Universidad de San Andrés

Departamento de Matemática y Ciencias

Maestría en Ciencia de Datos

Clustering en alta dimensión

Identificación de variables relevantes en datos mixtos

Autora: Griselda Martiarena

Directora: Marcela Svarc

2023

Maestría en Ciencia de Datos

Departamento de Matemática y Ciencias

Clustering en alta dimensión

Identificación de variables relevantes en datos mixtos

Griselda Martiarena **2023**

Directora: Marcela Svarc

Universidad de SanAndrés

Universidad de
SanAndrés

Resumen

La aceleración del progreso tecnológico ha generado nuevos desafíos y oportunidades para la supervisión bancaria. El desarrollo y la aplicación de herramientas que facilitan y potencian su tarea es una de ellas. En este sentido, este trabajo aborda la clusterización de las entidades financieras argentinas, a partir de una cantidad considerable de información pública estructurada, y la identificación de las variables relevantes en este proceso. Para ello, se apoya en el empleo de un algoritmo genético y de la metodología de *blinding* para la selección de variables. En particular, amplía la aplicación de este último enfoque, no sólo a conjuntos de datos con variables numéricas, categóricas y binarias, sino también a métodos de clustering por particiones, siendo este el principal aporte metodológico. Adicionalmente, procura ofrecer una herramienta que permita entender más a fondo el ecosistema bancario en un contexto de volúmenes crecientes de datos. Los resultados finales obtenidos son satisfactorios, confirmando la solidez y utilidad de la estrategia de ocultamiento.

PALABRAS CLAVES: K-medoides; Distancia de Gower; Coeficiente de Correlación Cofenética; Algoritmo Genético; Identificación de variables; Clustering; Alta dimensión; Datos mixtos.

Agradecimientos

A mi familia, por su apoyo y comprensión. A Marcela, por su valiosa ayuda y respaldo permanente en el desarrollo de este proyecto. Y a mis colegas del BCRA, por sus sugerencias constructivas.

Índice general

Índice de figuras

Índice de cuadros

CAPÍTULO 1

Introducción

La aceleración del progreso tecnológico evidenciado en los últimos años ha generado no sólo nuevos desafíos, sino también oportunidades sin precedentes para el mundo en general y la banca, en particular. En este sentido, la creciente implementación de innovación tecnológica, proceso acelerado e intensificado por la irrupción de la pandemia (OECD [21], Beerman et al. [3]), es una de ellas. En efecto, por ejemplo, el uso de tecnología puede permitir a las autoridades supervisoras de entidades financieras hacer más eficiente y efectiva su supervisión, al mejorar la identificación y monitoreo de fuentes de riesgo, como así también la precisión de la información y los tiempos de disponibilidad de la misma y de comunicación con el sector financiero (WB [30]).

En general, el empleo de la innovación tecnológica por parte de las autoridades financieras, en sentido amplio, para dar ayuda a su trabajo, se define bajo el término *Suptech (Supervisory technology)*, donde innovación tecnológica refiere a la aplicación de *big data* o inteligencia artificial en las herramientas utilizadas (Di Castri et al. [8]). Si bien varios desarrollos en la materia ya existían con anterioridad, las nuevas oportunidades que ofrece *Suptech* han resultado de una combinación de factores manifestados recientemente, tales como la disponibilidad de datos en forma masiva y la disposición de nueva infraestructura tecnológica (FSB [11]). De esta forma, ha surgido la posibilidad de desarrollar un amplio abanico de herramientas para dar soporte a una variedad de actividades, que incluye desde la recolección de datos hasta la supervisión prudencial (Beerman et al. [3]). En concreto, se pueden implementar herramientas para: análisis de sentimiento, análisis de conexiones entre agentes económicos, identificación de riesgos y clasificación en grupos de pares, etc.

En lo que respecta a este último ejemplo, la agrupación de las entidades financieras según criterios particulares siempre ha resultado de utilidad para los organismos reguladores y supervisores de las mismas para tareas tales como diseñar e implementar políticas con un criterio proporcional, desafiar las clasificaciones vigentes, entender con mayor profundidad o desde otros aspectos el ecosistema bancario y/o identificar características comunes relevantes. Este proceso de identificación se aborda en la literatura, en general, desde la perspectiva de la selección de variables en lo vinculado al desarrollo de procedimientos que buscan dar solución al problema. Bajo distintas metodologías de clusterización y tipos de datos, este tópico recibió basta atención en décadas pasadas (por ej. Steinly y Brusco [27] comparan ocho técnicas distintas de selección para clusterización basada y no basada en modelos; y Fop y Murphy [9] brindan una descripción general de las primeras, a partir de los trabajos

iniciales hasta el estado del arte del momento), y continúa aún sucintando mucho interés (Yuan et al. [31], Storlie et al. [28], Ghattas et al. [12], Chormunge y Jena [7]) frente a un contexto de disponibilidad creciente de información.

En líneas generales, Alelyani et al. [1] presenta una clasificación de los algoritmos de selección según la información utilizada, es decir, etiquetada o no (supervisado, no supervisado o semisupervisado), el tipo de resultado (subconjunto o ponderación) y la estrategia. En relación a este último aspectos, describe las siguientes categorías: filtro, envolvente, híbrido y embebido (*filter, wrapper, hybrid, embedded* respectivamente, en inglés). Los correspondientes a la primera categoría son ajenos al proceso de agrupación y se basan en ciertos criterios estadísticos para el análisis de la relevancia de las variables. En tanto, los que forman parte de la segunda estriban en la selección de aquellas variables que tienen el mayor poder discriminatorio bajo cierto clasificador. Estos últimos son más caros en términos computacionales y su desempeño depende del clasificador, pero resultan más precisos respecto de los primeros. Así, los híbridos son una combinación de ambos, al seleccionar preliminarmente varios conjuntos de variables en base a criterios estadísticos, para luego elegir el que clasifica con mayor precisión. Por último, los pertenecientes a la última categoría, embebido, llevan adelante la selección en el periodo de aprendizaje.

Dentro de los modelos del tipo envolvente disponibles, para el presente trabajo se opta por el de Fraiman et al. [10], quienes proponen una estrategia de ocultamiento de subconjuntos de variables en forma iterativa que no implica la omisión de ninguna variable redundante o no informativa a los fines de la clusterización, sino más bien su sustitución por una estimación en el proceso de clusterización. Esto representa una ventaja comparativa en relación a las otras alternativas, dado que no se pierde información, siempre que el agrupamiento original obtenido resulte razonable, o sea, no se evidencie una baja performance de la clusterización por la no remoción de variables que no serían útiles.

No obstante lo anterior, el procedimiento que introducen estos autores desafortunadamente únicamente se puede aplicar, en forma estricta, cuando la metodología de clustering empleada es por particiones, pero no cuando se utiliza un método jerárquico. Es por ese motivo que se presenta la necesidad de platear un nuevo procedimiento de selección de variables para este problema, el cual resulta una extensión natural de la problemática original y el principal aporte metodológico que presenta este trabajo.

De esta manera, en este trabajo se propone explorar en mayor profundidad la selección de variables en un proceso de clusterización que involucra una base de datos de alta dimensión con variables mixtas, iniciado por Caruso et al. [5], y extender dicho análisis a métodos jerárquicos de agrupamiento. El estudio basa su desarrollo en la propuesta metodológica de ocultamiento de variables redundantes o no informativas de Fraiman et al. [10], el empleo de un algoritmo genético, tomando como ejemplo a Alvarez y Svarc [2], como técnica computacionalmente viable de reducción de dimensión, y el uso de información pública que describe diferentes aspectos de las entidades que conforman el sistema financiero argentino. Esto, con el fin último de agrupar dichas entidades por criterios ajenos al riesgo al que están expuestas e identificar las variables que determinan la agrupación alcanzada. Como objetivo secundario, se procura

ofrecer una herramienta que permita entender más a fondo el ecosistema bancario en un contexto de volúmenes crecientes de datos.

El documento está organizado de la siguiente manera. En el Capítulo 2, se describen los datos disponibles, las técnicas de clustering empleadas, la distancia de Gower y los clusters resultantes. En el Capítulo 3, se plantean la metodología implementada y los algoritmos involucrados. Luego, en el Capítulo 4, se presentan los principales resultados de la metodología, y las variables y agrupaciones consecuentes. Finalmente, en el Capítulo 5, se plantean las principales conclusiones y establecen potenciales líneas de nuevos estudios. Cabe mencionar que al final del trabajo se presenta como Anexos información detallada de la base considerada y de los clusters resultantes, como así también el código R trabajado e implementado.

CAPÍTULO 2

Clusterización

2.1. Descripción de los datos a clusterizar

Para el presente estudio se consideró información pública de las entidades financieras que conforman el sistema financiero argentino, la cual se encuentra disponible en la sitio web del Banco Central de la República Argentina (BCRA)[1] y de la Comisión Nacional de Valores (CNV)[2]. Los datos corresponden al mes de junio de 2022 y resultan de las presentaciones periódicas (mensuales y trimestrales) que realizan las entidades en cumplimiento del régimen informativo requerido por el BCRA, en su función de regulador del funcionamiento del sistema financiero y supervisor de la actividad financiera y cambiaria (a través de la Superintendencia de Entidades Financieras y Cambiarias -SEFyC-).

A partir de los datos recopilados de la forma descripta, se dispuso de una base del tipo corte transversal balanceada de 79 registros (total de entidades activas a las fecha de estudio) y 61 variables, tanto del tipo numéricas, como binarias y categóricas. Del total de variables, 21 se obtuvieron en forma directa de la información disponibles, mientras que las restantes 40 se construyeron a partir de la misma. En este sentido, en líneas generales, las acciones llevadas adelantes fueron: agrupación de información, creación de variables dicotómicas y construcción de ratios e índices de concentración. Para este último caso, se empleó como medida de concentración el Índice de Herfindahl-Hirschman (HHI).[3]

Cabe señalar que las variables consideradas describen distintas características de las entidades en cuestión, en términos generales: dimensión, naturaleza, complejidad de sus operaciones y tipo de negocio, pero, salvo excepciones, no dan cuenta en forma directa de su grado de exposición a los distintos riesgos inherentes de la industria bancaria, tales como el riesgo de crédito, mercado y operacional. Como ya se mencionó en la introducción, no es el fin último de este trabajo la agrupación de las entidades financieras por su nivel o tipo de riesgo. Las variables empleadas se presentan a continuación (ver Cuadro 2.1) y, en el Anexo 1, se incluye una descripción más detallada de las mismas.

[1] http://www.bcra.gob.ar/PublicacionesEstadisticas/Entidades_financieras.asp

[2] https://www.argentina.gob.ar/cnv/empresas

[3] $HHI = \sum_{i=1}^{n} s_i^2$, donde s_i es la participación del agente i y n es la cantidad total de agentes que participan ($\sum_{i=1}^{n} s_i = 100$). A medida que el índice se acerca a 1000, indica un mayor nivel de concentración, mientras que a 0, lo contrario.

Cuadro 2.1: Variables para la clusterización

Nro.	Variable	Tipo
1	Concentración accionaria	numérica
2	Concentración accionaria por tipo de accionista	numérica
3	Concentración de préstamos - 10 mayores clientes	numérica
4	Concentración de préstamos - 60 mayores clientes	numérica
5	Concentración de depósitos - 10 mayores clientes	numérica
6	Concentración de depósitos - 60 mayores clientes	numérica
7	Cantidad de empresas asociadas	numérica
8	Cantidad de empresas subsidiarias	numérica
9	Participación de la cartera comercial en la cartera total	numérica
10	Participación de la cartera asimilable en la cartera total	numérica
11	Participación de la cartera consumo en la cartera total	numérica
12	Participación de los préstamos en el total de préstamos al SFNF	numérica
13	Participación de los depósitos en el total de depósitos del SFNF	numérica
14	Opera en ALADI (1: opera, 0: no opera)	binaria
15	Cantidad de cajas de ahorro de ayuda social	numérica
16	Cantidad de cuentas sueldo	numérica
17	Cantidad de cuentas corrientes	numérica
18	Cantidad de cuentas de ahorro	numérica
19	Cantidad de cuentas previsionales	numérica
20	Cantidad de empresas que disponen de cuentas sueldo	numérica
21	Cantidad de plazo fijos	numérica
22	Cantidad de operaciones por otros préstamos	numérica
23	Cantidad de operaciones por préstamos hipotecarios	numérica
24	Cantidad de operaciones por préstamos prendarios	numérica
25	Cantidad de tarjetas de crédito (plásticos)	numérica
26	Cantidad de tarjetas de débito	numérica
27	Cantidad de titulares por tarjetas de crédito	numérica
28	Dotación de personal	numérica
29	Ubicación de la casa matriz/casa central (1: CABA, 0: No CABA)	binaria
30	Concentración geográfica de las sucursales	numérica
31	Concentración geográfica de los cajeros	numérica
32	Concentración geográfica de las terminales	numérica
33	Cantidad de sucursales plenas	numérica
34	Cantidad de sucursales de operaciones especificas	numérica
35	Cantidad de sucursales móviles	numérica
36	Cantidad de dependencias automatizadas	numérica
37	Cantidad de cajeros automáticos	numérica
38	Cantidad de terminales de autoservicio	numérica
39	Cantidad de puestos de promoción	numérica
40	Cantidad de agencias complementarias de servicios financieros	numérica
41	Activos en el total del sistema financiero	numérica
42	Financiación al SP respecto del total de financiaciones	numérica
43	Financiación al SF en el total del sistema financiero	numérica
44	Adelantos respecto de total de préstamos al SPNF	numérica
45	Prestamos documentarios respecto de total de préstamos al SPNF	numérica
46	Préstamos hipotecarios respecto de total de préstamos al SPNF	numérica

continuación ...

... continuación

Nro.	Variable	Tipo
47	Préstamos prendarios respecto de total de préstamos al SPNF	numérica
48	Préstamos personales respecto de total de préstamos al SPNF	numérica
49	Otros préstamos respecto de total de préstamos al SPNF	numérica
50	Nocionales en derivados en el total del sistema financiero	numérica
51	Créditos por arrendamiento financiero (1: Si, 0: No)	binaria
52	Filiales en el exterior (1: Si, 0: No)	binaria
53	Depósitos del SP respecto del total de depósitos	numérica
54	Depósitos judiciales (1: Si, 0: No)	binaria
55	Ingresos por int. respecto del total de ingresos	numérica
56	Gastos de administración en el total del sistema financiero	numérica
57	Tenencia de títulos valores de fideicomisos financieros (1: Si, 0: No)	binaria
58	Cartera fideicomitida (1: Si, 0: No)	binaria
59	Tenencia de títulos privados en el total del sistema financiero	numérica
60	Fondos Comunes de Inversión (1: Si, 0: No)	binaria
61	Origen del capital	categórica

Nota: SPNF refiere a Sector Privado No Financiero.

Dada la cantidad de variables consideradas, es de esperar que varias de las mismas se encuentren correlacionadas, resultando redundantes, o sean variables no informativas, a los efectos de la clusterización. En efecto, considerando el Coeficiente de Correlación de Pearson (PCC)[4] para analizar la relación por pares de variables, las vinculadas al tamaño de las entidades, particularmente, evidencian una alta correlación entre ellas (ver Figura 2.1)[5].

En lo que respecta a las entidades financieras, a junio 2022, el sistema financiero argentino estaba compuesto por 79 entidades financieras[6]: 64 bancos (13 bancos estatales y 51 bancos privados -incluidos 3 bancos digitales y 1 banco cooperativo-) y 15 compañías financieras (incluidas 2 digitales y 8 estrechamente vinculadas al sector automotriz). A su vez, los bancos privados se dividían en aquellos con accionistas extranjeros (9 subsidiarias y 7 sucursales de bancos extranjeros) y aquellos con una participación mayoritaria de accionistas nacionales (35).

En tanto, a los efectos regulatorios, es decir, para la aplicación y cumplimiento de las normas que emite el BCRA, las entidades se dividían en tres grupos, según los establece dicho organismo, a saber: A (18 bancos),

[4] $PCC = \frac{\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}}$, donde x_i (respec. y_i) es la observación i correspondiente a la variable x (respec. y), N es la cantidad total observaciones y $\bar{x}$ (respec. $\bar{y}$) es la media de la variable x (respec. y). A medida que el coeficiente se acerca a 1 o -1, indica una mayor correlación positiva o negativa, respectivamente, mientras que a 0, no habría relación entre ambas variables.

[5] Se empleó el CCP, en detrimento de otras medidas más robustas (Coeficient de Correlación de Spearman o Kendall), al priorizarse la posibilidad de considerar variables binarias y limitar el uso del mismo a la descriptiva de los datos disponibles.

[6] También lo integraban otro tipo de entidades que no captan depósitos, tales como casas de cambio, otros proveedores no financieros de crédito, empresas emisoras de tarjetas de crédito y/o compra, billeteras virtuales y proveedores de servicios de pago, aunque no resultan objeto de estudio en el presente trabajo.

B (15 bancos) y C (46 bancos y compañías financieras). Esto, a partir de una variable indicativa de la dimensión de las mismas: monto de activos como porcentaje del activo total del sistema financiero. Así, las entidades del grupo A son aquellas cuya relación es mayor o igual al 1 %; las del grupo B, cuya relación es mayor o igual al 0,25 % pero menor al 1 %; y las del grupo C, cuyos activos representan menos del 0,25 % del total de activos del sistema financiero[7]. Cabe mencionar que, de las 79 entidades mencionadas, a los fines del presente trabajo se consideraron 74, dado que 3 no se encontraban operativamente activas al estar próximas a cerrar y 2 no estaban aún desarrollando efectivamente su negocio por tratarse de entidades relativamente nuevas (las 5 pertenecientes al grupo C). En Anexo 2, se presenta la lista de entidades financieras bajo estudio a junio 2022, agrupadas según lo establece el ente regulador.

