que apenas representan el 0,24% de los casos.

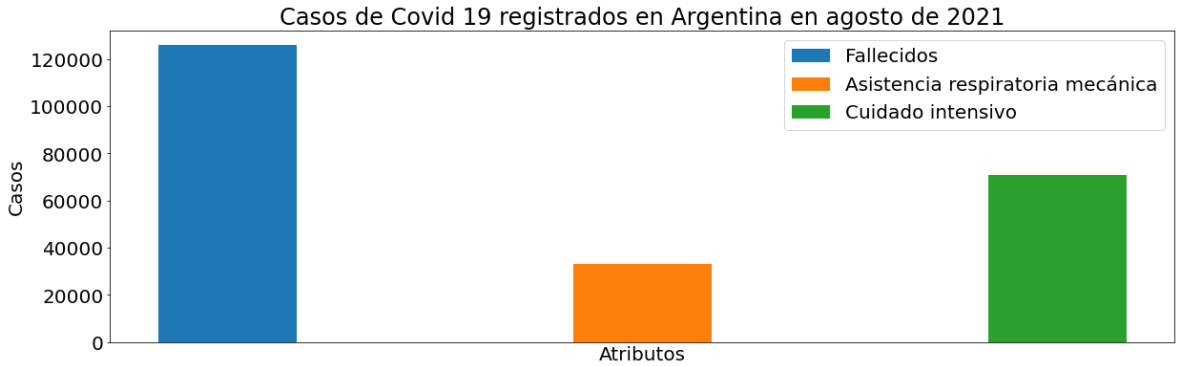
Los atributos “fallecido”, “asistencia respiratoria mecánica” y “cuidado intensivo”, son atributos binarios asimétricos para los cuales los resultados no son igualmente importantes [22]. Si bien se puede decir que dos personas que recibieron “cuidado intensivo” tienen algo en común, no está tan claro si se puede decir lo mismo de dos personas que no lo recibieron.

Por convención, se reemplaza el resultado más importante por 1, y el otro por 0, para que el acuerdo de dos unos se considere más significativo que el acuerdo de dos ceros.

Para describir variables cualitativas se utilizaron gráficos de barras donde es posible contar las observaciones.

El gráfico 4.2 muestra la cantidad de casos con valor 1 en los atributos fallecido, asistencia respiratoria mecánica y cuidado intensivo.

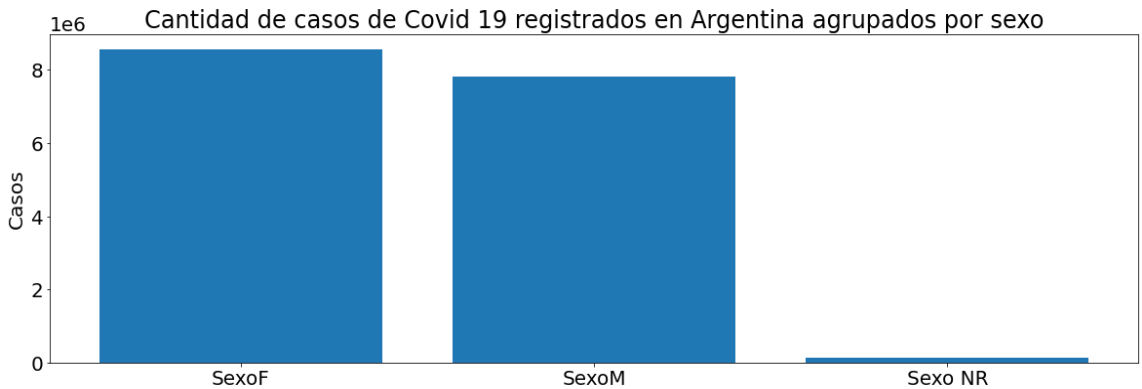
Gráfico 4. 2 Cantidad de casos por atributos



Para atributos nominales con más de 2 estados se aplicó la estrategia conocida como one-hot, la cual consiste en codificar los datos a un mayor número de variables binarias asimétricas, creando una nueva variable binaria para cada uno de los estados nominales y luego asignar el valor igual a 1 si se produce el estado correspondiente y a 0 en caso contrario.

El atributo sexo posee los estados nominales "M" masculino, “F" femenino y “NR” no registrado. Los tres estados son igualmente valiosos y tienen el mismo peso, por este motivo se reemplazaron por las variables binarias asimétricas “sexoF”, “sexoM” y “sexoNR”. El gráfico 4.3 muestra la cantidad de casos agrupados por sexo.

Gráfico 4. 3 Cantidad de casos agrupados por sexo



El atributo clasificación resumida se representó con las variables binarias asimétricas “descartado”, “confirmado”, “sospechoso” y “sin clasificar”.

El atributo origen de financiamiento posee los estados nominales “público” y “privado” al igual que en los casos anteriores no hay preferencia por qué resultado debe codificarse como 0 o 1. Este atributo fue representado por las variables binarias asimétricas “público” y “privado”.

El atributo “provincia de carga” se representó con 24 variables binarias asimétricas, una por cada provincia.

Los gráficos 4.4, 4.5 y 4.6 muestran la cantidad de casos agrupados por clasificación resumen, origen de financiamiento y provincias respectivamente.

Gráfico 4. 4 Cantidad de casos agrupados por Clasificación

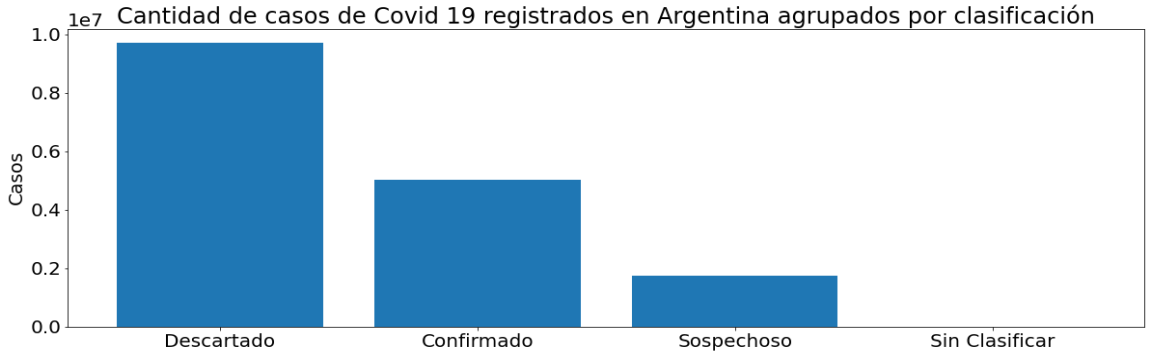


Gráfico 4. 5 Cantidad de casos agrupados por origen de financiamiento

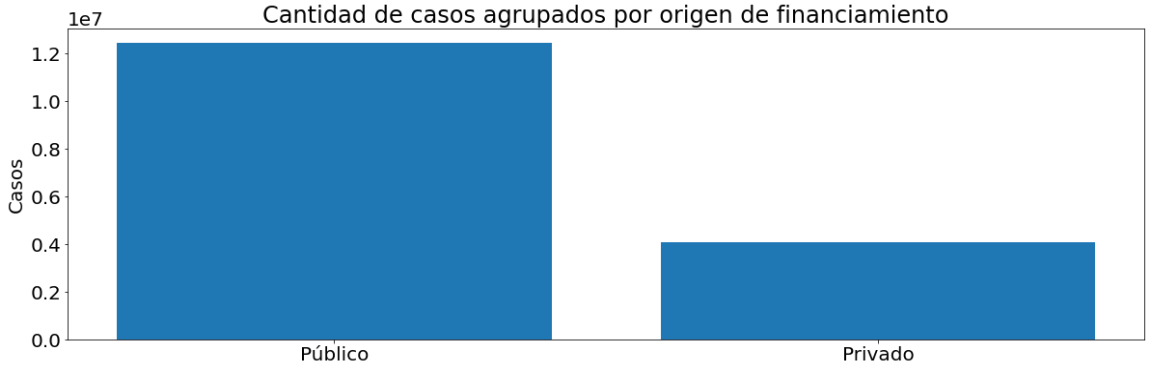
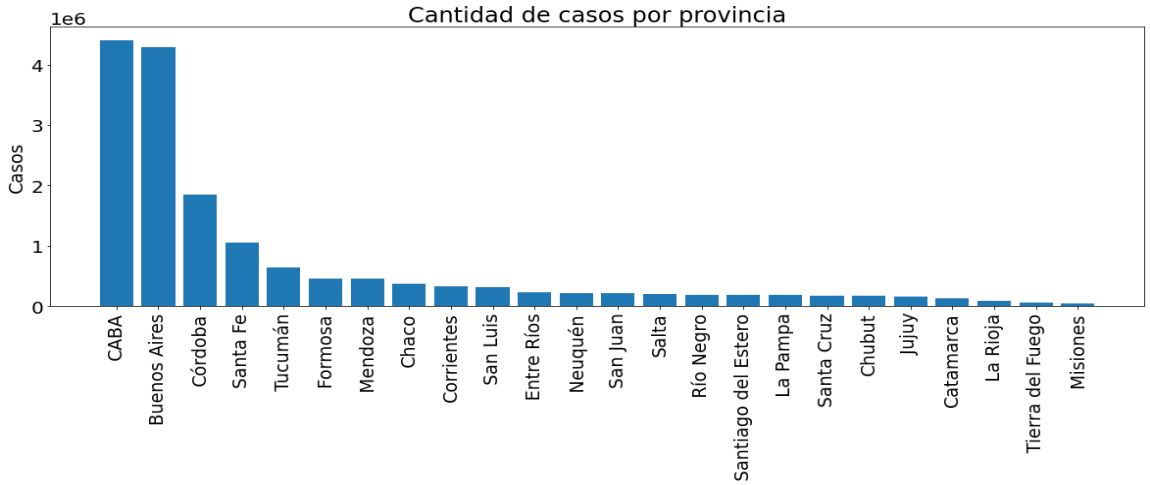


Gráfico 4. 6 Cantidad de casos agrupados por provincia



En la tabla 4.1 se muestra una fracción de los datos originales y en la tabla 4.2 los mismos datos transformados.

Tabla 4. 1 Datos Originales

Id evento caso	Sexo	Edad	Edad años meses	Cuidado intensivo	Fallecido	Asistencia respiratoria mecánica	Origen financiamiento	Clasificación resumen
17526667	F	15	Años	NO	NO	NO	Privado	Sospechoso
1754869	F	48	Años	NO	NO	NO	Público	Descartado
17570712	M	22	Años	NO	NO	NO	Público	Descartado
17588851	F	17	Años	NO	NO	NO	Público	Confirmado
17589437	M	38	Años	NO	NO	NO	Público	Confirmado
17604422	M	33	Años	NO	NO	NO	Público	Descartado

Tabla 4. 2 Datos Transformados

Id evento caso	SexoF	SexoM	SexoNR	Edad	Cuidado intensivo	Fallecido	Asistencia respiratoria mecánica	Privado	Público	Descartado	Confirmado	Sospechoso	Sin clasificar
17526667	1	0	0	15	0	0	0	1	0	0	0	1	0
1754869	1	0	0	48	0	0	0	0	1	1	0	0	0
17570712	0	1	0	22	0	0	0	0	1	1	0	0	0
17588851	1	0	0	17	0	0	0	0	1	0	1	0	0
17589437	0	1	0	38	0	0	0	0	1	0	1	0	0
17604422	0	1	0	33	0	0	0	0	1	1	0	0	0

4.2.1. Generación de vectores de características.

Para crear cualquier algoritmo de agrupación en clústeres con PySpark, es necesario realizar algunas transformaciones de datos.

Todos los atributos relevantes se convirtieron en un vector de características utilizando un **ensamblador de vectores.**

Un ensamblador de vectores es un transformador que combina una lista determinada de características en una sola columna de vector con la finalidad de entrenar modelos. El ensamblador de vectores acepta los tipos de atributos de entrada numérico, booleano y vectorial.

En cada fila, se extrajeron los atributos relevantes con excepción de la identificación del caso (id_evento_caso), se pasaron como parámetros de entrada al ensamblador de vectores y se transformaron en un vector de características.

Los atributos transformados fueron estandarizados para llevarlos a una escala comparable, ya que la distancia euclidiana siempre se ve más afectada por las variables en una escala más alta. Si una variable tiene una escala mucho mayor que el resto, determinará en gran medida el valor de distancia/similitud obtenida al comparar las observaciones, dirigiendo así la agrupación final. Escalar y centrar las variables antes de calcular la matriz de distancias para que tengan media 0 y desviación estándar 1, asegura que todas las variables tengan el mismo peso cuando se realice el agrupamiento. En el conjunto de datos analizado en este trabajo la edad oscila entre los valores 0 y 121, mientras que el resto de las variables binarias alternan entre 0 y 1.

4.3. Conjuntos de datos.

Los 8 atributos de interés edad, sexo, fallecido, asistencia respiratoria mecánica, cuidado intensivo, origen de financiamiento y clasificación resumen se transformaron en 13 atributos o 37 atributos dependiendo si se incluía o no la provincia de carga.

El conjuntos de datos con 37 características al que de aquí en adelante llamaremos Datos37 contiene los atributos Edad, sexoF, sexoM, sexoNR, fallecido, asistencia_respiratoria_mecanica, público, privado, cuidado_intensivo, confirmado, descartado, sin_clasificar, sospechoso, Tierra_del_Fuego, San_Luis, Chubut, Jujuy, Neuquen, CABA, Santa_Cruz, Entre_Rios, La_Pampa, Tucumán, Chaco, Mendoza, Santa_Fe, Corrientes, Buenos_Aires, Cordoba, Formosa, Salta, San_Juan, Santiago_del_Estero, Misiones, Catamarca, Rio_Negro, La_Rioja

El conjunto de datos con 13 características al que de aquí en adelante llamaremos Datos13 contiene los atributos Edad, sexoF, sexoM, sexoNR, fallecido, asistencia_respiratoria_mecanica, público, privado, cuidado_intensivo, confirmado, descartado, sin_clasificar, sospechoso.

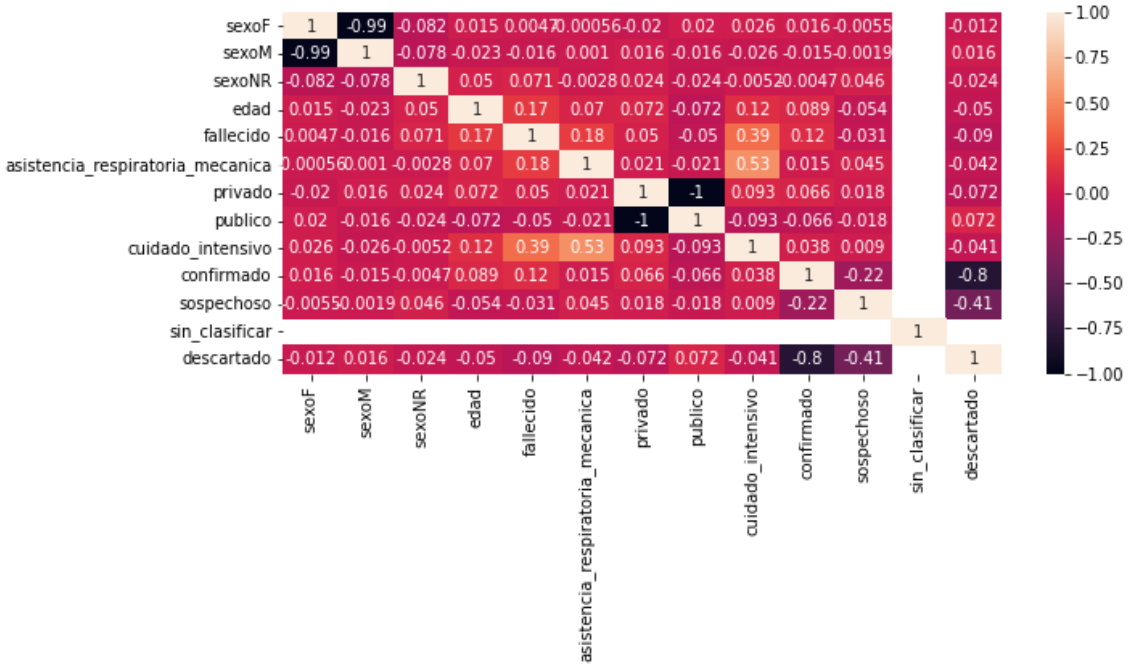
4.4. Matriz de correlación.

Los coeficientes de correlación son útiles para fines de agrupación porque miden el grado en que dos variables están relacionadas en términos de patrón o forma. [23]

La exploración de las correlaciones entre las características se analiza luego de transformar los atributos categóricos a variables numéricas para confirmar que no existan datos redundantes.

En el gráfico 4.8 se muestra la correlación entre las características de información del caso transformadas y una fracción de 0.001 del conjunto de datos.

Gráfico 4. 7 Matriz de correlación Pearson para una fracción del conjunto de datos de 1985 casos



Los valores cercanos a +1 indican la presencia de una fuerte relación positiva entre las dos características, mientras que los cercanos a -1 indican una fuerte relación negativa entre ambas características.

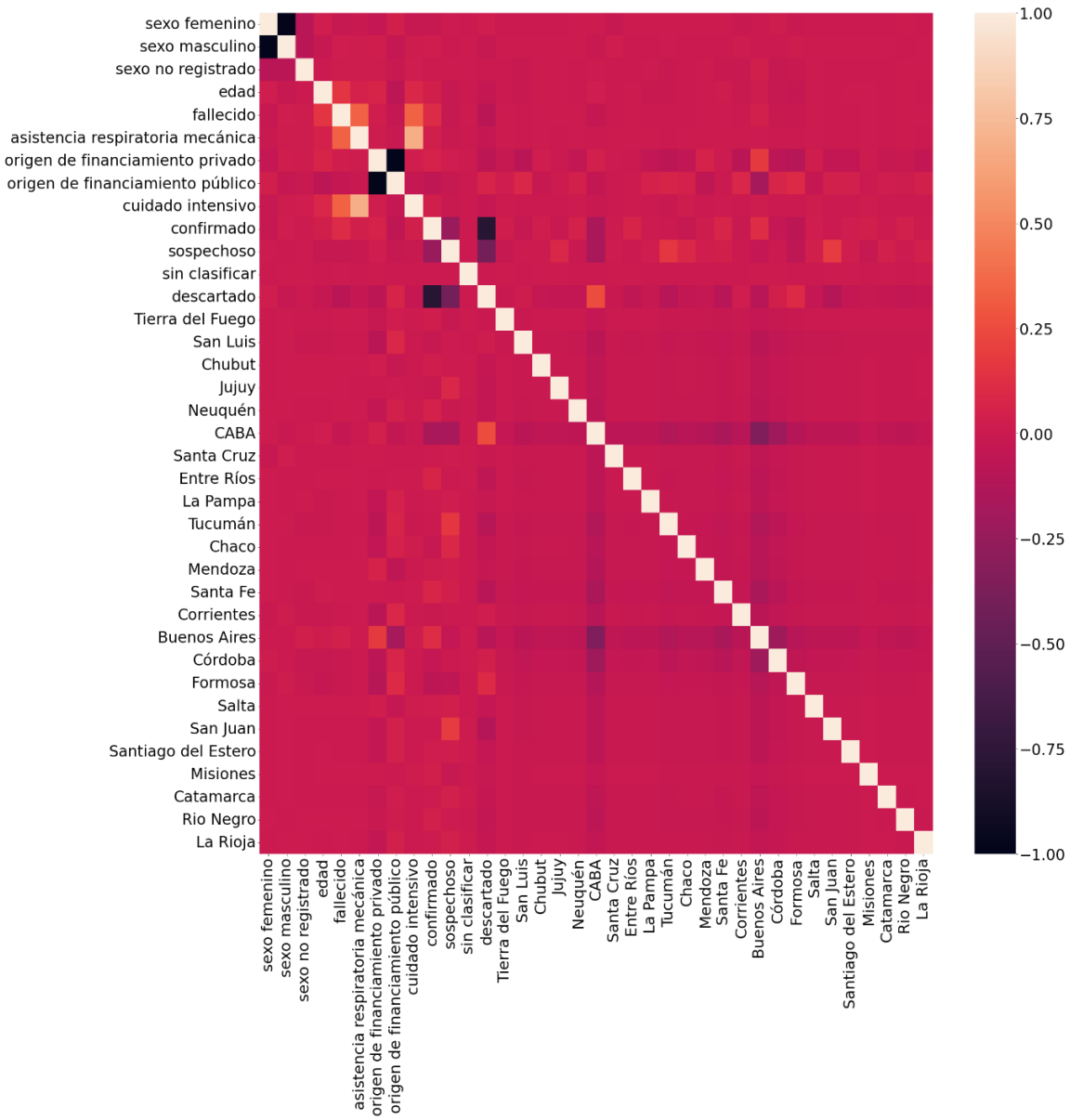
Los valores cercanos a cero indican que no existe ninguna relación entre las características.

La fila y columna en blanco asociados al atributo sin_clasificar indican la ausencia de observaciones con esta característica en la fracción del conjunto de datos. Por este motivo se vuelven a calcular las correlaciones sobre los conjuntos de datos completos.

En el gráfico 4.9 se muestra la matriz de correlación de Pearson sobre Datos37, no se imprimen los coeficientes de correlación para ganar legibilidad.

Se observan correlaciones negativas muy marcadas en los atributos sexo y origen de financiamiento (características representadas por sexoF/sexoM y público/privado) que indican que cuanto más aumenta una característica más disminuye la otra. La correlación de Pearson entre las variables cuidado intensivo y asistencia respiratoria mecánica tiene un valor de 0,681279, es un valor relativamente alto porque existe una relación positiva entre estas dos variables, ya que la asistencia respiratoria mecánica puede considerarse un tipo de cuidado intensivo.

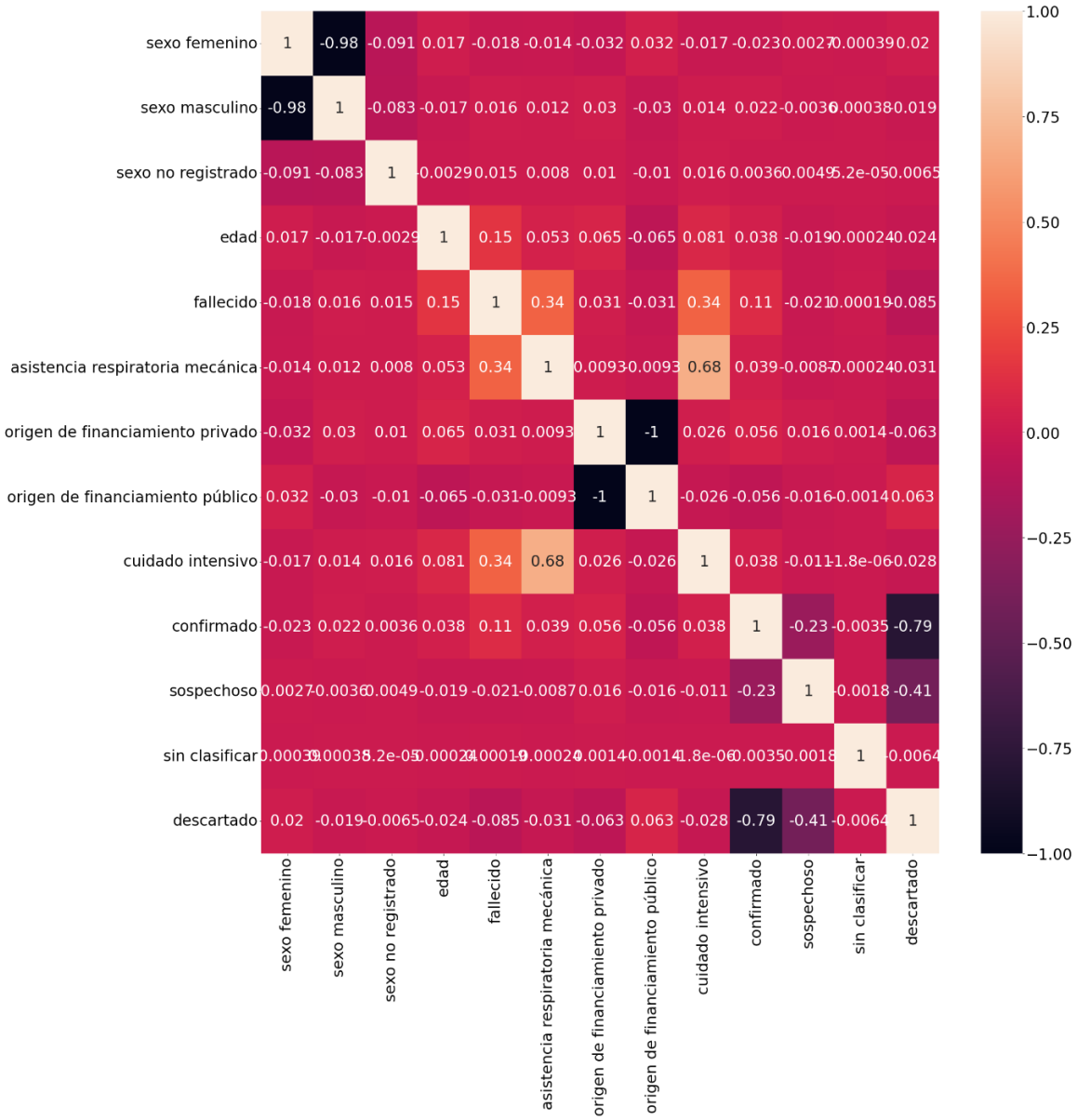
Gráfico 4. 8 Matriz de correlación de Pearson para el conjunto de datos completo



Las correlaciones negativas fuertes ocurren en los atributos público y privado con valor -1, sexoF y sexoM con un valor de -0.98 y descartado y confirmado con valor -0.79.

En el gráfico 4.10 se muestra la matriz de correlación de Pearson sobre Datos13. En esta matriz se incluyeron los coeficientes de correlación porque el tamaño de la matriz permite que sean legibles.

Gráfico 4. 9 Matriz de correlación de Pearson para el conjunto de datos completo con 13 atributos



5. Experimentación

En la primera sección de este capítulo se describen los ensayos realizados para medir tiempos de ejecución y comprobar que los algoritmos de agrupamiento seleccionados escalen de manera razonable en conjuntos de datos de gran tamaño.

