UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA

FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

"CARACTERIZACIÓN DEL PERFIL DEL INGRESANTE DE UNA UNIVERSIDAD PÚBLICA APLICANDO ALGORITMOS CLUSTERING K-PROTOTYPES Y K-MEDOIDS"

TESIS PARA OPTAR EL TÍTULO DE
INGENIERO ESTADÍSTICO INFORMÁTICO

PRESENTADO POR:
LEDVIR AYRTON WALTER CHAVEZ VALDERRAMA

LIMA – PERÚ

2020

UNIVERSIDAD NACIONAL AGRARIA
LA MOLINA

FACULTAD DE ECONOMÍA Y PLANIFICACIÓN

"CARACTERIZACIÓN DEL PERFIL DEL INGRESANTE DE UNA UNIVERSIDAD PÚBLICA APLICANDO ALGORITMOS CLUSTERING K-PROTOTYPES Y K-MEDOIDS"

TESIS PARA OPTAR EL TÍTULO DE

INGENIERO ESTADÍSTICO INFORMÁTICO

PRESENTADO POR:

LEDVIR AYRTON WALTER CHAVEZ VALDERRAMA

SUSTENTADA Y APROBADA ANTE EL SIGUIENTE JURADO:

M.A. Fernando René Rosas Villena PRESIDENTE	Mg. Jesús Walter Salinas Flores ASESOR
M.S. Luz Jeanet Bullón Camarena MIEMBRO	Mg. Iván Dennys Soto Rodríguez MIEMBRO

LIMA – PERÚ

2020

DEDICATORIA

Este trabajo fruto de mi esfuerzo y constancia va dedicado a mis padres, principales promotores de mis sueños, quienes me enseñaron que todas las cosas hay que valorarlas y trabajarlas, luchando día a día por alcanzar mis objetivos.

A mi madre, Mercy Valderrama Vizcardo, por mostrarme el camino hacia la superación, con su apoyo, consejos, comprensión, amor, ayudándome cuando la necesitaba, inspirarme a seguir adelante, sembrando en mí la semilla de la responsabilidad y el deseo de triunfar.

A mi padre, Walter Chavez Chacon, por el apoyo brindado, cada palabra de aliento, por desear y anhelar siempre lo mejor para mí, por su paciencia y preocupación constante.

A mis familiares, por su cariño y palabras de aliento que son de gran apoyo para lograr las metas que tengo trazadas, para un futuro mejor y ser orgullo para ellos, sin olvidarme de aquellos que ahora me acompañan espiritualmente y viven en mis recuerdos, porque gracias a sus consejos soy lo que soy.

A mis queridos amigos y colegas, que me brindaron la mano cuando más la necesitaba, por permitirme aprender más de la vida a su lado, a reír conmigo y ayudarme a no dejarme vencer antes las adversidades.

Y, sobre todo, una de las cosas que me ayudó a culminar este proyecto es la fe que tengo en Dios, pues por él, el forjador de mi camino, puedo sonreír ante todos mis logros y este trabajo de tesis ha sido una gran bendición en todo sentido.

AGRADECIMIENTOS

El principal agradecimiento es a Dios, por ayudarme a terminar este proyecto, gracias por darme la fuerza, el coraje para hacer este sueño realidad y por estar presente no solo en esta etapa tan importante de mi vida, sino en todo momento ofreciéndome lo mejor.

A mis maestros, que me ayudaron a tomar riesgos, a hacer de los temores, oportunidades de crecer y que a pesar de los frecuentes desvelos que me hicieron pasar, hoy veo los frutos de esas largas y a veces interminables noches, gracias por brindarme su tiempo, por el esfuerzo que dedicaron al compartir sus conocimientos, sin su instrucción profesional no habría llegado a este nivel, por educarme para la vida, por guiarme en la senda del saber y aconsejarme cuando más lo necesitaba.

Este trabajo de tesis no habría sido posible sin la exquisita colaboración del personal que labora en las oficinas de Estudios y Registros Académicos, del Centro de Admisión y Promoción y la Oficina de Bienestar Universitario y Asuntos Estudiantiles de la Universidad Nacional Agraria La Molina, quienes me brindaron el apoyo para obtener los datos necesarios para la aplicación de esta tesis, al mismo tiempo les agradezco cada charla que tuve con ellos, permitiéndome entender de manera más íntegra y enriquecedora la importancia que tiene desarrollar investigación en torno a la educación superior, buscando cada día aumentar el bienestar del universitario.

Quiero agradecer desde estas páginas al Mg. Jesús Walter Salinas Flores, mi docente asesor, por su continuo apoyo, el tiempo dedicado a revisar el manuscrito y haber hecho valiosísimas aportaciones al mismo, por las muy acertadas orientaciones y la nada fácil tarea que ha tenido al evitar que me derrumbe en las múltiples vicisitudes que han ocurrido durante la escritura de esta tesis. Es difícil que estas breves palabras puedan mostrar todo el agradecimiento que siento por él, en lo que respecta a este trabajo y, por supuesto, en lo personal. Gracias a la vida por este nuevo triunfo, a los miembros del Jurado Calificador por sus sugerencias y gracias a todas las personas que me apoyaron y creyeron en la realización de esta tesis.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

RESUMEN

En el presente trabajo de investigación se realizó un estudio comparativo de algoritmos no supervisados para la caracterización del perfil del ingresante de una universidad pública respecto a sus variables sociodemográficas, económicas y de rendimiento académico utilizando algoritmos de segmentación K-prototypes y K-medoids, con el fin de generar conocimientos valiosos y útiles para lograr una mejor comprensión de la diversidad de universitarios que ingresan y con ello conocer el tipo de estudiante que la institución forma, La aplicación se efectuó con datos de alumnos ingresantes a la Universidad Nacional Agraria La Molina de los ciclos académicos 2015-I y 2015-II de las modalidades de Concurso Ordinario y Dos Primeros Puestos de Colegios de Educación Secundaria con un total de 690 postulantes. Se realizó el preprocesamiento de los datos y la aplicación de algoritmos clustering trabajando tanto con variables cuantitativas como cualitativas, para luego determinar el número óptimo de conglomerados y el algoritmo más adecuado utilizando índices de validación interna. Se realizó la validación de los clusters obtenidos de manera univariada (análisis de variancia o ANOVA y prueba Chi cuadrado) y multivariada (algoritmo Boruta y árbol C5.0), por último, se determinó las variables más importantes para caracterizar el perfil de los ingresantes. Con la investigación realizada se logró identificar 3 tipos de alumnos: Ingresante previsto, Ingresante en proceso y el Ingresante en inicio; cada uno con características peculiares, las cuales permitirán a los responsables de las políticas educativas y en especial a los profesores consejeros saber el tipo de alumno que tienen a su cargo desde que ingresa a la universidad y empezar con ello políticas educativas como el emprendimiento del acompañamiento especializado, sistemático e integral; buscando la realización del paradigma del aprendizaje que la universidad se ha propuesto en su Modelo Educativo.

Palabras clave: perfil del ingresante, algoritmos de agrupamiento, segmentación, K-prototype, K-medoid.

ABSTRACT

In the present research work, a comparative study of unsupervised algorithms was carried out to characterization of the profile of the admitted student of a public university with respect their sociodemographic, economic and academic performance variables using K-prototypes and K-medoids segmentation algorithms, in order to generate valuable and useful knowledge to achieve a better understanding of the diversity of admitted university students and to know the kind of student institution will form academically. The application was effected in data of admitted students to the National Agrarian University La Molina of academic cycles 2015-I and 2015-II of the modalities Ordinary Admission exam and Top two Positions Secondary Schools with a total of 690 candidates of higher education. The data were preprocessed and the clustering algorithms were applied, I worked with quantitative and qualitative variables to determine the optimal number of clusters and the most appropriate algorithm using internal validation indices. The clusters obtained were validated using univariate analysis (variance analysis or ANOVA and Chi square test) and multivariate (Boruta and C5.0 tree algorithm). Finally, the most important variables were determined to characterize the profile of the admitted students. Based on the research work, it was possible to identify 3 kinds of students: Expected Admitted Student, Admitted Student in Process and Beginner Admitted Student, each with peculiar features, which will allow responsible of educational policies and in particular for the advisory teachers to know the kind of student, whom they are responsible from the moment they are admitted to the university and begin with educational policies such as specialist, systematic and integral monitoring always looking for the realization of the learning paradigm that the university has proposed in its Educational Model.

Keywords: admitted student profile, clustering algorithms, segmentation, K-prototype, K-medoid

I. INTRODUCCIÓN

En la actualidad, la gestión de los datos ha tomado gran importancia en el mundo de los negocios, debido a que la recopilación detallada y análisis profundo de estos, provee de información y conocimiento potencialmente valioso, antes insospechado, que permite la toma de decisiones de manera acertada y pertinente. Esto es fundamental para numerosas y variadas áreas tales como Marketing, Medicina, Ecommerce, Genética, entre otras; pero un área, relativamente nueva en este tema, que se ha visto enfrentada a grandes retos bajo el impacto de la globalización, el crecimiento económico y la innovación tecnológica es la educación superior, ámbito donde es necesario y relevante, poseer herramientas de gestión que permitan tomar decisiones académicas, dar respuestas a las preguntas de investigación y elaborar estrategias a partir del conocimiento oportuno de la información. Esto no solo repercute directamente sobre la funcionalidad de los departamentos académicos, también podrían influir sobre otras actividades como las evaluaciones y acreditaciones de carreras o de las mismas instituciones.

Hoy en día, se están desarrollando métodos para explorar datos provenientes de ambientes relacionados a la educación, buscando entender mejor a los estudiantes, profesores y actores relacionados en sus entornos educacionales con el fin de mejorar los procesos educativos, donde la analítica y estadística han sido llevados al ámbito tecnológico, donde prima la automatización de procesos y la gestión de grandes bases de datos a través de algoritmos de Machine Learning, fórmulas matemáticas que buscan descubrir la mejor solución para una situación dada, con lo cual se descubre información útil que ayuda a los docentes y responsables de las instituciones educativas a determinar la manera adecuada para guiar a sus estudiantes, maximizando su aprendizaje y contribuyendo a la mejora de la calidad de la educación superior.

Uno de los principales tipos de algoritmos de Machine Learning son los de “clustering” de datos, que tienen el propósito de agrupar datos por similitud, sin previa imposición de restricciones por parte del experto o analista, buscando la extracción de características donde

cada objeto queda representado por una colección de descriptores, permitiendo generar información valiosa que será convertida en conocimiento; ejemplos claros de estos algoritmos son K-medoids y K-prototype desarrollados en el presente trabajo de investigación. Esta metodología es de suma importancia en el ámbito de la educación superior, ya que con ella se puede conocer el perfil de los universitarios desde que ingresan, el cual es un tema que preocupa a las instituciones, debido al bajo desempeño que muestran algunos ingresantes en las carreras universitarias, lo que origina un bajo rendimiento académico y en casos extremos, la deserción. Los alumnos ingresantes tienen un perfil académico que es visible en los documentos presentados en la convocatoria al proceso de admisión, sin embargo, esta información dispersa o no analizada no brinda aproximación alguna sobre el perfil real del ingresante, lo cual no permite conocer realmente la población con la que se trabaja y por lo regular, no se tiene respuesta a la interrogante ¿quiénes y cómo son nuestros estudiantes?