Figura 2.1: Matriz de correlaciones

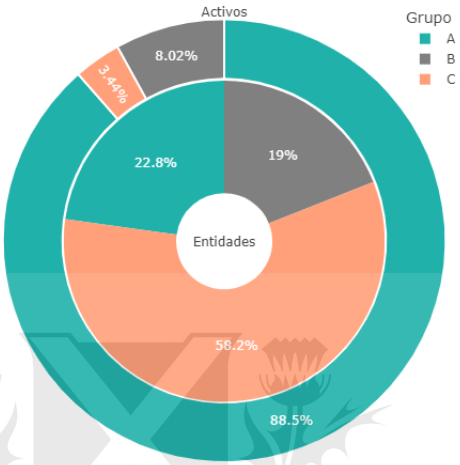
Bajo este criterio de agrupación, las entidades pertenecientes al grupo A, bancos públicos y privados mayormente universales, evidenciaban a junio 2022 un ratio de activos respecto del total de activos del sistema financiero de 4,92 % en promedio, concentrando gran parte del volumen de activos del sistema (88,54 %). Se destacaban 6 entidades, cuya participación superaba el 5 %, y en particular una, con un quinto de los activos totales del sistema. En tanto, la distribución del grupo B, desde esta perspectiva, era más homogénea, con un ratio promedio de 0,53 %. Por su parte, el grupo C, que reúne las restantes entidades del sistema, las cuales tienen negocios disímiles, solo representaba el 0,07 % de los activos en promedio (ver Figura 2.2). Las entidades que conforman el grupo A, además, concentraban casi el 89 % de los préstamos al sector privado no financiero (SPNF) y el 92 % de los depósitos del mismo.

Más allá de estas clasificaciones, es relevante notar que, en términos generales, el sistema financiero argentino es simple, focalizado fundamentalmente en la banca tradicional. Es decir, su actividad principal es captar depósitos y brindar préstamos, su tenencia de instrumentos complejos es baja y su actividad *cross-*

[7]Para mayor referencia, consultar: Comunicación "A" 7108.

border es limitada. Estas características, junto con su relativa concentración y las particularidades de las entidades más pequeñas, hacen difícil la tarea de encontrar una partición para las entidades financieras que exceda la dimensión tamaño.

Figura 2.2: Participación en el total de activos del sistema financiero



2.2. Técnicas de clustering implementadas

Para lograr una mayor discriminación de las entidades en grupo homogéneos, se aplicaron a los datos descriptos dos técnicas de clustering o agrupamiento alternativas pertenecientes a dos familias metodológicas del aprendizaje no supervisado, a saber: clustering por particiones (k-medias) y clustering jerárquico (método de Ward). Si bien existen diferentes métodos para la determinación y análisis de clusters, se emplearon los más tradicionales, no siendo el objetivo final del este trabajo la clusterización en si misma, sino la presentación de un procedimiento que permita encontrar las principales variables que determinan una agrupación dada ante datos mixtos y en alta dimensión.

El texto de estadística de James et al. [18] describe en forma simple y concisa ambas metodologías. En lo que respecta a la técnica por particiones k-medias (*k-means*, en inglés), esta permite particionar los datos en K subconjuntos disjuntos, a partir de la minimización de la varianza intragrupo (*within-cluster variation*, $W(C_k)$) total, donde K es especificada de antemano. Es decir, se reparten las observaciones en K grupos tal que la suma de la varianza intragrupo de los K grupos sea lo más pequeña posible. Existen muchas formas de definir el concepto de varianza intragrupo, aunque la más habitual involucra el cuadrado de la distancia Euclídea entre todas las observaciones de cada cluster o grupo. En definitiva, sean $x_1, \ldots, x_n$ una muestra aleatoria en $\mathbb{R}^V$, se busca determinar los conjuntos $C_1, \ldots, C_k$ que minimicen la siguiente expresión:

$$\sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,j \in C_k} \sum_{v=1}^{V} (x_{i,v} - x_{j,v})^2,$$

donde $|C_k|$ es el número de observaciones en el cluster k y V es el número de variables implicadas en el proceso de clustering.

En la práctica, en un primer paso, el algoritmo asigna en forma aleatoria todas las observaciones a los K clusters. En un segundo paso, para los K grupos se calcula su centroide[8], el cual es un vector conformado por las medias de las V variables de todas las observaciones del grupo k, y se reasigna cada observación al cluster cuyo centroide se encuentre más cercano en términos de distancia Euclídea. Este segundo paso se repite hasta que no se modifique más la conformación de los grupos.

En relación a la técnica de clustering jerárquico, a diferencia de la metodología k-medias, no es necesario la determinación de antemano de la cantidad de grupos, la cual es resultante del análisis del dendograma[9]. Este método puede ser aglomerativo o divisivo, según el algoritmo unifique o separe grupos en sus sucesivos pasos. En particular, en el clustering aglomerativo, técnica utilizada en el presente trabajo, el algoritmo inicia considerando cada observación como un cluster en sí mismo, y va fusionando de a dos grupos basándose en la menor disimilaridad por par de cluters. El algoritmo avanza hasta que todas las observaciones quedan agrupadas en un solo conjunto. Como medida de disimilaridad, generalmente, considera la distancia Euclídea entre dos observaciones, aunque puede ser reemplazada por otra medida de disimilaridad. Asimismo, incorpora la noción de enlace (*linkage*, en inglés) como el criterio considerado para calcular la disimilaridad entre dos grupos de observaciones.

Los tipos mas comunes de enlaces son cuatro, a saber: completo, simple, medio y centroide. Los dos primeros consideran como enlace la máxima y la mínima disimilaridad intercluster, respectivamente, mientras que el tercero tiene en cuenta el promedio de las disimilaridades interclusters, y el cuarto la disimilaridad entre los centroides de los clusters. Adicionalmente, existe el método de Ward, el cual, a diferencia de los métodos anteriores, su criterio de agrupación no es la menor disimilaridad, sino el menor cambio en la suma de las varianzas intraclusters[10].

En lo que respecta a este último, si bien originalmente el método emplea la distancia Euclídea, dado que el criterio considerado para calcular la disimilaridad se basa en la distancia entre las observaciones y su centroide, Murtagh y Legendre [20] señalan que el método de Ward también puede ser utilizado con disimilaridades distintas a la distancia Euclídea. Los autores muestran en términos teóricos como la suma de los cuadros de las mencionadas diferencias puede expresarse en términos de todos los pares de distancias entre las observaciones. Asimismo, los mismos presentan la fórmula sucinta de actualización de las disimilaridades utilizando el algoritmo de Lance-Williams. Cabe destacar que la librería hclust de R permite la clusterización a través de dicha metodología brindando como argumentos una matriz de disimilaridades y el enlace deseado.

[8] Centro de un agrupamiento, que no necesariamente puede coincidir con alguna de las observaciones disponibles, calculado como la media aritmética de las observaciones.

[9] Representación en forma de árbol del proceso de agrupamiento de las observaciones.

[10] En términos de Ward [29], en la minimización de una función objetivo, en este caso, la Suma de los Cuadrados de los Errores (SCE).

La elección del tipo de enlace y de la medida de disimilaridad es relevante, dado que determina el dendograma y, por ende, la conformación de los clusters. Para el presente trabajo, se empleó este último método por criterio experto, por dar lugar a la mejor agrupación relativa. Asimismo, se utilizó, tanto bajo el método jerárquico como el de por particiones, la distancia de Gower en reemplazo de la distancia Euclídea como medida de disimilaridad, al contar con una base de datos no solo con variables numéricas, sino también binarias y categóricas [11].

Si bien existen diversas medidas de disimilaridad, que incluyen distintas formas de medir la distancia más allá de la Euclídea, las mismas se concentran en determinado tipo de variables. En este sentido, diversos autores han presentado un resumen de medidas de similaridad y testeado empíricamente su desempeño con diferentes propósitos. Por ejemplo, Shirkhorshidi et al. [24] estudian 12 medidas de similaridad usadas frecuentemente para datos del tipo numérico continuo, mientras que Boriah et al. [4] estudian 14 medidas para datos del tipo categórico y Choi et al. [6] presentan 76 similaridades y medidas de distancia binarias empleadas en el último siglo. En cambio, la distancia de Gower, de la cual se brindan más detalles en la Sección 2.3, es una medida de aplicación más general, es decir, válida para datos mixtos.

Finalmente, la variedad en el tipo de datos también conllevó a la modificación del algortimo k-medias a k-medoides. Como ya se describió al principio de la presente sección, el primero considera en forma recursiva la distancia a los centroides como criterio de agrupación, concepto que es solo aplicable cuando se cuenta con variables numéricas dado que implica el cálculo de un promedio. Por lo tanto, fue necesario reemplazar dicha noción por una similar, la de medoide. Este concepto refiere al punto en el espacio que se corresponde con una observación, cuya distancia es la mínima con respecto a todo el resto de las observaciones del cluster.

Existen diferentes algoritmos para la implementación de la clusterización vía k-medoides. En particular, en este trabajo se utilizó el algoritmo *Partitioning Around Medoids (PAM)*, el cual es adecuado para conjuntos de datos chicos, que sería el caso presente en este estudio[12]. El texto de estadística de Han et al. [15] resume este procedimiento en forma sencilla. De manera aleatoria, se eligen al azar K observaciones representativas (medoides), conformándose K cluster según la distancia de las restantes observaciones a los mismos. Posteriormente, se evalúa si los reemplazos de los medoides por otras observaciones mejoraría la calidad de la clusterización, considerándose todas las sustituciones posibles. La calidad es medida por una función de costo igual a la disimilaridad promedio (o también suma de disimilaridades) entre las observaciones y la observación representativa de sus clusters. El proceso iterativo de reemplazo continúa hasta que la calidad del agrupamiento resultante no puede mejorarse más.

[11] Conviene notar que este reemplazo también atenúa el problema de pérdida de robustez del procedimiento k-medias por el efecto de potenciales valores extremos en el empleo del cuadrado de la distancia Euclídea en la minimización, al magnificar las grandes distancias [17].

[12] Para abordar grandes bases de datos cabría analizar emplear los algoritmos *Clustering LARge Applications* (CLARA) o *Clustering Large Applications based upon RANdomized Search* (CLARANS), ambos basados en muestreo.

Cabe mencionar que Kauffman y Rousseeuw [19], creadores del algoritmo, plantean un preproceso para la obtención de las K observaciones representativas originales en reemplazo del completo azar. En particular, dividen al algoritmo en dos etapas, la primera, denominada *BUILD*, arroja una clusterización inicial a través de la elección sucesiva de medoides. A partir de la selección de una primera observación representativa, cuya suma de disimilaridades respecto del resto de las observaciones es la menor posible, se eligen observaciones representativas hasta alcanzar los K clusters deseados, considerando el mismo criterio de selección que con la observación original. En definitiva, en forma sucesiva se resuelve el siguiente problema:

$$\max_i \sum_j C_{i,j};$$

$$C_{i,j} = max\left(D_j - d_{i,j}, 0\right);$$

donde D_j es la disimilaridad entre la observación j y el medoide más cercano, y $d_{i,j}$ es la disimilaridad entre la observación j y la propuesta de nuevo medoide i. La segunda etapa, denominada *SWAP*, busca mejorar los clusters resultantes, en línea con lo ya descripto en el párrafo precedente.

2.3. Distancia de Gower

Como se desprende de todo lo mencionado, la utilización de una medida correcta de disimilaridad es indispensable para el buen uso de las técnicas de clustering abordadas, al tratarse de metodologías de agrupamiento que se basan en el cálculo de la distancia entre los puntos en el espacio. En este orden de ideas, como ya se señaló en párrafos precedentes, Gower [14] presenta una alternativa válida para clusterizar a partir de variables de distintos tipos, la cual ha sido bastante utilizada.

Siguiendo al mencionado autor, la similaridad entre dos individuos i y j se define como el puntaje (*score*, en inglés) promedio resultante de todas las posibles combinaciones de las V variables:

$$S_{i,j} = \frac{\sum_{v=1}^{V} s_{i,j,v}}{\sum_{v=1}^{V} \delta_{i,j,v}},$$

donde $s_{i,j,v}$ es el puntaje resultante de la comparación de dos individuos i y j respecto de la variable v; $\delta_{i,j,v}$ es la posibilidad de realizar la comparación entre ambos individuos i y j respecto de la variable v, dado que no siempre es factible llevar adelantes la mencionada comprobación porque puede existir información faltante o no ser posible concluir frente a variables dicotómicas cuando una característica no existe en ambos individuos; y V el número total de variables.

El puntaje $s_{i,j,v}$ toma valor cero cuando ambos individuos i y j son considerados diferentes en esa característica v, y toma valor positivo, menor o igual que uno, cuando tienen cierto grado de similaridad. En tanto, $\delta_{i,j,v}$ es igual a uno cuando la variable v es posible de ser comparada entre los individuos i y j, y viceversa. Algunas convenciones al respecto son: si $\delta_{i,j,v} = 0$, $s_{i,j,v}$ se considera cero dado que no es posible la comparación. Asimismo, si $\sum_{v=1}^{V} \delta_{i,j,v} = 0$, es

decir la comparación no es factible para ninguna de las características, $S_{i,j}$ queda indefinido; en tanto, si todas son realizables, entonces $\sum_{v=1}^{V} \delta_{i,j,v} = V$.

La determinación del puntaje y de la factibilidad de la comparación depende del tipo de variable. Considerando los siguientes tres tipos de variables, se tiene:

- Variable numérica: $s_{i,j,v} = 1 - |x_{i,v} - x_{j,v}|/R_v$, donde $x_{1,v}, x_{2,v}, ..., x_{n,v}$ son los valores que toma la variable v y R_v es el rango de la misma en la muestra. De esta forma, cuando $x_{i,v} = x_{j,v}$ entonces $s_{i,j,v} = 1$, y cuando ambos valores son los extremos opuesto del rango de la muestra entonces $s_{i,j,v} = 0$, por lo que el puntaje entre valores intermedios siempre será una fracción positiva. En este caso, siempre $\delta_{i,j,v} = 1$.

- Variable categórica: $s_{i,j,v} = 1$, si ambos individuos comparten la característica, y $s_{i,j,v} = 0$ caso contrario. Al igual que en el caso de una variable numérica, siempre $\delta_{i,j,v} = 1$.

- Variable binaria: Tanto $s_{i,j,v}$ como $\delta_{i,j,v}$ pueden tomar valores 0 y 1, dependiendo de la presencia de la característica o no en los dos individuos a comparar (+ en caso afirmativo y − en caso negativo). De este modo, existen cuatro combinaciones posibles, a saber:

	Valores de la variable k			
Individuo i	+	+	−	−
Individuo j	+	−	+	−
$s_{i,j,v}$	1	0	0	0
$\delta_{i,j,v}$	1	1	1	0

Así, la similaridad de Gower toma valores entre 0 y 1, resultado 1 cuando dos individuos no difieren en nada, y 0 cuando lo hacen completamente. Ahora bien, la utilización de este concepto como reemplazo de la distancia de Euclídea, como medida de disimilaridad, implica calcular simplemente $1 - S_{i,j}$ para obtener la distancia de Gower.

2.4. Análisis de clusters

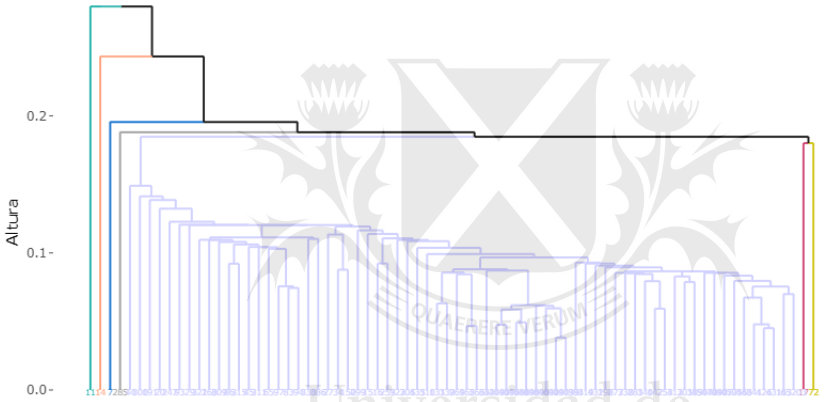
Previo al empleo de cualquiera de las metodología de clusterización mencionadas en la Sección 2.2, se llevó adelante el preprocesamiento necesario de los datos. De este modo, primero, se descompuso la variable categórica existente (origen del capital) en tres variables binarias. Segundo, se normalizaron los datos frente a las diferentes escalas que presentaban las distintas variables. Para ello, se empleó el método Min-Max con una escala entre 1 y 0, dado que el mismo no implica la asunción de distribución normal de los datos, mantiene la relación existente entre los mismos y es apta para variables del tipo binarias al no modificarlas.

Por último, se detectaron y excluyeron valores atípicos (*outliers*, en inglés), mejor denominados valores extremos dentro del marco del tema abordado, con el objetivo de no obtener resultados distorsionados, empleando la técnica de clustering jerárquico de enlace simple. Dadas sus características, que considera como enlace la mínima disimilaridad intercluster, se desprende fácilmente de un

dendograma aquellas observaciones que se encuentran en el espacio alejadas de las nubes de puntos, al unirse al árbol en las últimas instancias del algoritmo aglomerativo (ver Figura 2.3).