En la segunda sección se analizan los resultados obtenidos por los algoritmos de clustering estudiados mediante el uso de distintos criterios de validación.

En la tercera sección se describen las características de los grupos encontrados.

5.1. Tiempos de ejecución

A continuación, se detallan las variaciones en tiempos de ejecución de los algoritmos de agrupamiento disponibles en la librería MLlib de Spark, K-Means, Bisecting K-Means y Gaussian Mixture Model en función del entorno de ejecución y del tamaño de los 2 conjuntos de datos utilizados.

Para las pruebas se definió un valor de k que oscila entre 2 y 10, esta elección de valores de k se decide teniendo en cuenta el número usual seleccionado en los trabajos similares analizados en el capítulo 1 y que es un valor coherente con la cantidad de grupos que interesaría encontrar, analizando como se agrupan las 24 provincias. El valor de semilla se fijó en el valor 5380093 para poder reproducir los mismos resultados y para la evaluación se utilizó como métrica el índice Silhouette. En K-Means para la inicialización de centroides se utilizó el algoritmo K-Means|| que es el que viene configurado por defecto. Los modelos K-Means y Bisecting K-Means en las pruebas desde Colab se utilizaron con la medida de distancia Euclídea y Coseno y un número máximo de iteraciones igual a 20. Para las pruebas en el clúster de computadoras el número de iteraciones para los 3 modelos se fijó en 100 iteraciones. Las tablas con los resultados de los ensayos de detallan en el Anexo 1.1.

El gráfico 5.1 muestra la comparación de los tiempos de ejecución en segundos que demoró la creación de los 3 modelos para el conjunto Datos37 completo. El gráfico 5.2 muestra la comparación de los tiempos de ejecución en segundos que demoró la creación de los 3 modelos para el conjunto Datos13 completo. Las líneas continuas diferencian la ejecución en el clúster de la ejecución en Colab, siendo la primera notoriamente más eficiente para los 3 modelos.

Gráfico 5. 1 Comparación de tiempos de ejecución en Datos37

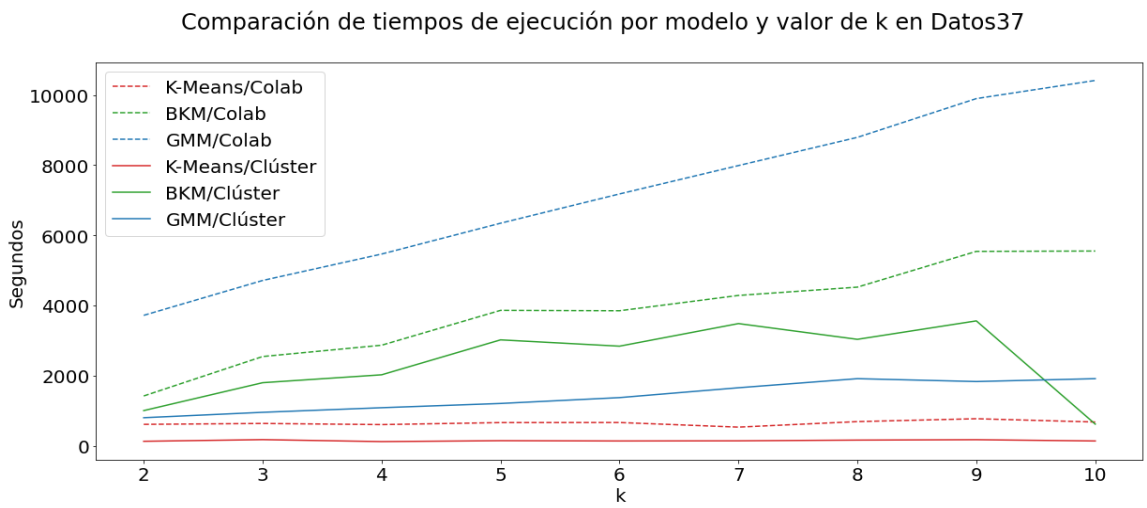
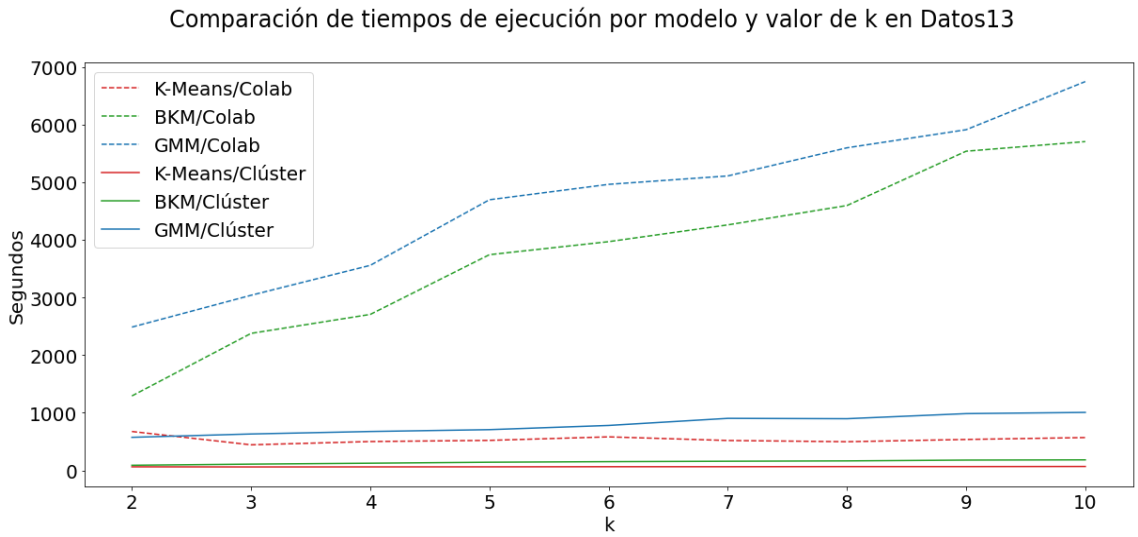


Gráfico 5. 2 Comparación de tiempos de ejecución en Datos13



El algoritmo K-Means en todos los ensayos realizados en este trabajo resultó el más eficiente en cuanto a tiempos de ejecución y el algoritmo de Mezclas Gaussianas en general el más costoso. Aunque esto no siempre es así, en el gráfico 5.1 en la ejecución en el clúster de computadoras el algoritmo Bisecting K-Means demoró más que Gaussian Mixture Model. Si bien ambos algoritmos tienen un orden de complejidad de O(n) al evaluar los tiempos de ejecución en el clúster se deben considerar distintos factores como la distribución de observaciones en los nodos, el número de procesadores y la selección de centroides iniciales entre otros. La comparación en tiempos de las pruebas realizadas se incluye en el Anexo 1.2

Por otra parte, se compararon los tiempos de ejecución de los tres modelos seleccionados en función del tamaño del conjunto de datos, para observar cómo varía el tiempo necesario para la creación del modelo y medir la escalabilidad.

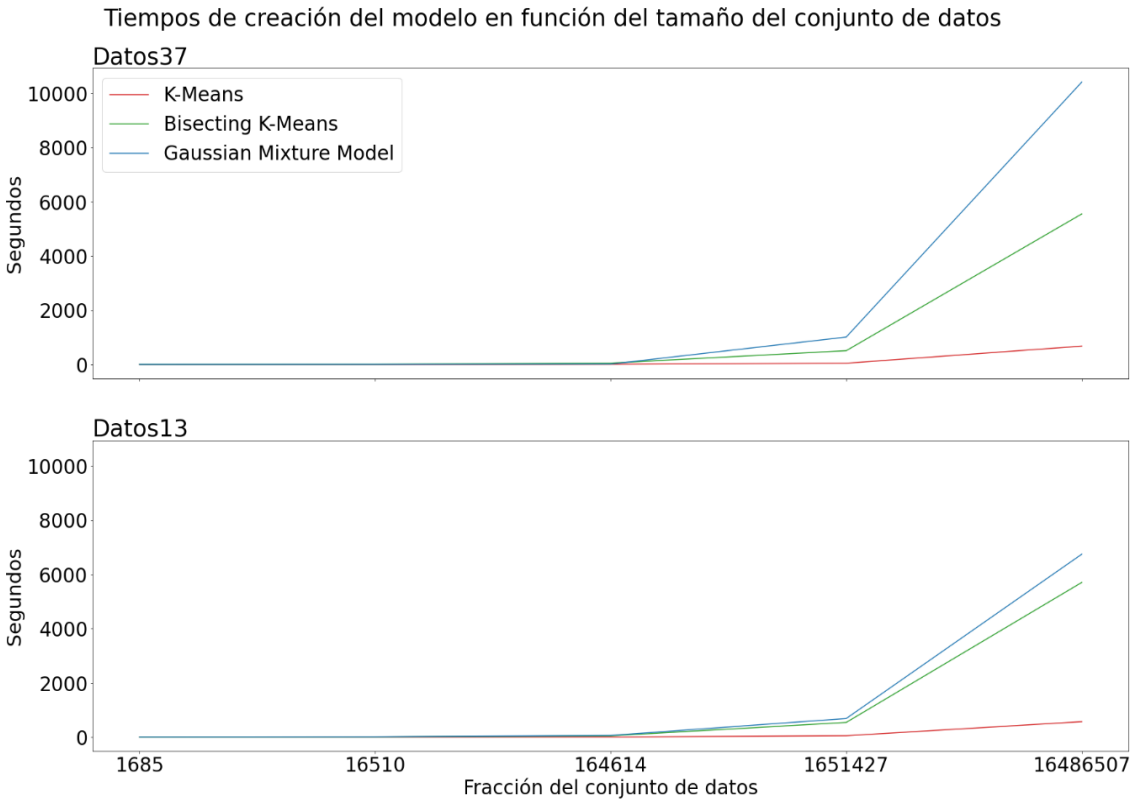
Tomando muestras aleatorias de ambos conjuntos de datos las diferencias significativas en tiempos de ejecución se observan con mayor claridad a medida que crece el número de observaciones.

El gráfico 5.3 muestra 2 gráficos de líneas donde se comparan los tiempos de ejecución de 5 fracciones tomadas de los 2 conjuntos de datos para k = 10. Estos subconjuntos se obtuvieron especificando los porcentajes 0.0001, 0.001, 0.01 y 0.1 en el método de muestreo aleatorio, obteniendo fracciones del conjunto de datos de 1685, 16510, 164614 y 1651427 muestras. El valor 16486507 corresponde al conjunto de datos completo. En el gráfico de líneas superior se observan los resultados de las fracciones tomadas de Datos37 y en el gráfico inferior los resultados correspondientes a Datos13.

Otras comparaciones correspondientes a distintos valores de k se pueden consultar en el Anexo 1.3

Una de las limitaciones del algoritmo Gaussian Mixture Model es que no escala bien cuando aumenta el número de dimensiones. Los algoritmos K-Means y Bisecting K-Means no se ven demasiado influenciados por el aumento del número de atributos, mientras que, en el caso del algoritmo de Gaussian Mixture Model, representado por las líneas azules este aumento tiene un efecto mucho más visible.

Gráfico 5. 3 Comparación de tiempos de ejecución en función del tamaño del conjunto de datos.



5.2. Validación de los clústers

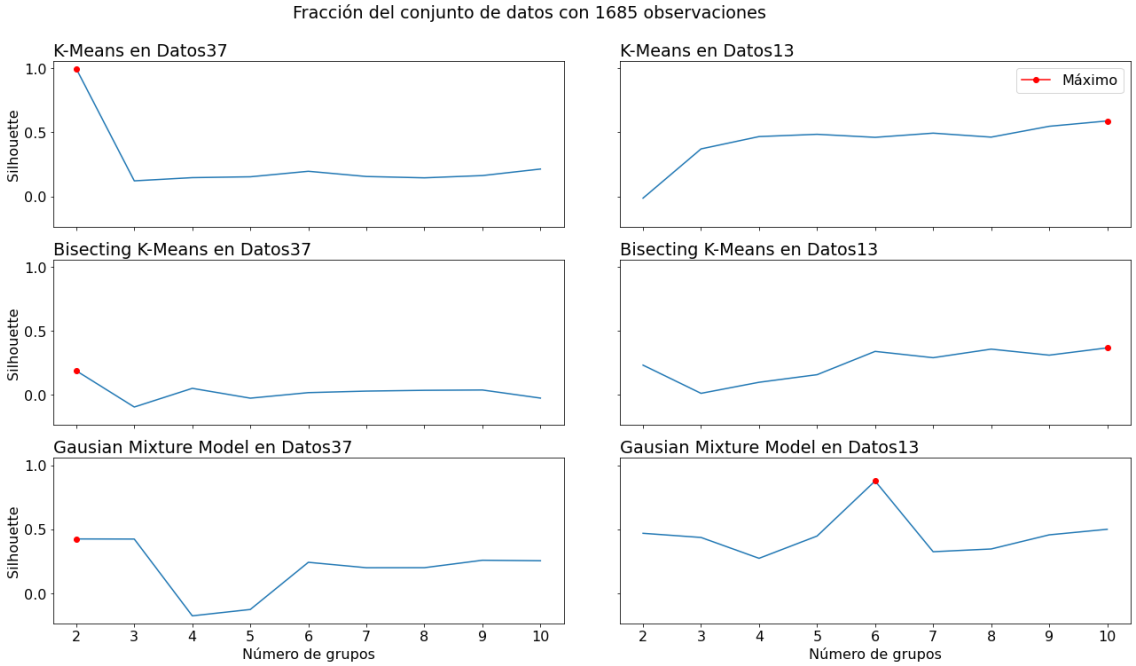
5.2.1. Índice Silhouette

El evaluador de resultados de agrupaciones, ClusteringEvaluator() de MLlib calcula el índice Silhouette permitiendo configurar 2 posibles métricas de distancia, la distancia euclidiana al cuadrado o la distancia coseno. Este evaluador espera dos columnas de entrada, la columna predicción con los resultados del entrenamiento del modelo y la columna de características con los atributos escalados.

El gráfico 5.4 muestra los coeficientes del índice Silhouette según el valor de k (entre 2 y 10) para los modelos K-Means, Bisecting K-Means y Gaussian Mixture Model, sobre una fracción de 0.0001 del conjunto de datos, considerando que un mayor valor indicará una mejor calidad de un resultado de agrupamiento.

En la columna de la izquierda de la matriz de gráficos de líneas se observan los coeficientes obtenidos utilizando como parámetro del modelo el vector de características que incluye las 24 provincias. La columna de la derecha muestra los coeficientes obtenidos al recibir un vector de 13 características. Las filas representan los modelos y el punto rojo destaca el número de agrupaciones (k) con mayor coeficiente Silhouette.

Gráfico 5. 4 Variación del Índice Silhouette para la fracción de 0.0001 observaciones



Los gráficos 5.5, 5.6, 5.7 y 5.8 muestran los resultados para fracciones del conjunto de datos de 0.001, 0.01, 0.1 y conjunto de datos completo respectivamente.

Gráfico 5. 5 Variación del Índice Silhouette para la fracción de 0.001 observaciones

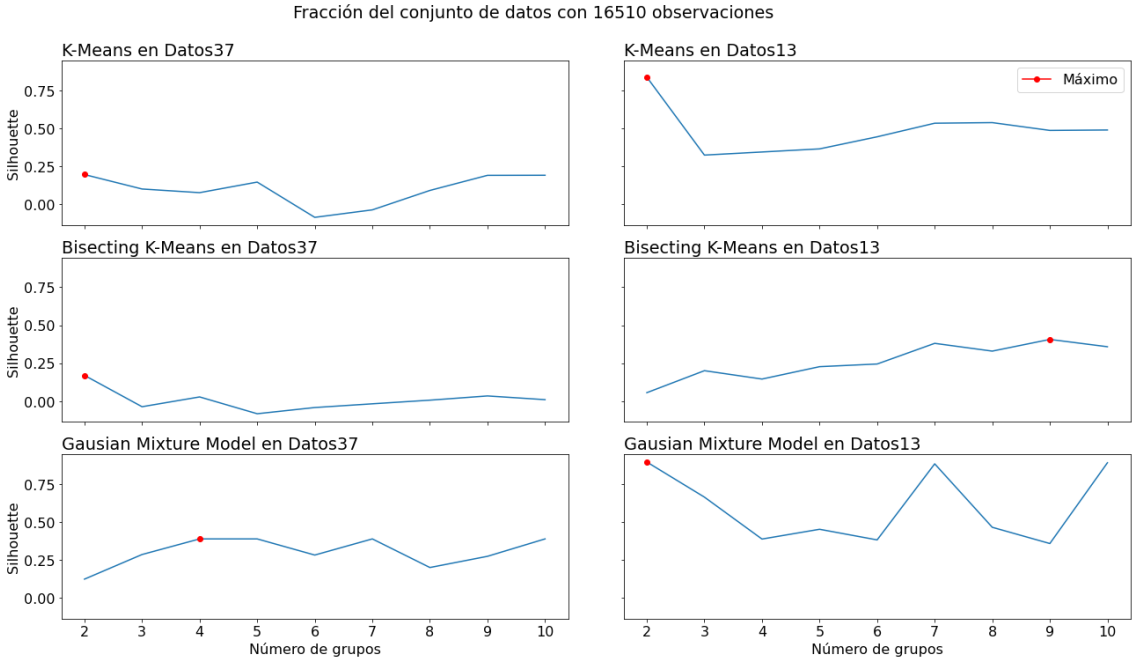


Gráfico 5. 6 Variación del Índice Silhouette para la fracción de 0.01 observaciones

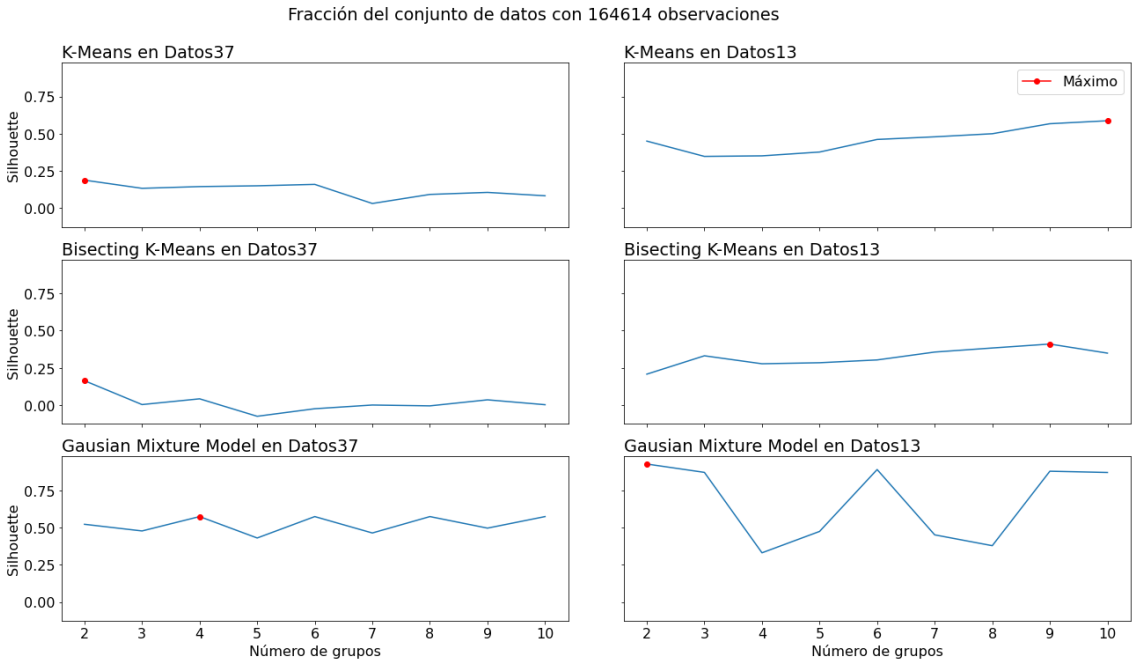


Gráfico 5. 7 Variación del Índice Silhouette para la fracción de 0.1 observaciones

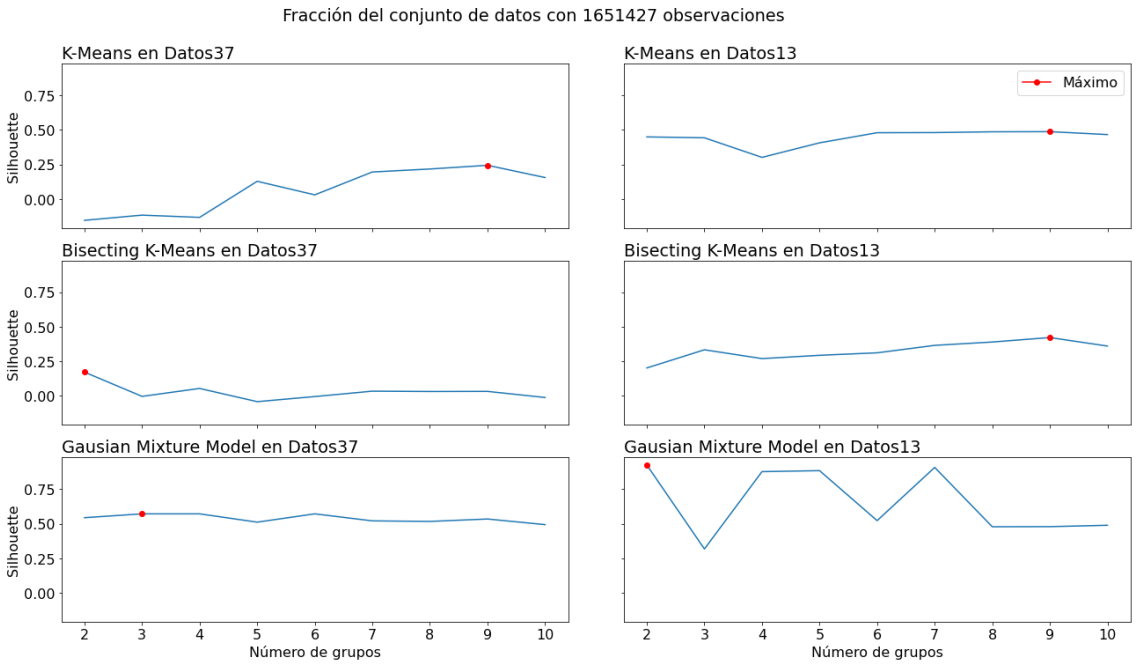
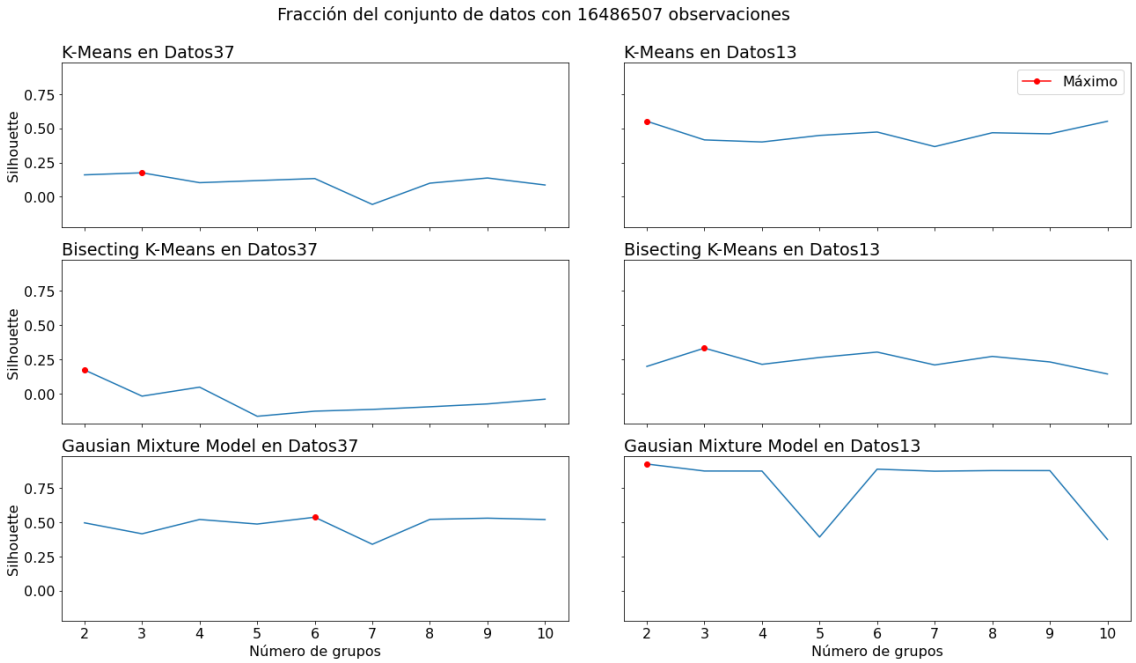
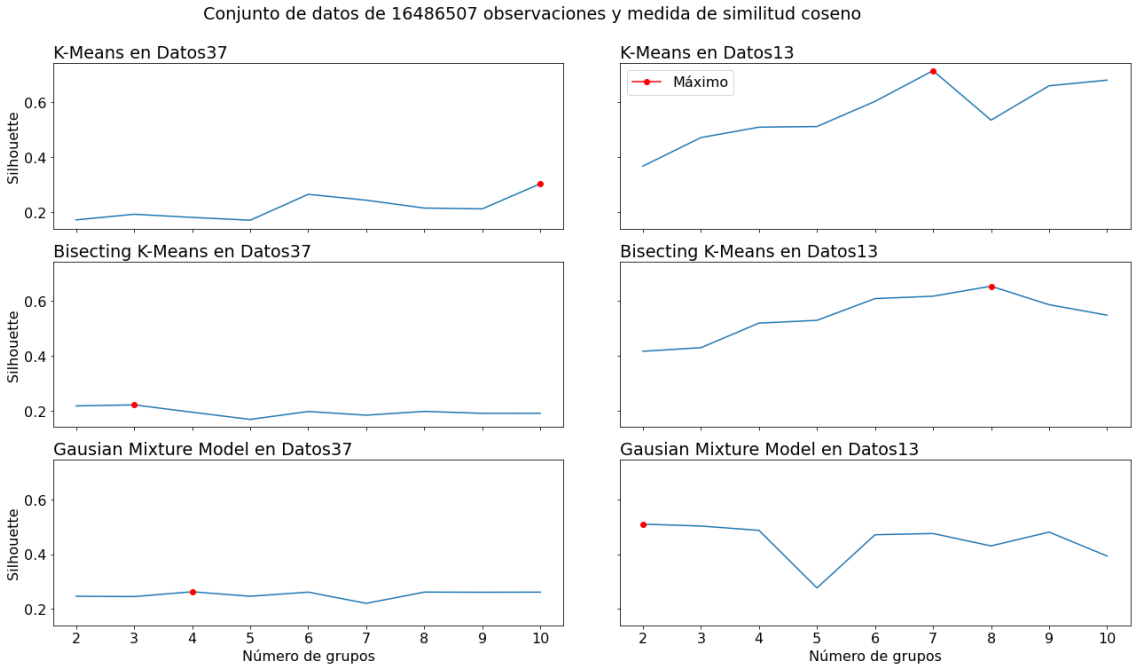


Gráfico 5. 8 Variación del Índice Silhouette para el conjunto de datos completo



El gráfico 5.9 muestra los resultados de la ejecución de los tres modelos configurados con métrica de distancia coseno para el conjunto de datos completo.