Esta preocupación por la mejora continua de la enseñanza-aprendizaje en las instituciones superiores se ven registradas en sus distintos documentos de gestión académica, ejemplo de ello lo encontramos en el modelo educativo de la Universidad Nacional Agraria La Molina (UNALM), en donde se plantea que el conocimiento de las características que poseen los universitarios es vital para la planificación académica de un profesional integral. En este documento se plantean un sinnúmero de políticas educativas, entre ellas, el emprendimiento del acompañamiento especializado, sistemático e integral para la realización del paradigma del aprendizaje, del docente y del estudiante desde que el universitario ingresa; es aquí donde el modelo educativo concibe al profesor como sujeto de la educación que impulsa y motiva las capacidades de los alumnos planificando, diseñando y conduciendo experiencias de aprendizaje, para lo cual su formación exige la adquisición y aplicación de herramientas didácticas suficientes que permitan establecer un vínculo con el estudiante, a fin de poder cumplir con los ideales y objetivos que la universidad se ha propuesto. Para complementar este propósito se plantean diferentes estrategias, tales como Programas de Formación Continua y el Sistema de Tutoría y Consejería Académica, este último tiene la finalidad de elevar la calidad del proceso educativo, a través de la atención personalizada de los problemas que influyen en el desempeño y rendimiento académico del estudiante; para ello es necesario contar con la información básica del perfil del ingresante, la cual permitirá reconocer las fortalezas y debilidades que los estudiantes presentan desde el inicio de su vida

universitaria y a partir de ello, emprender un proceso de orientación estudiantil; contribuyendo así a la dirección, control y administración de la prevención, asistencia y acompañamiento del estudiante de pregrado y posgrado, para superar la problemática que impide el normal desenvolvimiento en los estudios y complementando su desarrollo académico.

Adicionalmente a esto, la evaluación del desempeño académico del ingresante es básica y necesaria para controlar la progresión del rendimiento del alumno en una institución superior. Basándose en este tema crítico, la agrupación de estudiantes en diferentes categorías de acuerdo a sus conocimientos adquiridos en el ámbito escolar y/o universitario se ha convertido en una tarea compleja, ya que, con la segmentación tradicional de estudiantes basada solamente en sus puntajes promedios, es difícil obtener una visión completa y detallada del estado en el que se encuentra el rendimiento de los universitarios. Con la ayuda de los algoritmos de agrupamiento clustering, es posible descubrir las características claves del rendimiento de los estudiantes, utilizar esas características para predicciones futuras y con esto empezar con una atención personalizada del profesor hacia el estudiante desde su ingreso a la universidad.

Sin embargo, la tarea no sólo queda en agrupar a los individuos según sus características sino también en determinar qué algoritmo es adecuado según los tipos de datos disponibles y el particular propósito del análisis. De manera más objetiva, la validez de los clusters también debe investigarse, para lograr determinar y seleccionar el "mejor" algoritmo clustering existen índices de validación interna ampliamente utilizados tales como el índice de Dunn o Calinski Harabasz (un valor más alto de estos índices, indican un mejor agrupamiento de observaciones) y el índice de Davies-Bouldin (un valor más bajo de este índice, indica un mejor agrupamiento de observaciones), de esta manera se puede comparar técnicas de agrupamiento y determinar el algoritmo que muestra mejores resultados según los datos utilizados.

La estructura del presente trabajo de investigación desarrollado contiene cinco partes: primero, desarrollar de manera detallada los algoritmos de segmentación, específicamente clustering K-prototype y K-medoids, segundo, desarrollar estadísticas de validación clustering que permitan comparar la precisión de los algoritmos; tercero, aplicar los

algoritmos clustering para obtener el perfil del ingresante de la Universidad Nacional Agraria La Molina, para esto se utilizó el software R, cuarto, validar los resultados obtenidos en la aplicación para finalmente en la quinta parte, caracterizar el perfil de los ingresantes relacionado al rendimiento académico y situación socioeconómica.

1.1. Justificación de la investigación

Las instituciones de educación superior afrontan muchos problemas en todo momento, algunos de los más complejos que se pueden mencionar son: mejorar la calidad educativa, disminuir la deserción, reducir el traslado de carrera, aminorar la reprobación, prevenir el atraso estudiantil y evitar los bajos índices de eficiencia relacionado con las tasas de graduación, dichos problemas se ven reflejados en las aulas.

La deserción universitaria, en el Perú alcanza el 30%, señalado por Zaragoza (2017) y publicado por Aquino (2017), la proyección de ingresantes a diferentes universidades por año supera los 300,000 de este grupo entre 40,000 y 50,000 jóvenes abandonan sus estudios universitarios anualmente, donde el 70% de los que deciden no continuar pertenecen a universidades privadas y el 30% restante a estatales; las razones, por lo general, de esto son: problemas económicos, falta de vocación en la carrera profesional, falta de apoyo por parte de la universidad y plana universitaria (profesores/orientadores), expectativas defraudadas en la formación y bajo rendimiento académico.

Hoy en día, muchas instituciones de educación superior no conocen a sus estudiantes, lo cual genera el problema de la falta de conocimiento del tipo de estudiante que la institución gestiona y/o debe gestionar, esto debido al deficiente uso de la gestión de datos educacionales para conocer el perfil de los universitarios, no logrando así personalizar la atención a los alumnos para mejorar su aprendizaje desde su ingreso, pudiendo comprender mejor su desempeño y las condiciones en las cuales ellos aprenden, evitando la deserción de los mismos y mejorando el apoyo económico y/o académico que pueda brindar la institución hacia el estudiante.

Eckert et al. (2013) señala que es de suma importancia aprovechar toda información disponible en el ámbito de la enseñanza, lo cual permitirá descubrir conocimientos útiles para lograr una mejor comprensión del proceso de enseñanza – aprendizaje de los estudiantes y de su participación global en el proceso, orientado a la mejora de la calidad y la eficiencia del sistema educativo.

Todo ello es de sumo interés para lograr conocer y gestionar la diversidad de alumnos que ingresan a cada institución superior, ya que estos poseen diversas necesidades y su rendimiento académico varía, por lo que se hace necesario agrupar o segmentar a los ingresantes con características similares para reforzar su aprendizaje de forma personalizada desde el inicio de su vida universitaria. Esto se puede observar en la Universidad Nacional Agraria La Molina, donde los alumnos desde su ingreso tienen distintas características académicas o de preparación previa, de acuerdo a la formación recibida en la educación secundaria y en centros de preparación preuniversitaria. Las diferencias en el rendimiento académico son evidentes y tienen mucho que ver con el posible perfil de ingreso de los alumnos admitidos a través del concurso público de admisión, además los postulantes tienen la oportunidad de decidir por tres carreras de ingreso a la universidad, de esta manera, habrá estudiantes que ingresaron a la primera, segunda o tercera opción, probablemente, los primeros permanecerán en su carrera; los otros serán aquellos que buscarán la oportunidad de hacer el traslado a otras.

Adicionalmente, la universidad tiene estructurado su Modelo Educativo, el cual se sustenta en tres paradigmas para la formación universitaria: del aprendizaje, del estudiante y del docente, este último, además de ser un sujeto de la educación que impulsa y motiva las capacidades del alumno, es concebido como un académico transformador, crítico y reflexivo que acompaña el proceso de aprendizaje de los estudiantes para cumplir con los ideales y objetivos que la universidad propone, para ello es de suma importancia establecer una relación educativa, un vínculo con el estudiante con la finalidad de elevar la calidad del proceso educativo desde que el universitario ingresa; todo esto se complementa con los Programas de Formación Continua y el Sistema de Tutoría y Consejería Académica personalizada, para lograrlo es necesario conocer al estudiante que se gestiona desde su admisión en la institución, en virtud de lo cual es primordial contar con su información; sin embargo, no se cuenta con el perfil general del ingresante, tema que es relevante ante las

notables diferencias en rendimiento académico y por ello surge la necesidad de caracterizar el perfil de ingreso real y más adelante, plantear un perfil de ingreso deseado.

En la actualidad, es necesario caracterizar el perfil de ingreso de los estudiantes admitidos a la Universidad Nacional Agraria La Molina, para conocer mejor al alumno y adecuarlos al perfil requerido en el diseño curricular ofrecido. El resultado debe servir como instrumento de trabajo para mejorar las políticas del servicio educativo de la UNALM y las Facultades, principalmente para saber cómo y a quién brindar el apoyo necesario para mejorar su formación profesional y con ello reducir la brecha entre el perfil deseado y el perfil real de los ingresantes.

En resumen, el presente trabajo de investigación busca aportar de manera significativa en los siguientes ámbitos:

A. Tecnológica

Permite caracterizar el perfil del ingresante de la Universidad Nacional Agraria La Molina, utilizando algoritmos de Machine Learning, a partir de información de diferentes fuentes, promoviendo el proceso de gestión de datos, con el fin de contribuir como línea base para la mejora continua del proceso de enseñanza-aprendizaje en la institución.

B. Institucional

Con el conocimiento generado del trabajo de investigación, se busca mantener informada a la comunidad educativa y a los responsables de las políticas educativas en los diversos niveles, sobre el perfil del ingresante de la universidad, para contribuir a la toma de decisiones e impulsar cambios a favor de la calidad educativa. Con estos resultados, se busca adquirir una sinergia de esfuerzos estudiante – docente, para que este último tenga información del perfil del ingresante en las diferentes carreras dentro de la institución educativa superior; a partir de ello diseñe estrategias y renueve sus espacios de enseñanza de manera personalizada, mejorando la política de acompañamiento y logrando la realización del paradigma del aprendizaje.

C. Académica

La aplicación de algoritmos clustering en datos de estudiantes ha demostrado tener resultados muy prometedores en el rubro de educación superior, ejemplos de ellos se detallan en los antecedentes de la presente investigación; en general al aplicarlos se logró obtener información relevante, tal como el perfil de los estudiantes universitarios, información que es útil para los asesores en las instituciones, permitiéndoles generar una orientación a los educandos mucho más personalizada y oportuna, sobre los cursos ofrecidos en cada semestre, las regulaciones en los diversos aspectos de los procedimientos universitarios, la utilización de los diversos recursos de la universidad, las dificultades que pueden estar enfrentando académicamente, la organización de los planes de estudios, ajustar los cursos obligatorios y electivos dejando en el pasado las consejerías basadas en el simple promedio de notas del estudiante, creando la posibilidad de una visión más profunda de los datos de los estudiantes en las instituciones educativas superiores.

1.2. Objetivos de la investigación

1.2.1. Objetivo general

Caracterizar el perfil de los ingresantes de una universidad pública respecto a sus variables sociodemográficas, económicas y de rendimiento académico utilizando algoritmos de segmentación K-prototypes y K-medoids.

1.2.2. Objetivos específicos

- Determinar el algoritmo de segmentación más adecuado para el estudio de caso, utilizando indicadores de validación interna clustering.
- Encontrar el número de conglomerados óptimo para el estudio de caso, utilizando indicadores de validación interna clustering.
- Identificar las variables más importantes para caracterizar el perfil de los ingresantes de una universidad pública.