De esta manera, se excluyeron 6 entidades financieras, las cuales se destacan particularmente por su dimensión. Todas formaban parte del grupo A y coinciden con aquellas cuya participaciones en el total de activos del sistema financiero superaban el 5 %, representando en forma conjunta casi el 60 % de los activos del sistema. Así, quedaron 68 entidades a clasificar, a partir de 62 variables. Conviene mencionar que este accionar implicó la desaparición de una variable (filiales en el exterior) dado que solo se activaba para 2 entidades, ambas excluidas.

Figura 2.3: Dendograma. Enlace único. Distancia de Gower



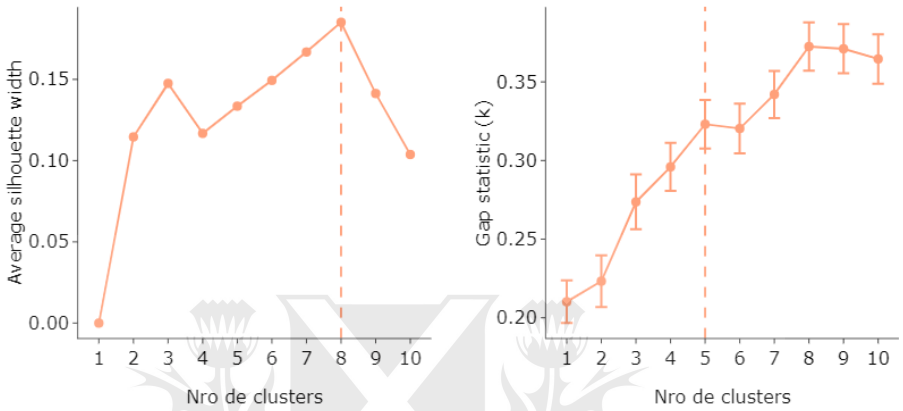
Dada la similitud de las entidades excluidas en base a las características mencionadas para su descripción, se las consideró por criterio experto como un cluster adicional a los estadísticamente conformados, aunque quedaron ajenos a todo procedimiento y análisis realizado en el presente trabajo. A priori, las entidades remanentes podrían agruparse en 7 grupos teniendo en cuenta criterios como los ya citados (dimensión, naturaleza, complejidad de sus operaciones y tipo de negocio): entidades remanentes del grupo A, entidades del grupo B, filiales de bancos extranjeros, bancos de propiedad y control del Estado, compañías financieras, entidades del grupo C divididas en al menos dos grupos para lograr una mayor granularidad en la clasificación.

Le elección de la cantidad óptima de clusters, especialmente bajo la metodología k-medias o k-medoides que implican su selección de antemano, es un tema que no es sencillo, ni directo. No existe un único criterio para su determinación. Dentro de los métodos de elección existentes, para el presente trabajo se emplearon dos: *Average Silhouette Width* y *Gap Statistic*, cuyos criterios de selección en ambos es el valor máximo de los mismos. En lo que respecta al primero, es la media del coeficiente de Silhouette de las observaciones bajo una partición en K grupos, donde dicho coeficiente es el ratio entre una medida de cohesión y una de separación de cada punto; toma valores entre -1 y 1. Por su parte, el segundo es la diferencia entre la varianza intraclusters de

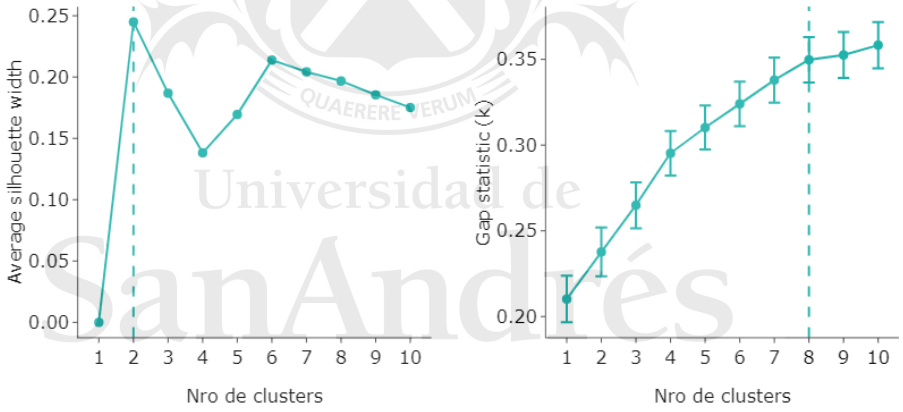
una partición de tamaño K de los datos de estudio y la resultante de datos generados en forma aleatoria bajo una distribución uniforme. De los resultados de ambos indicadores, la cantidad óptima de clusters se ubicaría en torno a 6 o 7 grupos (ver Figuras 2.4a y 2.4b).

Figura 2.4: Cantidad óptima de clusters

(a) K-medoides



(b) Método de Ward

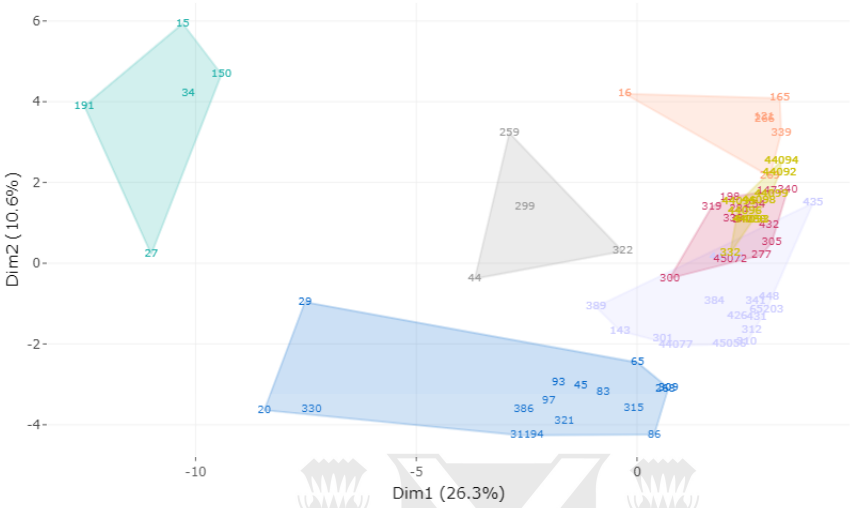


De esta forma, las 68 entidades financieras remanentes se agruparon en 7 clusters, conformándose cada uno con una cantidad razonable de entidades, (entre 4 y 17, 10 en promedio). Los agrupamientos bajo el método k-medoides (ver Figura 2.5) y el método de Ward (ver Figura 2.6) mostraron, en general, cantidades similares de entidades financieras en cada cluster, aunque diferencias en lo que hace a su composición en 17 entidades[13]. No obstante, las dos particiones resultaron en términos generales razonables, destacándose en ambas la clara agrupación de: los bancos universales de primera línea (no incluye los 6 bancos excluidos por resultar valores extremos, los cuales conformarían un octavo cluster *ad-hoc*), las sucursales de entidades del exterior, los bancos públicos y las compañías financieras dedicadas al negocio automotriz[14].

[13]Para contrarrestar el problema de etiquetado, se realizó un alineamiento de los nombres de los clusters, facilitando la comparación de los resultados.

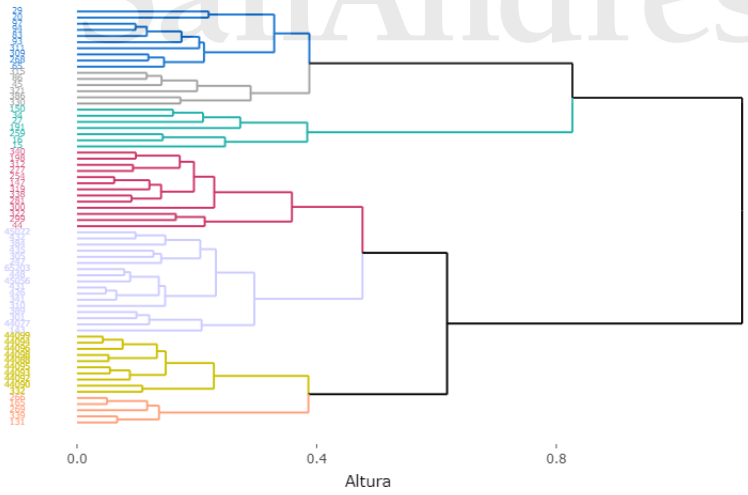
[14]Particularmente, proveen créditos prendarios para la compra de vehículos comercializados por la red de concesionarios correspondiente y otros servicios conexos.

Figura 2.5: Agrupamiento por particiones proyectado en las direcciones de las dos 1ras componentes principales. K-medoides



Si bien por criterio experto, bajo ambas metodologías, algunas entidades podrían ubicarse en otro cluster, el grupo es reducido y podría resultar de la aleatoriedad inherente que todo procedimiento estadístico presenta. Además, cabe destacar la heterogeneidad de las entidades que conforman el grupo C, al poseer muchas de ellas características únicas, lo cual dificulta cualquier intento de agrupación. Más allá de ello, cabría profundizar la investigación del porqué de la pertenencia de estas entidades a grupos determinados. En Anexo 3, se comparten más detalles de las agrupaciones alcanzadas.

Figura 2.6: Agrupamiento por jerarquía. Método de Ward



.

CAPÍTULO 3

Identificación de variables relevantes

3.1. Metodología aplicada

En resumen, a partir del empleo de las metodologías de clusterización mencionadas, se logró agrupar los 68 bancos en 7 grupos homogéneos considerando 62 características que los describen. No obstante, dada esta cantidad, no todas las variables serían determinantes en las agrupaciones logradas, al contarse posiblemente con variables redundantes y/o no informativas. Es aquí donde se presenta un desafío en como conocer las principales variables responsables de una determinada partición.

En este sentido, un primer acercamiento al problema conduciría a pensar en la técnica de Análisis de Componentes Principales (PCA). Ahora bien, esta técnica presenta algunas desventajas que se tornan relevantes en el presente estudio. Por un lado, la transformación de los datos inherente a la técnica, resultado de la implementación de combinaciones lineales de las variables originales, limita la interpretación de los resultados. Siendo el objetivo de este trabajo la identificación de las variables más relevantes en un proceso clusterización, este inconveniente se torna insoslayable. Adicionalmente, las variables que conforman las componentes, resultado de las proyecciones ortogonales de los datos maximizando la varianza de los mismos, no necesariamente coinciden con las responsables de una determinada clusterización que implica otros procesos matemáticos. Todo esto, más allá de que la técnica de PCA asume una relación lineal entre las variables.

El problema de la identificación de las variables relevantes, en general, es abordado en la literatura desde la selección de variables, sorteando los escollos mencionados. A través de los años, numerosas técnicas se han desarrollado tanto para diferentes metodologías de clasificación, como tipos de datos, tal como se describe en el Capítulo 1. Sin embargo, estos procedimientos subordinan los resultados de la clusterización a la bondad de la técnica de selección implementada. En este marco, Sosa Escudero et al. [26] y Caruso et al. [5] modifican el orden del procedimiento, seleccionando las variables en forma posterior al proceso de clusterización, luego de alcanzar una agrupación considerada satisfactoria que tiene en cuenta todas las variables. Para ello, los mencionados autores, basándose en Fraiman et al. [10], adoptan una estrategia

de ocultamiento (*blinding strategy*, en inglés) de aquellas variables redundantes o no informativas, y utilizan los resultados de la agrupación original para validar los resultados de este último procedimiento.

En concreto, explican que frente a la existencia de, por ejemplo, solo dos variables para un procedimiento de clustarización: X e Y, donde X es informativa, pero Y no, ya sea por estar correlacionada con X o por no ser informativa (por lo que aportaría poca información o ruido al proceso), su reemplazo por su mejor predicción basada en X, es decir, su esperanza condicional $E(Y|X)$ (regresión no paramétrica), no debería implicar alteraciones importantes en la agrupación resultante respecto de la original con ambas variables. Por lo tanto, la metodología se resume en encontrar mediante este procedimiento el subconjunto de variables de menor tamaño que reproduzca de la forma más precisa los resultados de una clusterización original. Es importante remarcar que el mencionado método siempre implica el empleo de todas las variables originales, incluso las redundantes bajo su esperanza condicional.

De la descripción anterior, se desprende que este método supone la determinación de un criterio de bondad de ajuste, en este caso, la concordancia entre los clusters obtenidos respecto de los correspondientes al conjunto completo de datos, como así también de un estimador de la esperanza condicional, tal como vecinos más cercanos (en inglés, *nearest neighbors*), técnica que fue implementada. Dentro del campo de aprendizaje supervisado, se trata de un metodología no paramétrica de clasificación basada en asignar una observación al grupo preponderante en los k vecinos más cercanos de la misma.

Un ejemplo concreto del procedimiento de *blinding* se describe a continuación para su mejor comprensión. Suponiéndose que se tienen 5 variables $(X_1, \ldots, X_5)$ y se seleccionan X_2 y X_4 como variables informativas (a priori), se debe aplicar el mencionado proceso a las tres restantes: X_1, X_3 y X_5. Para ello, se calcula la $E(X_i|X_2X_4)$ para $i = 1, 3, 5$ mediante vecinos más cercanos y reemplaza X_i. Es decir, teniendo en consideración solamente X_2 y X_4, se obtienen los k vecinos más cercanos de las n observaciones, para luego computar para las mismas la media aritmética de los valores de X_1, X_3 y X_5 correspondientes a los k vecinos ya determinados, llamándoselos $x_{i,(1)}, \ldots, x_{i,(k)}$. En definitiva: $E(X_i|X_2X_4) = \frac{1}{k}\sum_{j=1,\ldots,k} x_{i,(j)}$ para $i = 1, 3, 5$. Una vez calculada la misma, se reemplaza X_i para $i = 1, 3, 5$, por ella. Posteriormente, teniendo en cuenta tanto X_2 y X_4, como X_1, X_3 y X_5 reemplazadas, se reasignan las n observaciones a los centros más cercanos que resultaron de la clusterización original, obteniéndose la nueva partición.

Para llevar adelante el proceso descripto en forma exhaustiva, deberían testearse todas las combinaciones posibles de variables disponibles para la cluterización, comparándose las diferentes alternativas a través del criterio de bondad de ajuste determinado. No obstante, la resolución del problema enfrenta un limitante computacional frente a observaciones con muchas características, al crecer las combinaciones posibles en forma exponencial a medida que se agrega al problema una nueva variable. De este modo, Fraiman et al. [10] sugieren el empleo de un algoritmo *forward-backward*, en tanto, Alvarez y Svarc [2] plantean la utilización de un algoritmo genético (GA) para dar solución a este

problema[15].

3.2. Algoritmos implementados

Siguiendo a Goldberg [13] y Sivanandam y Deppa [23], un GA es un método de resolución de problemas de optimización que utiliza la selección natural y la genética como modelo de resolución, al combinar la supervivencia del más apto con el intercambio de información estructurada en forma aleatoria, para encontrar soluciones aproximadas a problemas de optimización y búsqueda. Frente a un amplio espacio de soluciones factibles, GA trabaja con una población o subconjunto de soluciones posibles, cada una representada a través de un individuo o cromosoma, a su vez conformado por determinadas características o genes.

A partir de una población inicial dada o determinada al azar, cada nueva generación o conjunto nuevo de individuos (cromosomas) surge como resultado de las características heredadas de los individuos más aptos de la generación pasada, y ocasionalmente de nuevas características. Para ello, de generación en generación, se emplean operadores de reproducción (combinación y mutación) y una función de *fitness* para la selección de los individuos más aptos, que se aplican directamente en los cromosomas. Cada cromosoma tiene un valor de *fitness* calculado, el cual se corresponde a una evaluación de cuán buena es la solución candidata. Luego de reiteradas iteraciones, el algoritmo se detiene cuando la población converge a una solución óptima, la que maximiza la función de *fitness*, o alcanza otro criterio de terminación, por ejemplo, máxima cantidad de iteraciones.

Una de las aplicaciones de los algoritmos genéticos, en particular de los binarios, es la selección de un subconjunto de variables. Los binarios son aquellos en los cuales los genes de los cromosomas se codifican o representan por 1 o 0. O sea, si una característica está presente en un individuo conlleva un 1, caso contrario un 0, resultando cada cromosoma en una cadena binaria. De esta manera, cada cromosoma se asemeja a una potencial solución a nuestro problema, donde 0 indica la ausencia de una variable y 1 la presencia de la misma[16] (Ver Algoritmo 1).

Para la implementación del algoritmo genético binario, se utilizó el paquete en R denominado GA [22] y la función que lleva el mismo nombre, a la cual se le debe proporcionar una función de *fitness* a maximizar. Para el presente trabajo y en línea con los desarrollos de Sosa Escudero et al. [26] y Alvarez y Svarc [2], la misma debió contemplar el procedimiento de *blinding*, es decir, la ejecución de regresiones no paramétricas y la reclusterización de los datos, y el cálculo de un indicador de bondad de ajuste o *fitness*: tasa correcta de clasificación o ratio de éxito (ratio entre las observaciones correctamente clasificadas respecto del total de observaciones; exactitud o *accuracy*, en inglés). Es necesario poner de resalto

[15] Si bien estos autores consideran el problema de selección de variables en otro contexto, la propuesta es válida para cualquier problema de selección de variables.

