“Predicción de fuga de clientes en una empresa de telecomunicaciones, mediante *machine learning*, utilizando variables financieras y operacionales”

Justo Matías Solís Zlatar

2019

UNIVERSIDAD ADOLFO IBÁÑEZ
FACULTAD DE INGENIERÍA Y CIENCIAS

“Predicción de fuga de clientes en una empresa de telecomunicaciones, mediante machine learning, utilizando variables financieras y operacionales”

Justo Matías Solís Zlatar

Profesor guía: Gonzalo Ruz, Ph.D

TESIS PARA OPTAR AL GRADO DE MAGISTER EN CIENCIAS DE LA INGENIERÍA – MENCIÓN EN TECNOLOGÍAS DE LA INFORMACIÓN

2019

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

GLOSARIO Y ABREVIATURAS

DX: Corresponde a la abreviatura de Desconexión (interrupción del servicio a la vivienda del cliente).

Desconexión Mora: Corresponde a la interrupción del servicio, producto de que el cliente no pagó su deuda, en un período de tiempo predeterminado.

Desconexión Voluntaria: Corresponde a la interrupción del servicio, producto de que el cliente manifiesta su intención de retirarse de la compañía, por diversos motivos (ofertas de otra empresa, mala calidad del servicio, entre otros).

I. RESUMEN DEL PROYECTO

La empresa con la que se trabajará en la presente tesis es líder en la industria de telecomunicaciones en Chile, entregando servicios de Internet Banda Ancha, Televisión por Cable y Telefonía Fija Ilimitada. Estos servicios son entregados a los distintos clientes del país, ya sea de forma individual, o como paquetes. La empresa posee alrededor de 2.9 millones de servicios activos, lo que es equivalente a 1.3 millones de clientes, y presentan alrededor de 60.000 ventas mensuales en servicios Fijos. Sin embargo, la compañía presenta una cantidad similar en Desconexiones, tanto mora como voluntaria, lo que trae como consecuencia que la tasa de *churn* de los clientes se encuentre entre 2% y 6%, especialmente en clientes que tienen una permanencia menor a 3 meses. Debido a esto, se procederá a crear un modelo de predicción de fuga de clientes, utilizando variables financieras y operacionales, con el objetivo de disminuir las tasas de fuga mora y voluntaria de la empresa.

Para el desarrollo de este trabajo, se utilizará la metodología CRISP – DM para segmentar a los clientes, y para la elaboración de un modelo de predicción, debido a que se incluye la comprensión del negocio, lo que permitirá que el desarrollo del proyecto se enfoque en las necesidades de la compañía. Junto con lo anterior, se aplicarán diversas técnicas de Minería de Datos y *Machine Learning* para obtener la predicción de fuga de clientes.

Finalmente, se procederá con las discusiones relacionadas a los resultados obtenidos en el desarrollo de la tesis, y su valor agregado, a diferencia de lo realizado en trabajos anteriores relacionados a predecir los clientes fugados.

II. DESCRIPCIÓN DEL PROBLEMA U OPORTUNIDAD

La empresa de telecomunicaciones posee cerca de 2.9 millones de servicios Fijos activos en Chile, entre Arica y Coyhaique, equivalente a 1.3 millones de clientes activos en Chile. Junto con lo anterior, presenta alrededor de 60.000 ventas mensuales en este tipo de servicios.

Ahora, si bien la compañía presenta un gran volumen de ventas mensual, la cantidad de desconexiones que se producen en el mismo período de tiempo es de, aproximadamente, 30.000 servicios. A continuación, se muestra la distribución de las desconexiones moras finalizadas.

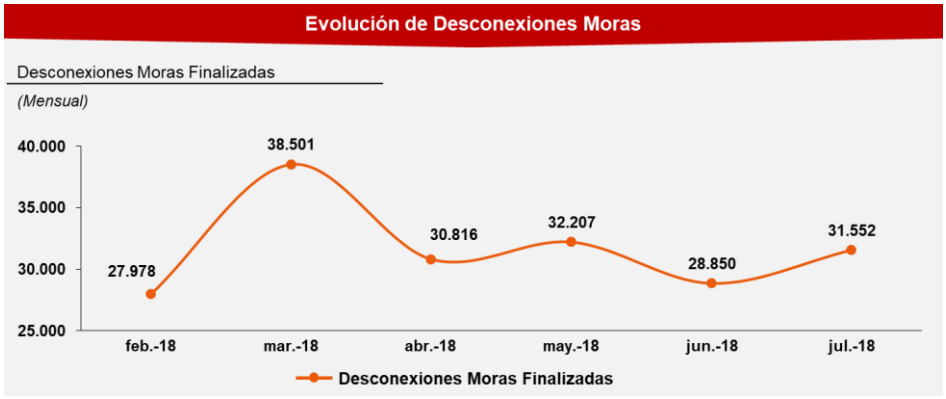


Figura 2.1 DX Moras Finalizadas en los últimos seis meses (Fuente: Empresa de telecomunicaciones).

En el caso de las desconexiones voluntarias finalizadas, la compañía presenta una cantidad de 35.000 desconexiones mensuales, aproximadamente, siendo algunas veces mayor que las mismas desconexiones mora. Las desconexiones voluntarias se pueden ver en el siguiente gráfico.

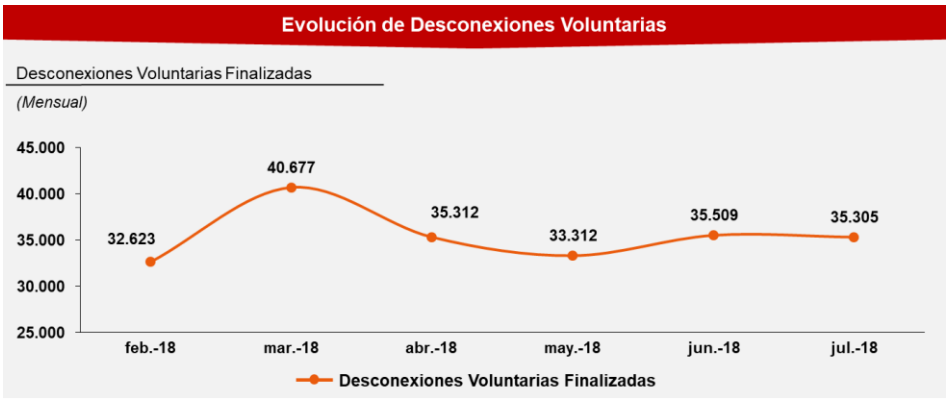


Figura 2.2 DX Voluntarias finalizadas en los últimos seis meses (Fuente: Empresa de telecomunicaciones).

Además, la tasa de *churn* de los clientes ha presentado una tendencia al alza, especialmente en aquellos que tienen una permanencia entre 0 y 3 meses en la compañía. Esto se puede ver reflejado en la siguiente tabla, en relación con la tasa de *churn* mora:

	CHURN MORA					
	feb-18	mar-18	abr-18	may-18	jun-18	jul-18
0 - 3 Meses	5,12%	6,59%	6,06%	6,27%	5,32%	5,52%
3 - 6 Meses	1,57%	2,44%	1,77%	1,86%	1,80%	2,46%
6 Meses a 1 año	0,67%	1,26%	0,87%	0,85%	0,84%	0,92%
1 a 1,5 años	0,65%	0,94%	0,72%	0,77%	0,72%	0,77%
1,5 a 2 años	0,49%	0,74%	0,45%	0,48%	0,45%	0,42%
2 a 2,5 años	0,38%	0,58%	0,40%	0,48%	0,40%	0,41%
2,5 a 3 años	0,31%	0,43%	0,28%	0,27%	0,25%	0,25%
Más de 3 años	0,17%	0,17%	0,13%	0,14%	0,12%	0,12%

Tabla 2.1. Tasa de *churn* mora de VTR, por permanencia (Fuente: Empresa de telecomunicaciones).

En el caso de la tasa de *churn* voluntaria, se puede ver nuevamente una tendencia al alza, especialmente en clientes con permanencia de 0 a 3 meses, y también de 2 a 2,5 años (en esta última categoría, alcanza un porcentaje cercano al 1%). Lo anterior se puede ver en la siguiente tabla:

	CHURN VOLUNTARIA					
	feb-18	mar-18	abr-18	may-18	jun-18	jul-18
0 - 3 Meses	1,58%	2,43%	2,18%	2,00%	1,88%	1,88%
3 - 6 Meses	1,02%	1,37%	1,40%	1,30%	1,29%	1,62%
6 Meses a 1 año	2,72%	1,34%	1,02%	0,93%	1,03%	1,05%
1 a 1,5 años	0,89%	1,21%	1,14%	1,08%	1,16%	1,18%
1,5 a 2 años	0,95%	1,07%	0,85%	0,76%	0,84%	0,78%
2 a 2,5 años	0,79%	1,06%	0,88%	0,87%	0,95%	0,90%
2,5 a 3 años	0,86%	0,96%	0,75%	0,68%	0,77%	0,70%
Más de 3 años	0,19%	0,56%	0,51%	0,47%	0,52%	0,49%

Tabla 2.2. Tasa de *churn* voluntario, por permanencia (Fuente: Empresa de telecomunicaciones).

Luego, el problema se enfoca en poder tomar acciones proactivas a los clientes con mayor probabilidad de fuga de la compañía, dado que actualmente se impulsan medidas cuando el cliente tiene una deuda con la compañía, o cuando manifiesta expresamente su intención de desconectar sus servicios, ya sea en las distintas sucursales de la empresa, o por medio del Call Center.

III. INTRODUCCIÓN

En las diversas industrias, en especial en la de telecomunicaciones, existen varios trabajos relacionados con la predicción y gestión de fugas de clientes (o *customer churn*), ya sea de forma voluntaria o involuntaria, cuyo objetivo es poder predecir los potenciales clientes que pueden abandonar la compañía, ya sea para irse a la competencia, o porque no han pagado su deuda.

Para poder realizar esta predicción, se han utilizado diversas variables para entrenar modelos de árboles de decisión, regresión logística, entre otros, y así poder identificar los clientes que presentan una mayor probabilidad de fuga. Un ejemplo de esto consiste en el diseño de un modelo de predicción, basado en árboles de decisión, utilizando información de una empresa de telecomunicaciones para servicios post pago de televisión digital, con el objetivo de poder predecir los potenciales clientes que se podrían fugar de forma voluntaria, y actuar de forma proactiva en su retención (Contreras et al., 2017). Otro caso de uso de árboles de decisión se encuentra en Keramati et al. (2016), donde se utilizaron datos de usuarios de servicios electrónicos bancarios, para modelar un árbol de decisión con RapidMiner y predecir la fuga de clientes.

Otra técnica de *machine learning* comúnmente utilizada son las redes neuronales. Un ejemplo de esto es el modelamiento de una red neuronal perceptrón multicapa (MLP), con una base de datos de clientes pertenecientes a una empresa de telecomunicaciones en Malasia, que fue entrenado por varios algoritmos de aprendizaje, como *Levenberg Marquardt back – propagation*, *Conjugate Gradient backpropagation with Fletcher – Reeves Updates*, entre otros, con el objetivo de obtener una predicción con mejor porcentaje de rendimiento, medido por la precisión de la predicción (Awang et al., 2018).

Mishra et al. (2018) utilizó una base de clientes pertenecientes a una empresa de telecomunicaciones para poder predecir la fuga de clientes. Para esto, procedió a reducir el desbalance en los datos, utilizando *Synthetic Minority Over – Sampling Technique* (SMOTE), junto con otras técnicas de reducción de atributos, como extracción de atributos correlacionados, *gain ratio*, *information gain*, y el método de evaluación de

atributos OneR. Luego, se entrenaron diversos modelos como *Classification and Regression Trees* (CART), *Bagged CART* y *Partial Decision Trees* (PART), para analizar su rendimiento, en base al área bajo la curva (AUC), *sensitivity* y *specificity*. Otro ejemplo del manejo de desbalance en los datos es el uso de un método combinado entre SMOTE y *Random Under – Sampling* (RUS) para obtener mejores resultados en la predicción de fuga de clientes, en una base de datos perteneciente a una empresa de Internet Banda Ancha de Indonesia (Hartati et al., 2018).

Otro método utilizado es *Particle Swarm Optimization* (PSO), que fue utilizado por Vijaya et al. (2017), quienes propusieron tres variantes distintas de PSO para realizar la predicción de fuga de clientes, las cuales eran: PSO incorporado con selección de atributos, PSO fusionado con *Simulated Annealing*, y PSO como una combinación entre selección de atributos y *Simulated Annealing*. Los resultados obtenidos por estos métodos fueron comparados con árboles de decisión, Naive – Bayes, K – vecinos más cercanos (k – NN), *Support Vector Machines* (SVM), *random forests*, entre otros. Otro ejemplo se presenta en Huang et al. (2018), quien propone un algoritmo híbrido entre PSO y *BP neural network*, con el objetivo de poder establecer el riesgo de fuga que poseen los clientes pertenecientes a una empresa de producción y venta de cigarrillos.

Además, se han utilizado clasificadores difusos (o *fuzzy classifiers*) para poder predecir la fuga de clientes, cuyos resultados han sido comparados con los obtenidos por redes neuronales, regresión lineal, C4.5, SVM, AdaBoost, *Gradient Boosting*, y *random forests*, demostrando que se pueden obtener buenos resultados para poder obtener un conjunto preciso de *churners* (Azeem et al., 2017).

Stripling et al. (2018) propone el uso de un clasificador que permite maximizar la medida de rentabilidad máxima esperada para fuga de clientes (EMPC) utilizando un algoritmo genético, llamado *ProfLogit*, con el objetivo de predecir la fuga de clientes, utilizando bases de datos pertenecientes a clientes de empresas de proveedores de servicios telefónicos.

Junto con lo anterior, se han propuesto algoritmos híbridos para predecir la fuga de clientes, obteniendo un mejor resultado, en base a su rendimiento y comprensibilidad.

De Caigny et al. (2018) utilizó datos de clientes pertenecientes a empresas de diversas industrias, y por medio de un algoritmo híbrido, llamado *logit leaf model* (LLM), realizó una predicción de fuga de clientes. Luego, esos resultados fueron comparados con los obtenidos por árboles de decisión, regresión logística, *random forests* y *logistic model trees*, concluyendo que el rendimiento predictivo obtenido por LLM fue significativamente mejor que los resultados obtenidos por los otros modelos. Otro caso se puede ver en el trabajo presentado por Ahmed et al. (2018), quien combinó múltiples *ensemble models*, creando dos *nested learners*, llamados *Boosted – Stacked learners* y *Bagged – Stacked learners*, para predecir la fuga de clientes en una base de datos balanceada, de un operador de telecomunicaciones de Asia del Sur.

Otro método híbrido propuesto para la predicción de fuga de clientes se presenta en Idris et al. (2017), donde utilizó programación genética (GP) con *AdaBoost*, para identificar factores que permitan representar el comportamiento de fuga, en una base de clientes de una empresa de telecomunicaciones. Junto con lo anterior, también se creó un algoritmo híbrido, que consiste en utilizar los métodos de regresión logística y Naive Bayes, para poder predecir la fuga de clientes, obteniendo un resultado con mayor tasa de aciertos que lo obtenido por cada uno de los modelos por separado (Günay et al., 2018).

Otro ejemplo de algoritmo híbrido se encuentra en Dulhare et al. (2018), quien propone un método compuesto por *Axiomatic Fuzzy Set* (AFS) y *parallel DBSCAN clustering*, obteniendo un mejor resultado que lo obtenido por *K – means*.

Además, se han diseñado otros métodos para predecir la fuga de clientes, como por ejemplo, usar los registros de llamadas que realizaron clientes de un proveedor de telecomunicaciones en un período de seis meses, y así, poder construir redes de llamadas semanales, con las que se obtuvieron series de tiempo para cada cliente. Luego, las series de tiempo fueron clasificadas mediante el método de *similarity forests* para poder predecir la fuga de clientes (Óskarsdóttir et al., 2018).

Amin et al. (2018) propone un enfoque de predicción de fuga de clientes, basado en la estimación de la certeza del clasificador, utilizando el factor distancia. Aquí, se tomaron

diversas bases de datos de la industria de telecomunicaciones, disponibles públicamente, y se dividieron en distintas zonas, basadas en el factor distancia. Luego, estas zonas se agruparon en dos categorías, según la certeza de los clasificadores utilizados (*accuracy*, *F – measure* y precisión), para predecir los clientes que tengan un comportamiento de fuga, de los que no lo tienen.

Sivasankar et al. (2018) propone un sistema de predicción de *churn* de clientes, que consiste en clasificar los datos recolectados por *Tera Data Center* de la Universidad de Duke, en Estados Unidos, mediante *hybrid probabilistic possibilistic fuzzy C – means clustering* (PPFCM), y luego realizar la predicción, utilizando redes neuronales artificiales (ANN).

Otro método propuesto para predecir la fuga de clientes es un algoritmo llamado *Particle Classification Optimization – based BP network for telecommunication customer churn prediction* (PBCCP), que se encarga de ejecutar, de forma iterativa, dos métodos: PCO y *particle fitness calculation* (PFC). Aquí, se aplicó este algoritmo en una base de registros de llamadas telefónicas de una empresa de telecomunicaciones de China, con el objetivo de poder predecir la fuga de clientes (Yu et al., 2018).

Otra forma de identificar los clientes fugados es por medio de *flat – rate pricing*. Moser et al. (2018) utilizaron una base de 21.490 clientes pertenecientes a un proveedor de servicios de Internet Premium, y por medio de un análisis de supervivencia de dos años, detectaron que un sesgo en la tarifa plana produce la fuga de los clientes de la compañía, debido a que lo atribuyen de forma más externa, a diferencia de los clientes que no pertenecen al segmento Premium.

También, se han utilizado técnicas metaheurísticas para realizar predicción de fuga de clientes. Un ejemplo es el uso de una forma híbrida del algoritmo *Firefly* como clasificador, en una base de datos de una empresa de telecomunicaciones, donde se reemplazó el bloque de comparación (o *comparison block*) por el algoritmo *Simulated Annealing* (Ammar et al., 2017).

Junto con las técnicas anteriormente mencionadas, se han propuesto indicadores que permiten identificar a posibles clientes que tienen intenciones de fuga. Un ejemplo de

esto es un índice llamado *diversion ratio*, que identifica los sesgos potenciales que causan la fuga de clientes, como el costo marginal de la firma, el cambio de tamaño de la demanda (producto de las circunstancias que van cambiando en el tiempo), y el aprendizaje sobre los atributos de los productos (Chen et al., 2016).

Si bien la fuga de clientes se aplica principalmente a industrias donde exista una interacción con el cliente de forma presencial, también se aplica a industrias o mercados de tipo online, donde la comunicación con el cliente no es presencial, o simplemente no existe. Un ejemplo es la predicción de fuga de clientes en un sitio Web de citas online de Taiwan, donde, para realizar la predicción, se utilizó un gráfico, llamado *Cumulative Sum control chart* (o *CUSUM control chart*), y recolectando solamente el tiempo entre llegadas (diferencia de tiempo entre dos eventos consecutivos) y *recency* (diferencia de tiempo entre el último cierre de sesión, y el inicio del último evento), se logró monitorear a cada cliente de forma individual, y así, predecir si un cliente se fugará o no (Chen, 2016).

También se puede encontrar ejemplos de predicción de fuga de clientes en los usuarios de juegos online. Un ejemplo se encuentra en la predicción de fuga de usuarios de un juego móvil, donde se introdujo un sistema de dos partes: predecir los clientes que se van a fugar (mediante el uso de técnicas tradicionales de *machine learning*); y enviar notificaciones personalizadas a esos clientes, como una medida para prevenir que se retiren del juego (Milošević et al., 2017).

Otra industria en donde se ha realizado modelos de predicción de fuga de clientes es el *e – commerce*. Aquí, se implementó un modelo de SVM, basados en la técnica de selección de parámetros AUC (llamada SVMauc). Luego, la capacidad de predicción del modelo se comparó con las técnicas de regresión logística, redes neuronales y SVM tradicional (Gordini et al., 2017). Otro caso en la misma industria consiste en el uso de las técnicas *K – means* y *Length – Recency – Frequency – Monetary* (o modelo LRFM) para segmentar los clientes, y luego clasificarlos entre clientes no fugados, parcialmente fugados, y fugados, usando tres técnicas de clasificación: árbol de decisión simple, ANN y *decision tree ensemble* (Rachid et al., 2018).

Otra industria donde se han desarrollado trabajos sobre la predicción de fuga de clientes es el *retail*. Patil et al. (2017) utilizó una base de datos de transacciones, pertenecientes a una tienda de regalos online del Reino Unido, para poder predecir los clientes fugados, utilizando *random forests*, SVM y *extreme gradient boosting*.

Rajamohamed et al. (2017) utilizó una base de usuarios de tarjetas de crédito, con el objetivo de poder predecir los posibles clientes fugados, y así, poder retenerlos. Para este caso, se utilizó el algoritmo *K – means* para segmentar los clientes, y luego se usaron diversas técnicas para realizar la predicción, como SVM, *random forests*, árboles de decisión, k – NN, y Naive Bayes. Finalmente, para poder escoger el mejor modelo, se utilizaron indicadores de precisión, *sensitivity*, *specificity*, *accuracy* y error de clasificación.

Otro caso de predicción de fuga de clientes en la industria bancaria es presentado por Mohanty et al. (2018), quien propone el uso de *Extreme Learning Machine* (ELM) para poder predecir los clientes fugados.

Culbert et al. (2018) propone un nuevo enfoque para la predicción de fuga de clientes, aplicado a la industria de fondos de pensiones en Australia (también llamado *Superannuation*). En este caso, se utilizó el algoritmo *extreme gradient boosting*, junto con *contrast sequential pattern mining*, con el objetivo de poder extraer patrones que precedan un evento de fuga.

Otras investigaciones que se han realizado sobre la fuga de clientes se centran en la forma en que se maneja la preparación de los datos para realizar el modelo de predicción. Por ejemplo, Coussement et al. (2017) utilizaron ocho técnicas de *data mining* para modelar una base de datos de una empresa de telecomunicaciones, y luego, comparó esos resultados con los obtenidos por un *logit model* optimizado, bajo el contexto de modelamiento de predicción de fuga de clientes.

A continuación, se muestra una tabla resumen de los trabajos anteriores.

Año	Título	Revistas	Autores	Técnicas
2018	*Time series for early churn detection: Using similarity*	*Expert Systems With Applications*	Óskarsdóttir, M.; Van Calster, T.; Baesens, B.;	*Similarity forests*

	based classification for dynamic networks		Lemahieu, W. y J. Vanthienen	
2018	*Customer churn prediction in telecommunication industry using data certainty*	*Journal of Business Research*	Amin, A.; Al – Obeidat, F.; Shah, B.; Adnan, A.; Loo, J. y S. Anwar	*Distance factor*
2017	*Design of a Predictive Customer Leakage Model using Decision Trees*	Revista Ingeniería Industrial – Año 16	Contreras, E.; Ferreira, F. y M. Valle	Árbol de decisión
2018	*A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees*	*European Journal of Operational Research*	De Caigny, A.; Coussement, K. y K. de Bock	LLM, *decision trees*, regr. logística, *random forests* y *LM Trees*
2016	*The gamma CUSUM chart method for online customer churn prediction*	*Electronic Commerce Research and Applications*	Chen, S.	*CUSUM control chart*
2017	*Early churn prediction with personalized targeting in mobile social games*	*Expert Systems with Applications*	Milošević, M.; Živić, N. e I. Andjelković	*Machine learning*
2017	*A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry*	*Decision Support Systems*	Coussement, K.; Lessmann, S. y G. Verstraeten	*Logit model*, *Bayesian network*, árbol de decisión J4.8, *MPNN*, *Naive Bayes*, *random forest*, *stochastic gradient boosting*
2017	*Customers churn prediction and marketing retention strategies. An application of SVM based on the AUC parameter – selection technique in B2B e – commerce industry*	*Industrial Marketing Management*	Gordini, N. y V. Veglio	SVM, SVMauc, regresión logística y redes neuronales.
2017	*Churn prediction on churn telecom data using hybrid firefly based classification*	*Egyptian Informatics Journal*	Ammar, A.; Ahmed, Q y D. Maheswari	Algoritmo *Firefly*, con *Simmulated Annealing*

2018	*Exploring nested ensemble learners using overproduction and choose approach for churn prediction in telecom industry*	*Neural Computing and Applications*	Ahmed, M.; Afzal, H.; Siddiqi, I.; Amjad, M. y K. Khurshid	*Boosted – Stacked learners* y *Bagged – Stacked learners*
2016	*Developing a prediction model for customer churn from electronic banking services using data mining*	*Financial Innovation*	Keramati, A.; Ghaneei, H. y S. Mirmohammadi	Árbol de decisión
2018	*Particle classification optimization – based BP network for telecom customer churn prediction*	*Neural Computing and Applications*	Yu, R.; An, X.; Jin, B.; Shi, J.; Move, O. y Y. Liu	PBCCP, PCO y PFC
2018	*Hybrid PPFCM – ANN model: an efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network*	*Neural Computing and Applications*	Sivasankar, E. y J.Vijaya	PPFCM y ANN
2017	*An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing*	*Cluster Computing*	Vijaya, J. y E. Sivasankar	PSO, árboles de decisión, Naive – Bayes, k – NN, SVM, *random forests*
2017	*Intelligent churn prediction for telecom using GP – AdaBoost learning and PSO undersampling*	*Cluster Computing*	Idris, A.; Iftikhar, A. y Z. Rehman	GP con AdaBoost
2017	*Improved credit card churn prediction based on rough clustering and supervised learning techniques*	*Cluster Computing*	Rajamohamed, R. y J. Manokaran	*K – means*, SVM, *random forests*, árboles de decisión, k – NN, y Naive Bayes

2017	*A churn prediction model for prepaid customers in telecom using fuzzy classifiers*	*Telecommunication Systems*	Azeem, M.; Usman, M. y A. Fong	*Fuzzy classifiers*, redes neuronales, regresión lineal, C4.5, SVM, AdaBoost, *Gradient Boosting*, y *random forests*
2018	*The Effect of a Service Provider's Competitive Market Position on Churn Among Flat – Rate Customers*	*Journal of Service Research*	Moser, S.; Schumann, J.H.; von Wangenheim, F.; Uhrich, F. y F. Frank	*Flat – rate pricing*
2018	*Clustering prediction techniques in defining and predicting customers defection: The case of e – commerce context*	*International Journal of Electrical and Computer Engineering*	Rachid, A.D.; Abdellah, A.; Belaid, B. y L. Rachid	*K – means*, modelo LRFM, árbol de decisión, ANN y *decision tree ensemble*
2018	*Predictive churn analysis with machine learning methods (Makine öğrenmesi yöntemleri ile kayip müşteri analizi)*	*26th IEEE Signal Processing and Communications Applications Conference*	Günay, M. y T. Ensari	Algoritmo híbrido entre regresión logística y Naive Bayes
2018	*An efficient hybrid clustering to predict the risk of customer churn*	*Proceedings of the 2nd International Conference on Inventive Systems and Control*	Dulhare, U.N. e I. Ghori	*K – means*, y un algoritmo híbrido entre AFS y *parallel DBSCAN clustering*
2018	*Customer churn prediction for retail business*	*2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*	Patil, A.P.; Deepshika, M.P.; Mittal, S.; Shetty, S.; Hiremath, S.S. y Y.E. Patil	*Random forests*, SVM y *extreme gradient boosting*
2018	*Churn prediction in telecommunication using machine learning*	*2017 International Conference on Energy, Communication, Data*	Mishra, K. y R. Rani	SMOTE, *gain ratio*, *information gain*, OneR, CART,

		Analytics and Soft Computing		*Bagged CART* y PART.
2018	*Handling imbalance data in churn prediction using combined SMOTE and RUS with bagging method*	*Journal of Physics: Conference Series*	Hartati, E.P.; Adiwijaya, y M.A. Bijaksana	SMOTE y RUS
2018	*Performance comparison of neural network training algorithms for modeling customer churn prediction*	*International Journal of Engineering and Technology*	Awang, M.K.; Ismail, M.R.; Makhtar, M.; Nordin A. Rahman, M., y A.R. Mamat	MLP, *Levenberg Marquardt back – propagation*, *BFGS Quasi – Newton backpropagation*, *Conjugate Gradient backpropagation*
2016	*Churn Versus Diversion in Antitrust: An Illustrative Model*	*Economica*	Chen, Y. y M. Schwartz	*Diversion ratio*
2018	*Profit maximizing logistic model for customer churn prediction using genetic algorithms*	*Swarm and Evolutionary Computation*	Stripling, E.; van den Broucke, S.; Antonio, K.; Baesens, B. y M. Snoeck	EMPC y ProfLogit
2018	*Customer Churn Prediction in Superannuation: A sequential pattern mining approach*	*Australasian Database Conference 2018: Databases Theory and Applications*	Culbert, B.; Fu, B.; Brownlow, J.; Chu, C.; Meng, Q. y G. Xu	*Extreme gradient boosting*, *contrast sequential pattern mining*
2018	*Churn and Non – churn of Customers in Banking Sector using Extreme Learning Machine*	*Proceedings of the Second International Conference on Computational Intelligence and Informatics*	Mohanty, R. y C.N.R. Sree	ELM
2018	*Application of Improved PSO - BP Neural Network in Customer Churn Warning*	*Procedia Computer Science*	Huang, J. y L. He	PSO y *BP neural network*

Tabla 3.1 Cuadro resumen de artículos sobre predicción de fuga de clientes (Fuente: Elaboración propia).

IV. HIPÓTESIS, OBJETIVO GENERAL Y OBJETIVOS ESPECÍFICOS DEL TRABAJO

En el caso de los clientes que poseen deuda, cuya antigüedad es mayor a lo definido en los flujos de cobranza, la compañía de telecomunicaciones posee una política de retención de clientes, que consiste en comunicarse con aquellos que están próximos a irse de la empresa, para que puedan pagar su deuda. En el caso de los clientes que se quieran ir de forma voluntaria, se realizan acciones de retención cuando el cliente va a alguna sucursal de la compañía, o cuando llama al Call Center.

Debido a lo anterior, se requiere un modelo de predicción de fuga de clientes que permita poder predecir, con un alto nivel de precisión, aquellos usuarios que tengan mayor probabilidad de irse de la compañía. Para poder desarrollar el modelo, se necesitan diversas variables que permitan abordar el problema, lo que posibilita plantear la siguiente hipótesis:

"La efectividad en la predicción de fuga de clientes mejora al considerar variables financieras y operacionales de manera conjunta, a diferencia de usar cada una por separado"

Producto de que ambas políticas de retención son reactivas en vez de proactivas, y la empresa no tiene un modelo que permita predecir los clientes que podrían fugarse de la compañía, es que se define como objetivo general lo siguiente: "implementar un modelo de predicción de fuga de clientes, a nivel nacional, con el fin de disminuir la tasa de fuga mora y voluntaria de la compañía de telecomunicaciones".

IV.1. Objetivos específicos

- Segmentar a los clientes de la compañía en *clusters*, con variables que permitan describirlos adecuadamente, en base a los servicios contratados.
- Obtener un modelo de predicción para los *clusters*, utilizando variables financieras y operacionales de los clientes, y comparando diversas técnicas de *machine learning*.
- Identificar, para cada modelo, las variables más importantes que se utilizaron en su creación, para poder crear políticas proactivas de retención de clientes.
- Crear e implementar un flujo de trabajo de los modelos de predicción para los *clusters*.

V. METODOLOGÍA

En este capítulo, se describirán las distintas metodologías que se utilizan para el desarrollo de proyectos de *Data Mining*, y se escogerá el procedimiento necesario para el desarrollo de la tesis.

V.1. Metodologías de Minería de Datos

A continuación, se procederá a describir las distintas metodologías utilizadas para desarrollar los proyectos de minería de datos, se compararán y finalmente, se escogerá la mejor para la ejecución de la tesis.

V.1.1. Metodología SEMMA

Esta metodología fue desarrollada por el Instituto SAS, que consiste en una guía para poder implementar modelos y aplicaciones de Minería de Datos, para solucionar problemas de diversas industrias. Su nombre es un acrónimo que corresponde a las fases que involucra esta metodología. Estos pasos son los siguientes:

a) *Sample*: Consiste en la selección de los datos y variables para modelar y particionar la base.
b) *Explore*: Consiste en entender los datos para descubrir relaciones entre las variables, y datos atípicos, mediante la visualización de datos.
c) *Modify*: Consiste en el uso de métodos para seleccionar, crear y transformar las variables para el modelamiento de datos.
d) *Model*: Consiste en usar varios métodos de modelamiento, usando las diversas variables de la base de datos, con el objetivo de crear modelos que permitan obtener el resultado deseado.
e) *Assess*: Consiste en evaluar los resultados de los modelos creados, para escoger el mejor, siguiendo ciertos parámetros definidos previamente.

Lo explicado anteriormente se puede representar en la siguiente imagen:

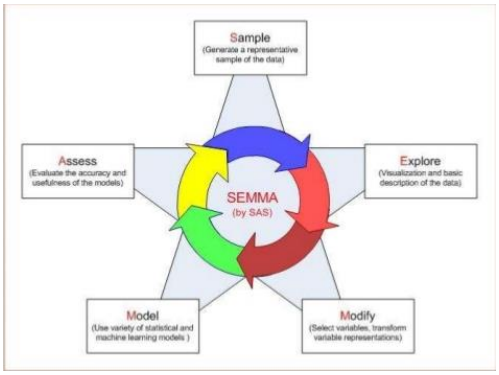


Figura 5.1 Metodología SEMMA (Fuente: https://www.slideshare.net/MohanBavirisetty/advanced-analytics-frameworks-platforms-and-metholodologies-v-10)

V.1.2. KDD

Knowledge Discovery in Databases (más conocido como KDD) es una metodología que se encarga de descubrir conocimientos e información de los datos que se encuentran contenidos en una base de datos (ya sea Excel, SQL, XML, etc.). Este proceso se realiza de forma iterativa, mediante el análisis de grandes volúmenes de datos, para extraer información, y obtener conclusiones. Este proceso se puede resumir en la siguiente imagen:

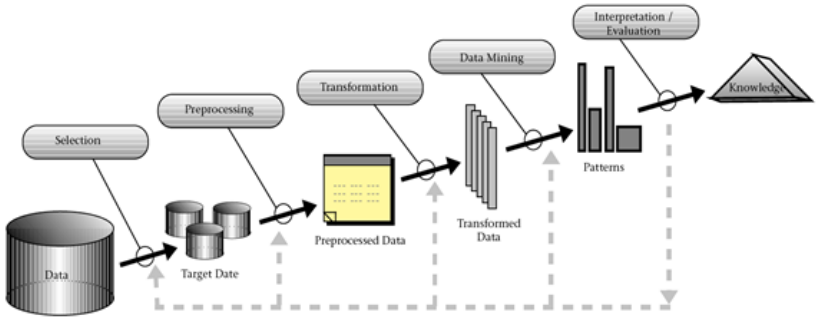


Figura 5.2 Metodología KDD (Fuente: https://www.researchgate.net/figure/Figura-2-El-proceso-KDD-CRISP-DM-por-su-parte-corresponde-a-una-metodologia-que_fig1_268687479)

Las etapas que se muestran en la imagen anterior consisten en lo siguiente:

a) Selección de datos: Consiste en seleccionar los datos a utilizar para obtener la información necesaria, y sus fuentes.

b) Preprocesamiento de datos: Consiste en la preparación y limpieza de los datos desde las distintas fuentes de datos. Aquí, se utilizan diversos métodos para el manejo de datos faltantes, inconsistentes, o datos en blanco, para poder trabajar con ellos.

c) Transformación de datos: Consiste en el tratamiento de los datos, y en la transformación y generación de nuevas variables en la base de datos. Además, se realizan otras operaciones, como normalización y consolidación de los datos procesados en las etapas anteriores.

d) *Data Mining*: Corresponde a la etapa de modelamiento de los datos, con el objetivo de poder identificar patrones previamente desconocidos, y que sean potencialmente útiles y comprensibles para la obtención de resultados.

e) Interpretación y Evaluación: Consiste en la identificación de patrones relevantes, mediante el uso de diversas medidas, y la evaluación de los resultados que se obtienen con los patrones resultantes.

V.1.3. CRISP – DM

CRISP – DM (*Cross Industry Standard Process for Data Mining*) corresponde a la descripción de procesos utilizados frecuentemente por los expertos en *Data Mining*. Las etapas principales de esta metodología se representan en la siguiente imagen:

Figura 5.3 Etapas de CRISP – DM (Fuente: https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining)

Las etapas que se representan en la figura anterior se describen a continuación:

a) *Business Understanding*: Consiste en la comprensión de los objetivos y requerimientos del proyecto desde una perspectiva empresarial o de negocios, no técnica. Con este

conocimiento, se buscará establecer los objetivos y criterios de *Data Mining*, junto con la definición de un plan de trabajo, para alcanzar los objetivos previamente establecidos.

b) *Data Understanding*: Consiste en entender los datos que se utilizarán para el proyecto, teniendo siempre en consideración los objetivos del negocio. Esta etapa comprende la recopilación inicial de los datos, exploración, identificación de problemas, entre otros.

c) *Data Preparation*: Consiste en la construcción del conjunto final de datos. En esta etapa, se encuentran como etapas a ejecutar la tabulación de los datos, selección de atributos, transformación y limpieza de datos, entre otros.

d) *Modeling*: Consiste en la selección y aplicación de las técnicas de modelamiento, o de *Data Mining*, en el conjunto final de datos.

e) *Evaluation*: Consiste en evaluar o determinar si los modelos utilizados en las etapas anteriores son los adecuados a las necesidades del proyecto y del negocio. Esta etapa tiene como objetivo determinar si existe alguna unidad de negocio importante que no ha sido completamente considerada en los modelos de las fases anteriores. Esta etapa termina cuando se toma la decisión de utilizar los resultados obtenidos en las etapas anteriores.

f) *Deployment*: Consiste en la integración de los modelos ejecutados anteriormente en la toma de decisiones de la empresa. Para esto, se debe hacer una planificación de esta integración, además de un monitoreo y un plan de mantenimiento. Una vez terminadas estas planificaciones, se procederá a realizar el informe final del proyecto y su revisión.

V.1.4. Comparación de metodologías

Al comparar las distintas metodologías que se describieron anteriormente, se puede ver que todas las metodologías son similares entre ellas, debido a que involucra obtención de los datos a utilizar, análisis de los datos, transformación de la base de datos mediante la creación de nuevas variables, modelamiento y evaluación de los resultados. Sin embargo, la metodología CRISP – DM es la única que incluye dentro del proceso, una etapa de comprensión del negocio, lo que permitirá enfocar el desarrollo del proyecto en las necesidades de la empresa de telecomunicaciones, a diferencia de, por ejemplo, la metodología SEMMA, debido a que se enfoca solamente en la base de datos, más que

en la comprensión e interpretación de los objetivos de la industria donde se aplicará la tesis.

Cabe mencionar que en la metodología KDD se puede incluir el análisis del negocio donde se enfocará el proyecto, antes de definir los datos a utilizar para el desarrollo de la tesis. Sin embargo, esta metodología no tiene un comportamiento cíclico, o de interacción entre cada una de las etapas, lo que no ocurre con la metodología CRISP – DM, debido a que permite una interacción continua entre cada una de las etapas del proceso.

VI. RESULTADOS OBTENIDOS

El desarrollo del presente trabajo se dividirá en dos partes: la primera parte consistirá en la segmentación de los clientes, utilizando variables socioeconómicas y financieras, como la edad, GSE, empaquetamiento, tiempo de permanencia, deuda del cliente, entre otras, mientras que la segunda parte consistirá en realizar un modelo de predicción de fuga de clientes, para los *clusters* que presentan mayor tasa de *churn*.

Además, para el desarrollo de ambas partes, se utilizará la metodología CRISP – DM. Junto con lo anterior, se destaca que la segmentación de clientes y la creación del modelo de predicción se encuentran relacionadas, dado que los segmentos obtenidos serán utilizados como datos de entrada para la implementación del modelo de predicción de *churn*, junto con variables financieras y operacionales, asociadas a los clientes. Esto se puede explicar en la siguiente imagen:

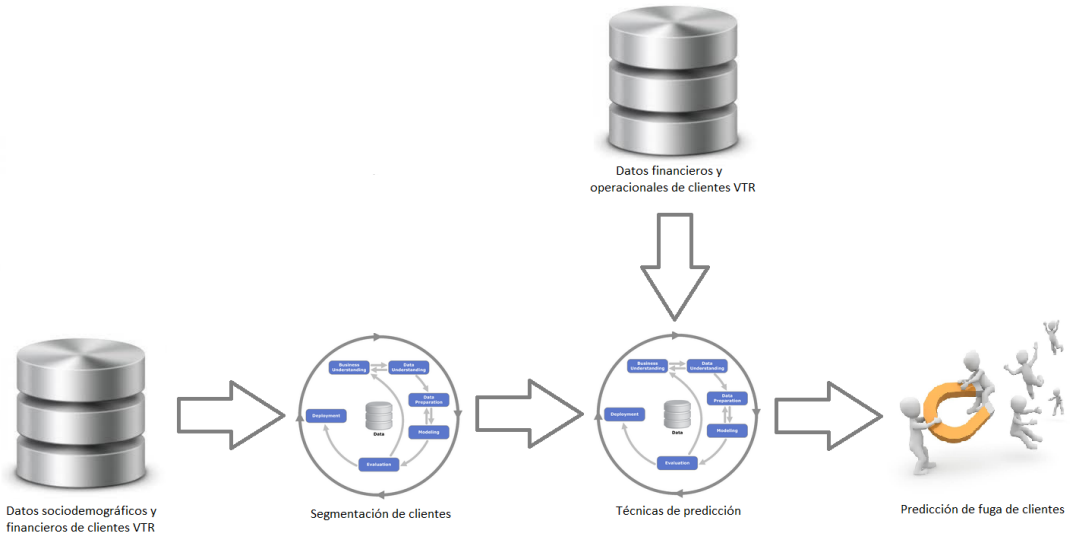


Figura 6.1 Desarrollo del trabajo (Fuente: Elaboración propia).

VI.1 Aplicación de CRISP – DM para la segmentación de clientes

Ahora, se procederá a aplicar la metodología CRISP – DM para obtener los distintos *clusters* de clientes, utilizando variables sociodemográficas y financieras, que permitan describir la situación del cliente con la compañía. A continuación, se procederá a describir los resultados obtenidos para cada una de las etapas que comprende esta metodología, asociada a la segmentación de los clientes.

VI.1.1. *Business Understanding*

En esta etapa, tal como se explicó en la metodología, se procederá a realizar una comprensión de los objetivos y requerimientos asociados a la segmentación de los clientes, desde un punto de vista empresarial, en vez de técnico. Además, se establecerá un plan de trabajo que permitirá alcanzar los objetivos establecidos.

Tal como se mencionó en la introducción, el problema que presenta la compañía se concentra en la cantidad de desconexiones que se producen mensualmente, ya sea por mora o voluntaria. Además, la tasa de *churn* ha presentado una tendencia al alza, especialmente en los clientes que tienen menos de 3 meses de permanencia en la compañía, y aunque la empresa tiene una política de retención de clientes, ésta no es proactiva, ya que consiste en comunicarse con los clientes, una vez que se está desconectando sus servicios, o están próximos a irse.

Debido a lo anterior, es importante implementar una política de retención proactiva, basada en un modelo estadístico, que permita mostrar resultados que sean de fácil comprensión para el ejecutivo de primera línea, y así tomar las medidas necesarias para retener al cliente, antes de que manifieste su intención de irse de la compañía, o que se desconecte por deudas impagas.

Ahora, se procederá a realizar la segmentación de los clientes, utilizando variables demográficas, y luego, se realizarán modelos de predicción para los *clusters* con mayor tasa de *churn*, con el objetivo de poder entregar una atención "personalizada", dependiendo del segmento al que pertenece el cliente.

VI.1.2. *Data Understanding*

Las variables a utilizar para realizar la segmentación de clientes son:

a) **Edad**: Corresponde a la edad de los clientes. Esta variable es importante para determinar tendencias en los servicios contratados, según los años que tenga el cliente.
b) **Grupo Socioeconómico (GSE)**: "El Índice Censal de Status Socioeconómico (ICSS) se basa en la clasificación según la posesión de bienes y el nivel de estudios del Jefe de Hogar. Su objetivo es clasificar a la población en segmentos que discriminen respecto a su poder adquisitivo de consumo, de su calidad material de vida, nivel cultural y estilo de vida"

(AIM, 2008). Un punto a destacar es que la cantidad de servicios que contrata un cliente está relacionado con el nivel socioeconómico del sector donde reside.

c) **Empaquetamiento**: Muestra cuántos servicios fueron contratados por el cliente, clasificándose en Mono (un servicio contratado), Dúo Pack (dos servicios contratados) y Triple Pack (todos los servicios contratados).

d) **Permanencia**: Consiste en la antigüedad del servicio activo, en días.

e) **Zona**: Corresponde a la zona donde pertenece la localidad en la que el servicio se encuentra instalado.

f) **Deuda**: Corresponde al monto adeudado que presenta el cliente.

g) **Flujo de cobranza**: Corresponde al tipo de flujo de cobranza que presenta el cliente. Existen tres tipos: Flujo Suave, Flujo Mediano y Flujo Duro.

h) **Profesión**: Corresponde a la profesión ejercida por el cliente.

i) **Cliente VIP**: Corresponde a una variable binaria para distinguir a clientes que son considerados como VIP.

j) **Sexo**: Corresponde a una variable binaria que sirve para identificar si los clientes son hombres o mujeres. En el caso de las empresas, esta variable aparece vacía, por lo que se completará con el valor 0.

k) **Días de deuda del cliente**: Corresponde a la cantidad de días que han transcurrido sin que el cliente pague su deuda.

l) **Ciclo de facturación**: Corresponde al ciclo que pertenece la cuenta del cliente, para que sea facturada.

m) **Tipo de boleta**: Variable binaria que muestra si el cliente recibe su boleta de forma electrónica (valor 1) o en papel (valor 0).

n) **Valor de facturación**: Corresponde al valor facturado al cliente en el último mes.

La base de datos a utilizar para la obtención de *clusters* corresponde a información de los clientes, con los servicios que se encuentran activos. Cabe mencionar que la llave utilizada para identificar a los distintos clientes de la compañía corresponderá a un campo combinado, formado por el RUT del cliente y el ID de la vivienda del mismo (este campo será llamado RUT_VIVIENDA). Se utilizará este campo como llave para identificar

a los clientes, debido a que un RUT puede estar asociado a una o más viviendas, y es probable que, en una vivienda, el cliente tenga contratado solamente un servicio, pero en la otra tenga contratado, por ejemplo, un Triple Pack.

Ahora, se procederá a analizar la variable de código de profesión, en comparación con la edad de los clientes. Para las cinco categorías que presentan más clientes, se pueden obtener *Boxplots*, con el objetivo de visualizar la distribución de las edades de los clientes en esas categorías. El gráfico se encuentra en Anexos (Anexo 1).

En la siguiente tabla, se muestra la descripción de los códigos de profesión que aparecen en los *boxplots*.

Código Profesión	Profesión
7	Dueña de Casa
8	Empleado
20	Otro
21	No informa
37	No tiene

Tabla 6.1 Profesiones (Fuente: Empresa de telecomunicaciones).

Ahora, cabe mencionar que los clientes que presentan como profesión 'Dueña de casa' presentan un promedio cercano a los 60 años de edad, a diferencia de los clientes con el código 37, que contiene a clientes de todas las edades. Un punto a mencionar es que en esa categoría, hay clientes que presentan una edad menor a 18 años (incluso, algunos con 0 años de edad), por lo que se revisará esta variable, para realizar algún procedimiento para poder reemplazar esos valores, o eliminar esos registros de la base. Un histograma de la edad de los clientes se encuentra en Anexos (Anexo 2), donde se pueden identificar casos de clientes que presentan una edad menor de 18 años, por lo que se revisará si esos valores serán sustituidos, utilizando un método de reemplazo, o los registros serán eliminados de la base.

Ahora, se realizarán diagramas de dispersión, con el objetivo de poder visualizar una cantidad aproximada de segmentos a obtener. Al realizar un histograma entre las variables EDAD y DIAS_PERMANENCIA, se puede obtener lo siguiente:

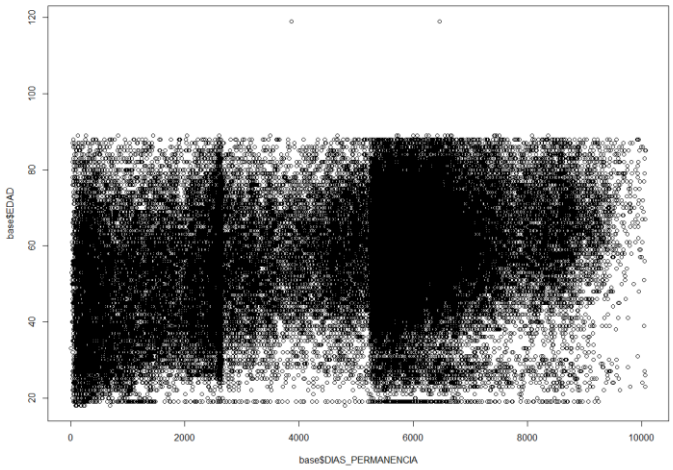


Figura 6.2 Histograma de DIAS_PERMANENCIA, con la edad de los clientes (Fuente: Elaboración propia).

Aquí, no es posible identificar una cantidad de segmentos a simple vista. Sin embargo, es posible identificar que se pueden obtener tres *clusters*, debido a que se puede identificar "bloques" de registros, los que se pueden interpretar como si fueran segmentos. Además, es posible identificar algunos *outliers* en la base, debido a la edad del cliente (alrededor de 120 años). Esos casos serán revisados para concluir si se eliminan de la base, o se mantienen.

Cabe mencionar que las otras variables son categóricas, por lo que se analizará una forma de trabajar con ellas para obtener una segmentación de clientes apropiada para continuar con el desarrollo de la tesis.

VI.1.3. *Data Preparation*

En esta sección, se procederá a la construcción de la base final de datos, para realizar la segmentación de los clientes. Para esto, se procederá a analizar los datos contenidos en la base inicial (o bruta), para verificar si hay que procesarlos. Luego, se analizará la varianza y las correlaciones existentes entre cada atributo de la base, y se eliminarán aquellos que tengan una varianza mínima y los que presentan una alta correlación entre ellos.

VI.1.3.1. Preprocesamiento de base de datos

Al revisar la base, se puede ver que existen registros donde la variable EDAD tiene valores negativos o tienen una edad menor a 18 años, lo que no está correcto. Para corregir esos

valores, se procederá a realizar un modelo de regresión para poder reemplazarlos con lo obtenido en el modelo, incluyendo los residuos calculados. Al realizar un gráfico de dispersión de las edades de los clientes (mayor o igual a 18 años), se obtiene lo siguiente:

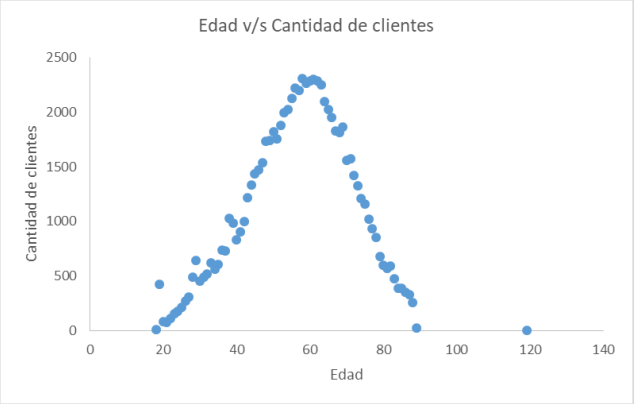


Figura 6.3 Gráfico de dispersión de las edades de clientes (Fuente: Elaboración propia).

Tal como se puede ver en la imagen, el comportamiento se puede representar como una **función cuadrática**. Debido a esto, se realizará un modelo de regresión polinómica de grado 2 para obtener los valores a reemplazar en la base de datos. Ese modelo se muestra en un gráfico, en la sección de Anexos (Anexo 3).

Luego, se procedió a estimar la cantidad de clientes para cada una de las edades, utilizando el modelo de regresión cuadrática, y se obtendrá la diferencia entre el valor real con el estimado.

En este caso, son 1.185 registros los que presentan esta inconsistencia. Luego, las edades con las que se reemplazarán serán **entre 21 y 30 años**, tal como se muestra en la siguiente tabla.

Edad	Q Clientes	Estimado	Diferencia
21	77	96	19
22	116	172	56
23	158	247	89
24	176	319	143
25	215	389	174
26	270	457	187
27	306	523	217
28	488	587	99
29	642	649	7
30	451	645	194
		Total:	**1.185**

Tabla 6.2 Resultados obtenidos para edades de 21 a 30 años (Fuente: Elaboración propia).

Junto con lo anterior, se procedió a agrupar las profesiones en cinco grupos:

1. Universitarios: Corresponde a las profesiones universitarias de los clientes, registradas en la base de datos, como abogados, arquitectos, médicos, ingenieros, entre otros.
2. Técnicos: Corresponde a aquellas profesiones que se pueden considerar de carácter técnico, como secretarias y contadores.
3. Uniformados: Son aquellos clientes que forman parte de las Fuerzas Armadas, o de alguna institución policial.
4. Empresas: Son las empresas que poseen servicios activos con la compañía, como por ejemplo, Pymes, hoteles, clínicas, restaurantes, etc.
5. Otros: En este grupo, se encuentran aquellos clientes, cuya profesión no se encuentra especificada en la base.

Otra modificación que se realizó a la base de datos es la transformación de seis variables categóricas, mediante el uso de variables *dummy*. Estas variables son **Empaquetamiento**, **Profesión**, **GSE**, **Flujo de Cobranza**, **Ciclo de Facturación** y **Zona**. Para la primera variable se crearon tres variables *dummies*; para la segunda, cinco variables *dummies*; para la tercera, seis variables *dummies*; para el flujo de cobranza, tres variables *dummies*; para el ciclo de facturación, diez variables *dummies*, y seis variables *dummies* para la última variable categórica.

Si se analiza la variable **Ciclo de facturación**, se puede obtener un gráfico circular, que se encuentra disponible en Anexos (Anexo 4).

En el gráfico, se puede ver que existen ciclos de facturación que poseen una mayor concentración de clientes, en comparación con otros ciclos (por ejemplo, el ciclo 14 contiene un 1,14% de clientes, y el ciclo 27 tiene un 25,09% de clientes). Cabe mencionar que a un cliente se le asigna un ciclo de facturación, de acuerdo al día de pago, previamente acordado, y se puede modificar, de acuerdo con lo que solicite el cliente. Debido a esto, se procederá a agrupar esta variable en tres conjuntos:

a) Ciclos de facturación grandes: Corresponde a una variable binaria que representa si el cliente pertenece a un ciclo de facturación que contenga, al menos, el 15% de la cartera de clientes.

b) **<u>Ciclos de facturación medianos</u>**: Corresponde a una variable binaria que representa si el cliente pertenece a un ciclo de facturación que contenga entre un 10% y un 15% de la cartera de clientes.

c) **<u>Ciclos de facturación pequeños</u>**: Corresponde a una variable binaria que representa si el cliente pertenece a un ciclo de facturación que contenga menos del 10% de la cartera de clientes.

Junto con lo anterior, cabe mencionar que la compañía presenta una distribución territorial de seis zonas (Norte, Centro, Sur, Metro Oriente, Metro Poniente y Metro Sur Oriente). Sin embargo, para el desarrollo de esta tesis, se unificarán las tres últimas áreas (debido a que pertenecen a la ciudad de Santiago) en una gran área, que se llamará **Zona Santiago**.