Gráfico 5. 9 Variación del Índice Silhouette para el conjunto de datos completo con medida de similitud coseno



5.2.1.1. Selección del número de agrupaciones según el índice Silhouette

La tabla 5.1 muestra los números de grupos (k) que obtuvieron un mayor coeficiente Silhouette en las ejecuciones de los modelos sobre Datos37 completo y fraccionado desde Colab y desde el clúster de computadoras. La tabla 5.2 muestra la misma información para el conjunto de datos Datos13. Los valores más altos del índice Silhouette se observan para k = 2 en K-Means sobre Datos37 y en Gaussian Mixture Model sobre Datos13.

La medida de distancia utilizada por defecto es la distancia euclidiana, en los ensayos en los que se utilizó medida del coseno se especifica explícitamente.

Tabla 5. 1 Mayores resultados del Índice Silhouette en Datos37

Tamaño en cantidad de casos	Conjunto de datos con 37 atributos					
	K-Means		Bisecting K-Means		Gaussian Mixture Model	
	k	Silhouette	k	Silhouette	k	Silhouette
1685	**2**	**0.99**	2	0.18	2	0.42
16510	2	0.19	2	0.16	4	0.38
164614	2	0.18	2	0.16	4	0.57
1651427	9	0.24	2	0.17	3	0.57
16486507	3	0.17	2	0.17	6	0.53
16486550 (clúster)	10	0.20	2	0.17	4	0.54
16486507 (coseno)	10	0.30	3	0.22	4	0.26

Tabla 5. 2 Mayores resultados del Índice Silhouette en Datos 13

Tamaño en cantidad de casos	Conjunto de datos con 13 atributos					
	K-Means		Bisecting K-Means		Gaussian Mixture Model	
	k	Silhouette	k	Silhouette	k	Silhouette
1685	10	0.58	10	0.36	6	0.87
16510	2	0.83	9	0.40	2	0.89
164614	10	0.58	9	0.41	**2**	**0.92**
1651427	9	0.48	9	0.42	**2**	**0.92**
16486507	2	0.55	3	0.33	**2**	**0.92**
16486550 (clúster)	10	0.61	3	0.33	**2**	**0.92**
16486507 (coseno)	7	0.71	8	0.65	2	0.50

5.2.2. Matriz de evidencia.

La matriz de evidencia se construye creando una matriz que tiene una fila y una columna para cada muestra y contabilizando en la entrada correspondiente al par de muestras las veces que coinciden en el mismo clúster, independientemente del modelo y de la etiqueta asignada.

Se dispone de los resultados de las ejecuciones de los 3 modelos sobre Datos37 y Datos13 almacenados en 36 archivos parquet en Google Drive. En cada archivo se guardó el id y el resultado de la predicción de los 3 modelos para cada caso. La tabla 5.3 muestra

un ejemplo de 5 casos correspondientes al archivo que almacena los resultados de la ejecución de los modelos configurados con medida de similitud coseno para k = 2 sobre Datos37

Tabla 5. 3 Resultados para Datos37, medida de similitud coseno, k = 2

Id_evento_caso	predKM	predBKM	predGMM
763429	Clúster 0	Clúster 1	Clúster 0
776683	Clúster 1	Clúster 0	Clúster 1
784302	Clúster 1	Clúster 0	Clúster 0
784915	Clúster 1	Clúster 0	Clúster 0
798754	Clúster 1	Clúster 0	Clúster 0

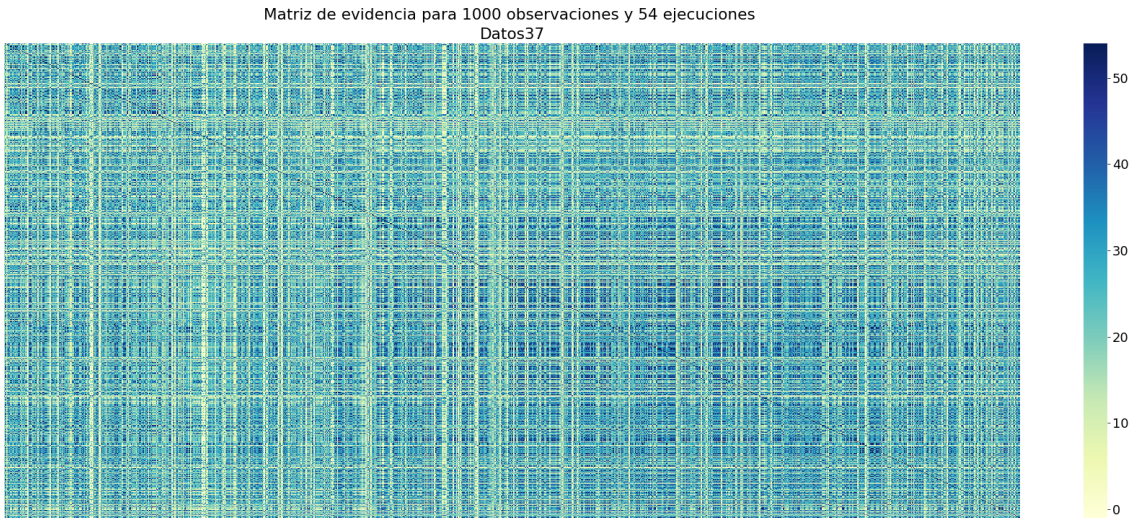
Los resultados de las ejecuciones se almacenaron en archivos separados por las medidas de distancia euclídea o coseno, los conjuntos de datos (Datos37 y Datos13) y los 9 valores utilizados para configurar el parámetro k.

Para generar la matriz de evidencia se tomó una muestra aleatoria de 1000 observaciones del archivo de predicciones de Datos37 para k = 2 y luego se recuperaron los mismos casos (por id_evento_caso) de los 17 archivos restantes.

Los 18 archivos se recorrieron contabilizando en cada entrada de la matriz y por cada modelo las veces en las que el par de casos tuvo la misma etiqueta de clúster.

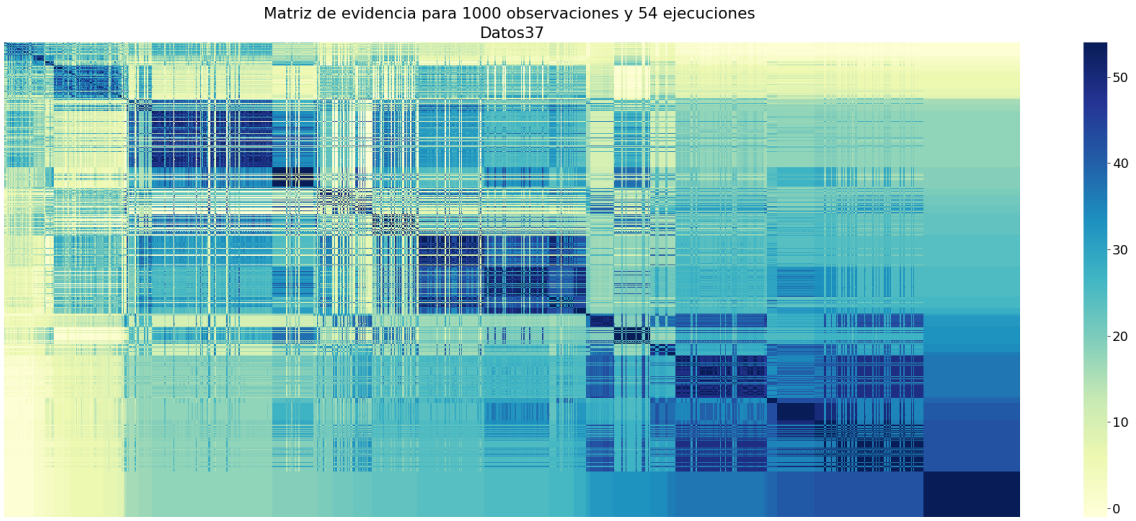
El gráfico 5.10 muestra la matriz de evidencia con los resultados de las 54 ejecuciones sobre Datos37 para 9 valores de k y 2 medidas de similitud.

Gráfico 5. 10 Matriz de evidencia sobre Datos37



En el gráfico 5.11 se muestra la matriz de evidencia sobre Datos37 ordenada por los contadores, tomando los valores de los ejes x e y mayores.

Gráfico 5. 11 Matriz de evidencia sobre Datos37 ordenada



Para generar la matriz de evidencia correspondiente a Datos13 se recuperaron de los 18 archivos correspondientes a los mismos casos que los utilizados para Datos37.

Los 18 archivos se recorrieron contabilizando en cada entrada de la matriz y por cada modelo las veces en las que el par de casos fueron etiquetados igual.

El gráfico 5.12 muestra la matriz de evidencia con los resultados de las 54 ejecuciones sobre Datos13 para 9 valores de k y 2 medidas de similitud. El gráfico 5.13 muestra la matriz de evidencia ordenada por los contadores.

Gráfico 5. 12 Matriz de evidencia sobre Datos13

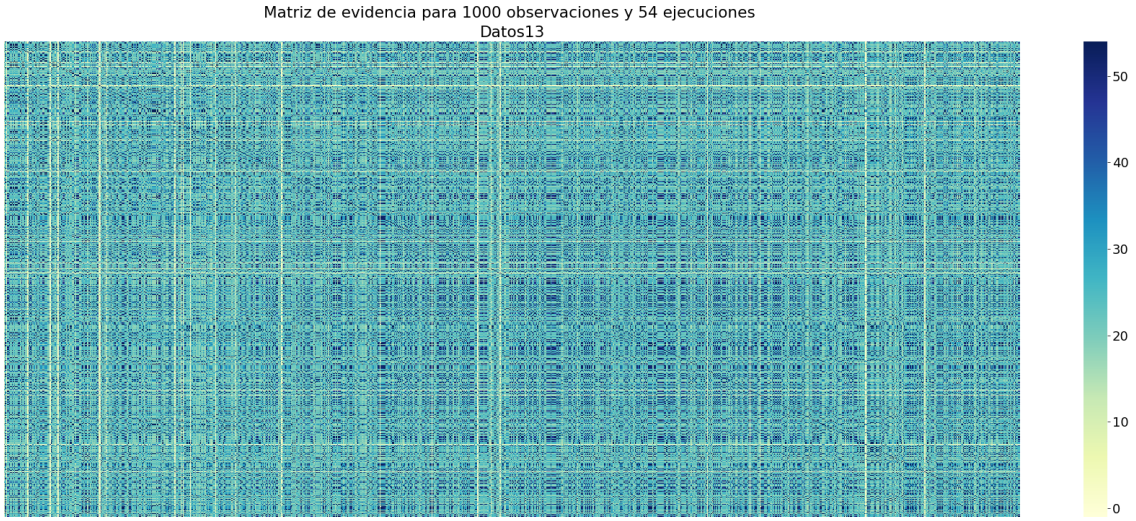
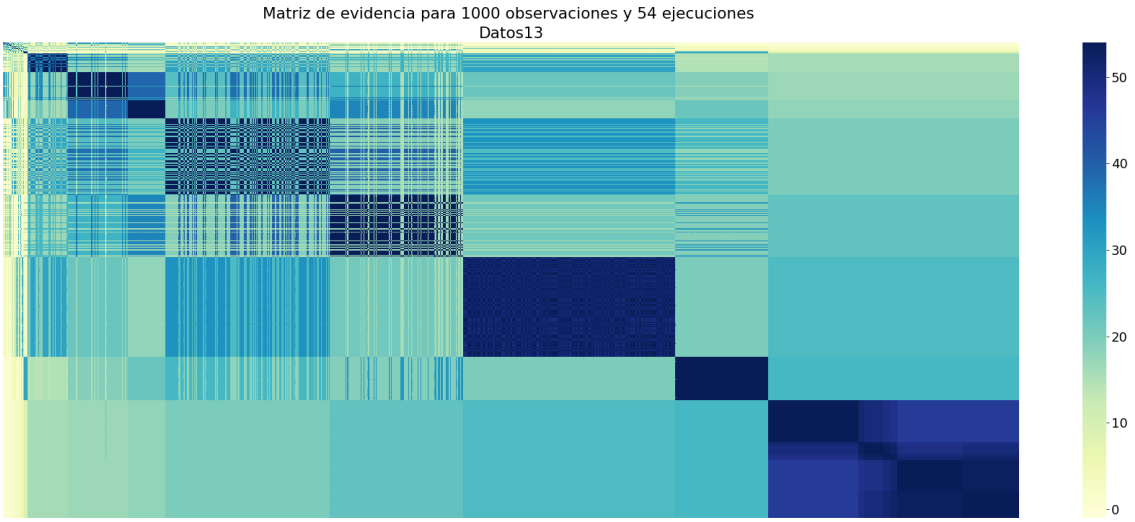


Gráfico 5. 13 Matriz de evidencias sobre Datos13 ordenada



Estas fracciones de 1000 muestras representan el 0.006 % del tamaño del conjunto de datos que se busca representar.

5.2.3. Matriz de similitud ideal

Para armar la matriz de similitud ideal se utilizó una fracción del conjunto de datos de 100 muestras almacenando en vectores la columna de identificación de cada caso (id_evento_caso) y la predicción del modelo correspondiente. Para k=2 los valores posibles de predicción serán 0 y 1, por este motivo se inicializó la matriz con algún valor distinto a estos valores, por ejemplo -1 y luego en los pares de puntos clasificados igual se reemplazó el valor de inicialización por el asignado por el modelo. El paso siguiente fue ordenar agrupando los casos por valor de predicción. En teoría, si tenemos agrupamientos bien separados, entonces la matriz debería mostrar una diagonal de bloques. De lo contrario, los patrones que se muestran en la matriz pueden revelar las relaciones entre los agrupamientos.

En los gráficos 5.14, 5.15 y 5.16 se muestran las matrices de similitud ideal para los resultados de los modelos configurados con k igual a 2

Gráfico 5. 14 Matriz de similitud ideal para k = 2 modelo Gaussian Mixture Model

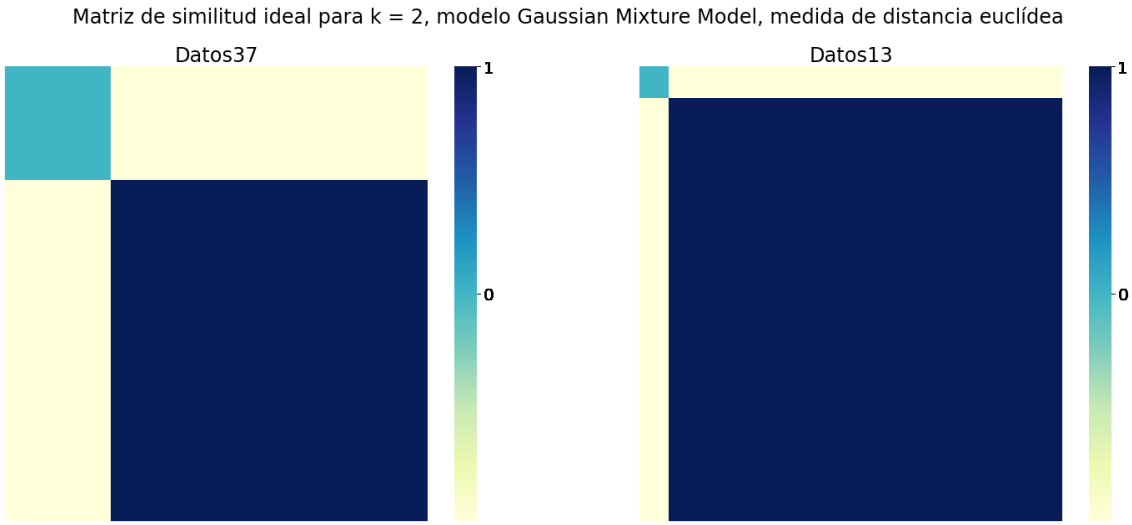


Gráfico 5. 15 Matriz de similitud ideal para k = 2 modelo K-Means

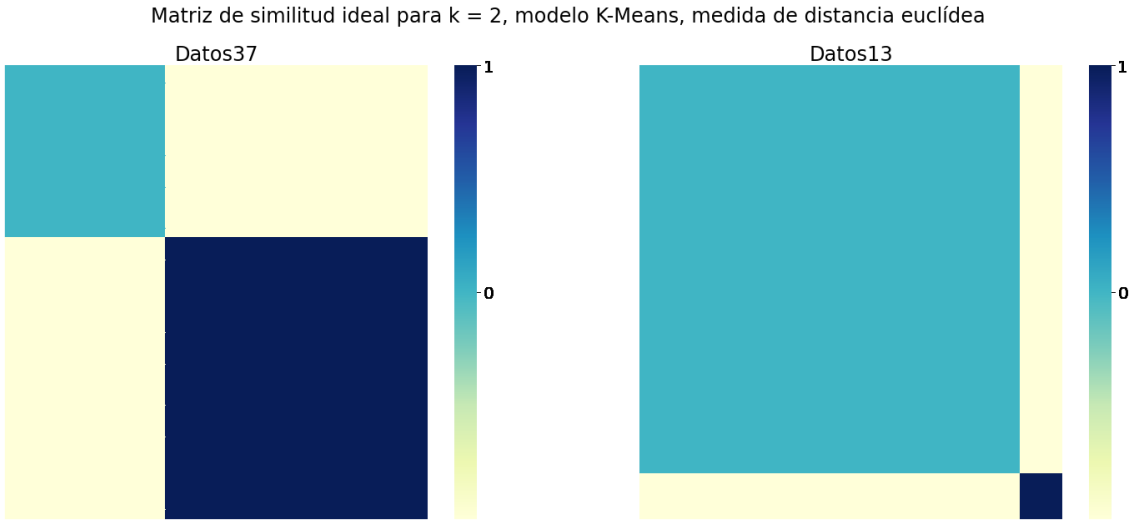
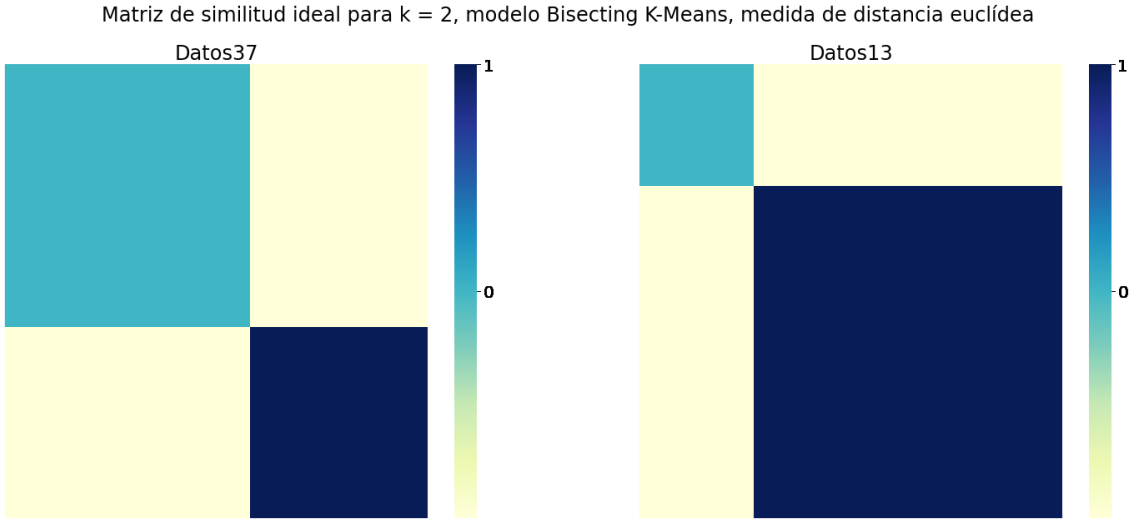


Gráfico 5. 16 Matriz de similitud ideal para k = 2 modelo Bisecting K-Means



5.2.4. Coincidencia en la clasificación de casos por los 3 modelos

Otro análisis llevado a cabo es el de comprobar las diferencias de los resultados obtenidos por las tres técnicas estudiadas. Como la cantidad de clústeres es la misma para todas las técnicas lo que se midió fue las diferencias entre los distintos agrupamientos obtenidos. Las diferencias entre los grupos son aquellas muestras que resultan ubicadas en diferentes clústeres. El objetivo de esta comparación es determinar cuántas muestras del conjunto de datos son ubicadas en el mismo clúster por las tres técnicas, brindando así una cierta confianza en los resultados.

Sobre los 36 archivos descriptos en 5.2.2 se aplicó la función listaFrecuenciasNum que clasificó las coincidencias de las predicciones asignadas por los modelos con la finalidad de representar gráficamente estos resultados.

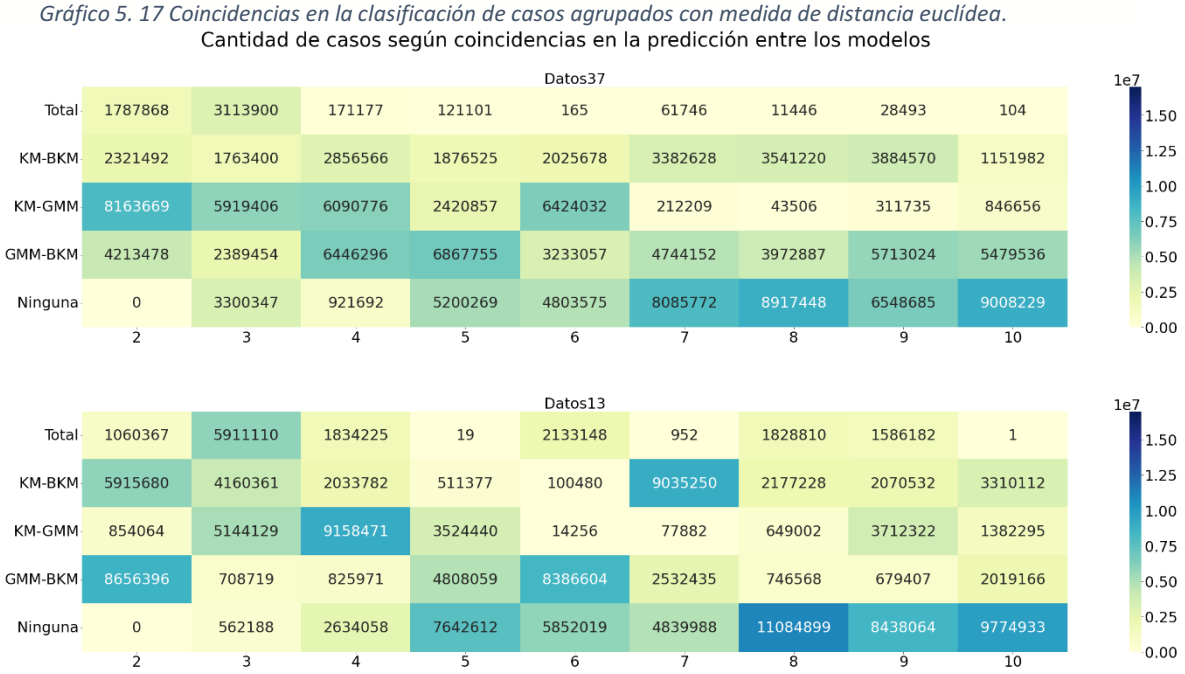
df.rdd.map(lambda t: (listafrecuenciasnum(t)))

La función listaFrecuenciasNum asigna el valor 0 a los id_evento_caso en los que los tres modelos coincidieron en el grupo asignado, el valor 1 si coincidían K-Means y Bisecting K-Means, el valor 2 para las coincidencias entre K-Means y Gaussian Mixture Model, el valor 3 para igual resultado entre Gaussian Mixture Model y Bisecting K-Means y finalmente el valor 4 para los casos sin coincidencias.

Esta clasificación busca representar la cantidad de casos que fueron asignados al mismo grupo, teniendo en cuenta que las agrupaciones encontradas por los modelos pueden incluir atributos distintos.

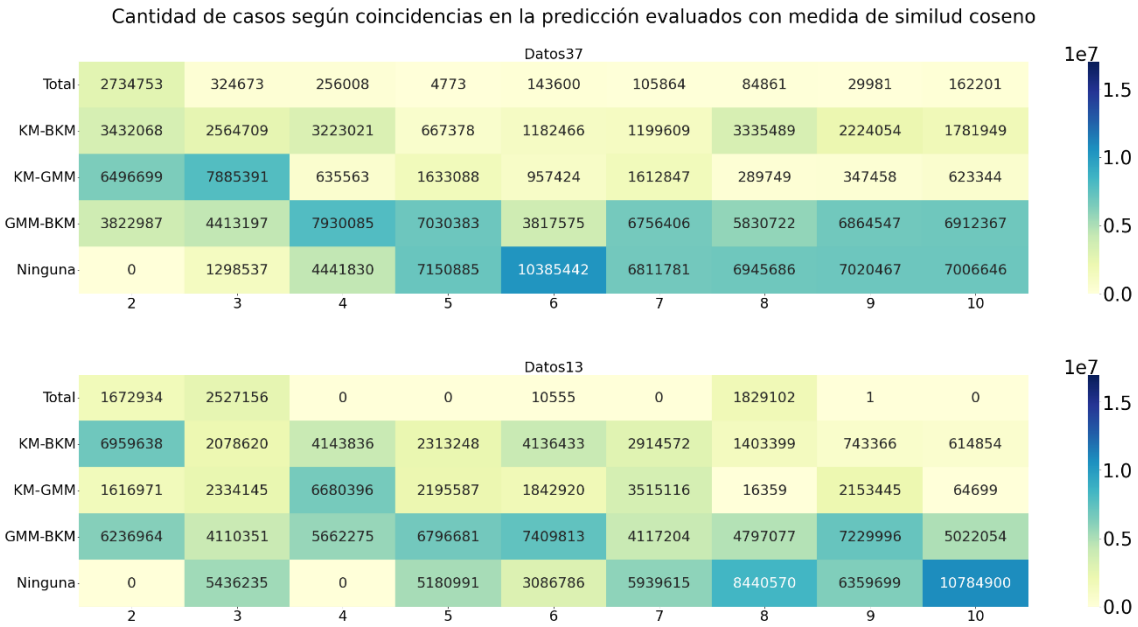
El gráfico 5.17 muestra en el mapa de calor superior la coincidencia en la clasificación de los 3 modelos representando en las filas la coincidencia y en las columnas el valor de k. La medida utilizada fue distancia euclídea. La misma información se muestra para Datos13 en el mapa de calor inferior.