[16] Suponiéndose que se tienen 5 variables $(X_1, \ldots, X_5)$, se podrían seleccionar como población inicial las siguientes tres combinaciones: (X_1, X_2 y X_3), (X_1, X_2 y X_4) y (X_1, X_2 y X_5), las que implicarían los siguientes vectores cromosomas o potenciales soluciones a testear y competir, para luego combinarse y/o mutar: (1, 1, 1, 0, 0), (1, 1, 0, 1, 0) y (1, 1, 0, 0, 1).

que dichos procedimientos fueron adaptados a la consideración de variables binarias e implementación de metodologías de clasificación no supervisadas del tipo jerárquico. De este modo, se estableció una función de *fitness* con determinadas nuevas particularidades (ver Algoritmo 2).

Algoritmo 1 Algoritmo Genético (GA)

1: Generar una población al azar de N cromosomas o individuos.
2: Evaluar el *fitness* de cada cromosoma de la población.
3: Crear una nueva población a partir de los siguientes pasos:
   a. Seleccionar pares de cromosomas (padres) según su *fitness* (en general, mejor *fitness*, mayor probabilidad de ser elegido).
   b. Con una cierta probabilidad, combinar los genes del par de cromosomas para obtener uno nuevo.
   c. Con una cierta probabilidad, mutar cada uno de los genes del nuevo cromosoma.
   d. Incorporar la descendencia a la nueva población.
4: Chequear si se satisface el o alguno de los criterios de terminación establecidos. En caso negativo, dirigirse nuevamente al Paso 2.
5: Presentar la mejor solución de la población vigente.

En particular, en lo que se refiere al primer aspecto (consideración de variables binarias), se empleó vecinos más cercanos como estimador no paramétrico para la esperanza condicional, se utilizó la distancia de Gower para la obtención de los vecinos y se realizó el cálculo de la moda, en reemplazo de la media, cuando la variable implicada era del tipo binaria. En tanto, en lo que respecta al segundo aspecto (implementación de metodologías del tipo jerárquico), se reemplazó el cálculo del ratio de éxito, como indicador de bondad de ajuste a maximizar en la función *fitness*, por el del Coeficiente de Correlación Cofenética (CCC; *Cophenetic Correlation Coefficient*, en inglés).

Como ya se señaló en la introducción, el procedimiento desarrollado por Fraiman et al. [10] desafortunadamente no es aplicable a metodologías de clustering distintas a la de por particiones, como por ejemplo, el método jerárquico. Los clusters por particiones se caracterizan por sus centros, por lo que el criterio para determinar la nueva partición con las variables ocultas se basa en asignarlas al centro original más cercano. Al considerar clusters jerárquicos, este criterio deja de tener sentido dado que los mismos están caracterizados por los dendrogramas, que reflejan toda la información contenida en las sucesivas matrices de disimilaridad. Así, en este caso, resulta natural comparar el dendrograma obtenido con las variables originales con aquel resultante de realizar el proceso de *blinding*. Una forma de realizar esta comparación es mediante la correlación cofenética.

El CCC, definido por Sokal y Rohlf [25], mide la correlación producto-momento ordinaria entre los elementos de las matrices de valores de *Cophenetic* (o matrices de distancias de *Cophenetic*) de dos árboles jerárquicos que se quieran comparar, donde el mencionado valor (o distancia) de dos observaciones

es el nivel del árbol (o disimilaridad) donde se conectan sus ramas. Al igual que el Coeficiente de Correlación de Pearson, puede tomar valores entre -1 y 1, indicando 1 un correlación positiva perfecta, -1 una correlación negativa perfecta y 0 que no hay correlación, es decir, ambos árboles no son similares en términos estadísticos. Por lo tanto, el CCC entre dos matrices de distancias de *Cophenetic* Y y Z es:

$$CCC(Y,Z) = \frac{\sum_{i<j}(Y_{i,j} - \overline{Y})(Z_{i,j} - \overline{Z})}{\sqrt{\sum_{i<j}(Y_{i,j} - \overline{Y})^2(Z_{i,j} - \overline{Z})^2}},$$

donde $Y_{i,j}$ (respec. $Z_{i,j}$) es la distancia cofenética entre dos observaciones i y j de la matriz Y (respec. Z) y $\overline{Y}$ (respec. $\overline{Z}$) es el promedio de las distancias de la matriz Y (respec. Z).

Algoritmo 2 Función de *fitness*

1: Procedimiento de *blinding* - parte 1: Correr la regresión no paramétrica llevando adelante los siguientes pasos:
   a. Determinar los k vecinos más cercanos de todas las observaciones considerando las variables (características) presentes en una potencial solución (cromosoma), vía fuerza bruta, utilizando la distancia de Gower.
   b. Asignar a los k vecinos obtenidos los valores correspondientes de cada variable no presente en la potencial solución.
   c. Calcular la media (variable numérica) o la moda (variables binaria) según corresponda de los k valores para cada observación.
   d. Reemplazar en cada variable cada observación por esta predicción obtenida basada en las variables seleccionadas.
2: Procedimiento de *blinding* - parte 2: Repetir el proceso de clusterización con el nuevo conjunto de datos obtenidos, según corresponda:
   - Método por Particiones. K-medoides: Calcular la matriz de disimilaridades respecto de los medoides finales originales y reagrupar las observaciones por criterio de menor distancia a los mismos.
   - Método Jerárquico. Método de Ward: Calcular la nueva matriz de disimilaridades y agrupar nuevamente las variables por el método de Ward.
3: Calcular el indicador de bondad de ajuste o *fitness* (a maximizar) según corresponda:
   - Método por Particiones. K-medoides: Tasa de éxito.
   - Método Jerárquico. Método de Ward: Coeficiente de Correlación Cofenética (CCC).
4: Devolver la tasa de éxito o el CCC, según corresponda.

Una vez implementado el GA con la función de *fitness* particular y reducido así el conjunto de potenciales variables relevantes, resultó viable el testeo y

comparación de todas las combinaciones posibles de variables resultantes hasta esa instancia, a través de la aplicación generalizada de la mencionada función. De esta forma, de todo el conjunto de soluciones factibles, se alcanzó aquella con mayor *accuracy* o CCC, de corresponder, y menor número de variables.

Es de importancia señalar que de obtenerse a partir del GA un subconjunto de variables de tamaño superior al computacionalmente viable, se procedió a la implementación de un algoritmo adicional, denominado algoritmo de selección aleatoria (ASA) a los fines de este trabajo, para reducir su tamaño en forma previa al testeo de todas las combinaciones factibles. En concreto, a partir de las soluciones resultantes del GA, se aplicó la estrategia de ocultamiento en forma aleatoria y de manera individual a las variables resultantes, sobreviviendo cada una solo en caso de obtenerse un *fitness* igual o menor al de la solución que la consideraba (ver Algoritmo 3).

Algoritmo 3 Algoritmo de Selección Aleatoria (ASA)

1: Elegir al azar un variable para ocultar.
2: Aplicar la función de *fitness*.
3: Comparar el *fitness* obtenido respecto del resultante sin ocultar dicha variable:
   a. De resultar superior, ocultar en forma permanente dicha variable.
   b. De resultar igual o inferior, no ocultar dicha variable.
4: Repetir todos los pasos n veces a determinar.

Una vez alcanzada una solución final, ya sea utilizando o no el ASA, se empleó el indicador *Adjusted Rand Index* (ARI), definido por Hubert y Arabie [16], como medida de validación externa para comparar los resultados del procedimiento desarrollado respecto de la agrupación original. El mismo resulta de la corrección del *Rand Index* (RI), indicador definido como la cantidad de pares de observaciones que se encuentran en el mismo cluster (no importa cual) tanto en la agrupación original como en la nueva, como así también que se ubican en clusters distintos en ambas particiones, respecto de la cantidad total de pares posibles de observaciones dados los datos disponibles. En particular, se trata de una corrección por azar, es decir, un ajuste que tiene como objetivo que el RI arroje 0 frente a comparaciones de particiones tomadas al azar (sujetas a la misma cantidad de grupos y observaciones en los mismos).

De este modo, a diferencia del RI, el ARI incorpora, tanto en el numerador como en el denominador, el valor esperado del RI. Utilizándose la forma general de corrección por azar:

$$\frac{\text{Índice} - \text{Valor esperado del índice}}{\text{Índice máximo} - \text{Valor esperado del índice}}$$

y asumiendo un RI máximo de 1, se tiene:

$$ARI = \frac{\sum_{i,j}\binom{n_{i,j}}{2} - \sum_{i}\binom{n_i}{2}\sum_{j}\binom{n_j}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_{i}\binom{n_i}{2} + \sum_{j}\binom{n_j}{2}] - \sum_{i}\binom{n_i}{2}\sum_{j}\binom{n_j}{2}/\binom{n}{2}},$$

donde $\binom{n_{i,j}}{2}$ es el número de pares de observaciones que se encuentran en el cluster i en la agrupación original y en el cluster j en la nueva, $\binom{n_i}{2}$ es el número de pares de observaciones que se encuentran en el cluster i en la agrupación original y que comparten cluster en la nueva, $\binom{n_j}{2}$ es el número de pares de observaciones que se encuentran en el cluster j en la agrupación nueva y que comparten cluster en la original, y $\binom{n}{2}$ es el número total de pares de observaciones que se encuentran en el mismo cluster tanto en la agrupación original, como en el nueva, independientemente si se trata del mismo cluster.

El mismo puede fluctuar entre -1 y 1, donde 1 implica que ambas particiones coinciden perfectamente, 0 cuando el *Rand Index* es igual a su valor esperado, y un valor negativo cuando su valor es inferior a su valor esperado.

CAPÍTULO 4

Resultados

4.1. Principales resultados

El código de todo el presente trabajo se desarrolló en lenguaje R, el cual se presenta en el Anexo 4. Asimismo, está a disposición de quien lo solicite. Previo a su ejecución, fue necesario el establecimiento de ciertos parámetros. En particular, en lo que respecta al GA, se estableció una tasa de combinación de 0,90 y de mutación de 0,10, dado que resultó de interés cubrir la mayor cantidad de posibilidades a partir de los cromosomas o alternativas seleccionadas naturalmente, dejando poca probabilidad de que se consideren variantes ya descartadas o alejadas de las seleccionadas. Asimismo, dada la naturaleza del problema (la identificación de las variables explicativas más relevantes), una vez calculados los *fitness* en cada generación, el mecanismo de selección optado entre generaciones fue el de mayor valor. En tanto, el número de individuos que pasa de generación se estableció en 50, para agilizar los tiempos computacionales. Es importante notar que la modificación de los diferentes hiperparámetros numéricos mencionados no implicó cambios considerables en los resultados.

Finalmente, se incorporó un parámetro β, parámetro de truncamiento de la distribución de valores del *fitness* calculados bajo GA, de forma tal de contar con la posibilidad de considerar soluciones con menor número de variables en detrimento de cierta de pérdida de ajuste. Así, se fijó $\beta = 0,90$, para no alejarse de las mejores soluciones y considerar alternativas con una cantidad de variables manipulables computacionalmente en forma posterior. Con $\beta = 1$, fue necesario recurrir a la aplicación del ASA para alcanzar un número de variables adecuado. Cabe indicar que también se incorporó un parámetro γ, parámetro de truncamiento de la distribución de valores de *fitness* calculados bajo ASA, con el mismo fin que bajo GA.

De esta manera, en lo que respecta al enfoque por particiones a través del algoritmo k-medoides (K-med 2-pasos), a partir de la aplicación del GA, se determinaron 47 variables a priori redundantes/no informativas a ocultar, subsistiendo así 15. Luego de testear en forma exhaustiva todos los posibles subconjuntos considerando la base de datos modificada, es decir, con información ocultada a partir del paso anterior, esta cantidad descendió a 8. Reemplazando todas las variables a priori no informativas/redundantes por sus estimaciones no paramétricas y llevando adelante nuevamente el proceso de clusterización con la nueva base de datos, el ratio de éxito resultó de 94 %, es decir, 64

entidades financieras de 68 mantuvieron el grupo original, solo 4 resultaron mal clasificadas. En términos de ARI, el índice alcanzó 0.85 (ver Cuadro 4.1).

Cuadro 4.1: Resultados del proceso de identificación

	K-med 2-pasos	K-med 3-pasos	Ward 3-pasos
q vbles GA	15	31	30
β	0,900	1,000	0,990
*fitness**	0,897	1,000	0,990
q vbles ASA	N/A	16	15
γ	N/A	0,985	0,995
*fitness**	N/A	0,971	0,993
q vbles ALL	8	7	9
*fitness**	0,941	0,956	0,984
Ratio de éxito	94 %	96 %	85 %
ARI	0,85	0,90	0,69

* Ratio de éxito bajo la metodología k-medoides y CCC bajo el método de Ward.

En tanto, de establecerse $\beta = 1$ y, por ende, aplicando adicionalmente el algoritmo de selección aleatoria (K-med 3-pasos), dado que la mejor solución luego del GA era de 31 variables informativas, se alcanzó como mejor solución 7 y 55 variables informativas y redundantes/no informativas, respectivamente. Dicha solución conllevó un ratio de éxito de 96 % (sólo 3 entidades financieras mal clasificadas) y un ARI 0,90, indicadores aún superiores a los obtenidos flexibilizando el parámetro β.

En relación al enfoque jerárquico a través del método de Ward (Ward 3-pasos), si bien los resultados no fueron tan satisfactorios, resultaron aceptables. Conviene mencionar que no se logró alcanzar una cantidad de soluciones factibles de procesar aún con $\beta = 0,90$ para la aplicación del GA. Por lo tanto, se optó por $\beta = 0,99$, determinándose así 30 variables a priori informativas y 42 no informativas a ocultar. Como resultado de la aplicación del ASA y el posterior testeo exhaustivo de todos los posibles subconjuntos con la base de datos modificada, se obtuvo como mejor solución 9 variables relevantes, con un CCC de 98 %, un ratio de éxito de 85 % y un ARI de 0,69. Así, 58 de 68 entidades mantuvieron su cluster original.

Es del caso mencionar que los resultados de estos últimos indicadores, ratio de éxito y ARI, fueron magros bajo todas las alternativas de procedimiento, si se consideran en el proceso de clusterización final solo las variables seleccionadas, es decir, descartándose y no ocultándose las variables a priori no informativas. Esto refuerza el buen desempeño del proceso de *blinding*.

4.2. Clusters resultantes y variables relevantes

Como se desprende de los párrafos previos, las composiciones de los clusters difirieron marginalmente respecto de los originales, resultando robusto el proceso de identificación frente a diferentes alternativas de clusterización y pasos de los procedimientos. No obstante, se evidenció que las entidades asignadas erróneamente no coincidieron entre los diferentes procesos de selección. En

tal sentido, cabe señalar como debilidad hallada del procedimiento general implementado, que éste resultó muy sensible a las condiciones iniciales de los algoritmos, en particular, a la elección de la semilla, limitación que también observa la metodología del tipo k-medias.

En relación a las variables identificadas como relevantes, en términos generales, las mismas mostraron coincidir parcialmente entre los diferentes métodos. De este modo, en forma exacta, lo hicieron las vinculadas al origen del capital (variables binarias resultado de la transformación de la variable categórica a los fines metodológicos) y la referida a donde está emplazada la casa central/matriz. En tanto, se correspondieron aproximadamente las relacionadas con la concentración de operaciones tradicionales (préstamos y depósitos), la concentración de los puestos de operación (cajeros y terminales), la participación en el mercado y el tamaño físico de la entidad (cantidad de personal y gastos vinculados al funcionamiento) (ver Cuadro 4.2). Estas coincidencias solo parciales responderían en buena medida a la cantidad de variables correlacionadas presentes en la base de datos con la cual se trabajó.

Cuadro 4.2: Variables identificadas

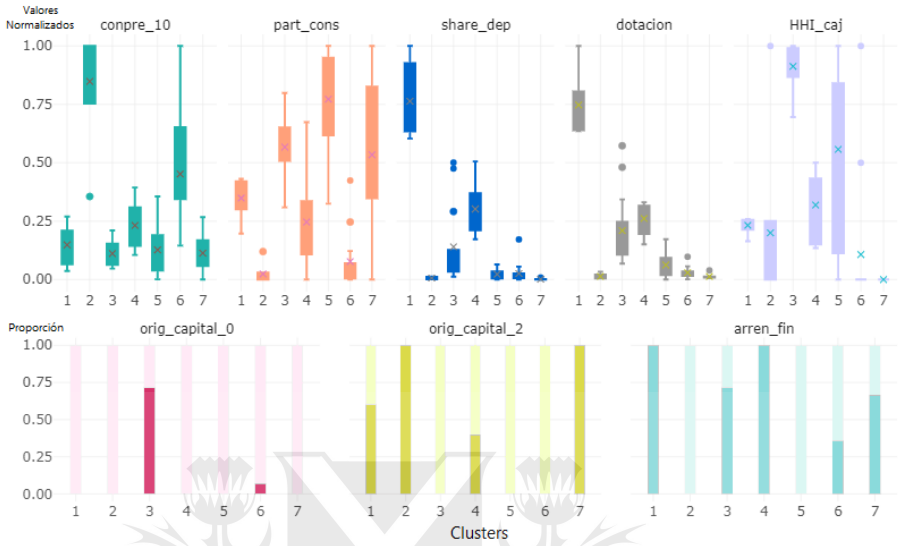
q vbles	K-med 2-pasos	K-med 3-pasos	Ward 3-pasos
1	conpre-10	conpre-60	condep-10
2	part-cons	prest-otros	condep-60
3	share-dep	share-act	q-ctaprev
4	dotación	q-cajau	gasto-adm
5	HHI-caj	HHI-ter	dep-jud
6	arren-fin	casa	casa
7	orig-k0	orig-k2	orig-k0
8	orig-k2		orig-k2
9			tit-fideic

Una cuestión relevante a notar es que, más allá de la técnica de clusterización empleada, el número de algoritmos implementados y la función de *fitness* elegida, la cantidad de variables seleccionadas no descendió por debajo de 7 bajo ningún procedimiento, indicando esto la presencia invariante de varias variables relevantes en la determinación del agrupamiento de las entidades financieras. Estas variables, asimismo, bajo todas las metodologías empleadas, representaron diversas características de las entidades en cuestión: tamaño, naturaleza y tipo de negocio; es decir, las agrupaciones resultantes respondieron a distintos aspectos que describen a las mismas y no sólo a uno de ellos, como por ejemplo tamaño.