Además, se procederá a agrupar la variable **GSE** en tres conjuntos:

a) **<u>Sector alto</u>**: Corresponde a una variable binaria que representa si el cliente es residente de un sector, considerado dentro del grupo ABC1.

b) **<u>Sector medio</u>**: Corresponde a una variable binaria que representa si el cliente es residente de un sector, considerado dentro del grupo C2 y/o C3.

c) **<u>Sector bajo</u>**: Corresponde a una variable binaria que representa si el cliente es residente de un sector, considerado dentro del grupo D y/o E.

Cabe mencionar que aquellos clientes que tengan su vivienda en una zona que no se encuentre identificada con un GSE, no serán incluidos en alguno de estos conjuntos, sino que en una categoría distinta.

Con respecto a las variables que no son categóricas, se menciona que serán normalizadas, con el objetivo de analizar su comportamiento, y también se realizará una estandarización de las mismas, como otra forma de analizarlas.

VI.1.3.1.1. Base con variables no binarias normalizadas

Aquí, se procederá a normalizar las variables que no son binarias. Es decir, las variables EDAD, DIAS_PERMANENCIA, VALOR_DEUDA, DIAS_DEUDA y VALOR_FACTURACION.

Para analizar el comportamiento de las variables anteriores, se realizarán histogramas, los que también ayudarán a identificar si es necesario aplicar una transformación a las

variables, como por ejemplo, aplicar una función logarítmica a un atributo. Un histograma de la variable normalizada EDAD se encuentra en Anexos (Anexo 5).

En ese gráfico, se puede ver que la variable EDAD no presenta una concentración en un valor (dado que fue normalizada), por lo que no será necesario aplicar una función para transformar los valores del atributo.

Revisando los histogramas del resto de las variables que fueron normalizadas, se puede ver que tienen el mismo comportamiento que el de la figura 5.8. Debido a esto, no será necesario aplicar una función para transformar los valores de los atributos. Los gráficos se muestran en la sección de Anexos (Anexo 6, 7, 8 y 9). Además, el código implementado para la normalización de la base se encuentra en Anexos (Anexo 10).

VI.1.3.1.2. Base con variables no binarias estandarizadas con puntaje Z

Ahora, se procederá a estandarizar las variables no binarias, utilizando el puntaje Z. Para realizar esta estandarización, se procederá a calcular la media y la desviación estándar de cada uno de los atributos, y luego, se ejecutará un ciclo *for*, que permitirá obtener los valores estandarizados para cada variable. Este código se encuentra en la sección de Anexos (Anexo 11).

Un histograma de la variable estandarizada EDAD se encuentra en la sección de Anexos (Anexo 12), la que no presenta una gran concentración de clientes en torno a un valor, por lo que no se utilizará una función para transformar los valores de este atributo.

Ahora, un histograma de la variable DIAS_PERMANENCIA se encuentra en la sección de Anexos (Anexo 13), donde tampoco se puede distinguir una alta concentración de clientes en torno a algún valor en la variable, por lo que no se procederá a realizar una transformación.

Luego, un histograma de la variable VALOR_DEUDA estandarizada se encuentra en la sección de Anexos (Anexo 14). En este caso, sí se puede identificar una mayor concentración de clientes, en torno al valor 0. Debido a esto, se procederá a utilizar la función logarítmica natural para transformar el atributo VALOR_DEUDA, eliminándose la variable original de la base.

El histograma con la variable ya transformada se encuentra en Anexos (Anexo 15), donde se puede ver que los valores no se encuentran concentrados en torno a 0, a excepción de un valor muy cercano a 0. A pesar de lo anterior, se utilizará esta transformación para trabajar en la clusterización.

El histograma de la variable DIAS_DEUDA se encuentra en la sección de Anexos (Anexo 16), donde se puede ver una gran concentración de clientes, cuando el valor de esta variable es cercano a 0. Debido a esto, se procederá a aplicar la función logarítmica natural, y se eliminará la variable original. Una vez aplicada la función en los valores de la variable, se obtiene el histograma que se encuentra en la sección de Anexos (Anexo 17). En este caso, ya no se puede identificar una mayor concentración en valores cercanos a 0, pero sí en valores entre -1 y -1,5. Sin embargo, a pesar de lo anterior, se utilizará esta variable en la base de datos para clusterizar los clientes.

Ahora, el histograma de la variable estandarizada VALOR_FACTURACION se encuentra en la sección de Anexos (Anexo 18), donde se puede identificar una gran concentración de valores de la variable en torno a 0. Producto de esto, se procederá a aplicar la función logarítmica natural. Una vez aplicada la función, se obtiene el histograma que se encuentra en Anexos (Anexo 19), donde se puede ver que hay una mayor dispersión en los valores de la variable, pero aún se presenta una mayor concentración en valores cercanos a 0 (aunque se encuentre en los valores negativos). Sin embargo, a pesar de esto, se utilizará esta variable para realizar la clusterización de clientes.

VI.1.3.1.3. Base con variables normalizadas y estandarizadas

Primero, se procederá a normalizar la base completa, es decir, las variables binarias y las que no lo son. El código utilizado para obtener esta base se encuentra en Anexos (Anexo 20). Ahora, para analizar el comportamiento de las variables anteriores, se realizarán histogramas, los que también ayudarán a identificar si es necesario aplicar una transformación a las variables. El histograma de la variable EDAD, ya normalizada, se encuentra en la sección de Anexos (Anexo 21).

Como la variable fue normalizada, se puede ver que hay una mayor concentración de clientes en torno al valor 0. Además, el resto de las variables presentan el mismo

comportamiento. Producto de esto, no se aplicarán funciones matemáticas para modificar las variables de la base.

Luego, se procederá a estandarizar la base a clusterizar, incluyendo variables binarias y no binarias, utilizando puntaje Z. El código utilizado para obtener esta base se encuentra en Anexos (Anexo 22). Para analizar estas variables, se realizarán histogramas de los atributos en la base, con el objetivo de ver si es necesario aplicar una función matemática para transformar las variables, o no. El histograma de la variable 'Svc_Mono' se encuentra en la sección de Anexos (Anexo 23).

En este caso, se puede ver que los valores están ubicados en dos extremos del gráfico: valores en torno a -2,5 y otros valores cercanos a 0,5. Esto se debe a que esta variable es binaria, y porque existe una gran cantidad de clientes que tiene contratado un servicio con la compañía (ya sea Fono, Cable o Internet). Luego, si se aplica una función matemática para transformar este tipo de variable, se obtiene que el comportamiento no cambia, por lo que no se realizará una transformación a las variables binarias contenidas en la base.

Además, cabe mencionar que las variables no binarias contenidas en la base ya fueron analizadas en la sección VI.1.3.1.2, por lo que se utilizarán las mismas variables transformadas, obtenidas en esa sección.

VI.1.3.2. Filtros

Una vez analizadas las bases, se procederá a revisar cada una de sus variables, para ver aquellas que presenten mayor correlación, y eliminar las irrelevantes. Se trabajará con cada una de las bases obtenidas en las secciones anteriores, para obtener una base sin variables correlacionadas e irrelevantes, y así obtener el mejor resultado de la clusterización de clientes de la compañía.

VI.1.3.2.1. Análisis con variables no binarias normalizadas entre 0 y 1

Para realizar este análisis, primero se utilizará una matriz de varianzas, cuyo objetivo es identificar si existen variables que no ayudan en el proceso (se asumirá que la variable cumple con esta condición cuando el valor de la varianza sea **inferior a 0,01**). En caso de que existan, se eliminarán de la base. Luego, se aplicará una matriz de correlación a las

variables restantes, con el objetivo de identificar las que son redundantes, o mayormente correlacionados (se considerará que dos variables están altamente correlacionadas cuando el valor de su correlación sea mayor a 0.9, en valor absoluto).

La tabla con las varianzas obtenidas en la base se encuentra en Anexos (Anexo 24), donde las variables que podrían ser consideradas como irrelevantes son 'PROF_4' (o sea, los clientes empresas) y 'FLAG_VIP'. Debido a esto, ambas variables serán eliminadas de la base.

Luego, se procedió a obtener la matriz de correlación de la base, obteniendo que las variables no presentaban una alta correlación entre ellas (es decir, una correlación mayor a 0,9), por lo que se trabajará con esas variables. La matriz de correlación obtenida se encuentra en la sección Anexos (Anexo 25).

VI.1.3.2.2. Análisis con variables no binarias estandarizadas con puntaje Z

Ahora, se realizará el análisis de la base con las variables no binarias normalizadas con puntaje Z.

Una vez normalizados los atributos, se procederá a calcular las varianzas de cada una de las variables que compone la base, cuyos resultados se encuentran en Anexos (Anexo 26). Ahora, tal como en el análisis de la base con las variables no binarias normalizadas entre 0 y 1, se considerará que una variable no ayuda en el proceso de clusterización (o no es relevante), cuando el valor de su varianza es **inferior a 0,01**. En este caso, tal como ocurrió anteriormente, las variables 'PROF_4' y 'FLAG_VIP' presentan una varianza menor a 0,01, por lo que serán eliminadas de la base.

Ahora, se procederá a obtener la matriz de correlación de la base. Una vez calculada la matriz, se obtuvo que las variables que presentaban mayor correlación son 'Svc_Mono' y 'Svc_Duo', por lo que se eliminará una de ellas de la base a clusterizar. Cabe mencionar que, si bien la proporción de clientes que tienen un servicio activo y que contrataron un dúo pack es similar, existen más usuarios que presentan solamente un servicio activo. Por tanto, se excluirá de la base la variable 'Svc_Duo'.

Finalmente, al obtener la matriz de correlación sin la variable 'Svc_Duo', tal como en el análisis anterior, se pudo determinar que el resto de los atributos no presentaban una

alta correlación entre ellos (es decir, una **correlación mayor a 0,9**), por lo que se trabajará con esas variables. La matriz de correlación obtenida se encuentra en la sección Anexos (Anexo 27).

VI.1.3.2.3. Análisis con variables binarias y no binarias normalizadas entre 0 y 1

En este análisis, se realizará una matriz de varianzas a las variables binarias y no binarias normalizadas que conforman la base, para identificar las variables que no ayudan en el proceso y excluirlas. Luego, se desarrollará una matriz de correlación, con el objetivo de identificar las variables que tengan una alta correlación, y excluir a una de ellas de la base.

Las varianzas calculadas de las variables que componen la base se encuentran en Anexos (Anexo 28). En este caso, las varianzas obtenidas son cercanas a 1, producto del proceso de normalización. Debido a esto, no se excluirá ninguna variable de la base a clusterizar. Luego, se procedió a obtener la matriz de correlación de la base. Una vez realizada la matriz de correlación, no se pudo identificar valores mayores a 0,9. Debido a esto, no se eliminarán variables de esta base. La matriz de correlación final se encuentra en la sección de Anexos (Anexo 29).

VI.1.3.2.4. Análisis con variables binarias y no binarias, estandarizadas con puntaje Z

En este caso, se analizará la base de variables binarias y no binarias, estandarizadas con puntaje Z. Las varianzas calculadas para esta base se encuentran en Anexos (Anexo 30), donde la mayoría de las variables de la base presentan una varianza igual a 1, producto de la estandarización. Debido a esto, no se excluirá ninguna variable de la base.

Al obtener la matriz de correlación de la base, las variables 'Svc_Mono' y 'Svc_Duo' presentaron mayor correlación entre ellas. Por tanto, se excluirá del análisis la variable 'Svc_Duo'.

Una vez excluido, no se lograron identificar otras variables que presentaban una alta correlación entre ellos. Cabe mencionar que la matriz de correlación obtenida para esta base se encuentra en la sección de Anexos (Anexo 31).

VI.1.4. *Modeling*

En esta sección, se procederá a aplicar distintos métodos de *clustering* para obtener una segmentación de los clientes de la compañía de telecomunicaciones. Para identificar la mejor segmentación, se comparará la **suma de los errores cuadráticos medios (SSE)** calculado en las distintas técnicas de clusterización, y se escogerá el que presente el menor error.

Las técnicas de *clustering* que se utilizarán para la segmentación de los clientes de la compañía, en cada una de las bases analizadas anteriormente, serán *K – Means*, método de Ward, DBSCAN y *K – Medoids*.

VI.1.4.1. Base de clientes con variables no binarias, normalizadas entre 0 y 1

Aquí, se utilizará la base de clientes, donde las variables no binarias se encuentran normalizadas entre 0 y 1. Sobre esta base, se aplicarán los distintos métodos de *clustering* mencionados anteriormente, y se registrará el mejor resultado obtenido para cada técnica.

VI.1.4.1.1. *K – Means*

En esta sección, se utilizará el algoritmo *K – Means* para obtener *clusters* de los clientes de la empresa de telecomunicaciones.

Para aplicar este algoritmo, se necesita una cantidad determinada de *clusters* a obtener, definido previamente. Para esto, se utilizará un método llamado "*Elbow Method*". En este caso, se realizará un gráfico con la cantidad de *clusters* utilizados para la segmentación, y el porcentaje de "explicabilidad" de los resultados obtenidos. Este gráfico se encuentra en la sección de Anexos (Anexo 32).

En ese gráfico, se puede ver que el "codo" se produce cuando el número de *clusters* se encuentra entre diez y veinte. Ahora, se desarrollará otro gráfico, que mostrará la cantidad de *clusters* y el error obtenido en el proceso de *clustering*. Este gráfico se encuentra en Anexos (Anexo 33).

Aquí, tal como en el gráfico anterior, el "codo" aparece cuando el número de *clusters* se encuentra entre diez y veinte. Luego, se optará por escoger **16 *clusters***, debido a que en este punto, se produce el "codo" en ambas curvas.

Para ejecutar *K – Means*, se utilizará el software R, en donde se implementará un código para obtener los *clusters*. Este código se encuentra en Anexos (Anexo 34). Ahora, cabe mencionar que el primer intento o ejecución del algoritmo puede que no entregue un buen resultado, por lo que se realizarán distintos intentos para encontrar el mejor resultado, el cual será el que entregue el menor error de todos los intentos ejecutados. En este caso, se realizaron veinte intentos, y se graficaron los distintos errores obtenidos en cada uno. Este gráfico se encuentra en Anexos (Anexo 35).

Al revisar el gráfico, el intento que presentó el menor error fue el **intento número 11**, por lo que se utilizarán los resultados obtenidos en ese intento como los *clusters* definitivos para esa base, con *K – Means*.

VI.1.4.1.2. Método de Ward

En esta sección, se utilizará el método de Ward, para obtener *clusters* de clientes. Este método se ejecutará en el software R, y su código se encuentra en la sección de Anexos (Anexo 36). Al revisar el código, se puede ver que se tomó una muestra aleatoria de 385 registros para realizar el dendrograma, ya que el proceso presenta dificultades al tomar la base de datos completa.

Cabe mencionar que se determinó la muestra aleatoria considerando una población de 1.300.000 clientes, un 95% de confianza, un valor sigma de 0,5 y un margen de error de 0,05.

Se ejecutó cinco veces el código, y los dendrogramas obtenidos, junto con la cantidad de *clusters* identificados se encuentran en la sección de Anexos (Anexos 37 a 46).

Ahora, si analizamos el error obtenido en cada intento, se puede concluir que el modelo que presentó el menor error fue el **intento 2**, tal como se puede ver en la siguiente tabla:

Intentos	SSE
1	3044,08
2	**2976,10**
3	3050,35
4	3044,37
5	2979,22

Tabla 6.3 Valores de SSE de cada intento del método de Ward (Fuente: Elaboración propia).

Por tanto, se puede concluir que es posible obtener 29 *clusters* de clientes.

VI.1.4.1.3. DBSCAN

En esta sección, se utilizará el algoritmo DBSCAN para obtener *clusters* de clientes de la empresa de telecomunicaciones.

Cabe mencionar que una de las ventajas que presenta este algoritmo es que no es necesario definir previamente la cantidad de *clusters* que se pretenden obtener, sino que se tienen que definir dos parámetros: ***épsilon*** (ε) y los **puntos mínimos** (que corresponde a la cantidad de puntos mínimos que debe tener cada *cluster* dentro de un círculo de radio ε). En este caso, el indicador que debe ser optimizado es ε, ya que el algoritmo es más sensible a este parámetro.

Para obtener un valor adecuado de épsilon, se calcularán las distancias de los k – vecinos más cercanos, las que serán graficadas y se identificará el "codo" en el gráfico resultante (tal como en "*Elbow Method*"). Además, el valor de *k* utilizado será el parámetro utilizado en DBSCAN como **puntos mínimos**. Sin embargo, dependiendo del valor de *k* utilizado, se obtendrá un valor de épsilon distinto, y por ende, una cantidad de *clusters* distinta. Producto de esto, se ejecutará este método varias veces, con distintos valores de *k*, con el objetivo de identificar el valor adecuado de este parámetro a ser utilizado y la cantidad final de *clusters* a obtener, bajo este algoritmo.

Un gráfico de los valores de *k* utilizados para calcular las distancias de los k – vecinos más cercanos, y el épsilon calculado para cada *k*, se muestra en Anexos (Anexo 47), donde se puede ver que no es posible identificar un "codo" en el gráfico, por lo que no es posible identificar un valor de ε con un valor de *k* adecuados para el desarrollo de DBSCAN (tal como en *Elbow Method* aplicado a *K – Means*). Debido a esto, se ejecutó DBSCAN con cada uno de los valores de épsilon y *k* obtenidos anteriormente, cuyos resultados se muestran en un gráfico que se encuentra en Anexos (Anexo 48).

En este caso, es posible identificar un quiebre en el gráfico, cuando $k = 3$ y por ende, $\varepsilon = 5.2$. El gráfico donde se puede identificar el valor de épsilon se encuentra en Anexos (Anexo 49).

Ahora, un punto importante a considerar en la ejecución de DBSCAN consiste en la cantidad de datos que son considerados como *outliers*, los que pueden variar

dependiendo del valor de *k* utilizado. Si este valor es pequeño (como el que se utilizó para el desarrollo de la tesis), entonces es probable que considere una cantidad mayor de puntos atípicos. Debido a esto, se debe considerar un valor de *k*, de tal forma que la cantidad de *outliers* no sea muy alta. Un gráfico de datos atípicos, por valor de *k* utilizado, se encuentra en Anexos (Anexo 50).

Además, un gráfico de los puntos atípicos con la cantidad de *clusters* obtenidos por DBSCAN se encuentra en Anexos (Anexo 51), donde se puede ver que el "codo" se produce cuando los valores de *k* se encuentran entre 3 y 5. Sin embargo, la cantidad de datos atípicos que son identificados en DBSCAN cuando *k* es igual a 3 ó 4 es muy alta. Producto de esto, se requiere un valor de *k* tal que el resultado obtenido no tenga una gran cantidad de *outliers*, por lo que se optaría para valores de *k* iguales a 5 ó 6. Luego, para el desarrollo de la tesis, se optará por *k* = 5, donde se obtienen **11 clusters**, y 35 *outliers*. Un gráfico de los *clusters* obtenidos y el código utilizado se encuentran en Anexos (Anexos 52 y 53).

VI.1.4.1.4. *K – Medoids*

En esta sección, se utilizará el método *K – Medoids* para obtener *clusters* de clientes, el cual fue ejecutado en R, mediante el algoritmo CLARA (*Clustering Large Applications*). El código utilizado se encuentra en la sección de Anexos (Anexo 54).

Ahora, tal como en el algoritmo *K – Means*, este método necesita que se define previamente la cantidad de *clusters* a obtener para realizar la segmentación. Debido a esto, se utilizará el método, llamado *Elbow Method*, con el objetivo de poder identificar la cantidad de *clusters* óptima para segmentar la base de clientes.

El gráfico de la cantidad de *clusters* con su respectivo error se encuentra en Anexos (Anexo 55), donde se puede ver que el "codo" se forma cuando se obtienen entre nueve y doce *clusters*. Debido a esto, se ejecutará el algoritmo *K – Medoids* con ***k* = 12**. Cabe mencionar que, tal como ocurre con *K – Means*, el primer intento no entrega necesariamente una buena solución, por lo que se realizarán varios intentos, y se utilizará el que tenga el menor error.

Después de ejecutar varias veces el algoritmo, con ***k* = 12** (y variando la cantidad de registros como muestra que toma el algoritmo para su ejecución en R), se obtiene el gráfico que se encuentra en Anexos (Anexo 56). Luego, en el **intento 7** se obtiene el mejor resultado para este algoritmo.

VI.1.4.2. Base de clientes con variables no binarias estandarizadas con puntaje Z

Aquí, se utilizará la base de clientes con variables no binarias, pero que fueron estandarizadas con puntaje Z, y se aplicarán los distintos métodos de *clustering* utilizados anteriormente, registrando los mejores resultados obtenidos por cada algoritmo.

VI.1.4.2.1. *K – Means*

En esta sección, se utilizará el algoritmo *K – Means* para obtener *clusters* de clientes de la empresa de telecomunicaciones.

Ahora, para aplicar este algoritmo, se necesita una cantidad determinada de *clusters*, definido previamente. Para encontrar la cantidad de *clusters*, se utilizará "*Elbow Method*". Primero, se graficarán la cantidad de *clusters* utilizados para la segmentación, y el porcentaje de "explicabilidad" de los resultados obtenidos. Este gráfico se muestra a continuación:

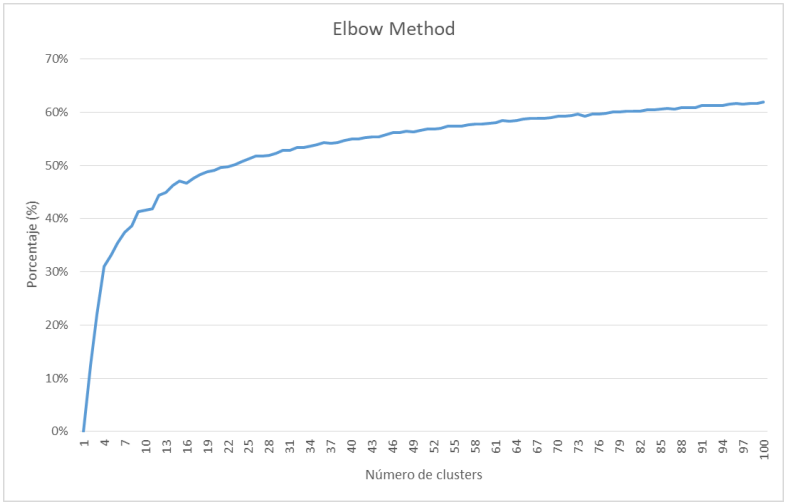


Figura 6.4 Número de *clusters* v/s tasa de explicabilidad (Fuente: Elaboración propia).

En ese gráfico, se puede ver que el "codo" o quiebre en la curva se produce cuando el número de *clusters* se encuentra entre siete y quince. Ahora, se desarrollará otro gráfico, que mostrará la cantidad de *clusters* y el error obtenido en el proceso de *clustering*. Este gráfico se muestra a continuación:

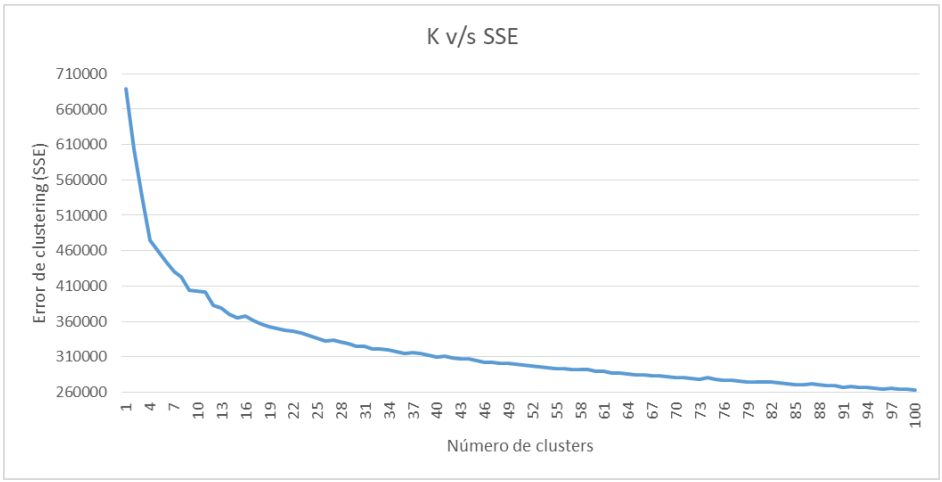


Figura 6.5 Número de *clusters* v/s error del algoritmo de *clustering* (Fuente: Elaboración propia).

En este caso, se puede ver que el "codo" se produce cuando la cantidad de *clusters* se encuentra entre siete y quince. Ahora, para esta base, se optará por obtener **15 clusters** de clientes.

Cabe mencionar que este algoritmo se ejecutó en R, y su código se encuentra adjunto en la sección de Anexos (Anexo 57). Ahora, cabe recordar que nunca se obtiene un buen resultado de clusterización cuando se ejecuta una vez el algoritmo, por lo que se ejecutará varias veces, y se optará por el resultado que presente el menor error. El gráfico con las diversas ejecuciones del algoritmo en R, con su respectivo error calculado, se muestra a continuación:

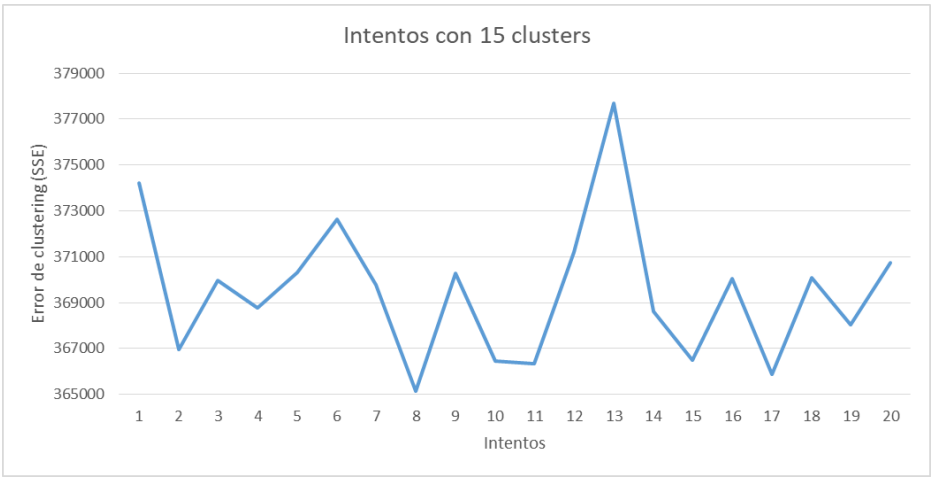


Figura 6.6 Distintos intentos de soluciones con *K – Means* (Fuente: Elaboración propia).

Como se puede ver en el gráfico anterior, el intento que presentó el menor error fue el **intento número 8**, por lo que se utilizarán los resultados obtenidos como los *clusters* definitivos para esa base.

VI.1.4.2.2. Método de Ward

Ahora, se utilizará el método de Ward para obtener *clusters* de clientes. Este método se ejecutará en R, y su código se encuentra en la sección de Anexos (Anexo 58).

Al revisar el código, se puede ver que se tomó una muestra aleatoria de 385 registros para realizar el dendrograma, ya que el proceso presenta dificultades al tomar la base de datos completa.

Cabe mencionar que se determinó la muestra aleatoria considerando una población de 1.300.000 clientes, un 95% de confianza, un valor sigma de 0,5 y un margen de error de 0,05.

Se ejecutó cinco veces el código en R, y los resultados obtenidos se muestran en la sección de Anexos (Anexos 59 a 68).

Ahora, si analizamos los valores de SSE obtenidos para cada uno de los intentos, se puede concluir que el modelo que presentó el menor valor fue el **intento 1**, tal como se puede ver en la siguiente tabla:

Intentos	SSE
1	**2451,29**
2	3580,94
3	3879
4	2485,79
5	3107,76

Tabla 6.4 Valores de SSE de cada intento con el método de Ward (Fuente: Elaboración propia).

Por tanto, para esta base, se puede concluir que se pueden obtener 29 *clusters* de clientes.

VI.1.4.2.3. DBSCAN

En esta sección, se utilizará el algoritmo DBSCAN para obtener *clusters* de clientes de la empresa de telecomunicaciones.

Cabe mencionar que una de las ventajas que presenta este algoritmo es que no es necesario definir previamente la cantidad de *clusters* que se pretenden obtener, sino que se tienen que definir dos parámetros: ***épsilon*** (ε) y los **puntos mínimos** (que corresponde a la cantidad de puntos mínimos que debe tener cada *cluster* dentro de un círculo de

radio ε). En este caso, el indicador que debe ser optimizado es ε, ya que el algoritmo es más sensible a este parámetro.

Para obtener un valor adecuado de épsilon, se calcularán las distancias de los k – vecinos más cercanos, las que serán graficadas y se identificará el "codo" en el gráfico resultante (tal como en "*Elbow Method*"). Además, el valor de *k* utilizado será el parámetro utilizado en DBSCAN como **puntos mínimos**. Sin embargo, dependiendo del valor de *k* utilizado, se obtendrá un valor de épsilon diferente, y por ende, una cantidad de *clusters* distinta. Producto de esto, se ejecutará este método varias veces, con distintos valores de *k*, con el objetivo de identificar el valor adecuado de este parámetro a ser utilizado y la cantidad final de *clusters* a obtener, bajo este algoritmo.

Un gráfico de los valores de *k* utilizados para calcular las distancias de los k – vecinos más cercanos, y el épsilon calculado para cada *k*, se muestra en Anexos (Anexo 69). Aquí, se puede ver que no es posible identificar un "codo" o quiebre en el gráfico, por lo que no es posible identificar un valor de ε con un valor de *k* adecuados para el desarrollo de DBSCAN (tal como en *Elbow Method* aplicado a *K – Means*). Debido a esto, se ejecutó DBSCAN con cada uno de los valores de épsilon y *k* obtenidos anteriormente, cuyos resultados se muestran en Anexos (Anexo 70).

En este caso, es posible identificar un quiebre en el gráfico, cuando *k* = 3 y por ende, ε = 5.1. Ahora, un punto importante a considerar en la ejecución de DBSCAN consiste en la cantidad de datos que son considerados como *outliers*, los que pueden variar dependiendo del valor de *k* utilizado. Si este valor es pequeño (como el que se mencionó anteriormente), entonces es probable que se estén considerado una cantidad mayor de puntos atípicos. Debido a esto, se debe considerar un valor de *k* tal, que la cantidad de *outliers* no sea muy alta. Un gráfico de la cantidad de datos atípicos identificados por DBSCAN, por valor de *k* utilizado, se encuentra en Anexos (Anexo 71).

Ahora, un gráfico de los puntos atípicos con la cantidad de *clusters* obtenidos por DBSCAN se encuentra en la sección de Anexos (Anexo 72). Analizando el gráfico, se puede ver que el "codo" se produce cuando los valores de *k* se encuentran entre 3 y 7. Sin embargo, la cantidad de datos atípicos que son identificados en DBSCAN cuando *k* es

gual a 3 ó 4 es muy alta. Producto de esto, se requiere un valor de *k* tal que el resultado obtenido no tenga una gran cantidad de *outliers*, por lo que se optaría para valores de *k* iguales a 5, 6 ó 7. Luego, para el desarrollo de la tesis, se optará por *k* = 6, donde se obtienen **20 clusters**, y 493 *outliers*. Cabe mencionar que, para realizar DBSCAN con *k* = 6, se asumió un valor de ε igual a 5.8, y el gráfico de los valores de épsilon en DBSCAN se encuentra en Anexos (Anexo 73).

La representación gráfica de los *clusters* obtenidos y el código utilizado para ejecutar este algoritmo se encuentran en Anexos (Anexos 74 y 75).

VI.1.4.2.4. *K – Medoids*

En esta sección, se utilizará el método *K – Medoids* para obtener *clusters* de clientes en esta base. Este método fue ejecutado en R, mediante el algoritmo CLARA (*Clustering Large Applications*), y el código utilizado se encuentra en la sección de Anexos (Anexo 76).

Ahora, tal como en el algoritmo *K – Means*, este método necesita que se defina previamente la cantidad de *clusters* a obtener para realizar la segmentación. Debido a esto, se utilizará *Elbow Method*, con el objetivo de poder identificar la cantidad de *clusters* óptima para segmentar la base de clientes. Al graficar la cantidad de *clusters* y el error obtenido en el proceso, se obtiene la imagen que se encuentra en Anexos (Anexo 77).

En este caso, se puede ver que el "codo" o quiebre se produce cuando la cantidad de *clusters* a obtener se encuentra entre siete y diez. Debido a esto, se ejecutará el algoritmo *K – Medoids* con ***k* = 10**. Cabe mencionar que, tal como ocurre con *K – Means*, el primer intento no entrega necesariamente una buena solución, por lo que se realizarán varios intentos, y se utilizará el que tenga el menor error.

Después de ejecutar varias veces el algoritmo, con *k* = 10 (y variando la cantidad de registros como muestra que toma el algoritmo para su ejecución en R), se obtienen el gráfico que se encuentra en Anexos (Anexo 78).

En este caso, se puede ver que el mejor resultado se obtiene en el **intento 7**, por lo que se utilizará ese resultado para el desarrollo de la tesis.

VI.1.4.3. Base de clientes con variables binarias y no binarias normalizadas entre 0 y 1

En esta sección, se utilizará la base de clientes con variables binarias y no binarias que fueron normalizadas entre 0 y 1, y se aplicarán los distintos métodos de *clustering* utilizados anteriormente, registrando los mejores resultados obtenidos por cada algoritmo.

VI.1.4.3.1. *K – Means*

Para aplicar este algoritmo, se necesita una cantidad determinada de *clusters*, definido previamente. Para encontrar esa cantidad, se utilizará "*Elbow Method*". Primero, se graficarán la cantidad de *clusters* utilizados para la segmentación, y el porcentaje de "explicabilidad" de los resultados obtenidos. Este gráfico se encuentra en la sección de Anexos (Anexo 79).

En el gráfico anterior, se puede ver que el "codo" o quiebre en la curva se produce cuando el número de *clusters* se encuentra entre diez y quince. Ahora, se desarrollará otro gráfico, que mostrará la cantidad de *clusters* y el error obtenido en el proceso de *clustering*. Este gráfico se encuentra en Anexos (Anexo 80).

En este caso, se puede ver que el "codo" se produce cuando la cantidad de *clusters* se encuentra entre siete y trece. Ahora, para esta base, se optará por obtener **13 *clusters*** de clientes.

Cabe mencionar que este algoritmo se ejecutó en el *software* R, y su código se encuentra adjunto en la sección de Anexos (Anexo 81). Ahora, cabe recordar que nunca se obtiene un buen resultado de clusterización cuando se ejecuta una vez el algoritmo, por lo que se ejecutará varias veces, y se optará por el resultado que presente el menor error. Las diversas ejecuciones del algoritmo en R, con su respectivo error calculado, se muestran en un gráfico, en la sección de Anexos (Anexo 82).

Como se puede ver en el gráfico anterior, el intento que presentó el menor error fue el **intento número 6**, por lo que se utilizarán los resultados obtenidos en ese intento como los *clusters* definitivos para esa base.

VI.1.4.3.2. Método de Ward

Ahora, se utilizará el método de Ward para esta base, con el objetivo de obtener *clusters* de clientes. Este método se ejecutará en R, y su código se encuentra en la sección de Anexos (Anexo 83).

Al revisar el código, se puede ver que se tomó una muestra aleatoria de 385 registros para realizar el dendrograma, ya que el proceso presenta dificultades al tomar la base de datos completa. Además, se determinó la muestra aleatoria considerando una población de 1.300.000 clientes, un 95% de confianza, un valor sigma de 0,5 y un margen de error de 0,05.

Se ejecutó cinco veces el código en R, y los resultados obtenidos se muestran en Anexos (Anexos 84 a 93).

Ahora, si analizamos los valores de SSE obtenidos para cada uno de los intentos, se puede concluir que el modelo que presentó el menor valor fue el **intento 1**, tal como se puede ver en la siguiente tabla:

Intentos	SSE
1	**10075,40**
2	10559,39
3	10182,14
4	10338,82
5	10425,93

Tabla 6.5 Valores de SSE calculados para cada intento con el método de Ward (Fuente: Elaboración propia).

Por tanto, para esta base, se puede concluir que se pueden obtener **30 *clusters*** de clientes.

VI.1.4.3.3. DBSCAN

Una de las ventajas que presenta este algoritmo es que no es necesario definir previamente la cantidad de *clusters* que se pretenden obtener, sino que se tienen que definir dos parámetros: **épsilon** (ε) y los **puntos mínimos** (que corresponde a la cantidad de puntos mínimos que debe tener cada *cluster* dentro de un círculo de radio ε). En este caso, el indicador que debe ser optimizado es **ε**, ya que el algoritmo es más sensible a este parámetro.

Para obtener un valor adecuado de épsilon, se calcularán las distancias de los k – vecinos más cercanos, las que serán graficadas y se identificará el "codo" en el gráfico resultante (tal como en "*Elbow Method*"). Además, el valor de *k* utilizado será el parámetro utilizado en DBSCAN como puntos mínimos. Sin embargo, dependiendo del valor de *k* utilizado, se obtendrá un valor de épsilon diferente, y por ende, una cantidad de *clusters* distinta. Producto de esto, se ejecutará este método varias veces, con distintos valores de *k*, con el objetivo de identificar el valor adecuado de este parámetro a ser utilizado y la cantidad final de *clusters* a obtener, bajo este algoritmo. El gráfico resultante se encuentra en Anexos (Anexo 94).

Se puede ver que, tal como en los análisis anteriores, no es posible identificar un "codo" o quiebre en el gráfico, por lo que no es posible identificar un valor de ε con un valor de *k* adecuados para el desarrollo de DBSCAN (tal como en *Elbow Method* aplicado a *K – Means*). Debido a esto, se ejecutó DBSCAN con cada uno de los valores de épsilon y *k* obtenidos anteriormente, cuyos resultados se encuentran en Anexos (Anexo 95).

En este caso, es posible identificar un quiebre en el gráfico, cuando *k* = 3 y por ende, ε = 8.6. Ahora, como se ha mencionado anteriormente, un punto importante a considerar en la ejecución de DBSCAN consiste en la cantidad de datos que son considerados como *outliers*, los que pueden variar dependiendo del valor de *k* utilizado. Si este valor es pequeño (como el que se mencionó anteriormente), entonces es probable que se esté considerado **una cantidad mayor de puntos atípicos** u *outliers*. Debido a esto, se debe considerar un valor de *k*, de tal forma que la cantidad de *outliers* no sea muy alta. Un gráfico con la cantidad de datos atípicos identificados por DBSCAN, por valor de *k* utilizado, se muestra en Anexos (Anexo 96).

Ahora, un gráfico de los puntos atípicos, con la cantidad de *clusters* obtenidos por DBSCAN en un solo gráfico, se muestra en Anexos (Anexo 97), donde se puede ver que el "codo" se produce cuando los valores de *k* se encuentran entre 3 y 7, tal como en el análisis de DBSCAN con la base de la sección anterior. Sin embargo, la cantidad de datos atípicos que son identificados en DBSCAN cuando *k* es igual a 3 ó 4 es muy alta. Producto de esto, se requiere un valor de *k* tal que el resultado obtenido no tenga una gran

cantidad de *outliers*, por lo que se optaría para valores de *k* iguales a 5, 6 ó 7. Luego, para el desarrollo de la tesis, se optará por ***k* = 5**, donde se obtienen 5 *clusters*, y 58 *outliers*. Cabe mencionar que, para realizar DBSCAN con k = 5, se asumió un valor de ε igual a 9.0, y su gráfico se encuentra en Anexos (Anexo 98).

Ahora, la representación gráfica de los *clusters* obtenidos y el código utilizado para la ejecución de este algoritmo se encuentra en Anexos (Anexos 99 y 100).

VI.1.4.3.4. *K – Medoids*

Este método fue ejecutado en R, mediante el algoritmo CLARA (*Clustering Large Applications*), y el código utilizado se encuentra en la sección de Anexos (Anexo 101). Ahora, tal como en el algoritmo *K – Means*, este método necesita que se define previamente la cantidad de *clusters* a obtener para realizar la segmentación. Debido a esto, se utilizará *Elbow Method*, con el objetivo de poder identificar la cantidad de *clusters* óptima para segmentar la base de clientes. El gráfico de la cantidad de *clusters* y el error obtenido en el proceso, se encuentra en Anexos (Anexo 102).

En este caso, se puede ver que el "codo" se produce cuando la cantidad de *clusters* se encuentra entre 4 y 12. Debido a esto, se ejecutará el algoritmo *K – Medoids* con ***k* = 7**. Cabe mencionar que, tal como ocurre con *K – Means*, el primer intento no entrega necesariamente una buena solución, por lo que se realizarán varios intentos, y se utilizará el que tenga el menor error.

El gráfico resultante después de ejecutar varias veces el algoritmo, con ***k* = 7** (y variando la cantidad de registros como muestra que toma el algoritmo para su ejecución en R), se encuentra en Anexos (Anexo 103).

En este caso, se puede ver que el mejor resultado se obtiene en el **intento 8**, por lo que se utilizará ese resultado para el desarrollo de la tesis.

VI.1.4.4. Base de clientes con variables binarias y no binarias estandarizadas con puntaje Z

En esta sección, se utilizará la base de clientes con variables binarias y no binarias que fueron estandarizadas con puntaje Z, y se aplicarán los distintos métodos de *clustering*

utilizados anteriormente, registrando los mejores resultados obtenidos por cada algoritmo.

VI.1.4.4.1. *K – Means*

Para aplicar el algoritmo de *K – Means*, se necesita una cantidad determinada de *clusters*, definido previamente. Para encontrar la cantidad de *clusters*, se utilizará "*Elbow Method*". Primero, se graficarán la cantidad de *clusters* utilizados para la segmentación, y el porcentaje de "explicabilidad" de los resultados obtenidos. Este gráfico se encuentra en Anexos (Anexo 104).

En el gráfico anterior, se puede ver que el "codo" o quiebre en la curva se produce cuando el número de *clusters* se encuentra entre **15** y **22**. Ahora, se desarrollará otro gráfico, que mostrará la cantidad de *clusters* y el error obtenido en el proceso de *clustering*. Este gráfico se encuentra en Anexos (Anexo 105).

En este caso, nuevamente se puede ver que el "codo" se produce cuando la cantidad de *clusters* se encuentra entre 15 y 22. Ahora, para esta base, se optará por obtener **22 *clusters*** de clientes.

Cabe mencionar que este algoritmo se ejecutó en el *software* R, y su código se encuentra adjunto en la sección de Anexos (Anexo 106). Ahora, cabe recordar que nunca se obtiene un buen resultado de clusterización cuando se ejecuta una vez el algoritmo, por lo que se ejecutará varias veces, y se optará por el resultado que presente el menor error. Las diversas ejecuciones del algoritmo en R, con su respectivo error calculado, se muestran en un gráfico, el cual se encuentra en Anexos (Anexo 107).

Ahora, el intento que presentó el menor error fue el **intento número 12**, por lo que se utilizarán los resultados obtenidos en ese intento como los *clusters* definitivos para esa base.

VI.1.4.4.2. Método de Ward

Ahora, se utilizará el método de Ward para esta base, con el objetivo de obtener *clusters* de clientes. Este método se ejecutará en R, y su código se encuentra en la sección de Anexos (Anexo 108). Al revisar el código, se puede ver que se tomó una muestra aleatoria

de 385 registros para realizar el dendrograma, ya que el proceso presenta dificultades al tomar la base de datos completa.

Cabe mencionar que se determinó la muestra aleatoria considerando una población de 1.300.000 clientes, un 95% de confianza, un valor sigma de 0,5 y un margen de error de 0,05.

Se ejecutó cinco veces el código en R, y los resultados obtenidos se encuentran en Anexos (Anexos 109 al 118).

Ahora, si analizamos los valores de SSE obtenidos para cada uno de los intentos, se puede concluir que el modelo que presentó el menor valor fue el **intento 1**, tal como se puede ver en la siguiente tabla:

Intentos	SSE
1	**9.308,521**
2	10.276,12
3	10.350,62
4	9.436,095
5	11.646,04

Tabla 6.6 Valores de SSE calculados de cada intento con el método de Ward (Fuente: Elaboración propia)

Por tanto, para esta base, se puede concluir que se pueden obtener **26 *clusters*** de clientes.

VI.1.4.4.3. DBSCAN

Como se ha mencionado anteriormente, una de las ventajas que presenta este algoritmo es que no es necesario definir previamente la cantidad de *clusters* que se pretenden obtener, sino que se tienen que definir dos parámetros: **épsilon** (ε) y los **puntos mínimos** (que corresponde a la cantidad de puntos mínimos que debe tener cada *cluster* dentro de un círculo de radio ε). En este caso, el indicador que debe ser optimizado es **ε**, ya que el algoritmo es más sensible a este parámetro.

Para obtener un valor adecuado de épsilon, se calcularán las distancias de los k – vecinos más cercanos, las que serán graficadas y se identificará el "codo" en el gráfico resultante (tal como en "*Elbow Method*"). Además, el valor de *k* utilizado será el parámetro utilizado en DBSCAN como puntos mínimos. Sin embargo, dependiendo del valor de *k* utilizado, se obtendrá un valor de épsilon diferente, y por ende, una cantidad de *clusters* distinta.

Producto de esto, se ejecutará este método varias veces, con distintos valores de *k*, con el objetivo de identificar el valor adecuado de este parámetro a ser utilizado y la cantidad final de *clusters* a obtener, bajo este algoritmo. Un gráfico de los valores de *k* utilizados para calcular las distancias de los k – vecinos más cercanos, y el épsilon calculado para cada *k* se muestra en Anexos (Anexo 119).

En ese gráfico, se puede ver que no es posible identificar un quiebre en el gráfico, por lo que no es posible identificar un valor de ε con un valor de *k* adecuados para el desarrollo de DBSCAN (tal como en *Elbow Method* aplicado a *K – Means*). Debido a esto, se ejecutó DBSCAN con cada uno de los valores de épsilon y *k* obtenidos anteriormente, cuyos resultados se muestran en Anexos (Anexo 120).

En este caso, es posible identificar un quiebre en el gráfico, cuando *k* = 4 y por ende, ε = 8.7. Ahora, un punto importante a considerar en la ejecución de DBSCAN es la cantidad de datos que son considerados como *outliers*, los que pueden variar dependiendo del valor de *k* utilizado. Si este valor es pequeño (como el que se mencionó anteriormente), entonces es probable que se esté considerado **una cantidad mayor de puntos atípicos** u *outliers*. Debido a esto, se debe considerar un valor de *k*, de tal forma que la cantidad de *outliers* no sea muy alta. Un gráfico con la cantidad de datos atípicos identificados por DBSCAN, por valor de *k* utilizado, se muestra en Anexos (Anexo 121). Además, al graficar los puntos atípicos con la cantidad de *clusters* obtenidos por DBSCAN en un solo gráfico, se obtiene lo que se encuentra en Anexos (Anexo 122).

Ahora, analizando el gráfico en Anexo 122, si bien el quiebre de *clusters* se produce cuando los valores de *k* están entre 3 y 6, se puede ver que el "codo" en el nuevo gráfico se produce cuando los valores de *k* se encuentran **entre 21 y 26**. Sin embargo, la cantidad de datos atípicos que son identificados en DBSCAN, cuando *k* es igual a cualquiera de los valores mencionados, es muy alta. Producto de esto, se requiere un valor de *k* tal que el resultado obtenido no tenga una gran cantidad de *outliers*. Luego, para el desarrollo de la tesis, se optará por ***k* = 24**, donde se obtienen 20 *clusters*, y 273 *outliers*. Cabe mencionar que, para realizar DBSCAN con k = 24, se asumió un valor de ε igual a 14.9, cuyo gráfico se encuentra en Anexos (Anexo 123).

Ahora, la representación gráfica de los *clusters* obtenidos se encuentra en Anexos (Anexo 124).

VI.1.4.4.4. *K – Medoids*

Este método fue ejecutado en R, mediante el algoritmo CLARA (*Clustering Large Applications*), y el código utilizado se encuentra en la sección de Anexos (Anexo 125). Ahora, tal como en el algoritmo *K – Means*, este método necesita que se define previamente la cantidad de *clusters* a obtener para realizar la segmentación. Debido a esto, se utilizará *Elbow Method*, con el objetivo de poder identificar la cantidad de *clusters* óptima para segmentar la base de clientes. Un gráfico de la cantidad de *clusters* con la medida de error, se encuentra en Anexos (Anexo 126).

En este caso, aunque el gráfico presente un comportamiento irregular, en comparación con los resultados de *K – Means*, se puede ver que el "codo" se produce cuando la cantidad de *clusters* definida en el algoritmo se encuentra entre siete y catorce. Debido a esto, se ejecutará el algoritmo *K – Medoids* con ***k* = 13**. Cabe mencionar que, tal como ocurre con *K – Means*, el primer intento no entrega necesariamente una buena solución, por lo que se realizarán varios intentos, y se utilizará el que tenga el menor error.

Después de ejecutar varias veces el algoritmo con ***k* = 13** (y variando la cantidad de registros como muestra que toma el algoritmo para su ejecución en R), se obtuvieron los resultados que se encuentran en Anexos (Anexo 127).

En este caso, se puede ver que el mejor resultado se obtiene en el **intento 6**, por lo que se utilizará ese resultado para el desarrollo de la tesis.

VI.1.4.5. Comparación de resultados

En esta sección, se procederá a comparar los distintos resultados obtenidos por los métodos desarrollados anteriormente, para identificar el método de clusterización que mejor clasifica a los clientes de la empresa de telecomunicaciones.

Los resultados obtenidos se encuentran en la siguiente tabla:

	K - Means	Método de Ward	DBSCAN	*K - Medoids*
Base con variables no binarias normalizadas entre 0 y 1	K = 16 SSE = 471.615,3	K = 29 SSE = 482.129,2	K = 11 SSE = 696.019,7	K = 12 SSE = 661.378,2
Base con variables no binarias estandarizadas	**K = 15** **SSE = 365.149,4**	K = 29 SSE = 407.637	K = 20 SSE = 602.785,98	K = 10 SSE = 561.641,6
Base con variables binarias y no binarias normalizadas entre 0 y 1	K = 13 SSE = 2.094.798	K = 30 SSE = 2.016.066	K = 5 SSE = 2.349.144,38	K = 7 SSE = 2.542.757
Base con variables binarias y no binarias estandarizadas	K = 22 SSE = 1.303.795	K = 26 SSE = 1.740.843	K = 24 SSE = 2.219.669,11	K = 13 SSE = 2.020.358

Tabla 6.7 Resultados de *clustering* (Fuente: Elaboración propia).

Tal como se puede ver en la tabla anterior, y considerando como criterio el valor del SSE, se puede concluir que el mejor resultado se obtuvo para el segundo tipo de análisis de la base, cuando se usa el método *K – Means*. Por tanto, se obtendrán **quince *clusters***, los que se describen en la siguiente tabla:

Clusters	Empaq.	Profesión	Rango etario	Sexo	GSE	Permanencia	Valor deuda	Días deuda	Ciclo de fact.	Flujo Cobr.	Tipo de boleta	Valor facturación	Zona
Cluster 1	Mono	Universitarios y Otros	Adulto "senior"	Masculino	Sector Alto	Mayor a 15 años	Entre $15.000 y $20.000	Menor a 5 días	Grande	Suave	Electrónica	Entre $45.000 y $50.000	Santiago
Cluster 2	Mono	Otros	Adulto joven	Femenino	Sector Medio	Menor a 5 años	Entre $15.000 y $20.000	Mayor a 15 días	Mediano	Duro	Electrónica	Entre $35.000 y $40.000	Santiago
Cluster 3	Mono	Otros	Adulto "senior"	Femenino	Sector Medio	Menor a 5 años	Entre $15.000 y $20.000	Mayor a 15 días	Mediano	Duro	Electrónica	Mayor a $50.000	Santiago
Cluster 4	Mono	Otros	Adulto mayor	Femenino	Sector Medio	Mayor a 15 años	Entre $15.000 y $20.000	Entre 5 y 10 días	Grande	Mediano	Electrónica	Entre $40.000 y $45.000	Santiago
Cluster 5	Mono	Universitarios	Adulto "senior"	Masculino	Sector Medio	Mayor a 15 años	Entre $15.000 y $20.000	Entre 10 y 15 días	Grande	Mediano	Electrónica	Entre $40.000 y $45.000	Santiago
Cluster 6	Mono	Universitarios y Otros	Adulto mayor	Masculino	Sector Medio	Mayor a 15 años	Entre $15.000 y $20.000	< 0 días	Grande	Mediano	Electrónica	Mayor a $50.000	Santiago
Cluster 7	Mono	Otros	Adulto "senior"	Masculino	Sector Medio	Mayor a 15 años	Mayor a $20.000	< 0 días	Grande	Mediano	Electrónica	Mayor a $50.000	Santiago
Cluster 8	Mono	Otros	Adulto "senior"	Femenino	Sector Medio	Menor a 5 años	Mayor a $20.000	Mayor a 15 días	Mediano	Duro	Electrónica	Entre $40.000 y $45.000	Santiago
Cluster 9	Mono	Universitarios	Adulto "senior"	Femenino	Sector Medio	Menor a 5 años	Entre $15.000 y $20.000	Mayor a 15 días	Mediano	Mediano	Electrónica	Entre $35.000 y $40.000	Santiago
Cluster 10	Mono	Otros	Adulto "senior"	Masculino	Sector Medio	Entre 10 y 15 años	Entre $15.000 y $20.000	< 0 días	Grande	Mediano	Electrónica	Entre $45.000 y $50.000	Santiago
Cluster 11	Mono	Otros	Adulto joven	Masculino	Sector Medio	Mayor a 15 años	Entre $15.000 y $20.000	Entre 10 y 15 días	Grande	Mediano	Electrónica	Mayor a $50.000	Santiago
Cluster 12	Mono	Universitarios y Otros	Adulto mayor	Masculino	Sector Medio	Mayor a 15 años	Mayor a $20.000	< 0 días	Grande	Mediano	Electrónica	Entre $45.000 y $50.000	Santiago
Cluster 13	Mono	Otros	Adulto "senior"	Masculino	Sector Medio	Mayor a 15 años	Entre $15.000 y $20.000	Menor a 5 días	Grande	Mediano	Electrónica	Mayor a $50.000	Santiago
Cluster 14	Mono	Universitarios y Otros	Adulto "senior"	Masculino	Sector Medio	Entre 10 y 15 años	Entre $15.000 y $20.000	Menor a 5 días	Grande	Mediano	Electrónica	Entre $45.000 y $50.000	Santiago
Cluster 15	Mono	Universitarios y Otros	Adulto "senior"	Masculino	Sector Alto	Mayor a 15 años	Mayor a $20.000	< 0 días	Grande	Mediano	Electrónica	Mayor a $50.000	Santiago

Tabla 6.8 Análisis de *clusters* (Fuente: Elaboración propia).

Como se puede ver en la tabla anterior, la interpretación y análisis de los *clusters* se realizó de forma categórica, y se realizó "examinando sus centroides, los cuales representan los valores promedio de los objetos contenidos en el conglomerado, en cada una de las variables" (Malhotra, 2008), en el caso de las variables cuantitativas (como la permanencia, el valor de la deuda, los días deuda y el valor de facturación). En el caso de las otras variables, se consideró el tipo de valor que más veces aparecían en los *clusters*. En la variable **Empaquetamiento**, se puede ver que en todos los *clusters* hay presencia de clientes Mono, dado que eran la cantidad de clientes que más se presentaban en cada *cluster*. En el caso de la variable **Profesión**, las categorías que se obtuvieron fueron,

principalmente, Universitarios y Otros, las que, tal como se mencionaron en la sección de Preprocesamiento de base de datos, consisten en clientes que poseen profesiones universitarias (abogados, arquitectos, médicos, etc.), o en clientes que no se encuentran especificadas sus profesiones en la base de datos.

En la variable **Sexo**, se puede ver que hay algunos *clusters* que poseen una alta presencia de hombres, como también hay otros que tienen una gran presencia de mujeres, y en la variable **GSE**, se puede ver que la mayoría de clientes se encuentran en sectores Altos y Medios (según la clasificación que se encuentra en la sección de Preprocesamiento de base de datos).

