ESTUDIO COMPARATIVO DE DIFERENTES ALGORITMOS DE CLUSTERING PARA LA ESTIMACIÓN DE GRUPOS DE EVALUADOS QUE COMPARTEN DEBILIDADES CONCEPTUALES SIMILARES

LAURA CRISTINA SCHIATTI SISÓ

TUTOR: LUZ STELLA ROBLES PEDROZO

UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SISTEMAS

Cartagena de Indias, D.T. y C., 2017

Agradecimientos

A Dios, por su guía;

A mi familia, por su apoyo;

A mis amigos, por su compañía;

A mis profesores, por su paciencia;

Índice de contenido

Lista de Tablas

Lista de Figuras

Lista de Gráficos

Lista de Abreviaturas

AA	Aprendizaje Automático
IRT	Item Response Theory
ROC	Rank Order Clustering
WSS	Within Sum of Squares
SSE	Sum of Squares Error

1. Descripción del Proyecto

1.1. Introducción

En esta investigación se busca hacer uso de algoritmos de clustering para agrupar evaluados que compartan debilidades conceptuales similares, y posteriormente, con base en los resultados, criterios de validación, pruebas estadísticas y análisis del comportamiento de los casos atípicos, determinar cuál es el mejor algoritmo. En este estudio comparativo es importante el análisis de casos atípicos pues representan el problema que será abordado.

1.2. Objetivos

Objetivo general

Diseño de un estudio comparativo entre diferentes algoritmos de clustering para la estimación de grupos de evaluados que comparten debilidades conceptuales similares, pudiendo así determinar qué técnica realiza el mejor agrupamiento (más óptimo).

Objetivos específicos

- Identificar y seleccionar las técnicas de clustering que van a ser aplicadas en esta investigación, sobre el conjunto de datos-base para el estudio, y a partir del cual, se espera generar los distintos grupos.
- Diseñar e implementar los algoritmos de agrupamiento (ya identificados para esta investigación), que permitirán clasificar los registros sobre el conjunto de datos.
- Estimar la precisión y validez de cada algoritmo de agrupamiento de forma tal que se pueda establecer cuál generó los clusters más completos.
- Diseñar e implementar una interfaz sobre el conjunto de grupos generados por cada técnica de agrupamiento, que facilite la presentación e interpretación de la información.

1.3. Resumen de los capítulos

Este trabajo se estructura de la siguiente manera. En el capítulo 2, se realiza una contextualización de la problemática planteada en este estudio y su justificación, del mismo modo se realiza una introducción a los principales conceptos y temáticas relacionadas con algoritmos de agrupamiento (o clustering). Tal introducción servirá para familiarse con las características principales de estos métodos y comprobar el

estado del arte. Para cerrar se describe de manera detallada el algoritmo de tecnología de grupos que será adaptado para dar solución al problema plateado en esta investigación.

En el capítulo 3, se describe en detalle la metodología seguida para alcanzar cada uno de los objetivos específicos planteados. Se analizan las observaciones a partir de las cuales se obtiene el conjunto de datos-base, según la estructura de los datos se seleccionan los algoritmos de clustering que serán comparados, se implementan en lenguaje *python* o *R* -según sea el caso-, se garantiza su idoneidad por medio de una prueba estadística y por último se validan sus resultados teniendo en consideración índices de validación previamente seleccionados y comportamiento de casos atípicos.

Tras la definición de la metodología, en el capítulo 4 se explica en detalle cómo fue la implementación de los tres algoritmos seleccionados, teniendo en cuenta tratamiento previo hecho a los datos, y el paso a paso seguido en cuanto a sentencias de R utilizadas y generalidades del algoritmo Rank Order Clustering implementado en *python*.

En el capítulo 5 se detallan los procesos de validación, pruebas y análisis de resultados. En la validación se utiliza el conjunto de datos estándar llamado iris como referencia teórica que permita determinar el nivel de validez de los algoritmos implementados. En las pruebas se aplican los algoritmos al conjunto de datos-base del estudio y se analizan los resultados en cuanto a autenticidad, precisión y aciertos en cuanto al agrupamiento. Finalmente, en el capítulo 6 se presentan las conclusiones y futuras líneas de investigación abiertas a consideración.

2. Contexto y Motivación

2.1. Introducción

Este capítulo tiene como objetivo contextualizar respecto al problema y motivación que llevaron a plantear la pregunta de investigación. De igual manera, se presentan a detalle los conceptos a utilizar para el desarrollo de esta investigación, directamente relacionado con la generación de agrupamientos, como lo son medidas de similitud, diferentes algoritmos existentes utilizados para agrupar objetos, criterios para determinar su desempeño y representaciones gráficas de los resultados; todos los conceptos facilitarán la comprensión de los mismos al ser aplicados al conjunto de estudiantes que serán agrupados. Por último, se presenta el estado del arte como referencia para conocer lo que se ha hecho respecto a la temática abordada en la investigación, para evitar duplicar esfuerzos y localizar errores que fueron superados.

2.2. Contexto y Descripción del problema

En todo proceso de aprendizaje, la evaluación es definida como el mecanismo que permite determinar en qué medida se han logrado los objetivos propuestos por cada unidad de enseñanza. Para esto, se asigna una expresión cuantitativa que representa o da cuenta del desempeño del estudiante. Sin embargo, no basta con emitir únicamente un tipo de calificación, es deseable que el juicio de valor hecho, posibilite una acción o una decisión, y es ahí donde el proceso de evaluación es más significativo que el de calificación, dado que entre otras cosas, facilita el identificar las falencias conceptuales de los sujetos evaluados a un nivel más específico puesto que identifica conceptos sobre los que la persona debe tener conocimiento.

La importancia de conocer -a nivel de detalle-, en qué conceptos está fallando el estudiante, a manera de evaluación-diagnóstico, radica en que se facilitan los procesos de retroalimentación, capacitación adicional y posibilita la realización de evaluaciones personalizadas con el objetivo de corroborar si hubo avance, y se pudieron superar las dificultades encontradas y adquirir los conocimientos deseados.

Una vez se tengan identificados los conceptos débiles encontrados en los evaluados y con base en un estudio previo hecho por Robles y Rodríguez-Artacho (2012), sería interesante poder agrupar a las personas que comparten las mismas debilidades conceptuales para que los procesos de retroalimentación sean más efectivos. En el proceso, se encontrarán sujetos fácilmente agrupables y otros considerados

atípicos, cuyas falencias, si bien son en su mayoría similares a los grupos identificados, manejan conceptos que no coinciden con tales grupos. Los puntos antes expuestos permiten formular con claridad la pregunta de investigación de este proyecto: ¿cuál sería el algoritmo de clustering que va a permitir clasificar mejor a los sujetos atípicos presentados dentro de una evaluación?

2.3. Justificación

En el análisis de conglomerados o clusters se busca particionar un conjunto de objetos o grupos, de tal forma que los objetos de un mismo grupo sean similares, y los objetos de grupos diferentes no sean similares, de ahí que el objetivo principal sea colocar las observaciones más parecidas en grupos a partir de una estructura que almacene tales datos (Robles et al., 2012).

Con el claro propósito de agrupar a los evaluados que compartan características similares, porque es la manera más efectiva de impartir procesos de retroalimentación y hacer que estos sean de mucho más provecho, se hace necesario conocer las distintas técnicas algorítmicas que se encuentran documentadas, y pueden utilizarse para identificar tales grupos.

Debido a que existen diversidad de técnicas de agrupamiento, y que la aplicación de cada una está sujeta al contexto en el que se aplica, a los tipos de datos registrados, a la cantidad de los mismos, a la cantidad de variables a analizar, etc., el objetivo de este estudio se centra en establecer un comparativo que mida el rendimiento de cada algoritmo, y permita identificar cuál sería la técnica óptima para dar tratamiento al conjunto de datos base de este estudio.

Adicionalmente, y teniendo en cuenta que los datos base de este estudio son sujetos evaluados, existen algunos que, durante el proceso de retroalimentación y agrupamiento, presentan comportamientos poco convencionales, siendo considerados estos como datos atípicos. Estos evaluados, por su comportamiento, pueden no ajustarse a ninguno de los grupos ya conformados. Esta particularidad hace que el análisis de las técnicas algorítmicas, sea un poco más preciso, de tal forma que permita ubicar a los evaluados en el mejor grupo, es decir, el grupo que mejor modele su conjunto de conceptos débiles.

Partiendo de allí, se puede considerar que uno de los beneficios obtenidos al lograr agrupar correctamente a los estudiantes teniendo en cuenta sus debilidades conceptuales es que el esfuerzo que tenga que hacer el instructor, o la persona que esté encargada de cumplir con el proceso de retroalimentación, sea mucho menor o vaya en el mismo sentido -esté centralizada- y no esté disperso en cuanto a que tenga personas con niveles de conocimiento diferentes. Entonces, se tiene que, si los estudiantes están en la misma línea, agrupados correctamente, la retroalimentación hecha por el docente será más precisa y acertada, y se tendrá información de la temática a reforzar en los estudiantes que realmente tienen la carencia, gracias a lograr que no se encuentren -en un grupo-, con estudiantes que ya conocen o dominan ese conjunto de conceptos en particular.

2.4. Marco Teórico

2.4.1. Generación de conglomerados - Clustering

La Inteligencia Artificial es el campo de la computación que estudia el diseño de entidades autónomas con capacidad de razonamiento, simulando la inteligencia humana. Sólo podemos considerar que un sistema es realmente inteligente si es capaz de observar su entorno y aprender de él dado que la inteligencia reside en adaptarse, tener capacidad de integrar nuevo conocimiento, resolver nuevos problemas, aprender de errores.

Una de las ramas de la Inteligencia Artificial es el Aprendizaje Automático (AA, o Machine Learning, por su nombre en inglés). Esta rama tiene como objetivo desarrollar técnicas que permitan a los sistemas adquirir conocimiento. De forma más concreta, se refiere al estudio, diseño y análisis de los sistemas que pueden aprender, capaces de generalizar comportamientos y reconocer patrones. Las aplicaciones del Machine Learning se dan principalmente en la solución de problemas como como tareas difíciles de programar (adquisición de conocimiento, reconocimiento de caras, voz, entre otras), aplicaciones auto adaptables (interfaces inteligentes, spam filters) y minería de datos (análisis de datos inteligente) entre otras.

Las técnicas de Machine Learning se pueden clasificar según el tipo de aprendizaje:

- *Aprendizaje supervisado:* adquisición de conocimiento, por experiencia, datos del pasado y datos actuales, a través de pares de datos del tipo atributos-etiquetas.

- *Aprendizaje no supervisado:* adquisición de conocimiento a través de información por atributos, que tiene como finalidad encontrar similitudes entre los datos.
- *Aprendizaje por refuerzo:* adquisición de conocimiento basado en utilidades/castigos que a través del tiempo generan un patrón de datos.

Una forma de aprendizaje no supervisado es la generación de agrupamientos (o clustering en inglés). El propósito del clustering es detectar grupos naturales en el conjunto de observaciones, es decir, grupos acordes a interpretaciones humanas de los datos. Los grupos son determinados de tal forma que los objetos de un grupo sean similares entre sí, y los objetos de grupos diferentes sean disímiles; para medir el grado de similitud se utiliza la información de una serie de atributos (o variables) para cada objeto.

En el análisis no supervisado, el número de grupos y la posible estructura de los mismos son desconocidos a priori, es decir, para la tarea de definir a qué grupo pertenece un sujeto no se cuenta con modelos o con algún patrón de ordenamiento definido.

En todo proceso de clustering se deben considerar diferentes elementos (*ver* Figura 2.1.), estos permiten definir una metodología orientada hacia la generación de conocimiento a partir de cierto conjunto de datos.

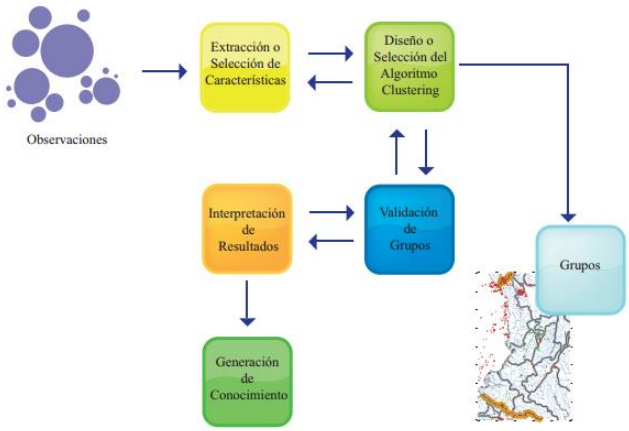


Figura 2.1. Elementos básicos de una tarea Clustering. En Reconocimiento de patrones espaciales sísmicos en el sur Occidente Colombiano mediante aprendizaje no supervisado por Duque, D., y Flórez, J., 2012.

Así, los elementos que interfieren en un proceso de clustering son:

a. Extracción o selección de características

Respecto al conjunto de datos se recomienda seleccionar y transformar las variables a utilizar, esto es, encontrar el conjunto de variables que mejor represente el concepto de similitud en el estudio que se esté desarrollando; así mismo, estandarizar los datos antes de aplicar alguna métrica o método debido a la naturaleza heurística de las técnicas de agrupamiento.

b. Diseño o selección del algoritmo de clustering

La decisión respecto al diseño o la selección del algoritmo de clustering que será utilizado se debe tomar considerando la *medida de similitud* utilizada para comparar objetos, y el *método de agrupamiento* seleccionado. Estos conceptos se definen así:

- Medidas de similitud. En el análisis de clusters, el significado de similitud suele tomarse como el nivel de proximidad entre elementos, y los casos del conjunto de datos más cercanos se consideran más similares. Así, las medidas de similitud son las encargadas de reconocer como semejantes, o diferentes los objetos que serán agrupados.

  Dentro de las medidas de similitud se encuentran medidas de distancia, coeficientes de correlación, coeficientes de asociación, y medidas probabilísticas de similitud. Considerando las características de las variables con las que se trabaja en esta investigación (se cuenta con un archivo que contiene conceptos débiles para cada estudiante), y los objetivos que se persiguen, se trabaja con *medidas de distancia*.