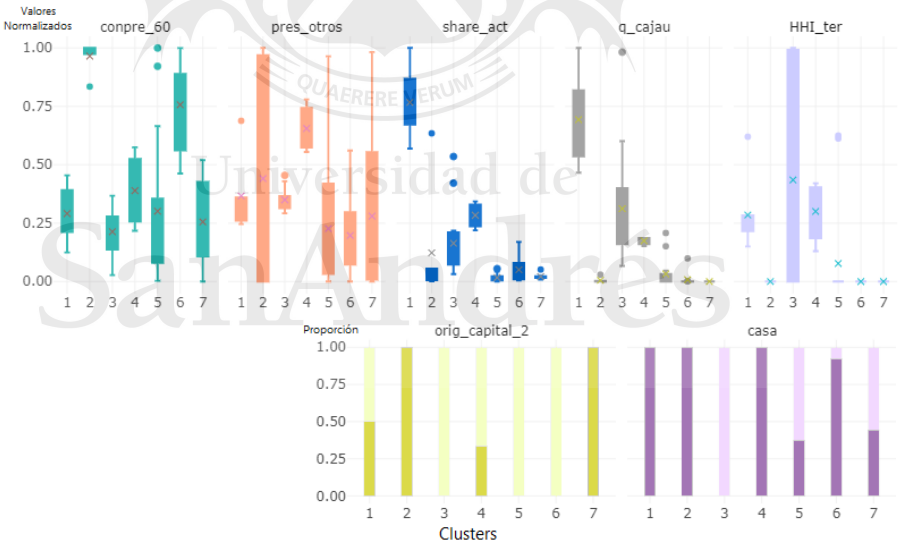
Al ampliarse el conjunto de estas variables (hasta 16), considerándose las variables identificadas como relevantes por los primeros algoritmos, o sea, sin realizar el testeo exhaustivo final, las coincidencias se incrementaron, tomando también relevancia la actividad de las entidades en lo que hace a fideicomisos, arrendamientos financieros y derivados (variables relacionadas con la complejidad de sus operaciones). En tanto, cabría analizarse otras combinaciones de variables resultado de la aplicación del procedimiento completo, es decir, soluciones *second best*, aunque esto implicaría una mayor cantidad de variables involucradas y/o un peor ajuste.

Figura 4.1: Distribución intracluster de las variables relevantes. K-medoides

(a) K-medoides 2-pasos



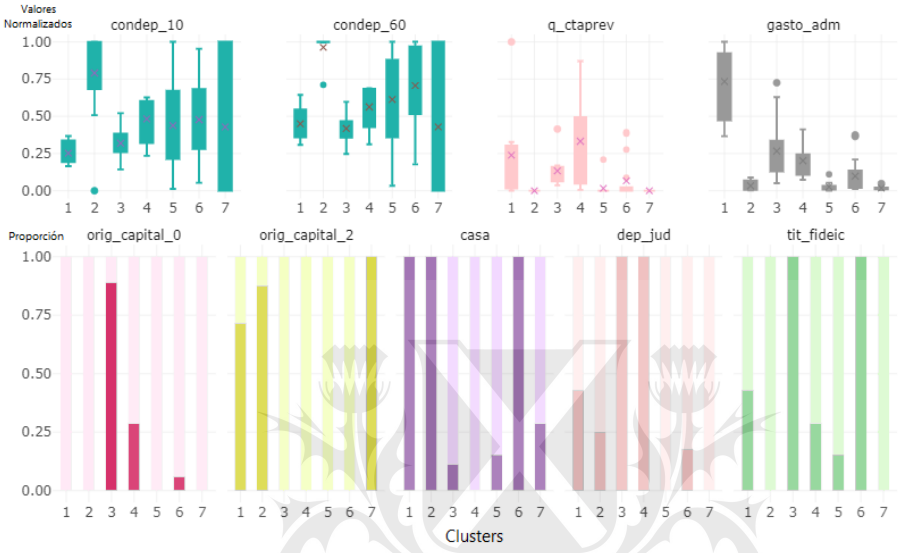
(b) K-medoides 3-pasos



Finalmente, es importante resaltar que del análisis de los valores que tomaron las variables coincidentes en forma exacta en los diferentes clusters finales bajo los distintos métodos de agrupamiento, no se presentaron divergencias sustanciales (ver Figuras 4.1a, 4.1b y 4.2). Asimismo, extendiéndose el análisis de la distribución intracluster al resto de las variables relevantes identificadas, se destacó el poder discriminatorio general entre clusters de algunas de ellas: la importancia relativa de la cartera de consumo en relación a las otras carteras (part-cons), la cantidad de personal (dotación) y la participación de los activos en el total de activos del sistema financiero (share-act), estando éstas dos últimas

vinculadas completamente al tamaño de la entidad. En tanto, las restantes evidenciaron cierto poder discriminatorio, es decir, para determinado cluster en particular.

Figura 4.2: Distribución intracluster de las variables relevantes. Ward 3-pasos



CAPÍTULO 5

Conclusiones

En un marco de creciente disponibilidad de datos y adelantos tecnológicos, resulta oportuno y provechoso el desarrollo de soluciones *Suptech* por parte de los organismos reguladores y/o supervisores, tales como métodos más robustos de agrupamiento de entidades financieras. En este sentido, este trabajo busca profundizar la propuesta metodológica de ocultamiento o *blinding* para la selección de variables en procesos de clusterización por particiones y jerárquicos que involucran datos en alta dimensión y variables mixtas. Para ello, a partir de una cuantía considerable de información pública, primero se agruparon las entidades que conforman el sistema financiero argentino en 7 grupos, los cuales resultan razonables, para luego avanzar en el proceso de selección o identificación de variables relevantes.

Los resultados finales obtenidos son satisfactorios, al lograr alcanzar subconjuntos de entre 7 y 9 variables que reproducen las 7 particiones originales con una tasa de éxito y un ARI en torno al 92 % y 0,81, respectivamente, sobresaliendo el desempeño del método por particiones en tres etapas. Desde el punto de vista puramente práctico, las resultados también son favorables. Las particiones obtenidas son explicadas fundamentalmente por una gama de variables que describen distintos aspectos de las entidades financieras. Aunque, dentro de los mismos, se destaca el criterio tamaño respecto de los demás, lo que resulta lógico considerando las particularidades del sistema financiero local.

De esta forma, no solo se logra ahondar el entendimiento de las agrupaciones resultantes, sino también se confirma la solidez del enfoque *blinding* dado su buen desempeño bajo distinto procedimientos. Adicionalmente, se destaca la implementación exitosa del mismo para métodos de clusterización jerárquicos. Como extensión de este trabajo se podría probar el código R desarrollado con otro conjunto de datos para reafirmar la robustez de los resultados, como así también profundizar en el análisis de sensibilidad frente a cambios en los hiperparámetros.

Anexos

Anexo 1: Detalle de las variables empleadas

1. Concentración accionaria: Sumatoria de la participación de cada accionista elevada al cuadrado (HHI).
2. Concentración accionaria por tipo de accionista: Sumatoria de la participación de cada tipo de accionista elevada al cuadrado (HHI).
3. Concentración de préstamos - 10 mayores clientes: Saldo de deuda 10 mayores clientes/Total préstamos de la entidad financiera.
4. Concentración de préstamos - 60 mayores clientes: Saldo de deuda 60 mayores clientes/Total préstamos de la entidad financiera.
5. Concentración de depósitos - 10 mayores clientes: Saldo de colocación 10 mayores clientes/Total depósitos en la entidad financiera.
6. Concentración de depósitos - 60 mayores clientes: Saldo de colocación 60 mayores clientes/Total depósitos en la entidad financiera.
7. Cantidad de empresas asociadas: Sumatoria de las empresas asociadas.
8. Cantidad de empresas subsidiarias: Sumatoria de las empresas subsidiarias.
9. Participación de la cartera comercial en la cartera total: Financiaciones y garantías otorgadas de cartera comercial/Total carteras de la entidad financiera.
10. Participación de la cartera asimilable en la cartera total: Financiaciones y garantías otorgadas de cartera asimilable a consumo/Total carteras de la entidad financiera.
11. Participación de la cartera consumo en la cartera total: Financiaciones y garantías otorgadas de cartera consumo/Total carteras de la entidad financiera.
12. Participación de los préstamos en el total de préstamos al SPNF: Préstamos al Sector Privado no Financiero y Residentes en el Ext./Total préstamos del sistema. (Incluye Capital, Intereses, Dif. Cotiz., Ajuste NIIF y otros Ajustes; pesos y ME).
13. Participación de los depósitos en el total de depósitos del SPNF: Depósitos del Sector Privado no Financiero y Residentes en el Ext./Total depósitos en el sistema.

14. Opera en ALADI (1: opera, 0: no opera): 1: opera en ALADI; 0: no opera en ALADI.

15. Cantidad de cajas de ahorro de ayuda social: Sumatoria de cajas de ahorros para el pago de planes o programa de ayuda social.

16. Cantidad de cuentas sueldo: Sumatoria de cuentas destinadas al pago de remuneraciones mediante acreditación bancaria.

17. Cantidad de cuentas corrientes: Sumatoria de cuentas con depósitos de efectivo constituidos bajo la forma de cuenta corriente bancaria.

18. Cantidad de cuentas de ahorro: Sumatoria de cuentas con depósitos de efectivo constituidos bajo el sistema de caja de ahorros.

19. Cantidad de cuentas previsionales: Sumatoria de cuentas destinadas al pago de haberes previsionales.

20. Cantidad de empresas que disponen de cuentas sueldo: Sumatoria de empresas con cuentas destinadas al pago de remuneraciones mediante acreditación bancaria.

21. Cantidad de plazo fijos: Sumatoria de operaciones a plazo fijo de individuos y de empresas.

22. Cantidad de operaciones por otros préstamos: Sumatoria de operaciones por otros préstamos.

23. Cantidad de operaciones por préstamos hipotecarios: Sumatoria de operaciones por préstamos hipotecarios.

24. Cantidad de operaciones por préstamos prendarios: Sumatoria de operaciones por préstamos prendarios.

25. Cantidad de tarjetas de crédito (plásticos): Sumatoria de tarjetas de crédito (plásticos).

26. Cantidad de tarjetas de débito: Sumatoria de tarjetas de débito.

27. Cantidad de titulares de tarjetas de crédito: Sumatoria de titulares de tarjetas de crédito.

28. Dotación de personal: Total de dotación de personal.

29. Ubicación de la casa matriz/casa central (1: CABA, 0: No CABA) : 1: casa matriz/casa central en CABA, 0: casa matriz/casa central en otra jurisdicción distinta de CABA.

30. Concentración geográfica de las sucursales: Sumatoria del porcentaje de sucursales planas en cada provincia elevado al cuadrado (HHI); se considera CABA una provincia más.

31. Concentración geográfica de los cajeros: Sumatoria del porcentaje de cajeros automáticos en cada provincia elevado al cuadrado (HHI); se considera CABA una provincia más.

32. Concentración geográfica de las terminales: Sumatoria del porcentaje de terminales de autoservicio en cada provincia elevado al cuadrado (HHI); se considera CABA una provincia más.

33. Cantidad de sucursales plenas: Sumatoria de sucursales planas (incluye casa matriz, casa central operativa y no operativa, y sucursales habilitadas).

34. Cantidad de sucursales de operaciones especificas: Sumatoria de oficinas, dependencias y agencias destinadas a la presentación de determinados servicios.

35. Cantidad de sucursales móviles: Sumatoria de agencias móviles.

36. Cantidad de dependencias automatizadas: Sumatoria de dispositivos y dependencias automáticas.

37. Cantidad de cajeros automáticos: Sumatoria de cajeros automáticos que funcionan dentro y fuera de las casas operativas.

38. Cantidad de terminales de autoservicio: Sumatoria de terminales de autoservicio habilitadas dentro y fuera de las casas operativas.

39. Cantidad de puestos de promoción: Sumatoria de puestos de promoción (incluye asesoramiento y recepción de solicitudes de operaciones).

40. Cantidad de agencias complementarias de servicios financieros: Sumatoria de locales (ej. empresas de cobranzas) que les fueron delegada la prestación de determinados servicios.

41. Activos en el total de activos del sistema financiero: Activos totales/Total activos del sistema.

42. Financiación al SP respecto del total de financiaciones: Préstamos al Sector Público no Financiero y tenencia de títulos públicos/Total préstamos a todos los sectores y tenencia total de títulos de la entidad financiera.

43. Financiación al SF en el financiamiento del sistema financiero: Préstamos al Sector Financiero/Total préstamos del sistema al SF. (Incluye solo capital).

44. Adelantos respecto de total de préstamos al SPNF: Adelantos al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

45. Préstamos documentarios respecto de total de préstamos al SPNF: Doc. a sola firma, descont. y comprados al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

46. Préstamos hipotecarios respecto de total de préstamos al SPNF: Préstamos hipotecarios al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

47. Préstamos prendarios respecto de total de préstamos al SPNF: Préstamos prendarios al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

48. Préstamos personales respecto de total de préstamos al SPNF: Préstamos personales al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

49. Otros préstamos respecto de total de préstamos al SPNF: Otros préstamos al Sector Privado no Financiero y Residentes en el Ext./Total préstamos de la entidad financiera. (Incluye solo capital).

50. Nocionales en derivados en el total del sistema financiero : Nocionales por opciones tomadas y compra/venta a término de moneda extranjera y títulos públicos sin entrega de subyacente/Total nocionales del sistema (Ctas. balance: 711035, 711099, 711101, 711102, 711117, 715099, 715101, 715102, 715117, 721099, 721101, 721102, 721108, 725099, 725101, 725102 y 725108).

51. Créditos por arrendamiento financiero (1: Si, 0: No): 1: Registra montos en la cta.150000 (Créditos por Arrendamiento Financiero) ; 0: no registra montos.

52. Filiales en el exterior (1: Si, 0: No): 1: Posee filiales en el exterior ; 0: no posee filiales en el exterior.

53. Depósitos del SP respecto del total de depósitos: Depósitos del Sector Público/Total depósitos de la entidad.

54. Depósitos judiciales (1: Si, 0: No): 1: Registra montos en las ctas. balance 311113, 311153, 311413, 311725, 311753, 315113, 315153, 315413, 315725 y 315753 (Dep. Jud.Cta.a la Vista y Dep.Jud. a Plazo Fijo) ; 0: no registra montos.

55. Ingresos por intereses respecto del total de ingresos: Ingresos financieros por intereses/Ingresos financieros y por servicios de la entidad.

56. Gastos de administración en el total sistema financiero: Gastos de administración/Total de gastos del sistema.

57. Tenencia de títulos valores de fideicomisos financieros (1: Si, 0: No): 1: Registra montos en las ctas. balance 121022, 125022, 121031, 125031, 121049, 121050, 125051 y 125052 (Títulos privados – Certificados de participación en fideicomisos financieros, Títulos privados – Títulos de deuda de fideicomisos financieros); 0: no registra montos.

58. Cartera fideicomitida (1: Si, 0: No): 1: Registra montos en las ctas. balance 711089, 715089, 721090 y 725090 (Fondos en fideicomisos y Cuenta de actividad fiduciaria acreedora por el contrario); 0: no registra montos.

59. Tenencia de títulos privados en el total del sistema financiero: Tenencia de obligaciones negociables y subordinadas/Total tenencias del sistema (Ctas. balance: 121019, 121020, 121045, 121046, 121047, 121048, 125019, 125020, 125047, 125048, 125049 y 125050).

60. Fondos Comunes de Inversión (1: Si, 0: No) : 1: Registra montos en las ctas. balance 121021 y 125021 (Títulos Privados - Fondos comunes de inversión); 0: no registra montos.

61. Origen del capital: Origen del capital según el grupo institucional de pertenencia (0: público, 1: privado nacional, 2: privado extranjero).