II. REVISIÓN DE LITERATURA

2.1.Antecedentes

2.1.1. Antecedentes extranjeros

En el plano internacional, se han realizado investigaciones que buscan destacar la aplicabilidad de técnicas de segmentación en el ámbito universitario, tal es el caso de Oyelade et al. (2010), quienes realizaron en Nigeria un estudio del perfil del estudiante, donde señalaron que la capacidad de supervisar el progreso del rendimiento académico de los universitarios es un tema crítico para la comunidad académica de educación superior. Esta capacidad era necesaria para la toma de decisiones efectivas de los planificadores académicos, por lo cual se implementó un sistema para analizar los resultados de los estudiantes basados en el análisis clustering evaluando el desempeño académico, descubriendo las características claves del rendimiento de los estudiantes, que fueron perfilados en 5 segmentos, aplicando una metodología simple y cuantitativa a un conjunto de datos provenientes de una universidad, llegando a la conclusión que el algoritmo clustering sirve como un buen punto de referencia para supervisar la progresión del rendimiento de los estudiantes en la institución superior y permite generar una mejora en la toma de decisiones de los planificadores académicos en torno al monitoreo del rendimiento de los estudiantes por semestre, lo que promueve mejoras en los resultados académicos en las sesiones académicas subsecuente.

Bedalli et al. (2015), realizaron en Albania una investigación donde aplicaron técnicas clustering a una colección de datos de estudiantes universitarios, donde el objetivo fue proporcionar información más útil a los asesores de su institución sobre los perfiles de los estudiantes, de tal manera que sirva como herramienta de apoyo en sus deberes tales como: la orientación de los estudiantes sobre los cursos ofrecidos en cada semestre, las regulaciones en los diversos aspectos de los procedimientos universitarios, la utilización de los diversos recursos de la universidad y las dificultades que pueden estar enfrentando, para ello

utilizaron análisis fuzzy clustering; los resultados de la investigación reflejaron que al usar este nuevo enfoque, la orientación que brinda el asesor no sólo se basaría en el promedio de calificaciones del estudiante y la información sobre la distribución de los cursos en los años anteriores, sino también que los estudiantes se podían perfilar de una manera más precisa de tal manera que se tenga más información útil en el proceso de asesoramiento estudiantil, además, se resaltó que el descubrir patrones ayuda en varios temas estratégicos como: optimizar la organización de los planes de estudios, ajustar los cursos obligatorios y electivos, preparar mejores métodos de enseñanza, entre otros; creando la posibilidad de una visión más profunda de los datos de los universitarios.

Finalmente, Singh et al. (2016), desarrollaron en la India un estudio de investigación utilizando el algoritmo clustering para categorizar a sus estudiantes en diferentes grupos, con el fin de ayudar a los alumnos y profesores a centrarse en las estrategias de mejora, mediante el seguimiento del rendimiento del estudiante. En el estudio se llegó a la conclusión que el algoritmo clustering es una herramienta que permite entender el desempeño académico del estudiante, por lo que tiene un papel muy importante en las instituciones superiores, siendo las variables importantes para la segmentación: las calificaciones de los estudiantes y si el alumno realizó o no proyectos complementarios o pasantías. Finalmente, señalaron, que la agrupación de los estudiantes en base a estas variables, permite obtener una visión completa del desempeño del estudiante y, a la vez, determinar los detalles de su rendimiento de semestre en semestre.

2.1.2. Antecedentes nacionales

Las investigaciones en el Perú no están ajenas al interés de analizar las características que poseen los estudiantes universitarios y la aplicabilidad de técnicas de segmentación en las instituciones superiores.

Tal es el caso del Instituto Nacional de Estadística e Informática (INEI) en el reporte del Segundo Censo Nacional Universitario 2010 ofreció algunos datos que han permitido caracterizar la población de estudiantes de la Universidad Nacional Agraria La Molina (UNALM). Los reportes señalaron que la UNALM tenía predominancia de estudiantes del sexo masculino (53.4%) sobre el de mujeres (46.6%). La mayor parte de alumnos estudiaron

en colegios particulares (59.0%) en comparación a aquellos que estudiaron en colegios estatales (41.0%). El 82.29% de los estudiantes provienen de la Región Lima, 4.41% de la Región Junín y 2.74% del Callao. De otras regiones, los estudiantes provienen en 1.8% de Ancash, 1.16% de Ayacucho y 1.06% de Ica. Además, que los estudiantes dependían económicamente mayoritariamente de sus padres (86.66%), lo cual es coherente con el 81.36% de estudiantes que indicó que no trabajaba al momento del Censo, también se observó que el porcentaje de estudiantes admitidos a la UNALM mediante el examen de admisión fue de 67.35% y por el Centro pre universitario de la UNALM de 24.13%, los cuales suman 91.47 % del total de estudiantes matriculados en el año 2010. Los ingresantes que no ingresan por el Centro pre universitario, postulan a la UNALM mediante el examen de admisión. Esto hace que mayoritariamente los estudiantes admitidos conozcan el proceso de admisión y la estructura de la prueba.

Arias (2015), realizó un trabajo de investigación con el objetivo de caracterizar el perfil de los ingresantes en los semestres académicos 2011-I, 2011-II, 2012-I y 2012-II de la carrera de Agronomía de la UNALM. La información de los 204 estudiantes provino de la ficha de evaluación de ingreso administrada por la Oficina de Bienestar Universitario y Asuntos Estudiantiles (OBUAE) y de la base de datos de la Oficina de Estudios y Registros Académicos. El primer grupo de variables consideradas estuvo relacionado a aspectos socio-demográficos (edad de ingreso y género), socio-educativas (grado de instrucción del padre y madre), socio-económicas (ingreso total familiar y situación económica del estudiante) y los resultados en las áreas de conocimiento en la educación secundaria, el segundo grupo de variables correspondió a los resultados en los temas considerados en el examen de admisión, la elección de opción de la carrera y el orden de mérito al ingresar, el tercer grupo de variables abarcó los resultados en las asignaturas cursadas en el primer semestre, los promedios ponderados semestral (PPS), acumulado (PPA) y el orden de mérito (OM) en cada semestre académico. Los datos fueron procesados mediante análisis multivariado usando las técnicas de análisis de componentes principales (ACP) y análisis de conglomerados o clustering (AC). Los resultados indicaron que la edad de ingreso de los estudiantes a la carrera de Agronomía estaba entre 17 a 19 años, con una moda de 18. Mediante la técnica ACP las mayores correlaciones positivas fueron encontradas con las asignaturas de Biología, Matemática Básica y Química General y los promedios ponderados semestral (PPS) y acumulado y (PPA). Las correlaciones negativas fueron identificadas en

las variables OM del primer y segundo semestre. Luego con la técnica del AC fueron identificados 8 perfiles de ingresantes en el semestre 2011-I y 7 perfiles de ingresantes en los semestres 2011-II, 2012-I y 2012-II.

Ochoa (2016), realizó un estudio comparativo de técnicas no supervisadas de minería de datos para segmentar a universitarios de la Universidad Católica de Santa María según su desempeño académico, utilizó los algoritmos K-means y PAM dentro del clustering particional y métodos de ward, single, complete, average, mcquitty, median y centroid dentro del clustering jerárquico aglomerativo. Eligió el algoritmo con el que obtuvo mejor calidad de agrupamiento utilizando medidas internas como las distancias intra-cluster e inter-cluster, y el coeficiente de silueta, obteniendo mejores resultados con la técnica de clustering particional K-means, logrando segmentar a los estudiantes en 3 grupos: alumnos con bajo, medio y avanzado rendimiento académico.

2.2. Marco teórico

2.2.1. Análisis clustering

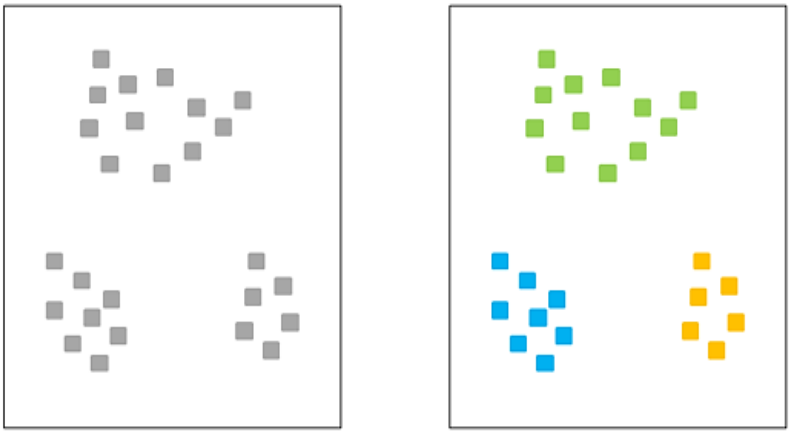
De acuerdo a Beca (2007) el proceso de agrupamiento es una de las más antiguas funciones cerebrales desarrolladas por el hombre. En el siglo V a.c. los filósofos griegos reflexionaban sobre la función cerebral de agrupamiento, a través de preguntas sobre cómo nos es posible conocer la realidad, enfrentar el problema de la descripción de las formas y de la materia mediante atributos o características. En general, es posible afirmar que el hombre identifica, observa, mide y realiza agrupamientos de objetos basándose en sus características con un fin particular. Desde la década de los 70 en adelante las áreas de inteligencia artificial, reconocimiento de patrones y técnicas de aprendizaje han trabajado en el agrupamiento de objetos y la generación de algoritmos que permitan realizarlo de manera automática. En base a estos estudios se han desarrollado diversos algoritmos clustering, los cuales reciben un conjunto de atributos o variables para cada objeto y entregan un particionamiento, donde cada cluster posee un patrón o características similares. La figura 1 ilustra un ejemplo de un resultado del proceso de clustering.

Wang et al. (2007) destacaron que el propósito de cualquier técnica clustering es generar una matriz de partición *U(X)* de un conjunto de datos X (que consta de n objetos, $X = \{x_1, x_2, \ldots, x_n\}$) para encontrar un número, *k* de clusters $(X_1, X_2, \ldots, X_k)$. La matriz de partición *U(X)* de tamaño $k \times n$ puede representarse como $U = [u_{ij}]_{k\times n}$, i = 1, . . . , n y j = 1, . . . , k, donde u_{ij} representa el grado de pertenencia del objeto x_i al clusters X_j. En una segmentación no difusa de los datos, se cumple la condición siguiente: $u_{ij} = 1$ si $x_i \in X_j$, de lo contrario $u_{ij} = 0$ el objetivo es clasificar el conjunto de datos X tal que:

$$n(X_i) \neq 0 \text{ para } i = 1, 2, \ldots, c$$

$$X_i \cap X_j = 0 \text{ para } i = 1, 2, \ldots, c, j = 1, 2, \ldots, c, \text{ donde } i \neq j,$$

$$\bigcup_{i=1}^{c} X_i = X$$



Datos sin agrupar

Datos después del proceso de agrupamiento

Figura 1: Resultado del proceso clustering

Rai et al. (2010) señalaron que el análisis clustering representa a los datos en segmentos, provocando la pérdida de ciertos detalles finos, pero logrando la simplificación de información, basándose en matemáticas, estadística y análisis numérico. Desde una perspectiva de técnicas de aprendizaje, el análisis clustering corresponden al mundo de patrones ocultos, la búsqueda de segmentos es un aprendizaje no supervisado, y el sistema resultante representa un concepto de datos. Por lo tanto, el análisis clustering es el aprendizaje no supervisado de un concepto de datos ocultos.