  Las medidas de distancia de uso más frecuente son las distancias Minkowski, Manhattan, Euclidiana, y Mahalanobis, entre otras. Se determinó trabajar con la distancia Euclidiana en este estudio debido a que es la métrica más conocida, sencilla de comprender e implementar, y la más recomendable cuando las variables, como en este caso, son homogéneas, o están medidas en las mismas unidades (nivel de pertenencia de un atributo o variables, en este caso concepto débil).

La distancia Euclidiana es una medida matemática estándar de distancia (raíz cuadrada de la suma de las diferencias elevadas al cuadrado), está definida por:

$$d(i,j) = \sqrt{\sum_{k=1}^{p}(\mathrm{X}_{ik} - \mathrm{X}_{jk})^2} \qquad (1)$$

Donde $x_i{}' = (x_{i1}, \dots, x_{ip})$ y $x_j{}' = (x_{j1}, \dots, x_{jp})$ son las observaciones de dos objetos o individuos i y j resultado de medir p variables numéricas $x_1, \dots, x_p$ sobre ellos.

- **Métodos de agrupamiento.** Los métodos de agrupamiento son procedimientos mediante los cuales se agrupan los elementos que son más similares dentro de un determinado grupo. Son varios los algoritmos propuestos por diferentes autores, cada uno representa una perspectiva diferente para la formación de los clusters, con resultados generalmente distintos sobre el mismo conjunto de datos. Decidir qué método utilizar para un problema específico depende de las variables consideradas, de la medida de similitud elegida, y de la tipología de resultados que se espera. La clasificación de los métodos de agrupamiento se ilustra a detalle en la Figura 2.2.

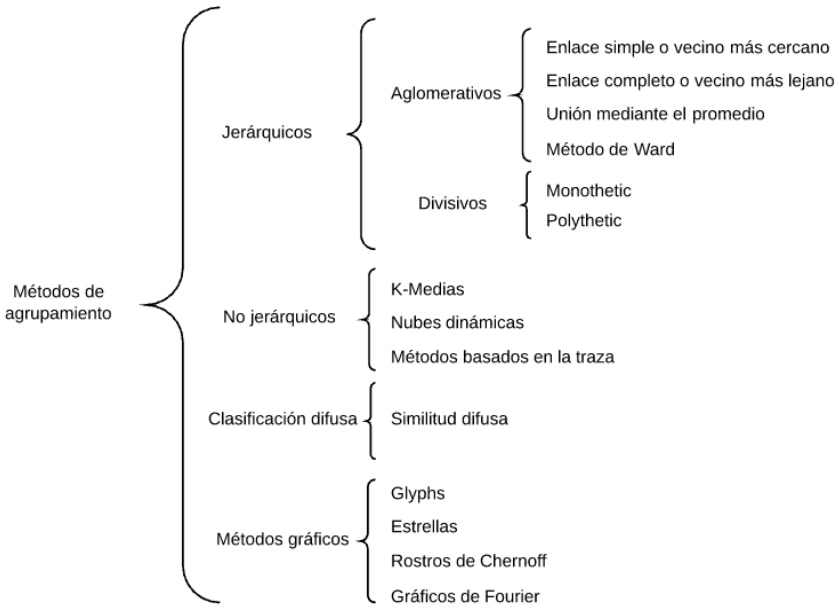


Figura 2.2. Cuadro sinóptico de clasificación de métodos de agrupamiento.

Métodos jerárquicos

Los métodos jerárquicos son, probablemente, los que más han sido desarrollados, y generan una descomposición jerárquica del conjunto de datos. Un método jerárquico puede ser clasificado como *aglomerativo* o *divisivo*, basado en cómo se forman los grupos. En los procedimientos aglomerativos cada uno de los objetos empieza formando un conglomerado, y luego se mezclan sucesivamente hasta que todos los objetos quedan dentro de un mismo conglomerado, y en los métodos de división, lo contrario.

Dentro de los métodos jerárquicos se encuentra el *método de Ward* (o de varianza mínima, 1963). Este método se basa en la pérdida de información que se produce al integrar los distintos individuos en clusters; puede medirse a través de la suma total de cuadrados de las desviaciones entre cada punto (individuo) y la media del grupo en el que se integra.

Se seleccionó el método de Ward por ser uno de los más utilizados en la práctica, pues tiene casi todas las ventajas del método de la media, y suele ser más discriminativo en la determinación de los niveles de agrupación. En comparación con los demás métodos aglomerativos, se tiene de acuerdo a investigaciones (e.g. Kuiper y Fisher, 1975), que Ward es capaz de acertar mejor con la clasificación óptima que otros métodos, y que es menos sensible a datos atípicos.

Para que el proceso de agrupamiento resulte óptimo, en el sentido de que los grupos formados no distorsionen los datos originales, Ward propuso que, en cada paso del análisis, se considere la posibilidad de la unión de cada par de grupos, y optar por la fusión de aquellos dos grupos que menos incrementen la suma de los cuadrados de las desviaciones al unirse.

La suma de cuadrados de Ward se calcula mediante:

$$SWC = \frac{1}{\left({}^{1}/_{n_h} + {}^{1}/_{n_k}\right)} \|\overline{\mathrm{X}}_h - \overline{\mathrm{X}}_k\|^2 \qquad (2)$$

Donde:

$\overline{\mathrm{X}}_h \; y \; \overline{\mathrm{X}}_k$ son los centroides.

$n_h \; y \; n_k$ son los tamaños de los grupos h y k respectivamente.

A manera de ejemplo, muy similar a la tipología de los casos que se trabajan en esta investigación, se presenta un caso de 5 individuos sobre los cuales se mide un atributo, ver Tabla 2.1.

Individuo	Atributo
A	3
B	7
C	8
D	11
E	14

Tabla 2.1. Caso-Ejemplo método de Ward con 5 individuos y un atributo.

El procedimiento en, por ejemplo, las tres primeras etapas sería:

Primera etapa: la *SCW* para cada uno de los individuos es cero. Los grupos iniciales son:

{A}, {B}, {C}, {D} y {E}

Segunda etapa: se deben organizar los cinco objetos en grupos de dos elementos cada uno, para esto se calcula una combinación de $\binom{5}{2}$ que es igual a diez posibles grupos; así, se producen las siguientes sumas de cuadrados:

$SCW_{\{A,B\}} = (3^2) + (7^2) - \frac{1}{2}(3+7)^2 = 8$ $\quad$ $SCW_{\{A,C\}} = 12.5$

$SCW_{\{A,D\}} = 32$ $\quad$ $SCW_{\{A,E\}} = 60.5$

$SCW_{\{B,C\}} = 0.5$ * $\quad$ $SCW_{\{B,D\}} = 12$

$SCW_{\{B,E\}} = 24.05$ $\quad$ $SCW_{\{C,D\}} = 5.5$

$SCW_{\{C,E\}} = 18$ $\quad$ $SCW_{\{D,E\}} = 4.5$

De acuerdo con los anteriores cálculos, los individuos B y C son fusionados porque tienen la menor *SCW* (marcada con *). Los grupos quedan organizados así:

{A}, {B, C}, {D} y {E}

Tercera etapa: se calcula la *SCW* para cada uno de los posibles agrupamientos $\binom{4}{2}$, entre los cuatro grupos encontrados en el paso anterior; resulta:

$SCW_{\{A,\},\{B,C\}} = (3^2 + 7^2 + 8^2) - \frac{1}{3}(3 + 7 + 8)^2 = 14$ $\quad SCW_{\{A,D\}} = 32$

$SCW_{\{A,E\}} = 60.5$ $\quad SCW_{\{D\}.\{B,C\}} = 8.67$

$SCW_{\{E\},\{B,C\}} = 28.67\ *$ $\quad SCW_{\{D,E\}} = 4.5\ *$

El grupo que registra la mayor homogeneidad es el conformado por D y E, ya que la fusión de éstos dos objetos produce la menor variabilidad. Los grupos que se han formado hasta aquí son:

{A}, {B, C} y {D, E}

En los pasos siguientes se continúa con los cálculos para tres grupos de dos individuos, etc.

Los resultados de un agrupamiento con métodos jerárquicos se pueden registrar en un diagrama en forma de árbol denominado *dendrograma* o *árbol de clasificación*. Un dendograma es una representación gráfica en forma de árbol que resume el proceso de agrupación jerárquica en un análisis de clusters.

El dendrograma de la Figura 2.3. muestra un ejemplo de la disposición de los objetos en cada uno de los conglomerados. En general, si se corta el dendograma mediante una línea horizontal, se determina el número de clusters en que se divide el conjunto de objetos. El eje vertical contiene los niveles de distancia bajo los cuales se conforman los grupos; así, para una distancia de 5.2 se tienen dos grupos (bajo la línea horizontal), estos son {1, 2} y {3, 4, 5}. Si se corta a una distancia de 2.5, se obtendrían tres clusters.

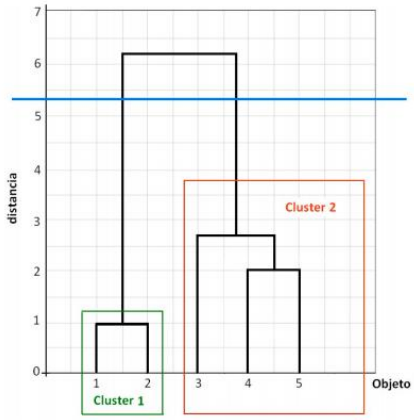


Figura 2.3 Dendrograma con dos grupos. En análisis conglomerados por de la Fuente, S., 2011..

Métodos no jerárquicos

A diferencia de los métodos de agrupación jerárquicos, los métodos no jerárquicos no han sido muy empleados o examinados; razón por la que se aplican e interpretan, a veces, de una manera poco correcta. Asimismo, exigen definir previamente el número de clusters y poseen algunos índices que indican el número óptimo de conglomerados.

En detalle, *k-medias* (o k-means en inglés) es un método que tiene como objetivo agrupar o clasificar un conjunto de m observaciones u objetos en k grupos, asignando cada objeto al cluster (de los k prefijados) con el centroide más próximo. Este método es el más sencillo y eficiente computacionalmente hablando; de igual manera, es fácil de implementar incluso para grandes conjuntos de datos, y consume poca memoria pues procesa los patrones secuencialmente.

Los tres algoritmos seleccionados para este estudio son:

b1. Algoritmo Jerárquico - Distancia Euclidiana - Método de Ward

Este algoritmo tiene en cuenta al momento de agrupar, distancia Euclidiana como medida de similitud y como método de agrupamiento el método de Ward.

El algoritmo jerárquico se describe formalmente como sigue:

Algoritmo 1. Jerárquico – Distancia Euclidiana - Método de Ward

1 - Cargar fichero de casos
2 - Calcular matriz de distancias usando distancia Euclidiana (ecuación 1)
3 - Fijar n clusters iniciales con SCW igual a 0
4 - **repetir**
a. Determinar todas las combinaciones sin repeticiones de pares de grupos c_i y c_j
b. Para cada par de grupos c_i y c_j calcular SCW usando la ecuación 6
c. Encontrar el menor SCW de los calculados
d. Fusionar los grupos c_i y c_j en un único grupo
5 - **hasta** todos los casos formen un único grupo

b2. Algoritmo Rank Order - Distancia Euclidiana - Método de Ward

El algoritmo Rank-Order Clustering (en adelante ROC), también llamado ordenamiento binario o algoritmo de King, en honor a su autor quien lo presentó en 1979, busca agrupar teniendo en cuenta características comunes. Hace parte de los algoritmos de tecnología de grupos que permiten la formación de grupos de máquinas y asignación de familias de partes a algunas de estas.

La tecnología de grupos es una práctica que surgió en diferentes áreas de producción, como el diseño y la manufactura, y busca agrupar piezas en familias con características geométricas, de procesamiento y de material similares. Se basa en disminuir el movimiento de piezas (de producción) al formar grupos de máquinas, capaces de procesar una familia de partes o componentes específica, con base en la información sobre la secuencia de operaciones que siguen dichas partes (Cascante y Coronado, 2007).

ROC busca manipular una matriz de incidencia hasta crear zonas cuyas posiciones contengan 1s (unos). Las matrices de incidencia son matrices binarias (sus elementos sólo pueden ser unos o ceros) empleadas para representar relaciones binarias entre dos elementos. En problemas de tecnología de grupos son usadas para relacionar las máquinas con los componentes a producir.

El algoritmo es descrito formalmente a continuación:

Algoritmo 2. Rank Order Clustering -Distancia Euclidiana - Método de Ward

1 - Leer fichero con matriz de incidencia
2 - **Para** i <= número de filas
a. calcular los pesos w_i de la fila.
3 - Ordenar las filas de forma descendente con base en el valor de w_i.
4 - **Para** j <= número de columnas
b. calcular los pesos w_j de la fila.
5 - Ordenar las columnas de forma descendente con base en el valor de w_j.

A manera de ejemplo sobre cómo opera este algoritmo se tiene un problema de 5 máquinas (A, B, C, D, E) y 6 partes (1, 2, 3,... , 6), como se muestra en la Tabla 2.2.

		Partes					
Máquina		1	2	3	4	5	6
	A			1		1	
	B		1	1			
	C	1			1		
	D		1	1		1	
	E	1			1		1

Tabla 2.2. Caso-Ejemplo algoritmo Rank Order con 5 máquinas y 6 partes.

El algoritmo sigue el siguiente procedimiento:

Paso 1: Se calculan los pesos (Wi) de las filas (*ver* Tabla 2.3).

		Partes						Wi
Máquinas		1	2	3	4	5	6	
	B. Wt:	2^5	2^4	2^3	2^2	2^1	2^0	
	A			1		1		$2^3+2^1 = 10$
	B		1	1				$2^4+2^3 = 24$
	C	1			1			$2^5+2^2=36$
	D		1	1		1		$2^4+2^3+2^1 = 26$
	E	1			1		1	$2^5+2^2+2^0= 37$

Tabla 2.3. Caso-Ejemplo Rank Order cálculo de los Wi por fila.