En el caso de la variable **Permanencia**, se puede ver que hay clientes que se encuentran en la compañía en diversos períodos de tiempo (menor a 5 años, entre 10 y 15 años y mayor a 15 años). Además, en la variable **Valor deuda**, se puede ver que los clientes presentan una deuda con la empresa, ya sea entre $15.000 y $20.000, o mayor a $20.000, mientras que en la variable **Días deuda**, se pueden identificar las siguientes categorías:

a) **< 0 días**: Esta categoría consiste en que la deuda que poseen los clientes que pertenecen al *cluster* aún no ha vencido.
b) **Menor a 5 días**: Son clientes que tienen una deuda ya vencida, y cuya antigüedad no supera los 5 días.
c) **Entre 5 y 10 días**: Son clientes que, tal como en la categoría anterior, tienen una deuda vencida, pero su antigüedad se encuentra entre 5 y 10 días.
d) **Entre 10 y 15 días**: Son clientes que poseen deuda vencida, y su antigüedad se encuentra entre 10 y 15 días.
e) **Mayor a 15 días**: Son clientes que poseen deuda vencida, con una antigüedad mayor a 15 días.

Cabe mencionar que si una deuda tiene una antigüedad mayor a 15 días, entonces la empresa aplicará acciones de cobranza, según su tipo de flujo, las que contemplan desde "cortar" sus servicios contratados, hasta retirar los equipos de la vivienda del cliente (siempre y cuando no pague su deuda).

Con respecto a la variable **Ciclo de facturación**, se puede ver que hay una gran presencia de ciclos del tipo Grande y Mediano (esta clasificación se realizó, de acuerdo a lo mencionado en la sección Preprocesamiento de base de datos), y en la variable **Flujo de Cobranza**, se puede ver que hay *clusters* con una gran cantidad de clientes con flujo de cobranza mediano y duro. La única excepción corresponde al *cluster* 1, el cual presenta una mayor cantidad de clientes con flujo de cobranza **suave**.

En la variable **Tipo de boleta**, se puede ver que todos los clientes presentan boletas de tipo electrónica, y en relación con la variable **Valor Facturación**, se puede ver que los clientes presentan una facturación desde $35.000. Finalmente, en la variable **Zona**, se puede ver que la mayoría de los clientes que pertenecen a esos *clusters* residen en Santiago.

VI.1.5. *Evaluation*

En esta etapa, se evaluará si los modelos que se utilizaron son los adecuados para obtener una solución al problema planteado, y a las necesidades del proyecto, establecidas en la etapa de *Business Understanding*.

Como se puede ver en el desarrollo anterior, lo que se obtuvo fue una segmentación de los clientes de la empresa de telecomunicaciones, utilizando variables demográficas y financieras, lo que permite tener un mejor entendimiento sobre su comportamiento, y así, poder entregar una atención "personalizada" a los clientes que pertenezcan a cada uno de los *clusters*.

Con respecto al *churn* de clientes, se puede calcular la tasa de *churn*, ya sea mora o voluntaria, asociada a cada *cluster*, lo que permitirá poder identificar los perfiles de clientes que presentan una mayor tasa de fuga, y así poder tomar políticas de retenciones proactivas a esos clientes (o estar más atentos a esos clientes, y evitar que se retiren de la compañía).

Las tasas de *churn* mora asociadas a cada *cluster* se muestran en la siguiente tabla:

	Churn Mora							
Cluster	**0 - 3 Meses**	**3 - 6 Meses**	**6 Meses a 1 año**	**1 a 1,5 años**	**1,5 a 2 años**	**2 a 2,5 años**	**2,5 a 3 años**	**Más de 3 años**
Cluster 1	1,36%	0,54%	0,26%	0,26%	0,17%	0,16%	0,10%	0,07%
Cluster 2	1,51%	0,58%	0,31%	0,28%	0,24%	0,20%	0,12%	0,09%
Cluster 3	1,43%	0,58%	0,29%	0,31%	0,18%	0,17%	0,13%	0,08%
Cluster 4	1,48%	0,52%	0,29%	0,26%	0,17%	0,19%	0,15%	0,08%
Cluster 5	1,46%	0,50%	0,29%	0,28%	0,18%	0,17%	0,13%	0,09%
Cluster 6	1,42%	0,54%	0,28%	0,22%	0,19%	0,17%	0,12%	0,07%
Cluster 7	1,36%	0,51%	0,29%	0,29%	0,17%	0,17%	0,10%	0,07%
Cluster 8	1,50%	0,50%	0,29%	0,30%	0,18%	0,14%	0,14%	0,08%
Cluster 9	1,36%	0,53%	0,29%	0,32%	0,16%	0,22%	0,11%	0,08%
Cluster 10	1,30%	0,50%	0,28%	0,28%	0,17%	0,18%	0,11%	0,08%
Cluster 11	1,37%	0,46%	0,28%	0,28%	0,18%	0,16%	0,12%	0,08%
Cluster 12	1,42%	0,51%	0,27%	0,27%	0,13%	0,18%	0,11%	0,07%
Cluster 13	1,48%	0,53%	0,28%	0,22%	0,15%	0,15%	0,12%	0,08%
Cluster 14	1,31%	0,50%	0,27%	0,27%	0,16%	0,18%	0,09%	0,07%
Cluster 15	2,57%	0,95%	0,50%	0,51%	0,32%	0,30%	0,17%	0,13%

Tabla 6.9 Tasas de *churn* mora por *cluster* (Fuente: Empresa de telecomunicaciones).

Para obtener las tasas de fuga por Mora que se muestran en la tabla anterior, se utilizaron los resultados de *churn* mora entregados por la compañía de telecomunicaciones en Junio del 2019. Ahora, como se puede ver en la tabla anterior, las tasas de fuga más altas se centran en los clientes, cuya permanencia está entre 0 y 3 meses, lo que coincide con lo presentado en la sección "Identificación de necesidades y oportunidades", en el capítulo 1.

Al analizar los valores de estas tasas, se puede ver que la tasa de fuga por mora de los clientes con permanencia hasta 3 meses en la compañía (o clientes nuevos) en un *cluster* de clientes es de, al menos, un 1,3%, lo que es muy grave, considerando que la tasa de fuga de los otros clientes es cercano (o menor) a 0,5%, exceptuando los clientes que tienen una permanencia en la compañía entre 3 y 6 meses, y que pertenezcan al *cluster* 15.

Ahora, al obtener las tasas de *churn* voluntario para cada *cluster*, se obtiene lo siguiente:

	Churn Voluntaria							
Cluster	**0 - 3 Meses**	**3 - 6 Meses**	**6 Meses a 1 año**	**1 a 1,5 años**	**1,5 a 2 años**	**2 a 2,5 años**	**2,5 a 3 años**	**Más de 3 años**
Cluster 1	0,34%	0,17%	0,11%	0,11%	0,06%	0,10%	0,07%	0,06%
Cluster 2	0,32%	0,24%	0,12%	0,12%	0,07%	0,12%	0,06%	0,06%
Cluster 3	0,27%	0,20%	0,11%	0,13%	0,07%	0,09%	0,09%	0,06%
Cluster 4	0,31%	0,16%	0,13%	0,12%	0,07%	0,09%	0,06%	0,06%
Cluster 5	0,32%	0,20%	0,11%	0,11%	0,07%	0,08%	0,08%	0,07%
Cluster 6	0,32%	0,20%	0,09%	0,11%	0,08%	0,07%	0,06%	0,06%
Cluster 7	0,31%	0,19%	0,11%	0,11%	0,09%	0,09%	0,06%	0,06%
Cluster 8	0,31%	0,17%	0,10%	0,10%	0,09%	0,09%	0,05%	0,06%
Cluster 9	0,28%	0,18%	0,11%	0,09%	0,08%	0,09%	0,09%	0,06%
Cluster 10	0,26%	0,17%	0,11%	0,09%	0,08%	0,11%	0,09%	0,05%
Cluster 11	0,32%	0,20%	0,09%	0,10%	0,09%	0,09%	0,07%	0,07%
Cluster 12	0,31%	0,16%	0,11%	0,15%	0,08%	0,11%	0,05%	0,06%
Cluster 13	0,30%	0,21%	0,12%	0,12%	0,08%	0,11%	0,10%	0,07%
Cluster 14	0,33%	0,20%	0,11%	0,11%	0,07%	0,07%	0,07%	0,06%
Cluster 15	0,36%	0,18%	0,11%	0,11%	0,08%	0,09%	0,10%	0,06%

Tabla 6.10 Tasas de *churn* voluntario por *cluster* (Fuente: Empresa de telecomunicaciones).

En este caso, a diferencia de las tasas de *churn* mora, las tasas de fuga voluntaria son inferiores al 1%, pero los mayores porcentajes se centran en los clientes nuevos. Debido a esto, es de suma importancia tomar medidas de forma proactiva para que los clientes que pertenezcan a los *clusters* que posean una alta tasa de fuga (tanto mora como voluntaria) y cuya permanencia sea menor a tres meses, no se retiren de la empresa.

Con estos resultados, es posible inferir qué tipo de clientes tiene una alta probabilidad de fuga, ya que, tal como se mencionó anteriormente, la compañía de telecomunicaciones debiera centrarse en aquellos *clusters* que presenten una alta tasa de *churn* mora y voluntaria, y así, cada vez que un cliente nuevo ingrese a la compañía, poder categorizarlo en alguno de los segmentos, y tomar las precauciones necesarias para evitar que se retire de la compañía.

Si consideramos la tasa de *churn* mora, con un límite mínimo de 1,5% (este límite fue definido previamente por el negocio), entonces los *clusters* que contiene clientes con una alta probabilidad de fuga son los siguientes: ***cluster* 2**, ***cluster* 8** y ***cluster* 15**.

Ahora, si consideramos la tasa de *churn* voluntaria, con un límite mínimo de 0,3% (tal como el límite anterior, fue establecido previamente por el negocio), entonces, a excepción de los *clusters* 3, 9 y 10, los demás segmentos corresponden a clientes con una alta tasa de fuga. Luego, al hacer el cruce con los *clusters* que presentan una alta tasa de fuga por mora, o involuntaria, entonces se puede decir que los *clusters* que presentan clientes con una mayor probabilidad de fuga son los ***clusters* 2, 8 y 15**.

Sin embargo, esta solución no responde la hipótesis planteada en la Introducción, por lo que se realizarán modelos de predicción de fuga de clientes, utilizando variables financieras y operacionales, pero solamente con los clientes pertenecientes a los *clusters* mencionados anteriormente.

VI.1.6. *Deployment*

Para implementar este modelo de segmentación, se deberá instalar el programa RStudio, con el objetivo de poder realizar la segmentación de los clientes en un cierto período de tiempo, con el objetivo de actualizar los perfiles de los *clusters* que se obtengan de los clientes.

Sobre la implementación de los resultados obtenidos, cabe mencionar que los perfiles del *cluster* al que pertenece cada cliente se mostrará en el programa utilizado por la compañía de telecomunicaciones para realizar gestión a los clientes, por medio de *scripts*, en donde se especificará si es de alto riesgo de fuga, o no. Producto de esto, no será necesario instalar un *software* adicional para cumplir con esta función.

VI.2. Aplicación de CRISP – DM para modelos de predicción

En esta sección, se procederá a aplicar la metodología CRISP – DM para obtener modelos de predicción asociados a los *clusters* que presentan una alta tasa de *churn* de clientes, cuya permanencia sea menor a 3 meses, utilizando variables financieras y operacionales relacionadas con el funcionamiento de los servicios contratados por los clientes, con el objetivo de poder predecir si un cliente se fugará de la compañía o no. Ahora, se describirán los resultados obtenidos para cada una de las etapas de la metodología, asociadas a la predicción de *churn* de clientes.

VI.2.1. *Business Understanding*

En esta etapa, se procederá a realizar una comprensión de los objetivos y requerimientos asociados a la creación de un modelo de predicción de fuga para los clientes de la empresa de telecomunicaciones, que presenten una alta tasa de *churn*, ya sea mora y/o voluntario, desde un punto de vista empresarial.

Ahora, tal como se ha mencionado anteriormente en este trabajo, el principal problema de la compañía consiste en la cantidad de desconexiones que se producen de forma mensual, ya sea por motivos involuntarios (debido, principalmente, al no pago de la deuda con la empresa) o voluntarios (inconformidad con la calidad del servicio, muerte de titular de la cuenta, entre otros), por lo que es de suma importancia desarrollar un modelo de predicción de *churn* que permita identificar a aquellos clientes que tienen mayor probabilidad de fugarse de la compañía. Una vez desarrollado este modelo, se podrán tomar acciones proactivas que permitirán evitar que el cliente desconecte sus servicios.

La creación de los modelos de predicción de fuga de clientes permitirá a la compañía poder realizar gestión “personalizada” a los clientes que cumplan con alguno de los

perfiles de segmentos que tengan una alta tasa de *churn* mora y/o voluntaria, y así poder optimizar recursos para la retención de esos clientes.

Para determinar o predecir si un cliente se fugará de la compañía, se identificarán variables de tipo financiera y operacional, que permita describir a los clientes de la forma más precisa posible, con el objetivo de obtener un perfil que permita identificar si un cliente es propenso a retirarse de la compañía o no.

VI.2.2. *Data Understanding*

Cabe mencionar que las variables a utilizar para el desarrollo del modelo de predicción de fuga de clientes se encuentran en la sección "Descripción del problema".

Sin embargo, a pesar de lo anterior, cabe recordar que, para solucionar lo planteado en la hipótesis, se realizarán tres modelos de predicción para cada uno de los *clusters* que se identificaron en la sección anterior. Uno de los modelos estará diseñado exclusivamente con variables financieras; el segundo se diseñará solamente con variables operacionales; y finalmente, el tercer modelo se desarrollará con una combinación de variables financieras y operacionales. El objetivo de la realización de estos modelos es poder identificar qué variables permiten predecir de mejor manera los clientes que se fugarán de la compañía.

Las variables financieras que se utilizarán para realizar el modelo de predicción de fuga de clientes serán las siguientes:

- **Facturación**: Corresponde al valor que se factura al cliente, producto de los servicios que contrató con la compañía de telecomunicaciones. Cabe mencionar que, a medida que un cliente aumente la cantidad de servicios contratados, entonces su valor a facturar aumentará. De igual forma, si el cliente desconecta alguno de sus servicios, entonces el valor de su facturación disminuirá. En el caso de este trabajo, se utilizará el valor facturado al cliente en los últimos tres meses.
- **Deuda**: Consiste en el monto adeudado por un cliente con la compañía. Esta variable se calculará, tomando en consideración los tres últimos meses.
- **GSE**: Corresponde a una variable binaria que muestra el grupo socioeconómico al que pertenece el sector donde reside el cliente. Para esta variable, se identificaron tres

tipos de sectores: sector alto (consiste en aquellas localidades pertenecientes al grupo ABC1), sector medio (consiste en aquellas localidades pertenecientes a los grupos C2 y C3) y sector bajo (son los sectores pertenecientes a los GSE D y E).

- **Flujo de cobranzas**: Esta variable, de tipo categórica, permitirá identificar a los clientes que se encuentre identificado con alguno de los tipos de flujo de cobranza, que existen actualmente en la empresa de telecomunicaciones. En este caso, solo existen tres tipos: flujo de cobranza suave (este flujo está presente, principalmente, en clientes que tengan PAC contratado, o en clientes VIP), flujo de cobranza mediano (este flujo está presente en clientes que han tenido un buen comportamiento de pago, y que han estado activos por un considerable período de tiempo, de forma ininterrumpida) y flujo de cobranza duro (en este flujo, se encuentran todos los clientes nuevos de la empresa de telecomunicaciones, y aquellos clientes que no tienen un buen comportamiento de pago).

Ahora, las variables operacionales que se utilizarán para el desarrollo de los modelos de predicción de fuga de clientes son:

- ***Truck Rolls***: Esta variable binaria corresponde a si se finalizó, al menos, un servicio técnico realizado en la vivienda del cliente. Para calcular esta variable, se usarán los datos de los últimos tres meses, para obtener un acumulado de la cantidad de servicios técnicos ejecutados en la vivienda del cliente.
- **Tiempo de respuesta**: Corresponde a la cantidad de días que la compañía ha demorado para resolver algún problema técnico a un cliente. Tal como en la variable *Truck Rolls*, se utilizarán los datos de los últimos tres meses para estimar el valor de esta variable, el cual será utilizado para predecir a clientes que se fugarán de la compañía. Si un cliente presentó más de 1 *truck roll* en el mes, entonces se calculará el promedio de tiempo de respuesta.
- **Fallas masivas en el sector**: Corresponderá a la cantidad de fallas masivas que han ocurrido en el sector donde reside el cliente, en los últimos tres meses.
- **Reiteraciones de servicios técnicos**: Es una variable binaria que se encargará de identificar si un cliente ha presentado servicios técnicos reiterados, o no.

Cabe mencionar que las variables financieras mencionadas anteriormente se pueden obtener mediante una consulta a la base de datos de la compañía. Sin embargo, las variables operacionales se obtienen a través de reportes generados por la gerencia de Operaciones de la empresa de telecomunicaciones, por lo que se requiere procesar la información contenida en los archivos, para obtener los valores de esas variables, y así, poder diseñar un modelo que permita predecir la fuga de clientes de la compañía.

VI.2.3. *Data Preparation*

En esta sección, se procederá a describir la construcción de las bases de datos a utilizar para el desarrollo del modelo de predicción de fuga de clientes. Tal como se mencionó en la sección anterior, se desarrollarán tres modelos para predecir *churn* de clientes, siendo uno conformado solamente por variables financieras; el otro modelo, por variables operacionales; y finalmente, un modelo que contenga ambos tipos de variables.

En el caso de la base de datos asociada al primer modelo, cabe destacar que esas variables fueron utilizadas para realizar la segmentación de los clientes de la compañía de telecomunicaciones. Debido a esto, se utilizará la misma información de esas variables para realizar el modelo de predicción (es decir, se usarán las variables que fueron estandarizadas para el desarrollo del modelo).

Con respecto a las variables operacionales, se destaca que los datos utilizados corresponden a los últimos tres meses cerrados que se tengan registros (en este caso, los meses serían Abril, Mayo y Junio del 2019). Además, en el caso de las variables **tiempo de respuesta** y **fallas masivas en el sector**, como ambas variables tienen unidades de medida distintas al resto de las variables utilizadas en este modelo, entonces se procederá a estandarizarlas con puntaje Z para predecir la fuga de clientes.

VI.2.4. *Modeling*

En esta sección, se procederá a desarrollar los modelos de predicción de *churn*, mediante el uso de variables financieras, variables operacionales y una combinación de ambos.

VI.2.4.1. Modelo de predicción de fuga de clientes con variables financieras

Para este modelo de predicción, se utilizarán tres técnicas que permitirán definir aquellos clientes con mayor probabilidad de fuga de la empresa de telecomunicaciones, y los que no presentan este comportamiento. Las técnicas que se utilizaron fueron: **modelo de naïve – Bayes**, **regresión logística** y **árbol de decisión**. Los motivos por los que se seleccionaron estos modelos para desarrollar el modelo de predicción son, principalmente, porque ambos entregan un resultado de tipo probabilístico para determinar si un cliente es un posible fugado o no, y también porque entregan una clara comprensión acerca de las variables utilizadas para el desarrollo del modelo de predicción.

Junto con los modelos anteriores, se procedió a obtener conjuntos de entrenamiento (70% de la base total) y de prueba (30% restante de la base total). Para esto, se utilizaron cinco técnicas que permitieron obtener distintos conjuntos de entrenamiento para el modelo de predicción. Estas técnicas son:

a) Conjunto desbalanceado: Corresponde a la técnica tradicional de generación de conjunto de entrenamiento. Aquí, se obtienen registros aleatorios de la base total para formar una base de entrenamiento, con el objetivo de, como su nombre lo indica, "entrenar" el modelo de predicción.
b) *Downsample*: Consiste en la generación de un conjunto de entrenamiento, privilegiando la clase que tenga menor proporción.
c) *Upsample*: Consiste en la generación de un conjunto de entrenamiento, pero priviliegiando la clase con mayor proporción de clientes.
d) SMOTE con variables por defecto en R: Consiste en una técnica que se utiliza para generar nuevos ejemplos artificiales de la clase minoritaria, usando los vecinos más cercanos.
e) SMOTE modificado: Es la misma técnica anterior, pero con una leve modificación, la que consiste en una disminución del sobremuestreo de la clase minoritaria.

Estos conjuntos de entrenamiento y de prueba serán aleatorios, y el proceso se ejecutará 30 veces para cada *cluster* seleccionado en el proceso de segmentación de clientes.

Los resultados obtenidos para los *clusters* 2, 8 y 15, para cada uno de los modelos de predicción descritos anteriormente se encuentran en Anexos (Anexo 128 al 136). Cabe mencionar que para seleccionar el mejor modelo de predicción, se consideró el valor del AUC.

En el caso del modelo de predicción utilizando variables financieras, se puede ver que, para el *cluster* 2, el mejor modelo corresponde al de **regresión logística**, donde el conjunto de entrenamiento se construye con SMOTE, con las variables por defecto, utilizadas en el *software* R. Para el *cluster* 8, se puede ver que el mejor modelo corresponde a **Naïve – Bayes**, donde el conjunto de entrenamiento se obtiene a través del método *Upsample*. Para el *cluster* 15, el mejor modelo corresponde al **árbol de decisión**, donde el conjunto de entrenamiento es desbalanceado.

En el caso del modelo de predicción con variables operacionales, para el *cluster* 2, se puede ver que, tal como en el modelo anterior, los mejores resultados se obtienen con **regresión logística**, y el conjunto de entrenamiento se obtiene con SMOTE, sin modificar las variables que entrega el *software* R.

VII. DISCUSIÓN

En esta sección, se procederá a discutir los resultados obtenidos en la sección anterior, y su importancia al respecto.

Tal como se puede apreciar en los resultados obtenidos, al procesar la base de clientes de la empresa de telecomunicaciones utilizando variables binarias (para los datos categóricos), y procesando solamente los datos continuos mediante estandarización con puntaje Z, o normalizándolos entre 0 y 1, se obtiene una cantidad de *clusters* de clientes que permiten una correcta interpretación, y también con poca probabilidad de unirlos (dado que al haber una gran cantidad de *clusters*, pueden existir algunos que tengan las mismas características, por lo que se deben fusionar), y también con un valor no muy elevado de error, dada la gran cantidad de variables que se procesaron para obtener los segmentos (producto del uso de variables binarias). La gran excepción en este caso sería la aplicación del método de Ward a estas bases, dado que se obtuvieron alrededor de 30 segmentos, por lo que es probable que algunos de ellos posean las mismas características, y por ende, se debieran fusionar.

Ahora, cuando se aplicó la estandarización con puntaje Z y la normalización entre 0 y 1 a las variables binarias, en conjunto con las variables continuas, se logró observar que, si bien se obtiene una cantidad adecuada de *clusters* (en relación con lo que una compañía puede manejar), no se obtiene un valor adecuado del error de clusterización, ya que este valor se encuentra cercano a los dos millones, por lo que se puede concluir que no se realizó una buena segmentación de los clientes, aún usando las mismas técnicas de *clustering* que en las bases que se describieron anteriormente (*K – Means*, método de Ward, DBSCAN y *K – Medoids*), y obteniendo una cantidad similar de *clusters*.

Además, en este trabajo, se compararon los distintos resultados obtenidos de segmentación de clientes, utilizando distintas técnicas de clusterización y de tratamiento de la base, con el objetivo de obtener la mejor solución al problema planteado. Junto con lo anterior, se calculó las tasas de *churn* mora y voluntaria, asociadas a cada *cluster*, lo que permite poder identificar a aquellos clientes que pueden tener una alta probabilidad de fuga. Esto ayudará a la empresa a poder identificar perfiles de clientes

que, en caso de que ingrese un cliente nuevo, saber identificarlo y etiquetarlo como un posible *churn*, y así, tomar las medidas necesarias para evitar que ese cliente se desconecte de la compañía.

Esto se puede comparar con lo presentado por Rajamohamed et al. (2017), quien clusterizó los usuarios de una tarjeta de crédito, utilizando *K – Means*, y con los resultados obtenidos, aplicó diversas técnicas de predicción, como SVM, árboles de decisión, etc., para poder predecir la fuga de clientes. Sin embargo, en este proyecto, no solamente se segmentó a los clientes utilizando *K – Means*, sino que también se utilizaron otras tres técnicas, por lo que es probable que si se hubieran usado más algoritmos de clusterización en la base de usuarios de tarjetas de crédito, se podría obtener un mejor resultado. Además, en Rajamohamed et al. (2017), no se calcularon las tasas de fuga de clientes asociadas a cada *cluster*, por lo que es probable que si se hubiera hecho, se obtendría una mejor predicción de fuga de clientes, enfocándose en solamente los clientes que tengan un perfil que indique una alta probabilidad de fuga.

Otro ejemplo de aplicación de segmentación de clientes para predecir la fuga de clientes se encuentra en Dulhare et al. (2018), quien tal como se comentó en “Estado del Arte”, propuso un modelo híbrido para predecir la fuga de clientes. Aunque para el desarrollo de la tesis se utilizó el método DBSCAN para segmentar a los clientes de la compañía de telecomunicaciones, no se obtuvo un buen resultado al aplicar este método, a diferencia de *K – Means*, con el que se obtuvo un buen resultado.

VIII. CONCLUSIONES

En esta sección se debe presentar de manera concisa y clara las conclusiones del trabajo realizado (extensión máxima 1 página).

IX. PROYECCIONES

En esta sección se deben identificar y discutir las posibles aplicaciones de los resultados o metodologías utilizadas a otras problemáticas. También se deben discutir aquellos aspectos que podrían requerirse para ampliar el campo de acción de la solución propuesta y se debe realizar un análisis crítico de los aspectos mejorables del proyecto (extensión máxima 2 páginas)

X. BIBLIOGRAFÍA

1. Óskarsdóttir, M.; Van Calster, T.; Baesens, B.; Lemahieu, W. y J. Vanthienen, (2018) *Time series for early churn detection: Using similarity based classification for dynamic networks*, en *Expert Systems With Applications*, 106, 55 – 65.
2. Amin, A.; Al – Obeidat, F.; Shah, B.; Adnan, A.; Loo, J. y S. Anwar, (2018) *Customer churn prediction in telecommunication industry using data certainty*, en *Journal of Business Research*. doi: 10.1016/j.jbusres.2018.03.003.
3. Contreras, E.; Ferreira, F. y M. Valle, (2017) *Design of a Predictive Customer Leakage Model using Decision Trees*, en Revista Ingeniería Industrial – Año 16, N°1, 7 – 23.
4. De Caigny, A.; Coussement, K. y K. de Bock, (2018) *A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees*, en *European Journal of Operational Research*, 269, 760 – 772.
5. Chen, S., (2016) *The gamma CUSUM chart method for online customer churn prediction*, en *Electronic Commerce Research and Applications*, 17, 99 – 111.
6. Milošević, M.; Živić, N. e I. Andjelković, (2017) *Early churn prediction with personalized targeting in mobile social games*, en *Expert Systems with Applications*, 83, 326 – 332.
7. Coussement, K.; Lessmann, S. y G. Verstraeten, (2017) *A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry*, en *Decision Support Systems*, 95, 27 – 36.
8. Gordini, N. y V. Veglio, (2017) *Customers churn prediction and marketing retention strategies. An application of support vector machines based on the AUC parameter – selection technique in B2B e – commerce industry*, en *Industrial Marketing Management*, 62, 100 – 107.
9. Ammar, A.; Ahmed, Q y D. Maheswari, (2017) *Churn prediction on churn telecom data using hybrid firefly based classification*, en *Egyptian Informatics Journal*, 18, 215 – 220.
10. Ahmed, M.; Afzal, H.; Siddiqi, I.; Amjad, M. y K. Khurshid, (2018) *Exploring nested ensemble learners using overproduction and choose approach for churn prediction in telecom industry*, en *Neural Computing and Applications*. doi: 10.1007/s00521-018-3678-8

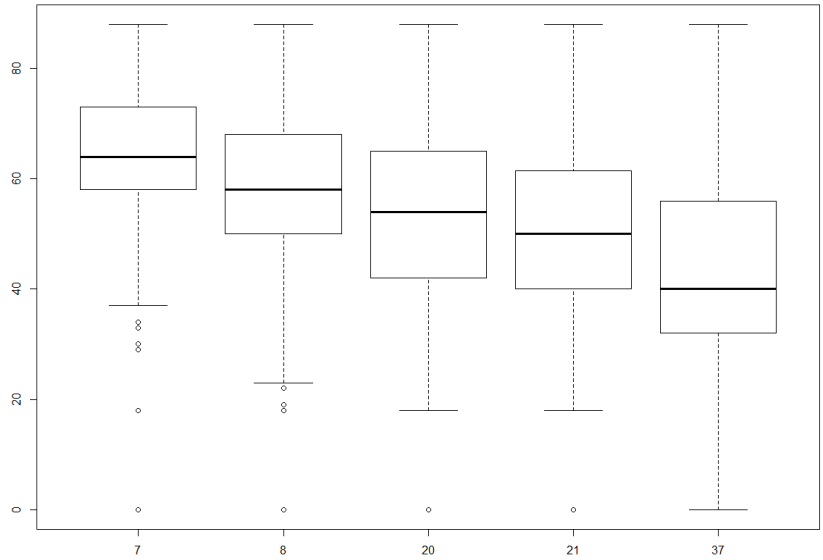
11. Keramati, A.; Ghaneei, H. y S. Mirmohammadi, (2016) *Developing a prediction model for customer churn from electronic banking services using data mining*, en *Financial Innovation*, 2 – 10.
12. Yu, R.; An, X.; Jin, B.; Shi, J.; Move, O. y Y. Liu, (2018) *Particle classification optimization – based BP network for telecommunication customer churn prediction*, en *Neural Computing and Applications*, 29, 707 – 720.
13. Sivasankar, E. y J.Vijaya, (2018) *Hybrid PPFCM – ANN model: an efficient system for customer churn prediction through probabilistic possibilistic fuzzy clustering and artificial neural network*, en *Neural Computing and Applications*. doi: 10.1007/s00521-018-3548-4.
14. Vijaya, J. y E. Sivasankar, (2017) *An efficient system for customer churn prediction through particle swarm optimization based feature selection model with simulated annealing*, en *Cluster Computing*. doi: 10.1007/s10586-017-1172-1.
15. Idris, A.; Iftikhar, A. y Z. Rehman, (2017) *Intelligent churn prediction for telecom using GP – AdaBoost learning and PSO undersampling*, en *Cluster Computing*. doi: 10.1007/s10586-017-1154-3.
16. Rajamohamed, R. y J. Manokaran, (2017) *Improved credit card churn prediction based on rough clustering and supervised learning techniques*, en *Cluster Computing*. doi: 10.1007/s10586-017-0933-1.
17. Azeem, M.; Usman, M. y A. Fong, (2017) *A churn prediction model for prepaid customers in telecom using fuzzy classifiers*, en *Telecommunication Systems*, 66, 603 – 614.
18. Moser, S.; Schumann, J.H.; von Wangenheim, F.; Uhrich, F. y F. Frank, (2018) *The Effect of a Service Provider's Competitive Market Position on Churn Among Flat – Rate Customers*, en *Journal of Service Research*, 21, 319 – 335.
19. Rachid, A.D.; Abdellah, A.; Belaid, B. y L. Rachid, (2018) *Clustering prediction techniques in defining and predicting customers defection: The case of e – commerce context*, en *International Journal of Electrical and Computer Engineering*, 8, 2367 – 2383.

20. Günay, M. y T. Ensari, (2018) *Predictive churn analysis with machine learning methods (Makine öğrenmesi yöntemleri ile kayip müşteri analizi)*, en *26th IEEE Signal Processing and Communications Applications Conference*, SIU 2018, 1 – 4.
21. Dulhare, U.N. e I. Ghori, (2018) *An efficient hybrid clustering to predict the risk of customer churn*, en *Proceedings of the 2nd International Conference on Inventive Systems and Control*, ICISC 2018, 673 – 677.
22. Patil, A.P.; Deepshika, M.P.; Mittal, S.; Shetty, S.; Hiremath, S.S. y Y.E. Patil, (2018) *Customer churn prediction for retail business*, en *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, ICECDS 2017, 845 – 851.
23. Mishra, K. y R. Rani, (2018) *Churn prediction in telecommunication using machine learning*, en *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing*, ICECDS 2017, 2252 – 2257.
24. Hartati, E.P.; Adiwijaya, y M.A. Bijaksana, (2018) *Handling imbalance data in churn prediction using combined SMOTE and RUS with bagging method*, en *Journal of Physics: Conference Series*, 971, artículo número 012007.
25. Awang, M.K.; Ismail, M.R.; Makhtar, M.; Nordin A. Rahman, M., y A.R. Mamat, (2018) *Performance comparison of neural network training algorithms for modeling customer churn prediction*, en *International Journal of Engineering and Technology*, 7, 35 – 37.
26. Chen, Y. y M. Schwartz, (2016) *Churn Versus Diversion in Antitrust: An Illustrative Model*, en *Economica*, 83, 564 – 583.
27. Stripling, E.; van den Broucke, S.; Antonio, K.; Baesens, B. y M. Snoeck, (2018) *Profit maximizing logistic model for customer churn prediction using genetic algorithms*, en *Swarm and Evolutionary Computation*, 40, 116 – 130.
28. Culbert, B.; Fu, B.; Brownlow, J.; Chu, C.; Meng, Q. y G. Xu, (2018) *Customer Churn Prediction in Superannuation: A sequential pattern mining approach*, en *Australasian Database Conference 2018: Databases Theory and Applications*, 123 – 134.
29. Mohanty, R. y C.N.R. Sree, (2018) *Churn and Non – churn of Customers in Banking Sector using Extreme Learning Machine*, en *Proceedings of the Second International Conference on Computational Intelligence and Informatics*, 51 – 58.

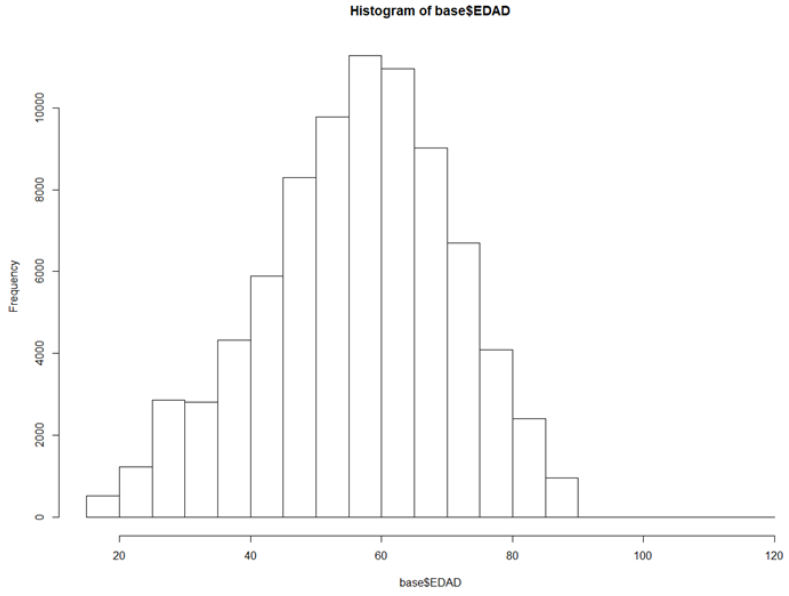
30. Huang, J. y L. He, (2018) *Application of Improved PSO - BP Neural Network in Customer Churn Warning*, en *Procedia Computer Science*, 131, 1238 – 1246.
31. AIM, (2008) *Grupos Socioeconómicos 2008*.
32. https://www.slideshare.net/MohanBavirisetty/advanced-analytics-frameworks-platforms-and-metholodologies-v-10
33. http://disi.unal.edu.co/~eleonguz/cursos/md/presentaciones/Sesion5_Metodologias.pdf
34. Han, J. y M. Kamber, (2001) *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, USA.
35. Malhotra, N., (2008) *Investigación de Mercados*, Quinta edición, Pearson Education.
36. http://aathosc.tripod.com/PuntajeZ22.htm
37. Tan, P.; Steinbach, M. y V. Kumar, (2006) *Introduction to Data Mining*, Pearson Education.

XI. ANEXOS

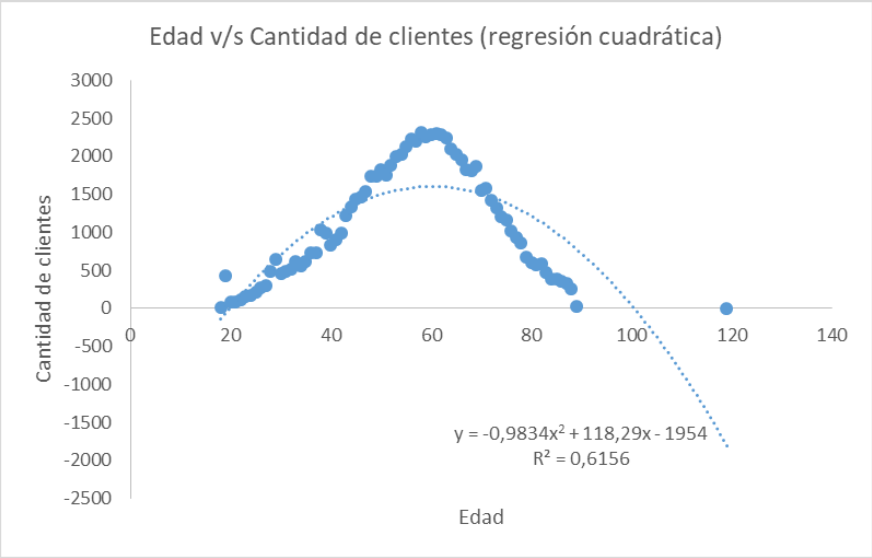
Anexo 1. *Boxplots* de las cinco profesiones con mayor cantidad de clientes (Fuente: Empresa de telecomunicaciones).



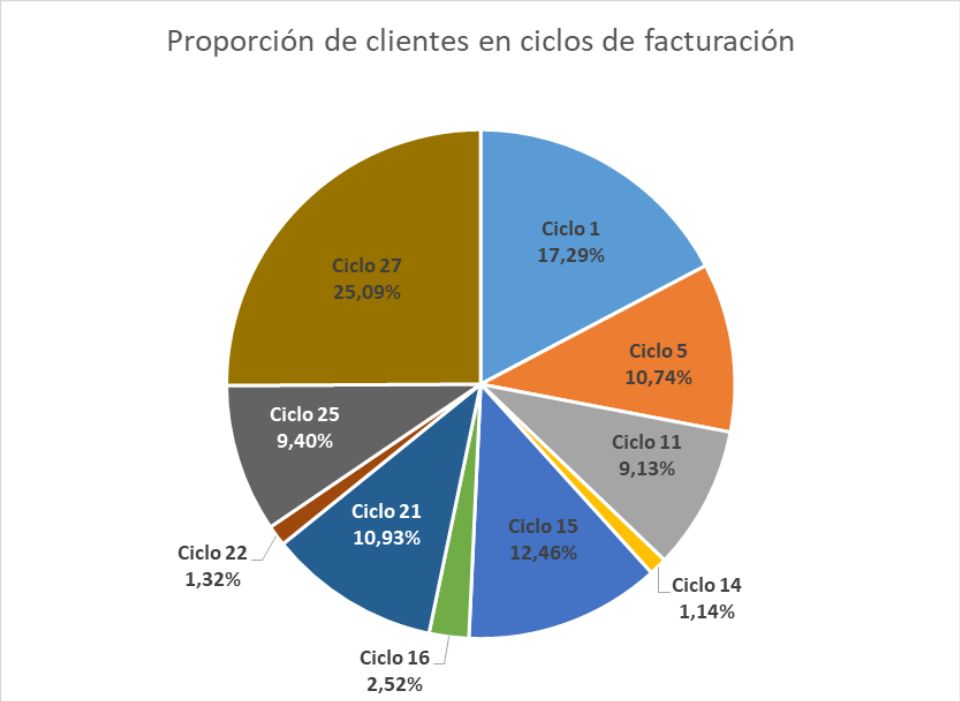
Anexo 2. Histograma de la variable EDAD (Fuente: Elaboración propia).



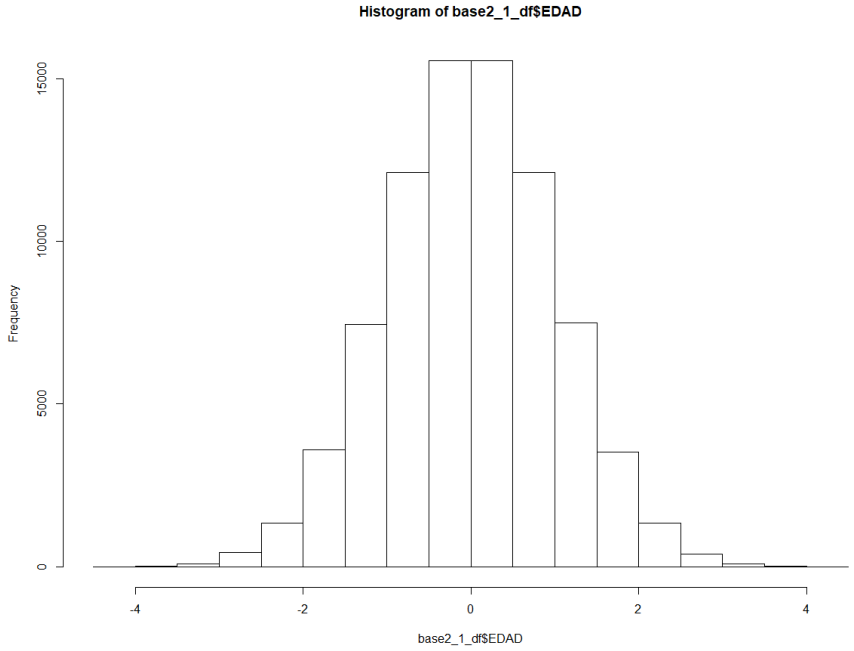
Anexo 3. Gráfico de dispersión con línea de tendencia polinómica de grado 2 (Fuente: Elaboración propia).



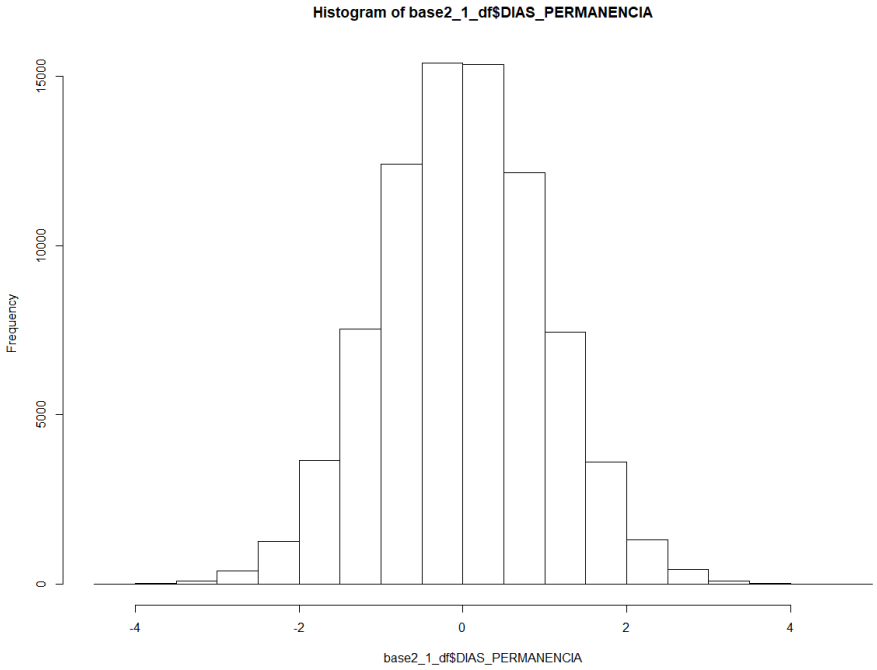
Anexo 4. Proporción de clientes que se encuentran en cada ciclo de facturación (Fuente: Empresa de telecomunicaciones).



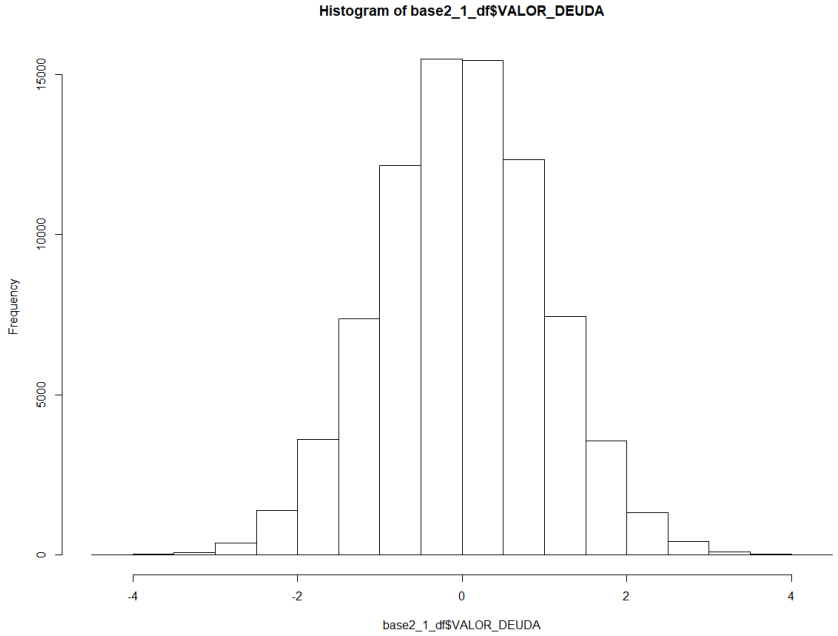
Anexo 5. Histograma de variable normalizada EDAD (Fuente: Elaboración propia).



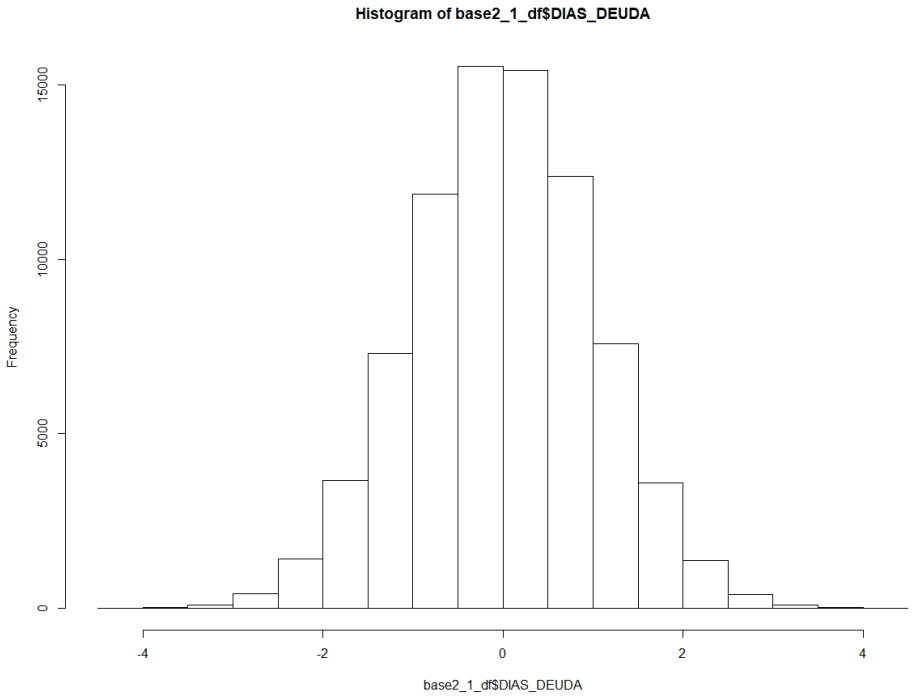
Anexo 6. Histograma de variable normalizada DIAS_PERMANENCIA (Fuente: Elaboración propia).



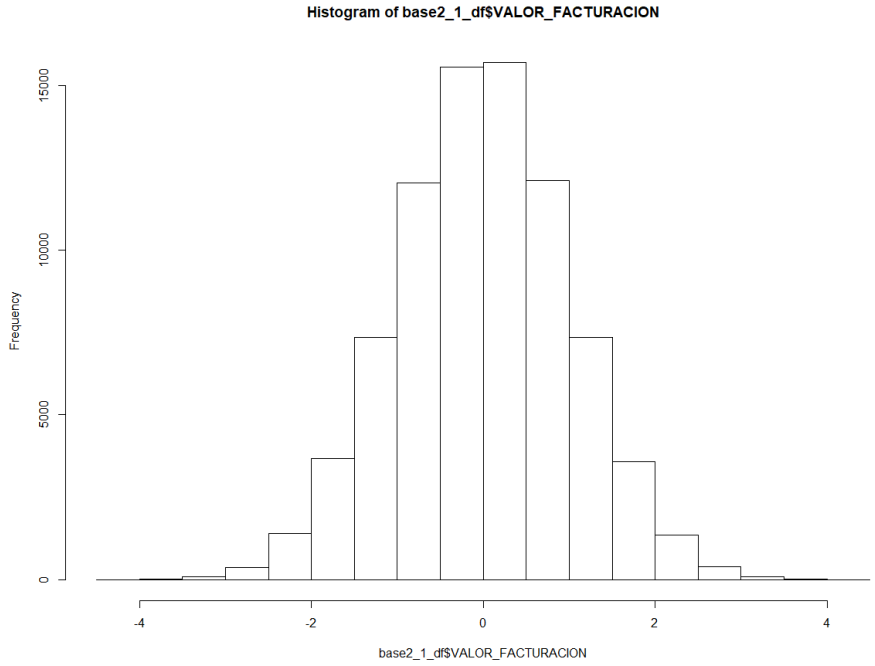
Anexo 7. Histograma de variable normalizada VALOR_DEUDA (Fuente: Elaboración propia).



Anexo 8. Histograma de variable normalizada DIAS_DEUDA (Fuente: Elaboración propia).



Anexo 9. Histograma de variable normalizada VALOR_FACTURACION (Fuente: Elaboración propia).