Según Velmurugan et al. (2011), el análisis clustering es una importante aplicación para el descubrimiento de conocimiento, análisis estadísticos y compresión de los datos; se ha desarrollado de diversas maneras en las técnicas de aprendizaje, reconocimiento de patrones, optimización y literatura estadística; en general, es la forma más común de aprendizaje no supervisado, es decir, que no existe un experto humano que haya asignado las clases. En la agrupación, la distribución y composición son los mismos datos los que determinarán la pertenencia al clúster.

Adicionalmente a esto Kaur et al. (2013) definieron que un clúster es una colección de objetos que son similares entre sí y son distintos a los objetos de otros clústers, un buen algoritmo de agrupamiento es capaz de identificar segmentos independientemente de la distribución de las observaciones. Otros requisitos de los algoritmos de agrupamiento son la escalabilidad, la capacidad de tratar con datos ruidosos y la insensibilidad al orden de los registros de entrada.

Por su parte Raulji (2014) señaló que los algoritmos clustering juegan un papel importante en el descubrimiento de conocimientos en grandes bases de datos: “El objetivo es que los objetos de un mismo grupo estén relacionados entre sí y no relacionados con los objetos de otros grupos. Clustering es una herramienta matemática que intenta descubrir estructuras o ciertos patrones en un conjunto de datos, donde los objetos tienen un cierto grado de similitud dentro de cada grupo. El análisis clustering es un proceso repetitivo de descubrimiento de conocimiento, se requerirá modificar el parámetro y el preprocesamiento hasta que el resultado obtenga las propiedades deseadas.”

2.2.2. Categorización de los algoritmos clustering

Según Soni et al. (2012) existen un gran número de métodos de agrupamiento, cada uno con sus particularidades, ya que dependen del tipo de datos utilizados y el objetivo específico que se quiere alcanzar. Todos los algoritmos clustering básicamente se pueden clasificar en dos grandes categorías: de partición y jerárquicos, los cuales son explicados brevemente y esquematizados en la figura 2.

El resumen de cada categorización se discute a continuación:

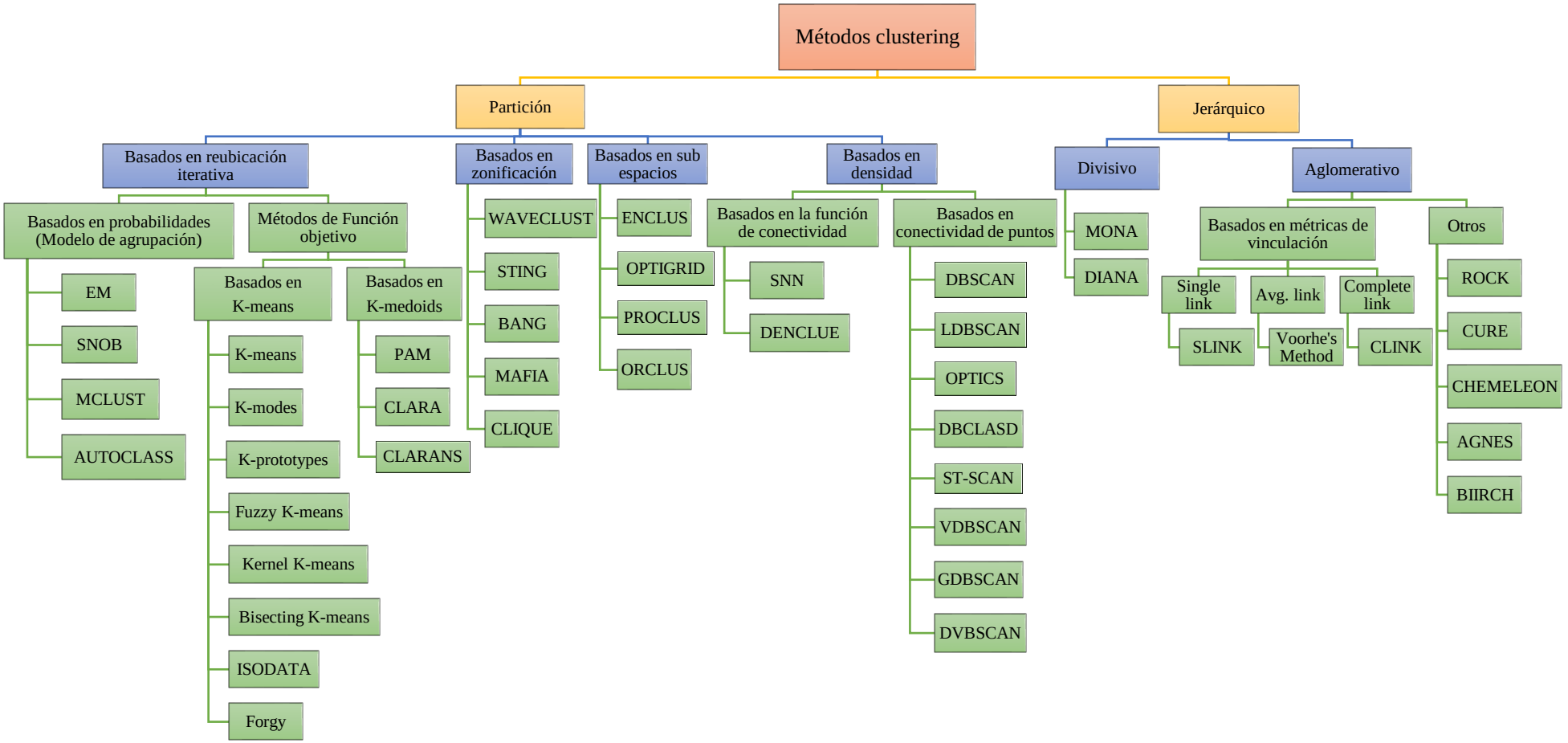


Figura 2: Categorización de diferentes métodos de agrupación

Fuente: Soni et al. (2012)

2.2.2.1. Algoritmos de partición

Los algoritmos clustering de partición, en general, crean *k* particiones a partir del conjunto de datos con *n* objetos, cada partición representa un clúster, donde $k \leq n$. Se divide los datos en subconjuntos o clusters basándose en algunos criterios de evaluación. Como la realización de todas las particiones posibles es computacionalmente inviable, ciertas heurísticas de optimización se utilizan en forma recursiva. Estos métodos se pueden basar en criterios de re-ubicación iterativa, zonificación, sub-espacios y densidad, los cuales se explicarán brevemente a continuación.

A. Algoritmos basados en re-ubicación iterativa

Un enfoque para la partición de datos, es partir de un punto de vista conceptual, encontrar estimadores de máxima verosimilitud de parámetros en modelos probabilísticos que dependen de variables no observables, se pueden conocer como modelos probabilísticos o simplemente agrupamiento basado en modelos. Aquí, el modelo supone que los datos provienen de una mezcla de varias poblaciones cuyas distribuciones y antecedentes se desea encontrar. Los algoritmos representativos son *EM*, *SNOB, AUTOCLASS* y *MCLUST*.

Otro enfoque para la partición se basa en una función objetivo, construyendo puntos equidistantes o centros de cada clúster, dependiendo de cómo se construyan los centroides, los algoritmos de partición iterativos se dividen en *k-means y k-medoids.*
El algoritmo *k-means*, es el más simple bajo este esquema, donde los centros son construidos a partir de las observaciones, a partir de este se han propuesto un gran número de variaciones, algunas de las cuales se pueden enumerar como: *ISODATA*, *Forgy, bisección de k-means*, *x-means*, *kernel k-means*, *k-means ++*, entre otros.

El algoritmo *k-medoids*, donde cada centroide no es construido, por lo contrario, está representado por uno de los objetos dentro de cada cluster*; PAM, CLARA y CLARANS* son tres algoritmos principales propuestos bajo el método.

B. Algoritmos basados en zonificación

Los algoritmos de agrupación basados en zonificación utilizan espacios multidimensionales, es decir, los conglomerados se consideran regiones más densas que su entorno, se difiere de los algoritmos de clustering convencionales en que no se preocupa en los puntos de datos sino al espacio de valores que rodea dichos puntos; para lograr ello, divide el espacio de objetos en un número finito de celdas que forman una estructura de zonas en donde se realizan todas las operaciones de agrupamiento.
Los algoritmos representativos basados en este método son: *STING*, *Wave Cluster* y *CLIQUE.*

C. Algoritmos basados en sub-espacios

Los algoritmos de agrupación de subespacio están diseñados con el objetivo de trabajar con los datos de alta dimensionalidad. Para ello, los métodos generalmente hacen uso del subespacio de la dimensión real, con ello, busca detectar todos los clusters en todos los subespacios, donde una observación de datos puede pertenecer a muchos clústeres diferentes, con cada *clúster* en algún subespacio. Los algoritmos representativos son: *CLIQUE, ENCLUS, MAFIA, PROCLUS y ORCLUS.*

D. Algoritmos basados en densidad

Estos algoritmos buscan conjuntos de puntos en el espacio, donde se agrupan puntos que están en regiones de alta densidad, esto es, donde los puntos sean muy cercanos y señalan como puntos atípicos, aquellos que se encuentran solos en regiones de baja densidad (cuyos vecinos más cercanos están demasiado lejos) y cuando la densidad en la vecindad supere algún umbral; es decir, para cada punto de datos dentro de un grupo dado; la vecindad de un radio dado debe contener al menos un número mínimo de puntos.
Los principales algoritmos representativos son *DBSCAN*, *OPTICS, DBCLASD, DENCLUE y SNN*.

2.2.2.2. Algoritmos jerárquicos

Los algoritmos jerárquicos tienen por objetivo agrupar clusters para formar un nuevo o bien separar alguno ya existente para dar origen a otros dos, de tal forma que, si sucesivamente se va efectuando este proceso de aglomeración o división, se minimice alguna distancia o bien se maximice alguna medida de similitud. Esta descomposición jerárquica puede representarse mediante un diagrama de árbol llamado dendrograma; cuyo nodo raíz representa el conjunto de datos completo y cada nodo hoja es un único objeto del conjunto de datos. Los resultados del agrupamiento pueden obtenerse cortando el dendrograma en diferentes niveles.

Hay dos enfoques generales para los algoritmos jerárquicos: aglomerativo (de abajo hacia arriba) y divisivo (de arriba hacia abajo)

A. Algoritmos aglomerativos

Los algoritmos aglomerativos, también conocidos como ascendentes, comienzan el análisis con tantos grupos como individuos haya. A partir de estas unidades iniciales se van formando grupos, de forma ascendente, hasta que al final del proceso todos los casos tratados estén englobados en un mismo conglomerado. La fusión entre conglomerados se basa en distancias entre dos clústeres. Hay diferentes nociones de distancia: enlace único, enlace promedio, enlace completo, entre otros.