Se ordenan las filas en forma descendente, de acuerdo con su respectivo peso (Wi). Para esto se asignan valores de 1 a 5 a cada fila, colocando 1 al mayor Wi y 5 al menor; y se intercambian las filas de manera que queden en orden (*ver* Tabla 2.4).

		Partes						Wi	Orden
Máquinas		1	2	3	4	5	6		
	B. Wt:	2^5	2^4	2^3	2^2	2^1	2^0		
	A			1		1		$2^3+2^1 = 10$	5
	B		1	1				$2^4+2^3 = 24$	4
	C	1			1			$2^5+2^2=36$	2
	D		1	1		1		$2^4+2^3+2^1 = 26$	3
	E	1			1		1	$2^5+2^2+2^0= 37$	1

Tabla 2.4. Caso-Ejemplo Rank Order ordenar filas en forma descendente.

Paso 2: se calculan los pesos (Wj) de las columnas (*ver* Tabla 2.5)

			Partes					
		B. WT.	1	2	3	4	5	6
Máquinas	E	2^4	1			1		1
	C	2^3	1			1		
	D	2^2		1	1		1	
	B	2^1		1	1			
	A	2^0			1		1	
Wj			$2^4+2^3 = 24$	$2^2+2^1= 6$	$2^2+2^1+ 2^0=7$	$2^4+2^3= 24$	$2^2+2^0= 5$	$2^4=16$

Tabla 2.5. Caso-Ejemplo Rank Order cálculo de los Wj por columna.

Se ordenan las columnas en forma descendente, de acuerdo con su respectivo peso (Wj). Para esto se asignan valores de 1 a 6 a cada columna, colocando 1 al mayor Wj y 5 al menor; y se intercambian las columnas de manera que queden en orden (*ver* Tabla 2.6).

			Partes					
		B. WT.	1	2	3	4	5	6
Máquinas	E	2^4	1			1		1
	C	2^3	1			1		
	D	2^2		1	1		1	
	B	2^1		1	1			
	A	2^0			1		1	
Wj			2^4+2^3 = 24	2^2+2^1= 6	2^2+2^1+ 2^0=7	2^4+2^3= 24	2^2+2^0= 5	2^4=16
Orden			1	5	4	2	6	3

Tabla 2.6. Caso-Ejemplo Rank Order ordenar columnas en forma descendente.

Luego de ordenar las columnas finaliza el algoritmo. Los resultados se ilustran en la Tabla 2.7. Se pueden apreciar dos grupos claramente definidos.

		Partes					
Máquinas	E	1	1	1			
	C	1	1				
	D				1	1	1
	B				1	1	
	A				1		1

Tabla 2.7. Caso-Ejemplo Rank Order resultados.

Limitaciones

Entrando en detalle sobre el desempeño de este algoritmo cabe destacar que funciona correctamente en caso de tener un número no tan grande de elementos en la matriz, pues se hace ineficiente en el momento de calcular los pesos correspondientes, pues dependen de potencias muy altas. Sin embargo, para efectos de problemas de agrupamiento de estudiantes, como el tratado en este estudio, funciona correctamente. A partir de pruebas realizadas se pudo fijar -para la matriz de incidencia- como máximo contar con 500 filas o 236 columnas, porque conlleva calcular potencias considerables, del orden de 2^{500} y 2^{236} respectivamente. Se puede resaltar que no es recomendable usar una matriz con dimensiones

iguales a los valores máximos antes sugeridos, debido a que implica recibir una alerta[1], y en ciertas ocasiones, resultados erróneos.

Además, por ser esencialmente un algoritmo de ordenamiento, con sus resultados solo se puede determinar -de manera intuitiva- los grupos generados con solo mirar la matriz final, esto significa que ROC no cuenta con un procedimiento ya establecido que particione y muestre explícitamente los grupos generados, es decir, la identificación de grupos y qué datos los conforman.

b3. Algoritmo K-means de MacQueen

Existe un poco de confusión en la literatura debido a que algunos autores se refieren a k-medias como un algoritmo específico, en lugar de tratarlo como un método general. Los algoritmos que implementan k-medias emplean variaciones de un proceso central que apenas muestran diferencias en el campo computacional, y difieren en cuanto a las características del conjunto de datos, los objetivos perseguidos y los resultados esperados. Los algoritmos más conocidos son los propuestos por Lloyd (1957), Forgy (1965) y MacQueen (1967).

Por un lado, los algoritmos de Lloyd y Forgy se caracterizan por funcionar correctamente para distribuciones de datos grandes, y por seleccionar aleatoriamente k casos, como los centroides iniciales; no obstante, necesitan almacenar los resultados de las últimas dos iteraciones (costoso para conjuntos de datos numerosos), y es posible, dadas ciertas condiciones, crear clusters vacíos. Ambos procedimientos se diferencian únicamente por el tipo de distribución de los datos de entrada, Lloyd funciona para distribuciones de datos discretas, y Forgy para continuas.

Por otro lado, el algoritmo de MacQueen se caracteriza por tomar los k primeros casos como centroides (uno para cada grupo), y por recalcular el centroide de un cluster inmediatamente después de que le sea asignado un caso, y no al final de cada ciclo, como ocurre en los otros algoritmos; esta última diferencia

[1] Too much output to process

es la principal ventaja de MacQueen, pues es más eficiente en cuanto a que actualiza los centroides frecuentemente, y recorre todos los casos antes de converger en una solución.

En este estudio, se decidió implementar el algoritmo de MacQueen considerando las ventajas mencionadas previamente, y debido a que tiene la virtud de ser el menos caro de los algoritmos en cuanto a cómputo total de operaciones, desde la configuración inicial hasta la final involucra sólo *k(2m−k)* cálculos de distancias, *(k−1)(2m−k)* comparaciones de distancias, y *m−k* cálculos de centroide siendo *m* el número de casos y *k* el número de clusters en que serán agrupados; asimismo es el algoritmo de k-medias más utilizado.

El algoritmo se describe formalmente como sigue:

Algoritmo 3. k-means de MacQueen

1 - Escoger el número de grupos
2 - Leer fichero de casos
3 - Determinar método para seleccionar los k centroides iniciales
4 - Asignar centroides iniciales
5 - **Mientras** criterio de parada **hacer**
Para i <= número de casos
a. Asignar el caso i al grupo más cercano (con mínima distancia euclidiana usando la ecuación 1)
b. Recalcular los centroides de los dos grupos afectados
6 - Recalcular centroides de todos los grupos

En el algoritmo se debe tener un determinado criterio de parada; en el caso de MacQueen se finaliza el algoritmo cuando no se produzca ninguna reasignación, es decir, hasta que los elementos logren estabilizarse en algún grupo. Otros criterios de parada pueden ser: un número máximo de iteraciones, el que los nuevos centroides disten de los centroides obtenidos en la iteración previa menos que una determinada distancia, o minimizar el error cuadrático medio.

Como un ejemplo simple del algoritmo, se cuenta con el conjunto de datos mostrado en la Tabla 2.8 que consiste en los valores de dos variables (A y B) para siete individuos (1, 2, …, 7).

Individuos	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

Tabla 2.8. Caso-Ejemplo algoritmo de MacQueen con 7 individuos y 2 variables.

Para este ejercicio, se desea agrupar a los individuos en 2 clusters. Como primer paso, se toman los dos primeros individuos -1 y 2- como los centroides de A y B respectivamente (*ver* Tabla 2.9).

	Individuos	Centroide
Grupo 1	1	(1.0, 1.0)
Grupo 2	4	(5.0, 7.0)

Tabla 2.9. Caso-Ejemplo MacQueen estructura de grupos iniciales.

A continuación, los individuos restantes son examinados en secuencia y asignados al cluster al que son más cercanos, en términos de distancia Euclidiana. El centroide de un cluster es recalculado cada vez que un nuevo miembro es adicionado. Esto conduce a los pasos mostrados en la Tabla 2.10:

	Cluster 1		Cluster 2	
Paso	Individuos	Centroide	Individuos	Centroide
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

Tabla 2.10. Caso-Ejemplo MacQueen primeras 6 iteraciones.

Ahora la partición inicial cambió, y los clusters tienen las características evidenciadas en la Tabla 2.11:

	Individuos	Centroide
Grupo 1	1, 2, 3	(1.8, 2.3)
Grupo 2	4, 5, 6, 7	(4.1, 5.4)

Tabla 2.11. Caso-Ejemplo MacQueen estructura de grupos.

En este punto no se tiene certeza de que los individuos se encuentren asignados en el cluster correcto. Así, se debe comparar la distancia de cada individuo al grupo en el que se encuentra con la del otro y se obtiene la Tabla 2.12:

Individuo	Distancia al centroide del Grupo 1	Distancia al centroide del Grupo 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Tabla 2.12. Caso-Ejemplo MacQueen distancias de cada individuo a cada cluster.

La distancia de un individuo al grupo en que se encuentra debe ser menor a la distancia respecto al otro cluster. Teniendo en cuenta la Tabla 2.12 se sabe que el individuo 3 es más cercano al cluster opuesto (grupo 2) que al grupo en que se encuentra, entonces, este es reubicado en el grupo 2.

	Individuos	Centroide
Grupo 1	1, 2	(1.3, 1.5)
Grupo 2	3, 4, 5, 6, 7	(3.9, 5.1)

Tabla 2.13. Caso-Ejemplo MacQueen estructura final de grupos.

La reubicación de individuos se repite iterativamente hasta que no sucedan más reubicaciones. Sin embargo, en este ejemplo cada individuo en este punto se encuentra en el grupo más cercano por tanto la solución final es la representada por la Tabla 2.13.

Considerando las diferentes medidas de similitud, los métodos de agrupamiento y la tecnología de grupos, se decidió para este estudio implementar los tres algoritmos descritos, porque tienen diferentes maneras de agrupar y diferentes rendimientos, en cuanto a aciertos respecto al agrupamiento, y por tanto alguno debe ser el mejor.

c. Validación de grupos

Una de las etapas clave en todo proceso clustering es la validación de los resultados, porque permite evaluar cuantitativamente el nivel de precisión y calidad de la agrupación. Una de las motivaciones es que casi todos los algoritmos de clustering encontrarán clusters en el conjunto de datos, aun cuando este conjunto no contenga clusters naturales en su estructura. Este procedimiento es una herramienta de apoyo al experto que le permita tomar decisiones más objetiva sobre la evaluación de particiones óptimas en un conjunto de datos.

Las medidas de evaluación que se aplican para verificar la validez de los clusters y se clasifican en:

- Medidas de validación Interna. La validación interna no precisa el conocimiento a priori de la partición correcta, esta consiste en estudiar los datos y cómo se agrupan. Es decir, evalúa la partición a partir de los datos y las distancias entre ellos. La ventaja principal de este tipo de técnicas es que no requiere el conocimiento de la partición correcta por lo que se puede utilizar en aplicaciones reales y no sólo en experimentos de laboratorio.

  Las medidas de validación interna tienen como base los criterios de *cohesión* y *separación.* La cohesión determina lo cercanos que están los objetos dentro del cluster; y la separación, lo distinto y bien separado que está un cluster con respecto a otro. Estas medidas pueden ser usadas para escoger el mejor algoritmo de clustering y determinar el número óptimo de grupos. Para este estudio se consideraron las métricas descritas en la Tabla 2.14.

Métrica	Descripción	Interpretación
Índice de Dunn	Mide separación entre clusters. Es la relación entre la menor distancia entre las observaciones que no están en el mismo grupo, y la mayor distancia entre los grupos.	Tiene un valor entre cero e infinito, y debe ser lo más alto posible. Un valor más alto indica un mejor rendimiento del algoritmo de clustering.
Índice de Silhouette	Este índice corresponde al promedio del valor Silhouette de cada observación y mide el grado de confianza en la asignación de clusters de una observación en particular.	Los valores próximos a 1 gozarán de una mayor confianza en la agrupación realizada, por el contrario, los valores próximos a -1 significan que la agrupación no es fiable.

Tabla 2.14. Métricas de validación interna de clusters.

- Medidas de validación Externa. Para utilizar validación externa se debe conocer previamente la partición correcta de los datos. En general, en un contexto real a priopi no se conoce la partición correcta, por lo cual este tipo de validación solo sirve en un contexto experimental para evaluar y comparar algoritmos de clustering. El problema de este tipo de técnicas es que no siempre se puede definir una única partición correcta, esto debido a que los clusters pueden estar solapados.

  Con respecto a las medidas externas, se dispone principalmente de tres tipos para realizar la comparación entre soluciones: las basadas en el conteo de pares de patrones, en las que las soluciones coinciden o no; las basadas en el análisis de correspondencia de conjuntos, y las basadas en la estadística y teoría de la información.

  Para este estudio se determinó aplicar una prueba estadística chi-cuadrado:

  c1. Prueba estadística chi-cuadrado

  Un parámetro es un dato considerado como necesario para analizar y evaluar situaciones de un contexto real. Es el parámetro el que hace posible entender comportamientos, decisiones, y

generalizar sus efectos a futuro. Por ejemplo: “Si nos basamos en los parámetros habituales, resultará imposible comprender esta situación”, “El paciente está evolucionando de acuerdo a los parámetros esperados”, “Estamos investigando, pero no hay parámetros que nos permitan establecer una relación con el caso anterior”, “La actuación del equipo en el torneo local es el mejor parámetro para realizar un pronóstico sobre su participación en el campeonato mundial”.

Bajo distintos contextos, el parámetro tiene su particular significado:
En Matemáticas, un parámetro es una variable que representa un valor numérico.
En Estadística, Un parámetro estadístico equivale a un conjunto de valores que representan estados de una población, y permiten modelar la realidad.
En Ciencias de la Computación, un parámetro es una variable. En general las variables pueden venir de una función o de un procedimiento y que cumplen con un conjunto de especificaciones para ejecutar un programa.

En Estadística Inferencial, se trabaja con métodos paramétricos porque ellos están soportados en conceptos y datos tomados a partir de muestreo de una población, con parámetros específicos, como son: la media (μ), la desviación estándar (σ), o la proporción (p). Los métodos paramétricos, deben ajustarse a algunas condiciones estrictas (supuestos), como el que los datos estén normalmente distribuidos.