Anexo 10. Código utilizado para generar base con variables no binarias normalizadas.

```
base2_1 <- cbind(base$RUT_VIVIENDA, base$Svc_Mono, base$Svc_Duo, base$Svc_Triple, base$PROF_1, base$PROF_2, base$PROF_3, base$PROF_4, base$PROF_5, rnorm(base$EDAD), base$FLAG_VIP, base$SEXO, var_sector_alto, var_sector_medio, var_sector_bajo, base$GSE_COM, rnorm(base$DIAS_PERMANENCIA), rnorm(base$VALOR_DEUDA), rnorm(base$DIAS_DEUDA), var_ciclo_pequeño, var_ciclo_mediano, var_ciclo_grande, base$FLUJO_SUAVE, base$FLUJO_MEDIANO, base$FLUJO_DURO, base$BOLETA_ELECT, rnorm(base$VALOR_FACTURACION), base$ZONA_NORTE, base$ZONA_CENTRO, base$ZONA_SUR, var_zona_stgo)
class(base2_1)
base2_1_df <- data.frame(base2_1)
class(base2_1_df)
colnames(base2_1_df) <- c("RUT_VIVIENDA", "Svc_Mono", "Svc_Duo", "Svc_Triple", "PROF_1", "PROF_2", "PROF_3", "PROF_4", "PROF_5", "EDAD", "FLAG_VIP", "SEXO",
```

```
"SECTOR_ALTO", "SECTOR_MEDIO", "SECTOR_BAJO", "GSE_COM",
"DIAS_PERMANENCIA", "VALOR_DEUDA", "DIAS_DEUDA", "CICLO_PEQUEÑO",
"CICLO_MEDIANO", "CICLO_DURO", "FLUJO_SUAVE", "FLUJO_MEDIANO",
"FLUJO_DURO", "BOLETA_ELECT", "VALOR_FACTURACION", "ZONA_NORTE",
"ZONA_CENTRO", "ZONA_SUR", "ZONA_SANTIAGO")
colnames(base2_1_df)
summary(base2_1_df)
```

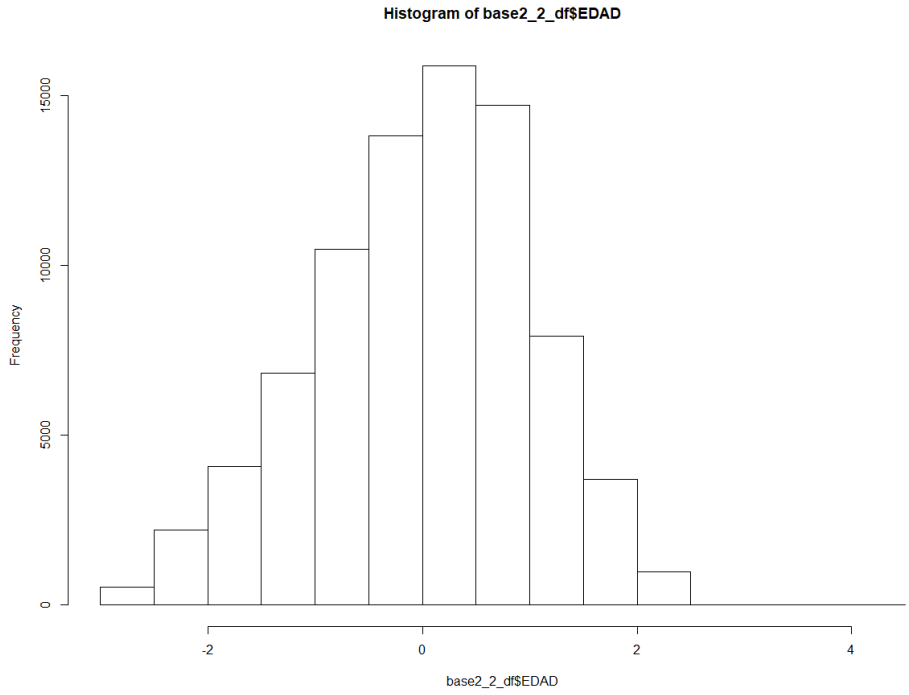
Anexo 11. Código utilizado para generar base de clientes con variables no binarias estandarizadas con puntaje Z.

```
#Estandarizando variable EDAD con puntaje Z
edad_valores <- base$EDAD
edad_Z <- rep(0,81107)
media_edad <- mean(base$EDAD)
desv_est_edad <- sd(base$EDAD)
for (x in valor_seq) {
  edad_Z[x] <- (edad_valores[x] - media_edad)/desv_est_edad
}
#Estandarizando variable DIAS_PERMANENCIA con puntaje Z
diasperm_valores <- base$DIAS_PERMANENCIA
diasperm_Z <- rep(0,81107)
media_diasperm <- mean(base$DIAS_PERMANENCIA)
desv_est_diasperm <- sd(base$DIAS_PERMANENCIA)
for (x in valor_seq) {
  diasperm_Z[x] <- (diasperm_valores[x] - media_diasperm)/desv_est_diasperm
}
#Estandarizando variable VALOR_DEUDA con puntaje Z
valordeuda_valores <- base$VALOR_DEUDA
valordeuda_Z <- rep(0,81107)
media_valordeuda <- mean(base$VALOR_DEUDA)
```

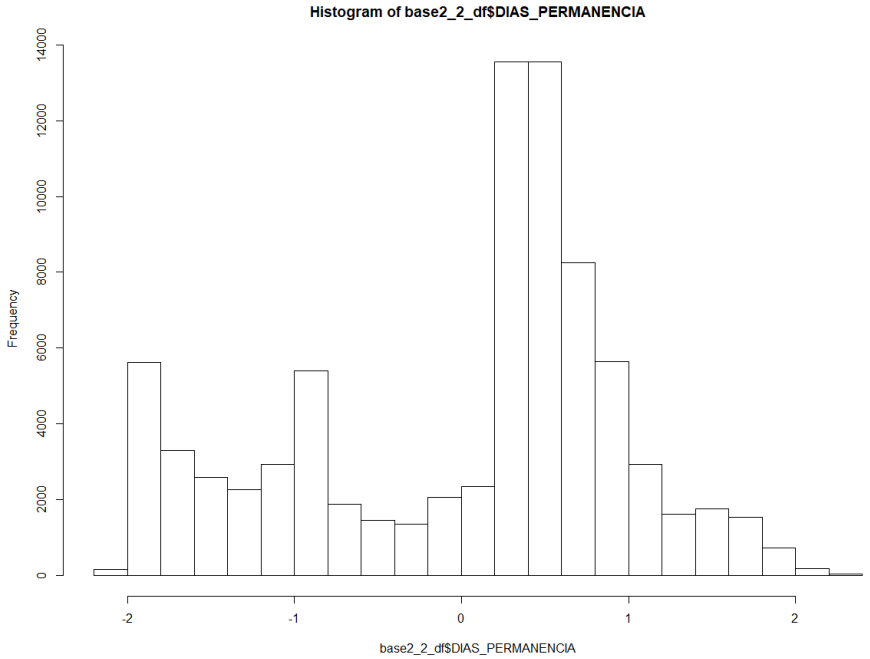
```
desv_est_valordeuda <- sd(base$VALOR_DEUDA)
for (x in valor_seq) {
  valordeuda_Z[x] <- (valordeuda_valores[x] - media_valordeuda)/desv_est_valordeuda
}
#Estandarizando variable DIAS_DEUDA con puntaje Z
diasdeuda_valores <- base$DIAS_DEUDA
diasdeuda_Z <- rep(0,81107)
media_diasdeuda <- mean(base$DIAS_DEUDA)
desv_est_diasdeuda <- sd(base$DIAS_DEUDA)
for (x in valor_seq) {
  diasdeuda_Z[x] <- (diasdeuda_valores[x] - media_diasdeuda)/desv_est_diasdeuda
}
#Estandarizando variable VALOR_FACTURACION con puntaje Z
valorfact_valores <- base$VALOR_FACTURACION
valorfact_Z <- rep(0,81107)
media_valorfact <- mean(base$VALOR_FACTURACION)
desv_est_valorfact <- sd(base$VALOR_FACTURACION)
for (x in valor_seq) {
  valorfact_Z[x] <- (valorfact_valores[x] - media_valorfact)/desv_est_valorfact
}
#Trabajando con base
base2_2 <- cbind(base$RUT_VIVIENDA, base$Svc_Mono, base$Svc_Duo,
base$Svc_Triple, base$PROF_1, base$PROF_2, base$PROF_3, base$PROF_4,
base$PROF_5, edad_Z, base$FLAG_VIP, base$SEXO, var_sector_alto,
var_sector_medio, var_sector_bajo, base$GSE_COM, diasperm_Z, valordeuda_Z,
diasdeuda_Z, var_ciclo_pequeño, var_ciclo_mediano, var_ciclo_grande,
base$FLUJO_SUAVE, base$FLUJO_MEDIANO, base$FLUJO_DURO, base$BOLETA_ELECT,
valorfact_Z, base$ZONA_NORTE, base$ZONA_CENTRO, base$ZONA_SUR,
var_zona_stgo)
```

```
class(base2_2)
base2_2_df <- data.frame(base2_2)
class(base2_2_df)
colnames(base2_2_df) <- c("RUT_VIVIENDA", "Svc_Mono", "Svc_Duo", "Svc_Triple", "PROF_1", "PROF_2", "PROF_3", "PROF_4", "PROF_5", "EDAD", "FLAG_VIP", "SEXO", "SECTOR_ALTO", "SECTOR_MEDIO", "SECTOR_BAJO", "GSE_COM", "DIAS_PERMANENCIA", "VALOR_DEUDA", "DIAS_DEUDA", "CICLO_PEQUEÑO", "CICLO_MEDIANO", "CICLO_DURO", "FLUJO_SUAVE", "FLUJO_MEDIANO", "FLUJO_DURO", "BOLETA_ELECT", "VALOR_FACTURACION", "ZONA_NORTE", "ZONA_CENTRO", "ZONA_SUR", "ZONA_SANTIAGO")
colnames(base2_2_df)
summary(base2_2_df)
```

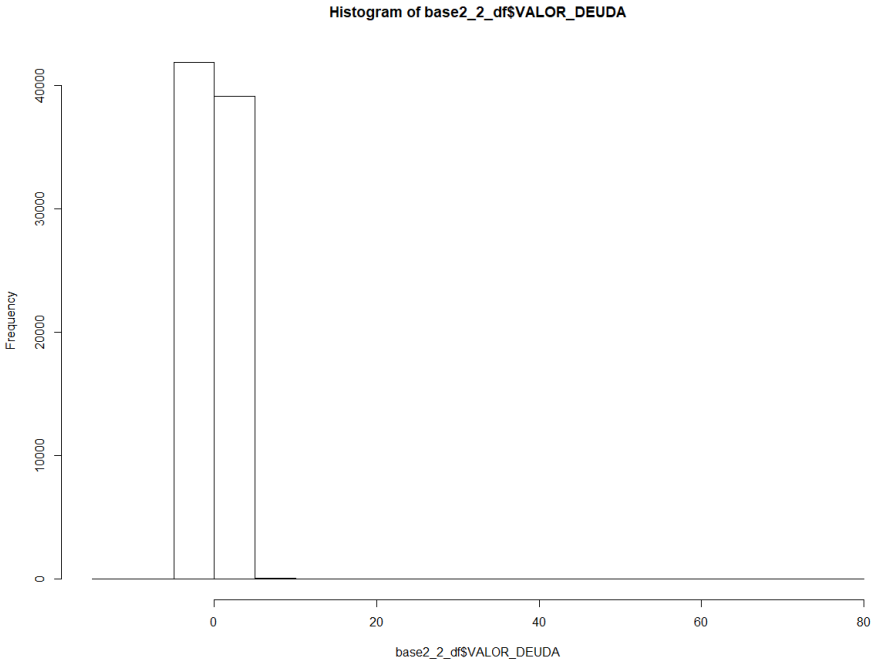
Anexo 12. Histograma de variable estandarizada EDAD (Fuente: Elaboración propia).



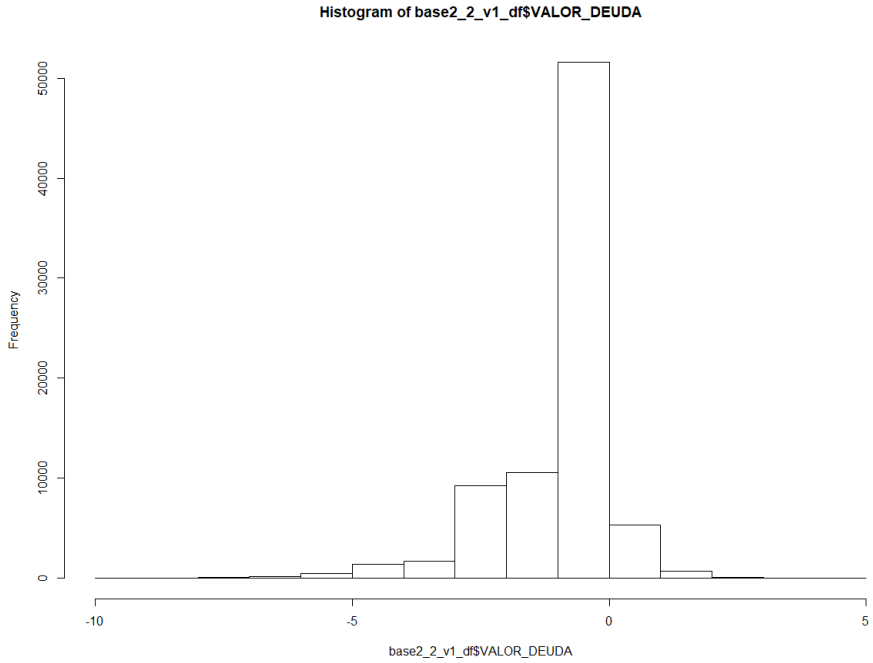
Anexo 13. Histograma de variable estandarizada DIAS_PERMANENCIA (Fuente: Elaboración propia).



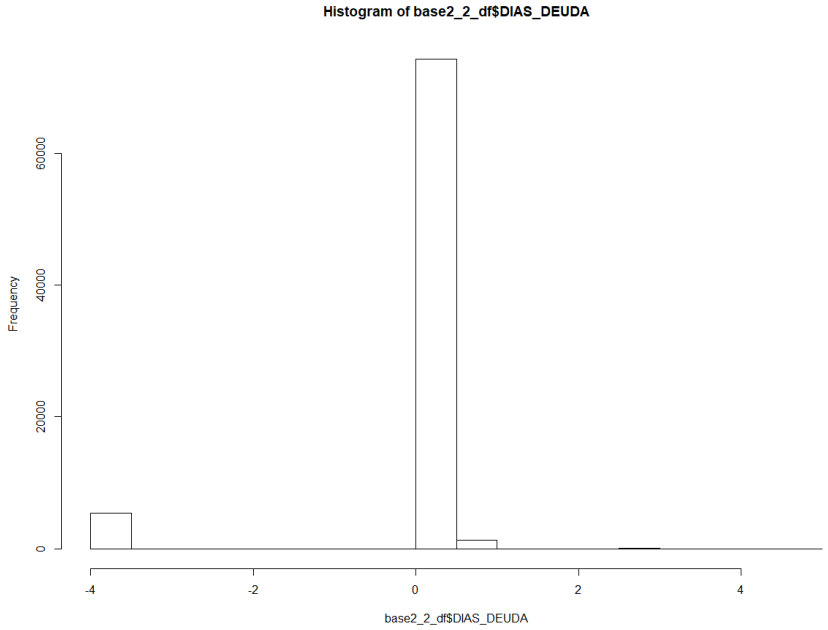
Anexo 14. Histograma de variable estandarizada VALOR_DEUDA (Fuente: Elaboración propia).



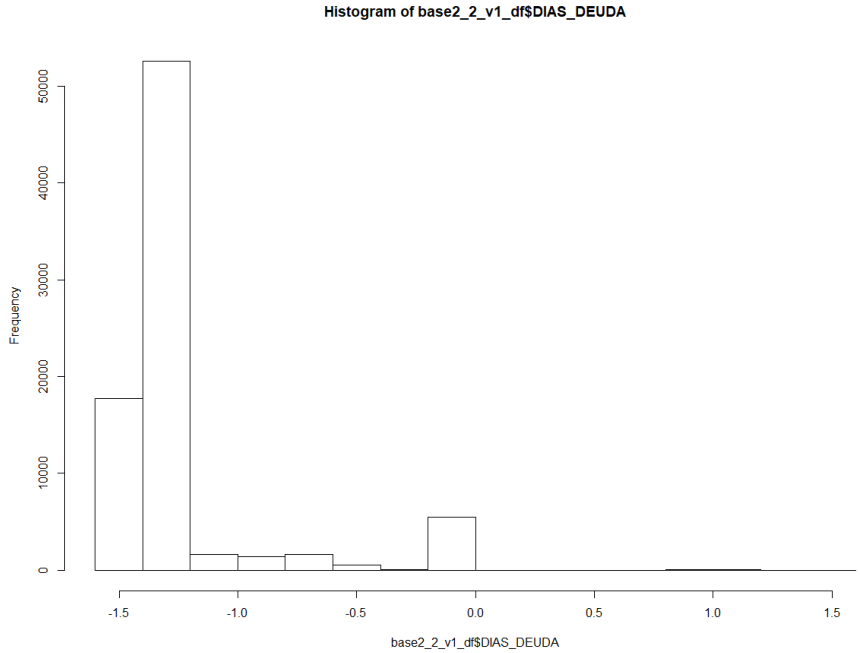
Anexo 15. Histograma de variable normalizada VALOR_DEUDA, una vez aplicada la función logaritmo natural (Fuente: Elaboración propia).



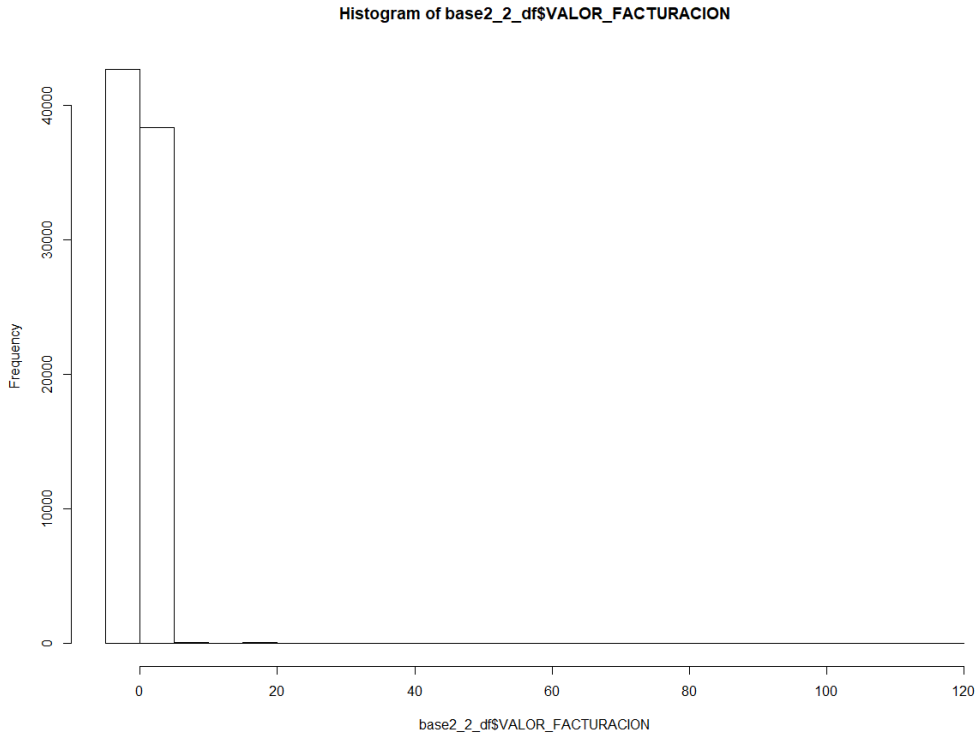
Anexo 16. Histograma de variable estandarizada DIAS_DEUDA (Fuente: Elaboración propia).



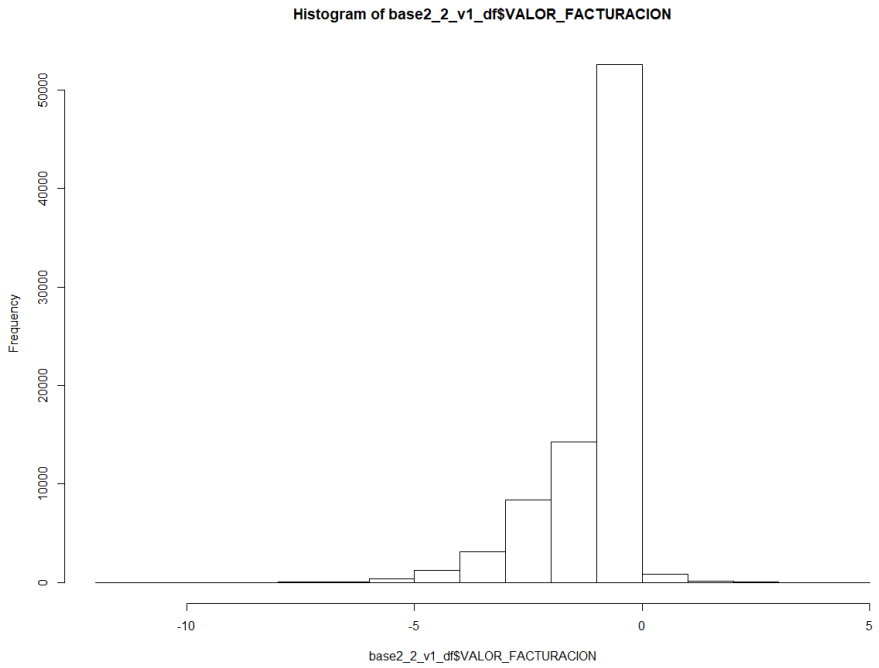
Anexo 17. Histograma de variable DIAS_DEUDA, transformada con función logarítmica natural (Fuente: Elaboración propia).



Anexo 18. Histograma de variable estandarizada VALOR_FACTURACION (Fuente: Elaboración propia).



Anexo 19. Histograma de variable estandarizada VALOR_FACTURACION con función logarítmica natural (Fuente: Elaboración propia).

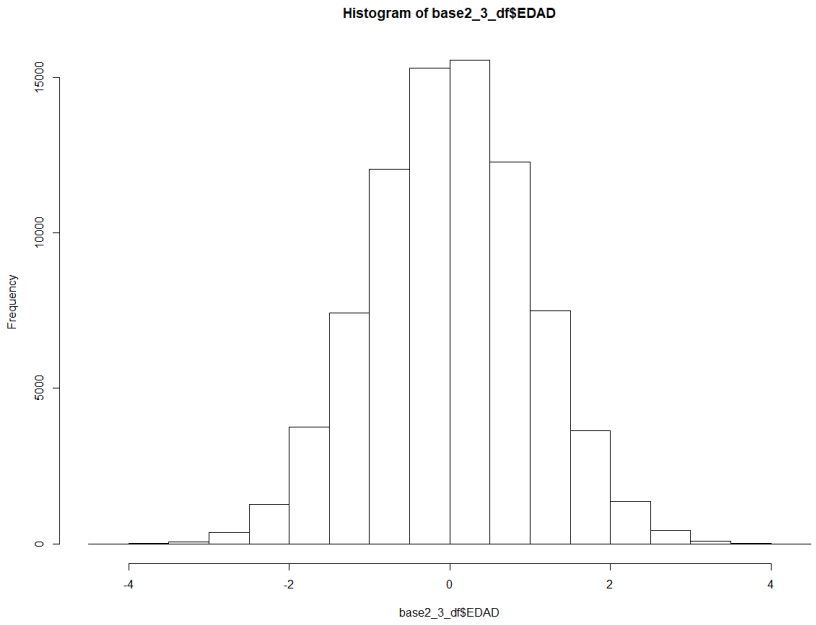


Anexo 20. Código utilizado para generar base con variables normalizadas.

```
base2_3 <- cbind(base$RUT_VIVIENDA, rnorm(base$Svc_Mono), rnorm(base$Svc_Duo),
rnorm(base$Svc_Triple), rnorm(base$PROF_1), rnorm(base$PROF_2),
rnorm(base$PROF_3), rnorm(base$PROF_4), rnorm(base$PROF_5),
rnorm(base$EDAD), rnorm(base$FLAG_VIP), rnorm(base$SEXO),
rnorm(var_sector_alto), rnorm(var_sector_medio), rnorm(var_sector_bajo),
rnorm(base$GSE_COM), rnorm(base$DIAS_PERMANENCIA),
rnorm(base$VALOR_DEUDA), rnorm(base$DIAS_DEUDA), rnorm(var_ciclo_pequeño),
rnorm(var_ciclo_mediano), rnorm(var_ciclo_grande), rnorm(base$FLUJO_SUAVE),
rnorm(base$FLUJO_MEDIANO), rnorm(base$FLUJO_DURO),
rnorm(base$BOLETA_ELECT), rnorm(base$VALOR_FACTURACION),
rnorm(base$ZONA_NORTE), rnorm(base$ZONA_CENTRO),
rnorm(base$ZONA_SUR), rnorm(var_zona_stgo))
class(base2_3)
base2_3_df <- data.frame(base2_3)
```

```
class(base2_3_df)
colnames(base2_3_df) <- c("RUT_VIVIENDA", "Svc_Mono", "Svc_Duo", "Svc_Triple", "PROF_1", "PROF_2", "PROF_3", "PROF_4", "PROF_5", "EDAD", "FLAG_VIP", "SEXO", "SECTOR_ALTO", "SECTOR_MEDIO", "SECTOR_BAJO", "GSE_COM", "DIAS_PERMANENCIA", "VALOR_DEUDA", "DIAS_DEUDA", "CICLO_PEQUEÑO", "CICLO_MEDIANO", "CICLO_DURO", "FLUJO_SUAVE", "FLUJO_MEDIANO", "FLUJO_DURO", "BOLETA_ELECT", "VALOR_FACTURACION", "ZONA_NORTE", "ZONA_CENTRO", "ZONA_SUR", "ZONA_SANTIAGO")
colnames(base2_3_df)
summary(base2_3_df)
```

Anexo 21. Histograma de variable EDAD en base normalizada.



Anexo 22. Código utilizado para generar base con variables estandarizadas con puntaje Z.

```
#Estandarizando variable Svc_Mono con puntaje Z
svcmono_valores_4 <- base$Svc_Mono
svcmono_norm_4 <- rep(0,81107)
media_svcmono_4 <- mean(base$Svc_Mono)
desv_est_svcmono_4 <- sd(base$Svc_Mono)
```

```
for (x in valor_seq) {
 svcmono_norm_4[x] <- (svcmono_valores_4[x] - media_svcmono_4) /
desv_est_svcmono_4
}
#Estandarizando variable Svc_Duo con puntaje Z
svcduo_valores_4 <- base$Svc_Duo
svcduo_norm_4 <- rep(0,81107)
media_svcduo_4 <- mean(base$Svc_Duo)
desv_est_svcduo_4 <- sd(base$Svc_Duo)
for (x in valor_seq) {
 svcduo_norm_4[x] <- (svcduo_valores_4[x] - media_svcduo_4)/desv_est_svcduo_4
}
#Estandarizando variable Svc_Triple con puntaje Z
svctriple_valores_4 <- base$Svc_Triple
svctriple_norm_4 <- rep(0,81107)
media_svctriple_4 <- mean(base$Svc_Triple)
desv_est_svctriple_4 <- sd(base$Svc_Triple)
for (x in valor_seq) {
 svctriple_norm_4[x] <- (svctriple_valores_4[x] - media_svctriple_4) /
desv_est_svctriple_4
}
#Estandarizando variable PROF_1 con puntaje Z
prof1_valores_4 <- base$PROF_1
prof1_norm_4 <- rep(0,81107)
media_prof1_4 <- mean(base$PROF_1)
desv_est_prof1_4 <- sd(base$PROF_1)
for (x in valor_seq) {
 prof1_norm_4[x] <- (prof1_valores_4[x] - media_prof1_4)/desv_est_prof1_4
}
```

```
#Estandarizando variable PROF_2 con puntaje Z
prof2_valores_4 <- base$PROF_2
prof2_norm_4 <- rep(0,81107)
media_prof2_4 <- mean(base$PROF_2)
desv_est_prof2_4 <- sd(base$PROF_2)
for (x in valor_seq) {
  prof2_norm_4[x] <- (prof2_valores_4[x] - media_prof2_4)/desv_est_prof2_4
}
#Estandarizando variable PROF_3 con puntaje Z
prof3_valores_4 <- base$PROF_3
prof3_norm_4 <- rep(0,81107)
media_prof3_4 <- mean(base$PROF_3)
desv_est_prof3_4 <- sd(base$PROF_3)
for (x in valor_seq) {
  prof3_norm_4[x] <- (prof3_valores_4[x] - media_prof3_4)/desv_est_prof3_4
}
#Estandarizando variable PROF_4 con puntaje Z
prof4_valores_4 <- base$PROF_4
prof4_norm_4 <- rep(0,81107)
media_prof4_4 <- mean(base$PROF_4)
desv_est_prof4_4 <- sd(base$PROF_4)
for (x in valor_seq) {
  prof4_norm_4[x] <- (prof4_valores_4[x] - media_prof4_4)/desv_est_prof4_4
}
#Estandarizando variable PROF_5 con puntaje Z
prof5_valores_4 <- base$PROF_5
prof5_norm_4 <- rep(0,81107)
media_prof5_4 <- mean(base$PROF_5)
desv_est_prof5_4 <- sd(base$PROF_5)
```

```
for (x in valor_seq) {
  prof5_norm_4[x] <- (prof5_valores_4[x] - media_prof5_4)/desv_est_prof5_4
}
#Estandarizando variable EDAD con puntaje Z
edad_valores_4 <- base$EDAD
edad_norm_4 <- rep(0,81107)
media_edad_4 <- mean(base$EDAD)
desv_est_edad_4 <- sd(base$EDAD)
for (x in valor_seq) {
  edad_norm_4[x] <- (edad_valores_4[x] - media_edad_4)/desv_est_edad_4
}
#Estandarizando variable FLAG_VIP con puntaje Z
vip_valores_4 <- base$FLAG_VIP
vip_norm_4 <- rep(0,81107)
media_vip_4 <- mean(base$FLAG_VIP)
desv_est_vip_4 <- sd(base$FLAG_VIP)
for (x in valor_seq) {
  vip_norm_4[x] <- (vip_valores_4[x] - media_vip_4)/desv_est_vip_4
}
#Estandarizando variable SEXO con puntaje Z
sexo_valores_4 <- base$SEXO
sexo_norm_4 <- rep(0,81107)
media_sexo_4 <- mean(base$SEXO)
desv_est_sexo_4 <- sd(base$SEXO)
for (x in valor_seq) {
  sexo_norm_4[x] <- (sexo_valores_4[x] - media_sexo_4)/desv_est_sexo_4
}
#Estandarizando variable VAR_SECTOR_ALTO con puntaje Z
sectalto_valores_4 <- var_sector_alto
```

```
sectalto_norm_4 <- rep(0,81107)
media_sectalto_4 <- mean(var_sector_alto)
desv_est_sectalto_4 <- sd(var_sector_alto)
for (x in valor_seq) {
  sectalto_norm_4[x] <- (sectalto_valores_4[x] - media_sectalto_4)/desv_est_sectalto_4
}
#Estandarizando variable VAR_SECTOR_MEDIO con puntaje Z
sectmedio_valores_4 <- var_sector_medio
sectmedio_norm_4 <- rep(0,81107)
media_sectmedio_4 <- mean(var_sector_medio)
desv_est_sectmedio_4 <- sd(var_sector_medio)
for (x in valor_seq) {
  sectmedio_norm_4[x] <- (sectmedio_valores_4[x] - media_sectmedio_4) /
desv_est_sectmedio_4
}
#Estandarizando variable VAR_SECTOR_BAJO con puntaje Z
sectbajo_valores_4 <- var_sector_bajo
sectbajo_norm_4 <- rep(0,81107)
media_sectbajo_4 <- mean(var_sector_bajo)
desv_est_sectbajo_4 <- sd(var_sector_bajo)
for (x in valor_seq) {
  sectbajo_norm_4[x] <- (sectbajo_valores_4[x] - media_sectbajo_4) /
desv_est_sectbajo_4
}
#Estandarizando variable GSE_COM con puntaje Z
com_valores_4 <- base$GSE_COM
com_norm_4 <- rep(0,81107)
media_com_4 <- mean(base$GSE_COM)
desv_est_com_4 <- sd(base$GSE_COM)
```

```
for (x in valor_seq) {
  com_norm_4[x] <- (com_valores_4[x] - media_com_4)/desv_est_com_4
}
#Estandarizando variable DIAS_PERMANENCIA con puntaje Z
diasperm_valores_4 <- base$DIAS_PERMANENCIA
diasperm_norm_4 <- rep(0,81107)
media_diasperm_4 <- mean(base$DIAS_PERMANENCIA)
desv_est_diasperm_4 <- sd(base$DIAS_PERMANENCIA)
for (x in valor_seq) {
  diasperm_norm_4[x] <- (diasperm_valores_4[x] - media_diasperm_4) /
desv_est_diasperm_4
}
#Estandarizando variable VALOR_DEUDA con puntaje Z
valordeuda_valores_4 <- base$VALOR_DEUDA
valordeuda_norm_4 <- rep(0,81107)
media_valordeuda_4 <- mean(base$VALOR_DEUDA)
desv_est_valordeuda_4 <- sd(base$VALOR_DEUDA)
for (x in valor_seq) {
  valordeuda_norm_4[x] <- (valordeuda_valores_4[x] - media_valordeuda_4) /
desv_est_valordeuda_4
}
#Estandarizando variable DIAS_DEUDA con puntaje Z
diasdeuda_valores_4 <- base$DIAS_DEUDA
diasdeuda_norm_4 <- rep(0,81107)
media_diasdeuda_4 <- mean(base$DIAS_DEUDA)
desv_est_diasdeuda_4 <- sd(base$DIAS_DEUDA)
for (x in valor_seq) {
  diasdeuda_norm_4[x] <- (diasdeuda_valores_4[x] - media_diasdeuda_4) /
desv_est_diasdeuda_4
```

```
}
#Estandarizando variable var_ciclo_pequeño con puntaje Z
ciclopequeño_valores_4 <- var_ciclo_pequeño
ciclopequeño_norm_4 <- rep(0,81107)
media_ciclopequeño_4 <- mean(var_ciclo_pequeño)
desv_est_ciclopequeño_4 <- sd(var_ciclo_pequeño)
for (x in valor_seq) {
 ciclopequeño_norm_4[x] <- (ciclopequeño_valores_4[x] - media_ciclopequeño_4) /
desv_est_ciclopequeño_4
}
#Estandarizando variable var_ciclo_mediano con puntaje Z
ciclomediano_valores_4 <- var_ciclo_mediano
ciclomediano_norm_4 <- rep(0,81107)
media_ciclomediano_4 <- mean(var_ciclo_mediano)
desv_est_ciclomediano_4 <- sd(var_ciclo_mediano)
for (x in valor_seq) {
 ciclomediano_norm_4[x] <- (ciclomediano_valores_4[x] - media_ciclomediano_4) /
desv_est_ciclomediano_4
}
#Estandarizando variable var_ciclo_grande con puntaje Z
ciclogrande_valores_4 <- var_ciclo_grande
ciclogrande_norm_4 <- rep(0,81107)
media_ciclogrande_4 <- mean(var_ciclo_grande)
desv_est_ciclogrande_4 <- sd(var_ciclo_grande)
for (x in valor_seq) {
 ciclogrande_norm_4[x] <- (ciclogrande_valores_4[x] - media_ciclogrande_4) /
desv_est_ciclogrande_4
}
#Estandarizando variable FLUJO_SUAVE con puntaje Z
```

```
flujosuave_valores_4 <- base$FLUJO_SUAVE
flujosuave_norm_4 <- rep(0,81107)
media_flujosuave_4 <- mean(base$FLUJO_SUAVE)
desv_est_flujosuave_4 <- sd(base$FLUJO_SUAVE)
for (x in valor_seq) {
 flujosuave_norm_4[x] <- (flujosuave_valores_4[x] - media_flujosuave_4) /
desv_est_flujosuave_4
}
#Estandarizando variable FLUJO_MEDIANO con puntaje Z
flujomediano_valores_4 <- base$FLUJO_MEDIANO
flujomediano_norm_4 <- rep(0,81107)
media_flujomediano_4 <- mean(base$FLUJO_MEDIANO)
desv_est_flujomediano_4 <- sd(base$FLUJO_MEDIANO)
for (x in valor_seq) {
 flujomediano_norm_4[x] <- (flujomediano_valores_4[x] - media_flujomediano_4) /
desv_est_flujomediano_4
}
#Estandarizando variable FLUJO_DURO con puntaje Z
flujoduro_valores_4 <- base$FLUJO_DURO
flujoduro_norm_4 <- rep(0,81107)
media_flujoduro_4 <- mean(base$FLUJO_DURO)
desv_est_flujoduro_4 <- sd(base$FLUJO_DURO)
for (x in valor_seq) {
 flujoduro_norm_4[x] <- (flujoduro_valores_4[x] - media_flujoduro_4) /
desv_est_flujoduro_4
}
#Estandarizando variable BOLETA_ELECT con puntaje Z
boletaelect_valores_4 <- base$BOLETA_ELECT
boletaelect_norm_4 <- rep(0,81107)
```

```
media_boletaelect_4 <- mean(base$BOLETA_ELECT)
desv_est_boletaelect_4 <- sd(base$BOLETA_ELECT)
for (x in valor_seq) {
 boletaelect_norm_4[x] <- (boletaelect_valores_4[x] - media_boletaelect_4) /
desv_est_boletaelect_4
}
#Estandarizando variable VALOR_FACTURACION con puntaje Z
valorfact_valores_4 <- base$VALOR_FACTURACION
valorfact_norm_4 <- rep(0,81107)
media_valorfact_4 <- mean(base$VALOR_FACTURACION)
desv_est_valorfact_4 <- sd(base$VALOR_FACTURACION)
for (x in valor_seq) {
 valorfact_norm_4[x] <- (valorfact_valores_4[x] - media_valorfact_4) /
desv_est_valorfact_4
}
#Estandarizando variable ZONA_NORTE con puntaje Z
zonanorte_valores_4 <- base$ZONA_NORTE
zonanorte_norm_4 <- rep(0,81107)
media_zonanorte_4 <- mean(base$ZONA_NORTE)
desv_est_zonanorte_4 <- sd(base$ZONA_NORTE)
for (x in valor_seq) {
 zonanorte_norm_4[x] <- (zonanorte_valores_4[x] - media_zonanorte_4) /
desv_est_zonanorte_4
}
#Estandarizando variable ZONA_CENTRO con puntaje Z
zonacentro_valores_4 <- base$ZONA_CENTRO
zonacentro_norm_4 <- rep(0,81107)
media_zonacentro_4 <- mean(base$ZONA_CENTRO)
desv_est_zonacentro_4 <- sd(base$ZONA_CENTRO)
```

```
for (x in valor_seq) {
 zonacentro_norm_4[x] <- (zonacentro_valores_4[x] - media_zonacentro_4) / desv_est_zonacentro_4
}
#Estandarizando variable ZONA_SUR con puntaje Z
zonasur_valores_4 <- base$ZONA_SUR
zonasur_norm_4 <- rep(0,81107)
media_zonasur_4 <- mean(base$ZONA_SUR)
desv_est_zonasur_4 <- sd(base$ZONA_SUR)
for (x in valor_seq) {
 zonasur_norm_4[x] <- (zonasur_valores_4[x] - media_zonasur_4)/desv_est_zonasur_4
}
#Estandarizando variable var_zona_stgo con puntaje Z
zonastgo_valores_4 <- var_zona_stgo
zonastgo_norm_4 <- rep(0,81107)
media_zonastgo_4 <- mean(var_zona_stgo)
desv_est_zonastgo_4 <- sd(var_zona_stgo)
for (x in valor_seq) {
 zonastgo_norm_4[x] <- (zonastgo_valores_4[x] - media_zonastgo_4) / desv_est_zonastgo_4
}
#Generación de base
base2_4 <- cbind(base$RUT_VIVIENDA, svcmono_norm_4, svcduo_norm_4, svctriple_norm_4, prof1_norm_4, prof2_norm_4, prof3_norm_4, prof4_norm_4, prof5_norm_4, edad_norm_4, vip_norm_4, sexo_norm_4, sectalto_norm_4, sectmedio_norm_4, sectbajo_norm_4, com_norm_4, diasperm_norm_4, valordeuda_norm_4, diasdeuda_norm_4, ciclopequeño_norm_4, ciclomediano_norm_4, ciclogrande_norm_4, flujosuave_norm_4,
```

flujomediano_norm_4, flujoduro_norm_4, boletaelect_norm_4, valorfact_norm_4, zonanorte_norm_4, zonacentro_norm_4, zonasur_norm_4, zonastgo_norm_4)

class(base2_4)

base2_4_df <- data.frame(base2_4)

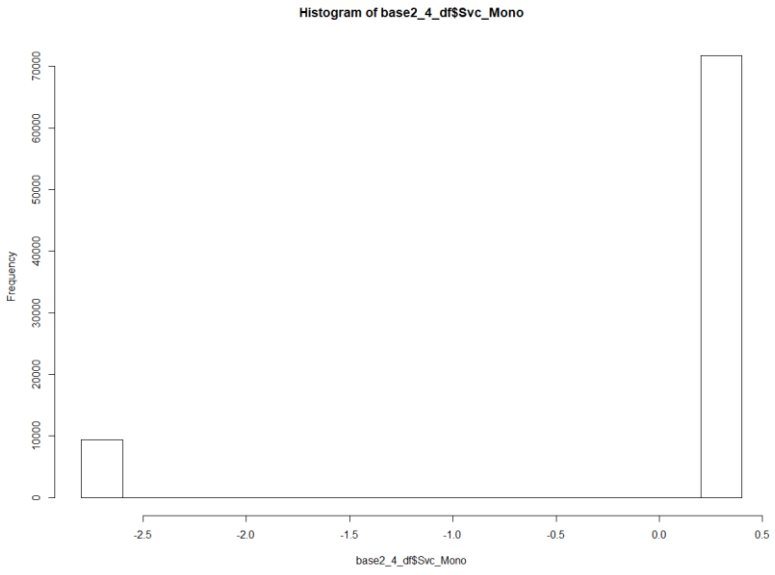
class(base2_4_df)

colnames(base2_4_df) <- c("RUT_VIVIENDA", "Svc_Mono", "Svc_Duo", "Svc_Triple", "PROF_1", "PROF_2", "PROF_3", "PROF_4", "PROF_5", "EDAD", "FLAG_VIP", "SEXO", "SECTOR_ALTO", "SECTOR_MEDIO", "SECTOR_BAJO", "GSE_COM", "DIAS_PERMANENCIA", "VALOR_DEUDA", "DIAS_DEUDA", "CICLO_PEQUEÑO", "CICLO_MEDIANO", "CICLO_GRANDE", "FLUJO_SUAVE", "FLUJO_MEDIANO", "FLUJO_DURO", "BOLETA_ELECT", "VALOR_FACTURACION", "ZONA_NORTE", "ZONA_CENTRO", "ZONA_SUR", "ZONA_SANTIAGO")

colnames(base2_4_df)

summary(base2_4_df)

Anexo 23. Histograma de variable 'Svc_Mono' estandarizada (Fuente: Elaboración propia).



Anexo 24. Tabla de varianzas de base con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

	Varianza
Svc_Mono	0,1023
Svc_Duo	0,0938
Svc_Triple	0,0108
PROF_1	0,2375
PROF_2	0,0671
PROF_3	0,0156
PROF_4	0,0067
PROF_5	0,2497
EDAD	0,9992
FLAG_VIP	0,0012
SEXO	0,2495
SECTOR_ALTO	0,2025
SECTOR_MEDIO	0,2499
SECTOR_BAJO	0,1209
GSE_COM	0,0788
DIAS_PERM	0,9984
VALOR_DEUDA	0,9912
DIAS_DEUDA	1,0012
CICLO_PEQUEÑO	0,1212
CICLO_MEDIANO	0,2248
CICLO_DURO	0,2442
FLUJO_SUAVE	0,1654
FLUJO_MEDIANO	0,2426
FLUJO_DURO	0,1629
BOLETA_ELECT	0,1441
VALOR_FACT	0,9936
ZONA_NORTE	0,1503
ZONA_CENTRO	0,1555
ZONA_SUR	0,1466
ZONA_SANTIAGO	0,2469

Anexo 25. Matriz de correlación de base con variables no binarias normalizadas entre 0 y 1.

	Svc_Mono	Svc_Duo	Svc_Triple	PROF_1	PROF_2	PROF_3	PROF_5	EDAD	SEXO	SECTOR_ALTO	SECTOR_MEDIO	SECTOR_BAJO	GSE_COM	DIAS_PERM	VALOR_DEUDA	DIAS_DEUDA
Svc_Mono	0,1023	-0,0926	-0,0096	-0,0020	-0,0008	0,0001	0,0024	-0,0024	-0,0004	-0,0126	0,0039	0,0070	0,0017	-0,0017	-0,0003	-0,0005
Svc_Duo	-0,0926	0,0938	-0,0011	0,0019	0,0007	-0,0001	-0,0023	0,0028	0,0002	0,0113	-0,0030	-0,0067	-0,0017	0,0014	0,0003	0,0006
Svc_Triple	-0,0096	-0,0011	0,0108	0,0001	0,0002	-0,0001	-0,0002	-0,0004	0,0002	0,0013	-0,0009	-0,0003	-0,0001	0,0003	0,0000	-0,0001
PROF_1	-0,0020	0,0019	0,0001	0,2375	-0,0280	-0,0061	-0,2007	0,0004	0,0078	0,0060	0,0006	-0,0029	-0,0037	-0,0022	-0,0001	-0,0001
PROF_2	-0,0008	0,0007	0,0002	-0,0280	0,0671	-0,0011	-0,0374	-0,0001	0,0004	0,0014	0,0012	-0,0013	-0,0013	-0,0005	0,0015	-0,0008
PROF_3	0,0001	-0,0001	-0,0001	-0,0061	-0,0011	0,0156	-0,0082	0,0001	0,0006	0,0001	0,0000	-0,0001	0,0000	0,0002	0,0000	0,0001
PROF_5	0,0024	-0,0023	-0,0002	-0,2007	-0,0374	-0,0082	0,2497	-0,0005	-0,0096	-0,0073	-0,0009	0,0047	0,0034	0,0026	-0,0016	0,0004
EDAD	-0,0024	0,0028	-0,0004	0,0004	-0,0001	0,0001	-0,0005	0,9992	-0,0009	-0,0008	0,0032	-0,0021	-0,0002	-0,0010	-0,0008	-0,0005
SEXO	-0,0004	0,0002	0,0002	0,0078	0,0004	0,0006	-0,0096	-0,0009	0,2495	0,0067	-0,0019	-0,0062	0,0014	-0,0014	-0,0005	-0,0005
SECTOR_ALTO	-0,0126	0,0113	0,0013	0,0060	0,0014	0,0001	-0,0073	-0,0008	0,0067	0,2025	-0,1385	-0,0397	-0,0243	-0,0003	0,0018	0,0025
SECTOR_MEDIO	0,0039	-0,0030	-0,0009	0,0006	0,0012	0,0000	-0,0009	0,0032	-0,0019	-0,1385	0,2499	-0,0691	-0,0423	-0,0001	-0,0006	-0,0042
SECTOR_BAJO	0,0070	-0,0067	-0,0003	-0,0029	-0,0013	-0,0001	0,0047	-0,0021	-0,0062	-0,0397	-0,0691	0,1209	-0,0121	0,0007	-0,0011	0,0008
GSE_COM	0,0017	-0,0017	-0,0001	-0,0037	-0,0013	0,0000	0,0034	-0,0002	0,0014	-0,0243	-0,0423	-0,0121	0,0788	-0,0003	0,0000	0,0009
DIAS_PERM	-0,0017	0,0014	0,0003	-0,0022	-0,0005	0,0002	0,0026	-0,0010	-0,0014	-0,0003	-0,0001	0,0007	-0,0003	0,9984	-0,0039	0,0009
VALOR_DEUDA	-0,0003	0,0003	0,0000	-0,0001	0,0015	0,0000	-0,0016	-0,0008	-0,0005	0,0018	-0,0006	-0,0011	0,0000	-0,0039	0,9912	0,0023
DIAS_DEUDA	-0,0005	0,0006	-0,0001	-0,0001	-0,0008	0,0001	0,0004	-0,0005	-0,0005	0,0025	-0,0042	0,0008	0,0009	0,0009	0,0023	1,0012
CICLO_PEQUEÑO	0,0031	-0,0027	-0,0004	0,0028	0,0002	-0,0002	-0,0031	0,0006	0,0051	-0,0035	0,0014	0,0016	0,0006	0,0015	0,0003	0,0004
CICLO_MEDIANO	0,0057	-0,0055	-0,0003	-0,0070	-0,0014	-0,0001	0,0089	-0,0018	-0,0074	-0,0186	0,0088	0,0079	0,0018	-0,0008	0,0009	0,0010
CICLO_DURO	-0,0119	0,0109	0,0011	0,0071	0,0019	0,0003	-0,0092	0,0005	0,0026	0,0248	-0,0108	-0,0106	-0,0033	-0,0006	-0,0022	-0,0006
FLUJO_SUAVE	-0,0015	0,0013	0,0002	0,0029	0,0004	0,0005	-0,0058	-0,0003	0,0196	0,0277	-0,0138	-0,0144	0,0005	0,0007	0,0016	-0,0002
FLUJO_MEDIANO	-0,0030	0,0036	-0,0006	0,0108	0,0029	-0,0001	-0,0119	0,0007	-0,0088	-0,0055	0,0083	0,0022	-0,0050	-0,0018	-0,0027	-0,0026
FLUJO_DURO	0,0045	-0,0048	0,0004	-0,0137	-0,0033	-0,0004	0,0177	-0,0004	-0,0108	-0,0222	0,0054	0,0122	0,0045	0,0011	0,0010	0,0028
BOLETA_ELECT	-0,0079	0,0069	0,0010	-0,0023	0,0000	0,0001	0,0019	-0,0007	0,0122	0,0211	-0,0042	-0,0156	-0,0013	0,0003	0,0000	0,0001
VALOR_FACT	-0,0018	0,0010	0,0008	0,0023	0,0002	-0,0001	-0,0025	0,0063	0,0007	-0,0013	0,0028	-0,0003	-0,0012	-0,0066	0,0024	0,0001
ZONA_NORTE	0,0031	-0,0026	-0,0005	0,0017	-0,0004	-0,0006	-0,0004	0,0004	0,0008	-0,0178	0,0044	0,0146	-0,0011	0,0019	-0,0002	0,0016
ZONA_CENTRO	0,0005	-0,0001	-0,0003	0,0009	0,0001	0,0004	-0,0012	0,0004	-0,0016	-0,0043	0,0054	0,0009	-0,0021	-0,0004	-0,0028	0,0018
ZONA_SUR	0,0072	-0,0066	-0,0006	-0,0005	-0,0005	0,0003	0,0000	0,0010	-0,0013	-0,0098	-0,0129	0,0042	0,0185	-0,0006	0,0005	-0,0019
ZONA_SANTIAGO	-0,0107	0,0093	0,0014	-0,0021	0,0008	0,0000	0,0016	-0,0018	0,0022	0,0319	0,0032	-0,0197	-0,0153	-0,0009	0,0026	-0,0014

	CICLO_PEQUEÑO	CICLO_MEDIANO	CICLO_DURO	FLUJO_SUAVE	FLUJO_MEDIANO	FLUJO_DURO	BOLETA_ELECT	VALOR_FACT	ZONA_NORTE	ZONA_CENTRO	ZONA_SUR	ZONA_SANTIAGO
Svc_Mono	0,0031	0,0057	-0,0119	-0,0015	-0,0030	0,0045	-0,0079	-0,0018	0,0031	0,0005	0,0072	-0,0107
Svc_Duo	-0,0027	-0,0055	0,0109	0,0013	0,0036	-0,0048	0,0069	0,0010	-0,0026	-0,0001	-0,0066	0,0093
Svc_Triple	-0,0004	-0,0003	0,0011	0,0002	-0,0006	0,0004	0,0010	0,0008	-0,0005	-0,0003	-0,0006	0,0014
PROF_1	0,0028	-0,0070	0,0071	0,0029	0,0108	-0,0137	-0,0023	0,0023	0,0017	0,0009	-0,0005	-0,0021
PROF_2	0,0002	-0,0014	0,0019	0,0004	0,0029	-0,0033	0,0000	0,0002	-0,0004	0,0001	-0,0005	0,0008
PROF_3	-0,0002	-0,0001	0,0003	0,0005	-0,0001	-0,0004	0,0001	-0,0001	-0,0006	0,0004	0,0003	0,0000
PROF_5	-0,0031	0,0089	-0,0092	-0,0058	-0,0119	0,0177	0,0019	-0,0025	-0,0004	-0,0012	0,0000	0,0016
EDAD	0,0006	-0,0018	0,0005	-0,0003	0,0007	-0,0004	-0,0007	0,0063	0,0004	0,0004	0,0010	-0,0018
SEXO	0,0051	-0,0074	0,0026	0,0196	-0,0088	-0,0108	0,0122	0,0007	0,0008	-0,0016	-0,0013	0,0022
SECTOR_ALTO	-0,0035	-0,0186	0,0248	0,0277	-0,0055	-0,0222	0,0211	-0,0013	-0,0178	-0,0043	-0,0098	0,0319
SECTOR_MEDIO	0,0014	0,0088	-0,0108	-0,0138	0,0083	0,0054	-0,0042	0,0028	0,0044	0,0054	-0,0129	0,0032
SECTOR_BAJO	0,0016	0,0079	-0,0106	-0,0144	0,0022	0,0122	-0,0156	-0,0003	0,0146	0,0009	0,0042	-0,0197
GSE_COM	0,0006	0,0018	-0,0033	0,0005	-0,0050	0,0045	-0,0013	-0,0012	-0,0011	-0,0021	0,0185	-0,0153
DIAS_PERM	0,0015	-0,0008	-0,0006	0,0007	-0,0018	0,0011	0,0003	-0,0066	0,0019	-0,0004	-0,0006	-0,0009
VALOR_DEUDA	0,0003	0,0009	-0,0022	0,0016	-0,0027	0,0010	0,0000	0,0024	-0,0002	-0,0028	0,0005	0,0026
DIAS_DEUDA	0,0004	0,0010	-0,0006	-0,0002	-0,0026	0,0028	0,0001	0,0001	0,0016	0,0018	-0,0019	-0,0014
CICLO_PEQUEÑO	0,1212	-0,0481	-0,0598	0,0056	-0,0020	-0,0037	-0,0037	-0,0023	0,0000	0,0079	-0,0008	-0,0071
CICLO_MEDIANO	-0,0481	0,2248	-0,1446	-0,0196	-0,0073	0,0268	0,0026	0,0044	0,0103	-0,0064	0,0111	-0,0150
CICLO_DURO	-0,0598	-0,1446	0,2442	0,0168	0,0159	-0,0327	0,0000	-0,0014	-0,0163	0,0017	-0,0154	0,0300
FLUJO_SUAVE	0,0056	-0,0196	0,0168	0,1654	-0,1225	-0,0428	0,0162	0,0038	-0,0090	0,0051	-0,0038	0,0077
FLUJO_MEDIANO	-0,0020	-0,0073	0,0159	-0,1225	0,2426	-0,1201	-0,0166	-0,0029	0,0033	0,0020	-0,0004	-0,0049
FLUJO_DURO	-0,0037	0,0268	-0,0327	-0,0428	-0,1201	0,1629	0,0004	-0,0009	0,0057	-0,0071	0,0042	-0,0028
BOLETA_ELECT	-0,0037	0,0026	0,0000	0,0162	-0,0166	0,0004	0,1441	0,0015	0,0000	-0,0087	-0,0089	0,0177
VALOR_FACT	-0,0023	0,0044	-0,0014	0,0038	-0,0029	-0,0009	0,0015	0,9936	-0,0010	-0,0008	0,0004	0,0015
ZONA_NORTE	0,0000	0,0103	-0,0163	-0,0090	0,0033	0,0057	0,0000	-0,0010	0,1503	-0,0355	-0,0329	-0,0820
ZONA_CENTRO	0,0079	-0,0064	0,0017	0,0051	0,0020	-0,0071	-0,0087	-0,0008	-0,0355	0,1555	-0,0344	-0,0856
ZONA_SUR	-0,0008	0,0111	-0,0154	-0,0038	-0,0004	0,0042	-0,0089	0,0004	-0,0329	-0,0344	0,1466	-0,0794
ZONA_SANTIAGO	-0,0071	-0,0150	0,0300	0,0077	-0,0049	-0,0028	0,0177	0,0015	-0,0820	-0,0856	-0,0794	0,2469

Anexo 26. Tabla de varianzas con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

	Varianzas
Svc_Mono	0,1023
Svc_Duo	0,0938
Svc_Triple	0,0108
PROF_1	0,2375
PROF_2	0,0671
PROF_3	0,0156
PROF_4	0,0067
PROF_5	0,2497
EDAD	1,0000
FLAG_VIP	0,0012
SEXO	0,2495
SECTOR_ALTO	0,2025
SECTOR_MEDIO	0,2499
SECTOR_BAJO	0,1209
GSE_COM	0,0788
DIAS_PERM	1,0000
VALOR_DEUDA	1,4586
DIAS_DEUDA	0,1386
CICLO_PEQUEÑO	0,1212
CICLO_MEDIANO	0,2248
CICLO_DURO	0,2442
FLUJO_SUAVE	0,1654
FLUJO_MEDIANO	0,2426
FLUJO_DURO	0,1629
BOLETA_ELECT	0,1441
VALOR_FACT	1,3845
ZONA_NORTE	0,1503
ZONA_CENTRO	0,1555
ZONA_SUR	0,1466
ZONA_SANTIAGO	0,2469

Anexo 27. Matriz de correlación de base con variables no binarias estandarizadas con puntaje Z (Fuente: Elaboración propia).