Gráfico 5. 17 Coincidencias en la clasificación de casos agrupados con medida de distancia euclídea.



El gráfico 5.18 muestra en el mapa de calor superior la coincidencia en la clasificación de los 3 modelos representando en las filas la coincidencia y en las columnas el valor de k. La medida utilizada fue distancia coseno. La misma información se muestra para Datos13 en el mapa de calor inferior.

Gráfico 5. 18 Coincidencias en la clasificación de casos agrupados con medida de similitud coseno



Un mapa de calor es una técnica de visualización de datos que muestra la magnitud de un fenómeno como color en dos dimensiones. Son muy útiles cuando se tienen conjuntos de datos muy grandes ya que permiten representar la densidad de elementos con la variación en el color, dando señales visuales obvias sobre cómo el fenómeno se agrupa.

Los resultados obtenidos para el conjunto Datos37 luego de las ejecuciones de los modelos parametrizados con medida de distancia euclídea muestran el mayor número de casos para k = 10 cuando no coinciden los modelos y un alto valor de coincidencias para k = 2 en los modelos K-Means y Gaussian Mixture Model.

Los resultados obtenidos para el conjunto Datos13 luego de las ejecuciones de los modelos parametrizados con medida de distancia euclídea muestran la mayor cantidad de casos en k = 8 cuando no coinciden los modelos, y un alto valor para k = 4 también en los modelos K-Means y Gaussian Mixture Model.

Los resultados obtenidos luego de las ejecuciones de los modelos parametrizados con medida de distancia coseno muestran para Datos37 el mayor número de casos cuando no coinciden los modelos para k = 6 y un alto valor para k = 4 en Bisecting K-Means y Gaussian Mixture Model. Para Datos13 la mayor cantidad de casos ocurre en k = 10 cuando no coinciden los modelos, y un alto valor para k = 6 también en los modelos Bisecting K-Means y Gaussian Mixture Model.

Para visualizar estas distribuciones se armaron 18 vectores por medida de distancia con las coincidencias entre las predicciones obtenidas utilizando Spark SQL y las librerías Numpy y Matplotlib.

En el histograma 5.19 se visualizan las coincidencias en cantidad de casos entre los conjuntos Datos37 y Datos13 en cada valor correspondiente al número de grupos (k). La medida de distancia parametrizada fue la euclídea. El eje y muestra la cantidad de casos (0 a 1000000), el eje x los valores 0 a 4 para coincidencia total, K-Means y Bisecting K-Means, K-Means y Gaussian Mixture Model, Gaussian Mixture Model y Bisecting K-Means y ninguna respectivamente. Las escalas de los ejes x e y se unificaron para facilitar la comparación visual.

En el histograma 5.20 se visualizan las coincidencias en cantidad de casos entre los conjuntos Datos37 y Datos13 en cada valor de k con las mismas consideraciones que en el gráfico anterior pero la medida de distancia parametrizada fue coseno.

Gráfico 5. 19 Cantidad de casos agrupados por coincidencias en la clasificación de los modelos parametrizados con distancia euclídea y cantidad de grupos (K) en Datos37 y Datos13

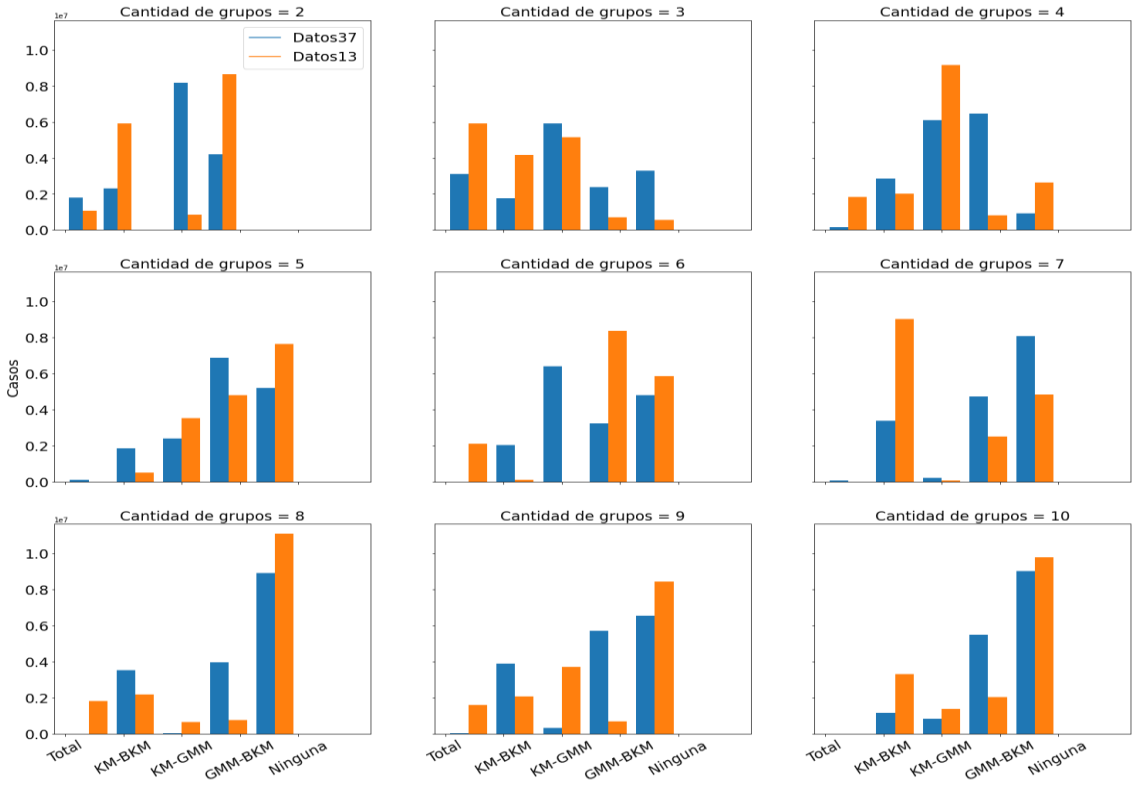
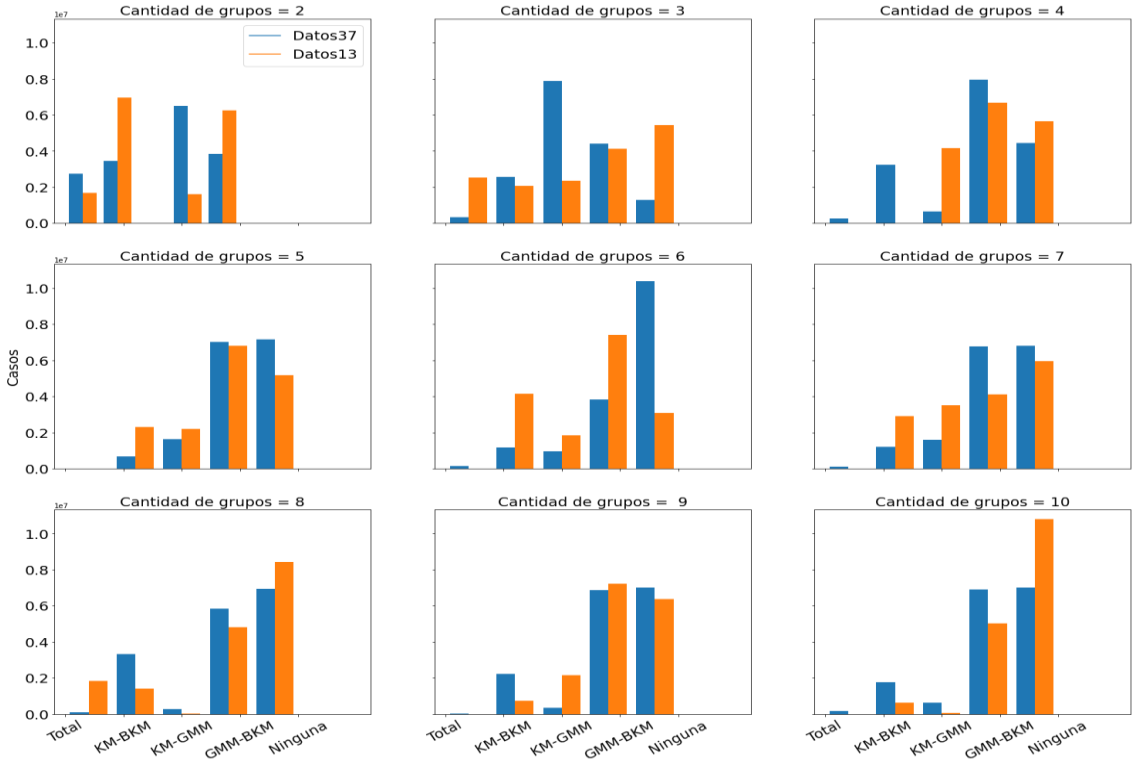


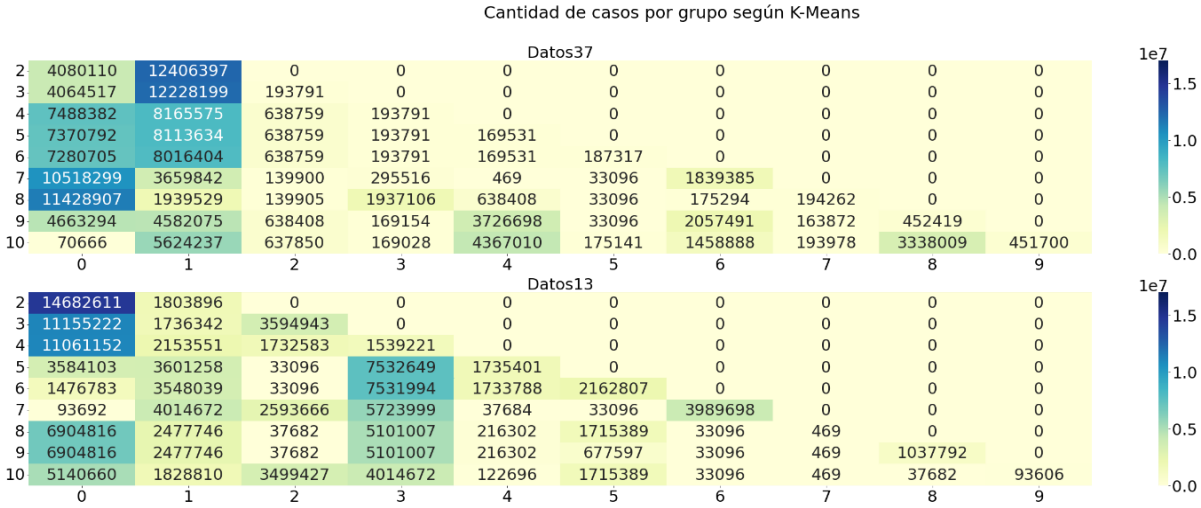
Gráfico 5. 20 Cantidad de casos agrupados por coincidencias en la clasificación de los modelos parametrizados con medida de distancia coseno y cantidad de grupos (K) en Datos37 y Datos13



5.2.5. Distribución de casos por modelo, agrupación y predicción.

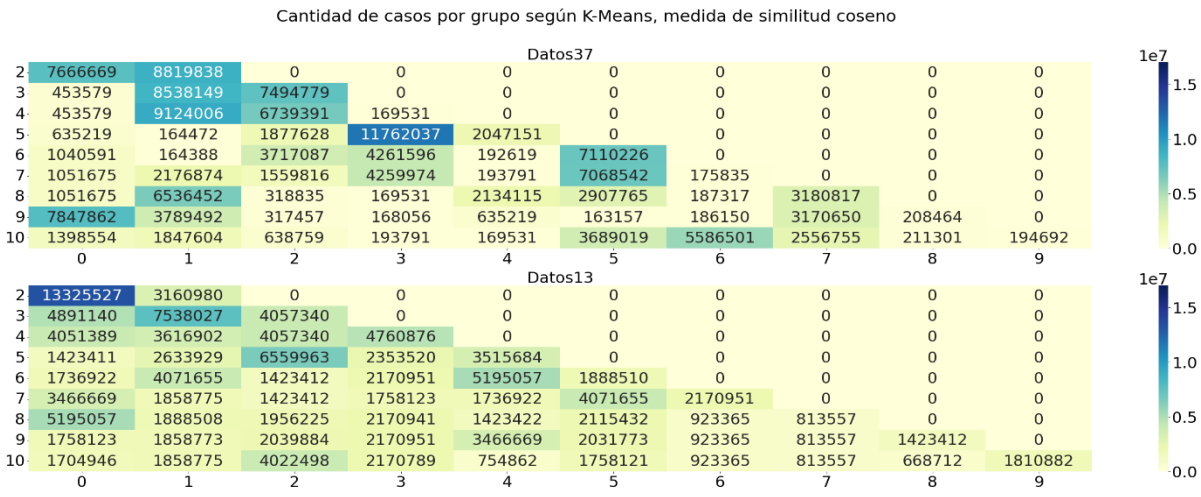
El gráfico 5.21 muestra en el mapa de calor superior la cantidad de casos indicando en cada fila el número de agrupaciones con el que fue configurado el conjunto de datos procesado y en cada columna la predicción del modelo K-Means. La medida de distancia parametrizada fue euclídea. En el mapa de calor inferior se muestra la misma información para Datos13.

Gráfico 5. 21 Cantidad de casos por valor de k y etiqueta de clúster según K-Means distancia euclídea



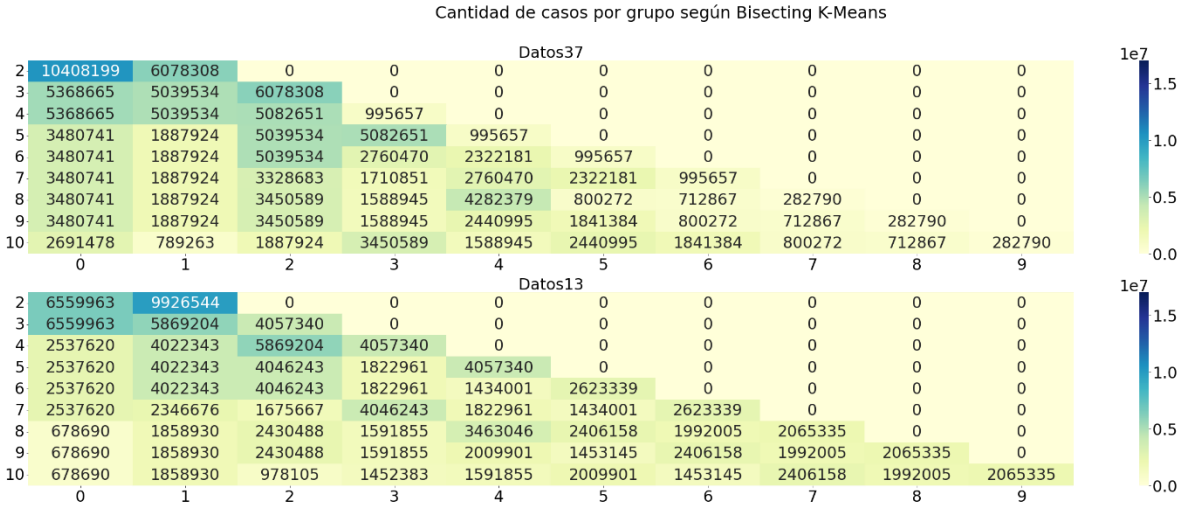
El gráfico 5.22 muestra en el mapa de calor superior la cantidad de casos indicando en cada fila el número de agrupaciones con el que fue configurado el conjunto de datos procesado y en cada columna la predicción del modelo K-Means. La medida de distancia parametrizada fue coseno. En el mapa de calor inferior se muestra la misma información para Datos13.

Gráfico 5. 22 Cantidad de casos por valor de k y etiqueta de clúster según K-Means distancia coseno



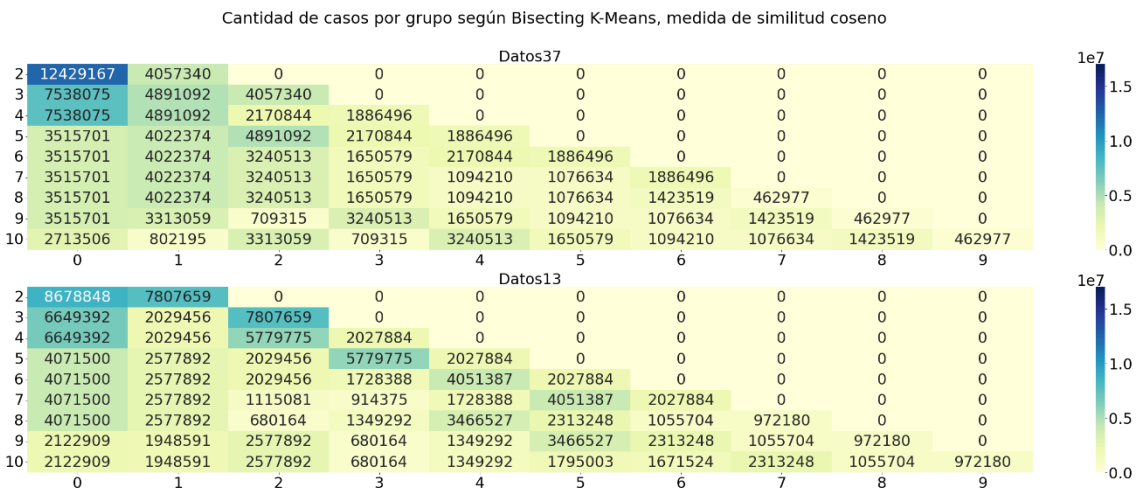
El gráfico 5.23 muestra en el mapa de calor superior la cantidad de casos indicando en cada fila el número de agrupaciones con el que fue configurado el conjunto de datos procesado y en cada columna la predicción del modelo Bisecting K-Means. La medida de distancia parametrizada fue euclídea. En el mapa de calor inferior se muestra la misma información para Datos13.

Gráfico 5. 23 Cantidad de casos por valor de k y etiqueta de clúster según Bisecting K-Means distancia euclídea



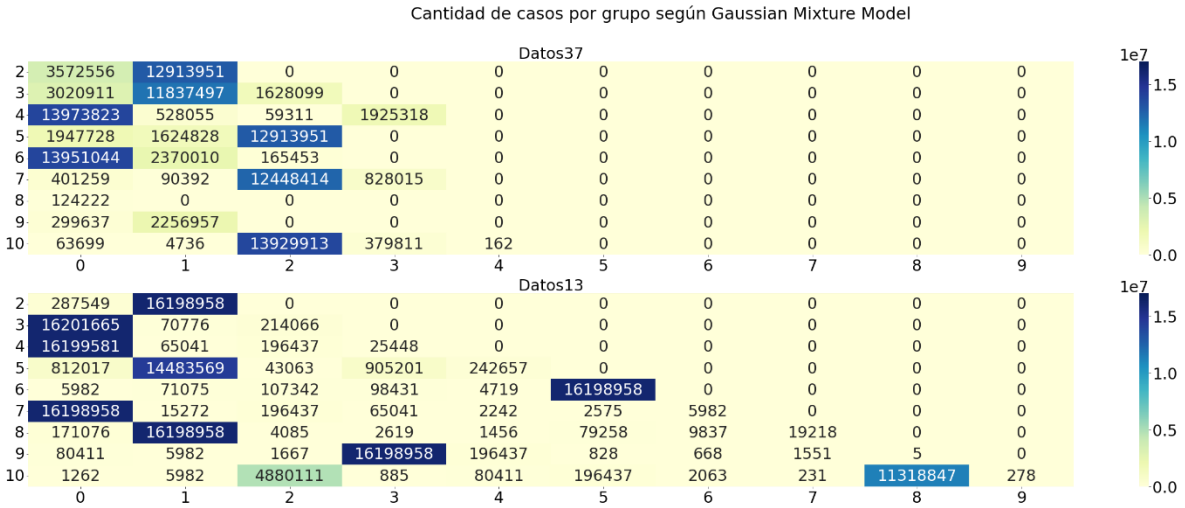
El gráfico 5.24 muestra en el mapa de calor superior la cantidad de casos indicando en cada fila el número de agrupaciones con el que fue configurado el conjunto de datos procesado y en cada columna la predicción del modelo Bisecting K-Means. La medida de distancia parametrizada fue coseno. En el mapa de calor inferior se muestra la misma información para Datos13.

Gráfico 5. 24 Cantidad de casos por valor de k y etiqueta de clúster según Bisecting K-Means distancia coseno



El gráfico 5.25 muestra en el mapa de calor superior la cantidad de casos indicando en cada fila el número de agrupaciones con el que fue configurado el conjunto de datos procesado y en cada columna la predicción del modelo Gaussian Mixture Model. En el mapa de calor inferior se muestra la misma información para Datos13.

Gráfico 5. 25 Cantidad de casos por valor de k y etiqueta de clúster según Gaussian Mixture Model



Los mapas de calor permiten visualizar fácilmente el tamaño de los grupos encontrados para cada valor de k.

5.3. Análisis de agrupamientos

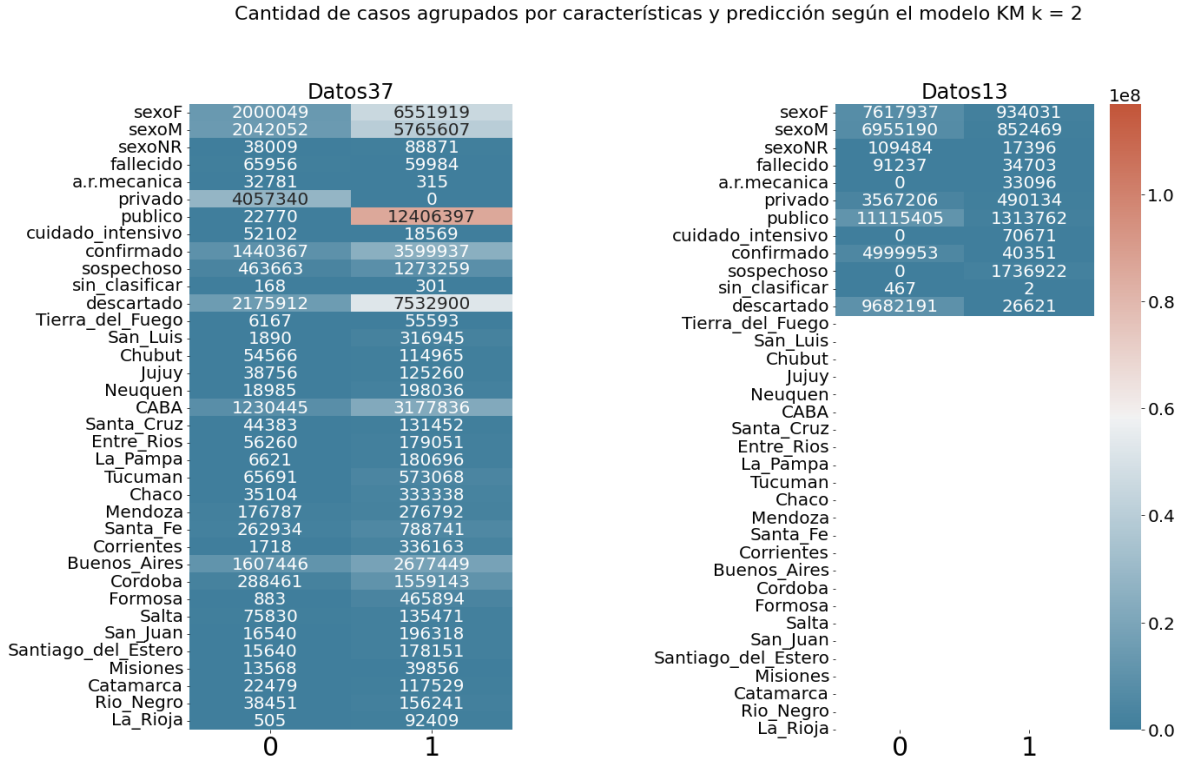
Continuando con la idea de coincidencia en la clasificación de los grupos encontrados por los modelos. ¿El valor de predicción asignado por los algoritmos, tiene los mismos atributos en los 3 modelos?

Para responder a esta pregunta en esta sección se analizará cómo quedan armados los grupos en función de los atributos. El valor de k seleccionado para este análisis corresponde al de mayor valor obtenido por el índice Silhouette al evaluar los agrupamientos, sólo se realizó el análisis para k igual a 2 ya que el análisis de los 9 valores de k sería demasiado extenso y los resultados arrojados por k = 2 ya dan una idea de la varianza de los resultados obtenidos por los tres modelos.

5.3.1. Distribución de los casos por atributo

El gráfico 5.26 muestra en el mapa de calor de la izquierda la cantidad de casos agrupados por atributos y predicción del modelo K-Means sobre Datos37. El mapa de calor de la derecha muestra la cantidad de casos agrupados por características y predicción del modelo K-Means sobre Datos13.