Así mismo, en Estadística Inferencial, se trabaja con métodos no paramétricos, estos métodos no requieren supuestos. Este tipo de pruebas no presuponen una distribución de probabilidad para los datos, porque son los datos observados los que determinan la distribución.

Casos en los que los métodos no paramétricos son utilizados:

- Algunos experimentos producen mediciones de respuesta que son difíciles de cuantificar.
- Se generan mediciones de respuesta que, aunque se pueden clasificar en categorías, la ubicación de la respuesta es una escala arbitraria.

En el análisis de los datos base de este estudio, es importante resaltar que:

- El tipo de datos a cualificar y/o cuantificar.
- No se cuenta con una clase de salida.
- La cantidad de características o variables de análisis, por cada registro, es variable.

A partir de las definiciones explicadas en este numeral, y del análisis de datos realizado, la prueba estadística que permitirá detectar diferencias entre el rendimiento de los algoritmos de clustering desarrollados en esta investigación, es un procedimiento estadístico no paramétrico. A lo anterior hay que agregar que, no se tiene certeza acerca del cumplimiento de supuestos, tales como el de normalidad de los datos base de este estudio (entre otros supuestos).

Dentro de los métodos no paramétricos, son varias las pruebas estadísticas (no paramétricas), que pueden ser utilizadas para realizar un experimento. Los más utilizados son:

- La Chi-cuadrado ($\mathcal{X}^2$).
- Los Coeficientes de Correlación en Independencia para Tabulaciones Cruzadas.
- Los Coeficientes de Correlación por Rangos Ordenados de Spearman y Kendall.

La Chi-cuadrado ($\mathcal{X}^2$)

- Es una prueba estadística para evaluar hipótesis acerca de la relación entre dos variables categóricas.
- Sirve para probar Hi correlacionales.
- Mide variables nominales u ordinales (o intervalos o razón reducidas a ordinales).

2.4.2. Selección del número óptimo de grupos: *método del codo*

La decisión sobre el número óptimo de clusters es subjetiva, especialmente cuando se incrementa el número de objetos pues si se seleccionan pocos, los clusters resultantes son heterogéneos y artificiales, mientras que si se seleccionan demasiados, la interpretación de los mismos suele resultar complicada.
Ciertos métodos de agrupamiento, como *k-means* recibe como parámetro de entrada el *número de grupos* en que serán divididos los elementos; para esto el experto necesita, -de alguna forma-, determinar

el número correcto de clusters según el conjunto de datos del estudio. A pesar de que no existen métodos totalmente exactos, se han implementado algunos que nos ayudan a elegir un número apropiado como el método del codo (o elbow method en inglés).

La idea del método del codo es ejecutar algún algoritmo que haga uso del método de k-means, en nuestro caso MacQueen, sobre el conjunto de datos para un rango de valores de k (k de 1 a 15 para este caso), y para cada valor de k calcular la suma de errores cuadrados (SSE en inglés).

A continuación, se genera un gráfico que relacione el SSE para cada valor de k. El número óptimo de grupos es entonces aquel punto que no varíe respecto al siguiente en cuanto al SSE. La idea es que queremos un pequeño SSE, pero que el SSE tiende a disminuir hacia 0 a medida que aumentamos k (el SSE es 0 cuando k es igual al número de puntos de datos en el conjunto de datos, porque cada punto de datos es su propio Cluster, y no hay error entre él y el centro de su clúster). Por lo tanto, la meta es elegir un pequeño valor de k que todavía tiene un SSE bajo, y el codo generalmente representa donde comenzamos a tener rendimientos decrecientes al aumentar k.

2.4.3. Generación de conglomerados en R

R es un poderoso entorno y lenguaje de programación creado para el análisis estadístico y gráfico de datos, es por esto que cuenta con un amplio abanico de herramientas que pueden ser usadas a través de comandos por consola. Para efectos de este proyecto, se hará uso de las funciones que nos brinda R relacionadas con análisis de clusters.

Se explicarán las funciones de R, y para conocer los argumentos que las funciones reciben, específicamente aquellos que serán usados para el análisis en este estudio se puede consultar el *Anexo I*, los demás argumentos de las funciones pueden ser consultados en la documentación oficial de R.

2.5. Estado del Arte

Para conocer el estado de aprendizaje -o nivel de conocimiento- de un evaluado, son utilizados como instrumentos los tests o pruebas, cuyos resultados se representan con una puntuación; sin embargo, no

es suficiente presentar al evaluado únicamente ese puntaje, y es por esto que son varias las investigaciones que han hecho aportes a este tema, desde diversas áreas en las que hacen uso, por ejemplo, de algunas técnicas de inteligencia artificial. Cabe destacar que dentro de los aportes hechos se encuentran la posibilidad de presentar a los evaluados y a los docentes, sugerencias o recomendaciones personalizadas, a partir de la prueba aplicada, y referida a los temas y conceptos evaluados en el test.

En aras de optimizar los procesos de aprendizaje, la evaluación moderna busca diagnosticar hasta que nivel entienden los estudiantes y conocer sus habilidades basándose en su *capacidad de aprendizaje* y no en una calificación (Chang, Chen, Li & Chiu, 2009). Para calcular la capacidad de aprendizaje los autores proponen un test en línea, combinado con la Teoría de Respuesta al Ítem (IRT, por sus siglas en inglés). Teniendo los resultados de la prueba, se usó el algoritmo K-Means para crear clusters, o grupos, de estudiantes con habilidades similares. Dicha clasificación será de gran utilidad para los docentes, pues estos pueden modificar el material de aprendizaje, y enseñar a estudiantes de acuerdo a sus aptitudes en los cursos, además de crear cursos nivelatorios que permitan aumentar la efectividad del proceso.

Por su parte, el trabajo de Wook, M., Wahab, N., Awang, N., Yahaya, Y., Mohd, M., & Yann, H. (2009) expone una propuesta que permite clasificar estudiantes de acuerdo a su desempeño académico utilizando dos técnicas de minería de datos, estas son Redes Neuronales Artificiales (RNA, por sus siglas en inglés) y una combinación de clustering y árboles de decisión. En cuanto a clustering se utilizó el algoritmo de K-Means; sin embargo, este algoritmo no tiene reglas específicas para definir los grupos y es por esto que se utilizaron sus resultados como entrada para la clasificación usando árbol de decisión que facilitan representar de forma más definida los grupos. La comparación de resultados probó que las RNA producen resultados precisos respecto al rendimiento.

Dalton, L., Ballarin, V., & Brun, M. (2009) destacan que es importante tener en cuenta que la selección de un algoritmo de clustering no es una decisión que se pueda tomar a la ligera pues este debe ser escogido con base en la naturaleza del problema, las características de los objetos que serán analizados, los grupos esperados y el tamaño del problema asi como el poder de computo disponible para su solución.

Debido a que los algoritmos de clustering generalmente varían resultan en diferentes conjuntos de generación de clusters, es importante evaluar el desempeño de tales métodos en términos de precisión y validez de los grupos; para esto, se cuenta con indices y criterios disponibles, de los cuales se deben seleccionar los apropiados para el problema y los objetivos perseguidos con el agrupamiento (Ansari, Z., Azeem, M.F., Ahmed, W., & Babu, A.).

2.6. Conclusiones

En resumen, los algoritmos de clustering reciben un conjunto de datos como entrada y mediante un proceso no supervisado lo particionan en cierto número de clusters o grupos. Se puede definir un cluster como un grupo de objetos homogéneos que poseen alguna medida de similitud entre ellos y que se muestran diferentes a los objetos agrupados en otros clusters. Esta técnica será utilizada para crear grupos de evaluados con conceptos débiles similares, como primer paso para generar procesos de retroalimentación más efectivos.

3. Metodología

3.1. Introducción

En este capítulo se describe en detalle la metodología de trabajo a seguir en este proyecto, es decir, los pasos y lineamientos generales seguidos a lo largo de la investigación; conocerlos a detalle facilita el proceso de implementación y análisis de resultados pues se sabe de dónde partir y a dónde se debe llegar.

3.2. Detalles del procedimiento propuesto

El objetivo principal de esta investigación se centra en establecer un comparativo entre diferentes algoritmos de clustering para la estimación de grupos de evaluados que comparten debilidades conceptuales similares, pudiendo así determinar qué técnica realiza el mejor agrupamiento. Es por esto que, para la realización de esta investigación, se diseñarán e implementarán diferentes algoritmos de agrupamiento que tienen como función, generar clusters a partir de los datos-base de trabajo, y de esta forma analizar la precisión con la que cada algoritmo realiza la clasificación de los registros. Una vez realizado este análisis, se espera identificar la técnica que mejor agrupa, y que pueda -este estudio-, ser base para proponer o sugerir aspectos a tener en cuenta para procesos de retroalimentación y nivelación de estudiantes.

En esta primera parte del estudio comparativo, se hará una revisión para la selección de las posibles técnicas de clustering que pueden ser utilizadas y probadas en esta investigación, teniendo en cuenta el tipo de datos que se tiene, así como la cantidad y el contexto en el que se aplica este estudio. Esta revisión se tendrá en cuenta técnicas de Aprendizaje no Supervisado (Machine Learning). Esta primera parte incluye el tratamiento que debe hacerse sobre los datos a ser utilizados, e implica la transformación de datos cualitativos a cuantitativos.

La segunda parte abordará el diseño e implementación de los algoritmos de clustering seleccionados previamente. Para construir tales algoritmos, se desarrollarán las instrucciones propias de los mismos, las cuales se encuentran basadas en diferentes teorías, y estructuras matemáticas y estadísticas, que permitirán determinar: la cantidad -que por defecto pueden llegar a formarse-, de grupos de estudiantes con dificultades conceptuales similares, así como la descripción detallada de elementos en cada grupo. El

anterior proceso va a sentar las bases para que: en primer lugar, sea más fácilmente identificable un número de grupos ideal por cada conjunto de datos; en segundo lugar, identificar el nivel de precisión con el que cada algoritmo asigna o reubica los datos atípicos.

La tercera parte de esta investigación busca establecer qué algoritmo generó los clusters más completos y precisos, para esto se calcularán de índices de validación interna, se aplicará la prueba estadística chi-cuadrado y se analizará la ubicación de datos atípicos.

Finalmente, para este estudio comparativo, se ha decidido presentar los resultados a través de una interfaz de usuario en la cual se pueda observar, de forma detallada: en primer lugar, las salidas y/o resultados generados por cada algoritmo; en segundo lugar, información relevante del proceso, como cantidad de grupos formados, cantidad de registros por grupo, técnica utilizada y valores de índices de validación interna; por último, las salidas podrán ser visualizadas por medio de representaciones gráficas (dendrogramas, tablas y demás), que sean amigables y faciliten la interpretación de los datos.

3.3. Conclusiones

Contar con una metodología de trabajo clara permite conocer los pasos a seguir, los cuales están orientados a alcanzar cada uno de los objetivos específicos planteados en la investigación. Se parte de la selección de los algoritmos de clustering que harán parte del estudio, luego se implementan cada uno de ellos, se calculan índices y criterios de validación, y, para finalizar se crea una interfaz de usuario para visualizar de forma más clara y detallada la operación de los algoritmos y los resultados de ejecutarlos sobre el conjunto de datos-base del estudio.

4. Implementación

4.1. Introducción

En este apartado se describirá el aspecto diseño o selección del algoritmo de clustering importante en todo proceso de generación y análisis de clusters. Para este estudio comparativo se decidió agrupar el mismo conjunto de observaciones en tres escenarios, cada uno con un algoritmo de clustering; para poder aplicar los algoritmos de clustering seleccionados para el estudio (Jerárquico, Rank Order y k-means de MacQueen), se aplicaron tratamientos a los datos para que estos coincidieran con las estructuras de datos de entrada de cada algoritmo.

4.2. Conocimientos previos – datos de entrada

Para este proyecto de investigación se cuenta con un conjunto de datos base obtenido de un estudio previo realizado por Robles et al. (2012); en tal estudio se aplica un test que evalúa siete conceptos diferentes (*ver* Figura 4.1) a un conjunto de personas, y a partir de los resultados fue generado archivo con los conceptos débiles para cada uno de los sujetos evaluados.

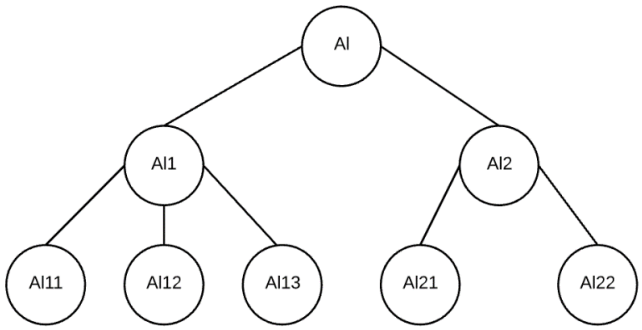


Figura 4.1. Árbol con la estructura temática evaluada en el test. En Descubrimiento de problemas de aprendizaje a través de test: fiabilidad y metodología de diagnóstico basado en clustering por Robles, L., y Rodríguez-Artacho M., 2012.

Como se evidencia en la Figura 4.2, el conjunto de datos tiene como primera columna los códigos de los evaluados y luego, frente a cada registro de estudiante los conceptos sobre los cuales tuviera debilidad, uno en cada columna, ordenados teniendo en cuenta el recorrido en preorden que se hizo del árbol que contiene la estructura temática evaluada en el test.

```
S012  A11   A12   A111  A112  A113  A121  A122
S021  A11   A12   A111  A112  A113  A121  A122
S024  A11   A12   A111  A112  A113  A121  A122
S025  A11   A12   A111  A112  A113  A121  A122
S027  A11   A12   A111  A112  A113  A121  A122
S033  A11   A12   A111  A112  A113  A121  A122
                    ...
S345  A11   A12   A111  A112  A113  A121  A122
S348  A11   A12   A111  A112  A113  A121  A122
S351  A12   A111  A121  A122
S354  A11   A12   A111  A113  A121  A122
S357  A12   A111  A113  A121  A122
S358  A11   A12   A111  A112  A113  A121  A122
S360  A11   A12   A111  A112  A113  A121  A122
```

Figura 4.2. Estructura archivo de conceptos débiles por evaluado.