	Snc_Monо	Snc_Triple	PROF_1	PROF_2	PROF_3	PROF_5	EDAD	SEXO	SECTOR_ALTO	SECTOR_MEDIO	SECTOR_BAJO	DIAS_PERM	VALOR_DEUDA	DIAS_DEUDA	CICLO_PEQUEÑO	CICLO_MEDIANO	CICLO_DURO	FLUJO_SUAVE	FLUJO_MEDIANO	FLUJO_DURO	BOLETA_ELECT	VALOR_FACT	ZONA_NORTE	ZONA_CENTRO	ZONA_SUR	ZONA_SANTIAGO
Snc_Monо	1,0000	-0,2901	-0,0130	-0,0097	-0,0151	-0,0120	-0,0025		-0,0876	-0,0241	-0,1197	-0,1899	0,0516	0,0283	-0,0754	-0,0379	-0,0754	-0,0114	-0,0190	-0,0347	-0,0654	-0,0316	-0,0249	-0,0038	-0,0586	-0,0676
Snc_Triple	-0,2901	1,0000	0,0025	0,0056	-0,0057	-0,0030	-0,0084	0,0046	0,0275	0,0170	0,0997	0,1597	-0,0991	-0,0121	0,0209	0,0379	0,0209	0,0054	0,0096	0,0263	0,0316	0,0086	0,0249	0,0038	0,0586	0,0273
PROF_1	-0,0130	0,0025	1,0000	-0,2223	-0,1010	-0,8241	0,1452	0,0319	0,0274	0,0025	-0,0170	-0,0841	-0,0216	-0,0166	0,0025	-0,0046	0,0296	0,0145	0,0450	-0,0696	-0,0127	-0,0213	0,0088	0,0046	-0,0026	-0,0085
PROF_2	-0,0097	0,0056	-0,2223	1,0000	-0,0354	-0,2888	0,0730	0,0930	0,0124	0,0090	-0,0141	-0,0969	-0,0151	-0,0030	-0,0046	-0,0177	-0,0046	-0,0039	0,0224	-0,0313	-0,0001	-0,0203	-0,0037	-0,0009	-0,0054	0,0063
PROF_3	0,0032	-0,0057	-0,1010	-0,0354	1,0000	-0,1312	-0,0075	0,0096	0,0026	-0,0001	-0,0033	0,0021	-0,0012	-0,0030	-0,0046	-0,0019	0,0042	0,0089	-0,0009	-0,0079	0,0032	0,0007	-0,0132	0,0072	0,0069	-0,0008
PROF_5	0,0151	-0,0030	-0,8241	-0,2888	-0,1312	1,0000	-0,1701	-0,1085	-0,0323	-0,0036	0,0273	-0,1075	0,0172	0,0316	-0,0177	0,0377	-0,0375	-0,0283	-0,0485	0,0877	0,0100	0,0240	-0,0021	-0,0061	0,0000	0,0065
EDAD	-0,0120	-0,0084	0,1452	0,0730	-0,0075	-0,1701	1,0000	0,0158	0,0608	-0,0004	-0,0373	0,3285	-0,0403	-0,0881	0,0179	-0,1030	0,0680	0,1183	-0,2128	-0,2227	-0,0396	-0,0485	0,0579	0,0221	-0,0250	0,1425
SEXO	-0,0025	0,0046	0,0319	0,0030	0,0096	-0,0385	0,0158	1,0000	0,0297	-0,0076	-0,0356	0,0545	-0,0111	-0,0330	0,0292	-0,0311	0,0106	0,0964	-0,0358	-0,0534	0,0643	0,0339	-0,0082	-0,0068	0,0087		
SECTOR_ALTO	-0,0876	0,0275	0,0274	0,0124	0,0026	-0,0323	0,0608	0,0297	1,0000	-0,6156	-0,2536	0,1415	-0,0204	-0,1043	-0,0226	-0,0872	0,1115	0,1514	-0,0250	-0,1220	0,1234	-0,0813	-0,1021	-0,0242	-0,0566	0,1425
SECTOR_MEDIO	0,0241	-0,0170	0,0025	0,0090	-0,0001	-0,0036	-0,0004	-0,0076	-0,6156	1,0000	-0,3976	-0,0293	0,0206	0,0359	0,0078	0,0373	0,0106	-0,0677	0,0338	0,0269	-0,0221	0,0023	0,0225	0,0275	-0,0676	0,0127
SECTOR_BAJO	0,0633	-0,0097	-0,0170	-0,0141	-0,0033	0,0273	-0,0373	0,0356	-0,2536	-0,3976	1,0000	-0,1038	-0,0171	0,0780	0,0129	0,0480	-0,0618	-0,1021	0,0130	0,0870	-0,1178	0,0639	0,1080	0,0069	0,0316	-0,1140
VALOR_DEUDA	-0,1899	0,0809	-0,0170	-0,0069	-0,0012	0,0172	-0,0403	-0,0111	-0,0204	0,0206	-0,0171	-0,0398	1,0000	0,0163	0,0572	0,0756	-0,0962	-0,0341	0,0184	0,0120	0,0679	0,0724	-0,0162	-0,0126	0,0043	0,0193
DIAS_DEUDA	0,0516	-0,0101	-0,0216	-0,0151	-0,0030	0,0316	-0,0881	-0,0330	-0,1043	0,0359	0,0780	-0,1559	0,0163	1,0000	-0,0293	0,0861	-0,0969	-0,1507	0,0342	0,1101	-0,0310	0,0667	0,0883	-0,0116	0,0421	-0,0921
CICLO_PEQUEÑO	0,0283	-0,0121	0,0166	0,0025	-0,0046	-0,0177	0,0179	0,0292	-0,0226	0,0078	0,0129	0,0277	0,0572	-0,0293	1,0000	-0,2917	-0,3475	0,0398	-0,0115	-0,0260	-0,0277	0,0368	0,0002	0,0574	-0,0057	-0,0413
CICLO_MEDIANO	0,0379	-0,0051	-0,0304	-0,0117	-0,0019	0,0377	-0,1030	-0,0311	-0,0872	0,0373	0,0480	-0,2276	0,0756	0,0861	-0,2917	1,0000	-0,6171	-0,1016	-0,0311	0,1403	0,0143	0,0065	0,0563	-0,0340	0,0609	-0,0639
CICLO_DURO	-0,0754	0,0209	0,0296	0,0145	0,0042	-0,0375	0,1146	0,0106	0,1115	-0,0439	-0,0618	0,2601	-0,0962	-0,0969	-0,3475	-0,6171	1,0000	0,0834	0,0654	-0,1639	0,0001	-0,0390	-0,0850	0,0085	-0,0812	0,1221
FLUJO_SUAVE	-0,0114	0,0054	0,0145	0,0039	0,0089	-0,0283	0,0680	0,0964	0,1514	-0,0677	-0,1021	0,2054	-0,0941	-0,1507	0,0398	-0,1016	0,0834	1,0000	-0,6117	-0,2610	0,1051	-0,0288	-0,0572	0,0315	-0,0241	0,0382
FLUJO_MEDIANO	-0,0190	-0,0096	0,0450	0,0224	-0,0009	-0,0485	0,1183	-0,0358	-0,0250	0,0338	0,0870	-0,0941	0,0184	0,0342	-0,0115	-0,0311	0,0654	-0,6117	1,0000	-0,6940	-0,0887	-0,0266	0,0172	0,0105	-0,0022	-0,0200
FLUJO_DURO	0,0347	0,0086	-0,0696	-0,0313	-0,0079	0,0877	-0,2128	-0,0534	-0,1220	0,0269	0,0870	-0,4573	0,0120	0,1101	-0,0260	0,1403	-0,1639	-0,2610	-0,6940	1,0000	0,0024	0,0615	-0,0001	0,0367	-0,0140	0,0270
BOLETA_ELECT	-0,0654	0,0263	-0,0127	-0,0001	0,0032	0,0100	-0,2227	0,0643	0,1234	-0,0221	-0,1178	-0,0623	0,0679	0,1051	-0,0277	0,0143	0,0001	0,1051	-0,0887	0,0024	1,0000	-0,0838	-0,0001	-0,0100	-0,0583	-0,0615	0,0838
VALOR_FACT	0,0316	0,0086	-0,0213	-0,0203	0,0007	0,0240	-0,0396	-0,0139	-0,0813	0,0023	0,0639	-0,0799	0,0724	0,0667	0,0368	0,0065	-0,0390	-0,0288	-0,0266	0,0615	-0,0838	1,0000	-0,0100	-0,0143	0,0574	-0,0251
ZONA_NORTE	0,0249	-0,0116	0,0088	-0,0037	-0,0132	-0,0021	-0,0485	0,0839	-0,1021	0,0225	0,1080	-0,0456	-0,0162	0,0883	0,0002	0,0563	-0,0850	-0,0572	0,0172	0,0367	-0,0001	-0,0100	1,0000	-0,2321	-0,2215	-0,4254
ZONA_CENTRO	0,0038	-0,0084	0,0046	0,0009	0,0072	-0,0061	0,0579	-0,0882	-0,0242	0,0275	0,0069	0,0673	-0,0126	-0,0116	0,0574	0,0085	0,0315	0,0105	-0,0446	-0,0583	-0,0143	-0,2321	1,0000	-0,2276	-0,4370		
ZONA_SUR	0,0586	-0,0151	-0,0026	-0,0054	0,0069	0,0000	0,0221	-0,0068	-0,0566	-0,0676	0,0316	0,0239	0,0043	0,0421	-0,0057	0,0609	-0,0812	-0,0241	-0,0022	0,0270	-0,0615	0,0574	-0,2215	-0,2276	1,0000	-0,4171
ZONA_SANTIAGO	-0,0676	0,0273	-0,0085	0,0063	-0,0008	0,0065	-0,0250	0,0087	0,1425	0,0127	-0,1140	0,0006	0,0193	-0,0921	-0,0413	-0,0639	0,1221	0,0382	-0,0200	-0,0140	0,0938	-0,0251	-0,4254	-0,4370	-0,4171	1,0000

Anexo 28. Tabla de varianzas con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

	Varianzas
Svc_Mono	1,0039
Svc_Duo	1,0030
Svc_Triple	1,0035
PROF_1	1,0030
PROF_2	1,0042
PROF_3	0,9946
PROF_4	0,9953
PROF_5	1,0024
EDAD	1,0067
FLAG_VIP	1,0032
SEXO	0,9979
SECTOR_ALTO	1,0003
SECTOR_MEDIO	0,9873
SECTOR_BAJO	0,9900
GSE_COM	1,0027
DIAS_PERM	0,9956
VALOR_DEUDA	1,0051
DIAS_DEUDA	1,0018
CICLO_PEQUEÑO	1,0035
CICLO_MEDIANO	1,0053
CICLO_DURO	1,0007
FLUJO_SUAVE	0,9987
FLUJO_MEDIANO	1,0018
FLUJO_DURO	1,0016
BOLETA_ELECT	1,0051
VALOR_FACT	1,0007
ZONA_NORTE	0,9966
ZONA_CENTRO	0,9975
ZONA_SUR	1,0068
ZONA_SANTIAGO	1,0013

Anexo 29. Matriz de correlación de base con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

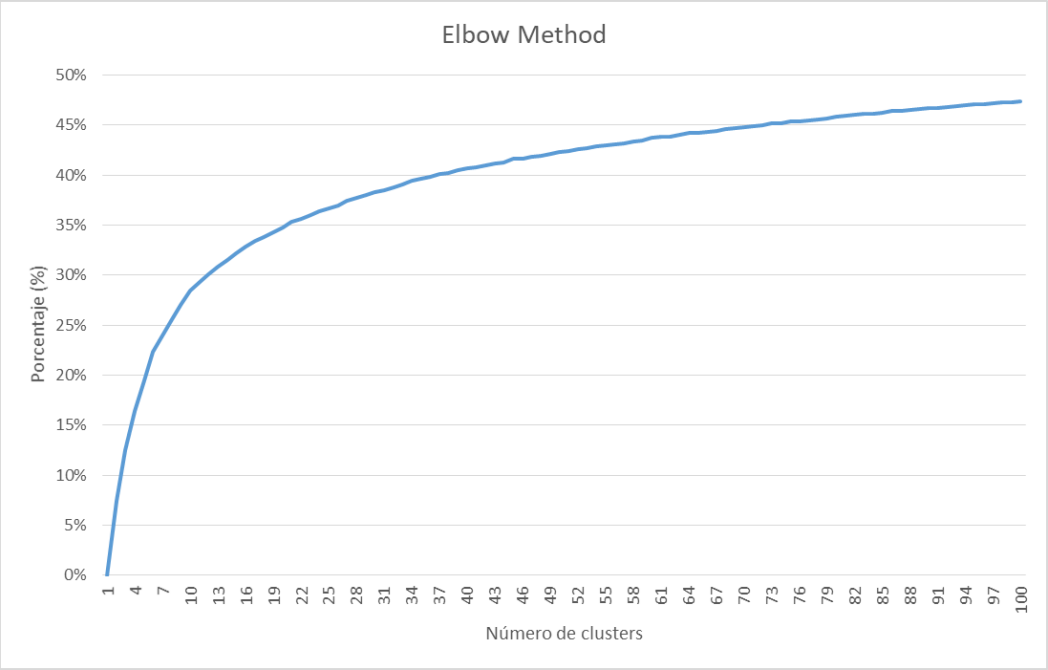
	Svc_Mono	Svc_Duo	Svc_Triple	PROF_1	PROF_2	PROF_3	PROF_4	PROF_5	EDAD	FLAG_VIP	SEXO	SECTOR_ALTO	SECTOR_MEDIO	SECTOR_BAJO	DIAS_PERM	VALOR_DELICIA	CICLO_MEDIANO	CICLO_DURO	FLUJO_SUAVE	FLUJO_MEDIANO	FLUJO_DURO	BOLETA_ELECT	VALOR_FACT	ZONA_NORTE	ZONA_CENTRO	ZONA_SUR	ZONA_SANTIAGO
Svc_Mono	1,0000	0,0012	-0,0005	-0,0023	-0,0015	0,0028	0,0036	0,0002	0,0025	-0,0002	0,0038	-0,0015	-0,0032	-0,0010	0,0023	-0,0002	-0,0004	0,0013	0,0052	-0,0006	0,0025	-0,0006	0,0056	0,0017	-0,0004	0,0029	0,0073	0,0016
Svc_Duo	0,0012	1,0000	0,0044	0,0018	0,0063	-0,0033	0,0036	-0,0061	0,0054	-0,0025	0,0027	0,0059	-0,0044	-0,0027	0,0062	-0,0002	-0,0002	-0,0029	0,0001	0,0019	0,0015	0,0019	0,0053	0,0070	-0,0067	-0,0058	0,0002	-0,0040
Svc_Triple	0,0005	0,0044	1,0000	0,0076	0,0031	0,0010	0,0012	0,0044	0,0044	0,0105	0,0091	0,0052	0,0018	-0,0034	-0,0063	-0,0002	0,0008	-0,0045	-0,0005	0,0000	-0,0001	-0,0022	-0,0046	-0,0007	0,0001	-0,0003	-0,0016	-0,0051
PROF_1	-0,0023	0,0018	0,0076	1,0000	0,0031	0,0010	0,0012	-0,0023	0,0033	-0,0011	-0,0044	-0,0065	-0,0020	0,0071	0,0024	0,0001	-0,0013	0,0005	-0,0058	0,0036	0,0021	-0,0061	0,0030	0,0038	-0,0038	0,0015	0,0003	0,0038
PROF_2	-0,0015	0,0063	-0,0042	0,0031	1,0000	0,0001	0,0062	-0,0009	0,0043	-0,0035	0,0027	-0,0015	0,0002	-0,0021	0,0021	-0,0002	-0,0025	-0,0015	0,0035	-0,0010	-0,0029	-0,0019	-0,0015	-0,0024	-0,0041	-0,0013	0,0024	0,0013
PROF_3	0,0028	-0,0033	0,0007	0,0010	0,0001	1,0000	-0,0009	0,0004	-0,0068	-0,0061	0,0007	-0,0064	-0,0008	0,0048	-0,0020	0,0001	0,0030	-0,0064	0,0012	0,0033	0,0034	0,0008	0,0008	-0,0046	0,0014	-0,0011	-0,0062	0,0009
PROF_4	0,0036	0,0036	0,0027	0,0012	0,0062	-0,0009	1,0000	0,0043	-0,0035	0,0027	-0,0015	0,0030	0,0019	0,0049	0,0021	-0,0002	-0,0055	0,0120	0,0002	0,0051	0,0005	-0,0003	0,0025	-0,0025	-0,0013	0,0024	0,0029	-0,0016
PROF_5	0,0002	-0,0061	-0,0042	-0,0023	-0,0006	0,0004	0,0021	1,0000	0,0009	-0,0061	0,0007	-0,0064	-0,0005	-0,0067	0,0020	-0,0002	-0,0044	0,0019	0,0015	-0,0007	-0,0039	-0,0029	-0,0001	0,0021	0,0002	0,0062	0,0029	-0,0050
EDAD	0,0025	0,0054	0,0044	0,0033	0,0010	-0,0068	0,0043	0,0009	1,0000	0,0013	-0,0059	0,0030	0,0043	0,0010	-0,0083	-0,0002	0,0001	0,0020	-0,0020	-0,0030	0,0044	-0,0095	0,0018	-0,0023	0,0023	0,0069	0,0012	-0,0027
FLAG_VIP	-0,0002	-0,0025	0,0105	-0,0011	0,0009	-0,0042	-0,0035	-0,0061	0,0013	1,0000	0,0078	0,0046	0,0013	-0,0027	-0,0049	0,0000	-0,0005	0,0000	-0,0013	-0,0030	0,0059	0,0006	-0,0009	-0,0001	-0,0054	-0,0090	0,0012	-0,0034
SEXO	0,0038	0,0027	0,0091	-0,0044	0,0008	-0,0004	0,0027	0,0007	-0,0059	0,0078	1,0000	0,0041	0,0002	0,0002	0,0011	-0,0001	-0,0014	-0,0003	0,0016	0,0065	-0,0038	0,0001	0,0038	0,0050	0,0038	0,0033	-0,0027	-0,0046
SECTOR_ALTO	-0,0015	0,0059	0,0052	-0,0065	0,0016	-0,0008	-0,0015	-0,0064	0,0030	0,0046	0,0041	1,0000	-0,0057	-0,0052	-0,0077	-0,0001	-0,0074	-0,0005	0,0037	0,0058	0,0035	0,0030	-0,0006	0,0056	-0,0036	0,0061	-0,0034	-0,0059
SECTOR_MEDIO	-0,0032	-0,0044	0,0018	-0,0020	0,0002	0,0038	-0,0025	-0,0005	0,0043	0,0013	-0,0016	-0,0057	1,0000	-0,0026	-0,0010	0,0001	0,0051	0,0042	0,0012	0,0069	-0,0031	0,0036	0,0013	-0,0019	-0,0033	-0,0010	0,0046	0,0006
SECTOR_BAJO	-0,0010	-0,0027	-0,0034	0,0071	-0,0001	0,0048	0,0019	-0,0067	0,0010	-0,0027	0,0002	-0,0052	-0,0026	1,0000	0,0001	0,0001	0,0048	0,0022	0,0087	-0,0003	-0,0032	0,0025	0,0034	-0,0007	0,0077	0,0036	-0,0015	-0,0009
DIAS_PERM	0,0023	0,0062	-0,0063	0,0024	-0,0021	-0,0020	0,0049	0,0020	-0,0083	-0,0049	0,0011	-0,0077	-0,0010	0,0001	1,0000	0,0002	0,0001	0,0002	-0,0040	0,0002	-0,0045	-0,0015	0,0032	-0,0035	-0,0049	-0,0044	-0,0002	0,0046
VALOR_DELICIA	-0,0018	-0,0053	-0,0036	0,0037	0,0029	0,0022	-0,0006	-0,0040	0,0013	0,0020	-0,0003	-0,0025	0,0024	0,0049	-0,0035	1,0000	-0,0012	0,0035	0,0008	-0,0028	-0,0002	-0,0004	0,0054	0,0051	0,0019	-0,0030	-0,0037	-0,0015
DIAS_CAMA	-0,0040	0,0022	0,0050	0,0021	0,0048	0,0045	0,0029	0,0007	-0,0037	-0,0033	-0,0007	-0,0004	-0,0038	-0,0025	0,0024	-0,0002	-0,0017	0,0004	0,0085	0,0003	0,0002	-0,0004	0,0027	0,0035	0,0048	-0,0011	0,0025	0,0035
CICLO_MEDIANO	-0,0027	0,0004	0,0014	-0,0012	0,0035	0,0037	-0,0004	-0,0037	0,0023	-0,0025	-0,0059	-0,0041	-0,0011	-0,0048	0,0007	-0,0002	1,0000	0,0041	0,0020	0,0033	0,0001	-0,0004	0,0003	0,0032	0,0046	0,0029	-0,0028	0,0035
CICLO_DURO	-0,0004	-0,0001	0,0004	-0,0013	0,0035	-0,0004	-0,0002	-0,0001	-0,0023	-0,0024	-0,0014	-0,0041	-0,0011	-0,0048	0,0007	-0,0002	0,0041	1,0000	0,0035	0,0053	0,0033	-0,0036	-0,0048	-0,0002	0,0009	0,0018	-0,0016	0,0035
FLUJO_SUAVE	0,0052	-0,0029	-0,0045	-0,0058	0,0035	0,0008	-0,0015	-0,0007	-0,0037	-0,0013	0,0016	0,0037	0,0042	0,0087	-0,0040	0,0002	0,0020	0,0035	1,0000	0,0035	-0,0016	-0,0039	-0,0040	-0,0033	0,0047	0,0009	-0,0018	0,0041
FLUJO_MEDIANO	0,0025	0,0015	0,0000	0,0036	-0,0010	0,0033	0,0051	-0,0007	-0,0030	-0,0030	0,0065	0,0058	0,0069	-0,0003	0,0002	-0,0002	0,0033	0,0053	0,0035	1,0000	0,0009	-0,0036	-0,0024	-0,0010	0,0082	-0,0038	0,0033	0,0041
FLUJO_DURO	-0,0006	0,0019	-0,0001	0,0021	-0,0029	0,0034	0,0005	-0,0039	0,0044	0,0059	-0,0038	0,0035	0,0012	-0,0032	0,0045	-0,0012	0,0001	0,0033	-0,0016	0,0009	1,0000	0,0016	-0,0015	-0,0006	-0,0018	-0,0042	-0,0008	0,0017
BOLETA_ELECT	0,0056	0,0053	-0,0022	-0,0061	-0,0019	0,0008	-0,0003	-0,0029	-0,0095	0,0006	0,0001	0,0030	0,0036	0,0025	-0,0015	-0,0017	-0,0004	-0,0036	-0,0039	-0,0036	0,0016	1,0000	0,0040	0,0012	-0,0018	0,0025	0,0018	0,0030
VALOR_FACT	0,0017	0,0070	-0,0046	0,0030	-0,0015	0,0008	0,0025	-0,0001	0,0018	-0,0009	0,0038	-0,0006	0,0013	0,0004	0,0032	0,0003	-0,0024	-0,0048	-0,0040	-0,0024	-0,0015	0,0040	1,0000	-0,0011	-0,0011	0,0063	-0,0031	0,0044
ZONA_NORTE	-0,0004	-0,0067	-0,0007	-0,0038	-0,0024	-0,0046	-0,0025	-0,0021	-0,0023	-0,0001	0,0050	0,0056	-0,0019	-0,0007	-0,0035	0,0001	-0,0032	-0,0002	-0,0033	-0,0010	-0,0006	-0,0012	-0,0011	1,0000	-0,0058	-0,0019	-0,0027	0,0025
ZONA_CENTRO	0,0029	-0,0058	0,0001	0,0015	-0,0041	0,0014	-0,0013	0,0002	0,0023	-0,0054	0,0038	-0,0036	0,0033	0,0077	-0,0049	0,0046	0,0046	0,0009	0,0047	0,0082	-0,0018	-0,0018	-0,0011	-0,0058	1,0000	0,0061	-0,0038	0,0046
ZONA_SUR	0,0073	0,0002	-0,0016	-0,0003	0,0013	-0,0011	0,0024	-0,0062	0,0069	-0,0090	0,0033	0,0061	-0,0010	0,0036	-0,0044	-0,0002	0,0029	0,0018	0,0009	-0,0038	-0,0042	0,0025	0,0063	-0,0019	0,0061	1,0000	-0,0031	0,0006
ZONA_SANTIAGO	0,0016	-0,0040	-0,0051	0,0038	-0,0015	0,0009	-0,0016	0,0029	-0,0050	0,0012	-0,0027	-0,0034	0,0046	0,0015	-0,0002	0,0	-0,0028	-0,0016	-0,0018	0,0033	-0,0008	0,0018	-0,0031	-0,0027	-0,0038	-0,0031	1,0000	

Anexo 30. Tabla de varianzas con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

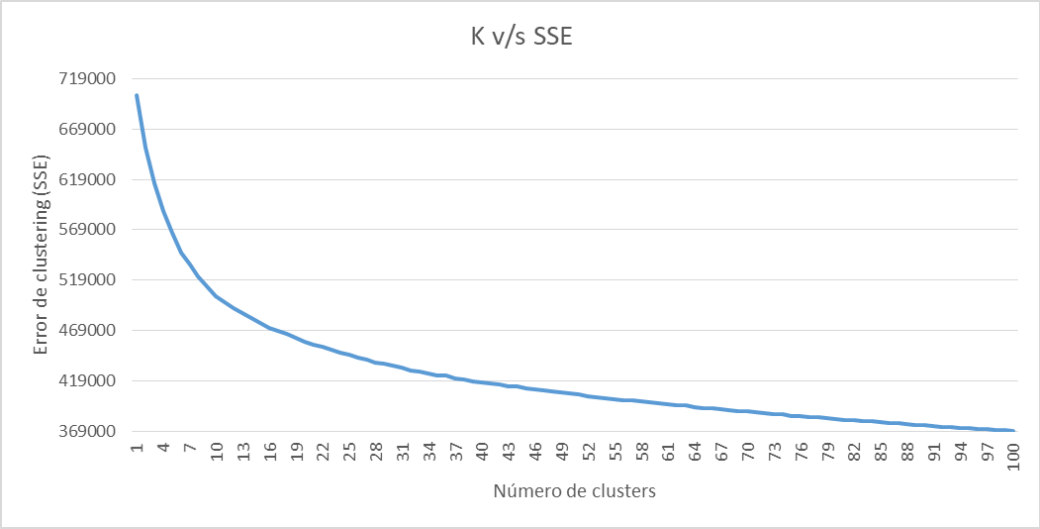
	Varianzas
Svc_Mono	1,0000
Svc_Duo	1,0000
Svc_Triple	1,0000
PROF_1	1,0000
PROF_2	1,0000
PROF_3	1,0000
PROF_4	1,0000
PROF_5	1,0000
EDAD	1,0000
FLAG_VIP	1,0000
SEXO	1,0000
SECTOR_ALTO	1,0000
SECTOR_MEDIO	1,0000
SECTOR_BAJO	1,0000
GSE_COM	1,0000
DIAS_PERM	1,0000
VALOR_DEUDA	1,4586
DIAS_DEUDA	0,1386
CICLO_PEQUEÑO	1,0000
CICLO_MEDIANO	1,0000
CICLO_GRANDE	1,0000
FLUJO_SUAVE	1,0000
FLUJO_MEDIANO	1,0000
FLUJO_DURO	1,0000
BOLETA_ELECT	1,0000
VALOR_FACT	1,3845
ZONA_NORTE	1,0000
ZONA_CENTRO	1,0000
ZONA_SUR	1,0000
ZONA_SANTIAGO	1,0000

Anexo 31. Matriz de correlación de base con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

Anexo 32. Número de *clusters* v/s tasa de explicabilidad (Fuente: Elaboración propia).



Anexo 33. Número de *clusters* v/s error del *clustering* (Fuente: Elaboración propia).

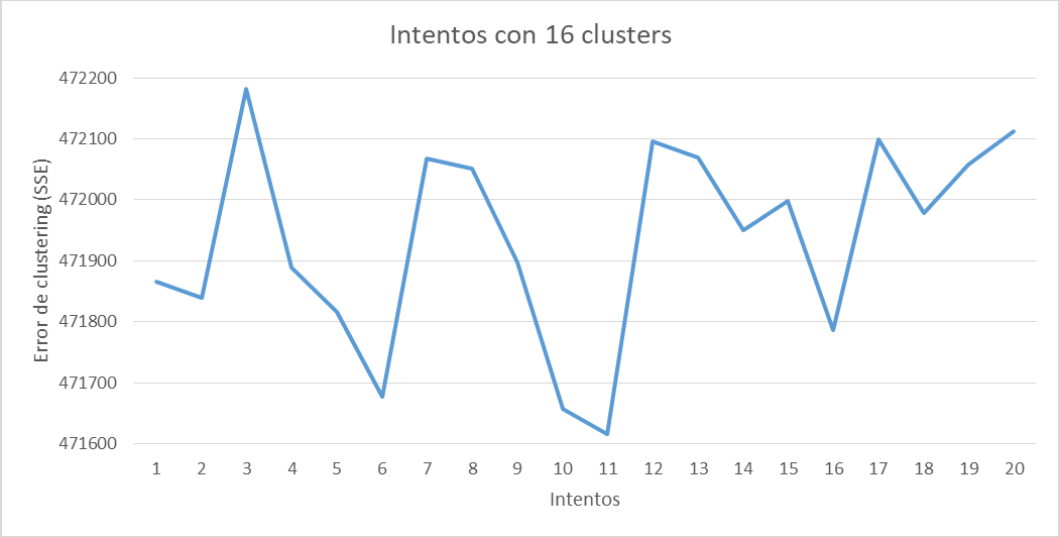


Anexo 34. Código utilizado para ejecutar *K – Means* en base con variables no binarias, normalizadas entre 0 y 1.

```
cl_base2_1_v1 <- kmeans(base2_1_v1_df[,c(-1)], 16, iter.max = 800, algorithm = "Lloyd")
#Aplica K - Means a la base especificada
cl_base2_1_v1
cl_base2_1_v1$tot.withinss
```

```
result_k1 <- data.frame(cbind(base2_1_v1_df, clusterNum = cl_base2_1_v1$cluster))
#Exportar base
write.csv(result_k1, file = "resultados_KMeans_base1_intento20.csv")
```

Anexo 35. Intentos de soluciones con *K – Means* (Fuente: Elaboración propia).



Anexo 36. Código utilizado para ejecutar el método de Ward en la base de clientes con variables no binarias, normalizadas entre 0 y 1.

```
#Se obtiene una muestra de la base total
muestreo <- sample(1:nrow(base2_1_v1_df), 385)
base_muestra <- base2_1_v1_df[c(muestreo),]
#Aplicando método de Ward
d <- dist(base_muestra, method = "euclidean")
ward_base1 <- hclust(d, method = "ward.D")
#Graficando dendrograma
plot(ward_base1)
groups <- cutree(ward_base1, k = 26) #Aquí, se especifican cuántos clusters existen en el dendrograma.
rect.hclust(ward_base1, k = 26, border = "red")
#Identificando los datos de cada cluster en la muestra
result_ward1_muestra <- data.frame(cbind(base_muestra, clusterNum = groups))
#Calcular SSE
```

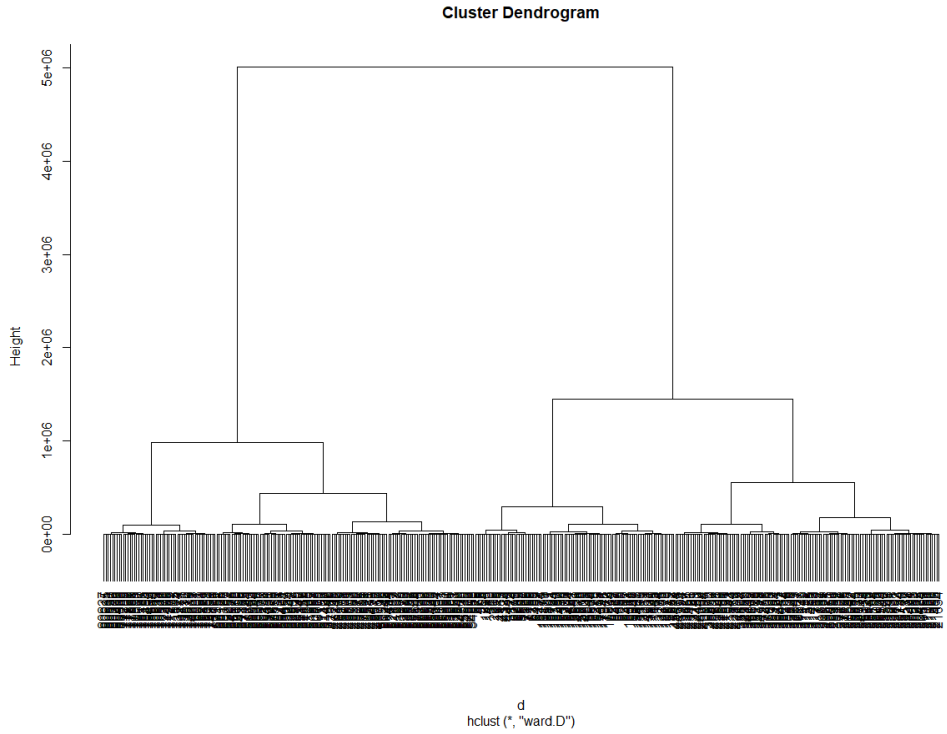
```
clusters_seq <- seq(1, 26)
sse_final <- rep(0, 26) #Cantidad de clusters
for (x in clusters_seq) {
 datos_cluster <- result_ward1_muestra[result_ward1_muestra$clusterNum==x,]
 centroide <- rep(0, 28)
 valor_seq_Ward <- seq(2, 29)
 for (y in valor_seq_Ward) {
  centroide[y - 1] <- mean(datos_cluster[,y])
 }
 filas <- nrow(datos_cluster)
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
  sse_variables <- rep(0, 28)
  for (w in valor_seq_Ward) {
   sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
  }
  sse_valores[z] <- sum(sse_variables)
 }
 sse_final[x] <- sum(sse_valores)
}
sse <- sum(sse_final)
sse
#Asignar clusters a cada dato de la base
centroides_ward <- matrix(nrow = 26, ncol = 28, rep(0, 728), byrow = TRUE) #El número
de filas son los clusters y los ceros son filas por columnas.
nfilas_centroides_ward <- nrow(centroides_ward)
seq_filas <- seq(1, nfilas_centroides_ward)
for (x in seq_filas) {
```

```
  datos_cluster <- result_ward1_muestra[result_ward1_muestra$clusterNum==x,]
  valor_seq_Ward <- seq(2, 29)
  for (y in valor_seq_Ward) {
    centroides_ward[x,y - 1] <- mean(datos_cluster[,y])
  }
}
seq_base <- seq(1, nrow(base2_1_v1_df))
seq_muestra <- seq(1, nrow(base_muestra))
cluster_base <- rep(0, 81107)
for (x in seq_base) {
  for (y in seq_muestra) {
    num_muestra <- base_muestra[y,1]
    if (num_muestra == x) {
      cluster_base[x] <- result_ward1_muestra$clusterNum[y]
      break
    }
  }
  dist_clusters <- rep(0, 26) #Colocar número de clusters
  seq_clust <- seq(1, 26) #Colocar número de clusters
  for (z in seq_clust) {
    distancias <- rep(0, 28)
    seq_dist <- seq(1, 28)
    for (xx in seq_dist) {
      distancias[xx] <- (base2_1_v1_df[x,xx + 1] - centroides_ward[z,xx])^2
    }
    sum_distancias <- sum(distancias)
    dist_clusters[z] <- sqrt(sum_distancias)
  }
  min_dist <- min(dist_clusters)
```

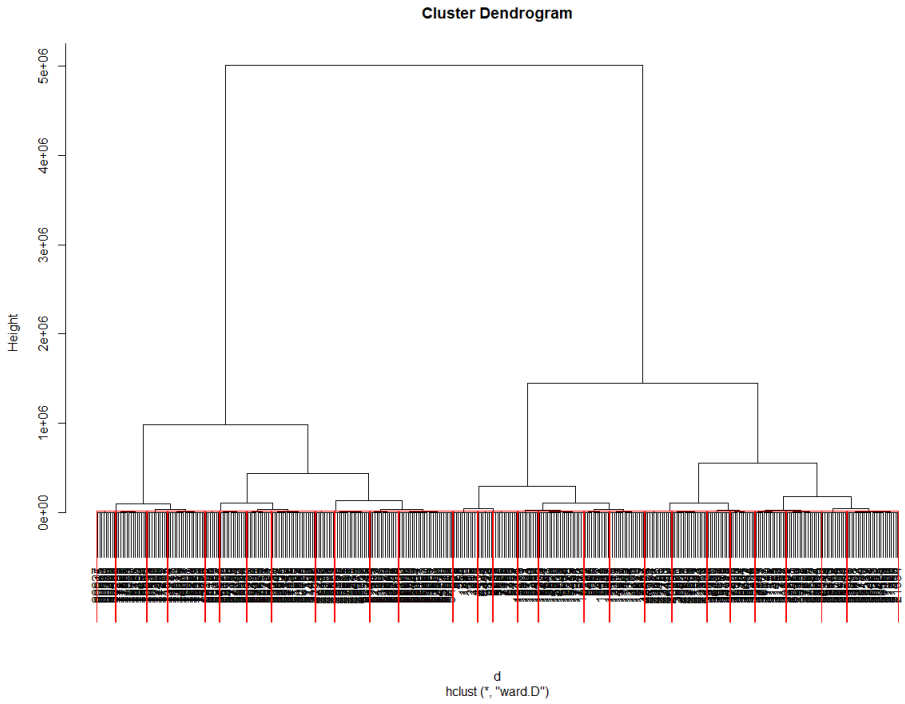
```
  for (zz in seq_clust) {
    valor_dist <- dist_clusters[zz]
    if (valor_dist == min_dist) {
      cluster_base[x] <- zz
      break
    }
  }
}
#Unir base total con etiquetas de cluster
result_ward1_total <- data.frame(cbind(base2_1_v1_df, clusterNum = cluster_base))
#Calcular SSE de base total
sse_final_total <- rep(0, 26) #Colocar cantidad de clusters
for (x in clusters_seq) {
  datos_cluster2 <- result_ward1_total[result_ward1_total$clusterNum==x,]
  centroide2 <- rep(0, 28)
  valor_seq_Ward2 <- seq(2, 29)
  for (y in valor_seq_Ward2) {
    centroide2[y - 1] <- mean(datos_cluster2[,y])
  }
  filas2 <- nrow(datos_cluster2)
  valor_filas2 <- seq(1, filas2)
  sse_valores2 <- rep(0, filas2)
  for (z in valor_filas2) {
    sse_variables2 <- rep(0, 28)
    for (w in valor_seq_Ward2) {
      sse_variables2 [w - 1] <- (datos_cluster2[z,w] - centroide2[w - 1])^2
    }
    sse_valores2[z] <- sum(sse_variables2)
  }
```

```
  sse_final_total[x] <- sum(sse_valores2)
}
sse_total <- sum(sse_final_total)
sse_total
```

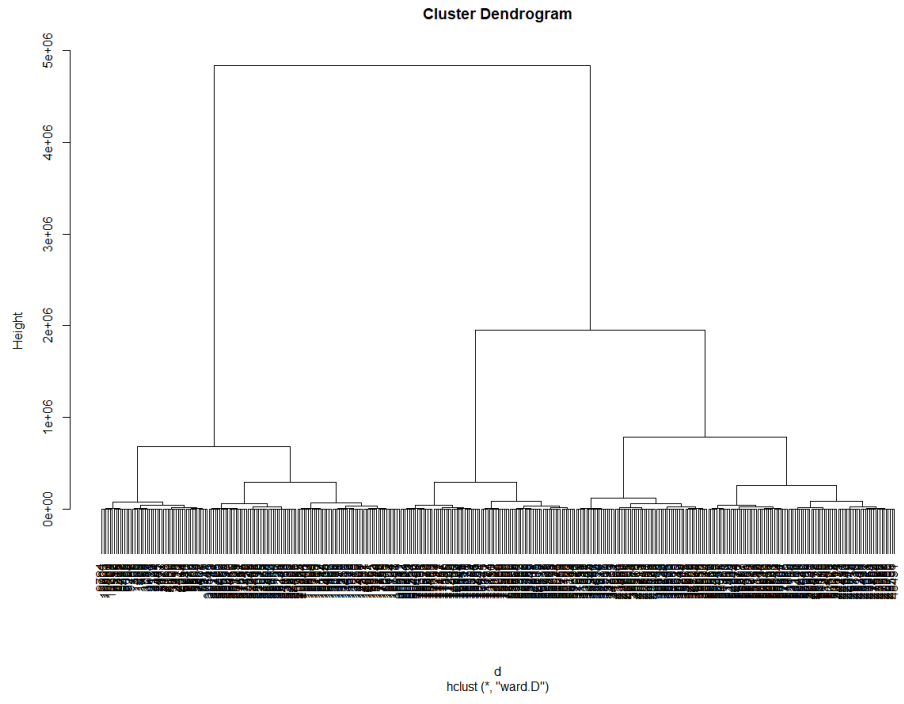
Anexo 37. Dendrograma intento 1 para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



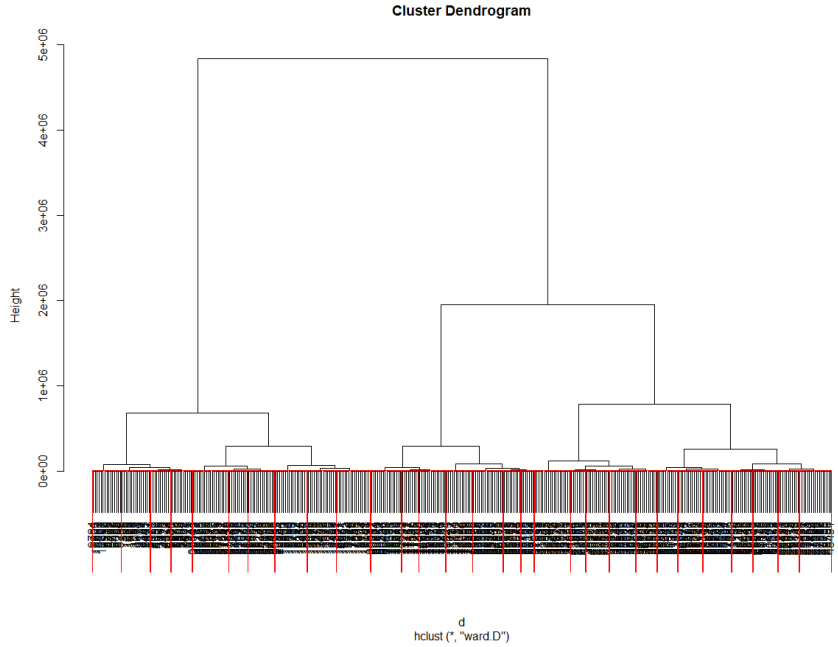
Anexo 38. Dendrograma intento 1 con 27 *clusters* identificados, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



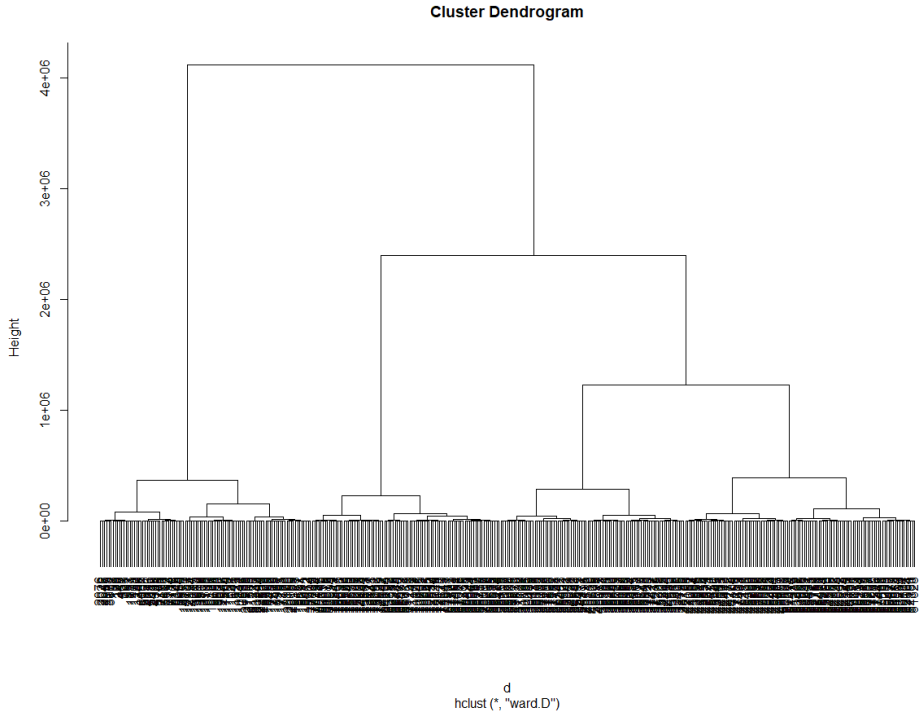
Anexo 39. Dendrograma intento 2 para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



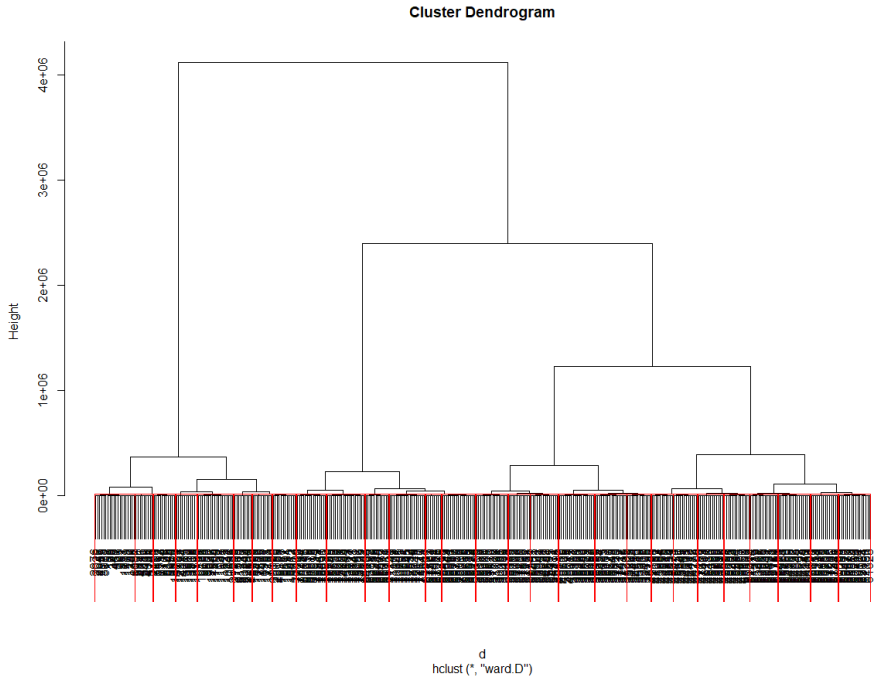
Anexo 40. Dendrograma intento 2 con 29 *clusters* identificados, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



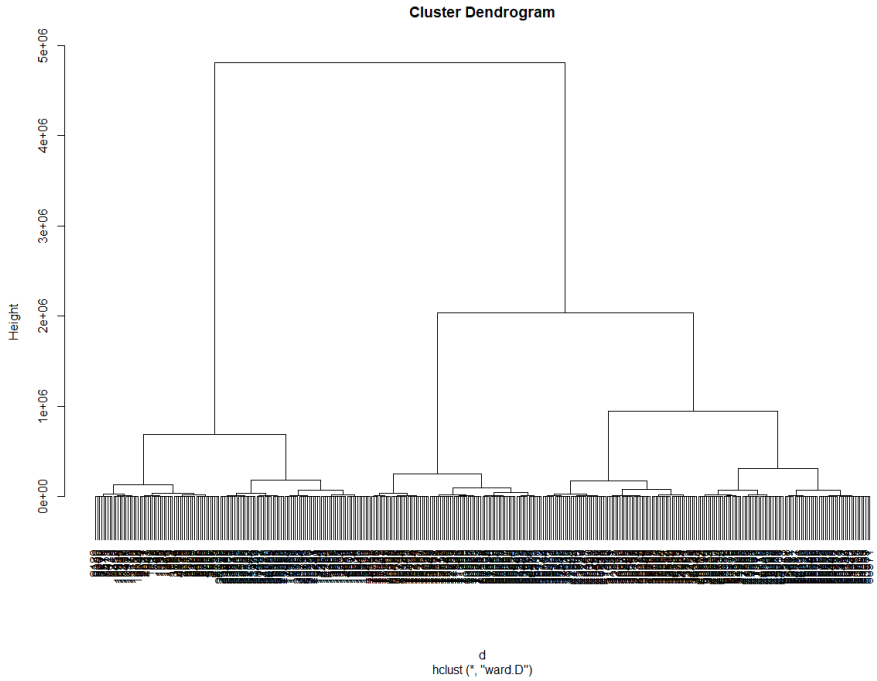
Anexo 41. Dendrograma intento 3 para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



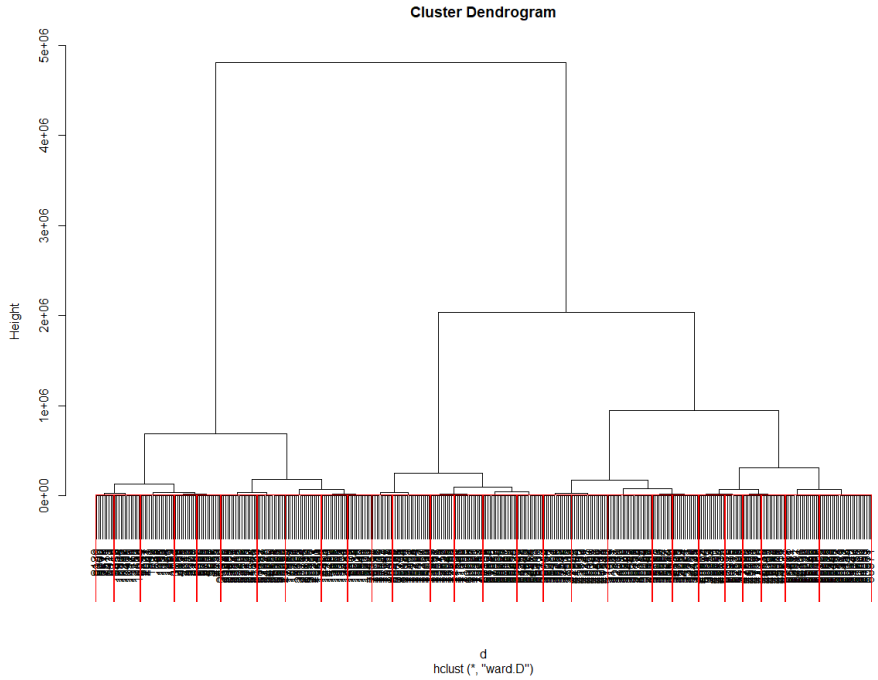
Anexo 42. Dendrograma intento 3 con 28 *clusters* identificados, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



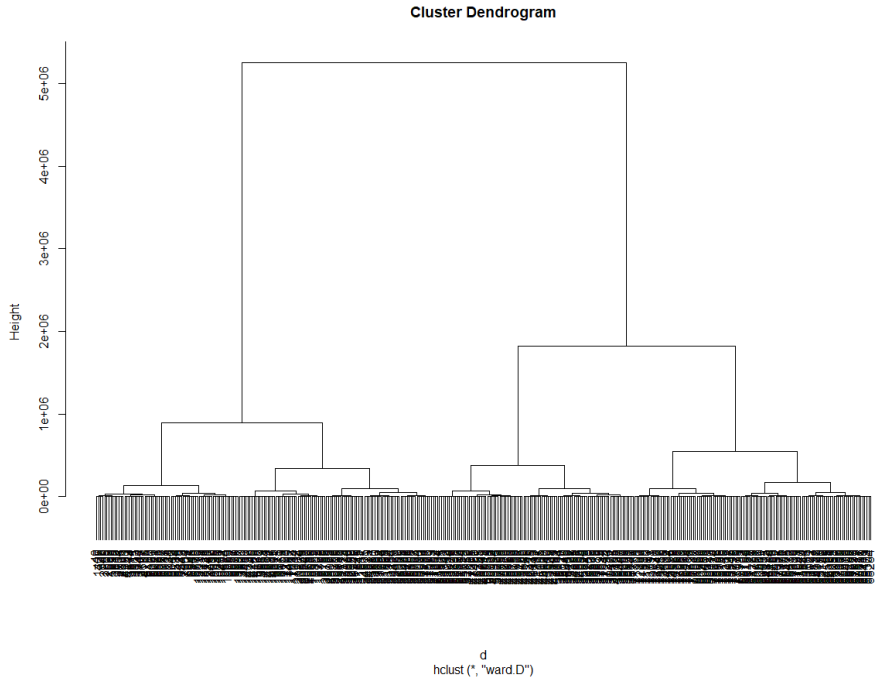
Anexo 43. Dendrograma intento 4 para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



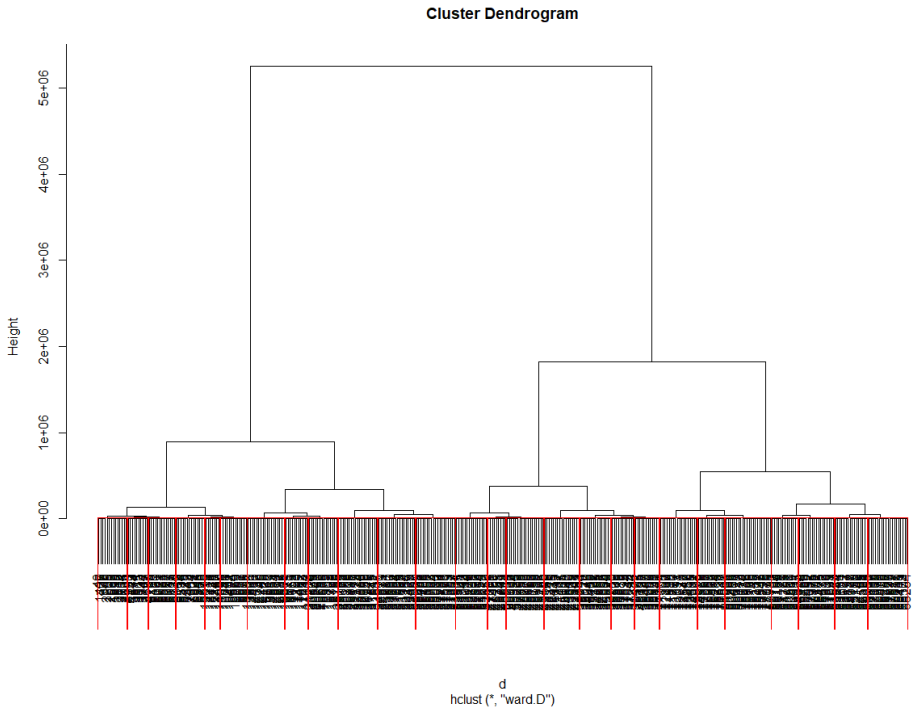
Anexo 44. Dendrograma intento 4 con 27 *clusters* identificados, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



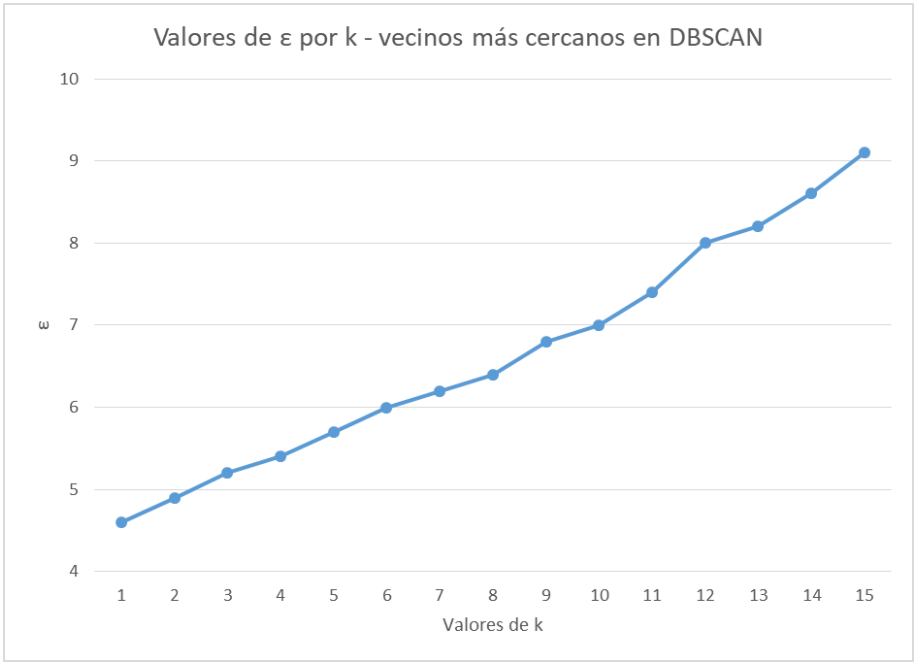
Anexo 45. Dendrograma intento 5 para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



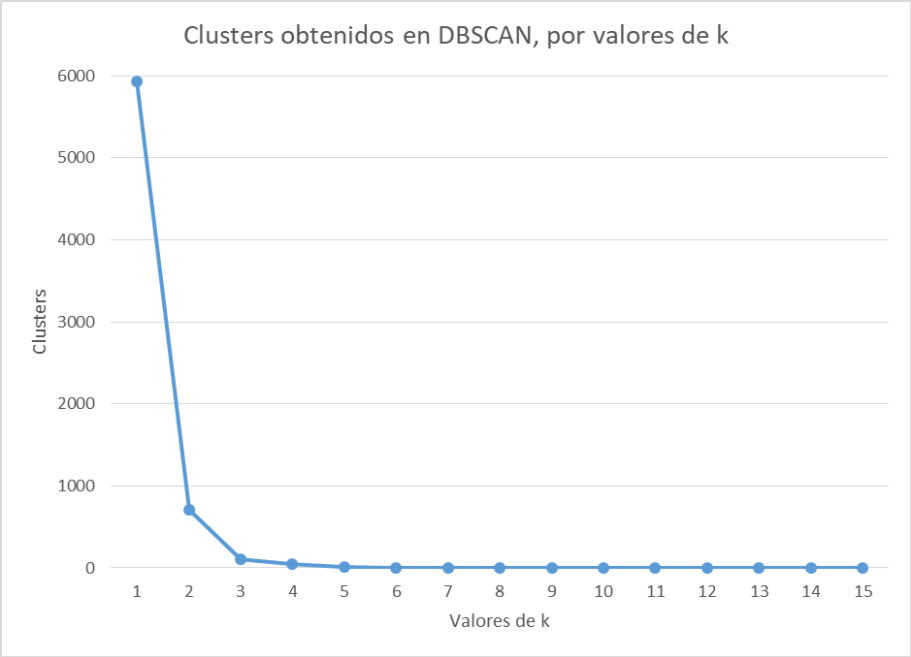
Anexo 46. Dendrograma intento 6 con 26 *clusters* identificados, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