B. Algoritmos divisivos

Los algoritmos divisivos o disociativos, también llamados descendentes, constituyen el proceso inverso al anterior. Comienzan con un conglomerado que engloba a todos los casos tratados y, a partir de este grupo inicial, a través de sucesivas divisiones, se van formando grupos cada vez más pequeños. Al final del proceso se tienen tantas agrupaciones como casos hayan sido tratados. Para un conjunto de datos que tiene n observaciones hay $2^{n-1} - 1$ divisiones posibles, lo cual es muy costoso computacionalmente. Dos algoritmos divisivos de agrupamiento: *DIANA y MONA*.

2.2.3. Medidas de distancia

Uno de los aspectos clave del análisis clustering es la elección de la medida que se desea utilizar para cuantificar la distancia entre los elementos. Existe una gran cantidad de medidas de distancia que se diferencian por el tipo de datos para el que han sido diseñadas: cuantitativos, categóricos, dicotómicos, entre otros. Estas medidas también se diferencian por el tipo de distancia evaluada: similaridad o disimilaridad.

Las medidas de similaridad evalúan el grado de parecido o proximidad existente entre dos elementos. Los valores más altos indican mayor parecido o proximidad entre los elementos comparados: cuando dos elementos se encuentran juntos, el valor de las medidas es máximo.

Las medidas de disimilaridad ponen el énfasis sobre el grado de diferencia o lejanía existente entre dos elementos. Los valores más altos indican mayor diferencia o lejanía entre los elementos comparados: cuando dos elementos se encuentran juntos, la distancia es nula.

En la figura 3, se presenta un resumen de algunas de las distancias que se pueden utilizar en los algoritmos clustering:

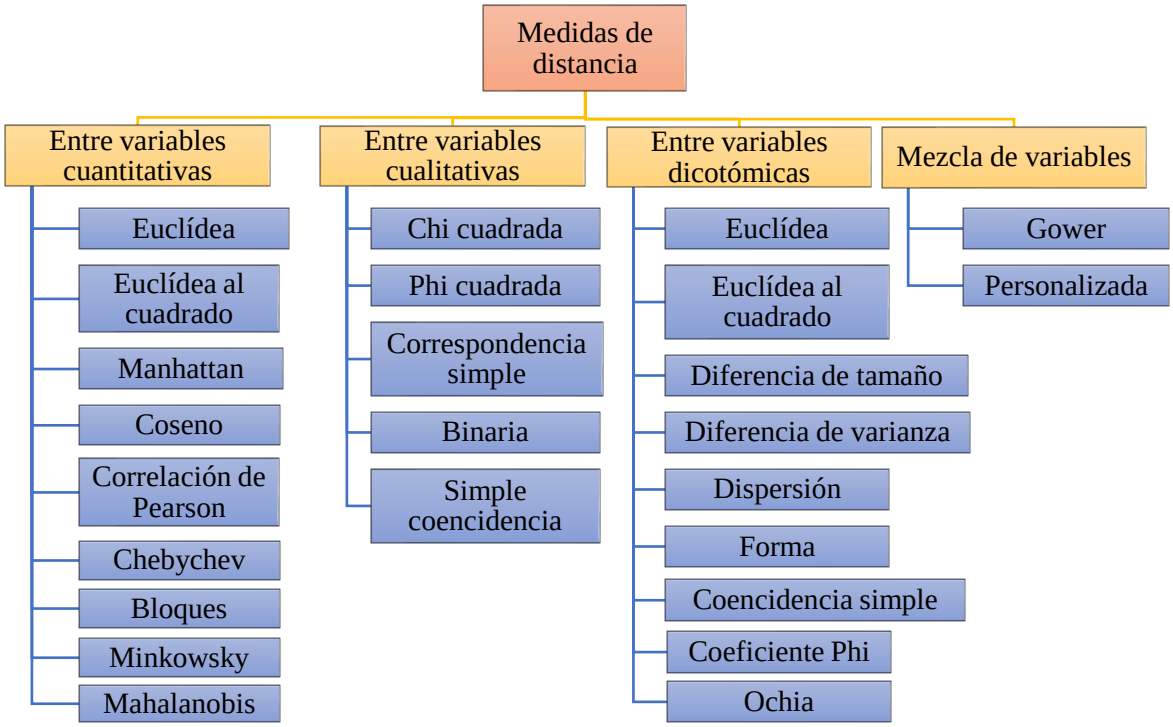


Figura 3: Medidas de distancias clustering

2.2.4. Algoritmo K-means

2.2.4.1. Descripción del algoritmo K-means

MacQueen (1967), señaló que el algoritmo K-means clustering es un método no supervisado, de agrupamiento particional o no jerárquico más utilizado para segmentar un determinado conjunto de datos en *k* grupos, donde *k* representa el número de grupos especificados previamente por el analista. En la agrupación de K-means, cada cluster está representado por su centro o centroide que corresponde a la media de los puntos asignados al cluster.

La idea básica detrás de la agrupación de K-means consiste en definir los grupos de manera que se minimice la variación intra-grupos (conocido como variación total dentro del grupo). El algoritmo estándar es el algoritmo de Hartigan et al. (1979), que define la variación intra-grupos como la suma de distancias al cuadrado entre los elementos y el correspondiente centroide:

$$W(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2 \qquad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Donde:

• x_i observación perteneciente al clúster C_k

• μ_k es el centroide asignados al clúster C_k

Cada observación (x_i) se asigna recursivamente a un grupo determinado de tal manera que la suma de los cuadrados (SS) de la distancia de observación a sus centros de conglomerados asignados μ_k es mínima. Finalmente, se define la variación intra-grupo del clúster de la siguiente manera:

$$Var.intra = \sum_{k=1}^{K} W(C_k) = \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2 \qquad \ldots\ldots\ldots\ldots\ldots (2)$$

La suma total de la variación intra-grupo de cuadrados mide la concentración (es decir, la bondad) de la agrupación y se desea que sea lo más pequeña posible.

El algoritmo K-means tiene las siguientes propiedades importantes:

1. Es eficiente en el procesamiento de grandes conjuntos de datos.
2. Con frecuencia finaliza en un óptimo local (centroides).
3. Funciona solo en valores numéricos.
4. Los grupos tienen formas convexas.

El primer paso al usar la agrupación de K-means es indicar la cantidad de agrupaciones *k* que se generarán en la solución final. El algoritmo comienza a seleccionar aleatoriamente *k* objetos del conjunto de datos para servir como los centros iniciales para los grupos. Los objetos seleccionados también se conocen como medios de agrupamiento o centroides.

A continuación, cada uno de los objetos restantes se asigna a su centroide más cercano, donde el más cercano se define utilizando medidas de distancia entre el objeto y el centroide del grupo. Este paso se denomina "etapa de asignación de clúster".

Después de la etapa de asignación, el algoritmo calcula el nuevo valor medio de cada grupo, el término "actualización de centroide" se usa para denominar esta etapa. Ahora que los centroides han sido recalculados, cada observación se verifica nuevamente para ver si podría estar más cerca de un grupo diferente, todos los objetos se reasignan de nuevo utilizando las medias de clúster actualizadas. La asignación de clúster y los pasos de actualización de centroides se repiten iterativamente hasta que las asignaciones de clúster dejan de cambiar (hasta lograr la convergencia). Es decir, cuando los conglomerados formados en la iteración actual son los mismos que los obtenidos en la anterior iteración.

En resumen, el algoritmo se detalla en los siguientes pasos:

- Paso 1: Seleccionar *k* elementos aleatorios como centroides iniciales de un conjunto de datos, uno para cada grupo.
- Paso 2: Asignar cada elemento a un cluster cuyo centroide sea el más cercano a él. Luego actualizar el centroide en cada clúster después de cada asignación.

- Paso 3: Después de que todos los elementos hayan sido asignados a un clúster, volver a probar la disimilitud de los objetos con los centroides actuales. Si se encuentra un elemento tal que su centro más cercano pertenece a otro cluster en lugar de su actual, reasignar el elemento al otro cluster y actualizar los centroides de ambos clústeres.
- Paso 4: Repetir el paso 3 hasta que ningún elemento haya cambiado de clúster.

Con el fin de tener un mejor entendimiento del paso a paso del algoritmo, se tiene un ejemplo de aplicación de este, en el anexo 1.

2.2.5. Algoritmo K-modes

2.2.5.1. Descripción del algoritmo K-modes

Según Huang (1998), el algoritmo de agrupamiento de K-modes es una extensión del algoritmo de agrupación de K-means. Sin embargo, el proceso de agrupamiento de K-means estándar no se puede aplicar a datos categóricos debido a la función de distancias entre observaciones y el uso de centroides para representar los centros de conglomerados. Para usar K-means en la agrupación de datos categóricos, se tendría que convertir cada categoría única a un atributo binario ficticio y usar 0 o 1 para indicar el valor categórico ausente o presente en un registro de datos. Este enfoque no es adecuado para datos categóricos de alta dimensión.

El enfoque K-modes modifica el proceso K-means estándar para agrupar datos categóricos reemplazando la función de distancia con una medida de disimilitud para datos categóricos, utilizando modas para representar los centros de los clusters y las modas se actualizan con los valores categóricos más frecuentes en cada iteración del proceso de agrupamiento. Estas modificaciones garantizan que el proceso de agrupación converja a un resultado mínimo local y se mantenga la eficiencia del proceso de agrupamiento.

Medida de disimilitud para datos categóricos

Sea X e Y dos elementos categóricos descritos por *m* atributos. La medida de disimilitud entre ambos elementos puede definirse como el total de falta de correspondencia (desajuste) entre las categorías de atributos correspondientes entre los dos elementos. Cuanto menor sea el número de desajustes, más similares serán los dos objetos. Esta medida a menudo se denomina coincidencia simple, y se presenta en la siguiente ecuación:

$$d(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \qquad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$\text{Donde: } \delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

Para agrupar un conjunto de datos categóricos X en *k* grupos, el proceso de agrupación de este algoritmo consiste en los siguientes pasos:

El primer paso es seleccionar aleatoriamente *k* objetos únicos como los centros de cluster iniciales. Luego, se calcula las distancias entre cada objeto y los centroides; asignando el objeto al grupo cuyo centro tiene la distancia más corta al objeto, a este paso se denomina "etapa de asignación de clúster"; se repite este procedimiento hasta que todos los objetos estén asignados a los clústeres.

La asignación de clúster y la “actualización de modes o centros” se repiten iterativamente hasta que las asignaciones de clúster dejan de cambiar (hasta lograr la convergencia). Es decir, cuando los conglomerados formados en la iteración actual son los mismos que los obtenidos en la anterior iteración.

Como es esencialmente igual que K-means, el algoritmo de agrupamiento de K-modes, posee las mismas propiedades, es decir, es eficiente para agrupar grandes datos categóricos y también produce resultados de clustering localmente óptimo, que dependen de los modes iniciales.