Gráfico 5. 26 Casos agrupados por características y predicción según K-Means

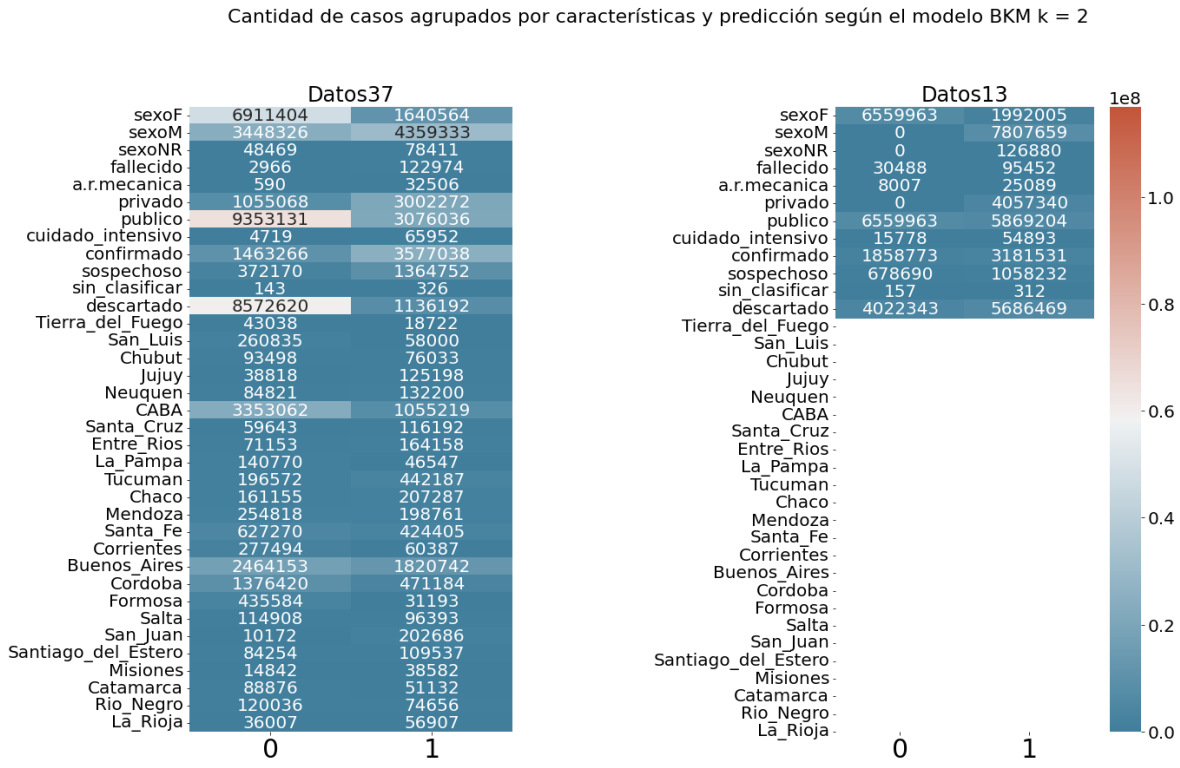


Analizando los atributos en que todos los casos quedaron etiquetados igual se observa en Datos37 que todos los casos con origen de financiamiento privado quedan en el grupo

0. Mirando Datos13 se puede ver que todos los casos que recibieron asistencia respiratoria mecánica, cuidado intensivo o fueron clasificados como caso sospechoso quedan en el grupo 1. Como vimos en el gráfico 5.8 el valor del índice Silhouette para Datos37 cuando k es igual a 2 fue menor a 0.25, un valor bajo que indica que la distancia entre los grupos no es significativa. En Datos13 para k igual a 2 el valor del índice Silhouette fue de 0.55 y resultó el mejor valor de k entre 2 y 10.

El gráfico 5.27 muestra en el mapa de calor de la izquierda la cantidad de casos agrupados por atributos y predicción del modelo Bisecting K-Means sobre Datos37. El mapa de calor de la derecha muestra la cantidad de casos agrupados por características y predicción del modelo K-Means sobre Datos13.

Gráfico 5. 27 Casos agrupados por características y predicción según BK-Means

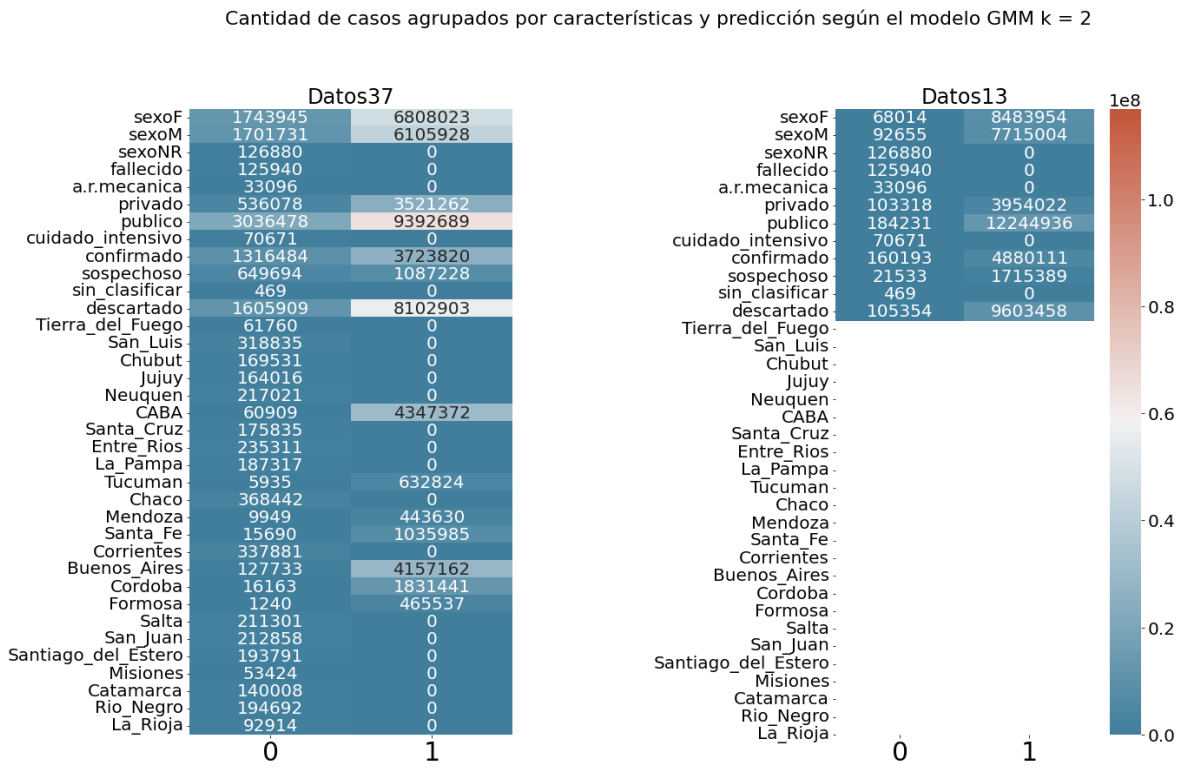


El algoritmo Bisecting K-Means para k = 2 agrupo una gran cantidad de casos ocurridos en mujeres, con origen de financiamiento público, clasificados como descartados, registrados en CABA y Buenos Aires en el grupo 0. En Datos13 en el grupo 1 se agruparon todos los casos correspondientes a hombres o sexo no registrado y origen de financiamiento privado. El valor del índice Silhouette para los 2 conjuntos de datos fue muy bajo, menor a 0.2 indicando una distancia entre los grupos poco significativa.

El gráfico 5.28 muestra en el mapa de calor de la izquierda la cantidad de casos agrupados por atributos y predicción del modelo Gaussian Mixture Model sobre Datos37.

El mapa de calor de la derecha muestra la cantidad de casos agrupados por características y predicción del modelo Gaussian Mixture Model sobre Datos13.

Gráfico 5. 28 Casos agrupados por características y predicción según GMM



Este algoritmo para los 2 conjuntos de datos agrupa todos los casos sin clasificar, de sexo no registrado, fallecidos, que recibieron asistencia respiratoria mecánica y cuidado intensivo en el grupo 0. En Datos37 incluye todos los casos de las provincias Tierra del Fuego, San Luis, Chubut, Jujuy, Neuquén, Santa Cruz, Entre Ríos, La Pampa, Chaco, Corrientes, Salta, San Juan, Santiago del Estero, Misiones, Catamarca, Río Negro y La Rioja, quedando en el grupo 1 la mayor parte de los casos de CABA, Tucumán, Mendoza, Santa Fe, Buenos Aires, Córdoba y Formosa.

El valor del índice Silhouette para Datos37 fue igual a 0.49 y a 0.92 en Datos13, explicando una mejor separación de los casos.

En términos de SQL los grupos encontrados por Gaussian Mixture Model en Datos13 etiquetados como 1 equivalen a la consulta

```
SELECT * FROM Datos  WHERE sexoNR = 0
AND fallecido = 0  AND cuidado_intensivo = 0
AND asistencia_respiratoria_mecanica = 0 AND  sin_clasificar = 0
```

donde tabla es una TempView creada en PySpark con las predicciones generadas por Gaussian Mixture Model.

5.3.1.1. Distribución del atributo edad

La edad es el único atributo del conjunto de datos incluido en el vector de características sin transformar antes de escalar los datos. En el histograma 5.29 se muestra la distribución de la edad para los 3 modelos ejecutados en Datos37 y Datos13 con medida de distancia euclídea unificando los ejes para simplificar la comparación de las distribuciones. Observando los gráficos correspondientes a K-Means y Bisecting K-Means se ven distribuciones similares inversas, es decir los casos que quedaron etiquetados como 0 en Bisecting K-Means parecen corresponder a los casos que quedaron etiquetados como 1 en K-Means, lo mismo se observa teniendo en cuenta las cantidades que se muestran en 5.21 y 5.23 y los atributos clasificación con valor descartado y origen de financiamiento con valor público claramente resaltados en 5.26 y 5.27 en grupos opuestos.

Gráfico 5. 29 Histogramas de distribución de edad en Datos37 para k = 2

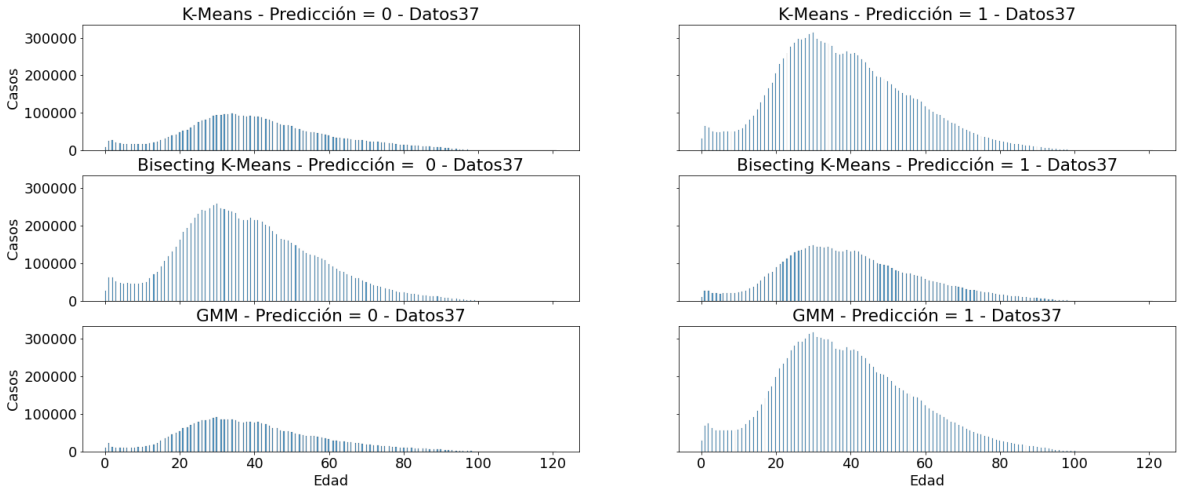
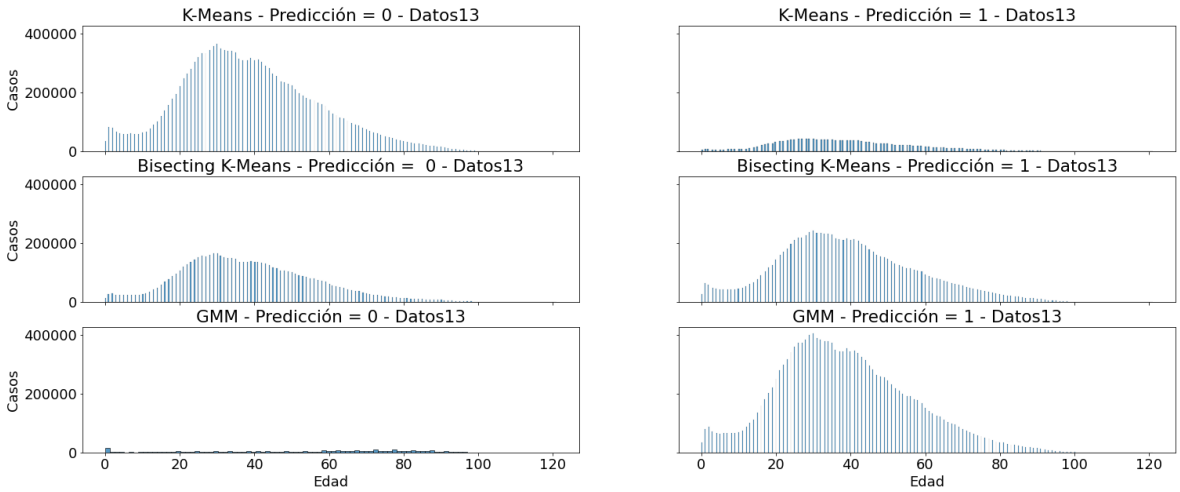
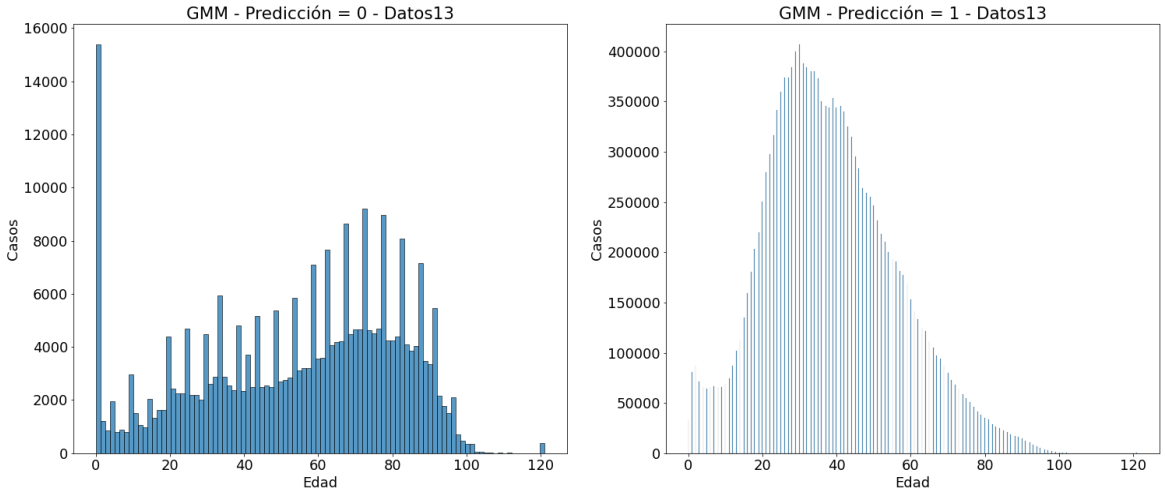


Gráfico 5. 30 Histogramas de distribución de edad en Datos13 para k = 2



El gráfico 5.31 muestra ampliado el detalle de distribución de casos en Datos13 sin unificar el eje y para observar en detalle la forma de la distribución.

Gráfico 5. 31 Histograma de distribución de edad en Datos13 para k = 2 según GMM



Las agrupaciones encontradas por Gaussian Mixture Model para k = 2 en Datos13 fueron las de mayor valor de índice Silhouette. Si bien la cantidad de casos que quedaron en los grupos es muy dispar la forma del histograma parece indicar que en el grupo 0 quedaron más casos correspondientes a edades igual a 0 o entre 60 y 100 años, y en el grupo 1 más casos entre 20 y 60 años.

En el Anexo A1.4 se incluye el resto de los histogramas con el eje y sin unificar y las cantidades de casos por edad.

5.3.2. Inclusión de las provincias

Todos los ensayos se realizaron sobre el conjunto Datos37 que incluye las 24 provincias y el conjunto Datos13. Se busca analizar por provincias siguiendo el antecedente de los trabajos realizados en otros países ¿Influye en los resultados de los modelos de agrupamiento incluir las provincias como atributos relevantes o no?

Teniendo en cuenta que el más alto valor del índice Silhouette igual a 0.92 ocurre en Datos13 luego de ejecutar el algoritmo Gaussian Mixture Model con el parámetro k igual a 2, se analizan las cantidades de casos agrupados por provincias en los tres modelos.

Los gráficos 5.32, 5.33 y 5.34 muestran la cantidad de casos agrupados por provincias etiquetadas como 0 y como 1 para cada modelo comparando en cada gráfico Datos37 y Datos13.

El modelo Gaussian Mixture Model parece separar en los casos etiquetados como 1 a las provincias con mayor cantidad de casos, en cambio en K-Means y Bisecting K-Means si bien difieren en la cantidad de casos por provincia etiquetados con valor 0 o 1, las distribuciones se ven más similares.

Gráfico 5. 32 Casos agrupados por provincias y valor de predicción según K-Means

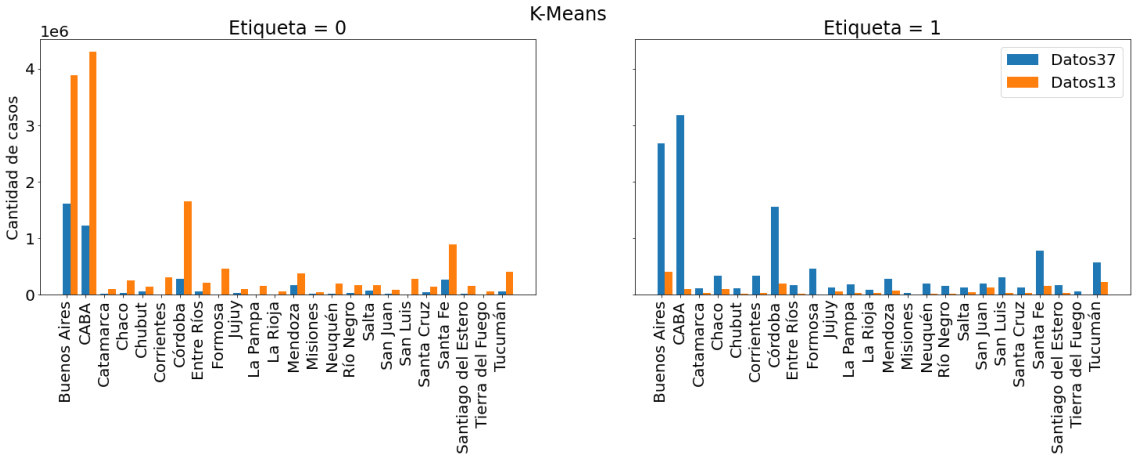


Gráfico 5. 33 Casos agrupados por provincias y valor de predicción según BK-Means

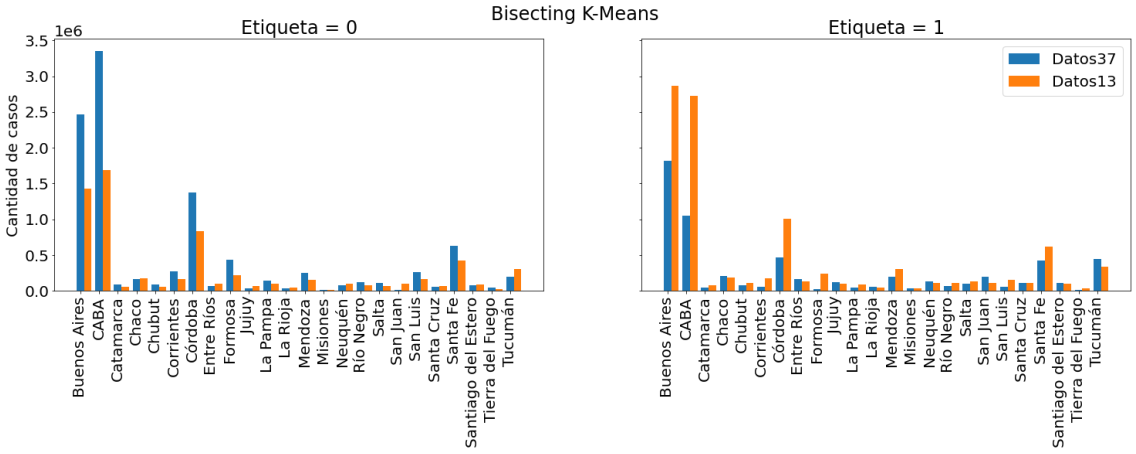
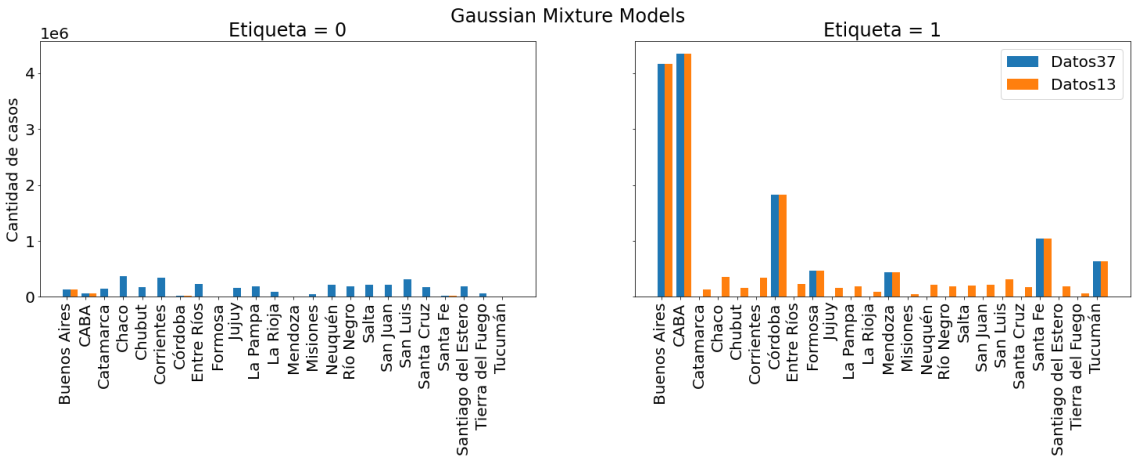


Gráfico 5. 34 Casos agrupados por provincias y valor de predicción según GMM



Analizando en detalle los resultados arrojados por el modelo Gaussian Mixture Model se puede observar en la tabla 5.4 que los casos registrados en CABA y en las provincias Buenos Aires, Córdoba, Formosa, Mendoza, Santa Fe y Tucumán que tuvieron al mismo tiempo valor 0 en los atributos fallecido, sexo no registrado, cuidado intensivo, asistencia respiratoria mecánica y casos sin clasificar en ambos conjuntos de Datos se agrupan igual. Es decir, los casos sospechosos, confirmados o descartados, que ocurrieron en hombres o

mujeres, que no tuvieron complicaciones son el grupo más numeroso. Agregar las provincias de carga a las características relevantes ocasiona una división diferente de la ocurrida en Datos13, las provincias Catamarca, Chaco, Chubut, Corrientes, Entre Ríos, Jujuy, La Pampa, La Rioja, Misiones, Neuquén, Río Negro, Salta, San Juan, San Luis, Santa Cruz, Santiago del Estero y Tierra del Fuego se agrupan en k = 0 al entrenar Datos37 permitiendo separar más provincias en el grupo 0 aunque la descripción del grupo por atributos ya no sería tan clara.

Como se puede observar en la matriz de similitud ideal 5.14 y en el gráfico 5.25 en ambos conjuntos de datos para k igual a 2 los casos etiquetados como 1 son el grupo más numeroso.

La tabla 5.4 muestra en detalle las cantidades de casos por provincia, resaltando en celeste los casos que se agrupan igual en Datos13 y Datos37.

Tabla 5. 4 Distribución de casos por provincias según GMM para k = 2

	Datos37		Datos 13	
Provincia	Predicción = 0	Predicción = 1	Predicción = 0	Predicción = 1
Buenos Aires	127733	4157162	127733	4157162
CABA	60909	4347372	60909	4347372
Catamarca	140008	0	1610	138398
Chaco	368442	0	5964	362478
Chubut	169531	0	3218	166313
Córdoba	16163	1831441	16163	1831441
Corrientes	337881	0	1349	336532
Entre Ríos	235311	0	4637	230674
Formosa	1240	465537	1240	465537
Jujuy	164016	0	3096	160920
La Pampa	187317	0	3157	184160
La Rioja	92914	0	1865	91049
Mendoza	9949	443630	9949	443630
Misiones	53424	0	1259	52165
Neuquén	217021	0	3420	213601
Río Negro	194692	0	4549	190143
Salta	211301	0	5480	205821
San Juan	212858	0	2751	210107
San Luis	318835	0	1883	316952
Santa Cruz	175835	0	2211	173624
Santa Fe	15690	1035985	15690	1035985
Santiago del Estero	193791	0	2609	191182
Tierra del Fuego	61760	0	872	60888
Tucumán	5935	632824	5935	632824

Los casos etiquetados con predicción 0 en Datos13 serían los que tienen valor 1 en alguno de los atributos fallecido, sexoNR, asistencia_respiratoria_mecanica, sin_clasificar o cuidado_intensivo.