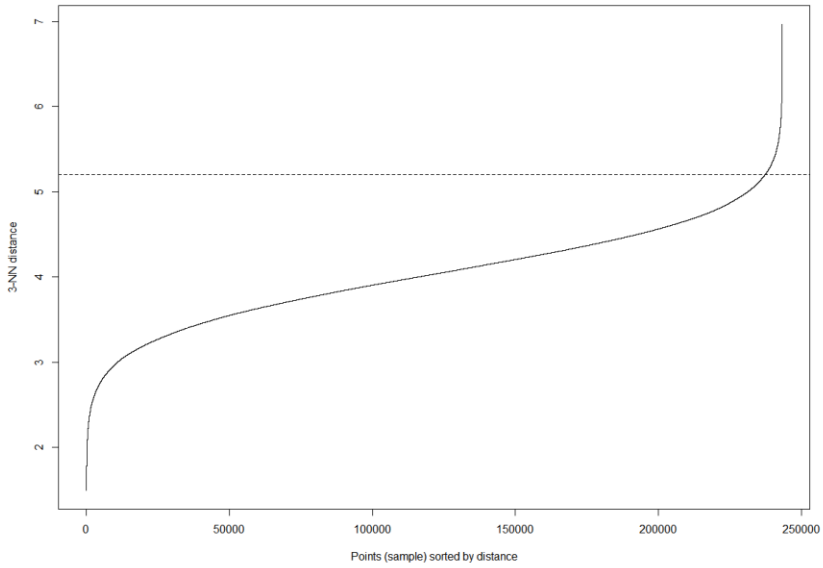
Anexo 47. Valores de épsilon (ε) por k – vecinos más cercanos, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



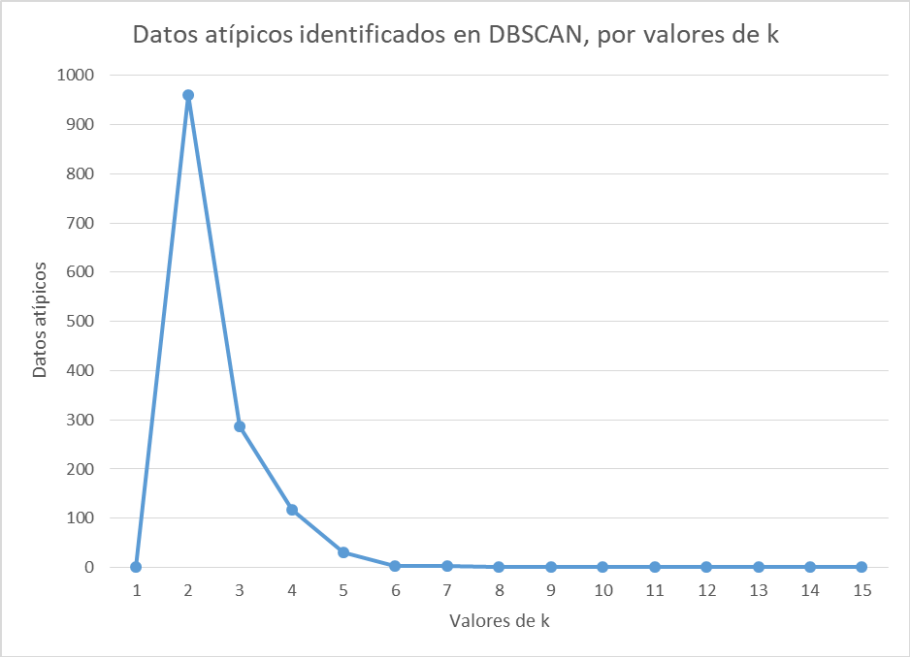
Anexo 48. Cantidad de *clusters* obtenidos en DBSCAN, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



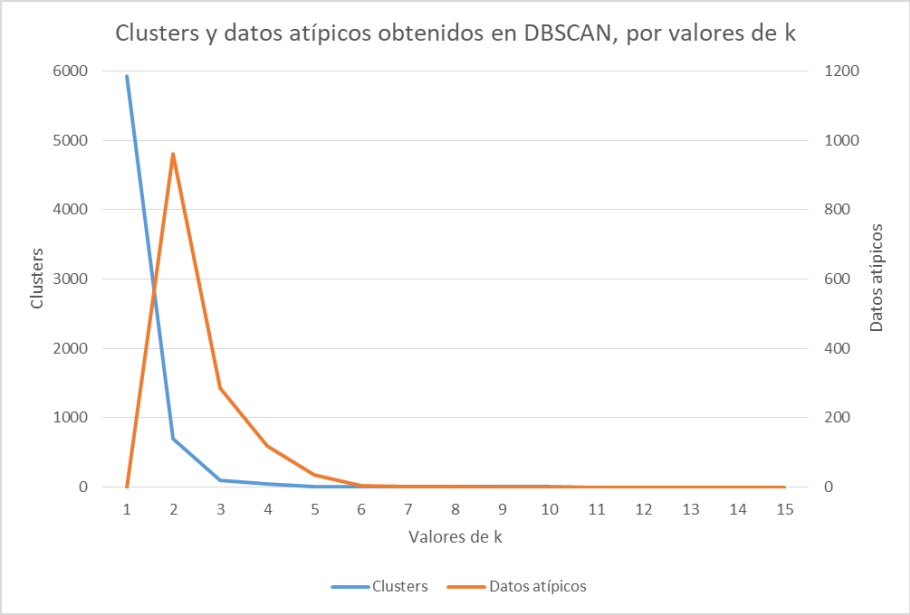
Anexo 49. Gráfico de distancias de k – vecinos más cercanos, con *k* = 3, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



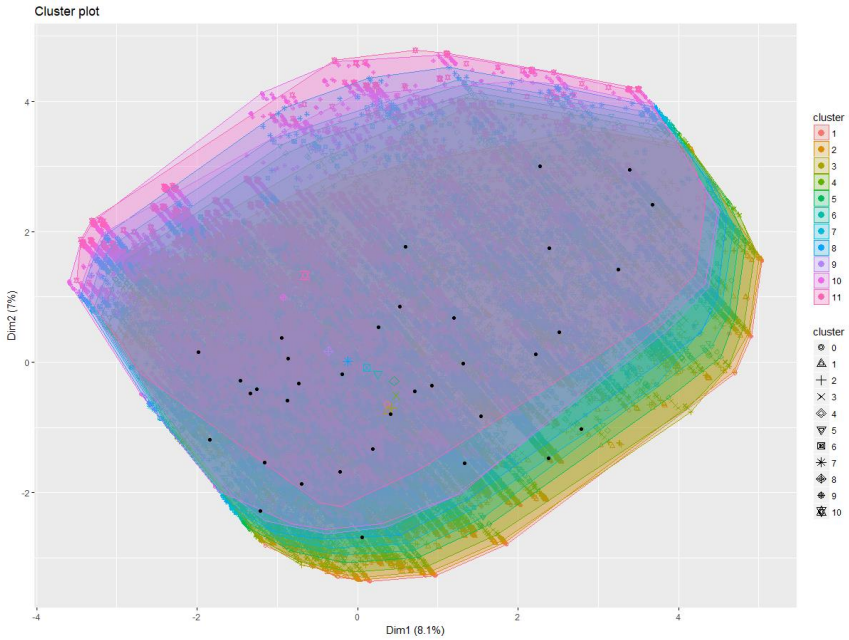
Anexo 50. Cantidad de datos atípicos para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 51. Gráficos de *clusters* obtenidos y cantidad de datos atípicos identificados en DBSCAN, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 52. *Clusters* obtenidos por DBSCAN para base con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 53. Código utilizado para ejecutar DBSCAN, en la base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

```
#Primero, se identificará valor de epsilon adecuado
library('dbscan')
library('factoextra')
dbscan::kNNdistplot(base2_1_v1_df, k = 3)
abline(h = 5.2, lty = 2)
#Aplicando DBSCAN
res.db <- dbscan(base2_1_v1_df, 5.7, 5)
res.db
#Graficando
fviz_cluster(res.db, base2_1_v1_df, geom = "point")
#Descargando base
result_dbscan1 <- data.frame(cbind(base2_1_v1_df, clusterNum = res.db$cluster))
write.csv(result_dbscan1, file = "resultados_DBSCAN_base1.csv")
```

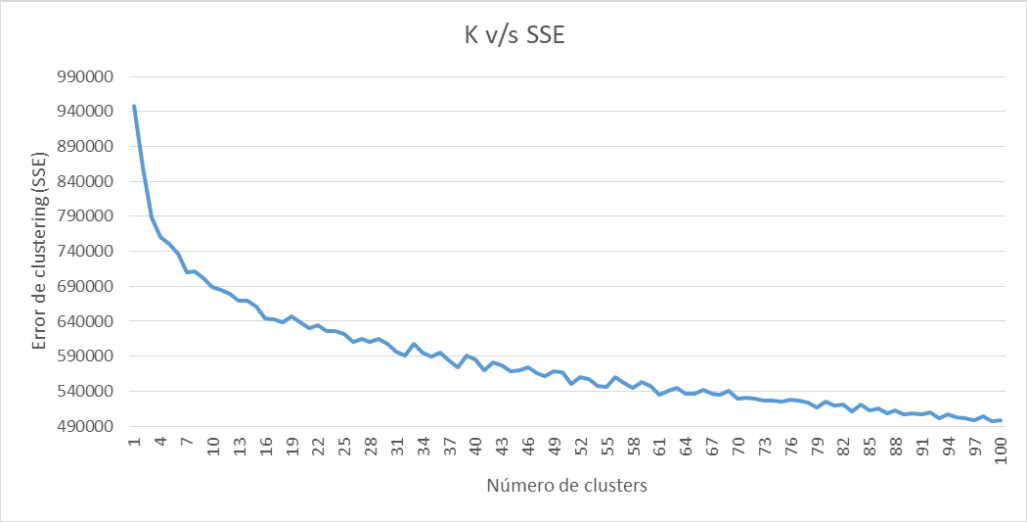
Anexo 54. Código utilizado para ejecutar *K – Medoids*, en la base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

```
library(cluster)
result_kmedoids1 <- clara(base2_1_v1_df[,c(-1)], 12, samples = 55)
result_kmedoids1$i.med
#Unir base con índices de clusters
result_db_kmedoids1 <- data.frame(cbind(base2_1_v1_df, clusterNum = result_kmedoids1$clustering))
#Calcular SSE
clusters_kmedoids_seq <- seq(1, 12)
sse_kmedoids_final <- rep(0, 12)
for (x in clusters_kmedoids_seq) {
 datos_cluster <- result_db_kmedoids1[result_db_kmedoids1$clusterNum==x,]
 centroide <- result_db_kmedoids1[result_kmedoids1$i.med[x],seq(2,29)]
 valor_seq_Ward <- seq(2, 29)
 filas <- nrow(datos_cluster)
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
  sse_variables <- rep(0, 28)
  for (w in valor_seq_Ward) {
   sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
  }
  sse_valores[z] <- sum(sse_variables)
 }
 sse_kmedoids_final[x] <- sum(sse_valores)
}
sse <- sum(sse_kmedoids_final)
sse
```

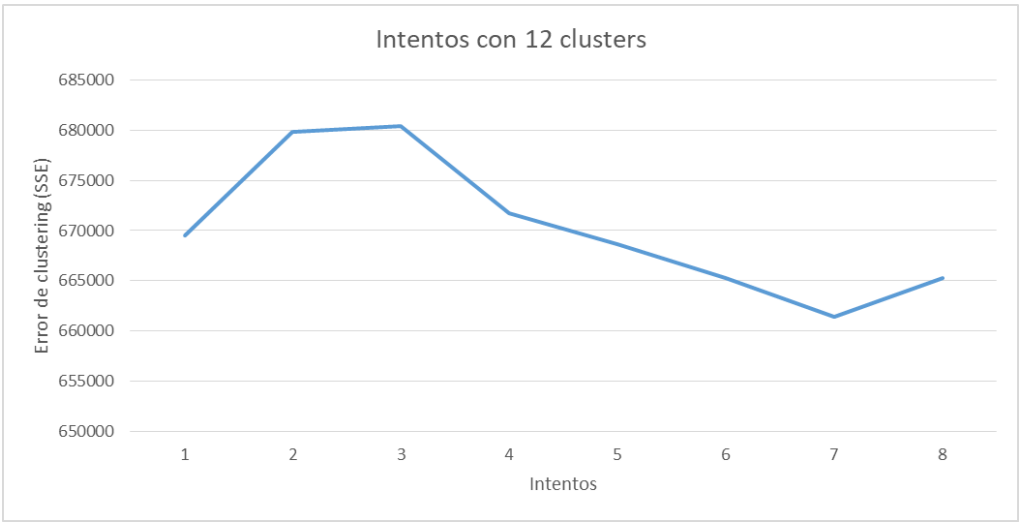
#Descargar base

write.csv(result_db_kmedoids1, file = "resultados_KMedoids_base1_intento7_661378-2.csv")

Anexo 55. Número de *clusters* v/s error de *K – Medoids* para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 56. Distintos intentos de soluciones con *K – Medoids*, para base de clientes con variables no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 57. Código utilizado para ejecutar *K – Means* para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

cl_base2_2_v1 <- kmeans(base2_2_v1_v2_df[,c(-1)], 15, iter.max = 800, algorithm = "Lloyd") #Aplica K - Means a la base especificada

```
cl_base2_2_v1
cl_base2_2_v1$tot.withinss
result_k2 <- data.frame(cbind(base2_2_v1_v2_df, clusterNum = cl_base2_2_v1$cluster))
write.csv(result_k2, file = "resultados_KMeans_base2_intento20.csv")
```

Anexo 58. Código utilizado para la ejecución del método de Ward, para base de clientes con variables no binarias, estandarizadas con puntaje Z.

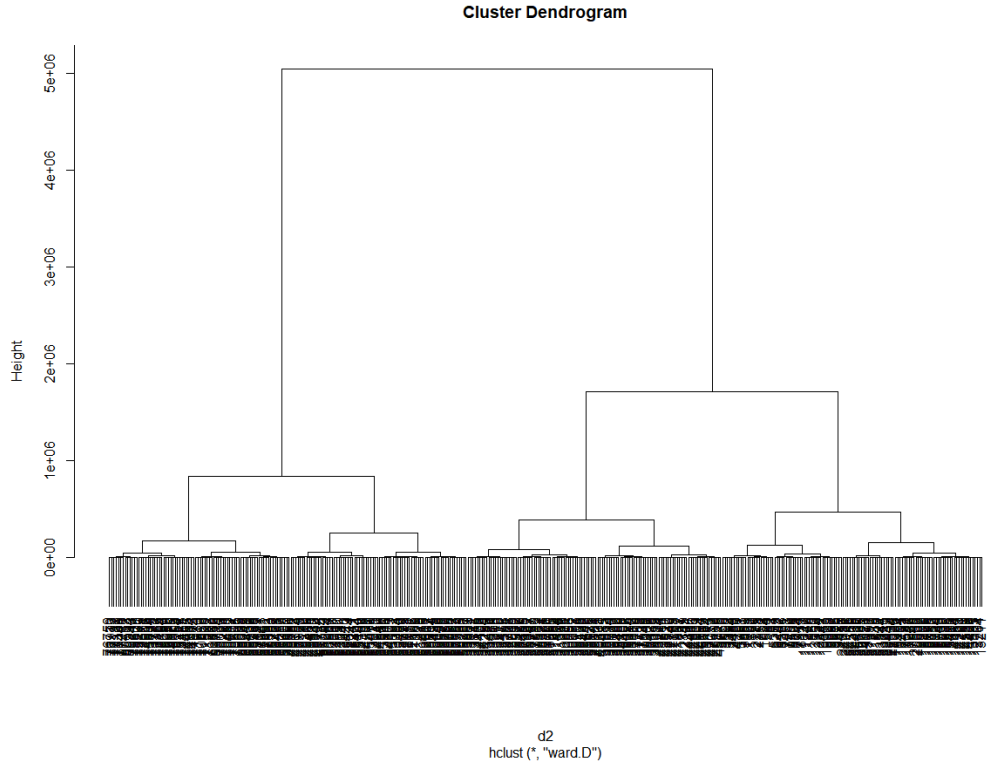
```
#Se obtiene una muestra de la base total
muestreo2 <- sample(1:nrow(base2_2_v1_v2_df), 385)
base_muestra2 <- base2_2_v1_v2_df[c(muestreo2),]
#Aplicando método de Ward
d2 <- dist(base_muestra2, method = "euclidean")
ward_base2 <- hclust(d2, method = "ward.D")
#Graficando dendrograma
plot(ward_base2)
groups <- cutree(ward_base2, k = 29) #Aquí, se especifican cuántos clusters existen en el dendrograma.
rect.hclust(ward_base2, k = 29, border = "red")
#Identificando los datos de cada cluster en la muestra
result_ward2_muestra <- data.frame(cbind(base_muestra2, clusterNum = groups))
#Calcular SSE
clusters_seq2 <- seq(1, 29) #Cantidad de clusters
sse_final2 <- rep(0, 29) #Cantidad de clusters
for (x in clusters_seq2) {
 datos_cluster <- result_ward2_muestra[result_ward2_muestra$clusterNum==x,]
 centroide <- rep(0, 26)
 valor_seq_Ward <- seq(2, 27)
 for (y in valor_seq_Ward) {
  centroide[y - 1] <- mean(datos_cluster[,y])
```

```
  }
  filas <- nrow(datos_cluster)
  valor_filas <- seq(1, filas)
  sse_valores <- rep(0, filas)
  for (z in valor_filas) {
    sse_variables <- rep(0, 26)
    for (w in valor_seq_Ward) {
      sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
    }
    sse_valores[z] <- sum(sse_variables)
  }
  sse_final2[x] <- sum(sse_valores)
}
sse2 <- sum(sse_final2)
sse2
#Asignar clusters a cada dato de la base
centroides_ward2 <- matrix(nrow = 29, ncol = 26, rep(0, 754), byrow = TRUE) #El número
de filas son los clusters y los ceros son filas por columnas.
nfilas_centroides_ward2 <- nrow(centroides_ward2)
seq_filas2 <- seq(1, nfilas_centroides_ward2)
for (x in seq_filas2) {
  datos_cluster <- result_ward2_muestra[result_ward2_muestra$clusterNum==x,]
  valor_seq_Ward <- seq(2, 27)
  for (y in valor_seq_Ward) {
    centroides_ward2[x,y - 1] <- mean(datos_cluster[,y])
  }
}
seq_base2 <- seq(1, nrow(base2_2_v1_v2_df))
seq_muestra2 <- seq(1, nrow(base_muestra2))
```

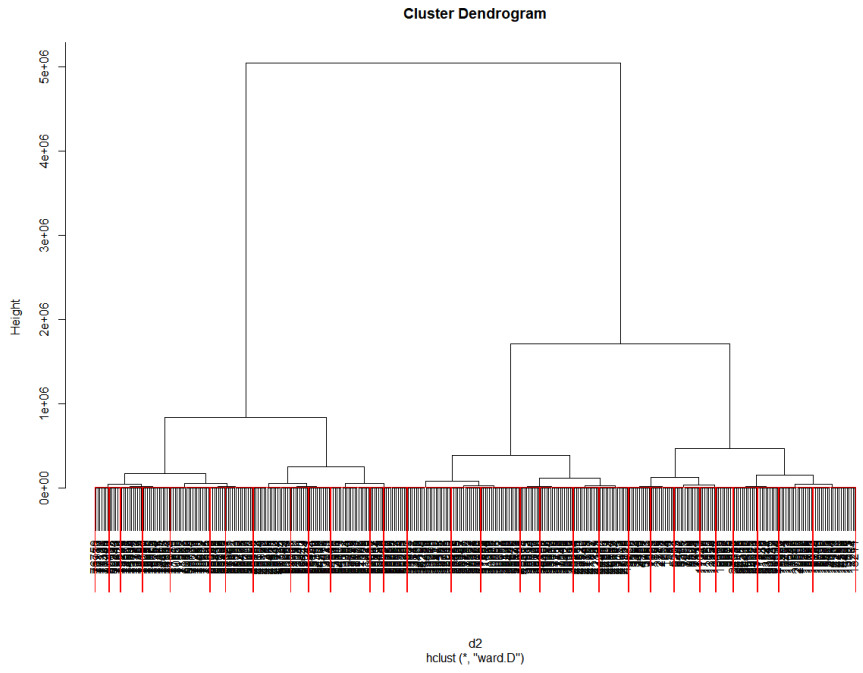
```
cluster_base2 <- rep(0, 81107)
for (x in seq_base2) {
  for (y in seq_muestra2) {
    num_muestra <- base_muestra2[y,1]
    if (num_muestra == x) {
      cluster_base2[x] <- result_ward2_muestra$clusterNum[y]
      break
    }
  }
  dist_clusters <- rep(0, 29) #Colocar número de clusters
  seq_clust <- seq(1, 29) #Colocar número de clusters
  for (z in seq_clust) {
    distancias <- rep(0, 26)
    seq_dist <- seq(1, 26)
    for (xx in seq_dist) {
      distancias[xx] <- (base2_2_v1_v2_df[x,xx + 1] - centroides_ward2[z,xx])^2
    }
    sum_distancias <- sum(distancias)
    dist_clusters[z] <- sqrt(sum_distancias)
  }
  min_dist <- min(dist_clusters)
  for (zz in seq_clust) {
    valor_dist <- dist_clusters[zz]
    if (valor_dist == min_dist) {
      cluster_base2[x] <- zz
      break
    }
  }
}
```

```
#Unir base total con etiquetas de cluster
result_ward2_total <- data.frame(cbind(base2_2_v1_v2_df, clusterNum = cluster_base2))
#Calcular SSE de base total
sse_final_total2 <- rep(0, 29) #Colocar cantidad de clusters
for (x in clusters_seq2) {
 datos_cluster2 <- result_ward2_total[result_ward2_total$clusterNum==x,]
 centroide2 <- rep(0, 26)
 valor_seq_Ward2 <- seq(2, 27)
 for (y in valor_seq_Ward2) {
  centroide2[y - 1] <- mean(datos_cluster2[,y])
 }
 filas2 <- nrow(datos_cluster2)
 valor_filas2 <- seq(1, filas2)
 sse_valores2 <- rep(0, filas2)
 for (z in valor_filas2) {
  sse_variables2 <- rep(0, 26)
  for (w in valor_seq_Ward2) {
   sse_variables2 [w - 1] <- (datos_cluster2[z,w] - centroide2[w - 1])^2
  }
  sse_valores2[z] <- sum(sse_variables2)
 }
 sse_final_total2[x] <- sum(sse_valores2)
}
sse_total2 <- sum(sse_final_total2)
sse_total2
#Exportar archivo
write.csv(result_ward2_total, file = "resultados_Ward_base2_intento5_405855-0.csv")
```

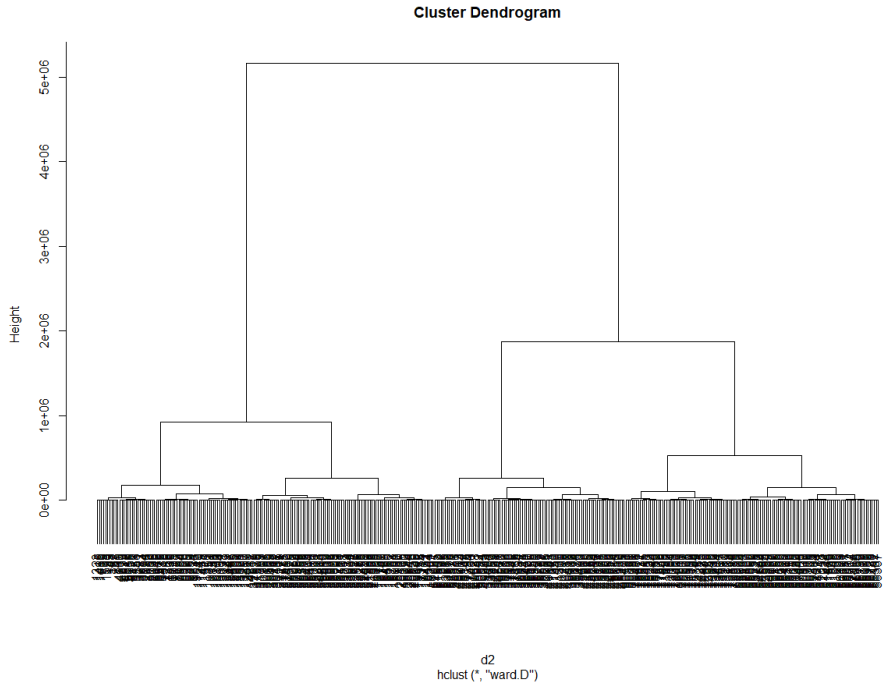
Anexo 59. Dendrograma intento 1 para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



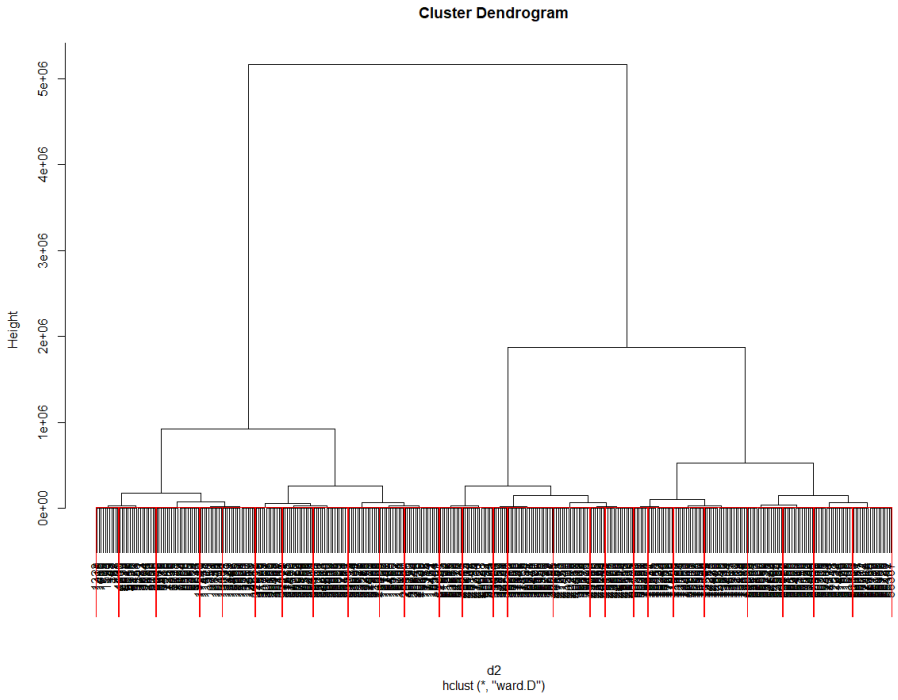
Anexo 60. Dendrograma intento 1 con 29 *clusters* identificados, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



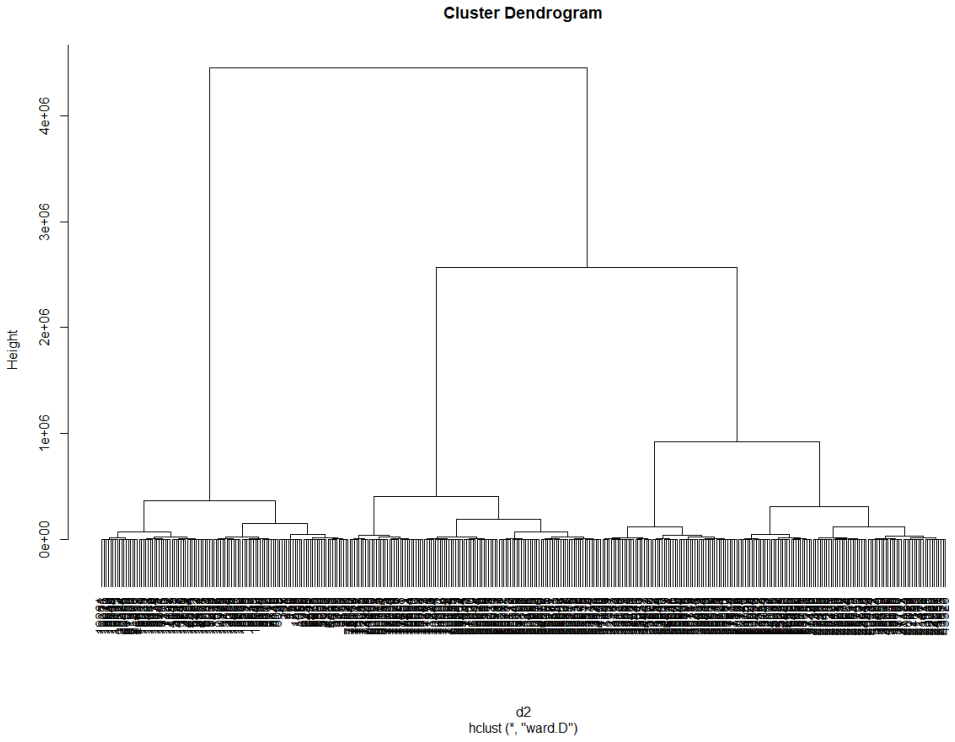
Anexo 61. Dendrograma intento 2 para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



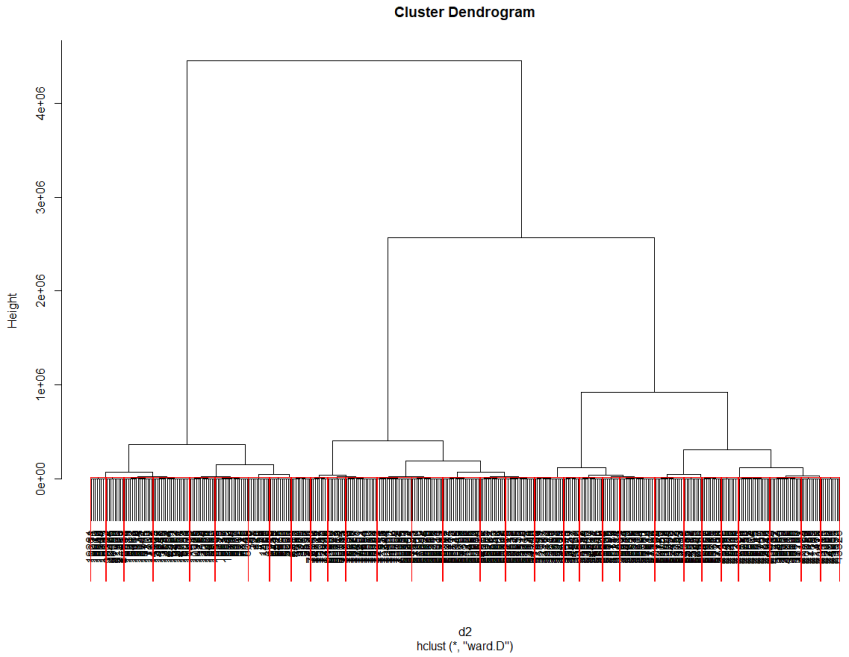
Anexo 62. Dendrograma intento 2 con 26 *clusters* identificados, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



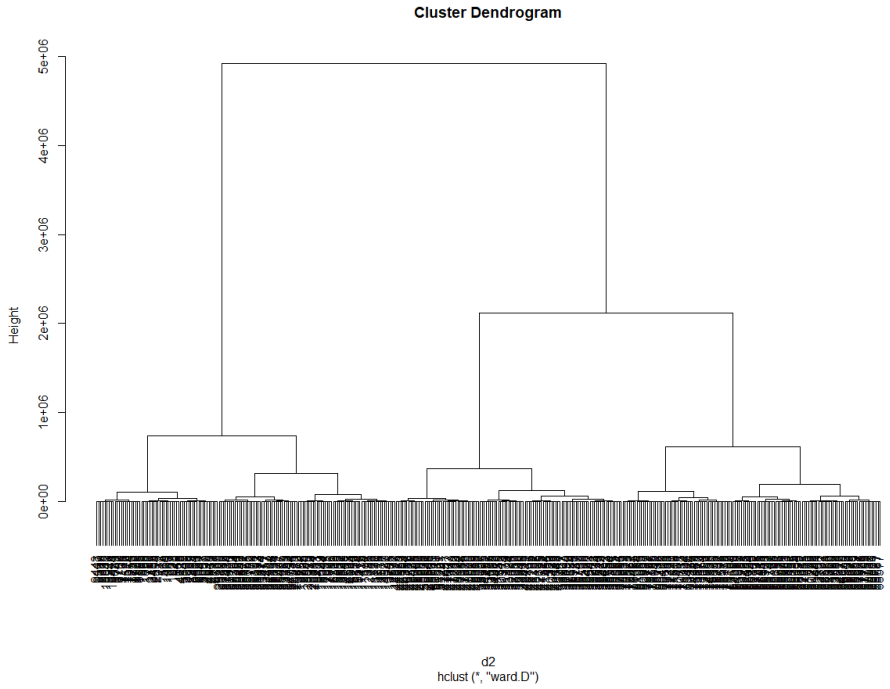
Anexo 63. Dendrograma intento 3 para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



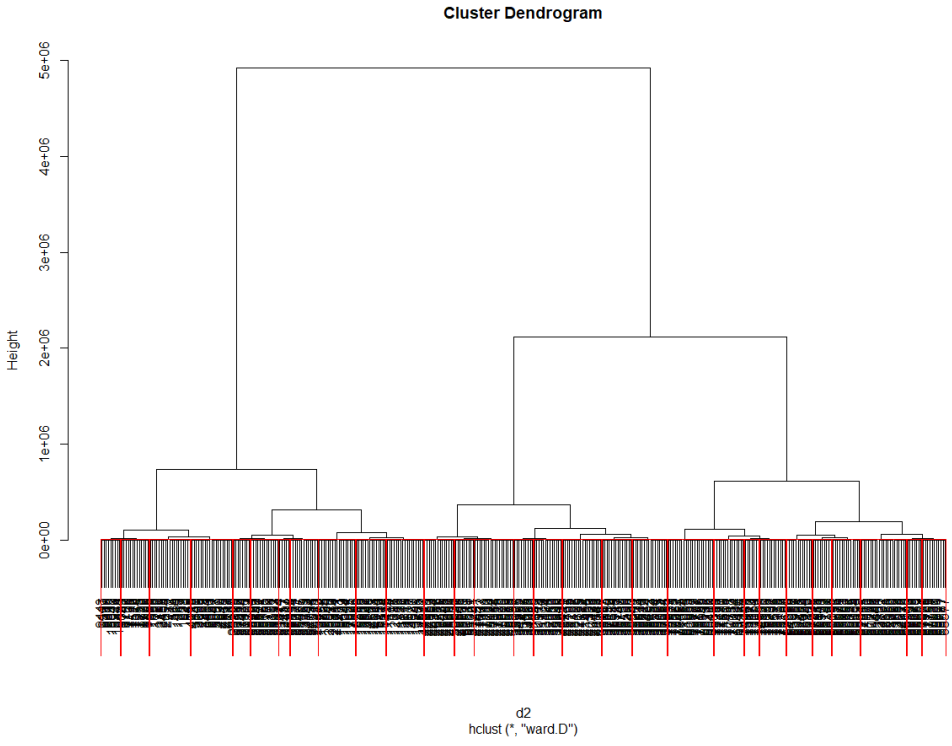
Anexo 64. Dendrograma intento 3 con 30 *clusters* identificados, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



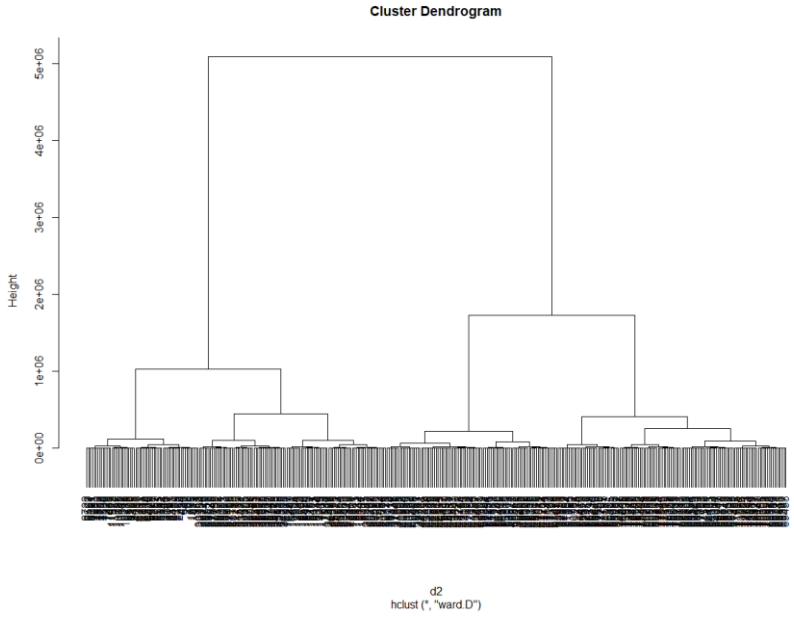
Anexo 65. Dendrograma intento 4 para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



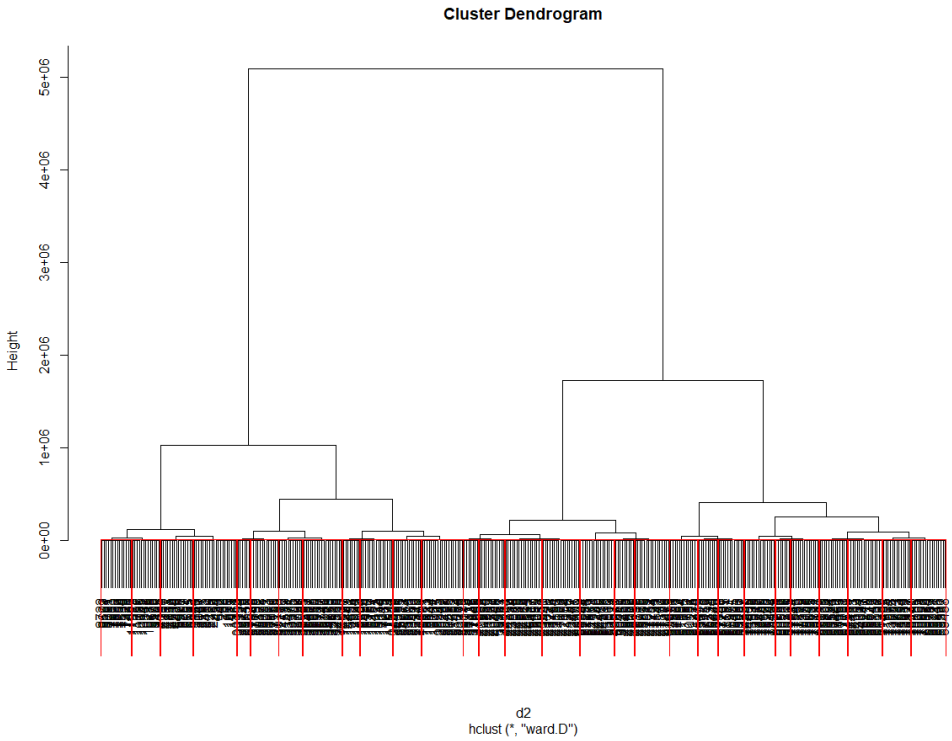
Anexo 66. Dendrograma intento 4 con 29 *clusters* identificados, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



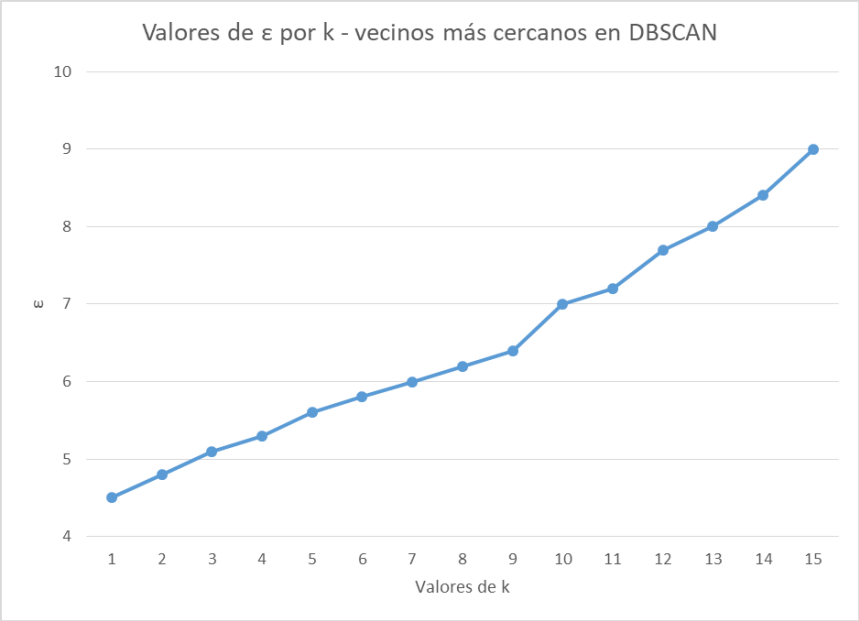
Anexo 67. Dendrograma intento 5 para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



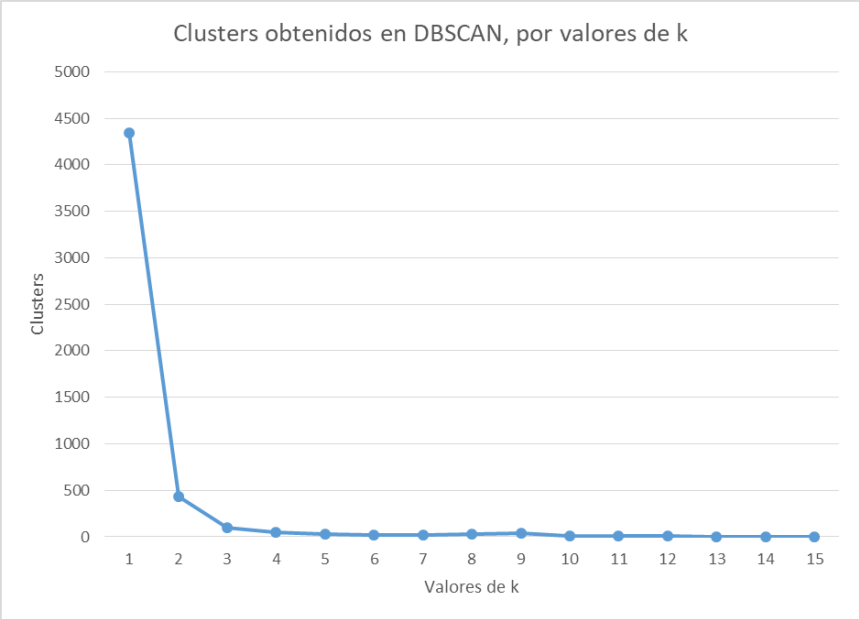
Anexo 68. Dendrograma intento 5 con 29 *clusters* identificados, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



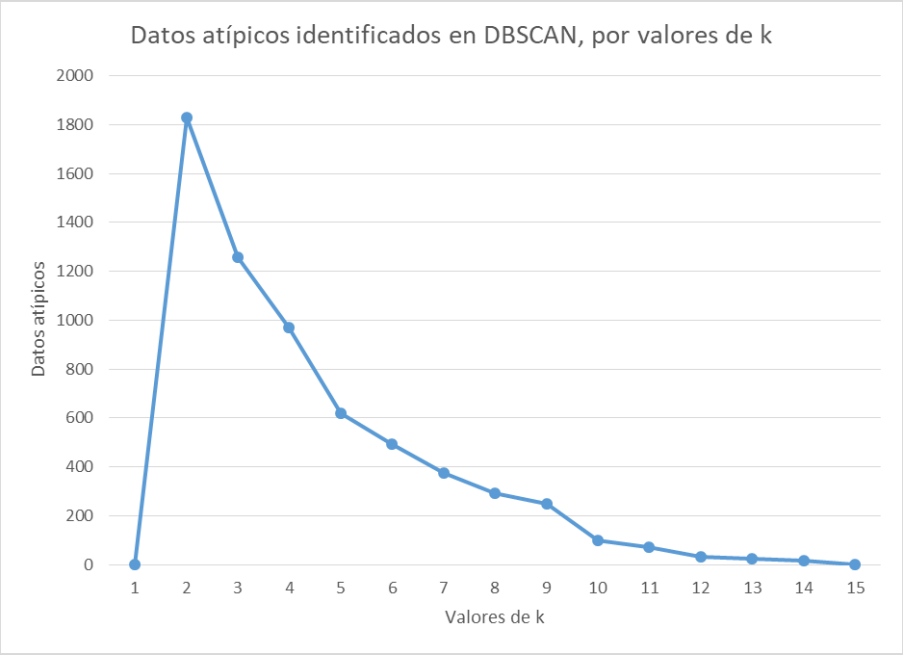
Anexo 69. Valores de épsilon (ε) por k – vecinos más cercanos para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



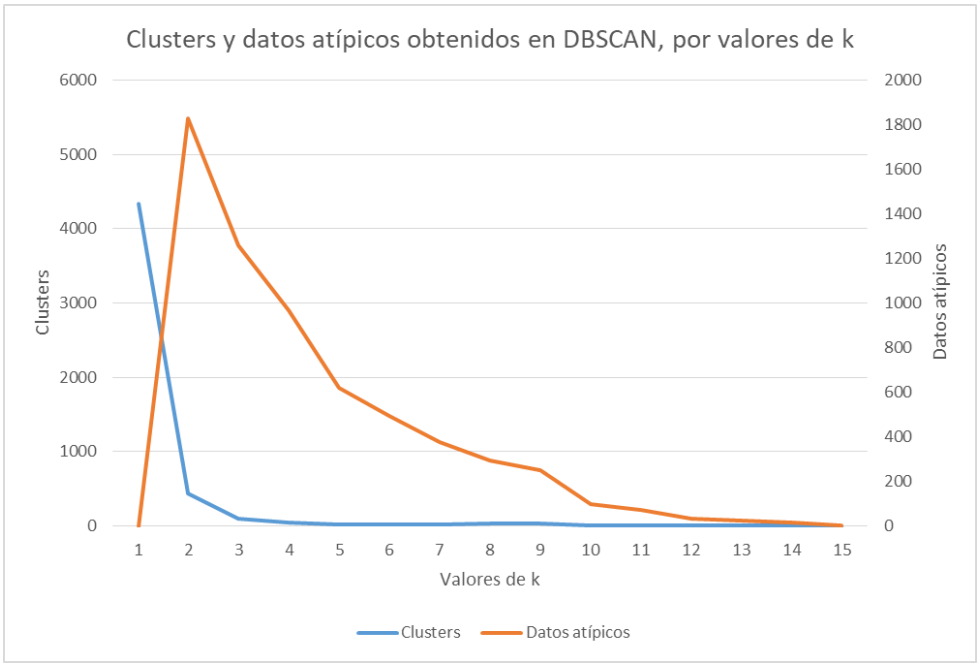
Anexo 70. Cantidad de *clusters* obtenidos en DBSCAN para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



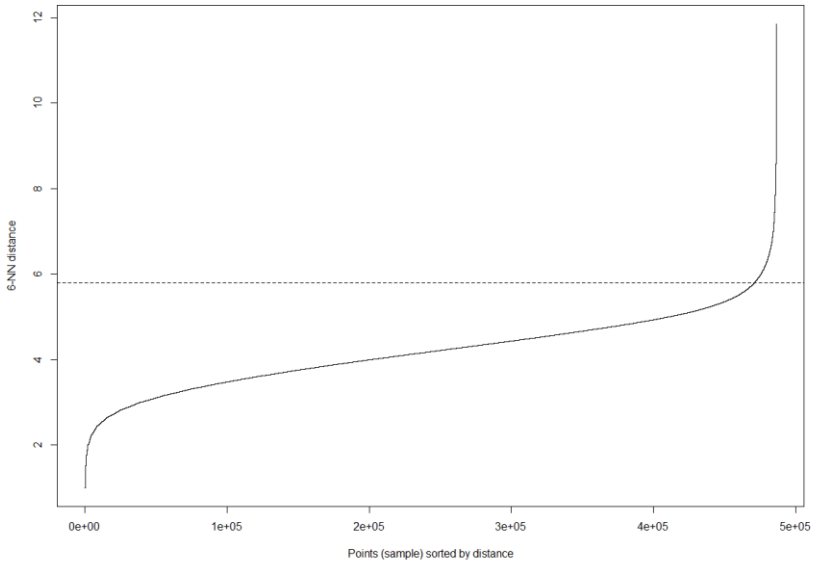
Anexo 71. Cantidad de datos atípicos en DBSCAN para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



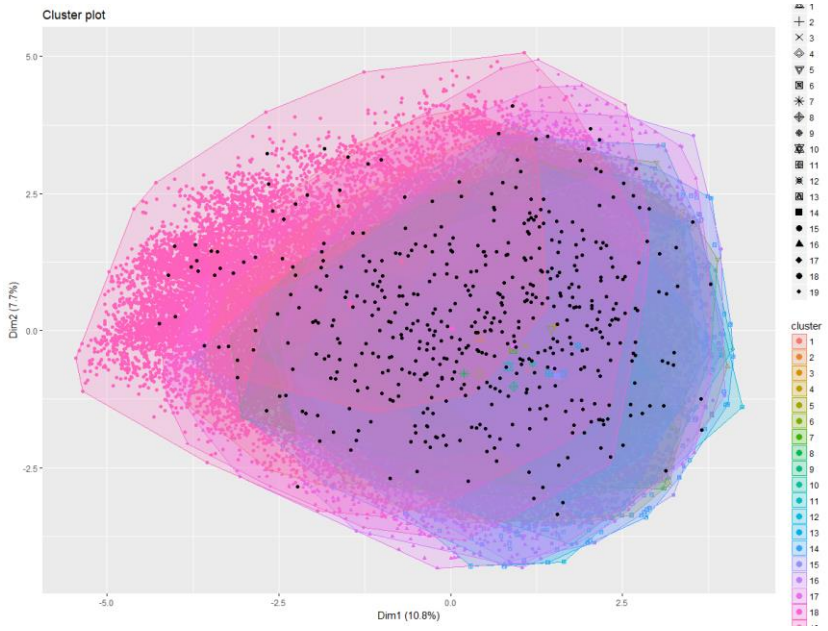
Anexo 72. Gráficos de *clusters* obtenidos y cantidad de datos atípicos identificados en DBSCAN, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 73. Gráfico de distancias de k – vecinos más cercanos, con *k* = 6, para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 74. *Clusters* obtenidos por DBSCAN para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 75. Código utilizado para ejecutar DBSCAN, para base de clientes con variables no binarias, estandarizadas con puntaje Z.

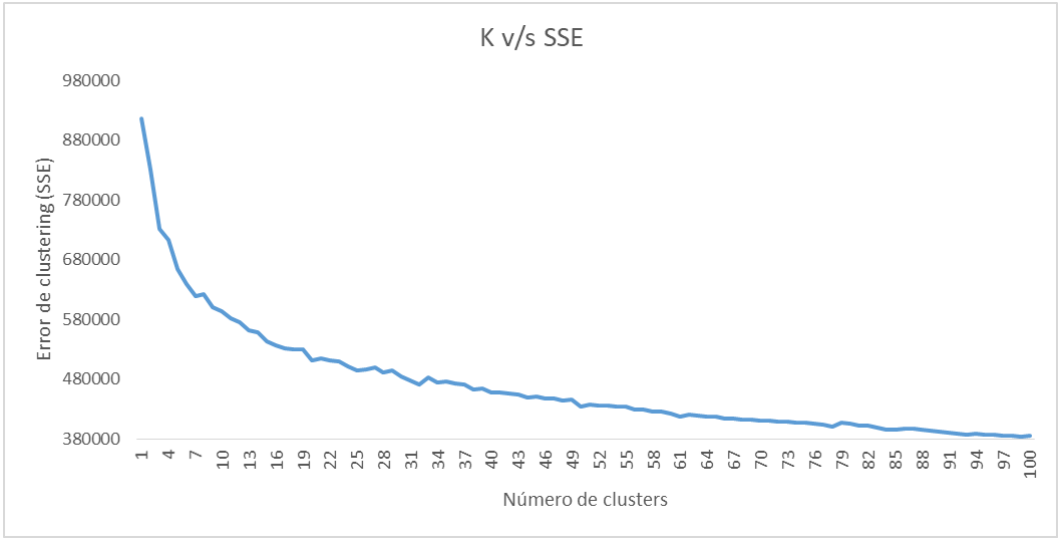
```
#Primero, se identificará valor de epsilon adecuado
library('dbscan')
library('factoextra')
dbscan::kNNdistplot(base2_2_v1_v2_df, k = 6)
abline(h = 5.8, lty = 2)
#Aplicando DBSCAN
res.db <- dbscan(base2_2_v1_v2_df, 5.8, 6)
res.db
#Graficando
fviz_cluster(res.db, base2_2_v1_v2_df, geom = "point")
#Descargando base
result_dbscan2 <- data.frame(cbind(base2_2_v1_v2_df, clusterNum = res.db$cluster))
write.csv(result_dbscan2, file = "resultados_DBSCAN_base2.csv")
```

Anexo 76. Código utilizado para ejecutar *K – Medoids* para base de clientes con variables no binarias, estandarizadas con puntaje Z.