En resumen, el algoritmo se detalla en los siguientes pasos:

- Paso 1: Seleccionar *k* elementos aleatorios como modes iniciales de un conjunto de datos, uno para cada grupo.
- Paso 2: Asignar cada elemento a un cluster cuyo centro sea el más cercano a él. Luego actualizar el mode en cada clúster después de cada asignación.
- Paso 3: Después de que todos los elementos hayan sido asignados a un clúster, volver a probar la disimilitud de los objetos con los centros actuales. Si se encuentra un elemento tal que su centro más cercano pertenece a otro cluster en lugar de su actual, reasignar el elemento al otro cluster y actualizar los modes de ambos clústeres.
- Paso 4: Repetir el paso 3 hasta que ningún elemento haya cambiado de clúster.

Con el fin de tener un mejor entendimiento del paso a paso del algoritmo, se tiene un ejemplo de aplicación de este, en el anexo 1.

2.2.6. Algoritmo K-prototypes

2.2.6.1. Descripción del algoritmo K-prototype

Según Huang (1998), los algoritmos K-means y K-modes se integran en el algoritmo K-prototype que se utiliza para agrupar elementos de tipo mixto. El algoritmo K-prototype es más útil porque los elementos encontrados con frecuencia en las bases de datos del mundo real son de tipo mixto. La disimilitud entre dos elementos de tipo mixto X e Y se puede medir con la siguiente ecuación:

$$d(X,Y) = \sum_{j=1}^{m_r}(x_j - y_j)^2 + \gamma \sum_{j=p+1}^{m_c} \delta(x_j, y_j) \quad \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Donde m_r y m_c son las cantidades de atributos numéricos y categóricos respectivamente, el primer término es la medida de distancia en los atributos numéricos y el segundo término es la medida de disimilitud en los atributos categóricos, donde el peso γ se usa para evitar

favorecer cualquier tipo de atributo. La influencia de γ en el proceso de agrupación se discutirá más adelante.

El algoritmo K-prototype se describe en los siguientes pasos:

- Paso 1: Seleccionar *k* prototypes iniciales de un conjunto de datos X, uno para cada grupo.
- Paso 2: Asignar cada elemento en X a un cluster cuyo prototype sea el más cercano a él de acuerdo con la ecuación. Luego actualizar el prototype en cada clúster después de cada asignación.
- Paso 3: Después de que todos los elementos hayan sido asignados a un clúster, volver a probar la disimilitud de los objetos con los prototypes actuales. Si se encuentra un elemento tal que su prototype más cercano pertenece a otro cluster en lugar de su actual, reasignar el elemento al otro cluster y actualizar los prototypes de ambos clústeres.
- Paso 4: Repetir el paso 3 hasta que ningún elemento haya cambiado de clúster.

Con el fin de tener un mejor entendimiento del paso a paso del algoritmo, se tiene un ejemplo de aplicación de este, en el anexo 1.

2.2.7. Algoritmo K-medoids (PAM)

2.2.7.1. Algoritmo K-Medoids

Según Bhat (2014), una de las técnicas de agrupamiento más populares es el algoritmo de agrupación K-means, la cual utiliza la media o centroide para representar el grupo, este enfoque, aunque fácil de entender e implementar tiene un inconveniente importante. En el caso de elementos con valores extremos, la distribución de datos es desigual, lo que puede terminar en un agrupamiento inadecuado. Esto hace que el algoritmo de agrupamiento K-means sea muy sensible a los valores atípicos y al ruido, reduciendo así su rendimiento.

Frente ello, se desarrolló otro enfoque para la agrupación, que se basa en similaridad, la partición alrededor de los medoids (algoritmo PAM), es conocida por ser más robusta a los outliers y al ruido que ocurren en un ambiente real sin control. En lugar de utilizar la media convencional o centroide, se utiliza medoids para representar los clusters. El medoid es una estadística que representa ese elemento de un conjunto de datos cuya disimilitud media a todos los demás elementos del conjunto es mínima. Por lo tanto, un medoid a diferencia de la media siempre es un elemento del conjunto de datos y es el más centralizado del conjunto de datos.

El trabajo del algoritmo K-medoids clustering es similar a K-means, también comienza con la selección aleatoria de *k* elementos de datos como centroides iniciales para representar los *k* clusters, los elementos restantes se incluyen en el grupo que tiene el medoid más cercano a ellos y posteriormente se determina un nuevo centro que puede representar mejor al grupo. En cada iteración, todos los elementos distintos a los centros, se asignan nuevamente a los clusters que tienen el medoid más cercano, provocando que los medoids alteren su ubicación. El algoritmo minimiza la suma de las distancias entre cada elemento de datos y su correspondiente medoid, este ciclo se repite hasta que ningún medoid cambie su colocación, esto marca el final del proceso y se tienen los clusters finales. Se forman *k* grupos que se centran alrededor de los medoids y todos los miembros se colocan en el grupo apropiado basado en el medoid más cercano.

La diferencia entre K-means y K-medoids es análoga a la diferencia entre la media y la mediana: donde la media indica el valor promedio de todos los datos recopilados, mientras que la mediana indica el valor central cuando todos los datos están distribuidos uniformemente alrededor de ella. La idea básica de este algoritmo es calcular primero los *k* objetos representativos que son llamados medoids. Después de encontrar el conjunto de centros, cada objeto del conjunto de datos se asigna al más cercano de ellos.

Según Arora et al. (2016), el algoritmo K-medoids se basa en técnicas de representatividad para reducir los inconvenientes del algoritmo K-means. Los *k* centros o medoids se seleccionan al azar de los datos y se forman los clusters al agrupar las *n* - *k* observaciones restantes a los medoids más cercanos, en cada cluster se procesan todos los datos para encontrar nuevos medoids de manera recursiva, de tal forma que se represente mejor a un

nuevo cluster, finalmente después de encontrar los nuevos medoids se enlazan todos los objetos de datos al clúster. La ubicación de cada centro puede cambiar en cada una de las $\frac{n!}{k!(n-k)!}$ iteraciones. Así se encuentran los *k* clústers que representan *n* objetos de datos; por defecto cuando los medoids no se especifican, el algoritmo busca el mejor conjunto de medoids con el objetivo de minimizar la función objetivo; el algoritmo ha sido diseñado para no depender del orden de las observaciones.

El algoritmo se basa en dos pasos:

- CONSTRUCCIÓN: Este paso selecciona secuencialmente *k* medoids "situados centralmente", para ser utilizados como iniciales.
- INTERCAMBIO: Si se puede reducir la función objetivo cambiando un objeto seleccionado por uno no seleccionado, se realiza el intercambio. Esto continúa hasta que la función objetivo ya no pueda ser disminuida.

El algoritmo se detalla en los siguientes pasos:

- Paso 1: Seleccionar *k* elementos aleatorios como los medoids de los n elementos de datos.
- Paso 2. Asociar cada elemento de datos al centro más cercano utilizando cualquiera de las métricas de distancia más comunes.
- Paso 3. Calcular el *costo total de cambio* TC_{ki}, el cual es la sumatoria de las medidas de distancias de las observaciones a su medoid C_i en cada cluster, se muestran en la siguiente ecuación: $TC_{ki} = costo(x, c) = \sum_{i=1}^{k} d^2_{(x_k, c_i)}$, donde x es cualquier objeto, c es el medoid, y d es la distancias del objeto al centro del cluster al que pertenece.
- Paso 4: Seleccionar de manera iterativa una observación no medoid, que sería el nuevo centro C_j, volver a probar la disimilitud de los objetos con el medoid nuevo y calcular el costo total TC_{kj}; si el costo total aumentó deshacer el intercambio y repetir este paso hasta probar todos los posibles medoids, buscando minimizar la función objetivo: $\min(TC_k)$.

Hay cuatro situaciones a considerar en este proceso:

- Afiliación de desplazamiento hacia fuera: un elemento x_i puede necesitar ser cambiado de cluster para ser medoid en el otro conglomerado.
- Actualización del medoid: un nuevo centro se encuentra para reemplazar el actual centro.
- Sin cambios: los elementos resultantes del clúster actual tienen la misma o más pequeña medida de distancia para todas las posibles redistribuciones consideradas.
- Cambio de cluster: un elemento exterior x_i se asigna al clúster actual con el nuevo (reemplazado) medoid.

Con el fin de tener un mejor entendimiento del paso a paso del algoritmo, se tiene un ejemplo de aplicación de este, en el anexo 1.

2.2.8. Distancia a utilizar

Sea $X = \{X_1, X_2, \dots, X_n\}$ un conjunto de *n* elementos, $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ un elemento *i* representado por los valores de *m* atributos y *k* un número entero positivo. El objetivo de agrupar X es encontrar una partición que divida los elementos de X en *k* agrupamientos disjuntos. Para un n dado, el número de particiones posibles es extremadamente grande.
No es práctico investigar cada partición, para determinar la mejor, dado un problema de agrupamiento. Una solución común es elegir un criterio que permita la agrupación de los objetos para guiar la búsqueda de una partición. Un criterio para determinar la disimilitud entre dos objetos de tipo mixto X e Y se presenta en la siguiente ecuación:

$$d(X_i, Y_l) = \sum_{j=1}^{m_r} ({x_{ij}}^r - {y_{lj}}^r)^2 + \gamma \sum_{j=p+1}^{m_c} \delta({x_{ij}}^c, {y_{lj}}^c) \quad \dots\dots\dots\dots\dots\dots \quad (5)$$

donde el primer término es la medida de distancia euclidiana al cuadrado en los atributos numéricos y el segundo término es la medida de disimilitud de coincidencia simple en los atributos categóricos.

Donde $\delta(p,q) = 0$ para $p = q$ y $\delta(p,q) = 1$ para $p \neq q$, ${x_{ij}}^r$ y ${q_{lj}}^r$ son valores de atributos numéricos, mientras que ${x_{ij}}^c$ y ${q_{lj}}^c$ son valores de atributos categóricos para el elemento *i* y el centro del cluster *l*; m_r y m_c son las cantidades de atributos numéricos y categóricos respectivamente, γ es un peso para atributos categóricos, introducido para evitar favorecer cualquier tipo de atributo. La influencia del peso γ en la agrupación se ilustra en la figura 4, suponga que los triángulos y los diamantes son un conjunto de elementos descritos por un atributo categórico y dos atributos numéricos, las formas triángulo y diamante representan dos valores del atributo categórico, mientras que los valores de los atributos numéricos se reflejan en las ubicaciones de los objetos, estos objetos están divididos en dos clusters.