Anexo 2: Entidades financieras a agrupar

Nro.	Código	Entidad	Grupo BCRA
1	7	BANCO DE GALICIA Y BUENOS AIRES S.A.U.	A
2	11	BANCO DE LA NACIÓN ARGENTINA	A
3	14	BANCO DE LA PROVINCIA DE BUENOS AIRES	A
4	15	INDUSTRIAL AND COMMERCIAL BANK OF CHINA	A
5	16	CITIBANK N.A.	A
6	17	BANCO BBVA ARGENTINA S.A.	A
7	20	BANCO DE LA PROVINCIA DE CÓRDOBA S.A.	A
8	27	BANCO SUPERVIELLE S.A.	A
9	29	BANCO DE LA CIUDAD DE BUENOS AIRES	A
10	34	BANCO PATAGONIA S.A.	A
11	44	BANCO HIPOTECARIO S.A.	A
12	45	BANCO DE SAN JUAN S.A.	A
13	65	BANCO MUNICIPAL DE ROSARIO	C
14	72	BANCO SANTANDER ARGENTINA S.A.	A
15	83	BANCO DEL CHUBUT S.A.	B
16	86	BANCO DE SANTA CRUZ S.A.	C
17	93	BANCO DE LA PAMPA SOCIEDAD DE ECONOMÍA M.	B
18	94	BANCO DE CORRIENTES S.A.	B
19	97	BANCO PROVINCIA DEL NEUQUÉN S.A.	B
20	131	BANK OF CHINA LIMITED SUCURSAL BS.AS.	C
21	143	BRUBANK S.A.U.	C
22	147	BANCO INTERFINANZAS S.A.	C
23	150	HSBC BANK ARGENTINA S.A.	A
24	165	JPMORGAN CHASE BANK, NATIONAL ASSOCIATION	C
25	191	BANCO CREDICOOP COOPERATIVO LIMITADO	A
26	198	BANCO DE VALORES S.A.	B
27	247	BANCO ROELA S.A.	C
28	254	BANCO MARIVA S.A.	B
29	259	BANCO ITAU ARGENTINA S.A.	A
30	266	BNP PARIBAS	C
31	268	BANCO PROVINCIA DE TIERRA DEL FUEGO	C
32	269	BANCO DE LA REP. ORIENTAL DEL URUGUAY	C
33	277	BANCO SAENZ S.A.	C
34	281	BANCO MERIDIAN S.A.	C
35	285	BANCO MACRO S.A.	A
36	299	BANCO COMAFI SOCIEDAD ANÓNIMA	B
37	300	BANCO DE INVERSIÓN Y COMERCIO EXTERIOR	B
38	301	BANCO PIANO S.A.	C
39	305	BANCO JULIO SOCIEDAD ANÓNIMA	C
40	309	BANCO RIOJA SOCIEDAD ANÓNIMA UNIPERSONAL	C
41	310	BANCO DEL SOL S.A.	C
42	311	NUEVO BANCO DEL CHACO S. A.	B
43	312	BANCO VOII S.A.	C

continuación ...

... continuación

Nro.	Código	Entidad	Grupo BCRA
44	315	BANCO DE FORMOSA S.A.	B
45	319	BANCO CMF S.A.	B
46	321	BANCO DE SANTIAGO DEL ESTERO S.A.	B
47	322	BANCO INDUSTRIAL S.A.	B
48	330	NUEVO BANCO DE SANTA FE SOCIEDAD ANÓNIMA	A
49	332	BANCO DE SERVICIOS FINANCIEROS S.A.	C
50	338	BANCO DE SERVICIOS Y TRANSACCIONES S.A.	C
51	339	RCI BANQUE S.A.	C
52	340	BACS BANCO DE CRÉDITO Y SECURITIZACIÓN S.A.	C
53	341	BANCO MASVENTAS S.A.	C
54	384	WILOBANK S.A.U.	C
55	386	NUEVO BANCO DE ENTRE RÍOS S.A.	B
56	389	BANCO COLUMBIA S.A.	C
57	426	BANCO BICA S.A.	C
58	431	BANCO COINAG S.A.	C
59	432	BANCO DE COMERCIO S.A.	C
60	435	BANCO SUCRÉDITO REGIONAL S.A.U.	C
61	448	BANCO DINO S.A.	C
62	44077	COMPAÑÍA FINANCIERA ARGENTINA S.A.	C
63	44088	VOLKSWAGEN FINANCIAL SERVICES COMPAÑÍA F.	C
64	44090	IUDU COMPAÑÍA FINANCIERA S.A.	C
65	44092	FCA COMPAÑÍA FINANCIERA S.A.	C
66	44093	GPAT COMPAÑÍA FINANCIERA S.A.U.	C
67	44094	MERCEDES-BENZ COMPAÑÍA FINANCIERA ARG.	C
68	44095	ROMBO COMPAÑÍA FINANCIERA S.A.	C
69	44096	JOHN DEERE CREDIT COMPAÑÍA FINANCIERA S.	B
70	44098	PSA FINANCE ARGENTINA COMPAÑÍA FINANCIERA	C
71	44099	TOYOTA COMPAÑÍA FINANCIERA DE ARGENTINA	C
72	45056	MONTEMAR COMPAÑÍA FINANCIERA S.A.	C
73	45072	REBA COMPAÑIA FINANCIERA S.A.	C
74	65203	CRÉDITO REGIONAL COMPAÑÍA FINANCIERA S.A	C

Anexo 3: Detalle de los clusters resultantes

Código	Entidad	Kmed	Ward
15	INDUSTRIAL AND COMMERCIAL BANK OF CHINA	1	1
27	BANCO SUPERVIELLE S.A.	1	1
34	BANCO PATAGONIA S.A.	1	1
150	HSBC BANK ARGENTINA S.A.	1	1
191	BANCO CREDICOOP COOPERATIVO LIMITADO	1	1
16	CITIBANK N.A.	2	1
131	BANK OF CHINA LIMITED SUCURSAL B.A.	2	2
165	JPMORGAN CHASE BANK, NATIONAL ASSOCIATION	2	2
266	BNP PARIBAS	2	2
269	BANCO DE LA REP. ORIENTAL DEL URUGUAY	2	2
339	RCI BANQUE S.A.	2	2
20	BANCO DE LA PROVINCIA DE CÓRDOBA S.A.	3	3
29	BANCO DE LA CIUDAD DE BUENOS AIRES	3	3
45	BANCO DE SAN JUAN S.A.	3	4
65	BANCO MUNICIPAL DE ROSARIO	3	3
83	BANCO DEL CHUBUT S.A.	3	3
86	BANCO DE SANTA CRUZ S.A.	3	4
93	BANCO DE LA PAMPA SOCIEDAD DE ECONOMÍA M.	3	3
94	BANCO DE CORRIENTES S.A.	3	3
97	BANCO PROVINCIA DEL NEUQUÉN S.A.	3	3
268	BANCO PROVINCIA DE TIERRA DEL FUEGO	3	3
309	BANCO RIOJA SOCIEDAD ANÓNIMA UNIPERSONAL	3	3
311	NUEVO BANCO DEL CHACO S. A.	3	3
315	BANCO DE FORMOSA S.A.	3	4
321	BANCO DE SANTIAGO DEL ESTERO S.A.	3	4
330	NUEVO BANCO DE SANTA FE SOCIEDAD ANÓNIMA	3	4
386	NUEVO BANCO DE ENTRE RÍOS S.A.	3	4
44	BANCO HIPOTECARIO S.A.	4	6
259	BANCO ITAU ARGENTINA S.A.	4	1
299	BANCO COMAFI SOCIEDAD ANÓNIMA	4	6
322	BANCO INDUSTRIAL S.A.	4	6
143	BRUBANK S.A.U.	5	5
301	BANCO PIANO S.A.	5	5
310	BANCO DEL SOL S.A.	5	5
312	BANCO VOII S.A.	5	6
341	BANCO MASVENTAS S.A.	5	5
384	WILOBANK S.A.U.	5	5
389	BANCO COLUMBIA S.A.	5	5
426	BANCO BICA S.A.	5	5
431	BANCO COINAG S.A.	5	5
435	BANCO SUCRÉDITO REGIONAL S.A.U.	5	5
448	BANCO DINO S.A.	5	5
44077	COMPAÑÍA FINANCIERA ARGENTINA S.A.	5	5
44090	IUDU COMPAÑÍA FINANCIERA S.A.	5	7

continuación ...

. . . continuación

Código	Entidad	Kmed	Ward
45056	MONTEMAR COMPAÑÍA FINANCIERA S.A.	5	5
65203	CRÉDITO REGIONAL COMPAÑÍA FINANCIERA S.A	5	5
147	BANCO INTERFINANZAS S.A.	6	6
198	BANCO DE VALORES S.A.	6	6
247	BANCO ROELA S.A.	6	5
254	BANCO MARIVA S.A.	6	6
277	BANCO SAENZ S.A.	6	6
281	BANCO MERIDIAN S.A.	6	6
300	BANCO DE INVERSIÓN Y COMERCIO EXTERIOR	6	6
305	BANCO JULIO SOCIEDAD ANÓNIMA	6	5
319	BANCO CMF S.A.	6	6
338	BANCO DE SERVICIOS Y TRANSACCIONES S.A.	6	6
340	BACS BANCO DE CRÉDITO Y SECURITIZACIÓN S.A.	6	6
432	BANCO DE COMERCIO S.A.	6	5
45072	REBA COMPAÑÍA FINANCIERA S.A.	6	5
332	BANCO DE SERVICIOS FINANCIEROS S.A.	7	7
44088	VOLKSWAGEN FINANCIAL SERVICES COMP. FIN.	7	7
44092	FCA COMPAÑÍA FINANCIERA S.A.	7	7
44093	GPAT COMPAÑÍA FINANCIERA S.A.U.	7	7
44094	MERCEDES-BENZ COMPAÑÍA FINANCIERA ARG.	7	7
44095	ROMBO COMPAÑÍA FINANCIERA S.A.	7	7
44096	JOHN DEERE CREDIT COMPAÑÍA FINANCIERA S.A.	7	7
44098	PSA FINANCE ARGENTINA COMPAÑÍA FINANCIERA	7	7
44099	TOYOTA COMPAÑÍA FINANCIERA DE ARGENTINA	7	7

Anexo 4: Código R implementado

Clusterización

```
### Preliminares
#### Se traen las librerias necesarias.
library(tidyverse)
library(readxl)
library(writexl)
library(repr)
library(cluster)
library(dendextend)
library(factoextra)
library(FNN)
library(mclust)
library(NbClust)
library(dplyr)
library(GA)
library(ggplot2)
library(reshape2)
library(psych)
library(StatMatch)
library(aricode)
library(caret)
library(gower)
library(plotly)
library(xtable)
library(ggdendro)
library(hrbrthemes)

#### Se carga la base.
base = read_excel("C:/Users/gamar/Desktop/Maestria/Data_Tesis/base_FINAL.xlsx",
sheet = "base jun22 (ene)")

#### Se trabaja con las primeras 65 variables (66 con el nombre de los bcos).
df = base[,1:66]
#str(df)

#### Se termina de acomodar la base.
df = as.data.frame(df)
row.names(df) = df[,1]
df = df[,-1]
vbles = colnames(df)
entidades = rownames(df)

#### Se les asigna el formato correcto a cada vble (4 categoricas y 9 binarias).
df$aladi = as.factor(df$aladi)
df$grupo_h = as.factor(df$grupo_h)
df$casa = as.factor(df$casa)
df$asocia1 = as.factor(df$asocia1)
df$asocia2 = as.factor(df$asocia2)
df$arren_fin = as.factor(df$arren_fin)
df$dep_jud = as.factor(df$dep_jud)
df$fil_ext = as.factor(df$fil_ext)
df$tit_fideic = as.factor(df$tit_fideic)
df$cart_fideic = as.factor(df$cart_fideic)
df$fci = as.factor(df$fci)
df$digital = as.factor(df$digital)
df$orig_capital = as.factor(df$orig_capital)
```

```
### Clusterizacion
#### Se eliminan las vbles que no interesan por implic. generar una descripcion.
```

```
dfbin = df
dfbin = dfbin %>% select(-c("grupo_h", "asocia1", "asocia2", "digital"))

#### Se transforman a vbles binarias las categoricas con dos niveles.
dfbin$aladi = ifelse(dfbin$aladi == "0",0, 1)
dfbin$casa = ifelse(dfbin$casa == "0",0, 1)
dfbin$arren_fin = ifelse(dfbin$arren_fin == "0",0, 1)
dfbin$dep_jud = ifelse(dfbin$dep_jud == "0",0, 1)
dfbin$fil_ext = ifelse(dfbin$fil_ext == "0",0, 1)
dfbin$tit_fideic = ifelse(dfbin$tit_fideic == "0",0, 1)
dfbin$cart_fideic = ifelse(dfbin$cart_fideic == "0",0, 1)
dfbin$fci = ifelse(dfbin$fci == "0",0, 1)

#### Se transforman a vbles binarias las categoricas con mas de dos niveles.
orig_capital = as.data.frame(dummy.code(dfbin$orig_capital))
orig_capital = rename(orig_capital, orig_capital_0 = "0")
orig_capital = rename(orig_capital, orig_capital_1 = "1")
orig_capital = rename(orig_capital, orig_capital_2 = "2")
dfbin = cbind(dfbin, orig_capital)
dfbin = dfbin %>% select(-c("orig_capital"))

#### Se normaliza (Min-Max scaling).
process = preProcess(dfbin, method=c("range"))
dfbin = predict(process, dfbin)

#### Se calcula matriz de disimilaridad aplicando Gower.
#Se transforma la ex vble categorica a logica para que sea
#considerada binaria asimetrica (no simetrica).
dfbin$orig_capital_0 = as.logical(dfbin$orig_capital_0)
dfbin$orig_capital_1 = as.logical(dfbin$orig_capital_1)
dfbin$orig_capital_2 = as.logical(dfbin$orig_capital_2)
dist_dfbin = as.dist(gower.dist(dfbin))
dfbin$orig_capital_0 = as.numeric(dfbin$orig_capital_0)
dfbin$orig_capital_1 = as.numeric(dfbin$orig_capital_1)
dfbin$orig_capital_2 = as.numeric(dfbin$orig_capital_2)

#### Se establece un nro de clusters preliminar.
nc = 8

#### Se prueba el metodo jerarquico simple para ver outliers.
dfbin_single = hclust(dist_dfbin, method="single")
fviz_dend(dfbin_single, k=nc, cex=0.5, main="single")

#### Se eliminan los bancos (outlieres) que no interesan.
dfbin = dfbin[-c(1, 2, 3, 6, 14, 35), ]
dfbin = dfbin %>% select(-c("fil_ext"))  #no tiene sentido al sacar BNA y Bapro

#### Se recalcula matriz de disimilaridad
dfbin$orig_capital_0 = as.logical(dfbin$orig_capital_0)
dfbin$orig_capital_1 = as.logical(dfbin$orig_capital_1)
dfbin$orig_capital_2 = as.logical(dfbin$orig_capital_2)
dist_dfbin = as.dist(gower.dist(dfbin))
dfbin$orig_capital_0 = as.numeric(dfbin$orig_capital_0)
dfbin$orig_capital_1 = as.numeric(dfbin$orig_capital_1)
dfbin$orig_capital_2 = as.numeric(dfbin$orig_capital_2)

#### Se analiza la cantidad de clusters posibles bajo kmeans
fviz_nbclust(dfbin, pam, diss=dist_dfbin, method="wss") +
  labs(subtitle="Elbow") + geom_vline(xintercept=3, linetype=2)
fviz_nbclust(dfbin, pam, diss=dist_dfbin, method="silhouette") +
  labs(subtitle = "Max Min Silhouette")
fviz_nbclust(dfbin, pam, diss=dist_dfbin, method="gap_stat") +
```

```
  labs(subtitle = "Gap Statistic")

#### Se analiza la cantidad de clusters posibles bajo hclust.
fviz_nbclust(dfbin, hcut, diss=dist_dfbin, method = "wss",
  hc_method="ward.D2") + labs(subtitle = "Elbow")
fviz_nbclust(dfbin, hcut, diss=dist_dfbin, method = "silhouette",
  hc_method="ward.D2") + labs(subtitle = "Max Min Silhouette")
fviz_nbclust(dfbin, hcut, diss=dist_dfbin, method = "gap_stat",
  hc_method="ward.D2")+labs(subtitle = "Gap Statistic")

#### Se establece el nro de clusters.
nc = 7

#### Se conforman clusters por k-medoids.
dfbin_pam = pam(dist_dfbin, diss = TRUE, nc)
dfbin_clus = dfbin_pam$clustering
dfbin_med = dfbin_pam$id.med
dfbin_clus

#### Se conforman clusters por metodo jerarquico completo, promedio y ward.d2.
dfbin_comp = hclust(dist_dfbin, method="complete")
fviz_dend(dfbin_comp, k=nc, cex=0.5, main="COMPLETE")
dfbin_compT = cutree(dfbin_comp, k=nc)

dfbin_aver = hclust(dist_dfbin, method="average")
fviz_dend(dfbin_aver, k=nc, cex=0.5, main="average")
dfbin_averT = cutree(dfbin_aver, k=nc)

dfbin_ward2 = hclust(dist_dfbin, method="ward.D2")
fviz_dend(dfbin_ward2, k=nc, cex=0.5, main="WARD.D2")
dfbin_ward2T = cutree(dfbin_ward2, k=nc)
dend_orig = as.dendrogram(dfbin_ward2)

#### Se exporta solucion.
clusters = as.data.frame(cbind(row.names(dfbin),dfbin_compT, dfbin_averT,
    dfbin_ward2T, dfbin_clus))
write_xlsx(clusters,"C:/Users/gamar/Desktop/Maestria/Tesis/clusters_FINAL.xlsx")
```