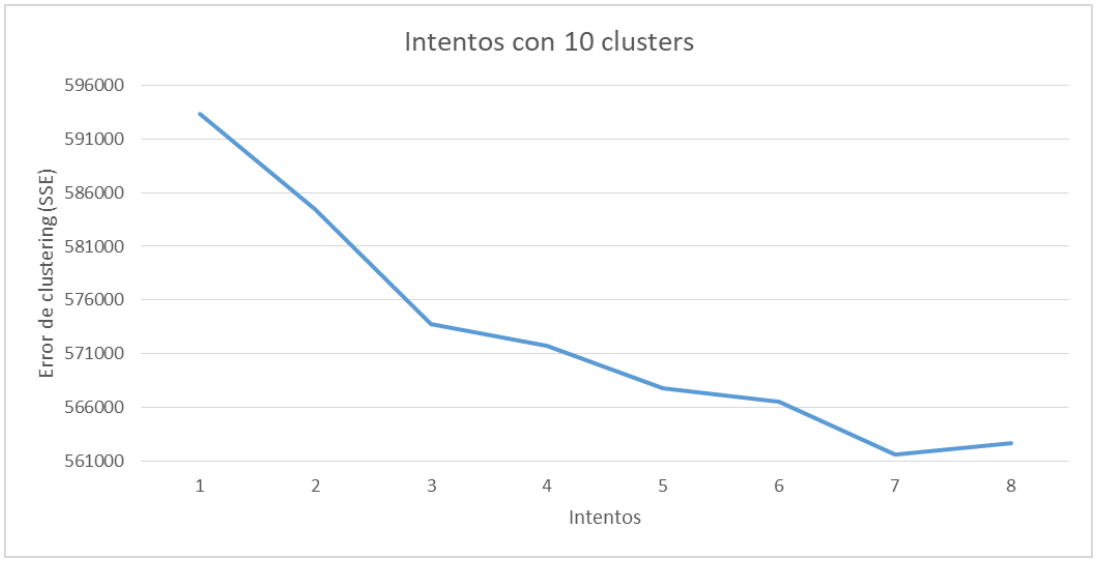
```
library(cluster)
result_kmedoids2 <- clara(base2_2_v1_v2_df[,c(-1)], 10, samples = 320)
result_kmedoids2$i.med
#Unir base con índices de clusters
result_db_kmedoids2 <- data.frame(cbind(base2_2_v1_v2_df, clusterNum = result_kmedoids2$clustering))
#Calcular SSE
clusters_kmedoids_seq <- seq(1, 10)
sse_kmedoids_final <- rep(0, 10)
for (x in clusters_kmedoids_seq) {
  datos_cluster <- result_db_kmedoids2[result_db_kmedoids2$clusterNum==x,]
  centroide <- result_db_kmedoids2[result_kmedoids2$i.med[x],seq(2,27)]
```

```
 valor_seq_Ward <- seq(2, 27)
 filas <- nrow(datos_cluster)
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
  sse_variables <- rep(0, 26)
  for (w in valor_seq_Ward) {
   sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
  }
  sse_valores[z] <- sum(sse_variables)
 }
 sse_kmedoids_final[x] <- sum(sse_valores)
}
sse <- sum(sse_kmedoids_final)
sse
#Descargar base
write.csv(result_db_kmedoids2, file = "resultados_KMedoids_base2_intento8_561641-6.csv")
```

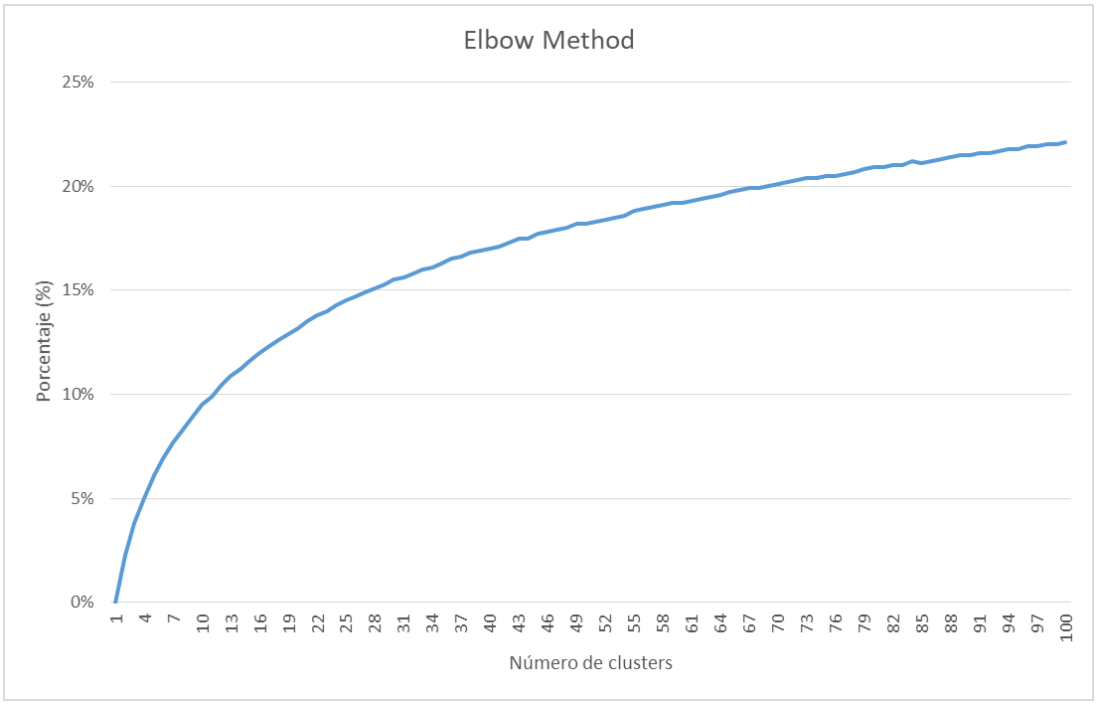
Anexo 77. Número de *clusters* v/s error de *K – Medoids* para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



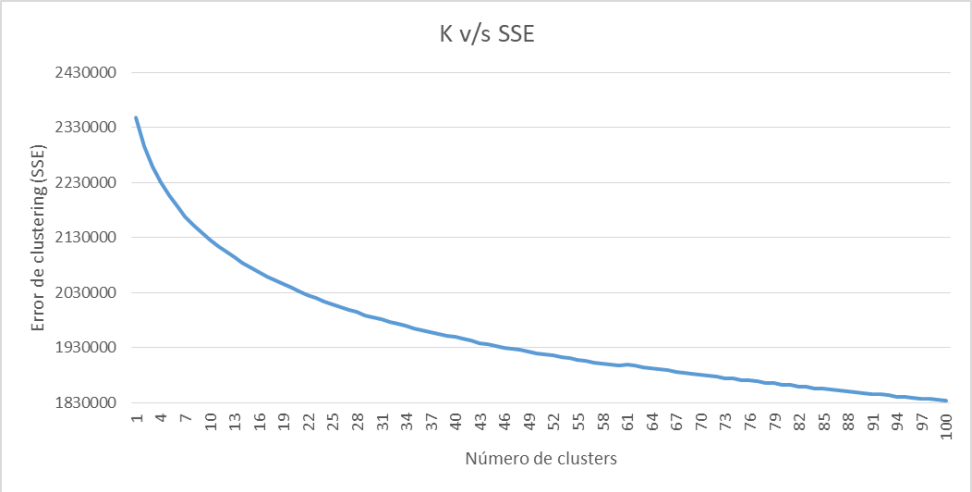
Anexo 78. Resultados de intentos de *K – Medoids* para base de clientes con variables no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 79. Número de *clusters* v/s porcentaje de explicabilidad para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



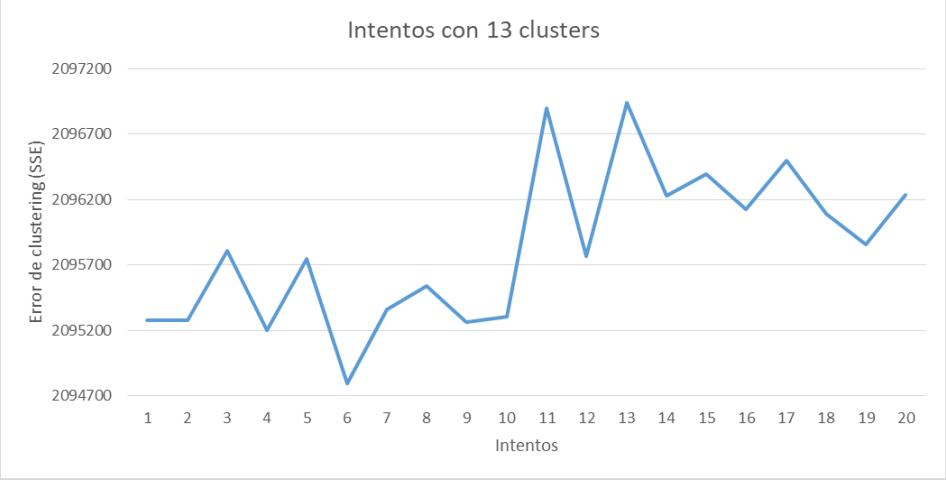
Anexo 80. Número de *clusters* v/s error de *clustering* para base con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 81. Código utilizado para ejecución de *K – Means*, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1.

```
cl_base2_3_v1 <- kmeans(base2_3_v1_df[,c(-1)], 13, iter.max = 850, algorithm = "Lloyd")
#Aplica K - Means a la base especificada
cl_base2_3_v1
cl_base2_3_v1$tot.withinss
result_k3 <- data.frame(cbind(base2_3_v1_df, clusterNum = cl_base2_3_v1$cluster))
write.csv(result_k3, file = "resultados_KMeans_base3_intento20.csv")
```

Anexo 82. Distintos intentos de *K – Means* con 13 *clusters* para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 83. Código utilizado para la ejecución del método de Ward, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1.

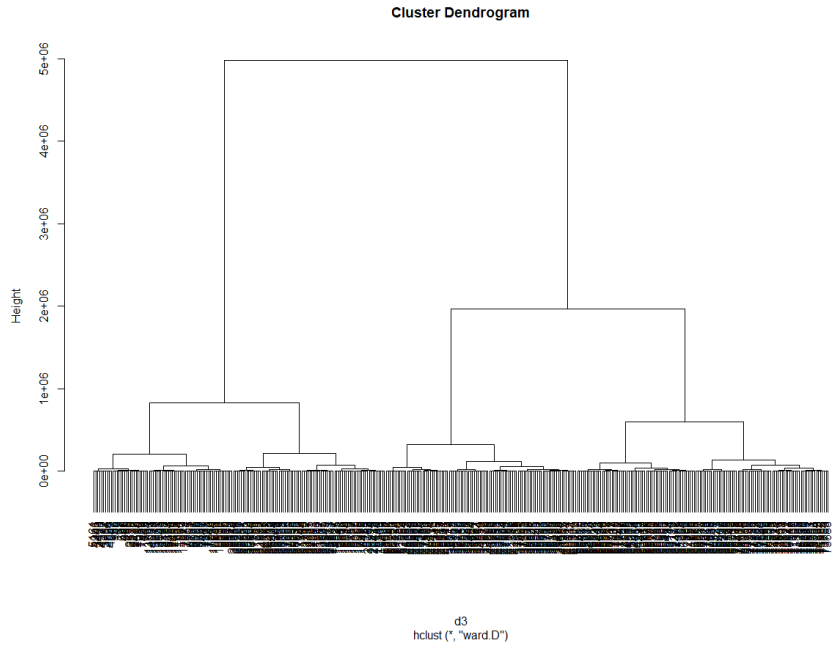
```
#Se obtiene una muestra de la base total
muestreo3 <- sample(1:nrow(base2_3_v1_df), 385)
base_muestra3 <- base2_3_v1_df[c(muestreo3),]
#Aplicando método de Ward
d3 <- dist(base_muestra3, method = "euclidean")
ward_base3 <- hclust(d3, method = "ward.D")
#Graficando dendrograma
plot(ward_base3)
groups <- cutree(ward_base3, k = 29) #Aquí, se especifican cuántos clusters existen en el dendrograma.
rect.hclust(ward_base3, k = 29, border = "red")
#Identificando los datos de cada cluster en la muestra
result_ward3_muestra <- data.frame(cbind(base_muestra3, clusterNum = groups))
#Calcular SSE
clusters_seq3 <- seq(1, 29) #Cantidad de clusters
sse_final3 <- rep(0, 29) #Cantidad de clusters
for (x in clusters_seq3) {
 datos_cluster <- result_ward3_muestra[result_ward3_muestra$clusterNum==x,]
 centroide <- rep(0, 29)
 valor_seq_Ward <- seq(2, 30)
 for (y in valor_seq_Ward) {
  centroide[y - 1] <- mean(datos_cluster[,y])
 }
 filas <- nrow(datos_cluster)
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
```

```
   sse_variables <- rep(0, 29)
   for (w in valor_seq_Ward) {
     sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
   }
   sse_valores[z] <- sum(sse_variables)
  }
  sse_final3[x] <- sum(sse_valores)
}
sse3 <- sum(sse_final3)
sse3
#Asignar clusters a cada dato de la base
centroides_ward3 <- matrix(nrow = 29, ncol = 29, rep(0, 841), byrow = TRUE) #El número
de filas son los clusters y los ceros son filas por columnas.
nfilas_centroides_ward3 <- nrow(centroides_ward3)
seq_filas3 <- seq(1, nfilas_centroides_ward3)
for (x in seq_filas3) {
  datos_cluster <- result_ward3_muestra[result_ward3_muestra$clusterNum==x,]
  valor_seq_Ward <- seq(2, 30)
  for (y in valor_seq_Ward) {
    centroides_ward3[x,y - 1] <- mean(datos_cluster[,y])
  }
}
seq_base3 <- seq(1, nrow(base2_3_v1_df))
seq_muestra3 <- seq(1, nrow(base_muestra3))
cluster_base3 <- rep(0, 81107)
for (x in seq_base3) {
  for (y in seq_muestra3) {
    num_muestra <- base_muestra3[y,1]
    if (num_muestra == x) {
```

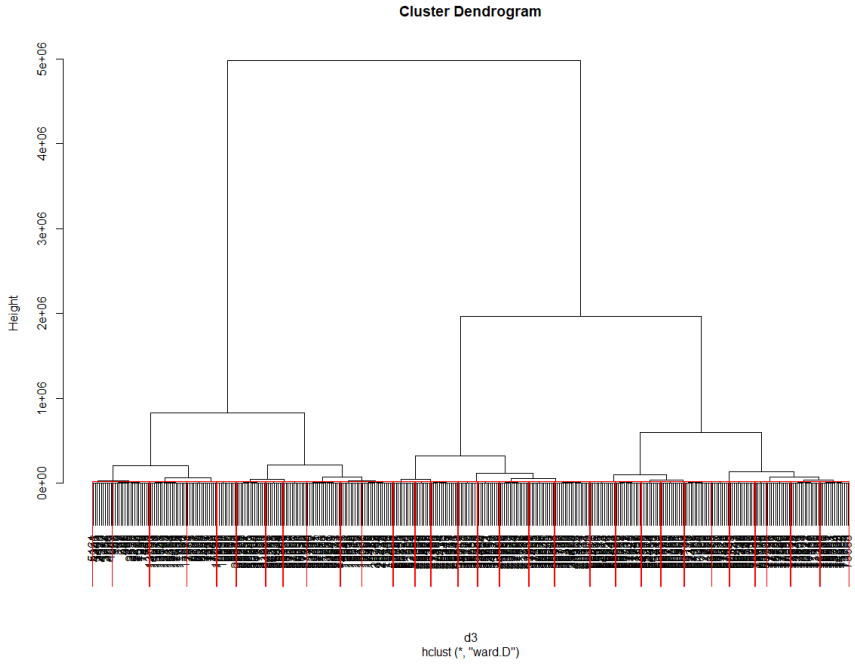
```
    cluster_base3[x] <- result_ward3_muestra$clusterNum[y]
    break
   }
  }
  dist_clusters <- rep(0, 29) #Colocar número de clusters
  seq_clust <- seq(1, 29) #Colocar número de clusters
  for (z in seq_clust) {
   distancias <- rep(0, 29)
   seq_dist <- seq(1, 29)
   for (xx in seq_dist) {
    distancias[xx] <- (base2_3_v1_df[x,xx + 1] - centroides_ward3[z,xx])^2
   }
   sum_distancias <- sum(distancias)
   dist_clusters[z] <- sqrt(sum_distancias)
  }
  min_dist <- min(dist_clusters)
  for (zz in seq_clust) {
   valor_dist <- dist_clusters[zz]
   if (valor_dist == min_dist) {
    cluster_base3[x] <- zz
    break
   }
  }
}
#Unir base total con etiquetas de cluster
result_ward3_total <- data.frame(cbind(base2_3_v1_df, clusterNum = cluster_base3))
#Calcular SSE de base total
sse_final_total3 <- rep(0, 29) #Colocar cantidad de clusters
for (x in clusters_seq3) {
```

```
 datos_cluster3 <- result_ward3_total[result_ward3_total$clusterNum==2,]
 centroide3 <- rep(0, 29)
 valor_seq_Ward3 <- seq(2, 30)
 for (y in valor_seq_Ward3) {
  centroide3[y - 1] <- mean(datos_cluster3[,y])
 }
 filas3 <- nrow(datos_cluster3)
 valor_filas3 <- seq(1, filas3)
 sse_valores3 <- rep(0, filas3)
 for (z in valor_filas3) {
  sse_variables3 <- rep(0, 29)
  for (w in valor_seq_Ward3) {
   sse_variables3 [w - 1] <- (datos_cluster3[z,w] - centroide3[w - 1])^2
  }
  sse_valores3[z] <- sum(sse_variables3)
 }
 sse_final_total3[x] <- sum(sse_valores3)
}
sse_total3 <- sum(sse_final_total3)
sse_total3
#Exportar archivo
write.csv(result_ward3_total, file = "resultados_Ward_base3_intento5_1920896.csv")
```

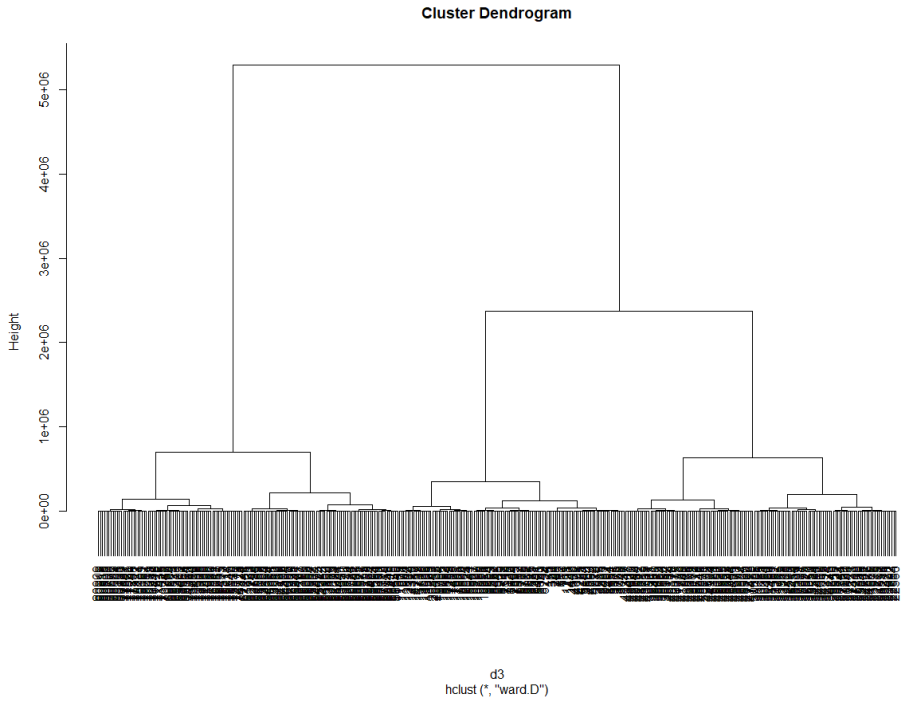
Anexo 84. Dendrograma intento 1 para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



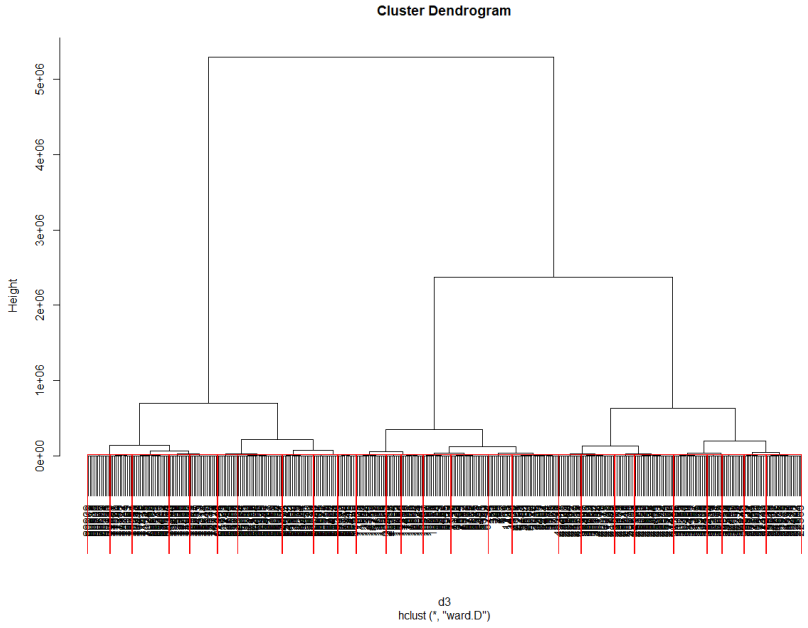
Anexo 85. Dendrograma intento 1 con 30 *clusters* identificados, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



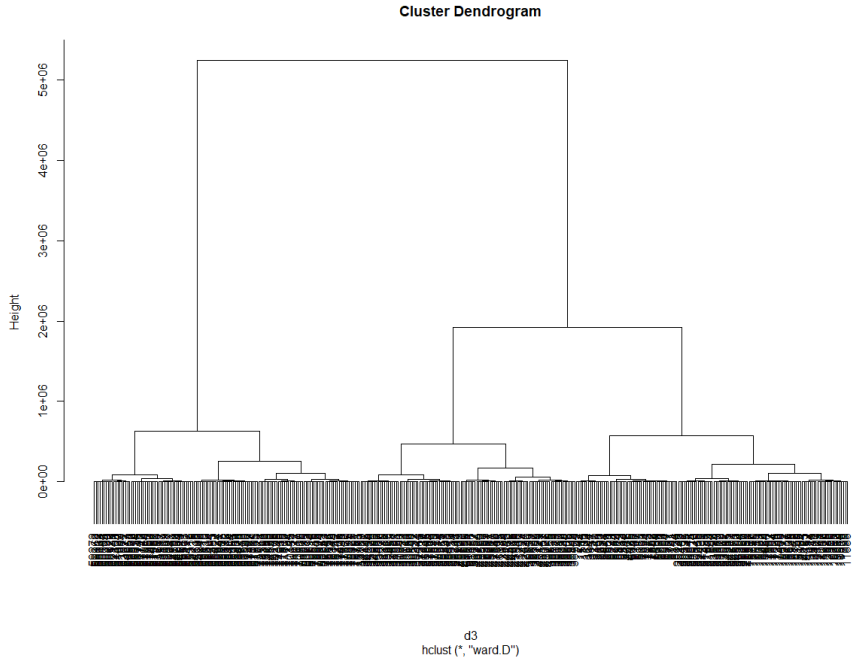
Anexo 86. Dendrograma intento 2 para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



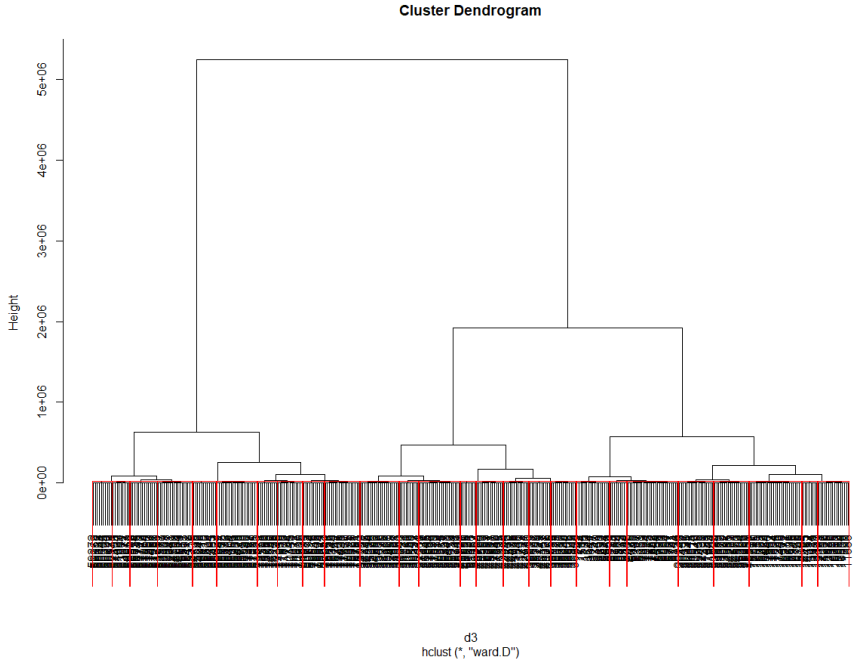
Anexo 87. Dendrograma intento 2 con 26 *clusters* identificados, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



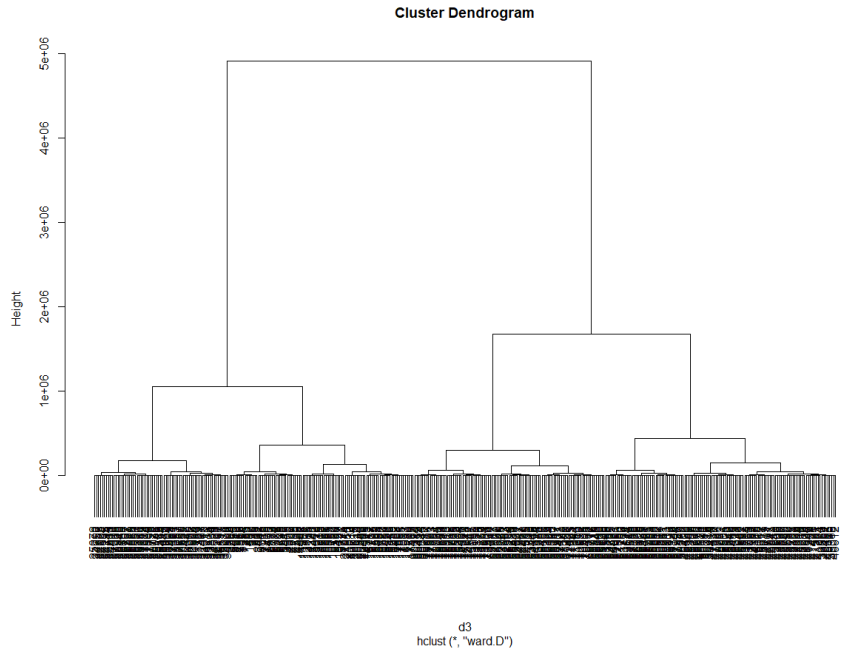
Anexo 88. Dendrograma intento 3 para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



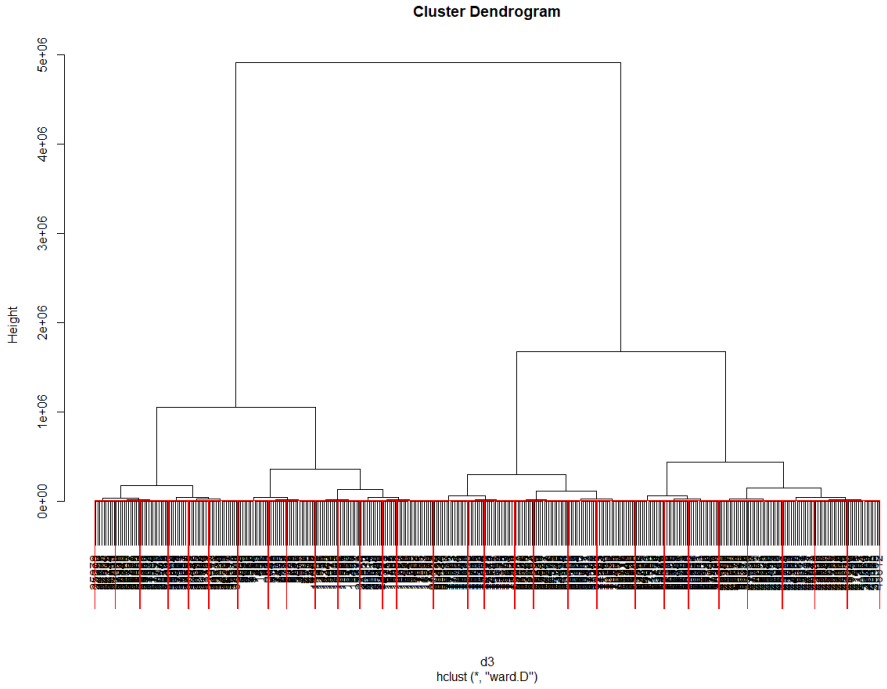
Anexo 89. Dendrograma intento 3 con 26 *clusters* identificados, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



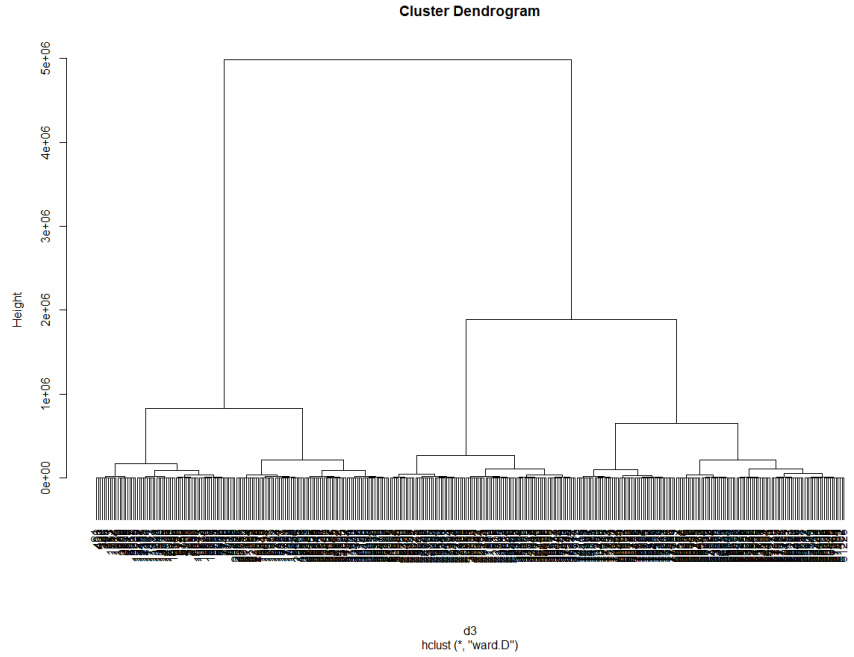
Anexo 90. Dendrograma intento 4 para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



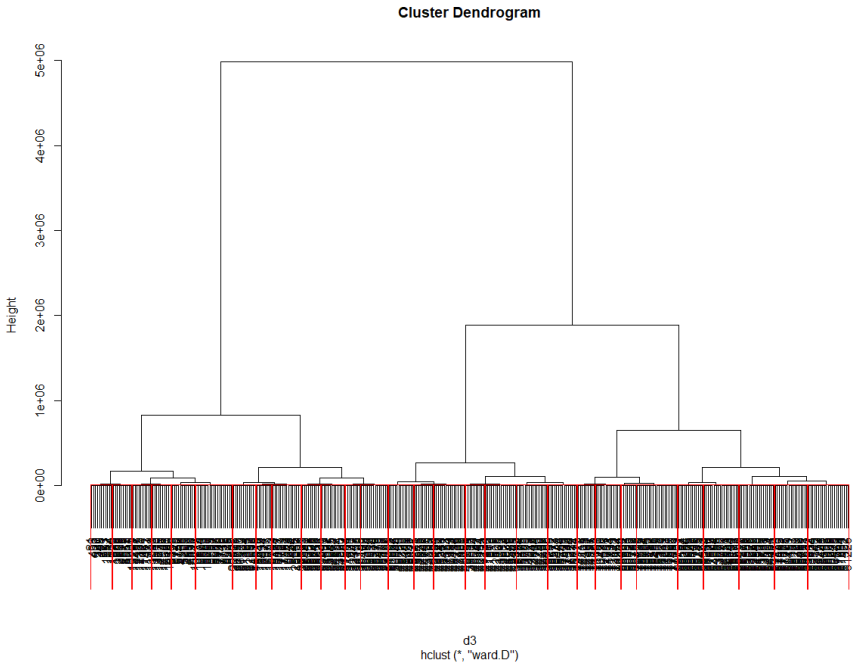
Anexo 91. Dendrograma intento 4 con 29 *clusters* identificados, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



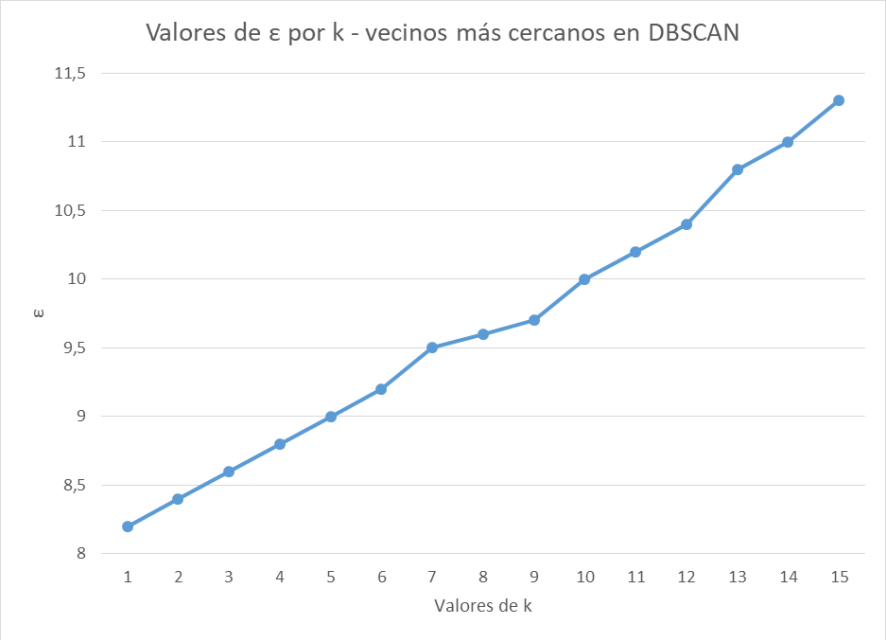
Anexo 92. Dendrograma intento 5 para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



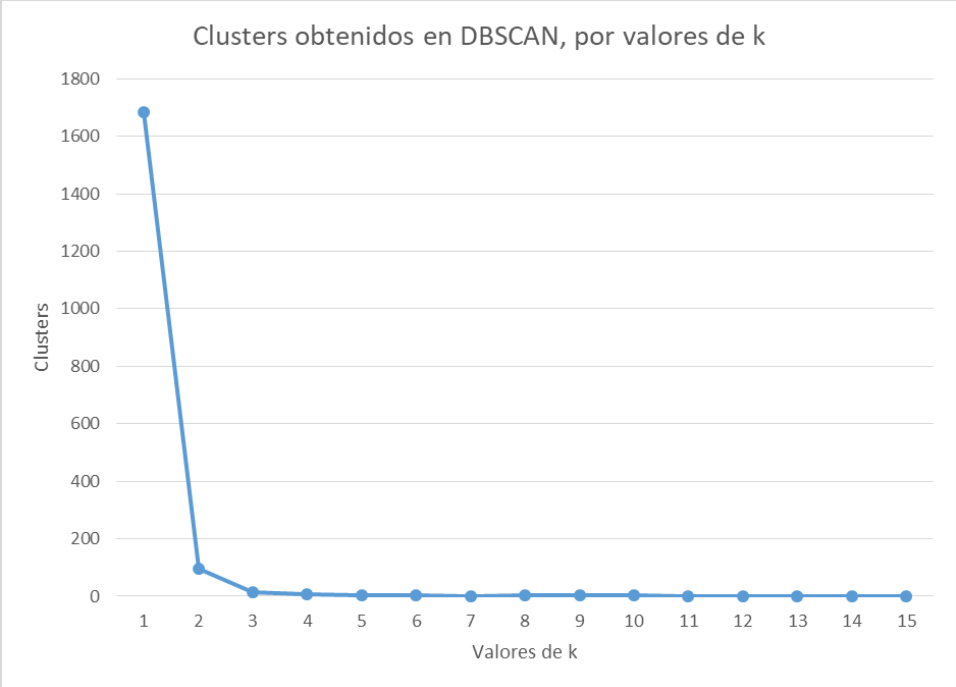
Anexo 93. Dendrograma intento 5 con 29 *clusters* identificados, para base de clientes con variables binarias y no binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



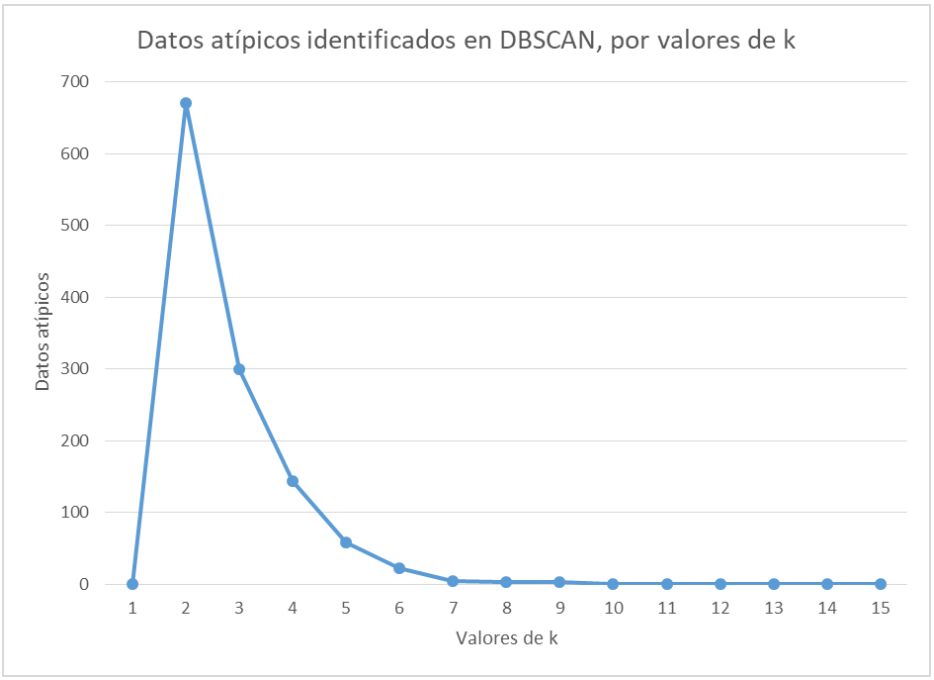
Anexo 94. Valores de épsilon (ε) por k – vecinos más cercanos, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



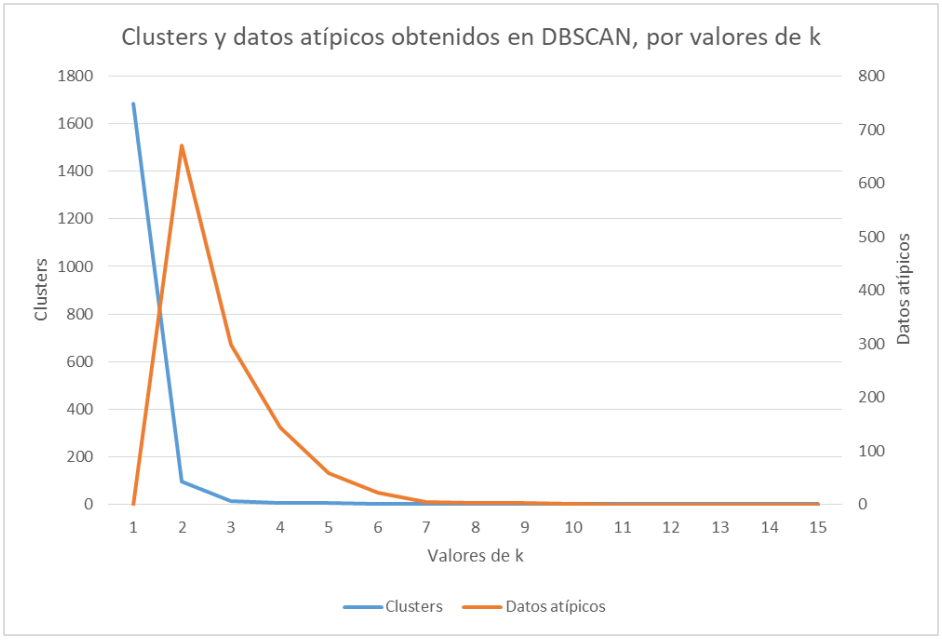
Anexo 95. Cantidad de *clusters* obtenidos en DBSCAN, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



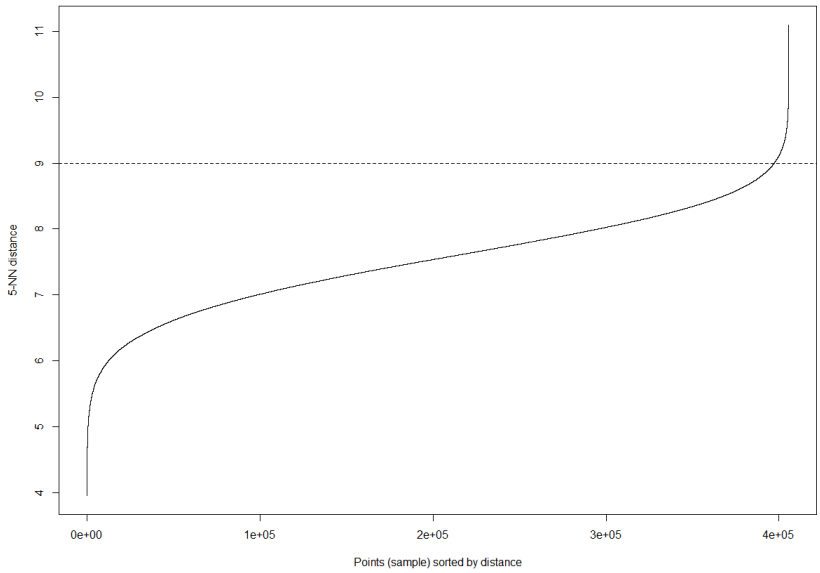
Anexo 96. Cantidad de datos atípicos en DBSCAN, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



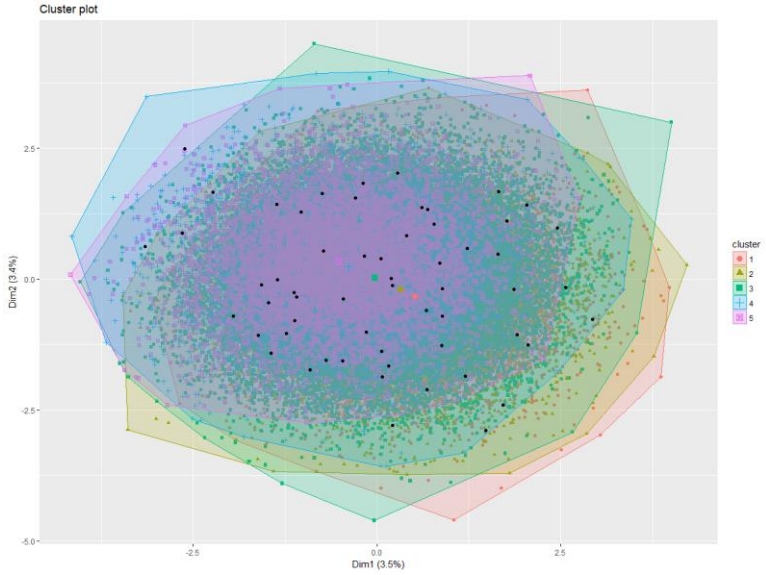
Anexo 97. Cantidad de *clusters* y datos atípicos obtenidos en DBSCAN, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 98. Gráfico de distancias de k – vecinos más cercanos, con *k* = 5, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 99. *Clusters* obtenidos por DBSCAN para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 100. Código utilizado para ejecución de DBSCAN, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

#Primero, se identificará valor de epsilon adecuado

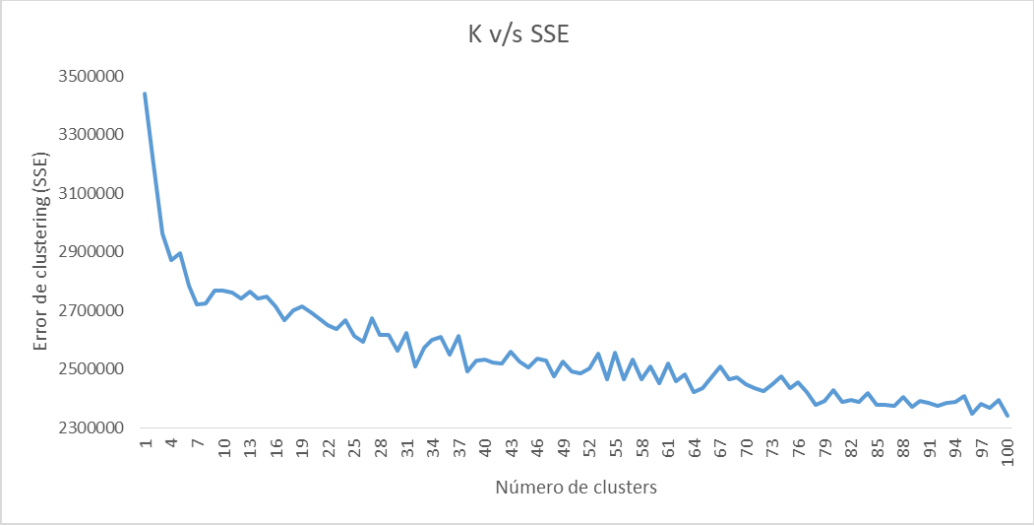
```
library('dbscan')
library('ggplot2')
library('factoextra')
dbscan::kNNdistplot(base2_3_v1_df, k = 5)
abline(h = 9.0, lty = 2)
#Aplicando DBSCAN
res.db <- dbscan(base2_3_v1_df, 9.0, 5)
res.db
#Graficando
fviz_cluster(res.db, base2_3_v1_df, geom = "point")
#Descargando base
result_dbscan3 <- data.frame(cbind(base2_3_v1_df, clusterNum = res.db$cluster))
write.csv(result_dbscan3, file = "resultados_DBSCAN_base3.csv")
```

Anexo 101. Código utilizado para ejecución de *K – Medoids*, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).

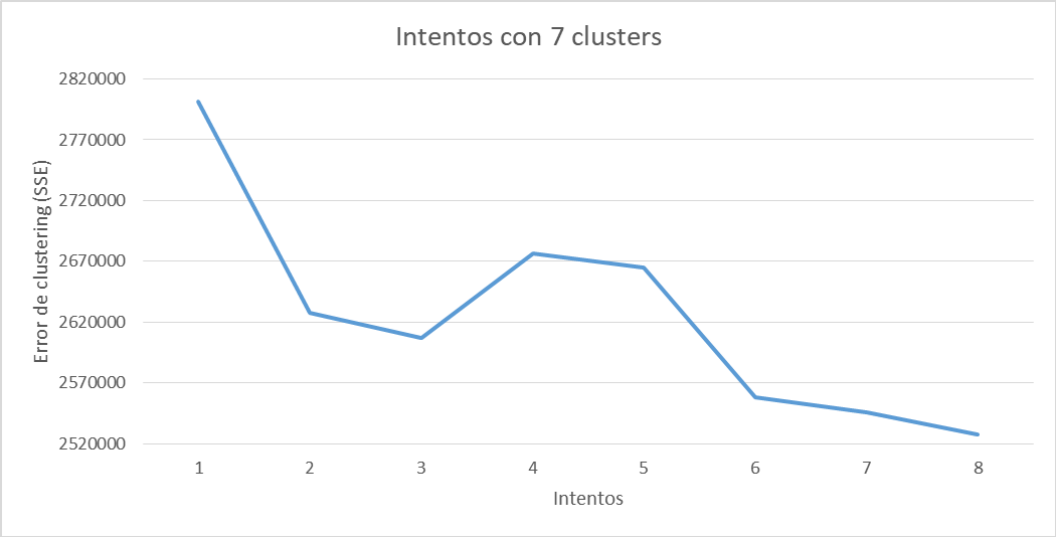
```
library(cluster)
result_kmedoids3 <- clara(base2_3_v1_df[,c(-1)], 7, samples = 620) #Usar samples para hacer las pruebas.
result_kmedoids3$i.med
#Unir base con índices de clusters
result_db_kmedoids3 <- data.frame(cbind(base2_3_v1_df, clusterNum = result_kmedoids3$clustering))
#Calcular SSE
clusters_kmedoids_seq <- seq(1, 7)
sse_kmedoids_final <- rep(0, 7)
for (x in clusters_kmedoids_seq) {
  datos_cluster <- result_db_kmedoids3[result_db_kmedoids3$clusterNum==x,]
  centroide <- result_db_kmedoids3[result_kmedoids3$i.med[x],seq(2,30)]
  valor_seq_Ward <- seq(2, 30)
```

```
 filas <- nrow(datos_cluster)
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
  sse_variables <- rep(0, 29)
  for (w in valor_seq_Ward) {
   sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
  }
  sse_valores[z] <- sum(sse_variables)
 }
 sse_kmedoids_final[x] <- sum(sse_valores)
}
sse <- sum(sse_kmedoids_final)
sse
#Descargar base
write.csv(result_db_kmedoids3, file =
"resultados_KMedoids_base3_intento8_2527828-0.csv")
```

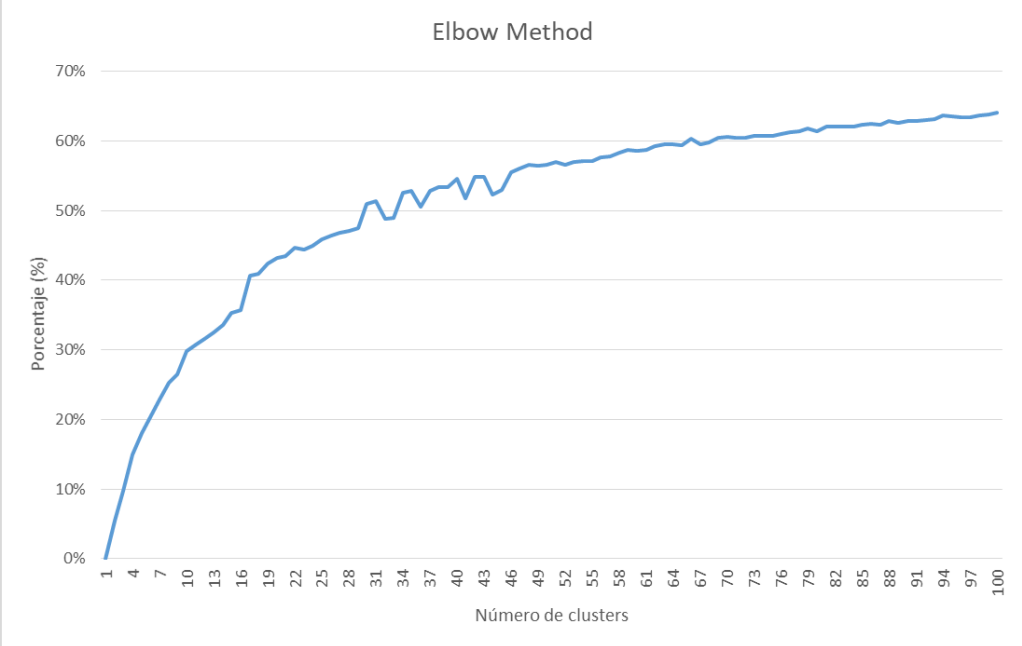
Anexo 102. Número de *clusters* v/s error de *K – Medoids*, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



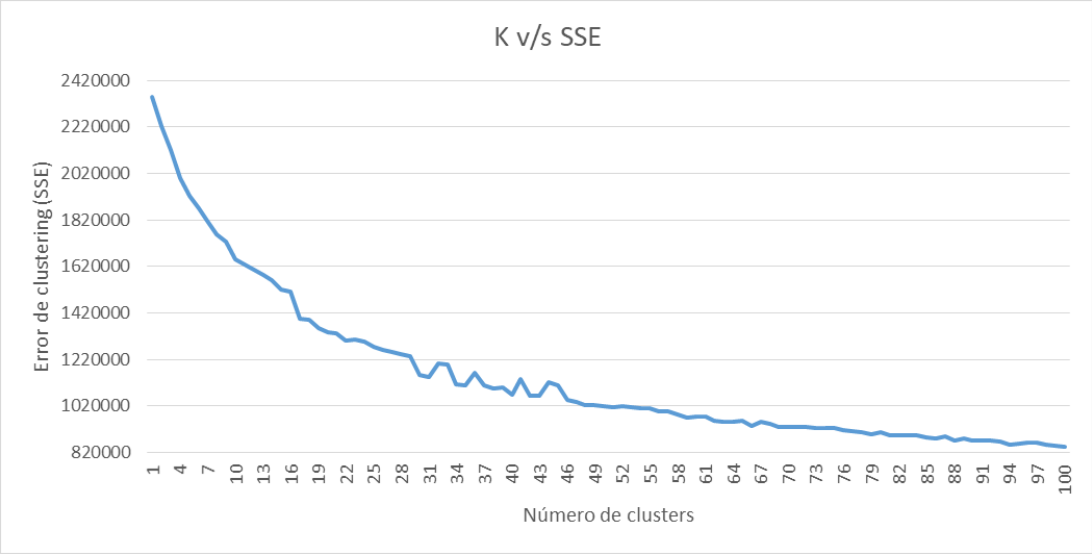
Anexo 103. Resultados de intentos de *K – Medoids*, para base de clientes con variables no binarias y binarias, normalizadas entre 0 y 1 (Fuente: Elaboración propia).