4.3. Número de grupos

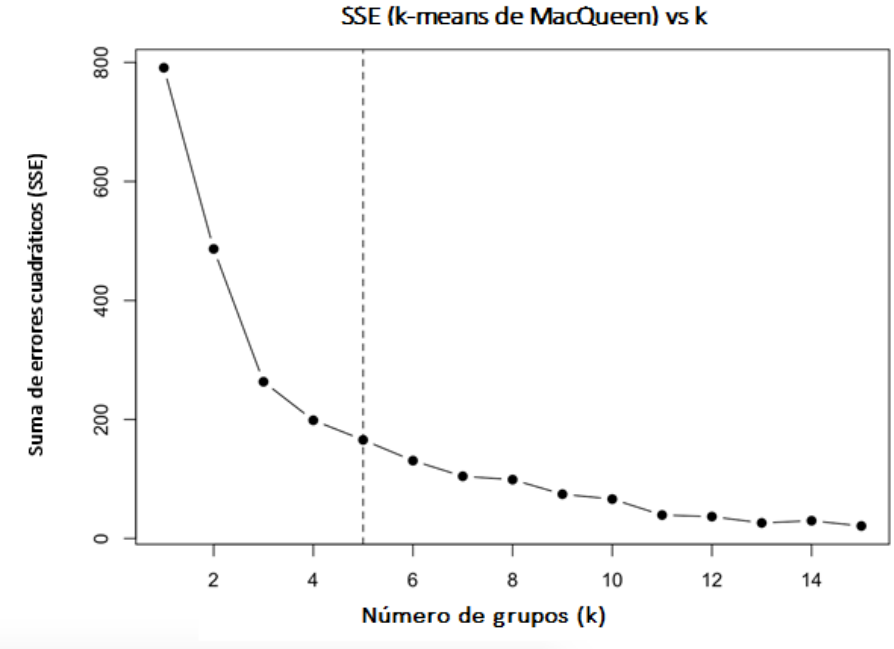
Para determinar el número de grupos óptimo para este estudio, se implementó el método del codo; el primer paso es aplicar el método k-means de MacQueen para distintos valores de k, utilizando R.

Se emplean ciertas sentencias, iniciando por `read.table`, utilizada para la lectura del archivo con índices como data frame (formato de tabla). Antes de agrupar, se debe preparar los datos, en este caso eliminar o estimar los datos que faltan y estandarizar las variables para facilitar su procesamiento, serán usadas para esto las funciones `na.omit` y `scale` respectivamente.

Se debe fijar el rango de valores de k para los cuales se calculará el SSE, en este caso 15, utilizando la sentencia `k.max <- 15`.

A la matriz de datos continuos (con la que se cuenta para aplicar el algoritmo k-means de MacQueen), descrita en el siguiente apartado, se le aplica en R el algoritmo `kmeans` para diferentes valores de k utilizando una función `sapply`. Para cada valor de k, calcular el SSE, este valor hace parte de los valores retornados por el método `kmeans`, específicamente el vector `withinss`.

Por último, se utiliza el comando `plot` para generar la gráfica (*ver* Gráfica 4.1) que muestra la variación del SSE respecto al número de grupos k.



Gráfica 4.1 Suma de errores cuadráticos vs número de grupos.

Teniendo en cuenta la Gráfica 4.1, se puede afirmar que a partir de 5 grupos la diferencia observada en la disimilitud dentro del grupo no es sustancial. En consecuencia, podemos decir con cierta confianza razonable que el número óptimo de clusters a utilizar es 5.

4.4. Implementación de los algoritmos de clustering

A continuación se procede a aplicar los tres algoritmos al conjunto de datos objeto de este estudio, estos son observaciones de 114 estudiantes respecto a sus conceptos débiles en una prueba que les fue aplicada (*ver* Figura 4.2).

Algoritmo Jerárquico – Distancia Euclidiana – Método de Ward

- Transformación de los datos de entrada

  El tratamiento de los datos consiste en crear una matriz de índices que contenga la sumatoria de los pesos de los conceptos débiles por estudiante. Así, se tiene la siguiente matriz de datos de entrada:

	IND
S012	95
S021	95
S024	95
S025	95
S027	95
S033	95
...	
S345	95
S348	95
S351	54
S354	91
S357	67
S358	95
S360	95

Figura 4.3. Matriz de datos de entrada Algoritmo Jerárquico – Distancia Euclidiana – Método de Ward.

- Detalles de la implementación

  A la matriz de índices de cada estudiante se le aplica en R la medida de similitud de distancia euclidiana y el método jerárquico aglomerativo de Ward para la determinación de los grupos.

  Se emplean ciertas sentencias, iniciando por `read.table`, utilizada para la lectura del archivo con índices como data frame (formato de tabla). Antes de agrupar, se debe preparar los datos, en este caso eliminar o estimar los datos que faltan y estandarizar las variables para facilitar su procesamiento, serán usadas para esto las funciones `na.omit` y `scale` respectivamente.

  Para determinar los grupos se aplica el método de distancia euclidiana usando la función `dist`, a esta se le debe especificar qué medida se va a calcular con el parámetro `method = "euclidean"`; y el método de agrupamiento `hclust` que viene con el paquete `stats`.

  Luego, para visualizar los grupos se genera un dendrograma usando la función `plot`, y se decide el número de grupos con `rect.hclust` para ver los cortes sobre el gráfico.

  De igual manera puede ser utilizada la función `cutree` para dividir la estructura resultante de `hclust` en determinado número de grupos, de tal manera que se pueda obtener un vector que contenga los elementos con etiquetas que definen quiénes hacen parte de qué grupo.

- Datos de salida - clusters generados

  Luego de ejecutar el algoritmo, se obtuvo el siguiente agrupamiento:

```
S360 S358 S348 S345 S337 S312 S310 S292 S288 S285 S282 S276 S273 S267 S260 S252 S245 S232 S228 S222 S218 S216
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S205 S198 S196 S195 S192 S176 S174 S171 S165 S162 S161 S159 S153 S140 S135 S123 S120 S111 S099 S095 S093 S087
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S078 S072 S057 S054 S053 S051 S044 S042 S039 S034 S033 S027 S025 S024 S012 S021 S354 S240 S233 S048 S060 S336
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2    2    2    5
S331 S303 S300 S294 S291 S261 S255 S249 S237 S219 S213 S210 S209 S183 S177 S150 S148 S144 S126 S121 S117 S108
   5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5    5
S084 S090 S186 S105 S138 S318 S357 S309 S264 S207 S189 S083 S141 S114 S351 S327 S325 S297 S283 S270 S243 S231
   5    5    5    5    5    4    4    4    4    4    4    4    4    3    3    3    3    3    3    3    3    3
S180 S110 S075 S081
   3    3    3    3
```

Figura 4.4. Grupos generados por el Algoritmo Jerárquico – Distancia Euclidiana – Método de Ward.

En resumen, se tienen los grupos presentes en la Tabla 4.1.

Grupo 1				Grupo 2	Grupo 3	Grupo 4	Grupo 5	
S360	S228	S135	S033	S354	S114	S318	S336	S144
S358	S222	S123	S027	S240	S351	S357	S331	S126
S348	S218	S120	S025	S233	S327	S309	S303	S121
S345	S216	S111	S024	S048	S325	S264	S300	S117
S337	S205	S099	S012	S060	S297	S207	S294	S108
S312	S198	s095	S021		S283	S189	S291	S084
S310	S196	S093			S270	S083	S261	S090
S292	S195	S087			S243	S141	S255	S186
S288	S192	S078			S231		S249	S105
S285	S176	S072			S180		S237	S138
S282	S174	S057			S110		S219	
S276	S171	S054			S075		S213	
S273	S165	S053			S081		S210	
S267	S162	S051					S209	
S260	S161	S044					S183	
S252	S159	S042					S177	
S245	S153	S039					S150	
S232	S140	S034					S148	

Tabla 4.1. Resumen grupos generados por el Algoritmo Jerárquico – Distancia Euclidiana – Método de Ward.

Algoritmo Rank Order – Distancia Euclidiana – Método de Ward

- Transformación de los datos de entrada

Se busca construir una matriz binaria donde cada fila es un estudiante, codificado con la letra S seguida del número del alumno, hasta el número total de alumnos, *m*. Cada columna, por su parte, representa un concepto evaluado, codificado con la letra A además del número de concepto, hasta el número total de conceptos, *n*. La parte interior de la matriz contiene información binaria de si un alumno reprobó (tiene débil) o no un concepto en particular, tomando valores 1 o 0 respectivamente. Esto es, Mij = 1 significa que para el estudiante i el concepto j es débil, y Mij = 0, lo contrario.

Para la construcción de una matriz binaria se traslada el conjunto de datos base a un archivo de excel para su manipulación; en primer lugar, se le adicionó la lista de códigos de los estudiantes, en la columna (J), y como primera fila los conceptos evaluados como sigue:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1											AI1	AI11	AI12	AI13	AI2	AI21	AI22
2	S012	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S012							
3	S021	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S021							
4	S024	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S024							
5	S025	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S025							
6	S027	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S027							
7	S033	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S033							
8	S034	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S034							
9	S039	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S039							
10	S042	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S042							
11	S044	AI1	AI2	AI11	AI12	AI13	AI21	AI22		S044							
12	S048	AI1	AI2	AI11	AI13	AI21	AI22			S048							

Figura 4.5. Tratamiento de datos algoritmo Rank Order, archivo de Excel con conceptos débiles por estudiante.

El paso siguiente consiste en rellenar la parte interior de la matriz utilizando una fórmula, el resultado ésta será 1 si un concepto hace parte de los conceptos débiles de un estudiante y 0 en el caso contrario; la fórmula tiene la siguiente estructura:

$$ArrayFormula(SI(O(celda1{:}celdan = concepto); 1; 0)) \quad (3)$$

Donde:

$celda1$ y $celdan$ delimitan el arreglo de celdas correspondiente a los conceptos débiles de un

estudiante.

concepto es el concepto evaluado que se buscará dentro de los conceptos débiles del estudiante.

Entrando en detalle, para aplicar la fórmula a una celda específica, por ejemplo, la K2, correspondiente a un estudiante (S012) y un concepto (Al1), la explicación sigue a continuación:

Paso 1: se usa la función lógica O[2] para comparar el intervalo de celdas correspondiente a los conceptos débiles del estudiante S012 con el concepto Al1 (cabecera de la columna), así

(O($B2:$H2=J$1)

Paso 2: por medio de una condición SI se establece que, si el resultado de la función O es TRUE, entonces, la celda adquiere valor de 1, en caso contrario 0.

SI(O($B2:$H2=J$1);1;0)

Paso 3: finalmente presionar *Ctrl + Shift + Enter* para aplicar la fórmula sobre la celda, debido a que es de tipo *single-cell array formula*[3], por esto, automáticamente aparece la etiqueta ArrayFormula.

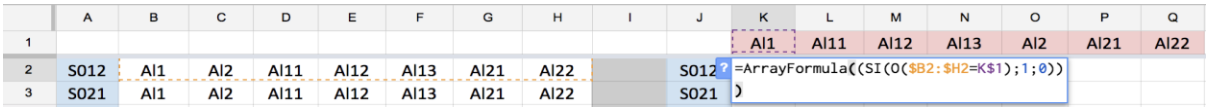


Figura 4.6. Tratamiento de datos algoritmo Rank Order, fórmula empleada para crear matriz binaria.

El resultado de aplicar la fórmula a cada concepto de cada estudiante se evidencia en la Figura 4.7.

[2]Función que retorna *true* si alguno de los argumentos especificados es verdadero desde el punto de vista lógico y *false* si todos los argumentos son falsos.

[3]Fórmulas que operan sobre intervalos de celdas y el resultado se almacena en una única celda.

	Al1	Al11	Al12	Al13	Al2	Al21	Al22
S012	1	1	1	1	1	1	1
S021	1	1	1	1	1	1	1
S024	1	1	1	1	1	1	1
S025	1	1	1	1	1	1	1
S027	1	1	1	1	1	1	1
S033	1	1	1	1	1	1	1
S034	1	1	1	1	1	1	1
S039	1	1	1	1	1	1	1
S042	1	1	1	1	1	1	1
S044	1	1	1	1	1	1	1
S048	1	1	0	1	1	1	1
S051	1	1	1	1	1	1	1

Figura 4.7. Tratamiento de datos algoritmo Rank Order, estructura final archivo de Excel con conceptos débiles por estudiante.

Paso 4: finalmente, se copia la matriz creada en un nuevo archivo y guardar con formato txt, para ser usado como datos de entrada del algoritmo ROC. La estructura definida para el archivo es mostrada en la Figura 4.8, en esta se evidencia, como notación para cada línea, la separación de términos por medio de tabulaciones (caracter "\t"). Este caracter será utilizado, por el método de lectura de archivo presente en el algoritmo, como clave para separar los términos que el archivo contiene.

Se requiere que al final del archivo de entrada, no se encuentren líneas en blanco, para evitar errores de lectura. Finalmente se obtiene la siguiente matriz de entrada:

```
      Al1   Al11  Al12  Al13  Al2   Al21  Al22
S012  1     1     1     1     1     1     1
S021  1     1     1     1     1     1     1
S024  1     1     1     1     1     1     1
S025  1     1     1     1     1     1     1
S027  1     1     1     1     1     1     1
S033  1     1     1     1     1     1     1
                  ...
S345  1     1     1     1     1     1     1
S348  1     1     1     1     1     1     1
S351  0     1     0     0     1     1     1
S354  1     1     0     1     1     1     1
S357  0     1     0     1     1     1     1
S358  1     1     1     1     1     1     1
S360  1     1     1     1     1     1     1
```

Figura 4.8. Matriz de datos de entrada Algoritmo Rank Order.