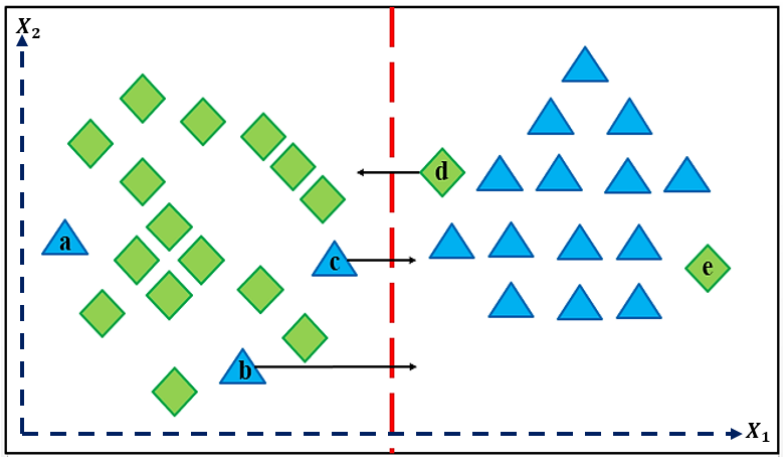


Figura 4: Resultado de la influencia de γ en el proceso de clustering

Si γ = 0, la agrupación solo depende de los atributos numéricos. El resultado será los dos grupos separados por la línea punteada vertical. Si γ > 0, entonces el objeto "c" puede cambiar al clúster de la derecha porque está cerca de ese clúster y su valor categórico es el mismo que el de la mayoría de los objetos en ese clúster.

De manera similar, el objeto "d" puede cambiar al clúster al de la izquierda, sin embargo, el objeto "a" aún puede permanecer en el clúster de la izquierda porque está demasiado alejado del cluster de la derecha, aunque tiene un valor categórico igual al de la mayoría de los objetos de ese clúster. De forma similar, el objeto "e" aún puede estar en el clúster de la derecha. El objeto "b" se vuelve incierto, dependiendo de si γ está sesgado hacia atributos numéricos o categóricos.

Si γ está sesgado al atributo categórico, el objeto b puede cambiar al clúster derecho, de lo contrario, puede permanecer en la izquierda. La elección de γ depende de la distribución de atributos numéricos, en términos generales, γ se relaciona con σ, la desviación estándar promedio de los atributos numéricos. En la práctica, σ puede usarse como una guía para determinar γ. Según, Huang (1998), un γ adecuado se encuentra entre 1/3σ y 2/3σ para los conjuntos de datos, el cálculo estimado de γ es de la siguiente manera:

$$\gamma = \frac{Promedio\ (Varianza\ o\ desviación\ estándar\ de\ las\ variables\ numéricas)}{Promedio(Heurística\ para\ variables\ categóricas)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (6)$$

donde la heurística para variables categóricas se calcula como: $1 - \sum_i p_i^2$ o $1 - max_i\ p_i$; p_i es la proporción de la categoría i en la variable cualitativa.

2.2.9. Validación de clúster

Según Wang et al. (2007), el análisis clústering tiene como objetivo identificar grupos de objetos similares, por lo tanto, ayuda a descubrir la distribución de patrones y correlaciones interesantes en grandes conjuntos de datos. Sin embargo, el análisis clústering es un método no supervisado y en la mayoría de los casos, el usuario no tendrá ningún conocimiento previo sobre el número de grupos en el que se está separando el conjunto de datos, ni el algoritmo más adecuado para estos, para llegar a ello, es necesario dividir conglomerados dispersos en dos o más clusters compactos, por lo contrario, si son pequeños, pueden combinarse más de un clúster separado.

La solución para encontrar el mejor algoritmo clustering y el número óptimo de conglomerados *k* se llama generalmente validez del cluster. Una vez que la partición se obtiene mediante un método de agrupación, la función de validez *cuantifica la precisión* de la estructura del conjunto de datos; para datos bidimensionales, los usuarios pueden verificar visualmente la validez de los resultados. Sin embargo, en el caso de grandes conjuntos de datos multidimensionales, la visualización efectiva del conjunto de datos sería difícil. Por lo tanto, el objetivo de la validez del clúster es encontrar *k* conglomerados óptimos con el mejor

algoritmo clustering buscando alcanzar una mejor precisión en la descripción de la estructura de datos multidimensionales.

La determinación del valor del parámetro (el número de clusters) es importante cuando se aplica algoritmos clustering, ya que la elección inadecuada de este genera particiones que no reflejan el agrupamiento deseado de los datos. Por ejemplo, la figura 5 (a) presenta un conjunto de datos, se observa que este tiene tres grupos desde el ángulo visual. Sin embargo, si se considera un algoritmo de agrupamiento con el valor de parámetros k=4 para dividir el conjunto de datos, el resultado del proceso de agrupación sería el esquema de agrupación presentado en figura 5 (b). En el ejemplo, el algoritmo de agrupación encontró los cuatro mejores clusters en los que nuestro conjunto de datos podría ser particionado. Sin embargo, este no es el particionamiento óptimo para el conjunto de datos considerado.

La partición obtenida por el algoritmo de agrupamiento en la figura 5 (b) representa incorrectamente la estructura del conjunto de datos, es decir, no encaja bien en el conjunto de datos. El agrupamiento óptimo para el conjunto de datos será un esquema con tres clústeres, como consecuencia, si se asigna un valor incorrecto a los parámetros del algoritmo de agrupación, el método obtendrá como resultado un esquema de partición que no es óptimo para el conjunto de datos específico lo que conduciría a tomar decisiones incorrectas.

El problema de decidir el algoritmo y el número de agrupaciones que encajan mejor en un conjunto de datos, así como en la evaluación de los resultados del agrupamiento, han sido objeto de varias investigaciones; idealmente, los clústeres resultantes deben tener buenas propiedades estadísticas (compactas, bien separadas, conectadas y estables).

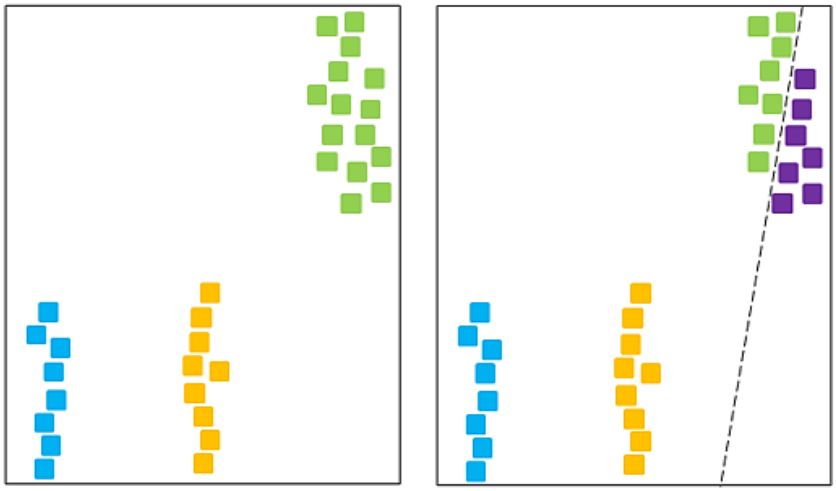


Fig. 5(a). Un conjunto de datos que consta de tres grupos

Fig. 5(b). Los resultados de la aplicación de algoritmos clustering cuando se piden cuatro grupos.

Figura 5: Comparación del número de clusters

Actualmente existe una gran cantidad de algoritmos de agrupamiento, por ello, decidir qué método de agrupación y determinar el número de clusters que es más apropiados para los datos analizados, puede ser una tarea compleja para el investigador que conduce el estudio.

2.2.10. Medidas de validación internas

Para comparar y verificar que los algoritmos de clustering agrupen objetos similares en un mismo clúster y objetos disímiles en diferentes clústeres se utilizan indicadores de validación interna, medidas que reflejan la cohesión y separación de las particiones de clúster.

La *cohesión* cuantifica el grado de proximidad del miembro de cada clúster a los otros miembros del mismo clúster, con ello se evalúa la homogeneidad dentro del clúster o varianza intragrupo; mientras que la *separación* entre clusters cuantifica el grado de heterogeneidad entre conglomerados o varianza intergrupal (usualmente midiendo la distancia entre los centroides). Ambas métricas permiten validar los resultados que se obtienen de una segmentación.

a) Índice de validación de Davies Bouldin

Según Chun (2012), el índice de Davies Bouldin es una métrica para evaluar el buen funcionamiento de los algoritmos de clustering. La fórmula de este índice se muestra en la siguiente ecuación:

$$DB = \frac{1}{k}\sum_{i=1}^{k} máx_{i\neq j}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right) \quad \text{..... (7)}$$

donde n es el número de clústeres, c_x denota el centro del clúster x, σ_x es la distancia media de todos los elementos del clúster x al centro c_x, y $d(c_i, c_j)$ es la distancia entre los centroides c_i y c_j , la idea de estas distancias se ve en la figura 6. El objetivo de los algoritmos de clustering es producir agrupamientos con baja distancia dentro del mismo clúster, y altas distancias entre los clústeres. Por lo tanto, el máximo valor de este índice $máx_{i\neq j}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right)$ representa el peor caso para el cluster i. La solución óptima es aquella que tiene el índice de Davies Bouldin más bajo.

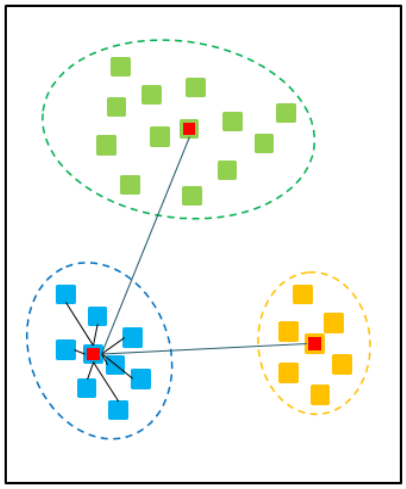


Figura 6: Distancias utilizadas para el cálculo del índice de Davies-Bouldin

b) Índice de Dunn

Chun (2012), explicó que el índice de Dunn es una métrica para evaluar el buen funcionamiento de los algoritmos de clustering. El objetivo de este índice es identificar un conjunto de clústeres que sean compactos, con una varianza pequeña entre los miembros del clúster, y que éstos estén bien separados de los miembros de otros clústeres. Un valor más alto del índice de Dunn indica un mejor rendimiento del algoritmo de clustering.

El índice de Dunn tiene un valor entre cero e infinito, por lo tanto, la distancia entre los miembros de un clúster debe ser los más baja posible y la distancia entre los clústeres lo más alta posible, como se tiene en la figura **7**.

$$ID = \frac{min\left(\underset{1\le i\le j\le k}{min} \; d(C_i, C_j)\right)}{max\left(\underset{1\le i\le k}{max} \; \Delta_k\right)} \qquad \text{..... (8)}$$

Donde:

k es el número de clusters,

$d(C_i, C_j)$ es la distancia entre centros de clústeres

$\Delta_i = \underset{x, y \in C_i}{max} \; d(x, y)$, sea x e y observaciones asignadas al mismo cluster C_i.