SELECT carga_provincia_nombre, COUNT(*) canti

```
FROM Datos13 WHERE fallecido = 1 OR  sexoNR = 1
OR asistencia_respiratoria_mecanica = 1
OR cuidado_intensivo = 1
OR sin_clasificar = 1
GROUP BY carga_provincia_nombre
```

Los casos etiquetados con predicción 1 en Datos13 son aquellos que se podrían obtener filtrando los mismos atributos que la consulta anterior con valor 0 en todos ellos al mismo tiempo:

```
SELECT carga_provincia_nombre, COUNT(*)  canti
FROM Datos13 WHERE fallecido = 0 AND sexoNR = 0
AND asistencia_respiratoria_mecanica = 0
AND cuidado_intensivo = 0
AND sin_clasificar = 0
GROUP BY carga_provincia_nombre
```

Para Datos37 la descripción de casos etiquetados de igual manera no es tan clara, ya que para obtener los casos etiquetados con predicción igual a 0 se filtrarán los atributos con valor 1 en alguno de los atributos fallecido, sexoNR, asistencia_respiratoria_mecanica, cuidado_intesivo sin_clasificar, y las 17 provincias Catamarca, Chaco, Chubut, Corrientes, Entre Ríos, Jujuy, La Pampa, La Rioja, Misiones, Neuquén, Río Negro, Salta, San Juan, San Luis, Santa Cruz, Santiago del Estero o Tierra del Fuego.

```
SELECT carga_provincia_nombre, COUNT(*)  canti FROM Datos37
WHERE fallecido = 1
OR sexoNR = 1 OR asistencia_respiratoria_mecanica = 1
OR cuidado_intensivo = 1 OR sin_clasificar = 1
OR catamarca = 1 OR  chaco = 1
OR  chubut = 1 OR  corrientes = 1
OR  entre_rios = 1 OR  jujuy = 1
OR  la_pampa = 1 OR  la_rioja = 1
OR  misiones = 1 OR  rio_negro = 1
OR  salta = 1 OR  san_juan = 1
OR  san_luis = 1 OR  neuquen = 1
OR  santa_cruz = 1 OR  santiago_del_estero = 1
OR  tierra_del_fuego = 1
GROUP BY carga_provincia_nombre
```

Para obtener los casos etiquetados con predicción igual a 1 se deben filtrar al mismo tiempo los atributos fallecido, sexoNR, asistencia_respiratoria_mecanica, cuidado_intensivo, sin clasificar y las 17 provincias Catamarca, Chaco, Chubut, Corrientes, Entre Ríos, Jujuy, La Pampa, La Rioja, Misiones, Neuquén, Río Negro, Salta, San Juan, San Luis, Santa Cruz, Santiago del Estero y Tierra del Fuego igualados a 0.

```
SELECT carga_provincia_nombre, COUNT(*) canti FROM Datos37
WHERE fallecido = 0
AND sexoNR = 0 AND asistencia_respiratoria_mecanica = 0
AND cuidado_intensivo = 0 AND sin_clasificar = 0
AND  catamarca = 0 AND  chaco = 0
AND  chubut = 0 AND  corrientes = 0
AND  entre_rios = 0 AND  jujuy = 0
AND  la_pampa = 0 AND  la_rioja = 0
AND  misiones = 0 AND  rio_negro = 0
AND  salta = 0 AND  san_juan = 0
AND  san_luis = 0 AND  neuquen = 0
AND  santa_cruz = 0 AND  santiago_del_estero = 0
AND  tierra_del_fuego = 0
GROUP BY carga_provincia_nombre
```

6. Conclusiones

El término agrupamiento hace referencia a un amplio abanico de técnicas cuya finalidad es encontrar patrones o grupos dentro de un conjunto de observaciones. Es un método de aprendizaje no supervisado, ya que el proceso no tiene en cuenta a qué grupo pertenece realmente cada observación si es que existe tal información. No es posible saber con certeza si la solución obtenida es correcta o no, o si un determinado agrupamiento es mejor o peor que otro.

Es un proceso subjetivo, ya que los resultados obtenidos dependerán del algoritmo de agrupamiento seleccionado, el conjunto de datos disponible, la medida de similitud utilizada para comparar objetos, los atributos seleccionados y de la escala en la que se encuentren estos atributos.

Los atributos seleccionados como relevantes en este trabajo fueron transformados a variables binarias y normalizados para tratar de minimizar el impacto relativo a las distintas escalas, pero no existían en los atributos distancias reales salvo en el atributo edad.

Los modelos estudiados K-Means, Bisecting K-Means y Gaussian Mixture Model fueron algunos de los disponibles en MLlib, la biblioteca de aprendizaje automático de Spark. Los tres algoritmos requieren que se indique de antemano el número de clústers k que se van a crear, tarea que puede ser complicada si no se dispone de información adicional sobre los datos con los que se trabaja.

Se han desarrollado varias estrategias para ayudar a identificar potenciales valores óptimos de k como el índice Silhouette, pero son solo orientativas.

En todos los ensayos realizados en este trabajo el algoritmo K-Means demostró ser el más eficiente en cuanto a tiempo de ejecución, pero no en cuanto a la calidad de los grupos encontrados.

Los valores del Índice Silhouette más alto se obtuvieron para los resultados del algoritmo Gaussian Mixture Model en el conjunto de datos que no incluía a las provincias. Una de las limitaciones del algoritmo Gaussian Mixture Model es que no escala bien en cuanto a tiempos de creación del modelo cuando aumenta el número de dimensiones, en cambio los algoritmos K-Means y Bisecting K-Means no se vieron demasiado influenciados por el aumento del número de atributos.

Los resultados obtenidos de los ensayos utilizando Gaussian Mixture Model con el número de grupos (k) igual a 2 podrían dar indicios sobre qué atributos tener en cuenta para agrupar las provincias.

Incluir las provincias como atributos relevantes o no dependerá del objetivo a analizar ya que como indica el valor del índice Silhouette la calidad de agrupamiento disminuye.

Las visualizaciones de la matriz de evidencia y la matriz de similitud ideal pueden revelar las relaciones entre los casos existan grupos o no.

La gran cantidad de información surgida de la pandemia del Covid 19 está disponible para analizar y desarrollar y algoritmos herramientas que en un futuro sean capaces de mitigar los efectos de este tipo de pandemias. Actualmente es posible descargar el archivo histórico de casos registrados desde el 01/03/2020 hasta el 04/06/2022. El tamaño del archivo es de 6407967 KB, más del doble del tamaño del conjunto de datos utilizado en este estudio.

Como trabajo futuro se sugiere explorar diferentes medidas de distancia y otras técnicas de clustering como algoritmos basados en densidad además de reiterar los ensayos realizados en el conjunto de datos actual.

El objetivo de este trabajo fue brindar un panorama general sobre los diferentes resultados que se pueden obtener según el algoritmo que se utilice y como se lo configure.

Es importante utilizar todos los algoritmos posibles junto a técnicas de “ensemble” para obtener mayor conocimiento y brindar una cierta confianza en los resultados ya que el conocimiento obtenido, validado convenientemente, podría ser de gran utilidad para establecer políticas de salud.

Bibliografía

1. **Tan, P.-N., Steinbach, M., and Kumar, V.** *Introduction to Data Mining.* s.l. : Addison-Wesley., 2006.

2. **Hernández Orallo J., Ramírez Quintana M.J., Ferri Ramírez C.** *Introducción a la minería de datos.* s.l. : Pearson Educación S.A., 2004. ISBN: 84-205-4091-9 .

3. *MapReduce: Simplified Data Processing on Large Clusters.* **Dean, Jeffrey and Ghemawat, Sanjay.** [ed.] USENIX. s.l. : Association OSDI '04: 6th Symposium on Operating Systems Design and Implementation, 2004.

4. *Fast and Interactive Analytics over Hadoop NETWORKED SYSTEMS.* **Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy.** [ed.] USENIX. Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. : Symposium on Networked Systems Design and Implementation, 2012.

5. *MLlib: Machine Learning in Apache Spark. Journal of Machine Learning Research.* **Meng, Xiangrui , et al.** [ed.] Mark Reid. 2016.

6. **ml-clustering. [Online] https://spark.apache.org/docs/latest/ml-clustering.html.**

7. *A Clustering Approach to Classify Italian Regions and Provinces Based on Prevalence and Trend of SARS-CoV-2 Cases.* Maugeri, A., Barchitta, M., & Agodi, A. 17, 2020, International journal of environmental research and public health, Vol. 15, p. 5286.

8. *"Clustering K-Means Algorithms and Econometric Lethality Model by Covid-19, Peru 2020".* Flores Arocutipa, J.P., Jinchuña Huallpa, J., Carbajal Navarro, G. y Bermejo Peralta, L. 2021, International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 1, p. 12.

9. *Ranking and Clustering Iranian Provinces Based on COVID-19 Spread: K-Means Cluster Analysis.* Madadizadeh, F., & Sefidkar, R. 2021, Journal of Environmental Health and Sustainable Development.

10. Bahmani, B., Moseley, B., Vattani, A., Kumar, R., Vassilvitskii, S. *Scalable K-Means ++.* s.l. : proc. VLDB Endow., 2012. pp. 622-633. Vol. 5.

11. Tizón Galisteo, Daniel. Big Data Clustering Máster Thesis. *Universidad Nacional de Educación a Distancia (España).* [Online] 2017. http://e-spacio.uned.es/fez/view/bibliuned:master-ETSInformatica-IAA-Dtizon.

12. Apache/Spark. *KMeans.scala.* [Online] https://github.com/apache/spark/blob/master/mllib/src/main/scala/org/apache/spark/mllib/clustering/KMeans.scala.

13. Das, Angel. K Means clustering using Pyspark on Big Data. [Online] 2021. [Cited: 10 7, 2021.] https://towardsdatascience.com/K-Means-clustering-using-pyspark-on-big-data-6214beacdc8b.

14. covid-19-casos-registrados-en-la-republica-argentina. [Online] 2021. [Cited: 08 10, 2021.] http://datos.salud.gob.ar/dataset/covid-19-casos-registrados-en-la-republica-argentina.

15. Informes diarios . [Online] 08 10, 2021. https://www.argentina.gob.ar/coronavirus/informes-diarios/reportes/agosto2021.

16. Colaboratory. [Online] [Cited: 08 14, 2022.] https://colab.research.google.com/?hl=es#scrollTo=Nma_JWh-W-IF.

17. Apache Spark 3.3.0. [Online] https://spark.apache.org/docs/latest/ml-guide.html.

18. ¿Por qué deberías de usar ficheros Parquet si procesas muchos datos? [Online] https://datos.gob.es/es/blog/por-que-deberias-de-usar-ficheros-parquet-si-procesas-muchos-datos.

19. Singh, Pramod. *Machine Learning with PySpark.* Bangalore : Apress, 2019. ISBN-13 (pbk): 978-1-4842-4130-1.

20. QlikView para Windows. [Online] https://www.qlik.com/. Versión 11.20.13607.0 SR 17 64-bit Edition(x64).

21. Cupas, Carol. Qlik Sense vs QlikView: Diferencias y similitudes. [Online] 8 23, 2021. https://openwebinars.net/blog/qlik-sense-vs-qlikview-diferencias-y-similitudes/.

22. Kaufman Leonard, Rousseeuw Peter J. *Finding Groups in Data: An Introduction to Cluster Analysis.* [ed.] Wiley Series in Probability and Statistics. s.l. : John Wiley & Sons, Inc., 2005. 9780470316801.

23. Clustering with Python by Joaquín Amat Rodrigo. [Online] Attribution 4.0 International (CC BY 4.0). https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html.

24. *Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means.* Gopalani, Satish & Arora, Rohan. 1, 2015, International Journal of Computer Applications, Vol. 113, pp. 0975–8887.

25. Albon, Chris. *Machine Learning with Python Cookbook.* s.l. : O'Reilly Media, Inc., 2018. ISBN: 978-1-491-98938-8.

26. MLlib. [Online] https://spark.apache.org/mllib/.

27. PySpark Clustering. [Online] http://spark.apache.org/docs/latest/api/python/reference/pyspark.ml.html#clustering.

28. PySpark Documentation. [Online] http://spark.apache.org/docs/latest/api/python/.

29. Witten Ian H., Frank Eibe, Hall Mark A., Christopher J. Pal. *Data Mining. Practical Machine Learning Tools and Techniques.* s.l. : Morgan Kaufman, 2017.

30. *Increase the Performance of K-Means Clustering Algorithm Using Apache Spark.* Xie, C. 1, 2017, International Journal of Internet of Things and its Applications, Vol. 1, pp. 13-28.

31. *Aproximación al reajuste automático de centroides mediante la heurística.* Colomé Abril, Xavier. Barcelona : s.n., 2012.

Anexo 1

Este Anexo se divide en 5 secciones. En la primera se incluyen 16 tablas correspondientes a los valores de índice Silhouette y tiempos de ejecución de 5 tamaños de los conjuntos de datos utilizados para los ensayos. Las tablas A1.2. 1, A1.2. 2, A1.2. 3 y A1.2. 4 corresponden a los resultados de los ensayos de los conjuntos de datos utilizados para comparar ejecuciones desde Colab y desde el Clúster, la edad se había manejado con otro criterio. Las tablas A1.2. 5 y A1.2. 6 muestran los resultados de las ejecuciones con medida de similitud coseno.

La segunda sección incluye gráficos de tiempos de ejecución de los ensayos en función del valor del tiempo en segundos y del valor de k, sin unificar el eje y para visualizar los cambios entre los modelos

La tercera sección incluye gráficos de tiempos de ejecución en función del tiempo en segundos y del tamaño del conjunto de datos, con el eje y unificado.

La cuarta sección muestra otros gráficos de distribución de edad sin unificar el eje y además del detalle de cantidades para cada edad y etiqueta de clúster según los resultados del modelo Gaussian Mixture Model para k = 2

La quinta sección incluye ejemplos del armado de la matriz de evidencia y algunos resultados parciales.

Contenido

1. Resultados de ejecuciones en Colab

Tabla A1.1. 1 Coeficientes del índice Silhouette y tiempos para 1685 observaciones y 37 atributos

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.9973542565	0.1871225615	0.4235036247	0:00:01	0:00:01	0:00:00
3	0.1207656146	-0.09616334901	0.4226900322	0:00:00	0:00:02	0:00:01
4	0.1459238843	0.04990962085	-0.1773784371	0:00:00	0:00:02	0:00:00
5	0.1525697048	-0.02692260845	-0.1276162068	0:00:00	0:00:04	0:00:01
6	0.1955904551	0.01627399633	0.2409844832	0:00:00	0:00:04	0:00:00
7	0.1548264424	0.02828420909	0.1985626595	0:00:01	0:00:04	0:00:00
8	0.1444829323	0.03452126194	0.1988580052	0:00:01	0:00:04	0:00:00
9	0.1620839992	0.03694579612	0.2569951484	0:00:01	0:00:05	0:00:00
10	0.213118501	-0.02629767974	0.2530396477	0:00:01	0:00:06	0:00:01

Tabla A1.1. 2 Coeficientes del índice Silhouette y tiempos para 1685 observaciones y 13 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	-0.01446777107	0.2311729594	0.4676249945	0:00:41	0:00:01	0:00:00
3	0.3704190567	0.0108246236	0.4351211012	0:00:00	0:00:02	0:00:01
4	0.4668270251	0.09724686005	0.2721428243	0:00:01	0:00:02	0:00:05
5	0.48455941	0.1566554169	0.4462635648	0:00:01	0:00:03	0:00:07
6	0.460851139	0.3386250585	0.8762952156	0:00:01	0:00:03	0:00:06
7	0.4935389005	0.2893391369	0.3236218904	0:00:00	0:00:03	0:00:01
8	0.462504549	0.3563004915	0.3451087833	0:00:00	0:00:03	0:00:06
9	0.5471194989	0.3091817801	0.4554704468	0:00:00	0:00:05	0:00:06
10	0.5889622352	0.3661447731	0.4989702093	0:00:01	0:00:04	0:00:07

Tabla A1.1. 3 Coeficientes del índice Silhouette y tiempos para 16510 observaciones y 37 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.1954412524	0.1699717429	0.1249999388	0:00:01	0:00:02	0:00:01
3	0.1005717582	-0.03615297218	0.2868794702	0:00:01	0:00:05	0:00:01
4	0.07650063508	0.0282662281	0.3895087327	0:00:01	0:00:05	0:00:17
5	0.1459607371	-0.08227149989	0.3895087327	0:00:01	0:00:06	0:00:05
6	-0.08585864465	-0.04145350272	0.2833837717	0:00:01	0:00:07	0:00:01
7	-0.03666736067	-0.0170188526	0.3895087327	0:00:01	0:00:07	0:00:21
8	0.09109609708	0.006961652508	0.2014453371	0:00:01	0:00:08	0:00:01
9	0.1903976867	0.03475474909	0.275076112	0:00:01	0:00:10	0:00:01
10	0.1910084381	0.0102895375	0.3895087327	0:00:01	0:00:09	0:00:03

Tabla A1.1. 4 Coeficientes del índice Silhouette y tiempos para 16510 observaciones y 13 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.8358413321	0.05624999847	0.8949711518	0:00:39	0:00:02	0:00:07
3	0.3234731064	0.2007689906	0.6642352779	0:00:01	0:00:04	0:00:07
4	0.3438486156	0.1457408415	0.3885631068	0:00:01	0:00:04	0:00:08
5	0.3642836949	0.2271350312	0.4528383675	0:00:01	0:00:06	0:00:09

6	0.4436850936	0.2446626274	0.3829614143	0:00:01	0:00:06	0:00:10
7	0.5330438552	0.3805144067	0.8830343016	0:00:01	0:00:07	0:00:11
8	0.5369596365	0.3294238965	0.4658719296	0:00:01	0:00:07	0:00:12
9	0.4856595429	0.4063608977	0.3591232632	0:00:01	0:00:08	0:00:13
10	0.4884873798	0.3582632984	0.8901215725	0:00:01	0:00:08	0:00:13

Tabla A1.1. 5 Coeficientes del índice Silhouette y tiempos para 164614 observaciones y 37 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting KMeans	GMM	KM	BKM	GMM
2	0.1885011979	0.1641007062	0.5222257337	0:00:04	0:00:15	0:00:38
3	0.1335497431	0.003832212194	0.47739167	0:00:04	0:00:22	0:00:48
4	0.1451886644	0.04248808929	0.5739357588	0:00:04	0:00:26	0:00:56
5	0.150420298	-0.07476776834	0.430174897	0:00:06	0:00:35	0:01:04
6	0.1600055353	-0.02407338926	0.5739357588	0:00:05	0:00:37	0:00:10
7	0.03170724477	0.0008230070049	0.4638236817	0:00:06	0:00:40	0:00:12
8	0.0926128971	-0.004861259125	0.5739357588	0:00:04	0:00:41	0:00:12
9	0.106196376	0.03573824189	0.4964032185	0:00:04	0:00:47	0:00:16
10	0.08348819447	0.002898996244	0.5739357588	0:00:07	0:00:50	0:00:14

Tabla A1.1. 6 Coeficientes del índice Silhouette y tiempos para 164614 observaciones y 13 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.4504933034	0.2080800276	0.926698484	0:00:44	0:00:13	0:00:26
3	0.3477993781	0.3313311194	0.8705653265	0:00:03	0:00:23	0:00:29
4	0.3514996515	0.2777696101	0.3307208335	0:00:03	0:00:31	0:00:38
5	0.3771268435	0.2847259845	0.4741806479	0:00:03	0:00:34	0:00:42
6	0.4616102794	0.3034897327	0.8896858106	0:00:03	0:00:36	0:00:44
7	0.4793196762	0.35633819	0.4511884692	0:00:03	0:00:39	0:00:51
8	0.4999857272	0.3836439747	0.3785385376	0:00:03	0:00:40	0:00:59
9	0.5671656407	0.4101421103	0.8783408864	0:00:04	0:00:48	0:01:00
10	0.5871193404	0.3491324747	0.8700637599	0:00:05	0:00:49	0:01:03

Tabla A1.1. 7 Coeficientes del índice Silhouette y tiempos para 1651427 observaciones y 37 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	-0.15466483	0.171960794	0.543468095	0:00:36	0:02:15	0:05:37
3	-0.116802793	-0.003385368	0.571293928	0:00:39	0:03:58	0:07:09
4	-0.133162706	0.053882829	0.571293928	0:00:36	0:04:33	0:08:35
5	0.1278671	-0.041625545	0.511236573	0:00:34	0:05:53	0:10:02
6	0.029758015	-0.004790161	0.571293928	0:00:43	0:06:26	0:11:28
7	0.195157762	0.034255936	0.520575287	0:00:47	0:06:56	0:12:56
8	0.216764103	0.031762337	0.516489645	0:00:44	0:07:00	0:14:24
9	0.243559481	0.032592391	0.534050933	0:00:47	0:08:07	0:15:37
10	0.154972754	-0.011508371	0.493250224	0:00:46	0:08:27	0:16:50

Tabla A1.1. 8 Coeficientes del índice Silhouette y tiempos para 1651427 observaciones y 13 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.448949412	0.2021835	0.924471859	0:01:49	0:02:18	0:04:10
3	0.442433837	0.33347894	0.317251194	0:00:57	0:03:57	0:05:02
4	0.300905808	0.269874327	0.876568418	0:00:48	0:04:28	0:05:44
5	0.406074487	0.293576735	0.883743727	0:00:38	0:05:55	0:06:27
6	0.479246104	0.311606467	0.522302295	0:00:52	0:06:37	0:07:59
7	0.480219077	0.365563709	0.907183652	0:00:46	0:06:49	0:08:05
8	0.485865625	0.38975531	0.478059808	0:00:45	0:07:19	0:09:21
9	0.486849011	0.422037688	0.478780495	0:00:41	0:08:39	0:10:09
10	0.465827968	0.360555515	0.488782129	0:00:52	0:09:02	0:11:26

Tabla A1.1. 9 Coeficientes del índice Silhouette y tiempos para 16486507 observaciones y 37 atributos.

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.159652258	0.173133563	0.494777426	0:10:10	0:23:39	1:01:57
3	0.174690818	-0.018155364	0.414701434	0:10:35	0:42:23	1:18:31
4	0.102544039	0.047507621	0.519178581	0:10:03	0:47:45	1:31:04
5	0.117796797	-0.165269048	0.485823863	0:11:00	1:04:19	1:45:43
6	0.132708457	-0.127976302	0.535660827	0:11:02	1:04:08	1:59:37
7	-0.056344921	-0.115092465	0.33808363	0:08:51	1:11:24	2:13:06
8	0.098965318	-0.09586657	0.519972687	0:11:28	1:15:21	2:26:30
9	0.136801242	-0.074941149	0.528799581	0:12:50	1:32:20	2:44:55
10	0.085644323	-0.040261082	0.51899194	0:11:16	1:32:30	2:53:32

Tabla A1.1. 10 Coeficientes del índice Silhouette y tiempos para 16486507 observaciones y 13 atributos

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.551116576	0.198850319	0.924154258	0:11:15	0:21:32	0:41:26
3	0.414945309	0.332553573	0.873091405	0:07:24	0:39:37	0:50:38
4	0.399787822	0.214004291	0.872880799	0:08:21	0:45:06	0:59:17
5	0.447546289	0.264300883	0.390746467	0:08:41	1:02:22	1:18:15
6	0.472363192	0.303568957	0.886622853	0:09:42	1:06:08	1:22:42
7	0.366136005	0.209595659	0.871787127	0:08:39	1:10:59	1:25:08
8	0.467109305	0.271906238	0.876374878	0:08:18	1:16:35	1:33:16
9	0.459305078	0.231572693	0.876159906	0:08:57	1:32:17	1:38:30
10	0.550902453	0.143682108	0.373621466	0:09:30	1:35:04	1:52:25

1.1. Conjunto de datos ejecutados en Colab y el clúster de computadoras

Tabla A1.2. 2 Coeficientes del índice Silhouette y tiempos para 16486550 observaciones y 37 atributos (Clúster).

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.12497911	0.17303938	0.44653483	00:02:07	00:16:42	00:13:19
3	0.17180834	-0.01819167	0.45815803	00:02:54	00:29:57	00:15:53
4	-0.01231126	0.04746864	0.54196120	00:01:58	00:33:42	00:18:04
5	-0.03513567	-0.16520863	0.51746772	00:02:24	00:50:18	00:20:05

6	-0.01917515	-0.12785413	0.50720772	00:02:16	00:47:18	00:22:52
7	0.02167324	-0.11497988	0.53686423	00:02:20	00:58:02	00:27:33
8	0.19010190	-0.09579980	0.50334436	00:02:41	00:50:33	00:31:52
9	0.16295759	-0.07453884	0.53076837	00:02:52	00:59:19	00:30:30
10	0.20833456	-0.03959039	0.52521601	00:02:16	00:10:16	00:31:52

Tabla A1.2. 3 Coeficientes del índice Silhouette y tiempos para 16486550 observaciones y 13 atributos (Clúster).

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.42054238	0.19886650	0.92413557	00:01:02	00:01:29	00:09:34
3	0.25925491	0.33252076	0.43648382	00:01:00	00:01:49	00:10:32
4	0.34513778	0.24659855	0.32926133	00:01:01	00:02:05	00:11:15
5	0.41886924	0.28676508	0.88873441	00:01:02	00:02:22	00:11:47
6	0.44261337	0.33009355	0.23954792	00:01:04	00:02:32	00:13:00
7	0.40620762	0.19644265	0.45998924	00:01:04	00:02:39	00:15:03
8	0.48298827	0.25415546	0.37257875	00:01:06	00:02:45	00:14:57
9	0.50850367	0.21506599	0.27684688	00:01:06	00:02:59	00:16:25
10	0.61150377	0.31859507	0.42334458	00:01:08	00:03:03	00:16:46

Tabla A1.2. 4 Coeficientes del índice Silhouette y tiempos para 16486550 observaciones y 37 atributos(Colab).

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.10283717	0.17303938	0.45199872	00:08:43	00:22:47	01:04:47
3	0.16627948	-0.01819167	0.52521601	00:10:40	00:41:31	01:16:26
4	0.14557527	0.0580809	0.43250675	00:08:55	0:46:55	01:31:28
5	0.13281214	0.00747143	0.52521601	00:07:38	01:04:11	01:43:49
6	0.08934556	-0.01819167	0.52707812	00:07:56	01:07:14	01:52:36
7	0.11229207	-0.06202213	0.50744793	00:09:27	01:11:06	02:13:52
8	0.08356671	-0.05047113	0.46320278	00:11:14	01:17:10	02:27:38
9	0.0765883	-0.00987628	0.43601926	00:13:22	01:32:37	02:54:00
10	0.14921378	-0.00739929	0.52521601	00:12:37	01:35:07	02:58:02

Tabla A1.2. 5 Coeficientes del índice Silhouette y tiempos para 16486550 observaciones y 13 atributos (Colab).