- Detalles de la implementación

  El algoritmo ROC, fue programado en lenguaje *python*. Se parte de tener el archivo de entrada construido con las especificaciones antes descritas, el proceso que se sigue consiste en aplicarle el algoritmo, para obtener el ordenamiento de estudiantes deseados.

  Inicialmente, se crearon un arreglo de pesos binarios, llamado *binaryArray,* y otro arreglo para los pesos w_i de cada fila, *linesWeightArray*.

Arreglo de pesos binarios

j	6	5	4	3	2	1	0
2^j	64	32	16	8	4	2	1

Este peso w_i sale del producto punto del arreglo de potencias binarias 2^j y la fila del *estudiante S012*. Esto es $w_i = (64, 32, 16, 8, 4, 2, 1).(1,1,1,1,1,1,1) = 127$

	Conceptos							
Estudiantes	AI1	AI11	AI12	AI13	AI2	AI21	AI22	**Wi**
S012	1	1	1	1	1	1	1	127
S021	1	1	1	1	1	1	1	127
S024	1	1	1	1	1	1	1	127
				.				.
				.				.
				.				.
S357	0	1	0	1	1	1	1	47
S358	1	1	1	1	1	1	1	127
S360	1	1	1	1	1	1	1	127

Matriz de incidencia (M)

Figura 4.9. Algoritmo Rank Order - primer paso de la implementación.

A partir de los w_i se creó otro vector *orderedLinesWeight*, que contiene valores de 1 al 114, uno para cada estudiante, de tal manera que se conozca, en orden descendente, como deben ser organizadas las filas. Después se procede a copiar cada fila en una nueva matriz, *orderedMatrix,* de tamaño igual al de la matriz de incidencia, en la posición que le corresponde según el vector de *orderedLinesWeight.* Se tiene entonces:

	Conceptos							
Estudiantes	AI1	AI11	AI12	AI13	AI2	AI21	AI22	**Wi**
S012	1	1	1	1	1	1	1	127
S021	1	1	1	1	1	1	1	127
S024	1	1	1	1	1	1	1	127
				.				.
				.				.
				.				.
S110	0	1	0	0	1	1	1	39
S081	0	1	0	0	1	1	1	39
S075	0	1	0	0	1	1	1	39

Matriz de incidencia (M)

Figura 4.10. Algoritmo Rank Order - cálculo de Wi.

Luego se repite el proceso, con la diferencia de que se busca reordenar las columnas. Es así, que se crea un arreglo w_j con los pesos correspondientes a cada columna, *columnsWeightArray*. El proceso se ilustra en la Figura 4.11.

	Conceptos						
Estudiantes	AI1	AI11	AI12	AI13	AI2	AI21	AI22
S360	1	1	1	1	1	1	1
S358	1	1	1	1	1	1	1
S348	1	1	1	1	1	1	1
				.			
				.			
				.			
S110	0	1	0	0	1	1	1
S081	0	1	0	0	1	1	1
S075	0	1	0	0	1	1	1
Wj	2076918743413931051412198531478323 2L	2076918743413931051412198531688038 3L	2076918743413931051412198525134438 4L	2076918743413931051187018556977561 6L	2076918743413931051412198316880383L	2076918743413931049835938662108364 7L	2076918743413931049835938662108364 7L

Arreglo de pesos binarios

i	2^i
113	1,04E+41
112	5,19E+40
111	2,60E+40
.	.
.	.
.	.
2	4
1	2
0	1

Matriz de incidencia (M)

Este peso w_j sale del producto punto del arreglo de potencias binarias 2^i y la columna del *concepto AI1*

Esto es $w_j = (1{,}04E+41,\ 5{,}19E+40,\ 2{,}60E+40,\ \ldots,4,\ 2,\ 1).(1,1,1,\ \ldots,\ 0,0,0) =$

$= 2076918743413931051412198531478323 2L$

Figura 4.11. Algoritmo Rank Order - segundo paso de la implementación.

Usando los pesos w_j se creó el vector *orderedColumnsWeight*, con valores de 1 al 7, uno para cada concepto, para así obtener, en orden descendente, como deben ser organizadas las columnas de la matriz. Finalmente, se copian las columnas en orden en una nueva matriz, *finalMatrix,* de tamaño igual al de la matriz de incidencia, en la posición que le corresponde. Es entonces cuando se obtiene la matriz ordenada final (*ver* Figura 4.12).

	Conceptos						
Estudiantes	AI22	AI2	AI11	AI1	AI12	AI13	AI21
S360	1	1	1	1	1	1	1
S358	1	1	1	1	1	1	1
S348	1	1	1	1	1	1	1
				.			
				.			
				.			
S110	1	1	1	0	0	0	1
S081	1	1	1	0	0	0	1
S075	1	1	1	0	0	0	1

Matriz de incidencia (M)

Figura 4.12. Algoritmo Rank Order - estructura matriz de resultados de la implementación.

- Valor agregado

Como fue mencionado previamente, el algoritmo ROC se limita a ordenar los datos, por tanto, sus resultados no posibilitan identificar claramente qué grupos se forman y quiénes hacen parte de cada uno, es por esto que se decidió establecer un procedimiento para generar índices -uno para cada estudiante-.

El índice se genera de la siguiente forma:

Paso 1: se utilizó la matriz resultante del ROC cuya primera columna son los códigos de los estudiantes, su primera fila coincide con los conceptos evaluados y se tiene frente a cada registro de estudiante los conceptos sobre los cuales tuviera debilidad representados con 1 o con 0 en caso contrario.

Paso 2: se asignó a cada columna (concepto) una potencia de 2, desde 0 hasta el valor que corresponde al concepto de la primera columna, con incrementos unitarios, de derecha a izquierda.

Paso 3: para cada fila (estudiante), se calculó una sumatoria correspondiente a la multiplicación del valor de la matriz para cada concepto por la potencia asignada, en el paso anterior, a la columna en que el concepto se encuentra. Finalmente, para guardar los índices se generó el archivo *indices.txt*, que contiene los estudiantes y sus respectivos índices.

Paso 4: los índices calculados servirán para definir qué grupos se generan, cuáles y cuántos estudiantes los conforman. Sin embargo, debido a que existen casos atípicos, cuyos índices no coinciden con alguno de los valores calculados, se dispuso aplicar un algoritmo en R que determine los grupos.
Luego de calcular los índices se obtiene:

```
IND
S360   127
S358   127
S348   127
S345   127
S337   127
S312   127
     ...
S231   113
S180   113
S110   113
S081   113
S075   113
```

Figura 4.13. Matriz de datos a los que se le aplicará distancia Euclidiana y método de Ward para particionar.

Finalmente, teniendo los índices, se procede a aplicar distancia Euclidiana y método de Ward. Para esto se consideran las mismas sentencias empleadas en la implementación del Algoritmo Jerárquico, expuesto en el apartado previo.

- Datos de salida – clusters generados

  Luego de ejecutar el algoritmo, se obtuvo el siguiente agrupamiento:

```
S360 S358 S348 S345 S337 S312 S310 S292 S288 S285 S282 S276 S273 S267 S260 S252 S245 S232 S228 S222 S218 S216
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S205 S198 S196 S195 S192 S176 S174 S171 S165 S162 S161 S159 S153 S140 S135 S123 S120 S111 S099 S095 S093 S087
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S078 S072 S057 S054 S053 S051 S044 S042 S039 S034 S033 S027 S025 S024 S021 S012 S186 S138 S105 S336 S331 S303
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2
S300 S294 S291 S261 S255 S249 S237 S219 S213 S210 S209 S183 S177 S150 S148 S144 S126 S121 S117 S108 S090 S084
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
S354 S240 S233 S060 S048 S318 S114 S357 S309 S264 S207 S189 S141 S083 S351 S327 S325 S297 S283 S270 S243 S231
   3    3    3    3    3    4    4    4    4    4    4    4    4    4    5    5    5    5    5    5    5    5
S180 S110 S081 S075
   5    5    5    5
```

Figura 4.14. Grupos generados por el Algoritmo Rank Order – Distancia Euclidiana – Método de Ward.

En resumen, se tienen los grupos presentes en la Tabla 4.2.

Grupo 1				Grupo 2		Grupo 3	Grupo 4	Grupo 5
S360	S228	S135	S033	S336	S144	S354	S318	S351
S358	S222	S123	S027	S331	S126	S240	S114	S327
S348	S218	S120	S025	S303	S121	S233	S357	S325
S345	S216	S111	S024	S300	S117	S048	S309	S297
S337	S205	S099	S012	S294	S108	S060	S264	S283
S312	S198	s095	S021	S291	S084		S207	S270
S310	S196	S093	S186	S261	S090		S189	S243
S292	S195	S087	S105	S255			S141	S231
S288	S192	S078	S138	S249			S083	S180
S285	S176	S072		S237				S110
S282	S174	S057		S219				S075
S276	S171	S054		S213				S081
S273	S165	S053		S210				
S267	S162	S051		S209				
S260	S161	S044		S183				
S252	S159	S042		S177				
S245	S153	S039		S150				
S232	S140	S034		S148				

Tabla 4.2. Resumen grupos generados por el Algoritmo Rank Order – Distancia Euclidiana – Método de Ward.

Algoritmos k-means de MacQueen

- Transformaciones a los datos de entrada

  Para aplicar el algoritmo de k-medias se requiere de una matriz de datos continuos debido a que es un tipo de agrupamiento basado en el cálculo de centroides y medias; es por esto que se generó una matriz similar a la creada para el algoritmo ROC, con la diferencia de que en su interior contiene la proporción de respuestas correctas por cada concepto de un estudiante tomando valores de 0 a 1. Esto es, Mij = 1 el estudiante domina completamente el concepto y 0 no lo domina.

  Para este caso, la estructura de la matriz proviene de cierta prueba desarrollada en la tesis de Robles et al. (2012). La matriz de datos es:

	A11	A111	A112	A113	A12	A121	A122
S012	0.375	0.0	0.0	0.0	0.538	0.0	0.0
S021	0.167	0.0	0.0	0.0	0.308	0.0	0.0
S024	0.167	0.0	0.0	0.0	0.308	0.0	0.0
S025	0.375	0.0	0.0	0.0	0.538	0.0	0.0
S027	0.167	0.0	0.0	0.0	0.308	0.0	0.0
S033	0.167	0.0	0.0	0.0	0.308	0.0	0.0
			. . .				
S345	0.375	0.364	0.0	0.0	0.538	0.333	0.385
S348	0.208	0.364	0.0	0.0	0.231	0.333	0.385
S351	0.792	0.364	0.571	0.75	0.846	0.333	0.385
S354	0.417	0.364	0.571	0.75	0.231	0.333	0.385
S357	0.625	0.364	0.571	0.75	0.538	0.333	0.385
S358	0.208	0.364	0.0	0.0	0.231	0.333	0.385
S360	0.417	0.364	0.0	0.0	0.538	0.333	0.385

Figura 4.15. Grupos generados por el Algoritmo k-means de MacQueen.

- Detalles de la implementación

  A la matriz de datos continuos antes descrita se le aplica en R el método de agrupamiento `kmeans` para agrupar a los estudiantes.

  Se emplean ciertas sentencias, iniciando por `read.table`, utilizada para la lectura del archivo con índices como data frame (formato de tabla). Antes de agrupar, se debe preparar los datos, en este caso

eliminar o estimar los datos que faltan y estandarizar las variables para facilitar su procesamiento, serán usadas para esto las funciones `na.omit` y `scale` respectivamente.

Para agrupar se utiliza el método `kmeans` de R; se debe especificar el algoritmo que será empleado con el parámetro `algorithm`, para este caso `"MacQueen"`. Este método retorna un objeto con diferentes componentes, el `cluster` es el que contiene el vector de membresía, es decir un vector en el que se detalla a qué grupo pertenece qué estudiante.

- Datos de salida – clusters generados

Luego de ejecutar el algoritmo, se obtuvo el siguiente agrupamiento:

```
S360 S358 S348 S345 S337 S312 S310 S292 S288 S285 S282 S276 S273 S267 S260 S252 S245 S232 S228 S222 S218 S216
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S205 S198 S196 S195 S192 S176 S174 S171 S165 S162 S161 S159 S153 S140 S135 S123 S120 S111 S099 S095 S093 S087
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
S078 S072 S057 S054 S053 S051 S044 S042 S039 S034 S033 S027 S025 S024 S021 S012 S186 S138 S105 S336 S331 S303
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    2    2    2
S300 S294 S291 S261 S255 S249 S237 S219 S213 S210 S209 S183 S177 S150 S148 S144 S126 S121 S117 S108 S090 S084
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
S354 S240 S233 S060 S048 S318 S114 S357 S309 S264 S207 S189 S141 S083 S351 S327 S325 S297 S283 S270 S243 S231
   3    3    3    3    3    4    4    4    4    4    4    4    4    4    5    5    5    5    5    5    5    5
S180 S110 S081 S075
   5    5    5    5
```

Figura 4.16. Grupos generados por el Algoritmo k-means de MacQueen.

En resumen, se tienen los grupos presentes en la Tabla 4.3.

Grupo 1				Grupo 2		Grupo 3	Grupo 4	Grupo 5
S360	S228	S135	S033	S336	S144	S354	S318	S351
S358	S222	S123	S027	S331	S126	S240	S114	S327
S348	S218	S120	S025	S303	S121	S233	S357	S325
S345	S216	S111	S024	S300	S117	S048	S309	S297
S337	S205	S099	S012	S294	S108	S060	S264	S283
S312	S198	s095	S021	S291	S084		S207	S270
S310	S196	S093	S186	S261	S090		S189	S243
S292	S195	S087	S105	S255			S141	S231
S288	S192	S078	S138	S249			S083	S180
S285	S176	S072		S237				S110
S282	S174	S057		S219				S075

S276	S171	S054		S213				S081
S273	S165	S053		S210				
S267	S162	S051		S209				
S260	S161	S044		S183				
S252	S159	S042		S177				
S245	S153	S039		S150				
S232	S140	S034		S148				

Tabla 4.3. Resumen grupos generados por el Algoritmo k-means de MacQueen

4.5. Interfaz de usuario

En aras de facilitar la visualización tanto de los resultados como del procedimiento que sigue cada algoritmo para agrupar, se decidió crear una interfaz de usuario. La interfaz de usuario en este caso es una aplicación web llamada ClusteringApp, desarrollada con las tecnologías Django para el backend, y JavaScript, HTML5 y CSS3 para el frontend. La aplicación cuenta con un panel de opciones en las cuales se encuentran los tres algoritmos, luego de seleccionar cualquiera de estos, se puede cargar un archivo con los datos que se desea agrupar, y se podrá visualizar un paso a paso para llegar a la solución.