Identificación de variables relevantes

```
### Preliminares
#### Se establecen definiciones iniciales para la f de fitness.

#Otras variables necesarias para la funcion de fitness.
X = dfbin; clusters = dfbin_clus; medoides = dfbin_med

#Funcion para calcular la moda.
moda = function(x) {
  u = unique(x)
  tab = tabulate(match(x, u))
  u[tab == max(tab)]
}

#Funcion para calcular knn por Gower.
gower.knn = function(x, y, k){
  matriz = gower.dist(data.x=x, data.y=y)
  indices = apply(matriz, 1, order)
  indices.k = indices[1:k,]
  indices.k = t(indices.k)
  return(indices.k)
}
```

```
### Clusterizacion con Kmed - 2 pasos
#### 1) Se corre un GA para achicar las alter. de combinaciones de vbles.
#### Se contruye la funcion de FITNESS.
f_fit = function(cromo, op, k, beta) {
  if (sum(cromo==1)>0){

    #Se determinan los k vecinos mas cercanos de todos los registros a
    #partir de las vbles con 1 y obtengo sus indices.
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)

    X1 = X[,which(cromo==1)]

    nn = gower.knn(x=X1, y=X1, k=k+1)   #k vecinos mas cercanos de c/obs.
    nn = nn[,-c(1)]    #se elimina el primer vecino por ser la misma obs.

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)

    #Se construyen los nuevos valores de las vbles que toman 0 y rearma la base.
    Xnew = X

    X0 = X[,which(cromo==0)]          #matriz con las vbles que toman 0.
    q_X0 = ncol(X0)                   #nro de vbles que toman 0.

    X0new = c()
    for (i in 1:q_X0){                #se toma una por una las vbles que toman 0.
      X0_ind = X0[,i]
      nn_valor = nn
      nn_valor[ ] = X0_ind[c(nn)]     #se le asigna a nn indices el valor que le
                                      #corresponde de la vble tomada.
      if (all(X0_ind %in% 0:1)){      #se calcula el promedio o la moda segun
                                      #corresponda (cont/disc).
        nn_prom = apply(nn_valor, 1, moda)
      } else {
        nn_prom = apply(nn_valor, 1, mean)
      }
      X0new = cbind(X0new, nn_prom)  #se agrupan las "nuevas" variables.
    }

    X0new = as.data.frame(X0new)
    colnames(X0new) = colnames(X0)   #se le ponen los nombres a las variables.

    Xnew[,which(cromo==0)] = X0new   #se reemplaza con las "nuevas" variables.

    #Se asignan las nuevas observaciones a los clusters existentes.
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)
    Xnew$orig_capital_0 = as.logical(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.logical(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.logical(Xnew$orig_capital_2)

    Xmed = X[medoides,]              #se queda con los medoides originales.
    #Se calcula la matriz de disimil respecto de medoides org.
    dist_medoid = gower.dist(data.x=Xmed, data.y=Xnew)
    #Se queda con el cluster de menor dist de cada agente.
    clustersnew = apply(dist_medoid, 2, which.min)

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
```

```
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)
    Xnew$orig_capital_0 = as.numeric(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.numeric(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.numeric(Xnew$orig_capital_2)

    #Se comparan asignaciones y calcula ratio de exitos.
    dif = ifelse(clusters-clustersnew == 0, 1, 0)
    psuccess = sum(dif)/nrow(X)
  } else {
    psuccess=0
  }
  #Se establece un limite superior para el psuccess.
  limpsuccess = ifelse(psuccess-beta>0,0,psuccess)  #beta: tolerancia en la max.

  #Se devuelve el resultado.
  if (op==1) {          #op=1 devuelve p de suceso resultante,
                        #op=2 devuelve cluster de la reg.
    return(limpsuccess)
  } else {
    return(clustersnew)
  }
}

#### Se corre FUNCION GA para aplicar el algoritmo genetico.
GA = ga(type = "binary",                #optimization data type
        fitness = function(cromo) f_fit(cromo, op=1, k=3, beta=0.90), #ff
        nBits = ncol(X),                #total number of variables
        #population = gabin_Population, #f que det la pobl. inicial (binary)
        #selection = lrSelection,        #f que det la nueva gener. (linear-rank)
        crossover = gabin_uCrossover,   #f para generar el crossover (uniform)
        #mutation = raMutation,          #f para generar la mutacion (uniform)
        popSize = 100,                  #the number of individuals
        pcrossover = 0.9,               #crossover rate prob (default 0.8)
        pmutation = 0.1,                #mutation rate prob (default 0.1)
        elitism = max(1, round(1000 * 0.05)), #the number of individuals that
                                              #pass to next iteration (default)
        monitor = plot,                 #plot the result at each iteration
        maxiter = 50,                   #total runs or generations
        run=15,                         #stopping criteria
        #maxFitness = -Inf,             #upper bound de la fitness
        names = colnames(X),
        keepBest = TRUE,                #keep the best solution at the end
        suggestions = diag(rep(1,ncol(X))), #matriz de soluciones para inicio
        seed = 36                       #for reproducibility purposes
)
summary(GA)
plot(GA)

#### Se analizan los resultados.
GA@summary
psuccess_ga = GA@fitnessValue
sol_ga = as.data.frame(GA@solution)
g_sol_ga = apply(sol_ga,1,sum)
g_sol_ga
min(g_sol_ga)
psuccess_ga

#### 2) Se testean todas las combinaciones resultantes y se valida eleccion.
#### Se contruye una f que permita det que vbles explican la clusterizacion.
fvr = function(Xgen){                   #Xgen=base resultante del GA.
```

```
  #Se crea una grilla con todas combin. de activacion de las vbles elegidas.
  comb = expand.grid(rep(list(0:1), ncol(Xgen)))
  comb = comb[-c(1),]                  #se elimina la 1ra comb. por ser todo 0.
  q_comb = nrow(comb)                  #nro de combinaciones finales.
  colnames(comb) = colnames(Xgen)

  Xcomb = as.data.frame(matrix(0,q_comb,ncol(X))) #se agregan las vbles
                                       #descartadas por el GA pero con 0.
  colnames(Xcomb) = colnames(X)
  Xcomb[c(colnames(comb))] = comb

  #Se calcula la probabilidad de exito para cada combinacion.
  f_obj = c()                          #funcion objetivo.
  for (j in 1:q_comb){
    Xcomb_j = Xcomb[j,]
    p_success = f_fit(Xcomb_j, op=1, k=3, beta=1)
    f_obj = cbind(f_obj, p_success)
  }

  #Se ordenan y observan las mejores combinaciones.
  psucess_final = t(f_obj)
  mejor_comb = cbind(comb, psucess_final)
  total = rowSums(mejor_comb[,-(length(mejor_comb))])
  mejor_comb = cbind(mejor_comb, total)
  mejor_comb = mejor_comb[order(-psucess_final, total),]
  return(mejor_comb)
}

#### Se le aplica a la funcion las salidas de GA.
Xga_ind = GA@solution[which.min(g_sol_ga),]  #vec con la fila que nos interesa.
Xga_ind = Xga_ind[Xga_ind==1]                #vector con las vbles relevantes.
Xgen = X[,names(Xga_ind)]                    #base con las vbles elegidas.
sol = fvr(Xgen)
sol

#### Se clusteriza con la mejor solucion.
sol_ind = sol[1,]                            #se queda con la mejor solucion.
sol_ind = sol_ind %>% select(-c("psucess_final", "total"))
dffin = dfbin[colnames(sol_ind[ ,which(sol_ind==1)])] #valores de la mejor sol.

#Se calcula k-medoids.
if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.logical(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.logical(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
    dffin$orig_capital_2=as.logical(dffin$orig_capital_2)
    }

dist_dffin = gower.dist(dffin)
dffin_pam = pam(dist_dffin, diss = TRUE, nc)
dffin_clus = dffin_pam$clustering; dffin_med = dffin_pam$id.med

if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.numeric(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.numeric(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
```

```
    dffin$orig_capital_2=as.numeric(dffin$orig_capital_2)
    }

#### Se clusteriza con la mejor sol + reg no param para vbles redundantes y no inf.
sol_ind1 = sol_ind[,sol_ind==1]
sol_ind01 = as.data.frame(rbind(colnames(dfbin), rep(0, dim(dfbin)[2])))
colnames(sol_ind01) = sol_ind01[1,]
sol_ind01 = sol_ind01[-1, ]
sol_ind01[,colnames(sol_ind[ ,which(sol_ind==1)])] = 1
dfreg_clus = f_fit(sol_ind01, op=0, k=3, beta=1)

#### Se comparan los clusters.
ari_bin_vs_fin = ARI(dfbin_clus, dffin_clus)
ari_bin_vs_reg = ARI(dfbin_clus, dfreg_clus)
ari_bin_vs_fin
ari_bin_vs_reg

### Se exporta solucion.
comp_clustKmed = as.data.frame(cbind(dfbin_clus, dffin_clus, dfreg_clus))
write_xlsx(comp_clustKmed,
    "C:/Users/gamar/Desktop/Maestria/Data_Tesis/comp_clustKmed_FINAL.xlsx")
```

```
### Clusterizacion con Kmed - 3 pasos
#### 1) Se corre un GA para achicar las alternativas de combinaciones de vbles.
#### Se contruye la funcion de FITNESS.
f_fit = function(cromo, op, k, beta) {
  if (sum(cromo==1)>0){

    #Se determinan los k vecinos mas cercanos de todos los registros a
    #partir de las vbles con 1 y obtengo sus indices.
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)

    X1 = X[,which(cromo==1)]

    nn = gower.knn(x=X1, y=X1, k=k+1)   #k vecinos mas cercanos de c/obs.
    nn = nn[,-c(1)]          #se elimina el primer vecino por ser la misma obs.

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)

    #Se construyen los nuevos valores de las vbles que toman 0 y rearma la base.
    Xnew = X

    X0 = X[,which(cromo==0)]       #matriz con las vbles que toman 0.
    q_X0 = ncol(X0)                #nro de vbles que toman 0.

    X0new = c()
    for (i in 1:q_X0){             #se toma una por una las vbles que toman 0.
      X0_ind = X0[,i]
      nn_valor = nn
      nn_valor[ ] = X0_ind[c(nn)]  #se le asigna a nn/indices el valor que le
                                   #corresponde de la vble tomada.
      if (all(X0_ind %in% 0:1)){   #se calcula el promedio o la moda segun
                                   #corresponda (cont/disc).
        nn_prom = apply(nn_valor, 1, moda)
      } else {
        nn_prom = apply(nn_valor, 1, mean)
      }
      X0new = cbind(X0new, nn_prom)  #se agrupan las "nuevas" variables.
```

```
    }

    X0new = as.data.frame(X0new)
    colnames(X0new) = colnames(X0)   #se le ponen los nombres a las variables.

    Xnew[,which(cromo==0)] = X0new   #se reemplaza con las "nuevas" variables.

    #Se asignan las nuevas observaciones a los clusters existentes.
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)
    Xnew$orig_capital_0 = as.logical(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.logical(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.logical(Xnew$orig_capital_2)

    Xmed = X[medoides,]              #se queda con los medoides originales.
    #Se calcula la matriz de disimil respecto de medoides org.
    dist_medoid = gower.dist(data.x=Xmed, data.y=Xnew)
    #Se queda con el cluster de menor dist de cada agente.
    clustersnew = apply(dist_medoid, 2, which.min)

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)
    Xnew$orig_capital_0 = as.numeric(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.numeric(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.numeric(Xnew$orig_capital_2)

    #Se comparan asignaciones y calcula ratio de exitos.
    dif = ifelse(clusters-clustersnew == 0, 1, 0)
    psuccess = sum(dif)/nrow(X)
  } else {
    psuccess=0
  }
  #Se establece un limite superior para el psuccess.
  limpsuccess = ifelse(psuccess-beta>0,0,psuccess) #beta: tolerancia en la max.

  #Se devuelve el resultado.
  if (op==1) {              #op=1 devuelve p de suceso resultante,
                            #op=2 devuelve cluster de la reg.
    return(limpsuccess)
  } else {
    return(clustersnew)
  }
}

#### Se corre FUNCION GA para aplicar el algoritmo genetico.
GA = ga(type = "binary",                #optimization data type
        fitness = function(cromo) f_fit(cromo, op=1, k=3, beta=1),  #ff
        nBits = ncol(X),                #total number of variables
        #population = gabin_Population, #f que det la pobl. inicial (binary)
        #selection = lrSelection,       #f que det la nueva gener. (linear-rank)
        crossover = gabin_uCrossover,   #f para el crossover (uniform)
        #mutation = raMutation,         #f para la mutacion (uniform)
        popSize = 100,                  #the number of individuals
        pcrossover = 0.9,               #crossover rate prob (default 0.8)
        pmutation = 0.1,                #mutation rate prob (default 0.1)
        elitism = max(1, round(1000 * 0.05)), #the number of individuals that
                                              #pass to next iteration (default)
        monitor = plot,                 #plot the result at each iteration
        maxiter = 50,                   #total runs or generations
        run=15,                         #stopping criteria
```

```
        #maxFitness = -Inf,            #upper bound de la fitness
        names = colnames(X),
        keepBest = TRUE,               #keep the best solution at the end
        suggestions = diag(rep(1,ncol(X))), #matriz de soluciones para inicio
        seed = 36                      #for reproducibility purposes
)
summary(GA)
plot(GA)

#### Se analizan los resultados.
GA@summary
psuccess_ga = GA@fitnessValue
sol_ga = as.data.frame(GA@solution)
g_sol_ga = apply(sol_ga,1,sum)
sol_ga_ind = sol_ga[which.min(g_sol_ga),] #vec con la fila que nos interesa.
g_sol_ga
min(g_sol_ga)
psuccess_ga

#### Se contruye una f INTERMEDIA para acotar el conjunto de vbles del GA.
fvr0 = function(alfa, gama, seed){
  i=1
  set.seed(seed)
  while(i<alfa){      #alfa: cantidad de iteraciones para sacar vbles al azar.

    #Se elige al azar una vble para no considerar.
    sol_ga1 = sol_ga_ind[,which(sol_ga_ind==1)]
    vble_rand = sample(1:ncol(sol_ga1),1)
    sol_ga1_new = sol_ga1[-c(vble_rand)]

    #Se agregan las vbles descartadas por el GA y el azar pero con 0.
    sol_ga_new = as.data.frame(matrix(0,1,ncol(sol_ga)))
    colnames(sol_ga_new) = colnames(sol_ga)
    sol_ga_new[c(colnames(sol_ga1_new))] = sol_ga1_new

    #Se calcula el coef de correlacion para para la nueva alternativa.
    psuccess_new = f_fit(sol_ga_new, op=1, k=3, beta=1)

    #Se compara el coef de correlacion obtenido con la vble eliminada con el
    #ultimo obtenido.
    #Si el nuevo coef de correl. es mayor o igual me quedo con la nueva base.
    if (psuccess_new-psuccess_ga*gama >= 0){
      psuccess_ga = psuccess_new                #gama: tolerancia en la max.
      sol_ga_ind = sol_ga_new
    } else {
      psuccess_ga = psuccess_ga
      sol_ga_ind = sol_ga_ind
    }
    i=i+1
  }
  resultado = list("psuccess" = psuccess_ga, "q_vles"= sum(sol_ga_ind),
    "sol" = sol_ga_ind)
  return(resultado)
}

#### Se le aplica a la funcion las salidas de GA.
sol_int = fvr0(alfa=1000, gama=0.985, seed=34)
sol_int

#### 2) Se testean todas las combinaciones resultantes y se valida eleccion.
#### Se contruye una f que permita det que vbles explican la clusterizacion.
fvr = function(Xgen){                  #Xgen=base resultante del GA.
```

```
  #Se crea una grilla con todas las combin. de activacion de las vbles elegidas.
  comb = expand.grid(rep(list(0:1), ncol(Xgen)))
  comb = comb[-c(1),]                  #se elimina la 1ra comb. por ser todo 0.
  q_comb = nrow(comb)                  #nro de combinaciones finales.
  colnames(comb) = colnames(Xgen)

  Xcomb = as.data.frame(matrix(0,q_comb,ncol(X))) #se agregan las vbles
                                       #descartadas por el GA pero con 0.
  colnames(Xcomb) = colnames(X)
  Xcomb[c(colnames(comb))] = comb

  #Se calcula la probabilidad de exito para cada combinacion.
  f_obj = c()                          #funcion objetivo.
  for (j in 1:q_comb){
    Xcomb_j = Xcomb[j,]
    p_success = f_fit(Xcomb_j, op=1, k=3, beta=1)
    f_obj = cbind(f_obj, p_success)
  }

  #Se ordenan y observan las mejores combinaciones
  psucess_final = t(f_obj)
  mejor_comb = cbind(comb, psucess_final)
  total = rowSums(mejor_comb[,-(length(mejor_comb))])
  mejor_comb = cbind(mejor_comb, total)
  mejor_comb = mejor_comb[order(-psucess_final, total),]
  return(mejor_comb)
}

#### Se le aplica a la funcion las salidas del paso INTERMEDIO.
sol_int0 = as.data.frame(sol_int$sol)
sol_int1 = sol_int0[colSums(sol_int0) != 0]  #vector con las vbles relevantes.
Xgen = X[,names(sol_int1)]                   #base con las vbles elegidas.
sol = fvr(Xgen)
sol

#### Se clusteriza con la mejor solucion.
sol_ind = sol[2,]                            #se queda con la mejor solucion.
sol_ind = sol_ind %>% select(-c("psucess_final", "total"))
dffin = dfbin[colnames(sol_ind[ ,which(sol_ind==1)])] #valores de la mejor sol.

#Se calcula k-medoids.
if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.logical(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.logical(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
    dffin$orig_capital_2=as.logical(dffin$orig_capital_2)
    }

dist_dffin = gower.dist(dffin)
dffin_pam = pam(dist_dffin, diss = TRUE, nc)
dffin_clus = dffin_pam$clustering; dffin_med = dffin_pam$id.med

if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.numeric(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.numeric(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
```

```
    dffin$orig_capital_2=as.numeric(dffin$orig_capital_2)
    }

#### Se clusteriza con la mejor solucion + reg no param para vbles redundantes y no inf.
sol_ind1 = sol_ind[,sol_ind==1]
sol_ind01 = as.data.frame(rbind(colnames(dfbin), rep(0, dim(dfbin)[2])))
colnames(sol_ind01) = sol_ind01[1,]
sol_ind01 = sol_ind01[-1, ]
sol_ind01[,colnames(sol_ind[ ,which(sol_ind==1)])] = 1
dfreg_clus = f_fit(sol_ind01, op=0, k=3, beta=1)

#### Se comparan los clusters.
ari_bin_vs_fin = ARI(dfbin_clus, dffin_clus)
ari_bin_vs_reg = ARI(dfbin_clus, dfreg_clus)
ari_bin_vs_fin
ari_bin_vs_reg

comp_clustKmed = as.data.frame(cbind(dfbin_clus, dffin_clus, dfreg_clus))
write_xlsx(comp_clustKmed,
    "C:/Users/gamar/Desktop/Maestria/Data_Tesis/comp_clustKmed_3p_FINAL.xlsx")
```

```
### Clusterizacion con Ward - 3 pasos
#### 1) Se corre un GA para achicar las alternativas de combinaciones de vbles.
#### Se contruye la funcion de FITNESS.
f_fit = function(cromo, op, k, beta) {
  if (sum(cromo==1)>0){

    #Se determinan los k vecinos mas cercanos de todos los registros
    #a partir de las vbles con 1 y obtengo sus indices
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)

    X1 = X[,which(cromo==1)]

    nn = gower.knn(x=X1, y=X1, k=k+1)   #k vecinos mas cercanos de c/obs.
    nn = nn[,-c(1)]          #se elimina el primer vecino por ser la misma obs.

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)

    #Se construyen los nuevos valores de las vbles que toman 0 y rearma la base.
    Xnew = X

    X0 = X[,which(cromo==0)]        #matriz con las vbles que toman 0.
    q_X0 = ncol(X0)                 #nro de vbles que toman 0.

    X0new = c()
    for (i in 1:q_X0){              #se toma una por una las vbles que toman 0.
      X0_ind = X0[,i]
      nn_valor = nn
      nn_valor[ ] = X0_ind[c(nn)]   #se le asigna a nn/indices el valor que le
                                    #corresponde de la vble tomada.
      if (all(X0_ind %in% 0:1)){    #se calcula el promedio o la moda segun
                                    #corresponda (cont/disc).
        nn_prom = apply(nn_valor, 1, moda)
      } else {
        nn_prom = apply(nn_valor, 1, mean)
      }
      X0new = cbind(X0new, nn_prom) #se agrupan las "nuevas" variables.
    }
```

```
    X0new = as.data.frame(X0new)
    colnames(X0new) = colnames(X0)  #se le ponen los nombres a las variables.

    Xnew[,which(cromo==0)] = X0new  #se reemplaza con las "nuevas" variables.

    #Se asignan las nuevas observaciones a los clusters existentes.
    X$orig_capital_0 = as.logical(X$orig_capital_0)
    X$orig_capital_1 = as.logical(X$orig_capital_1)
    X$orig_capital_2 = as.logical(X$orig_capital_2)
    Xnew$orig_capital_0 = as.logical(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.logical(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.logical(Xnew$orig_capital_2)

    #Se clusteriza por el metodo jerarquico ward.d2.
    dist_Xnew = gower.dist(Xnew)
    Xnew_ward2 = hclust(as.dist(dist_Xnew), method="ward.D2")
    dend_reg = as.dendrogram(Xnew_ward2)
    clustersnew = cutree(Xnew_ward2, k=nc)

    X$orig_capital_0 = as.numeric(X$orig_capital_0)
    X$orig_capital_1 = as.numeric(X$orig_capital_1)
    X$orig_capital_2 = as.numeric(X$orig_capital_2)
    Xnew$orig_capital_0 = as.numeric(Xnew$orig_capital_0)
    Xnew$orig_capital_1 = as.numeric(Xnew$orig_capital_1)
    Xnew$orig_capital_2 = as.numeric(Xnew$orig_capital_2)

    #Se comparan los dendogramas (original vs. regresion np)
    coef_cor = cor_cophenetic(dend_orig, dend_reg)
  } else {
    coef_cor=0
  }

  #Se establece un limite superior para el coeficiente.
  limpcoef = ifelse(coef_cor-beta>0,0,coef_cor)   #beta: tolerancia en la max.

  #Se devuelve el resultado.
  if (op==1) {          #op=1 devuelve p de suceso resultante,
                        #op=2 devuelve cluster de la reg.
    return(limpcoef)
  } else {
    return(clustersnew)
  }
}

#### Se corre FUNCION GA para aplicar el algoritmo genetico.
GA = ga(type = "binary",                #optimization data type
        fitness = function(cromo) f_fit(cromo, op=1, k=3, beta=0.99), #ff
        nBits = ncol(X),                #total number of variables
        #population = gabin_Population, #f que det la pobl. inicial (binary)
        #selection = lrSelection,       #f que det la nueva gener. (linear-rank)
        crossover = gabin_uCrossover,   #f para el crossover (uniform)
        #mutation = raMutation,         #f para la mutacion (uniform)
        popSize = 100,                  #the number of individuals
        pcrossover = 0.9,               #crossover rate prob (default 0.8)
        pmutation = 0.1,                #mutation rate prob (default 0.1)
        elitism = max(1, round(1000 * 0.05)), #the number of individuals that
                                              #pass to next iteration (default)
        monitor = plot,                 #plot the result at each iteration
        maxiter = 50,                   #total runs or generations
        run=15,                         #stopping criteria
        #maxFitness = -Inf,             #upper bound de la fitness
```

```
        names = colnames(X),
        keepBest = TRUE,                #keep the best solution at the end
        suggestions = diag(rep(1,ncol(X))), #matriz de soluciones para inicio
        seed = 36                       #for reproducibility purposes
)
summary(GA)
plot(GA)

#### Se analizan los resultados.
GA@summary
coef_cor_ga = GA@fitnessValue
sol_ga = as.data.frame(GA@solution)
g_sol_ga = apply(sol_ga,1,sum)
sol_ga_ind = sol_ga[which.min(g_sol_ga),]  #vec con la fila que nos interesa
g_sol_ga
min(g_sol_ga)
coef_cor_ga

#### Se contruye una f INTERMEDIA para acotar el conjunto de vles del GA.
fvr0 = function(alfa, gama, seed){
  i=1
  set.seed(seed)
  while(i<alfa){          #alfa: cantidad de iter para sacar vbles al azar.

    #Se elige al azar una vble para no considerar.
    sol_ga1 = sol_ga_ind[,which(sol_ga_ind==1)]
    vble_rand = sample(1:ncol(sol_ga1),1)
    sol_ga1_new = sol_ga1[-c(vble_rand)]

    #Se agregan las vbles descartadas por el GA y el azar pero con 0.
    sol_ga_new = as.data.frame(matrix(0,1,ncol(sol_ga)))
    colnames(sol_ga_new) = colnames(sol_ga)
    sol_ga_new[c(colnames(sol_ga1_new))] = sol_ga1_new

    #Se calcula el coef de correlacion para para la nueva alternativa.
    coef_cor_new = f_fit(sol_ga_new, op=1, k=3, beta=1)

    #Se compara el coef de correlacion obtenido con la vble eliminada con el
    #ultimo obtenido.
    #Si el nuevo coef correl. es mayor o igual me quedo con la nueva base.
    if (coef_cor_new-coef_cor_ga*gama >= 0){
      coef_cor_ga = coef_cor_new              #gama: tolerancia en la max.
      sol_ga_ind = sol_ga_new
    } else {
      coef_cor_ga = coef_cor_ga
      sol_ga_ind = sol_ga_ind
    }
    i=i+1
  }
  resultado = list("coef_cor" = coef_cor_ga, "q_vles"= sum(sol_ga_ind),
    "sol" = sol_ga_ind)
  return(resultado)
}

#### Se le aplica a la funcion las salidas de GA.
sol_int = fvr0(alfa=1000, gama=0.995, seed=41)
sol_int

#### 2) Se testean todas las combinaciones resultantes y se valida eleccion.
#### Se contruye una f que permita det que vbles explican la clusterizacion.
fvr = function(Xgen){                   #Xgen=base resultante del GA.
```

```
  #Se crea una grilla con todas las combin. de activacion de las vbles elegidas.
  comb = expand.grid(rep(list(0:1), ncol(Xgen)))
  comb = comb[-c(1),]                #se elimina la 1ra comb. por ser todo 0.
  q_comb = nrow(comb)                #nro de combinaciones finales.
  colnames(comb) = colnames(Xgen)

  Xcomb = as.data.frame(matrix(0,q_comb,ncol(X))) #se agregan las vbles
                                     #descartadas por el GA pero con 0.
  colnames(Xcomb) = colnames(X)
  Xcomb[c(colnames(comb))] = comb

  #Se calcula la probabilidad de exito para cada combinacion.
  f_obj = c()                         #funcion objetivo.
  for (j in 1:q_comb){
    Xcomb_j = Xcomb[j,]
    coef_cor = f_fit(Xcomb_j, op=1, k=3, beta=1)
    f_obj = cbind(f_obj, coef_cor)
  }

  #Se ordenan y observan las mejores combinaciones
  coefcor_final = t(f_obj)
  mejor_comb = cbind(comb, coefcor_final)
  total = rowSums(mejor_comb[,-(length(mejor_comb))])
  mejor_comb = cbind(mejor_comb, total)
  mejor_comb = mejor_comb[order(-coefcor_final, total),]
  return(mejor_comb)
}

#### Se le aplica a la funcion las salidas del paso INTERMEDIO.
sol_int0 = as.data.frame(sol_int$sol)
sol_int1 = sol_int0[colSums(sol_int0) != 0]   #vector con las vbles relevantes.
Xgen = X[,names(sol_int1)]                    #base con las vbles elegidas.
sol = fvr(Xgen)
sol

#### Se clusteriza con la mejor solucion.
sol_ind = sol[5,]                             #se queda con la mejor solucion.
sol_ind = sol_ind %>% select(-c("coefcor_final", "total"))
dffin = dfbin[colnames(sol_ind[ ,which(sol_ind==1)])] #valores de la mejor sol.

#Se calcula hclust jerarquico
if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.logical(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.logical(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
    dffin$orig_capital_2=as.logical(dffin$orig_capital_2)
    }

dist_dffin = gower.dist(dffin)
dffin_ward2 = hclust(as.dist(dist_dffin), method="ward.D2")
dffin_ward = cutree(dffin_ward2, k=nc)

if("orig_capital_0" %in% names(dffin)=="orig_capital_0"){
    dffin$orig_capital_0=as.numeric(dffin$orig_capital_0)
    }
if("orig_capital_1" %in% names(dffin)=="orig_capital_1"){
    dffin$orig_capital_1=as.numeric(dffin$orig_capital_1)
    }
if("orig_capital_2" %in% names(dffin)=="orig_capital_2"){
```

```
    dffin$orig_capital_2=as.numeric(dffin$orig_capital_2)
    }

#### Se clusteriza con la mejor solucion + reg no param para vbles redundantes y no inf.
sol_ind1 = sol_ind[,sol_ind==1]
sol_ind01 = as.data.frame(rbind(colnames(dfbin), rep(0, dim(dfbin)[2])))
colnames(sol_ind01) = sol_ind01[1,]
sol_ind01 = sol_ind01[-1, ]
sol_ind01[,colnames(sol_ind[ ,which(sol_ind==1)])] = 1
dfreg_ward = f_fit(sol_ind01, op=0, k=3, beta=1)

#### Se comparan los clusters.
ari_bin_vs_fin = ARI(dfbin_ward2T, dffin_ward)
ari_bin_vs_reg = ARI(dfbin_ward2T, dfreg_ward)
ari_bin_vs_fin
ari_bin_vs_reg

comp_clustCor = as.data.frame(cbind(dfbin_ward2T, dffin_ward, dfreg_ward))
write_xlsx(comp_clustCor,
    "C:/Users/gamar/Desktop/Maestria/Data_Tesis/comp_clustWard_3p_FINAL2.xlsx")
```