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.42054238	0.1988665	0.91025619	00:13:11	00:23:22	0:45:18
3	0.33252076	0.33252076	0.87874738	00:08:23	00:42:26	0:52:21
4	0.29198738	0.21397034	0.90715862	00:14:24	00:48:32	1:01:25
5	0.44746811	0.26420054	0.87255528	00:07:44	1:05:31	1:09:23
6	0.3860968	0.30347274	0.29141909	00:07:51	1:09:42	1:29:25
7	0.41693838	0.20936078	0.32692998	00:08:55	1:14:28	1:32:23
8	0.51873652	0.27009469	0.88147154	00:09:02	1:16:17	1:31:42
9	0.37824752	0.23674942	0.40557485	00:09:22	1:45:58	1:50:27
10	0.58676708	0.34391863	0.3749018	00:08:59	1:38:45	2:03:48

1.2. Resultados con medida de similitud coseno

Tabla A1.2. 5 Coeficientes del índice Silhouette y tiempos para 16486507 observaciones, 37 atributos y medida de similitud coseno

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.17210974	0.21717868	0.24654724	00:11:01	00:22:28	00:58:47
3	0.19257489	0.22092162	0.24541446	00:10:41	00:40:46	01:14:13
4	0.18103166	0.19420256	0.2628858	00:08:37	00:43:43	01:28:06
5	0.17089334	0.16795419	0.24658083	00:09:05	01:05:22	01:42:00
6	0.26516492	0.19682208	0.26160158	00:10:13	01:06:51	01:59:08
7	0.24377502	0.18341818	0.22072488	00:10:54	01:12:26	02:14:36
8	0.21510899	0.19722079	0.26167646	00:10:37	01:22:41	02:29:17
9	0.21221043	0.19015291	0.26113703	00:09:42	01:33:22	02:32:55
10	0.30373765	0.19031125	0.2615553	00:11:08	01:34:16	03:02:16

Tabla A1.2. 6 Coeficientes del índice Silhouette y tiempos para 16486507 observaciones, 13 atributos y medida de similitud coseno

k	*Coeficiente Silhouette*			*Tiempo en segundos*		
	K-Means	Bisecting K-Means	GMM	KM	BKM	GMM
2	0.36796905	0.4161293	0.50943525	00:10:45	00:24:56	00:43:07
3	0.47197123	0.42942637	0.50243013	00:09:36	00:44:53	00:51:08
4	0.5098261	0.51891853	0.48673372	00:09:34	00:50:48	00:59:41
5	0.51226833	0.52896293	0.27696799	00:10:25	01:11:44	01:18:42
6	0.60386453	0.60821619	0.47068432	00:10:18	01:16:49	01:23:09
7	0.71569058	0.6168347	0.47523969	00:10:20	01:20:12	01:24:27
8	0.53564288	0.65342566	0.42997817	00:09:33	01:18:30	01:39:38
9	0.66072394	0.58601343	0.48037633	00:09:47	01:32:00	01:42:35
10	0.68128016	0.54788151	0.39312931	00:10:29	01:36:37	01:55:38

2. Tiempos de ejecución

Gráfico A1.2. 1 Tiempos de ejecución en función del valor de k sin unificar el eje y

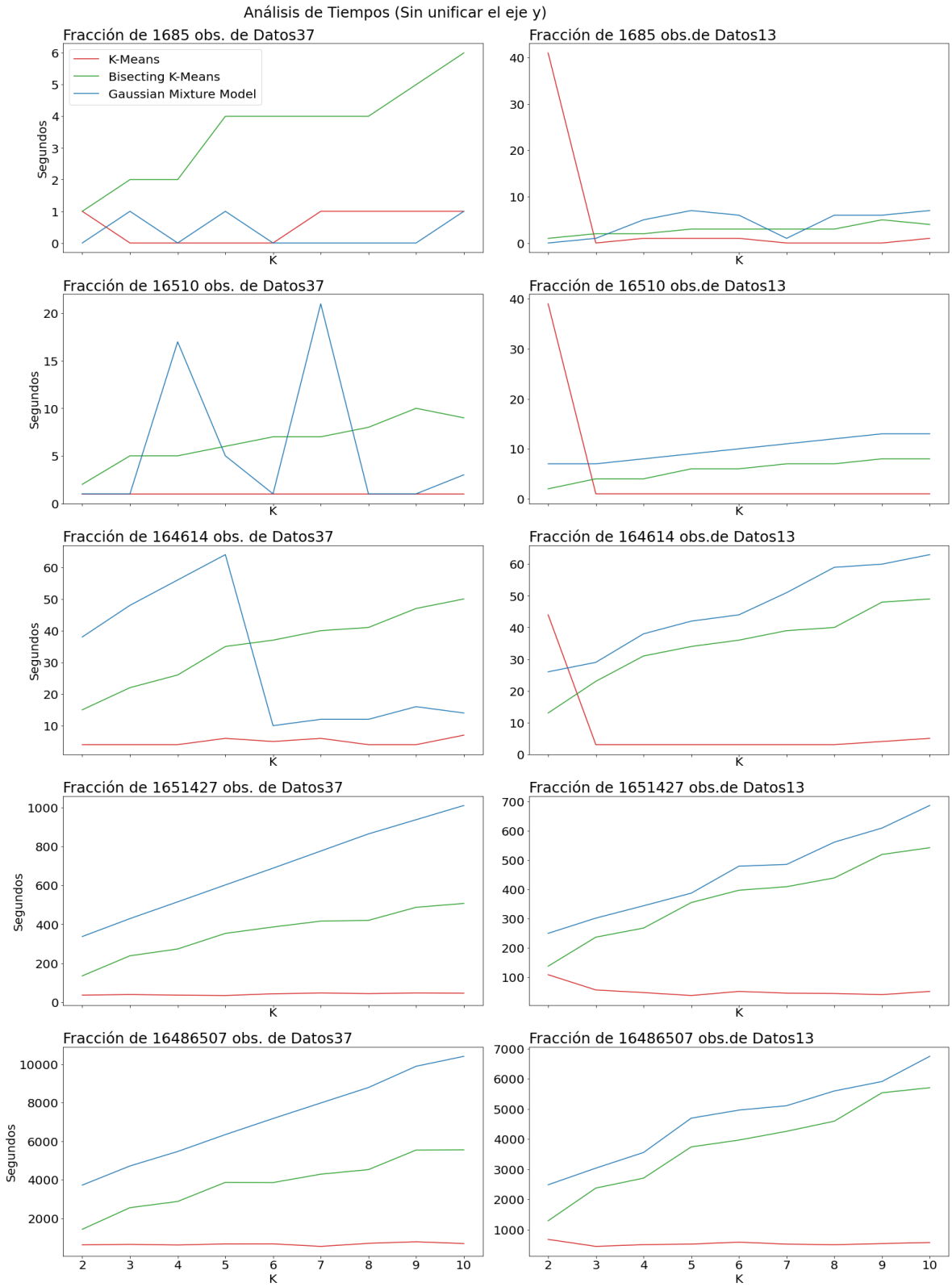
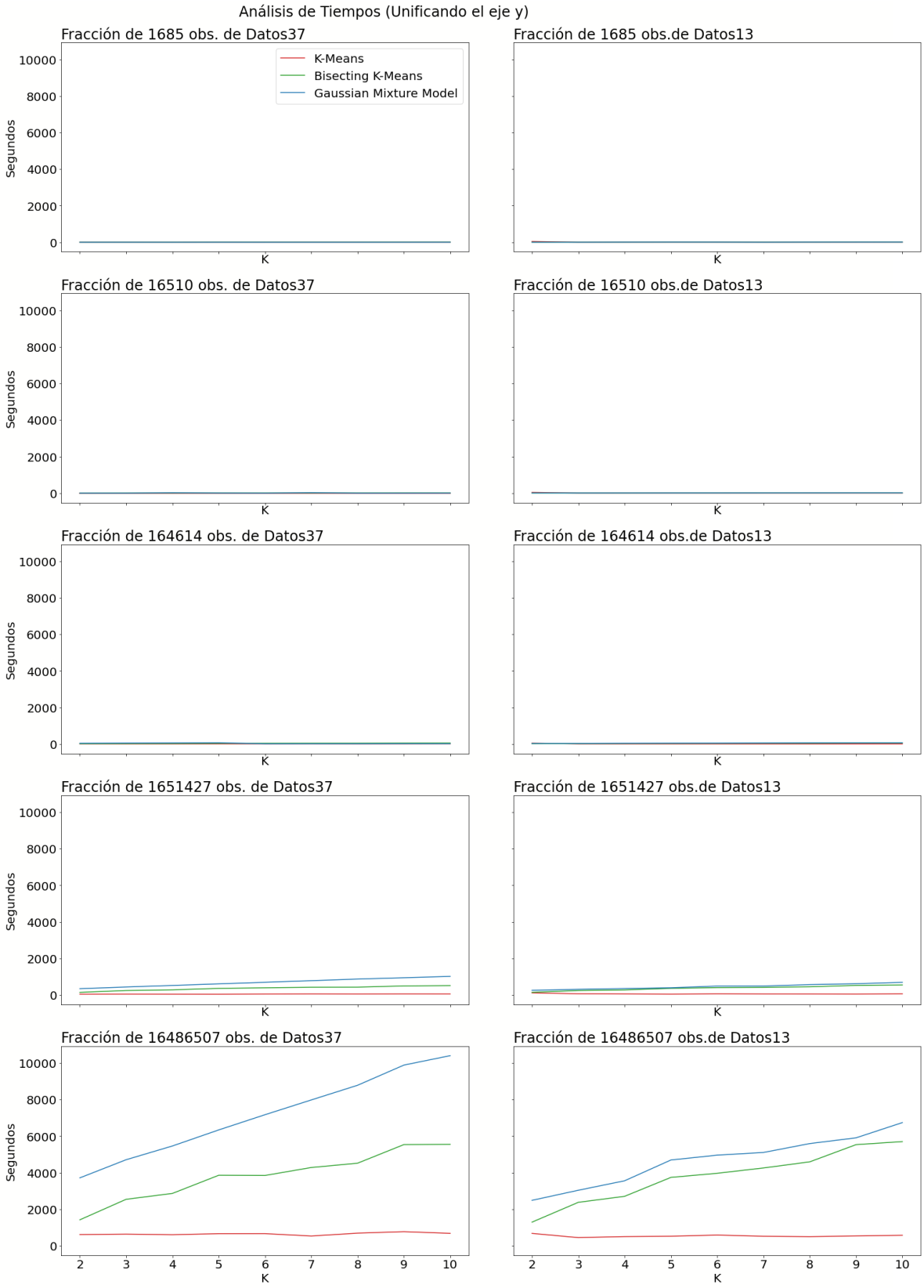
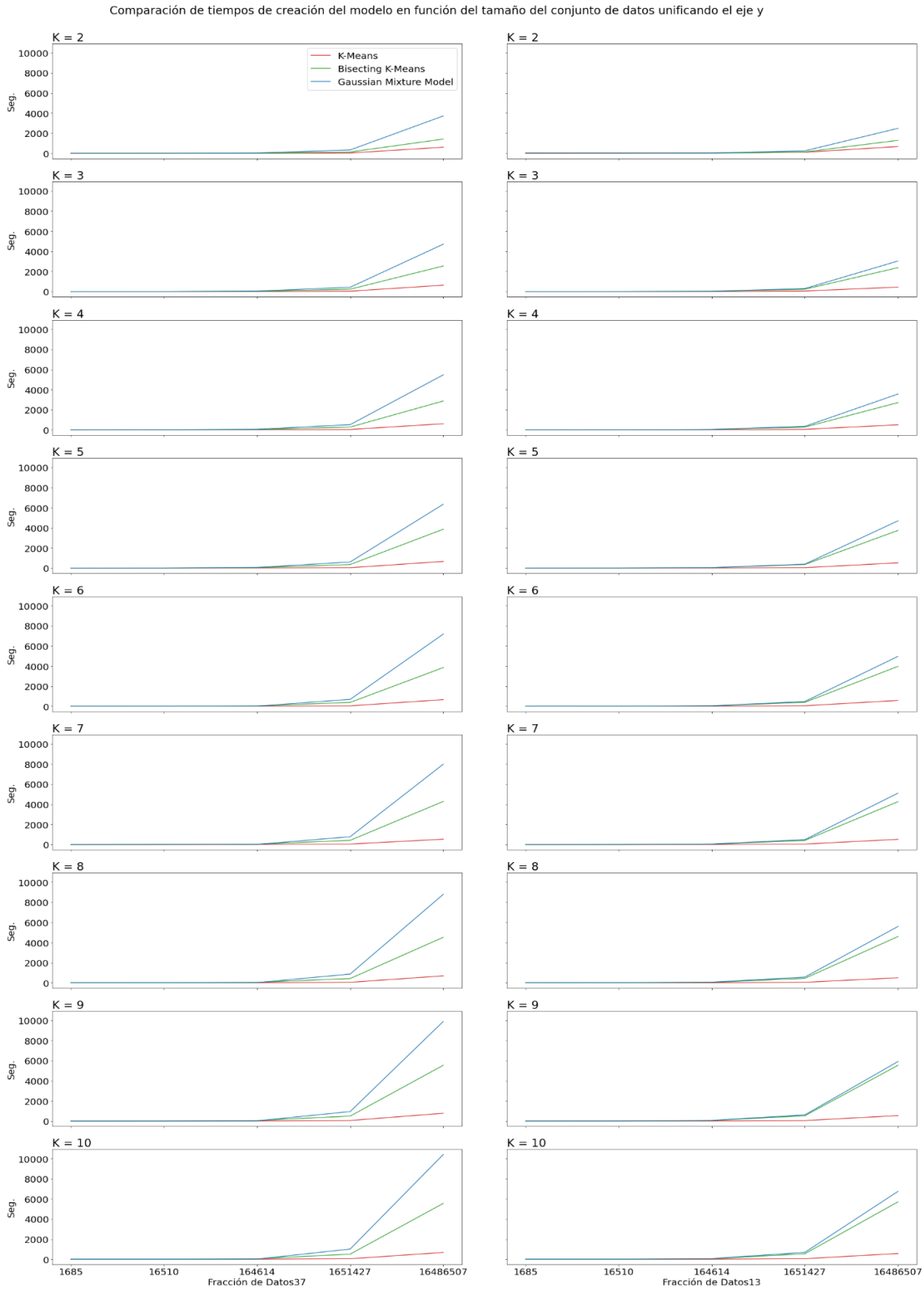


Gráfico A1.2. 2 Tiempos de ejecución en función del valor de k unificando el eje y



3. Tiempos de ejecución en función del tamaño del conjunto de datos

Gráfico A1.3. 1 Tiempos de ejecución en función del tamaño del conjunto de datos sin unificar el eje y



4. Distribución de edad

Gráfico A1.4. 1 Distribución de casos por edad y predicción según K-Means en Datos 37

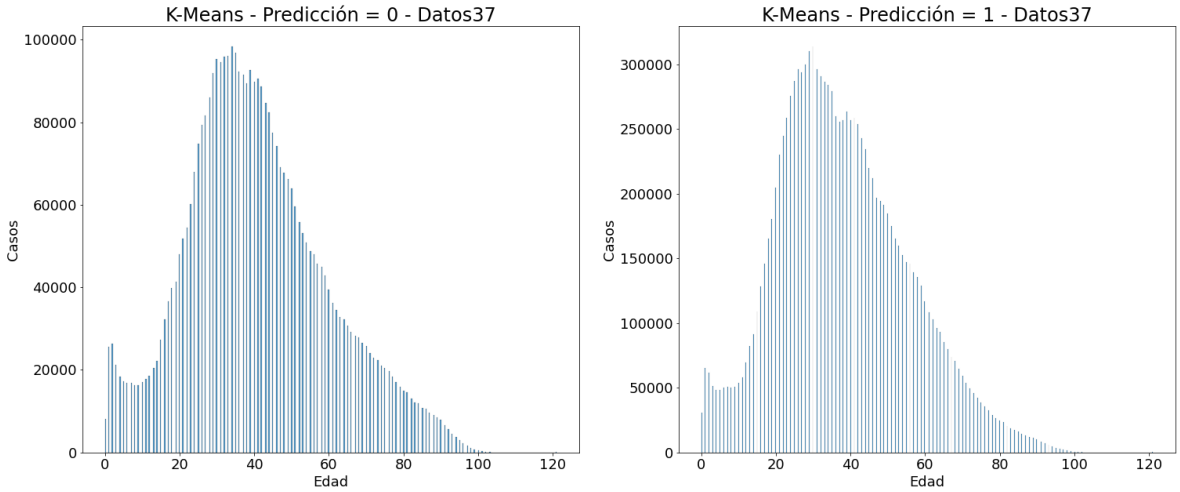


Gráfico A1.4. 2 Distribución de casos por edad y predicción según Bisecting K-Means en Datos 37

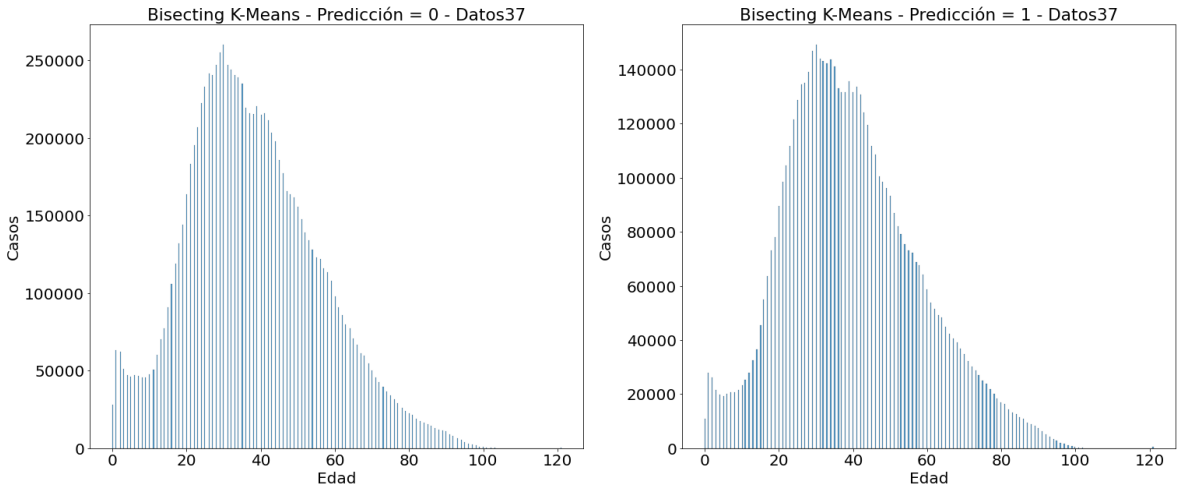


Gráfico A1.4. 3 Distribución de casos por edad y predicción según GMM en Datos 37

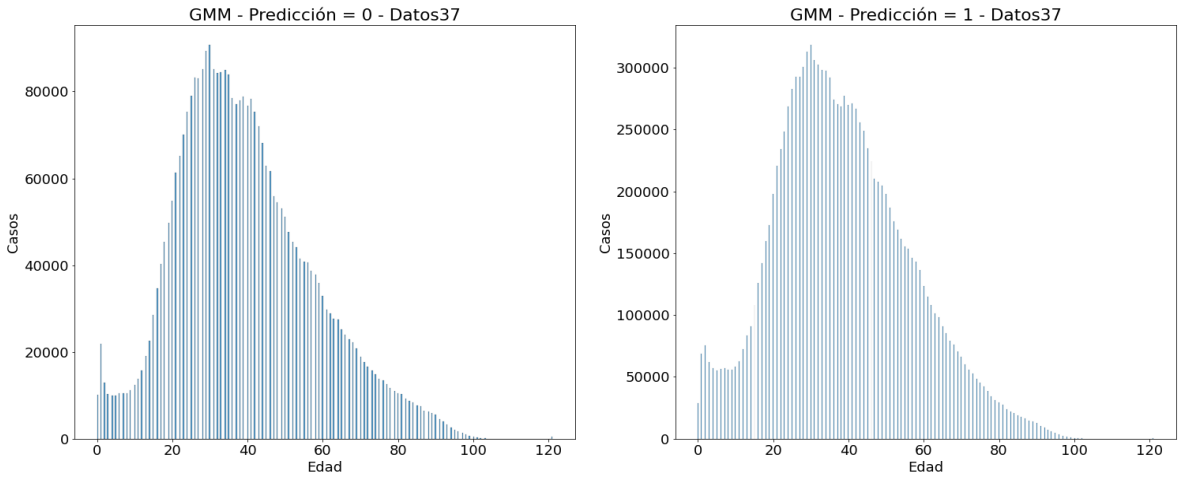


Gráfico A1.4. 4 Distribución de casos por edad y predicción según K-Means en Datos 13

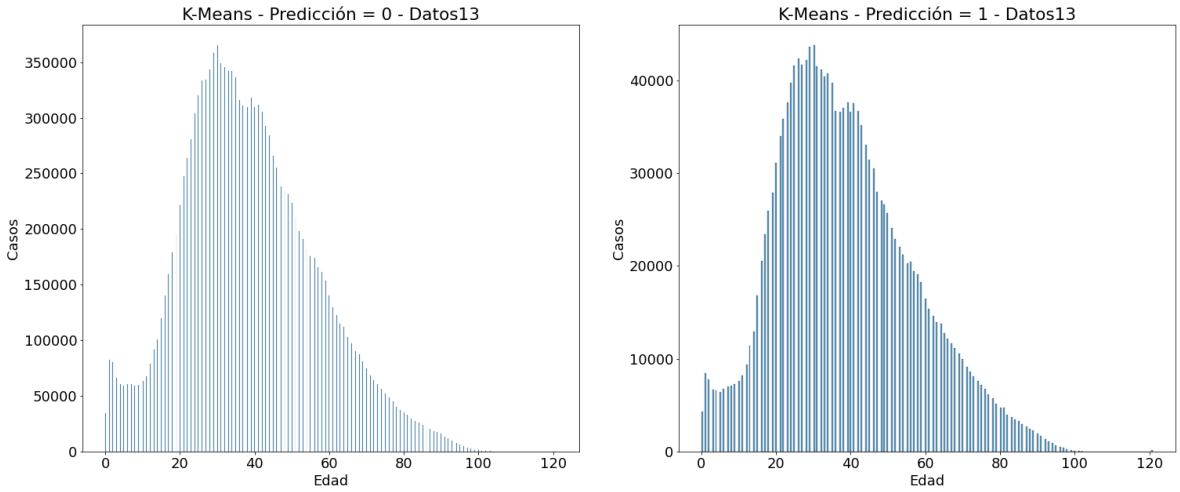


Gráfico A1.4. 5 Distribución de casos por edad y predicción según Bisecting K-Means en Datos 13

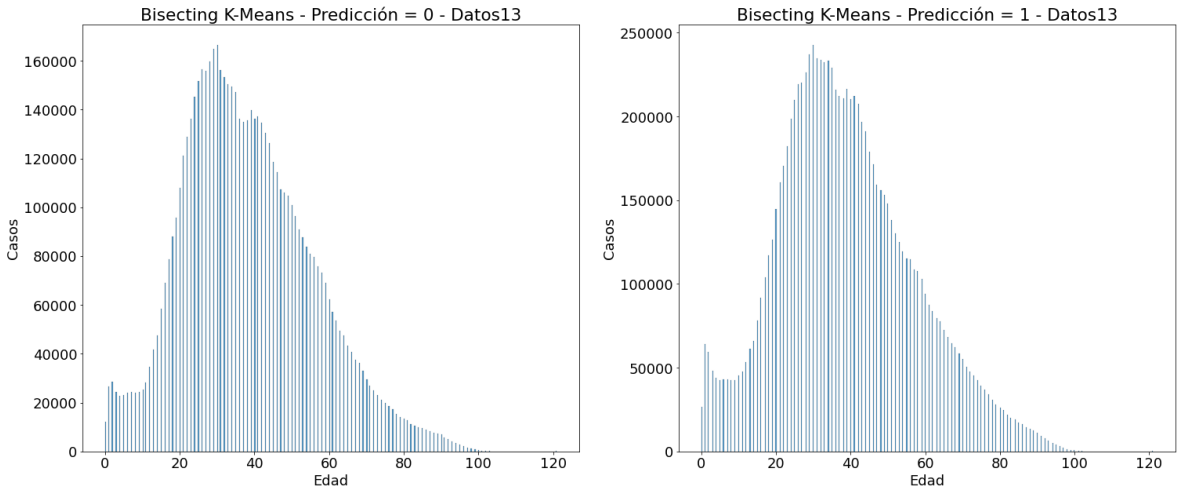
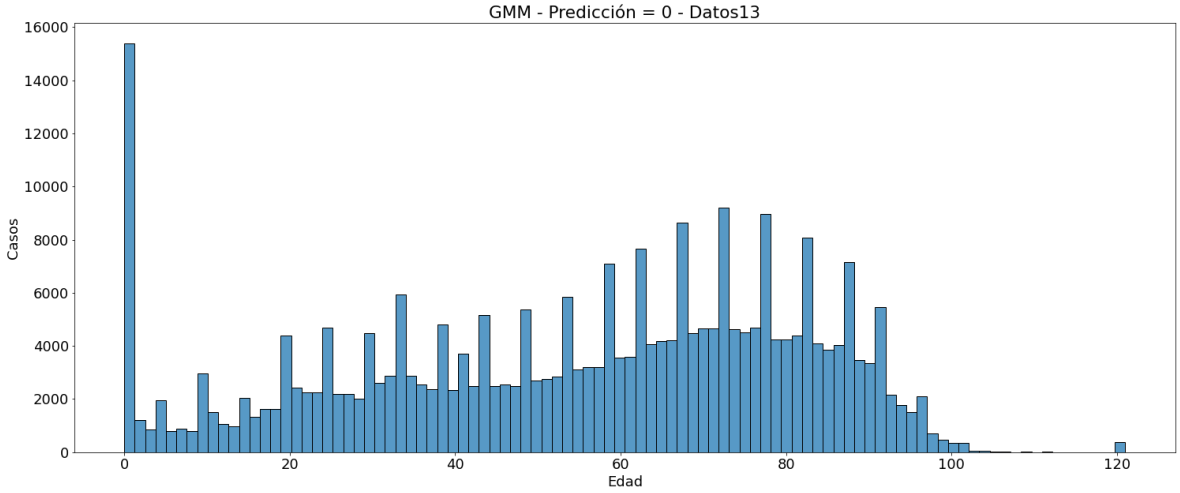


Gráfico A1.4. 6 Distribución de casos por edad y predicción según GMM en Datos 13 predicción = 0