4.6. Descripción del entorno de desarrollo

a. **Python**. Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, lo que significa que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, que usa tipado dinámico, y es multiplataforma. Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia Pública General de GNU, a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

   En esta investigación, se trabajará con la versión de Python 2.7.10, en el entorno pyCharm 5.0.3.

b. **R Project.** R es un lenguaje de programación especialmente orientado al análisis estadístico y a la representación gráfica de los resultados obtenidos. Es un proyecto GNU. Por lo tanto, los usuarios son

libres de modificarlo y extenderlo. Se trata de un lenguaje basado en comandos, en lugar de pinchar y arrastrar iconos o menús con el ratón se escriben comandos o instrucciones que son ejecutados. Una sucesión de instrucciones o comandos de R, que implementa un flujo de trabajo para realizar una tarea se denomina script o guion en R. Existe una amplia variedad de entornos de desarrollo para R que facilitan escribir scripts de R tales como: R commander, RKWard y RStudio.

En esta investigación, se trabajará con la versión 0.99.903 de RStudio.

4.7. Conclusiones

Conocer a detalle el proceso de implementación de cada uno de los algoritmos de clustering objeto de este estudio, permite tener una visión clara respecto a los diferentes tratamientos hechos al conjunto de datos-base utilizado, además del funcionamiento de cada algoritmo. De igual manera, es de gran utilidad conocer la aplicación ClusteringApp desarrollada que permite visualizar de forma más precisa los procedimientos y resultados de los algoritmos.

5. Validación y Pruebas

5.1. Introducción

Con los solos resultados que se generaron en el capítulo anterior con cada algoritmo no es suficiente, se debe validar que los algoritmos cumplen correctamente con su función de agrupar tomando como referencia una base de datos ampliamente conocida que tenga agrupados o clasificados los registros. Se decidió trabajar con iris, esta es una base de datos que es usada para hacer clasificación pues cuenta con una clase de salida, sin embargo, para efectos de la investigación se decidió trabajar con ese conjunto omitiendo la clase. Se ejecutaron los algoritmos implementados en este estudio sobre los datos de iris para verificar si los datos son agrupados de la manera como son presentados originalmente.

5.2. Prueba estadística chi-cuadrado

Para la prueba estadística se decidió trabajar con la base de datos iris, como referencia teórica. iris, es quizás la base de datos más conocida que se encuentra en la literatura de Reconocimiento de Patrones [R. A. Fisher 1936]. El conjunto de datos contiene 3 clases, de 50 instancias cada una, donde cada clase se refiere a un tipo de planta de Iris.

Información acerca de los atributos:

1. Sepal length in cm
2. Sepal width in cm
3. Petal length in cm
4. Petal width in cm
5. Class:

- Iris Setosa
- Iris Versicolor
- Iris Virginica

<u>Objetivo de la prueba</u>

La prueba estadística tiene como objetivo evidenciar que existen diferencias en el rendimiento de los tres algoritmos realizados para esta investigación, y que por tanto uno de ellos es el mejor. Rendimiento

medido en porcentaje: número de datos que registra en cada grupo generado versus el número de datos registrados en el referente teórico.

Para llevar a cabo este objetivo, se describe a continuación el procedimiento:

- Identificar la base de datos estándar sobre la cual realizar las pruebas. Este conjunto de datos debe tener claramente identificados los grupos.
- Analizar y extraer los grupos.
- Calcular la proporción de registros que cada grupo tiene, respecto al total.
- Ejecutar cada uno de los tres algoritmos sobre el conjunto de datos estándar.
- Identificar la proporción de registros que queda en cada grupo, respecto al total, para las salidas generadas por cada algoritmo.

De acuerdo con la información obtenida, -salida generada por cada algoritmo-, y teniendo en cuenta que este tipo de experimentos forman parte de pruebas que no precisan plantear inferencias sobre parámetros de la población (media y dispersión), llamadas también Pruebas no Paramétricas, el estadístico de prueba que pudiera ser utilizado, es la prueba Chi-cuadrado.

		Algoritmo 1	Algoritmo 2	Algoritmo 3
		Jerarquico - hclust Distancia Euclediana Método Ward	Rank Order Distancia Euclediana Método Ward	kmeans Distancia Euclediana (MacQueen)
Número de Datos Observados	Iris Setosa	n1 = 48	n1 = 50	n1 = 50
	Iris Versicolor	n2 = 74	n2 = 50	n2 = 62
	Iris Virgínica	n3 = 27	n3 = 50	n3 = 38
Número de Datos Esperado	Iris Setosa	50	50	50
	Iris Versicolor	50	50	50
	Iris Virgínica	50	50	50

Tabla 5.1. Número de Datos Observados y Esperados

El estadístico de prueba chi-cuadrado para este experimento tiene 3 categorías (Iris Setosa, Iris Versicolor, Iris Virginica), lo que significa que tendrá (k – 1) = 2 grados de libertad, debido a que la única restricción lineal impuesta para el número de datos es que:

$$n1 + n2 + n3 = 150$$

<u>Prueba de hipótesis:</u>

Se quiere probar que la proporción de datos que cada algoritmo agrupa o incluye en cada grupo generado (3 grupos), es igual al referente teórico (0.3334).

Probar que: p(Iris Setosa) = 0.3334
p(Iris Versicolor) = 0.3334
p(Iris Virginica) = 0.3334

Hipótesis para pruebas chi-cuadrado:

Ho =	Algoritmo 1	Algoritmo 2	Algoritmo 3
	p(Iris Setosa) = 0.3334	p(Iris Setosa) = 0.3334	p(Iris Setosa) = 0.3334
	p(Iris Versicolor) = 0.3334	p(Iris Versicolor) = 0.3334	p(Iris Versicolor) = 0.3334
	p(Iris Virgínica) = 0.3334	p(Iris Virgínica) = 0.3334	p(Iris Virgínica) = 0.3334

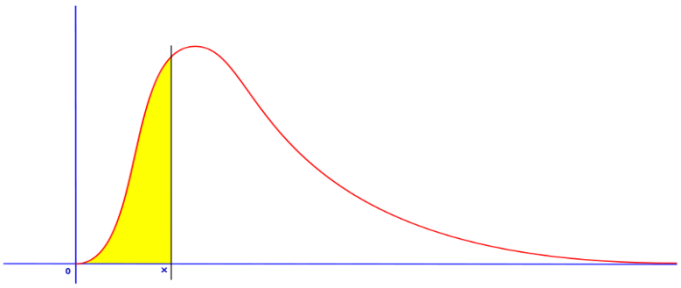
Ha = $p_i \neq p_j$, para algún i, j

Por tanto, con un α = 0.05, se debe calcular:

$$\mathcal{X}^2_{k-1,\propto} = \mathcal{X}^2_{2,0.05}$$

Utilizando la Tabla de Puntos Porcentuales de las Distribuciones $\mathcal{X}^2$ se obtiene que:

$$\mathcal{X}^2_{2,0.05} = 5.991$$



Gráfica 5.1. Distribución de probabilidad de la prueba.

Siendo x = 5.9991, valor de la probabilidad acumulada de cero a 5.991.

Se rechaza Ho si y solo si:

$$\mathcal{X}^2_{experimental} > 5.991$$

Con un nivel de 95% de confianza.

$$\mathcal{X}^2_{experimental} = \mathcal{X}^2_1 + \mathcal{X}^2_2 + \mathcal{X}^2_3$$

Para calcular $\mathcal{X}^2$ experimental se utilizan los valores observados y esperados Tabla 5.1, los cuales se resumen en la siguiente tabla:

	Alg1 $e_o(e_i)$	Alg2 $e_o(e_i)$	Alg3 $e_o(e_i)$
Iris Setosa	48(50)	50(50)	50(50)
Iris Versicolor	74(50)	50(50)	62(50)
Iris Virginica	27(50)	50(50)	38(50)

Tabla 5.2. Relación datos observados y esperados.

Y se calculan los valores de chi-cuadrado para cada algoritmo, denotado con $\mathcal{X}^2_1, \mathcal{X}^2_2$ y $\mathcal{X}^2_3$, como sigue:

$$\mathcal{X}^2{}_1 = \sum\left(\frac{(48-50)^2}{50} + \frac{(74-50)^2}{50} + \frac{(27-50)^2}{50}\right) = 22{,}18$$

$$\mathcal{X}^2{}_2 = \sum\left(\frac{(50-50)^2}{50} + \frac{(50-50)^2}{50} + \frac{(50-50)^2}{50}\right) = 0$$

$$\mathcal{X}^2{}_3 = \sum\left(\frac{(50-50)^2}{50} + \frac{(62-50)^2}{50} + \frac{(38-50)^2}{50}\right) 5{,}76$$

$$\mathcal{X}^2_{experimental} = \mathcal{X}^2_1 + \mathcal{X}^2_2 + \mathcal{X}^2_3 = 22{,}18 + 0 + 5{,}76 = 27{,}94$$

$$\mathcal{X}^2_{experimental} = 27{,}94 > 5.991$$

Por tanto, se rechaza Ho. Hay evidencia que indica que al menos uno de los algoritmos está generando grupos con proporción de datos diferente al referente teórico.

5.3. Análisis de casos atípicos

A partir de los resultados obtenidos de cada algoritmo se procede a analizar cómo estos agrupan los casos atípicos, para así determinar cuál es más funcional para este estudio en cuanto a ese aspecto.

El primer paso es identificar los casos atípicos para el conjunto de datos-base de estudiantes. Los casos atípicos fueron seleccionados teniendo en cuenta la distribución que se tiene de los estudiantes de acuerdo a sus conceptos débiles, esta distribución se ilustra en la Figura 5.1.

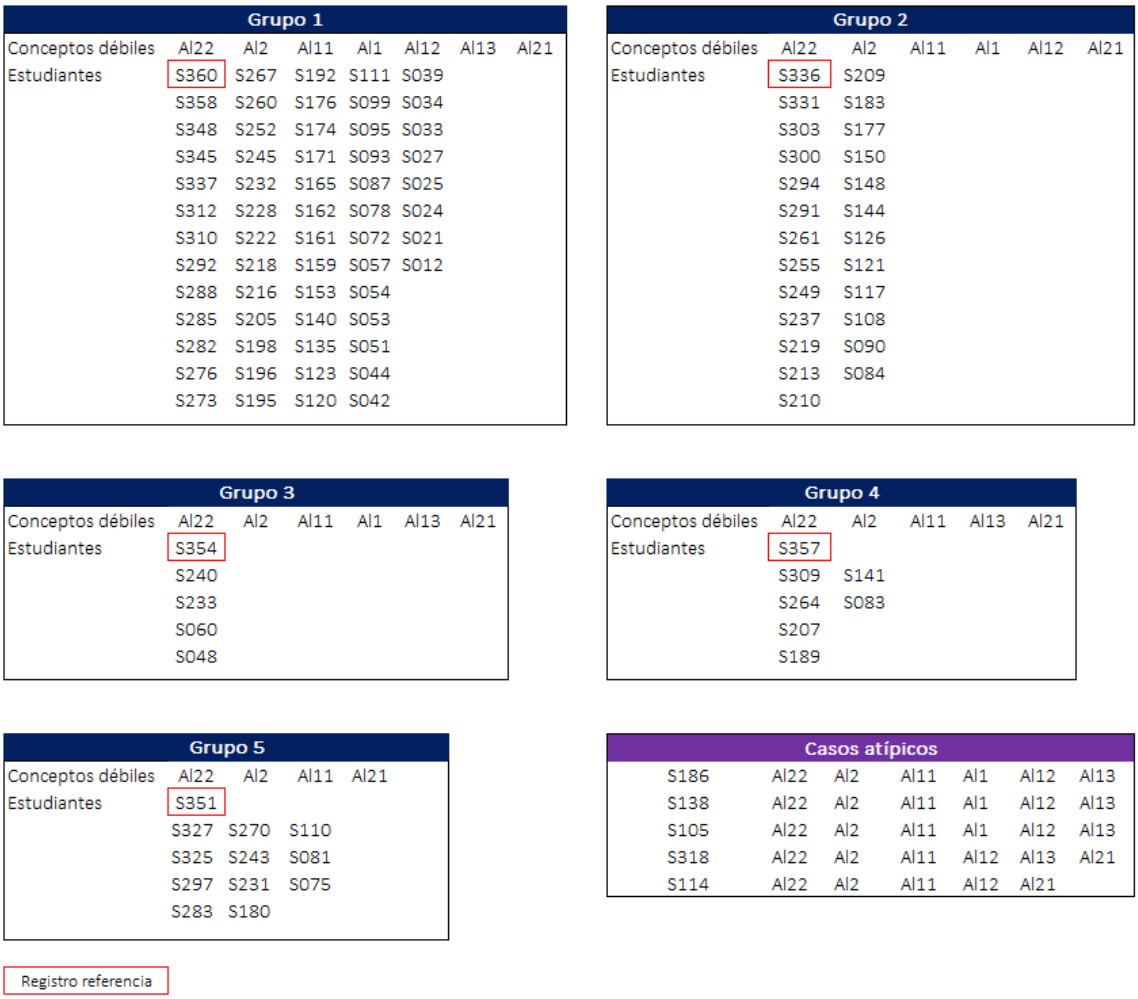


Figura 5.1. Distribución de estudiantes en grupos de acuerdo a sus conceptos débiles.

El registro referencia (encerrado en rojo) de cada grupo es aquel que facilitará identificar en qué grupo fue asignado cada caso atípico.

El siguiente paso es correr cada algoritmo sobre el conjunto de datos; estos resultados se evidencian en la Figura 5.6.