Bibliografía

[1] Alelyani, S., Tang, J., Liu, H. 2014. Data Clustering. Algorithms and Applications. Chapman and Hall/CRC, Capítulo 2: Feature Selection for Clustering: A Review, 35-41.

[2] Alvarez, A., Svarc, M. 2021. A variable selection procedure for depth measures. *AStA Adv Stat Anal*, 105, 247–271.

[3] Beerman, K., Prenio, J., Zamil, R. 2021. Suptech tools for prudential supervision and their use during the pandemic. *FSI Insights on policy implementation*, No 37.

[4] Boriah, S., Chandola, V., Kumar, V. 2008. Similarity Measures for Categorical Data: A Comparative Evaluation. *Proceedings of the 2008 SIAM International Conference on Data Mining (SDM)*, 243-254.

[5] Caruso, G., Sosa Escudero, W., Svarc, M. 2015. Deprivation and the Dimensionality of Welfare: A Variable-Selection Cluster-Analysis Approach. *Review of Income and Wealth*, Volume 61, Issue 4, 702-722.

[6] Choi, S., Cha, S., Tappert, C. 2009. A Survey of Binary Similarity and Distance Measures. *Systemics, Cybernetics and Informatics*, Volume 8, Number 1, Year 2010.

[7] Chormuge, S., Jena, S. 2017. Correlation based feature selection with clustering for high dimensional data. *Journal of Electrical Systems and Information Technology*, Volume 5, Issue 3, 542-549.

[8] Di Castri, S., Hohl, S., Kulenkampff, A., Prenio, J. 2019. The suptech generations. *FSI Insights on policy implementation*, No 19.

[9] Fop, M., Murphy, T.B. 2018. Variable selection methods for model-based clustering. *Statistics Surveys*, Vol. 12, 1-48.

[10] Fraiman, R., Justel, A., Svarc, M. 2008. Selection of Variables for Cluster Analysis and Classification Rules. *Journal of the American Statistical Association*, 103:483, 1294-1303.

[11] Financial Stability Board. 2020. The Use of Supervisory and Regulatory Technology by Authorities and Regulated Institutions. Market developments and financial stability implications.

[12] Ghattas, B., Michel, P., Boyer, L. 2019. Assessing variable importance in clustering: a new method based on unsupervised binary decision trees. *Computational Statistics*, 34 (1), 301-321.

[13] Goldberg, D. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

[14] Gower, J.C. 1971. A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, Vol. 27, No. 4, 857-871.

[15] Han, J., Kamber, M., Pei, J. 2012. Data mining: concepts and techniques. Elseiver, Tercera edición.

[16] Hubert, L., Arabie, P. 1985. Comparing partitions. *Journal of Classification*, 193–218.

[17] Hastie, T., Tibshirani, R., Friedman, J. 2009. The Elements of Statistical Learning, Data Mining, Inference, and Prediction. Springer, Segunda Edición.

[18] James, G., Witten, D., Hastie, T., Tibshirani, R. 2013. An Introduction to Statistical Learning, with Applications in R. Springer.

[19] Kaufman L., Rousseeuw, P. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley Series in Probability and Statistics, Capítulo 2: Partitioning Around Medoids (Program PAM), 68-125.

[20] Murtagh, F., Legendre, P. 2014. Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?. *Journal of Classification*, 31, 274-295.

[21] OECD. 2021. Artificial Intelligence, Machine Learning and Big Data in Finance: Opportunities, Challenges, and Implications for Policy Makers.

[22] Scrucca, L. 2013. GA: A Package for Genetic Algorithms in R. *Journal of Statistical Software*, 53/4, 1-37.

[23] Sivanandam, S.N., Deppa, S.N. 2008. Introduction to Genetic Algorithms. Springer.

[24] Shirkhorshidi, A.S., Aghabozorgi, S., Wah, T.Y. 2015. A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data. *PLoS ONE*, 10(12): e0144059.

[25] Sokal, R., Rohlf, J. 1962. The Comparison of Dendrograms by Objective Methods. *Taxon*, Vol. 11, No. 2, 33-40.

[26] Sosa Escudero, W., Caruso, G., Svarc, M. 2013. *Poverty and the Dimensionality of Welfare*, Poverty and Social Exclusion: New Methods of Analysis. Routledge, Primera Edición, Capítulo 3, 38-53.

[27] Steinley, D., Brusco, M.J. 2008. Selection of Variables in Cluster Analysis: An Empirical Comparison of Eight Procedures. *Psychometrika*, 73, 125–144.

[28] Storlie, C., Myers, S., Katusic, S., Weaver, A., Voigt, R., Croarkin, P., Stoeckel, R., Port, J. 2018. Clustering and variable selection in the presence of mixed variable types and missing data. *Statistic in Medice*, Vol. 37, Issue 19, No. 2, 2884-2899.

[29] Ward, J.H. 1963. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, Vol. 58, Issue 301, 236-244.

[30] World Bank Group. 2018. From Spreadsheets to Suptech. Technology Solutions for Market Conduct Supervision.

[31] Yuan, S., De Roover, K., Van Deun, K. 2022. Simultaneous clustering and variable selection: A novel algorithm and model selection procedure. *Behavior Research Methods*.