4.1. Cantidad de casos por edad y predicción en Datos37 y Datos 13

Tabla A1.4. 6 Cantidad de casos por edad y predicción en Datos37 y Datos 13 según GMM (Datos correspondientes a la tercer fila de los gráficos 5.29 y 5.30)

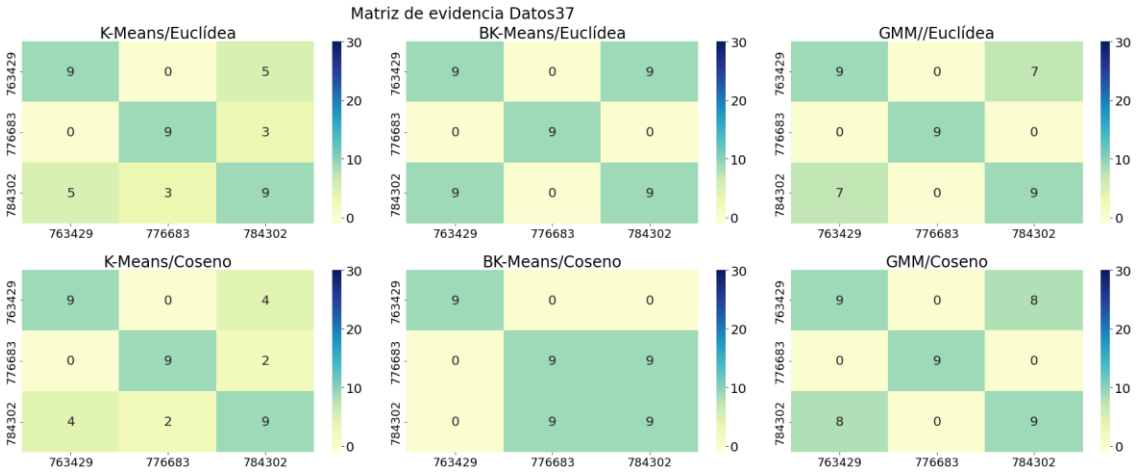
	Datos37		Datos13	
	Etiqueta			
edad	0	1	0	1
0	10058	28663	5149	33572
1	21834	69041	10234	80641
2	12867	75278	1223	86922
3	10265	62260	850	71675
4	10007	56919	1003	65923
5	10033	55288	956	64365
6	10408	56551	800	66159
7	10453	56921	893	66481
8	10522	55977	794	65705
9	11107	55795	1148	65754
10	12436	58327	1829	68934
11	13838	62300	1517	74621
12	15741	72413	1077	87077
13	19095	83653	982	101766
14	22591	91136	892	112835
15	28589	107838	1164	135263
16	34719	125985	1318	159386
17	40296	142104	1636	180764
18	45383	159726	1622	203487
19	49804	172397	2102	220099
20	54855	197979	2275	250559
21	61364	220360	2427	279297
22	65260	234322	2262	297320
23	70033	248581	2259	316355
24	75364	268552	2333	341583
25	78964	282762	2370	359356
26	83172	292868	2205	373835
27	83080	292661	2194	373547
28	85090	300709	2023	383776
29	89451	312593	2086	399958
30	90753	318352	2386	406719
31	85095	305814	2600	388309
32	84315	302697	2868	384144
33	84396	298287	2951	379732
34	84944	297748	2977	379715
35	83972	291988	2879	373081
36	78411	273971	2539	349843
37	77128	270369	2377	345120
38	78056	268618	2431	344243

	Datos37		Datos13	
	Etiqueta			
edad	0	1	0	1
61	29777	114923	3578	141122
62	28926	108393	3799	133520
63	27618	101372	3870	125120
64	27536	98011	4077	121470
65	25218	90642	4192	111668
66	23993	85100	4216	104877
67	22970	78963	4294	97639
68	22248	76403	4355	94296
69	20840	70658	4487	87011
70	18887	66050	4647	80290
71	17737	59946	4671	73012
72	16621	56141	4538	68224
73	15772	52801	4675	63898
74	14914	48693	4639	58968
75	13885	45148	4525	54508
76	13442	42108	4701	50849
77	12591	38484	4581	46494
78	11715	34238	4390	41563
79	11017	31202	4251	37968
80	10451	29260	4248	35463
81	10233	27538	4390	33381
82	9227	24013	4089	29151
83	8651	22148	3997	26802
84	8303	20809	4090	25022
85	7620	19199	3845	22974
86	7417	17839	4024	21232
87	6518	16190	3620	19088
88	6208	14817	3529	17496
89	5900	13780	3457	16223
90	5574	12767	3365	14976
91	4592	10474	2900	12166
92	3985	9128	2550	10563
93	3297	7318	2174	8441
94	2623	5930	1788	6765
95	2132	4653	1508	5277
96	1686	3505	1179	4012
97	1296	2619	933	2982
98	935	1850	702	2083
99	617	1156	463	1310

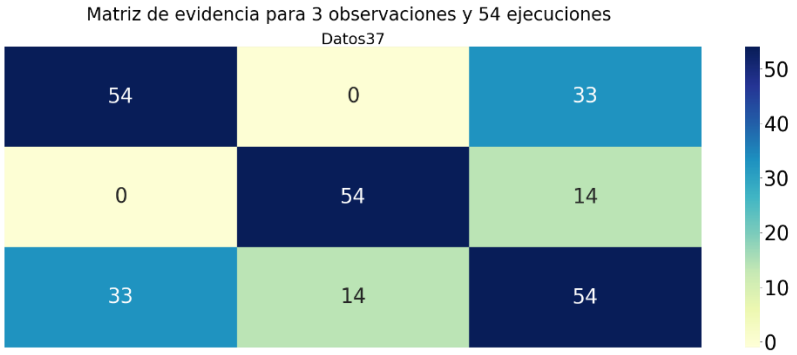
39	78795	277260	2378	353677	100	443	726	351	818
40	76736	269821	2327	344230	101	283	443	232	494
41	78346	271163	3712	345797	102	163	233	115	281
42	75373	266931	2476	339828	103	90	150	68	172
43	71932	255644	2622	324954	104	47	55	39	63
44	68149	249004	2534	314619	105	28	51	24	55
45	62928	234507	2501	294934	106	23	25	18	30
46	61638	224427	2554	283511	107	9	15	9	15
47	55961	210370	2504	263827	108	8	6	6	8
48	54479	207533	2701	259311	109	13	9	10	12
49	53109	204580	2670	255019	110	3	1	3	1
50	51238	197793	2708	246323	111	13	1	13	1
51	47594	186936	2762	231768	112	3	2	3	2
52	45352	175768	2858	218262	113	1	1	1	1
53	44060	168852	2906	210006	114	0	1	0	1
54	41585	161814	2947	200452	115	2	0	2	0
55	40825	155204	3122	192907	116	1	0	1	0
56	40576	153689	3191	191074	117	0	2	0	2
57	38758	146022	3189	181591	118	0	2	0	2
58	37859	142942	3544	177257	119	2	3	2	3
59	35977	136240	3555	168662	120	24	16	24	16
60	32986	123444	3549	152881	121	423	193	350	266

5. Matriz de evidencia

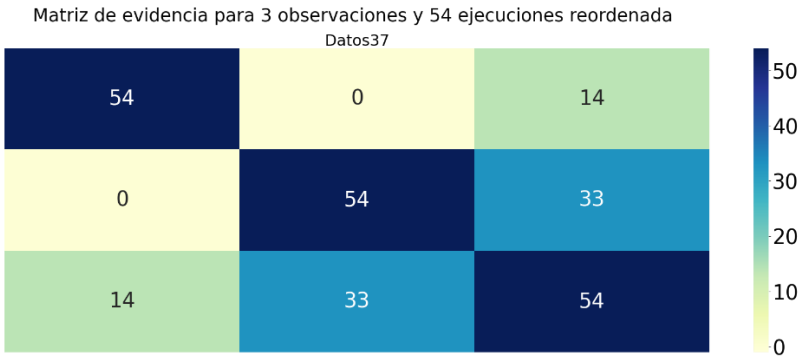
Ejemplo de armado de la matriz de evidencia para 3 observaciones luego de 27 ejecuciones. Estas 27 ejecuciones corresponden a 9 valores de k configurados en los 3 modelos



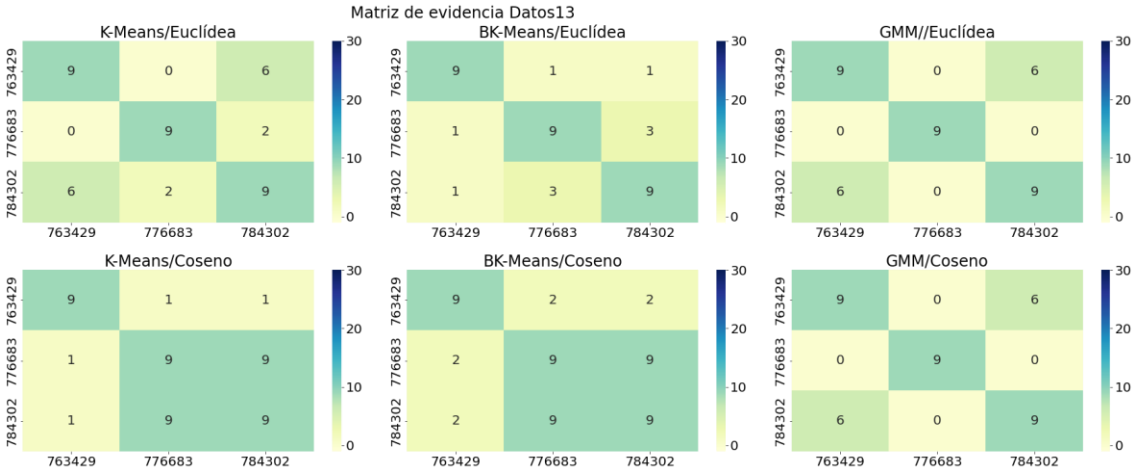
La suma de cada celda da como resultado la matriz que se muestra a continuación



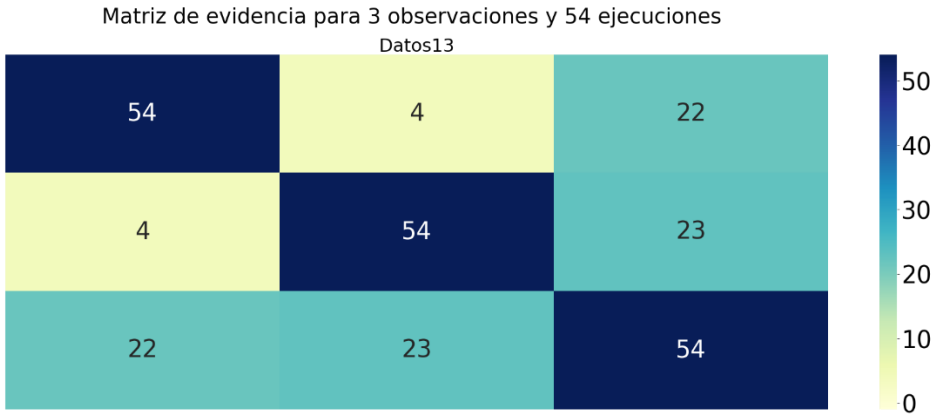
Ordenando por cantidad de ejecuciones (máximos valores de los ejes)



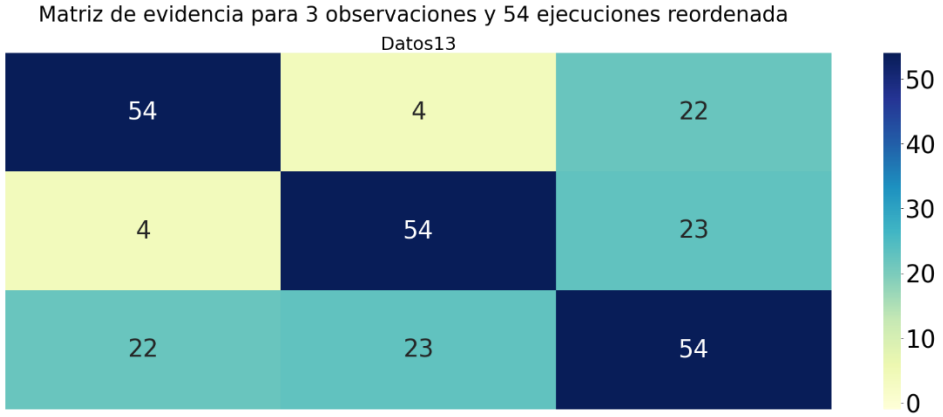
Misma idea para las mismas 3 muestras en Datos13



Sumando por celda se obtiene:

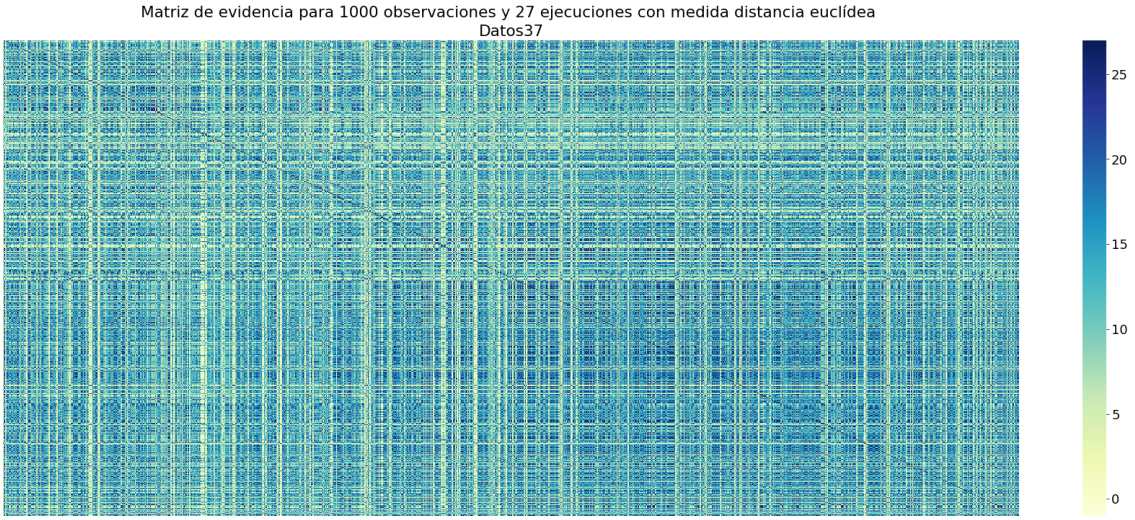


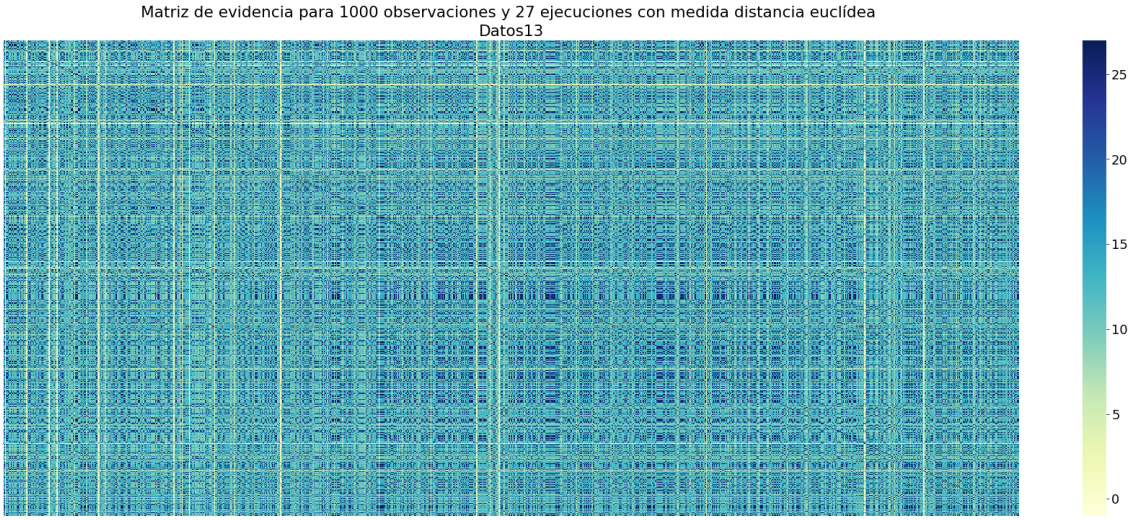
Y reordenando en este caso queda igual

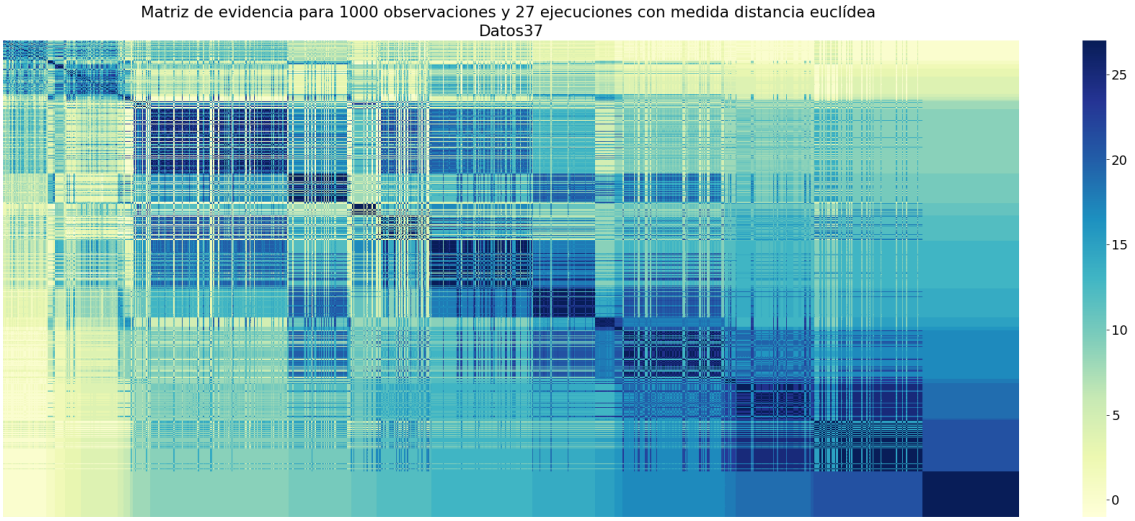


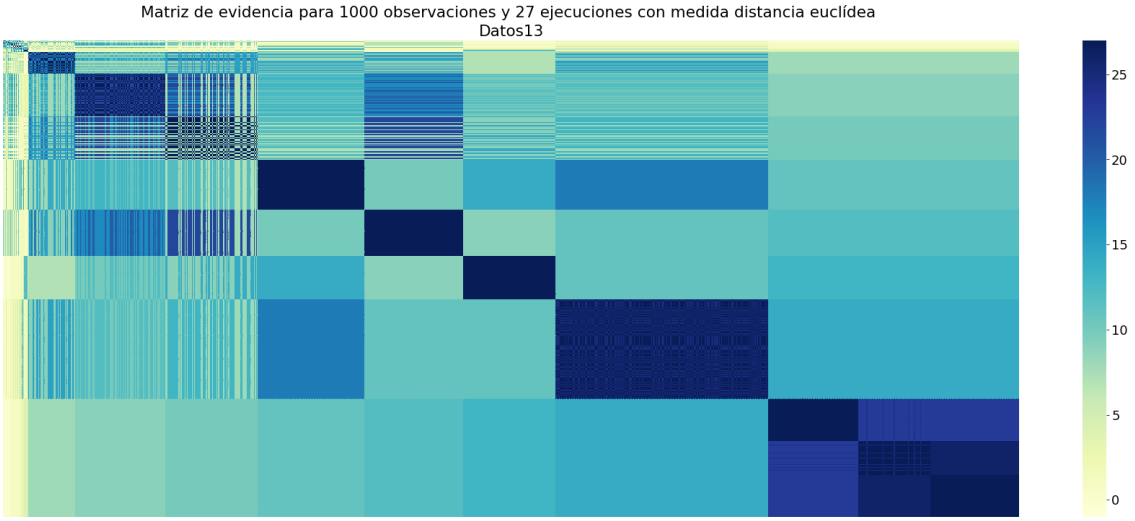
5.1. Matriz de evidencia / Distancia euclídea

A continuación, se incluyen las matrices calculadas por medida de similitud, es decir para 1000 muestras de cada conjunto de datos se suman 27 ejecuciones para distancia euclídea y luego las 27 correspondientes a similitud coseno.

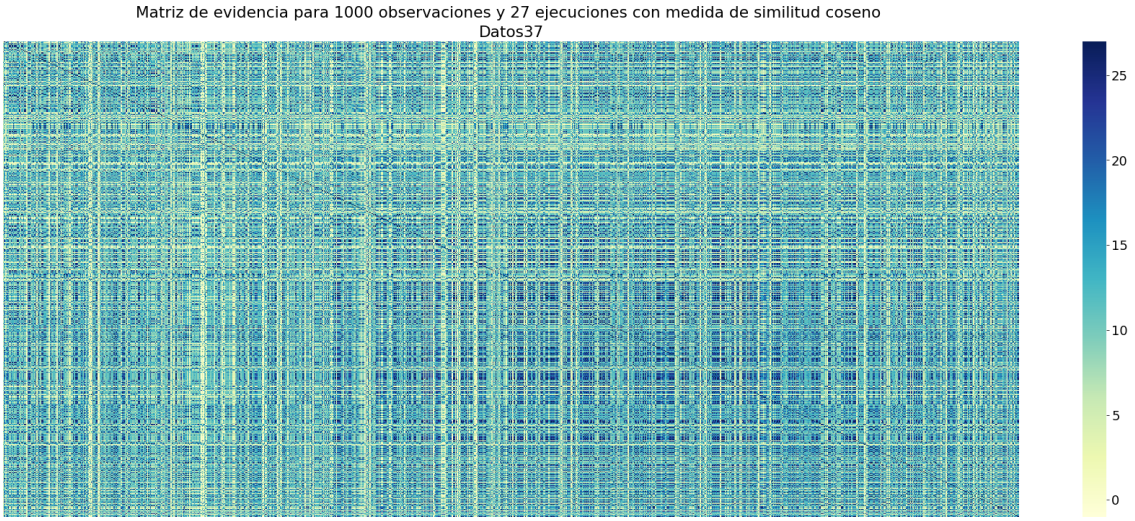


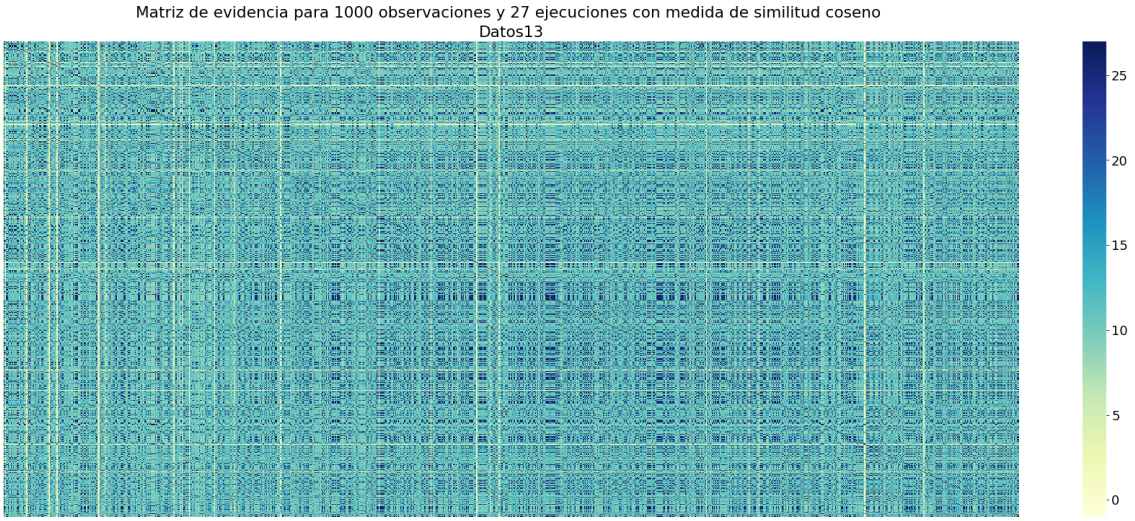




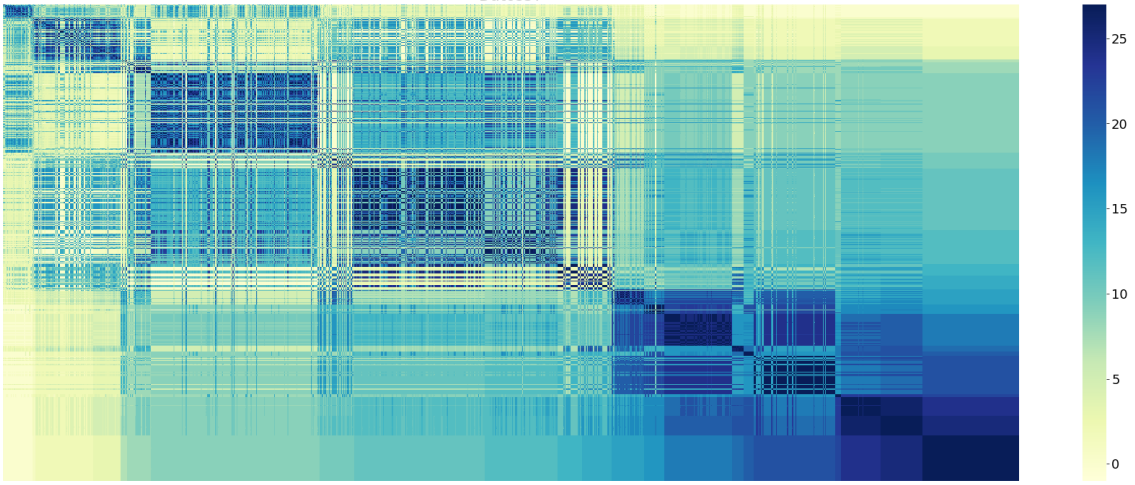


5.2. Matriz de evidencia / Distancia coseno





Matriz de evidencia para 1000 observaciones y 27 ejecuciones con medida de similitud coseno
Datos37



Matriz de evidencia para 1000 observaciones y 27 ejecuciones con medida de similitud coseno
Datos13