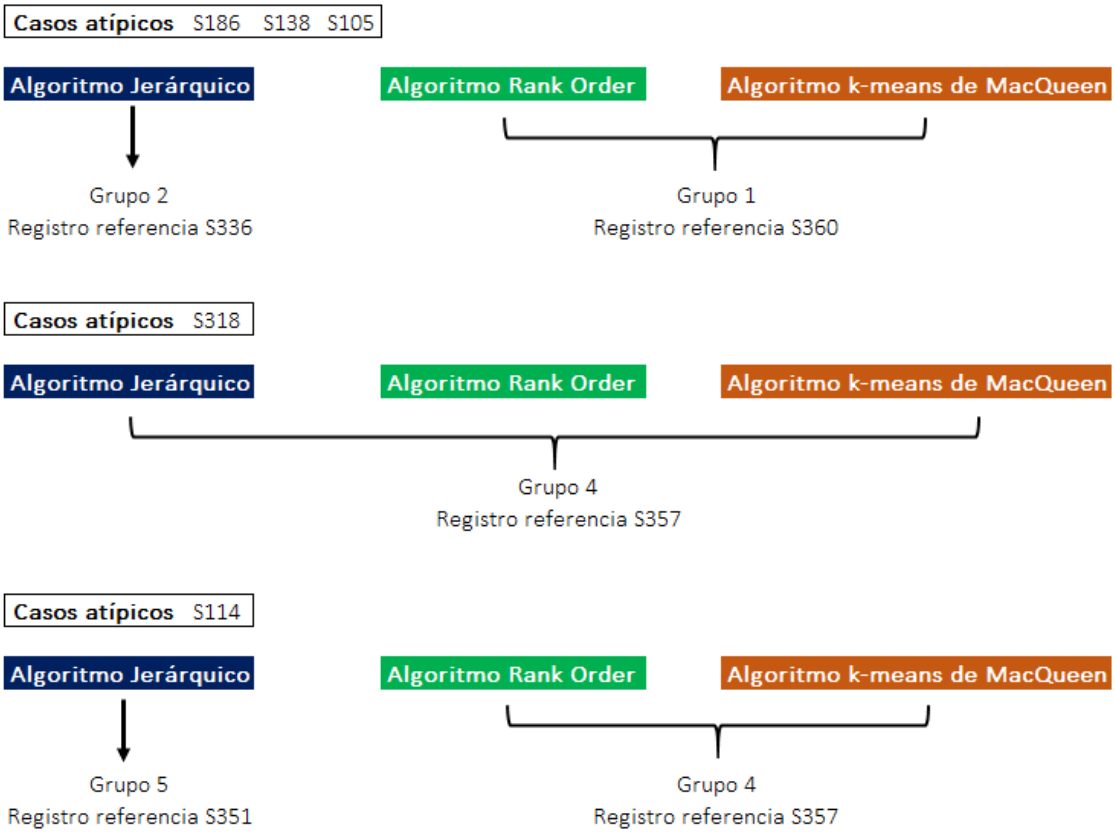


Figura 5.2. Agrupamiento de casos atípicos con algoritmos Jerárquico, Rank Order y k-means de MacQueen.

Por último se debe analizar los resultados para cada caso atípico, estos son:

Estudiantes S186, S138 y S105

- **Algoritmo Jerárquico:** determinó que los estudiantes deben hacer parte del grupo 2; en este caso, fueron agrupados incorrectamente el cual no cuenta con todos los conceptos débiles que el estudiante debe reforzar
- **Algoritmos Rank Order y k-means de MacQueen:** en este caso, los algoritmos agruparon de forma acertada pues en este grupo los estudiantes podrán reforzar todos sus conceptos débiles.

Estudiante S318

- **Algoritmos Jerárquico, Rank Order y k-means de MacQueen:** en este caso, los tres algoritmos fallan porque incluyen al estudiante en un grupo al que le hace falta el concepto Al12.

Estudiante S114

- **Algoritmos Jerárquico, Rank Order y k-means de MacQueen:** los tres algoritmos fallan porque, a pesar, de que incluyen al estudiante en los grupos 5 y 4, a ambos les hace falta el concepto Al12.

En conclusión se tiene que los casos atípicos son 5 estudiantes, de estos se logró ubicar correctamente a 3 utilizando los algoritmos Rank Order y k-means de MacQueen; por su parte, el algoritmo Jerárquico fallo en agrupar a todos los atípicos, al incluirlo en grupos en los que no se encuentran todos sus conceptos débiles y por tanto, no serviría el proceso de retroalimentación,

5.4. Medición de la calidad de los resultados

Para medir la calidad de los resultados se acude a índices de validación interna, estos miden la bondad de la estructura interna del agrupamiento realizado por cada algoritmo; en este caso se consideran los índices de Dunn y Silhouette. Debido a que estos índices miden separación inter-cluster y cohesión intra-cluster respectivamente, se busca maximizar sus valores; así, se considera el algoritmo más acertado aquel cuyos índices sean los mayores.

Los valores calculados en cada caso se muestran en la siguiente tabla:

	Jerárquico Distancia Euclidiana Método Ward	Rank Order Distancia Euclidiana Método Ward	K-means (MacQueen)
Índice de Dunn	1	3	0.2926338
Índice de Silhouette	0.9673971	0.974685	0.5512583

Tabla 5.3. Valores de índices Dunn y Silhouette para cada algoritmo de clustering.

Considerando los valores mostrados en la Tabla 5.3, se tiene que el algoritmo que obtiene los mejores índices de validación, es decir, los más altos, es el algoritmo Rank Order.

6. Conclusiones y Trabajos Futuros

6.1. Conclusiones

A continuación, se presentan las conclusiones más importantes que se extraen del presente trabajo de grado:

- Los algoritmos de tipo no jerárquico hacen suposiciones fuertes acerca de la naturaleza de los datos y requieren de parámetros de entrada como el número de clases y la ubicación inicial de centroides, mientras que los algoritmos jerárquicos solo usan información referente al conjunto de datos.
- La prueba estadística chi-cuadrado ayudó a determinar que los tres algoritmos son válidos y que agrupan de manera diferente.
- El mejor agrupamiento para un caso atípico es ser incluidos en un grupo que contenga todos sus conceptos débiles e incluso más, nunca menos porque en el proceso de retroalimentación no reforzaría todos los conceptos que tiene débiles.
- A partir del agrupamiento de casos atípicos se puede concluir que el algoritmo Jerárquico realiza el peor agrupamiento. Por su parte los algoritmos Rank Order y k-means (MacQueen) tienen un comportamiento acertado.
- En cuanto a índices de validación Rank Order tiene el mejor desempeño. Seguido del agrupamiento Jerárquico y por último k-means.
- A pesar de tener buenos índices de validación, el algoritmo jerárquico no agrupa correctamente datos atípicos.
- El algoritmo más óptimo es Rank Order pues tiene buen desempeño en cuanto a los índices de validación y además agrupa casos atípicos satisfactoriamente.
- Se cumplió el objetivo de identificar el mejor algoritmo; minimizando el riesgo de que no se clasificaran correctamente los datos y haciendo que los datos atípicos se ubicarán correctamente.
- Se logró modificar el algoritmo Rank Order, permitiendo que este tuviera un método de particionamiento, calculando un índice y haciendo uso de la distancia Euclidiana y método de Ward.
- Los tratamientos de los datos afectan considerablemente el resultado de aplicarles los algoritmos, pues estos son sensibles a la cantidad de variables, cantidad de datos, presencia de datos atípicos, tipo de datos, estructura prueba busca validar que los algoritmos presentados en capítulos anteriores agrupan asertivamente.

6.2. Trabajos Futuros

En este apartado se presentan algunas líneas de investigación que pueden ser objeto de interés, y se derivan del presente estudio.

- En relación con el algoritmo Rank Order Clustering, como medida para contrarrestar su limitación en cuanto a tamaño de la matriz de incidencia y en aras de optimizar resultados se propone la implementación del algoritmo Direct Clustering Technique, que se basa en la idea de mover bloques y moverlos conservando las relaciones entre componentes y máquinas.
- En cuanto al método de agrupamiento k-medias sería interesante comparar los resultados del algoritmo de MacQueen, aplicado en este estudio, con los otros algoritmos Lloyd, Forgy y Hartigan que implementan k-medias.
- Se tiene previsto otro trabajo de grado que prueba otras técnicas de agrupamiento, entre ellas razonamiento basado en casos.
- Desarrollar un diseño de experimentos, en que se tengan en cuenta repeticiones para algoritmos como k-means que generan en cada corrida resultados ligeramente diferentes.
- Crear un mecanismo que permita identificar de manera automática los datos atípicos para posteriormente hacer una reubicación de esos registros más asertiva, más pertinente.

Referencias Bibliográficas

Ansari, Z., Azeem, M. F., Ahmed, W., & Babu, A. V. (2015). Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *arXiv preprint arXiv:1507.03340*.

Campo, D. N., Stegmayer, G., & Milone, D. H. (2014). Análisis de estabilidad en clusters solapados. Inteligencia Artificial, 17(53), 79-89.

Cascante, M. & Coronado, J. (2007). *Tecnología de grupos*. Nota Técnica NT-2202-000-VP. Universidad de los Andes. Pág 1 -19.

Chang, W. C., Chen, S. L., Li, M. F., & Chiu, J. Y. (2009, December). Integrating IRT to Clustering Student's Ability with K-Means. In *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on* (pp. 1045-1048). IEEE.

Dalton, L., Ballarin, V., & Brun, M. (2009). Clustering algorithms: on learning, validation, performance, and applications to genomics. *Current genomics*, *10*(6), 430-445.

De la Fuente, S. (2011). *Identificación de clusters en un dendrograma* [Figura]. Recuperado de http://www.fuenterrebollo.com/Economicas/ECONOMETRIA/SEGMENTACION/CONGLOMERADOS/conglomerados.pdf

Duque, D., y Flórez, J. (2012). *Reconocimiento de patrones espaciales sísmicos en el sur Occidente Colombiano mediante aprendizaje no supervisado.* (tesis de pregrado). Pontificia Universidad Javeriana Cali. Cali, Colombia.

Fisher, R.A. "The use of multiple measurements in taxonomic problems" Annual Eugenics, 7, Part II, 179-188 (1936); also in "Contributions to Mathematical Statistics" (John Wiley, NY, 1950).

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. Journal of intelligent information systems, 17(2-3), 107-145.

Morissette, L., & Chartier, S. (2013). The k-means clustering technique: General considerations and implementation in Mathematica. *Tutorials in Quantitative Methods for Psychology*, *9*(1), 15-24.

Robles, L., y Rodríguez-Artacho M. (2012). *Descubrimiento de problemas de aprendizaje a través de test: fiabilidad y metodología de diagnóstico basado en clustering.* (tesis de maestría). Universidad Nacional de Educación a Distancia - UNED. Madrid, España.

Villagra, A. *Metaheurísticas aplicadas a Clustering.* (tesis de maestría). Universidad Nacional de San Luis. San Luis, Argentina.

Wook, M., Yahaya, Y. H., Wahab, N., Isa, M. R. M., Awang, N. F., & Seong, H. Y. (2009, December). Predicting NDUM student's academic performance using data mining techniques. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on* (Vol. 2, pp. 357-361). IEEE.

Anexos

Anexo I. Listado de funciones en R

`read.table(file, header=TRUE, ...)`

Donde:

`file` representa el nombre del fichero del cual se leerán los datos, este debe incluir la ruta de ubicación del archivo. Si no se especifica una ruta absoluta, el nombre del archivo es relativo al directorio de trabajo actual, o puede ser una URL.

`header` es un valor lógico usado para indicar si se toma la primera línea del archivo como las cabeceras de las columnas.

`na.omit(object, ...)`

Donde `object` es el objeto que será tratado con la función.

`scale(x, ...)`

Donde `x` es la matriz que será estandarizada.

`dist(x, method = "euclidean", ...)`

Donde:

`x` es una matriz numérica, data frame u objeto "dist".

`method` representa la medida de distancia que será utilizada. Puede tomar valores de `"euclidean", "maximum", "manhattan", "canberra", "binary"` o `"minkowski"`.

`hclust(d, method="", ...)`

Donde:

`d` es la matriz de distancias como la producida por la función dist.

`method` representa el método de agrupamiento que será utilizado. Puede tomar valores de `"ward", "single", "complete", "average", "mcquitty",`

"median" o "centroid". En versiones recientes de R, se actualizó el método ward convirtiéndose en ward.D.

plot(x, hang=-1, cex.axis=0.6, cex=1.6, sub="Distancia", ...)

Donde:

x es el objeto resultante de aplicar hclust.

hang permite visualizar las etiquetas de los elementos, que se encuentran en el eje x, todas al mismo nivel.

cex.axis tamaño de las etiquetas tick del eje.

cex tamaño del texto (es un multiplicador, para textos menores, usar números menor de 1). 1,5 es 50% más grande, 0,5 es 50% más pequeño, etc.

sub es el valor del subtítulo de la gráfica.

rect.hclust(tree, k = NULL, border="red", ...)

Donde:

tree es el árbol resultante de hclust.

k es el número de grupos en que será dividido tree.

border es el vector con los colores de la frontera para los rectángulos.

abline(h=1.2 ,lty=2 ,col="blue", ...)

Donde:

h es el valor del eje y por el cual será trazada la línea.

lty es el tipo de línea, puede tomar valores de 1 a 6.

col es el color de la línea.

cutree(tree, k = 5, ...)

Donde:

tree es el árbol resultante de hclust.

`k` es el número k de grupos en que será dividido tree.

`kmeans(x, algorithm = "MacQueen", ...)`

Donde:

`x` es la matriz de datos.

`algorithm` representa el algoritmo de agrupamiento que será utilizado. Puede tomar valores de `"Hartigan-Wong", "Lloyd", "Forgy", o "MacQueen"`.

`intCriteria(x, vector, crit)`

Donde:

`x` es la matriz de datos.

`vector` es el vector de particiones (o vector de membresía).

`crit` es el vector que contiene los nombres de los índices que se desea calcular.