Anexo 104. Número de *clusters* v/s tasa de explicabilidad, para base de clientes con variables no binarias y binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



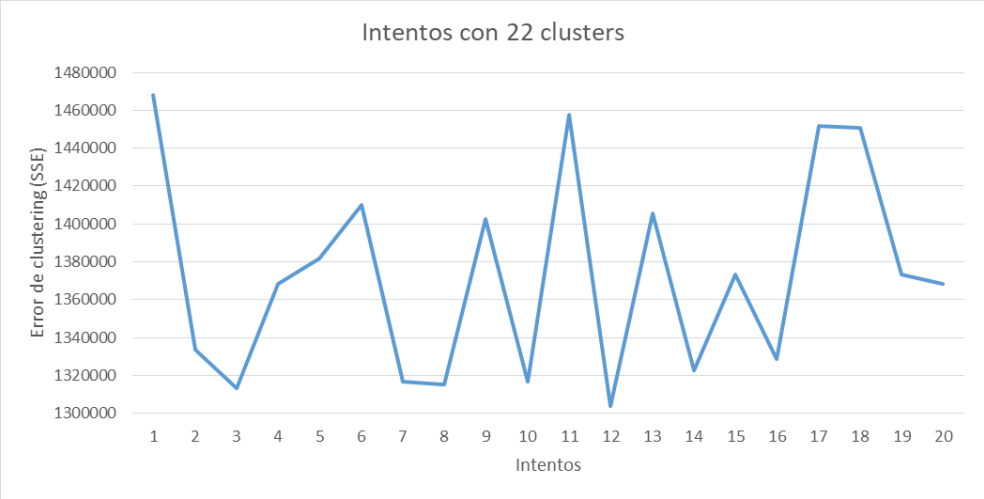
Anexo 105. Número de *clusters* v/s error del *clustering*, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 106. Código utilizado para ejecutar *K – Means*, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z

```
cl_base2_4_v1 <- kmeans(base2_4_v1_v2_df[,c(-1)], 22, iter.max = 850, algorithm = "Lloyd") #Aplica K - Means a la base especificada
cl_base2_4_v1
cl_base2_4_v1$tot.withinss
result_k4 <- data.frame(cbind(base2_4_v1_v2_df, clusterNum = cl_base2_4_v1$cluster))
write.csv(result_k4, file = "resultados_KMeans_base4_intento20.csv")
```

Anexo 107. Distintos intentos de *K – Means* con 22 *clusters* para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 108. Código utilizado para la ejecución del método de Ward, para base de clientes con variables no binarias y binarias, estandarizadas con puntaje Z.

```
#Se obtiene una muestra de la base total
muestreo4 <- sample(1:nrow(base2_4_v1_v2_df), 385)
base_muestra4 <- base2_4_v1_v2_df[c(muestreo4),]
#Aplicando método de Ward
d4 <- dist(base_muestra4, method = "euclidean")
ward_base4 <- hclust(d4, method = "ward.D")
#Graficando dendrograma
plot(ward_base4)
groups <- cutree(ward_base4, k = 29) #Aquí, se especifican cuántos clusters existen en el dendrograma.
rect.hclust(ward_base4, k = 29, border = "red")
#Identificando los datos de cada cluster en la muestra
result_ward4_muestra <- data.frame(cbind(base_muestra4, clusterNum = groups))
#Calcular SSE
clusters_seq4 <- seq(1, 29) #Cantidad de clusters
sse_final4 <- rep(0, 29) #Cantidad de clusters
```

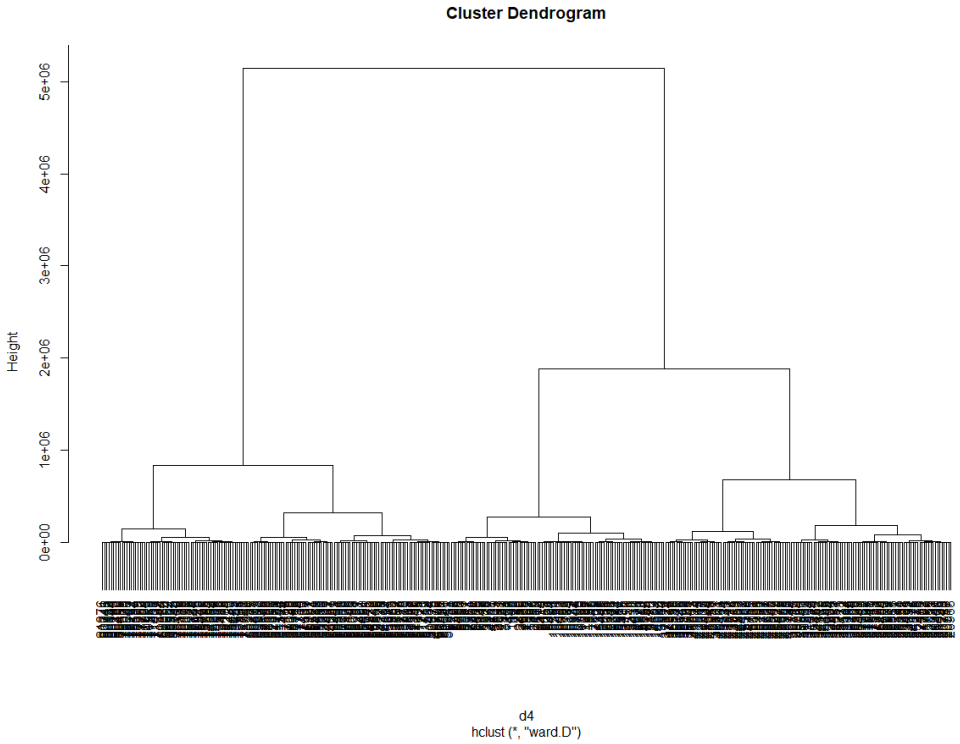
```
for (x in clusters_seq4) {
  datos_cluster <- result_ward4_muestra[result_ward4_muestra$clusterNum==x,]
  centroide <- rep(0, 29)
  valor_seq_Ward <- seq(2, 30)
  for (y in valor_seq_Ward) {
    centroide[y - 1] <- mean(datos_cluster[,y])
  }
  filas <- nrow(datos_cluster)
  valor_filas <- seq(1, filas)
  sse_valores <- rep(0, filas)
  for (z in valor_filas) {
    sse_variables <- rep(0, 29)
    for (w in valor_seq_Ward) {
      sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
    }
    sse_valores[z] <- sum(sse_variables)
  }
  sse_final4[x] <- sum(sse_valores)
}
sse4 <- sum(sse_final4)
sse4
#Asignar clusters a cada dato de la base
centroides_ward4 <- matrix(nrow = 29, ncol = 29, rep(0, 841), byrow = TRUE) #El número
de filas son los clusters y los ceros son filas por columnas.
nfilas_centroides_ward4 <- nrow(centroides_ward4)
seq_filas4 <- seq(1, nfilas_centroides_ward4)
for (x in seq_filas4) {
  datos_cluster <- result_ward4_muestra[result_ward4_muestra$clusterNum==x,]
  valor_seq_Ward <- seq(2, 30)
```

```
  for (y in valor_seq_Ward) {
    centroides_ward4[x,y - 1] <- mean(datos_cluster[,y])
  }
}
seq_base4 <- seq(1, nrow(base2_4_v1_v2_df))
seq_muestra4 <- seq(1, nrow(base_muestra4))
cluster_base4 <- rep(0, 81107)
for (x in seq_base4) {
  for (y in seq_muestra4) {
    num_muestra <- base_muestra4[y,1]
    if (num_muestra == x) {
      cluster_base4[x] <- result_ward4_muestra$clusterNum[y]
      break
    }
  }
  dist_clusters <- rep(0, 29) #Colocar número de clusters
  seq_clust <- seq(1, 29) #Colocar número de clusters
  for (z in seq_clust) {
    distancias <- rep(0, 29)
    seq_dist <- seq(1, 29)
    for (xx in seq_dist) {
      distancias[xx] <- (base2_4_v1_v2_df[x,xx + 1] - centroides_ward4[z,xx])^2
    }
    sum_distancias <- sum(distancias)
    dist_clusters[z] <- sqrt(sum_distancias)
  }
  min_dist <- min(dist_clusters)
  for (zz in seq_clust) {
    valor_dist <- dist_clusters[zz]
```

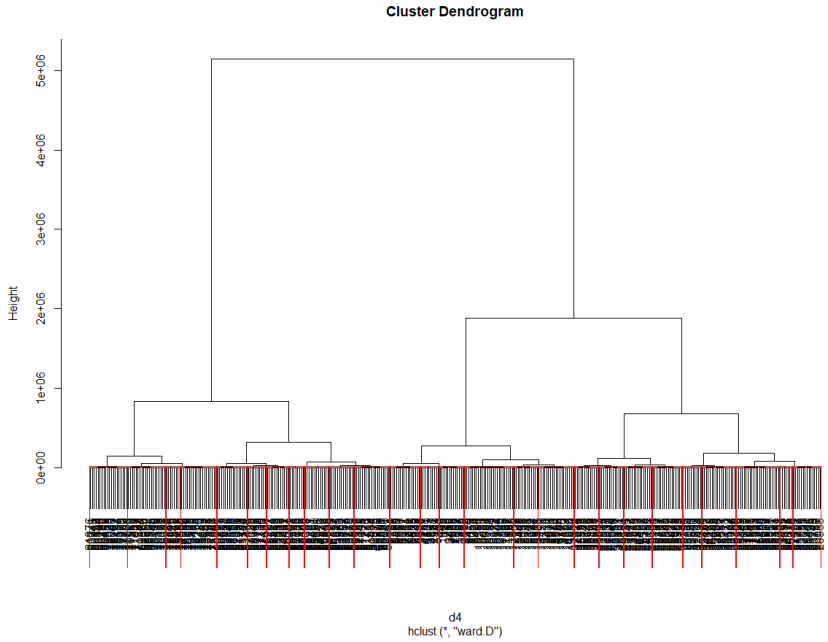
```
   if (valor_dist == min_dist) {
     cluster_base4[x] <- zz
     break
   }
  }
}
#Unir base total con etiquetas de cluster
result_ward4_total <- data.frame(cbind(base2_4_v1_v2_df, clusterNum =
cluster_base4))
#Calcular SSE de base total
sse_final_total4 <- rep(0, 29) #Colocar cantidad de clusters
for (x in clusters_seq4) {
  datos_cluster4 <- result_ward4_total[result_ward4_total$clusterNum==x,]
  centroide4 <- rep(0, 29)
  valor_seq_Ward4 <- seq(2, 30)
  for (y in valor_seq_Ward4) {
    centroide4[y - 1] <- mean(datos_cluster4[,y])
  }
  filas4 <- nrow(datos_cluster4)
  valor_filas4 <- seq(1, filas4)
  sse_valores4 <- rep(0, filas4)
  for (z in valor_filas4) {
    sse_variables4 <- rep(0, 29)
    for (w in valor_seq_Ward4) {
      sse_variables4 [w - 1] <- (datos_cluster4[z,w] - centroide4[w - 1])^2
    }
    sse_valores4[z] <- sum(sse_variables4)
  }
  sse_final_total4[x] <- sum(sse_valores4)
```

```
}
sse_total4 <- sum(sse_final_total4)
sse_total4
#Exportar archivo
write.csv(result_ward4_total, file = "resultados_Ward_base4_intento5_1590799.csv")
```

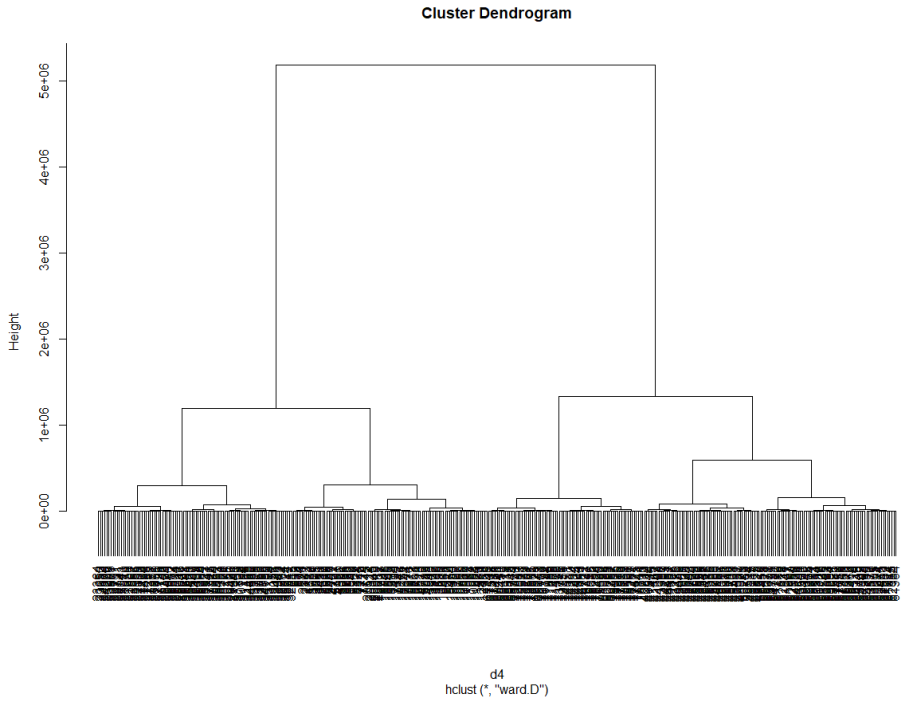
Anexo 109. Dendrograma intento 1 para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



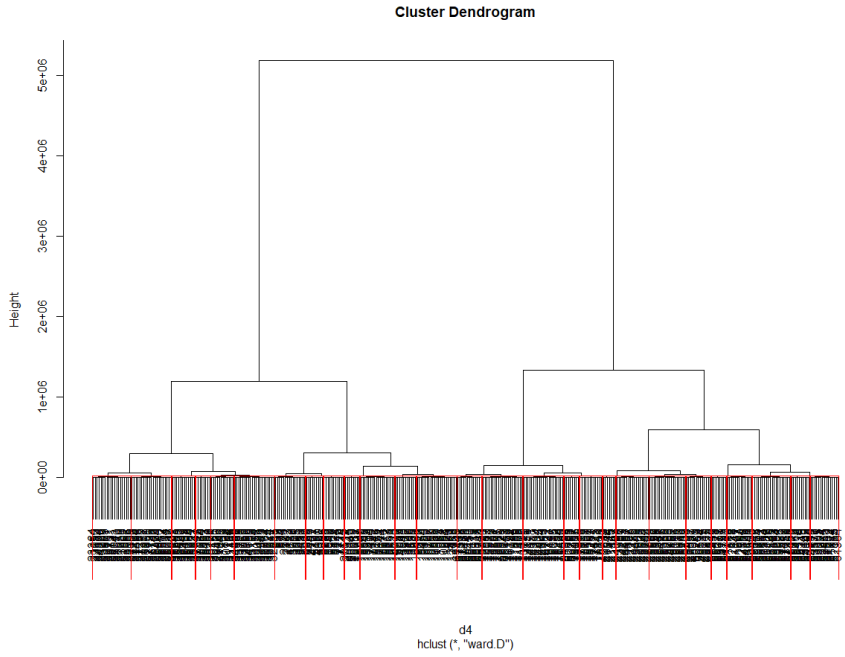
Anexo 110. Dendrograma intento 1 con 26 *clusters* identificados, para base de clientes con variables no binarias y binarias, estandarizados con puntaje Z (Fuente: Elaboración propia).



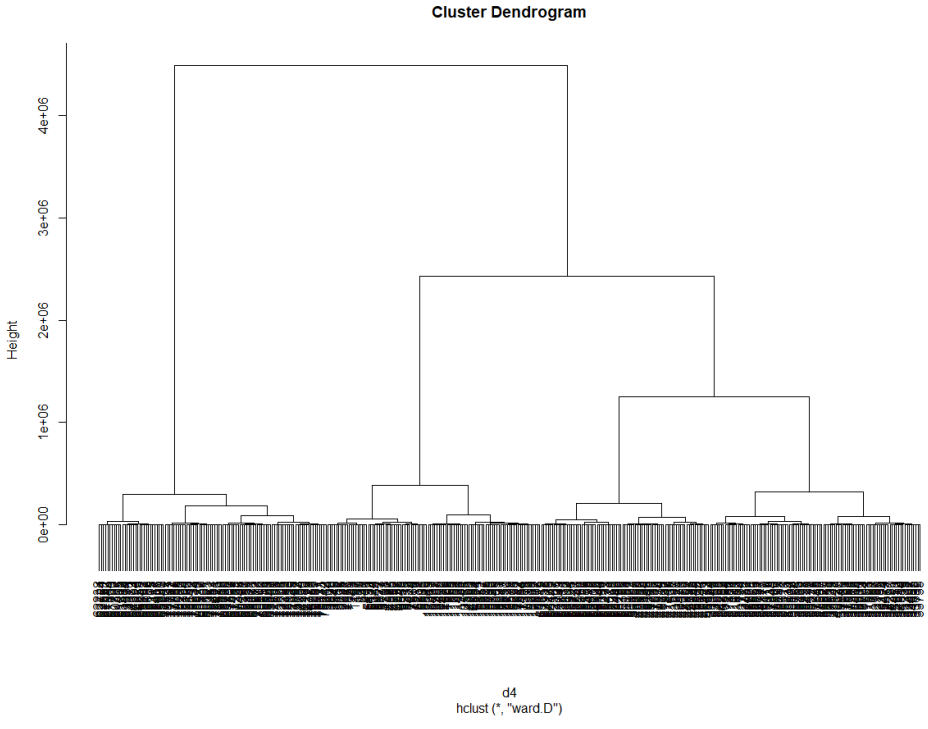
Anexo 111. Dendrograma intento 2 para base de clientes con variables no binarias y binarias, estandarizados con puntaje Z (Fuente: Elaboración propia).



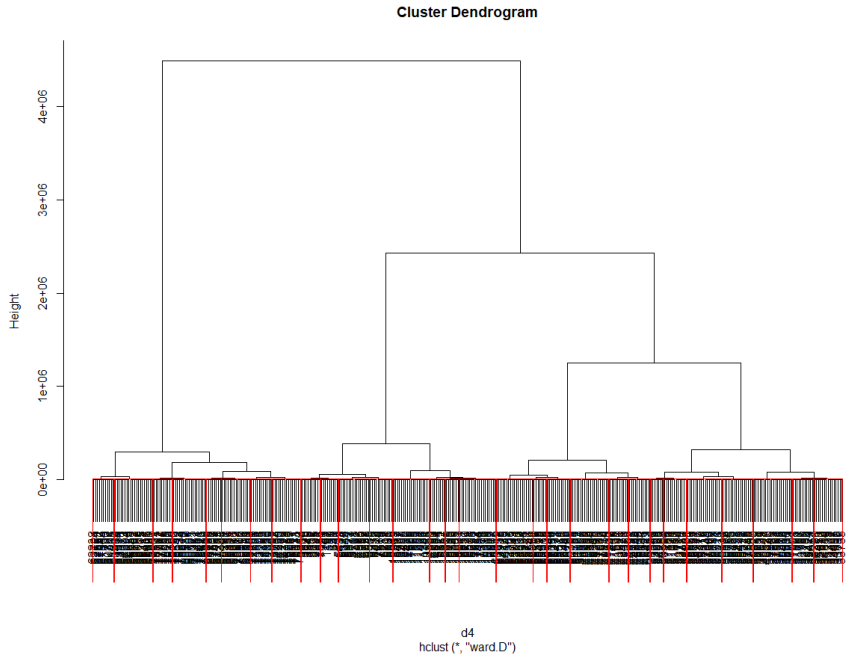
Anexo 112. Dendrograma intento 2 con 27 *clusters* identificados, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



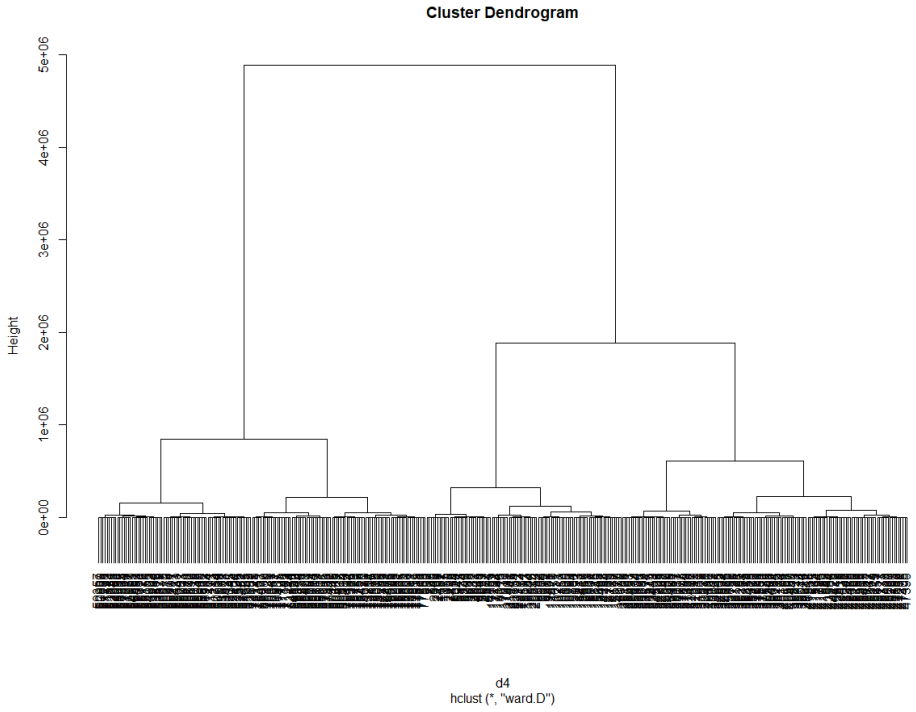
Anexo 113. Dendrograma intento 3 para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



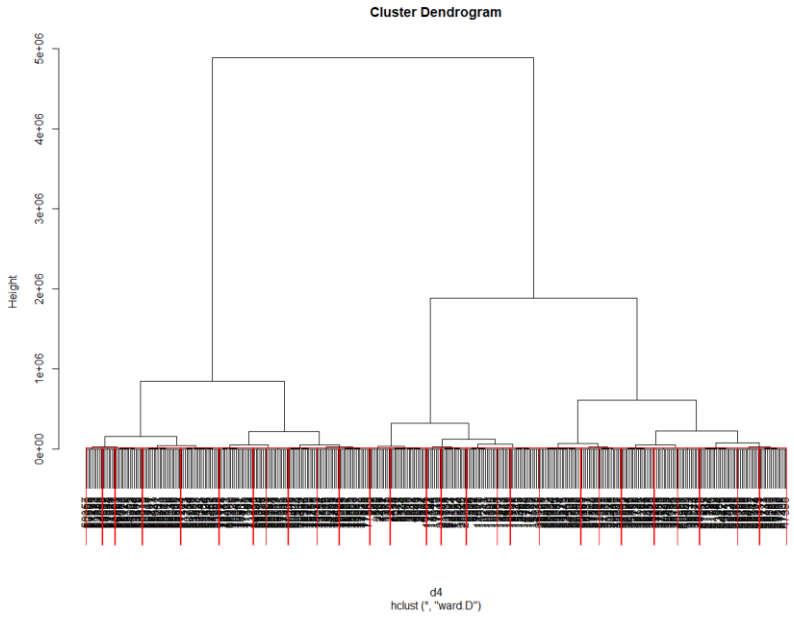
Anexo 114. Dendrograma intento 3 con 29 *clusters* identificados, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



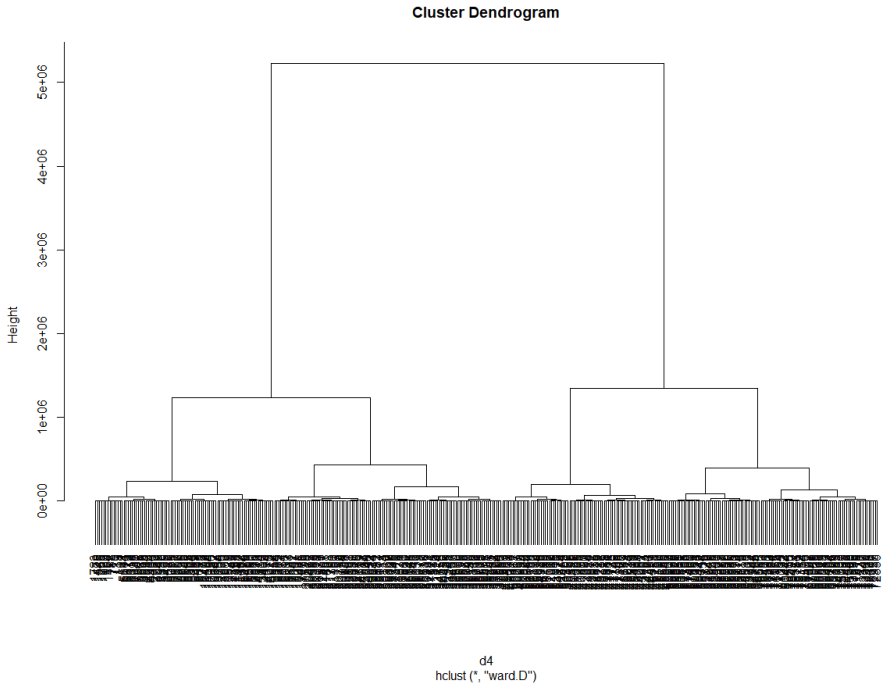
Anexo 115. Dendrograma intento 4 para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



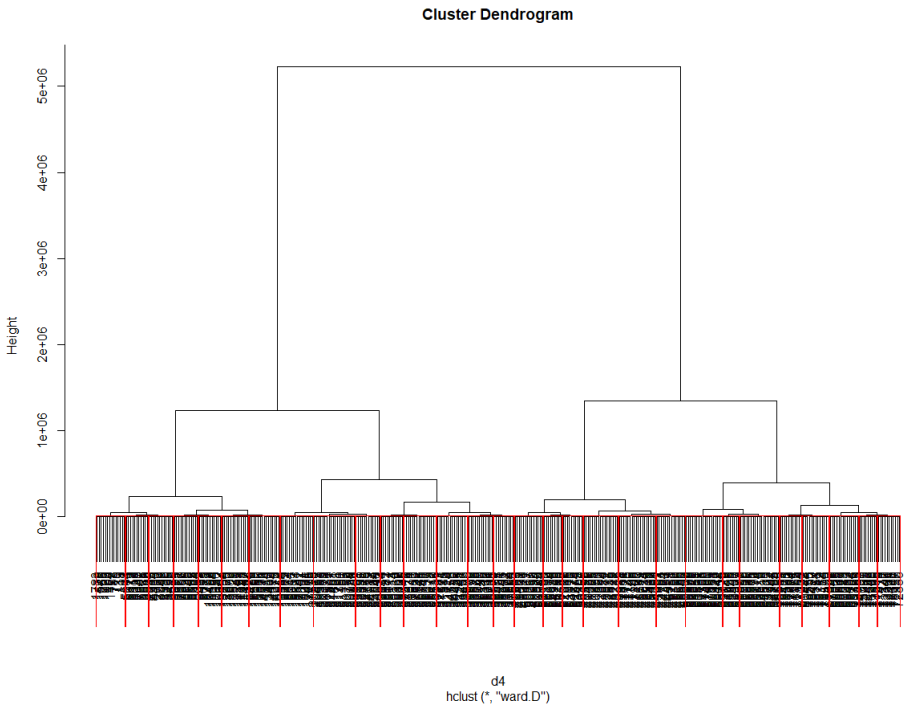
Anexo 116. Dendrograma intento 4 con 27 *clusters* identificados, para base de clientes con variables binarias y no binarias, estandarizados con puntaje Z (Fuente: Elaboración propia).



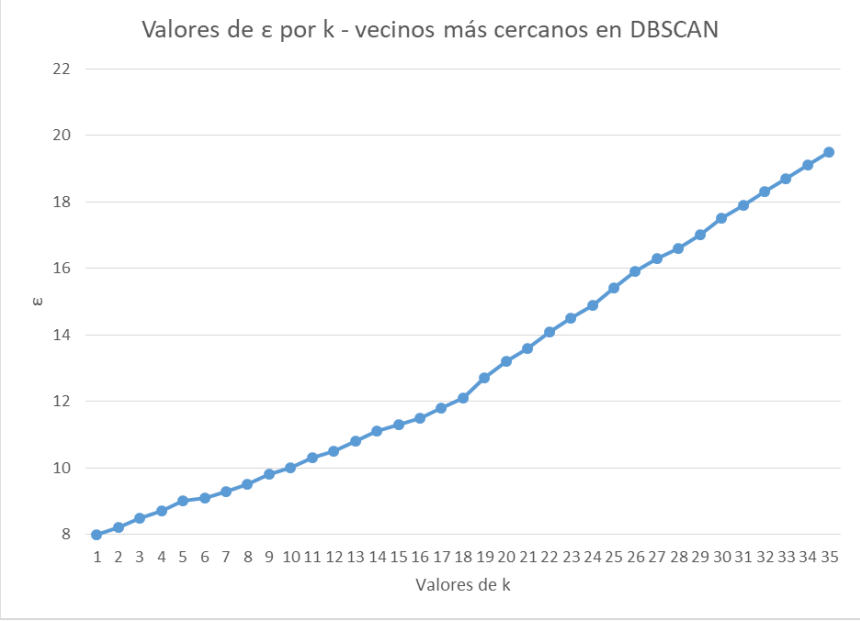
Anexo 117. Dendrograma intento 5 para base de clientes con variables binarias y no binarias, estandarizados con puntaje Z (Fuente: Elaboración propia).



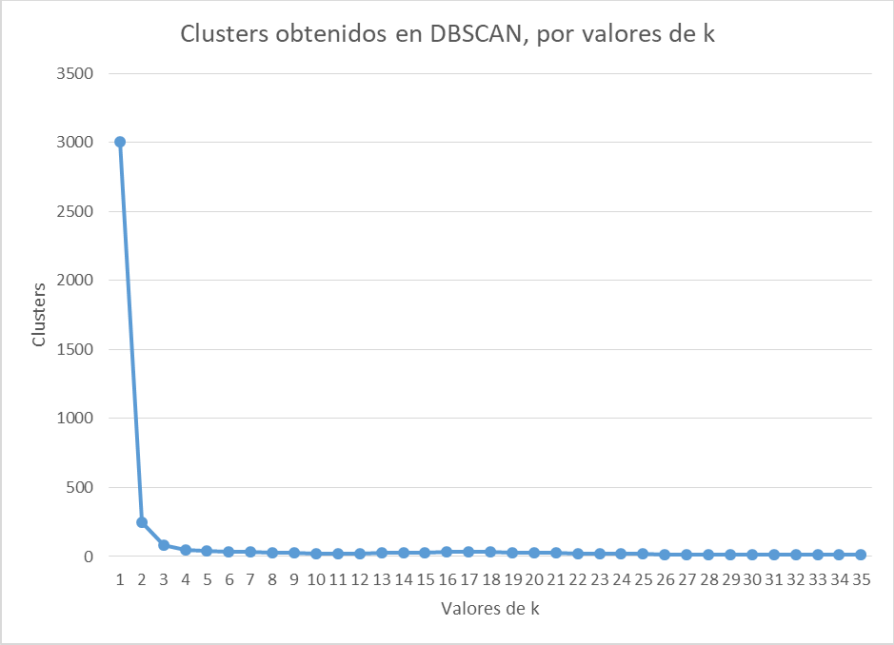
Anexo 118. Dendrograma intento 5 con 29 *clusters* identificados, para base de clientes con variables binarias y no binarias, estandarizados con puntaje Z (Fuente: Elaboración propia).



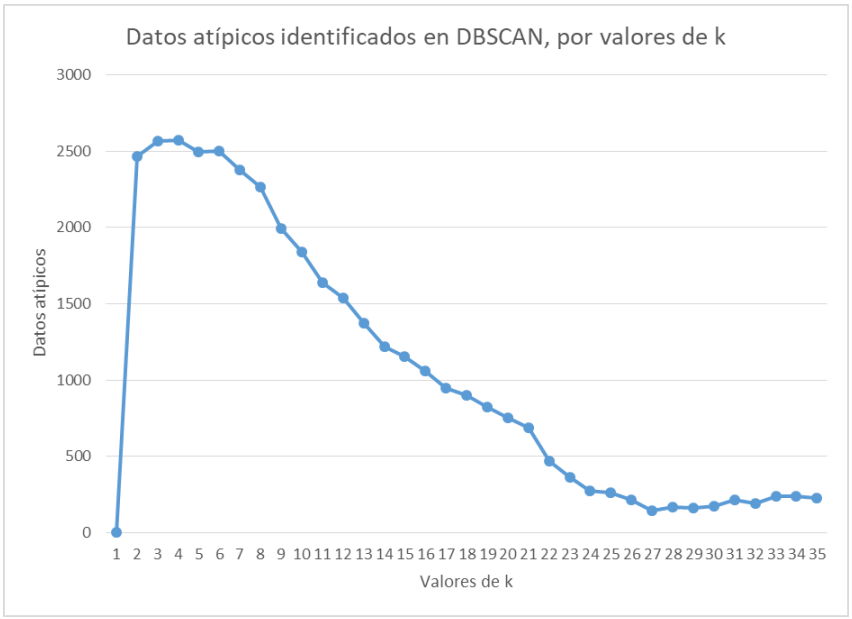
Anexo 119. Valores de épsilon por k – vecinos más cercanos, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



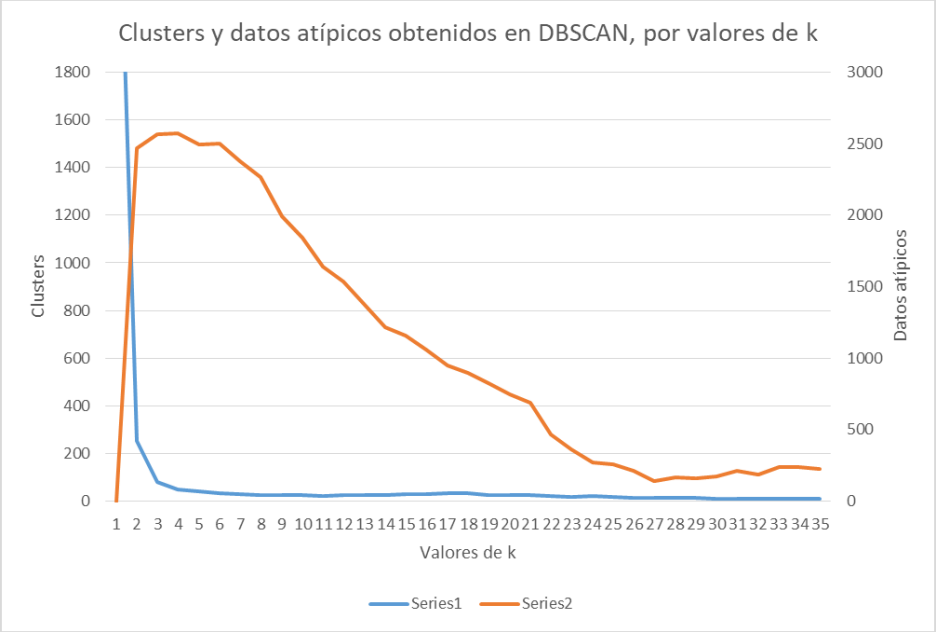
Anexo 120. Cantidad de *clusters* obtenidos en DBSCAN para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



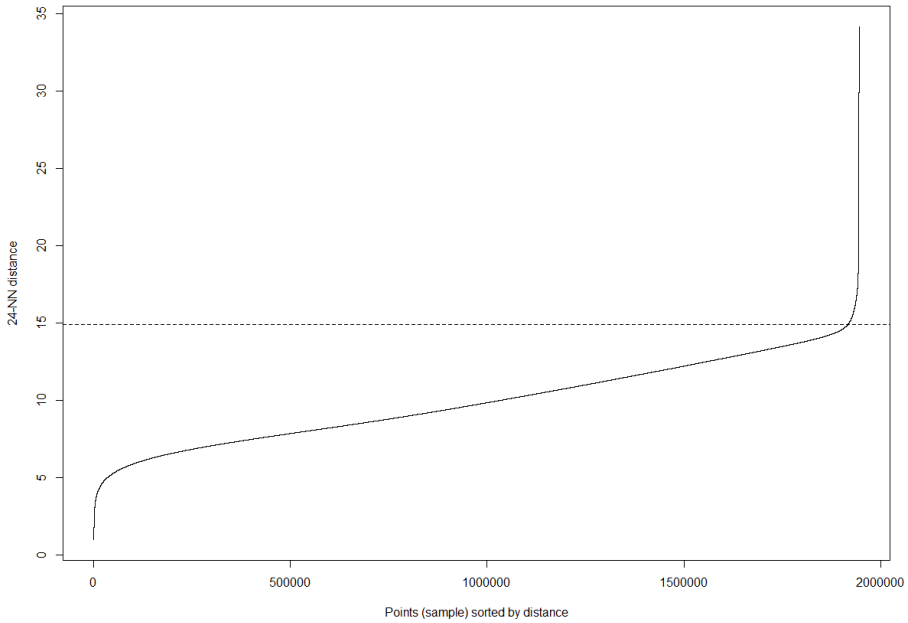
Anexo 121. Cantidad de datos atípicos en DBSCAN, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



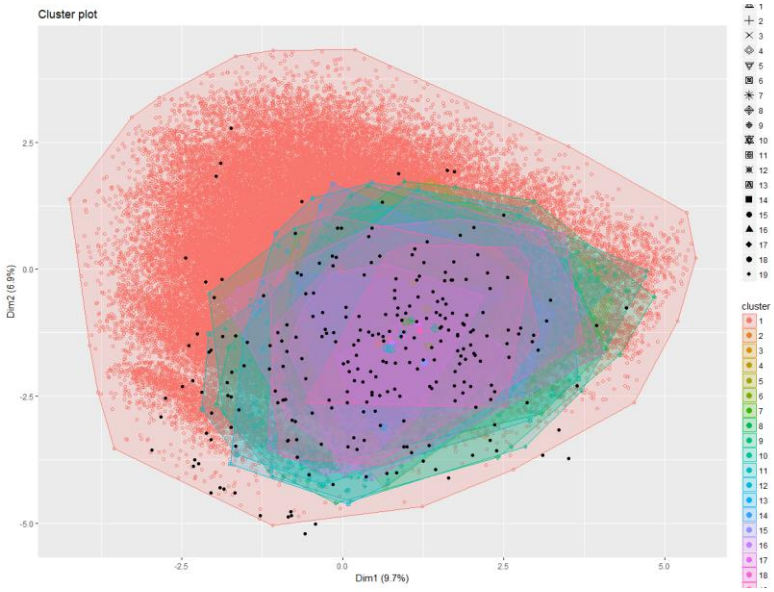
Anexo 122. Cantidad de *clusters* y datos atípicos obtenidos en DBSCAN, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 123. Gráfico de distancias de k – vecinos más cercanos, con *k* = 24, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 124. *Clusters* obtenidos por DBSCAN, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

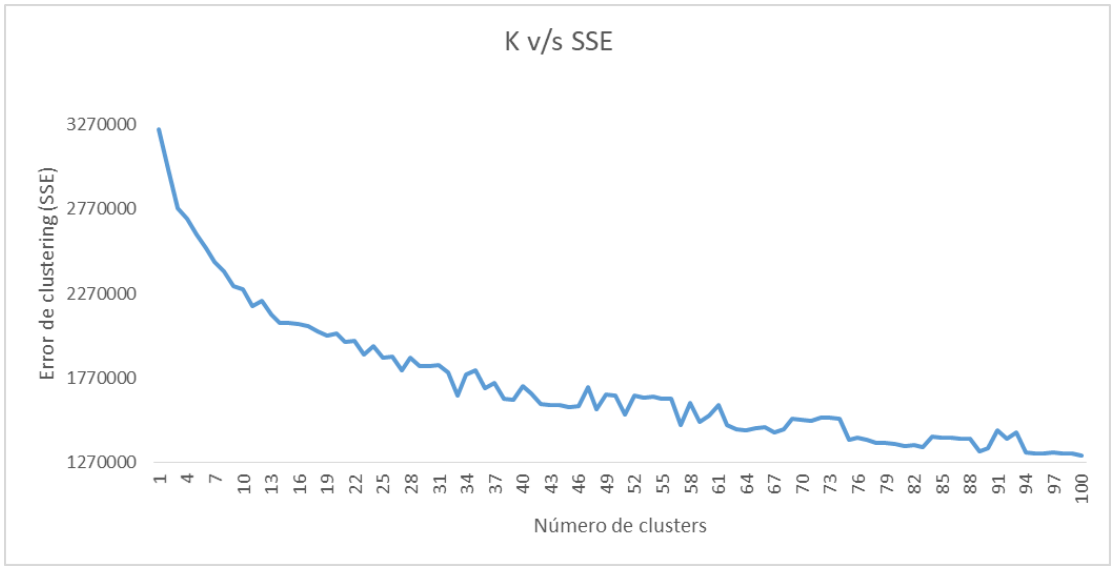


Anexo 125. Código para ejecutar *K – Medoids* con base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).

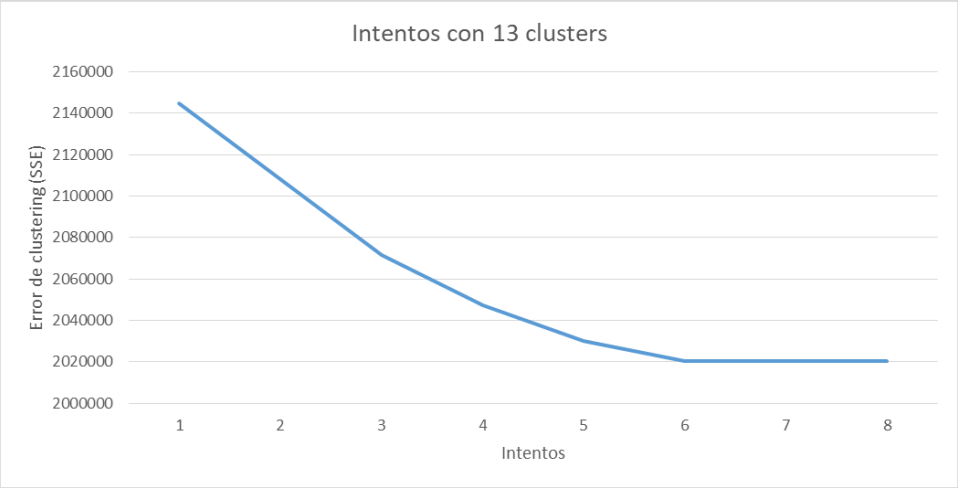
```
library(cluster)
result_kmedoids4 <- clara(base2_4_v1_v2_df[,c(-1)], 13, samples = 2060) #Usar samples para hacer las pruebas.
result_kmedoids4$i.med
#Unir base con índices de clusters
result_db_kmedoids4 <- data.frame(cbind(base2_4_v1_v2_df, clusterNum = result_kmedoids4$clustering))
#Calcular SSE
clusters_kmedoids_seq <- seq(1, 13)
sse_kmedoids_final <- rep(0, 13)
for (x in clusters_kmedoids_seq) {
  datos_cluster <- result_db_kmedoids4[result_db_kmedoids4$clusterNum==x,]
  centroide <- result_db_kmedoids4[result_kmedoids4$i.med[x],seq(2,30)]
  valor_seq_Ward <- seq(2, 30)
  filas <- nrow(datos_cluster)
```

```
 valor_filas <- seq(1, filas)
 sse_valores <- rep(0, filas)
 for (z in valor_filas) {
  sse_variables <- rep(0, 29)
  for (w in valor_seq_Ward) {
   sse_variables [w - 1] <- (datos_cluster[z,w] - centroide[w - 1])^2
  }
  sse_valores[z] <- sum(sse_variables)
 }
 sse_kmedoids_final[x] <- sum(sse_valores)
}
sse <- sum(sse_kmedoids_final)
sse
#Descargar base
write.csv(result_db_kmedoids4, file =
"resultados_KMedoids_base4_intento8_2020358-0.csv")
```

Anexo 126. Número de *clusters* v/s error de *K – Medoids*, para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 127. Resultados de intentos de *K – Medoids* para base de clientes con variables binarias y no binarias, estandarizadas con puntaje Z (Fuente: Elaboración propia).



Anexo 128. Resultados de *cluster* 2 de modelo de predicción con variables financieras.

	Naïve - Bayes	Regresión logística	Árbol de decisión
Desbalanceado	Accuracy: 89,48% ± 3,59% Sensitivity: 99,44% ± 0,13% Specificity: 1,26% ± 1,11% Precision: 89,92% ± 3,62% F1 - Score: 94,40% ± 2,00% AUC: 53,95% ± 9,30%	Accuracy: 98,46% ± 0,57% Sensitivity: 99,41% ± 0,12% Specificity: 3,69% ± 6,16% Precision: 99,04% ± 0,56% F1 - Score: 99,22% ± 0,29% AUC: 55,28% ± 8,63%	Accuracy: 90,71% ± 17,25% Sensitivity: 96,10% ± 18,15% Specificity: 0,90% ± 1,02% Precision: 91,18% ± 17,46% F1 - Score: 93,56% ± 17,73% AUC: 50,70% ± 7,86%
Downsample	Accuracy: 67,55% ± 22,15% Sensitivity: 99,42% ± 0,20% Specificity: 0,91% ± 0,68% Precision: 67,74% ± 22,40% F1 - Score: 78,09% ± 19,55% AUC: 53,07% ± 8,23%	Accuracy: 83,13% ± 13,89% Sensitivity: 99,41% ± 0,15% Specificity: 0,78% ± 0,64% Precision: 83,53% ± 14,04% F1 - Score: 90,02% ± 10,27% AUC: 52,45% ± 9,29%	Accuracy: 57,05% ± 33,60% Sensitivity: 79,54% ± 40,45% Specificity: 0,56% ± 0,32% Precision: 57,12% ± 34,02% F1 - Score: 65,21% ± 35,54% AUC: 52,70% ± 7,37%
Upsample	Accuracy: 71,01% ± 16,44% Sensitivity: 99,41% ± 0,17% Specificity: 0,86% ± 0,58% Precision: 71,24% ± 16,60% F1 - Score: 81,81% ± 12,86% AUC: 54,05% ± 9,54%	Accuracy: 90,36% ± 3,64% Sensitivity: 99,42% ± 0,14% Specificity: 1,11% ± 0,87% Precision: 90,82% ± 3,64% F1 - Score: 94,89% ± 2,06% AUC: 54,21% ± 8,80%	Accuracy: 88,33% ± 6,83% Sensitivity: 99,43% ± 0,15% Specificity: 1,10% ± 0,88% Precision: 88,77% ± 6,91% F1 - Score: 93,66% ± 4,01% AUC: 52,95% ± 5,40%
SMOTE	Accuracy: 63,11% ± 22,60% Sensitivity: 99,38% ± 0,19% Specificity: 0,81% ± 0,59% Precision: 63,27% ± 22,85% F1 - Score: 74,90% ± 17,99% AUC: 51,99% ± 10,25%	**Accuracy: 84,39% ± 5,03%** **Sensitivity: 99,45% ± 0,15%** **Specificity: 1,00% ± 0,76%** **Precision: 84,76% ± 5,06%** **F1 - Score: 91,44% ± 3,02%** **AUC: 57,43% ± 10,59%**	Accuracy: 73,86% ± 19,33% Sensitivity: 99,42% ± 0,15% Specificity: 0,68% ± 0,56% Precision: 74,13% ± 19,55% F1 - Score: 83,35% ± 14,69% AUC: 51,91% ± 10,23%
SMOTE modificado	Accuracy: 60,97% ± 22,45% Sensitivity: 99,34% ± 0,37% Specificity: 0,88% ± 0,66% Precision: 61,09% ± 22,66% F1 - Score: 73,13% ± 18,97% AUC: 51,37% ± 9,59%	Accuracy: 80,56% ± 13,76% Sensitivity: 99,46% ± 0,13% Specificity: 1,04% ± 0,61% Precision: 80,88% ± 13,91% F1 - Score: 88,44% ± 10,31% AUC: 56,52% ± 7,91%	Accuracy: 63,23% ± 22,84% Sensitivity: 99,42% ± 0,19% Specificity: 0,74% ± 0,38% Precision: 63,38% ± 23,12% F1 - Score: 74,84% ± 18,98% AUC: 52,96% ± 6,76%

Anexo 129. Resultados de *cluster* 8 de modelo de predicción con variables financieras.

	Naïve - Bayes	Regresión logística	Árbol de decisión
Desbalanceado	Accuracy: 88,95% ± 0,87% Sensitivity: 97,56% ± 0,27% Specificity: 6,72% ± 1,35% Precision: 90,89% ± 0,91% F1 - Score: 94,11% ± 0,49% AUC: 68,49% ± 1,83%	Accuracy: 96,19% ± 1,43% Sensitivity: 97,23% ± 0,28% Specificity: 10,30% ± 7,52% Precision: 98,90% ± 1,57% F1 - Score: 98,05% ± 0,76% AUC: 67,33% ± 1,89%	Accuracy: 91,70% ± 1,68% Sensitivity: 97,43% ± 0,32% Specificity: 6,57% ± 2,40% Precision: 93,94% ± 1,92% F1 - Score: 95,64% ± 0,91% AUC: 68,31% ± 1,86%
Downsample	Accuracy: 30,04% ± 11,19% Sensitivity: 99,45% ± 0,59% Specificity: 3,70% ± 0,54% Precision: 28,22% ± 11,96% F1 - Score: 42,96% ± 10,70% AUC: 68,09% ± 2,13%	Accuracy: 88,89% ± 3,83% Sensitivity: 97,50% ± 0,26% Specificity: 6,30% ± 1,84% Precision: 90,89% ± 4,04% F1 - Score: 94,03% ± 2,25% AUC: 66,91% ± 2,33%	Accuracy: 82,78% ± 23,38% Sensitivity: 94,20% ± 17,80% Specificity: 5,49% ± 2,38% Precision: 84,59% ± 24,80% F1 - Score: 87,52% ± 22,84% AUC: 65,21% ± 2,76%
Upsample	**Accuracy: 29,96% ± 11,56%** **Sensitivity: 99,45% ± 0,43%** **Specificity: 3,77% ± 0,82%** **Precision: 28,11% ± 12,22%** **F1 - Score: 42,80% ± 10,89%** **AUC: 68,54% ± 1,85%**	Accuracy: 89,83% ± 1,31% Sensitivity: 97,48% ± 0,28% Specificity: 6,36% ± 1,31% Precision: 91,92% ± 1,40% F1 - Score: 94,61% ± 0,74% AUC: 67,25% ± 2,05%	Accuracy: 83,89% ± 16,65% Sensitivity: 97,49% ± 0,68% Specificity: 4,98% ± 2,14% Precision: 85,74% ± 17,87% F1 - Score: 89,95% ± 12,83% AUC: 63,94% ± 2,53%
SMOTE	Accuracy: 30,84% ± 3,90% Sensitivity: 99,42% ± 0,32% Specificity: 3,73% ± 0,43% Precision: 29,00% ± 4,06% F1 - Score: 44,75% ± 4,75% AUC: 68,20% ± 1,92%	Accuracy: 88,64% ± 2,49% Sensitivity: 97,49% ± 0,27% Specificity: 5,97% ± 1,33% Precision: 90,64% ± 2,67% F1 - Score: 93,92% ± 1,42% AUC: 67,17% ± 2,10%	Accuracy: 89,37% ± 11,05% Sensitivity: 97,35% ± 0,56% Specificity: 5,44% ± 2,53% Precision: 91,60% ± 11,84% F1 - Score: 93,80% ± 9,08% AUC: 63,89% ± 2,86%
SMOTE modificado	Accuracy: 28,81% ± 4,42% Sensitivity: 99,49% ± 0,35% Specificity: 3,66% ± 0,40% Precision: 26,88% ± 4,62% F1 - Score: 42,13% ± 5,38% AUC: 68,49% ± 2,12%	Accuracy: 88,28% ± 2,54% Sensitivity: 97,50% ± 0,26% Specificity: 5,97% ± 1,23% Precision: 90,25% ± 2,66% F1 - Score: 93,72% ± 1,45% AUC: 67,03% ± 2,28%	Accuracy: 83,01% ± 16,66% Sensitivity: 97,51% ± 0,62% Specificity: 4,96% ± 1,61% Precision: 84,75% ± 17,82% F1 - Score: 89,31% ± 14,05% AUC: 64,47% ± 2,35%

Anexo 130. Resultados de *cluster* 15 de modelo de predicción con variables financieras.

	Naïve - Bayes	Regresión logística	Árbol de decisión
Desbalanceado	Accuracy: 51,52% ± 3,84% Sensitivity: 80,79% ± 1,92% Specificity: 35,38% ± 2,29% Precision: 41,07% ± 6,79% F1 - Score: 54,13% ± 5,76% AUC: 62,45% ± 1,43%	Accuracy: 59,71% ± 5,70% Sensitivity: 78,72% ± 3,01% Specificity: 39,37% ± 3,34% Precision: 59,62% ± 15,51% F1 - Score: 66,51% ± 8,43% AUC: 62,90% ± 1,22%	**Accuracy: 58,82% ± 3,27%** **Sensitivity: 79,71% ± 2,51%** **Specificity: 38,40% ± 2,06%** **Precision: 56,26% ± 8,83%** **F1 - Score: 65,44% ± 5,43%** **AUC: 64,42% ± 1,08%**
Downsample	Accuracy: 40,00% ± 9,82% Sensitivity: 52,91% ± 41,10% Specificity: 32,18% ± 2,69% Precision: 19,00% ± 18,27% F1 - Score: 26,51% ± 24,81% AUC: 61,85% ± 1,33%	Accuracy: 43,08% ± 13,38% Sensitivity: 47,03% ± 42,00% Specificity: 33,78% ± 4,52% Precision: 24,86% ± 25,44% F1 - Score: 31,13% ± 30,17% AUC: 62,44% ± 1,32%	Accuracy: 30,50% ± 2,41% Sensitivity: 19,16% ± 38,99% Specificity: 29,77% ± 1,38% Precision: 1,51% ± 3,16% F1 - Score: 2,80% ± 5,81% AUC: 63,99% ± 1,29%
Upsample	Accuracy: 42,51% ± 9,38% Sensitivity: 63,43% ± 35,82% Specificity: 32,72% ± 2,61% Precision: 23,87% ± 17,39% F1 - Score: 33,23% ± 23,30% AUC: 62,12% ± 1,36%	Accuracy: 46,79% ± 11,83% Sensitivity: 57,32% ± 38,22% Specificity: 34,63% ± 3,56% Precision: 31,53% ± 22,62% F1 - Score: 40,19% ± 27,38% AUC: 62,49% ± 1,30%	Accuracy: 30,33% ± 2,34% Sensitivity: 15,05% ± 34,29% Specificity: 29,71% ± 1,32% Precision: 1,32% ± 3,26% F1 - Score: 2,40% ± 5,91% AUC: 63,77% ± 1,44%
SMOTE	Accuracy: 46,91% ± 7,02% Sensitivity: 73,68% ± 25,25% Specificity: 33,98% ± 2,21% Precision: 31,95% ± 13,23% F1 - Score: 43,85% ± 17,77% AUC: 62,04% ± 1,44%	Accuracy: 53,13% ± 6,62% Sensitivity: 75,69% ± 20,62% Specificity: 36,40% ± 2,38% Precision: 43,59% ± 12,92% F1 - Score: 55,13% ± 15,53% AUC: 62,45% ± 1,36%	Accuracy: 34,67% ± 3,03% Sensitivity: 79,48% ± 31,92% Specificity: 30,91% ± 1,56% Precision: 8,09% ± 4,31% F1 - Score: 14,55% ± 7,53% AUC: 63,00% ± 1,47%
SMOTE modificado	Accuracy: 37,04% ± 9,40% Sensitivity: 39,30% ± 42,97% Specificity: 31,44% ± 2,20% Precision: 13,53% ± 17,92% F1 - Score: 18,73% ± 24,53% AUC: 62,46% ± 1,26%	Accuracy: 43,38% ± 12,23% Sensitivity: 46,70% ± 41,58% Specificity: 33,58% ± 3,41% Precision: 24,94% ± 23,55% F1 - Score: 32,08% ± 29,02% AUC: 62,46% ± 1,31%	Accuracy: 31,63% ± 2,66% Sensitivity: 43,02% ± 46,83% Specificity: 30,06% ± 1,37% Precision: 3,33% ± 4,01% F1 - Score: 6,13% ± 7,31% AUC: 63,30% ± 1,64%

Anexo 131. Resultados de *cluster* 2 de modelo de predicción con variables operacionales.

	Naïve - Bayes	Regresión logística	Árbol de decisión
Desbalanceado	Accuracy: 57,93% ± 47,21% Sensitivity: 99,15% ± 0,37% Specificity: 0,28% ± 0,33% Precision: 58,03% ± 47,81% F1 - Score: 59,79% ± 45,93% AUC: 48,53% ± 3,92%	Accuracy: 48,62% ± 39,07% Sensitivity: 99,45% ± 0,26% Specificity: 0,47% ± 0,40% Precision: 48,60% ± 39,67% F1 - Score: 56,64% ± 33,22% AUC: 50,94% ± 4,33%	Accuracy: 7,51% ± 5,17% Sensitivity: 99,83% ± 0,32% Specificity: 0,71% ± 0,16% Precision: 6,90% ± 5,25% F1 - Score: 12,49% ± 8,56% AUC: 51,55% ± 1,32%
Downsample	Accuracy: 58,54% ± 46,60% Sensitivity: 82,58% ± 37,57% Specificity: 0,31% ± 0,38% Precision: 58,66% ± 47,28% F1 - Score: 60,15% ± 47,04% AUC: 50,03% ± 4,60%	Accuracy: 36,78% ± 44,65% Sensitivity: 59,36% ± 49,31% Specificity: 0,43% ± 0,36% Precision: 36,61% ± 45,30% F1 - Score: 38,82% ± 45,63% AUC: 49,90% ± 5,24%	Accuracy: 40,68% ± 15,52% Sensitivity: 99,42% ± 0,28% Specificity: 0,63% ± 0,24% Precision: 40,56% ± 15,73% F1 - Score: 55,63% ± 18,20% AUC: 50,73% ± 2,21%
Upsample	Accuracy: 73,07% ± 42,53% Sensitivity: 98,90% ± 0,74% Specificity: 0,16% ± 0,27% Precision: 73,46% ± 43,10% F1 - Score: 74,34% ± 41,83% AUC: 47,81% ± 3,14%	Accuracy: 33,53% ± 43,13% Sensitivity: 99,73% ± 0,36% Specificity: 0,49% ± 0,35% Precision: 33,31% ± 43,76% F1 - Score: 36,58% ± 41,71% AUC: 52,38% ± 5,01%	Accuracy: 13,78% ± 34,08% Sensitivity: 13,26% ± 34,38% Specificity: 0,55% ± 0,25% Precision: 13,30% ± 34,49% F1 - Score: 13,28% ± 34,44% AUC: 52,41% ± 1,68%
SMOTE	Accuracy: 45,69% ± 46,43% Sensitivity: 99,52% ± 0,42% Specificity: 0,50% ± 0,59% Precision: 45,63% ± 47,11% F1 - Score: 48,22% ± 45,62% AUC: 51,13% ± 4,65%	**Accuracy: 26,34% ± 39,03%** **Sensitivity: 99,69% ± 0,40%** **Specificity: 0,53% ± 0,32%** **Precision: 26,03% ± 39,61%** **F1 - Score: 29,47% ± 38,52%** **AUC: 53,47% ± 4,83%**	Accuracy: 6,94% ± 24,07% Sensitivity: 6,62% ± 25,19% Specificity: 0,58% ± 0,21% Precision: 6,41% ± 24,39% F1 - Score: 6,51% ± 24,78% AUC: 51,45% ± 2,91%
SMOTE modificado	Accuracy: 35,22% ± 45,27% Sensitivity: 89,45% ± 30,33% Specificity: 0,47% ± 0,38% Precision: 35,01% ± 45,92% F1 - Score: 36,90% ± 45,04% AUC: 50,28% ± 4,80%	Accuracy: 22,27% ± 37,05% Sensitivity: 92,89% ± 25,26% Specificity: 0,55% ± 0,31% Precision: 21,89% ± 37,59% F1 - Score: 24,75% ± 37,13% AUC: 53,08% ± 4,89%	Accuracy: 6,87% ± 23,77% Sensitivity: 6,63% ± 25,22% Specificity: 0,59% ± 0,21% Precision: 6,32% ± 24,06% F1 - Score: 6,47% ± 24,63% AUC: 50,97% ± 2,67%