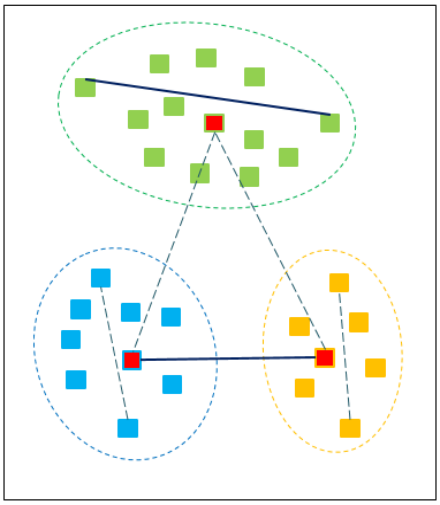


Figura 7: Distancias utilizadas para el cálculo del índice de Dunn

c) Índice de Calinski Harabasz

Calinski et al. (1974), señalan que este índice se basa en la idea del ANOVA, está definido como la razón entre la dispersión interior de los clusters y la dispersión entre los clusters. El objetivo es *maximizar* el valor de la función CH defina de la siguiente forma:

$$CH = \frac{SSB/(k-1)}{SSW/(N-k)} \qquad \text{..... (9)}$$

Donde:
k es el número de clusters
N es el número total de observaciones
SSW es la varianza intracluster (Sum of Squared Within): medida interna especialmente usada para evaluar la *cohesión* de los clústeres que el algoritmo de agrupamiento generó, como se tiene en la figura 8.

$$SSW = \sum_{i=1}^{k} \sum_{x \epsilon C_i} d^2(m_i, x) \qquad \text{..... (10)}$$

Siendo k el número de clústeres, x un punto del clúster C_i y m_i el centro del clúster Ci
SSB es la varianza interclúster (Sum of Squared Between): medida de separación utilizada para evaluar la distancia entre clusters (Separación), como se tiene en la figura 9.

$$SSB = \sum_{j=1}^{k} n_j d^2(c_j, \bar{x}) \qquad \text{..... (11)}$$

Siendo k el número de clústeres, n_j el número de elementos en el clúster j, c_j el centroide del clúster j y $\bar{x}$ es el centro de la data set.

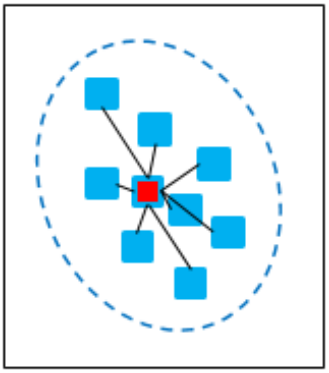


Figura 8: Distancias utilizadas para el cálculo del SSW

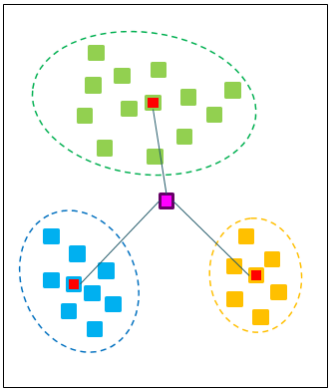


Figura 9: Distancias utilizadas para el cálculo del índice de SSB

2.2.11. Árboles de decisión

Según Maimon et al. (2010), los árboles de decisión son un conjunto de técnicas supervisadas no paramétricas, se organizan en una estructura en forma de árbol, cuyo objetivo principal es el aprendizaje inductivo a partir de observaciones y reglas lógicas, son expresados como una partición recursiva del espacio de entrada en función de los valores de los atributos.

El árbol de decisión consta de nodos que forman una estructura de árbol, lo que significa que es un árbol dirigido con un nodo llamado "raíz" que no tiene ramas entrantes, todos los demás nodos tienen exactamente una rama entrante. Un nodo con ramas salientes se denomina nodo interno o de prueba, todos los demás nodos se denominan hojas (también conocidos como nodos terminales o de decisión), los cuales corresponden a una decisión, la cual coincide con una de las variables objetivo del problema a resolver. En un árbol de decisión, cada nodo

interno divide los valores de un atributo en dos o más subespacios no solapantes, en el caso de los atributos numéricos, la condición se refiere a un rango y en atributos cualitativos se refiere a una clase.

Cada hoja se asigna a una clase que representa el valor corresponden a una decisión, la cual coincide con una de las variables clase del problema a resolver más apropiado. Alternativamente, la hoja puede contener un vector de probabilidad que indica la probabilidad de que el atributo objetivo tenga un cierto valor. Las observaciones se clasifican al seguir las reglas desde la raíz del árbol hasta una hoja, de acuerdo con el resultado de las pruebas a lo largo del camino.

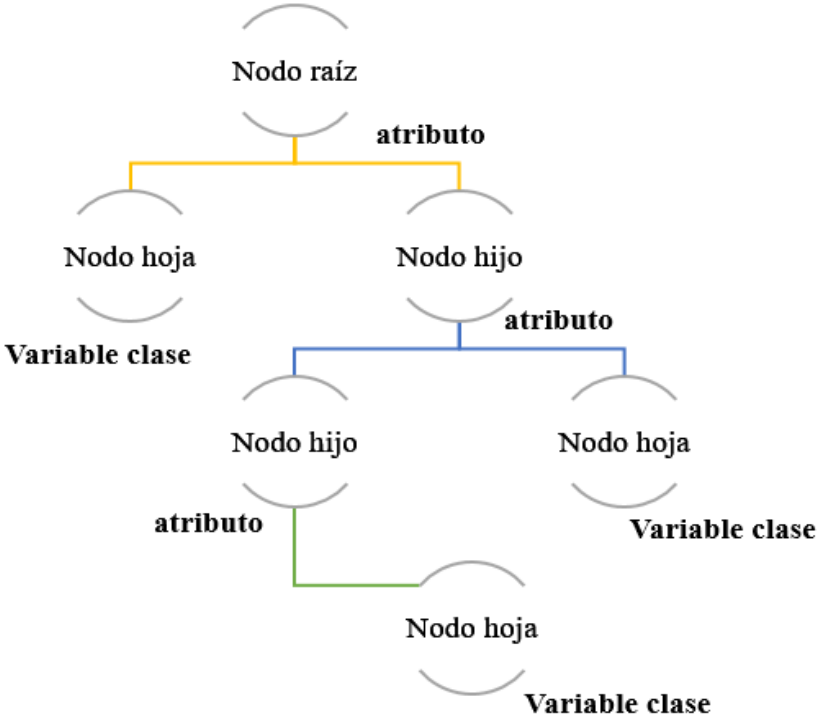


Figura 10: Estructura de un árbol de decisión

2.2.11.1. Tipos de árboles de decisión

Los árboles de decisión utilizados principalmente son de dos tipos:

- Árbol de clasificación son los árboles de decisión donde la variable objetivo que se busca predecir es categórica, a partir de una o más variables productoras continuas y/o categóricas.

- Árbol de regresión son los árboles de decisión donde la variable objetivo que se busca predecir es cuantitativa, a partir de una o más variables productoras continuas y/o categóricas.

Según Amat (2017), a pesar de la sencillez con la que se puede resumir el proceso de construcción de un árbol, es necesario establecer una metodología que permita crear los subespacios, o lo que es equivalente, decidir donde se introducen las divisiones: en qué predictores y en qué valores de estos. Es en este punto donde se diferencian los algoritmos de árboles de regresión y clasificación.

En el caso de los árboles de regresión, el criterio más frecuentemente empleado para identificar las divisiones es el Residual Sum of Squares (RSS). El objetivo es encontrar los J subespacios $(R_1, \dots, R_j)$ que minimizan el Residual Sum of Squares (RSS) total:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - y_{R_j})^2 \qquad \dots\dots\dots\dots\dots\dots (12)$$

donde y_{R_j}es la media de la variable respuesta en la región R_j. No es posible considerar todas las posibles particiones del espacio de los predictores. Por esta razón, se recurre a lo que se conoce como *recursive binary splitting* (división binaria recursiva), este permite encontrar el punto de corte s para cada varible, tal que, si se distribuyen las observaciones en las regiones $\{X|X_j < s\}$ y $\{X|X_j \geq s\}$, se consigue la mayor reducción posible en el RSS.

En caso de los árboles de clasificación, el criterio más frecuentemente empleado para encontrar nodos lo más puros/homogéneos posible, es la *entropía*, una forma de medir la impureza asociada con una variable aleatoria, es decir, qué tan mezclados están los valores de la clase, la entropía aumenta con el aumento de la incertidumbre o aleatoriedad y disminuye con una disminución de la incertidumbre o aleatoriedad.

Dado que el conjunto de datos consta de n observaciones. El atributo de la variable objetivo tiene c clases diferentes (i = 1, 2, ... c). Cada observación pertenece a una de las c clases y

p_i se refiere a la proporción de observaciones de la variable respuesta que caen en el nivel i de clase. La entropía de información se define de la siguiente manera:

$$\boldsymbol{E_c} = -\sum_{i=1}^{c} \hat{p}_i log_2^{\hat{p}_i} \qquad \text{..... (13)}$$

El atributo que reduce la impureza al nivel máximo (o tiene el índice *mínimo* de Entropía) se selecciona como el atributo de división, si el caso es absolutamente homogéneo y contiene un tipo similar de conjuntos de datos, la entropía es cero. El proceso para realizar las divisiones consiste en:

1. Para cada posible división, se calcula el valor de la medida E_c en cada uno de los nodos hijos resultantes.
2. Se suman los valores ponderados por la fracción de observaciones que contiene cada nodo hijo.

$$SplitInfo\ (S) = \sum_{i=1}^{m} \frac{n\ observaciones\ nodo\ hijo_i}{n\ observaciones\ nodo\ padre} \times \boldsymbol{E_{c_i}}$$

..... (14)

3. La división con menor o mayor valor (dependiendo de la medida empleada) se selecciona como división óptima.

Si se utiliza como medida de pureza la entropía, para calcular la cantidad de información que uno gana al elegir un atributo en particular se cuantifica como:

- Ganancia de información (se elige el atributo que tiene la mayor ganancia de información)

Indicador que permite determinar el atributo a seleccionar para continuar el proceso de división, se calcula como:

$$InfoGain(S_2) = SplitInfo(S_1) - SplitInfo\ (S_2) \qquad \text{..... (15)}$$

- Proporciones de ganancia (se elige el atributo que tiene la mayor proporción de ganancia)

Otra manera de determinar el atributo a seleccionar es la proporción de ganancia, la cual se calcula como:

$$GainRatio(S_2) = InfoGain(S_2)/Entropy(S_2) \quad \ldots\ldots\ldots\ldots\ldots\ldots (16)$$

Donde S_1 representa al nodo padre y S_2 al nodo hijo o variable que se quiere seleccionar

Según Gupta et al. (2017) y R Patel et al. (2014) existen muchos algoritmos para construir un árbol de decisión, estos dependen del tipo de variable que usan como dependientes e independientes, los más utilizados son:

i. ID3 (Dicotomizador iterativo)

ID3 es un algoritmo desarrollado por Ross Quinlan en 1986 que se utiliza para generar un árbol de decisión a partir de un conjunto de datos. Para construir un árbol de decisión, ID3 se utiliza una búsqueda exhaustiva de arriba hacia abajo a través de los conjuntos dados, donde cada atributo en cada nodo del árbol se prueba para seleccionar el atributo que es mejor para la clasificación de un conjunto dado. Por lo tanto, el atributo con la mayor ganancia de información se puede seleccionar como el atributo de prueba del nodo actual.

Para construir un modelo de árbol de decisión, ID3 solo acepta atributos categóricos. ID3 no proporciona resultados precisos cuando hay ruido y cuando se implementa en serie. Por lo tanto, los datos se procesan previamente antes de construir un árbol de decisión. Para construir un árbol de decisión, la ganancia de información se calcula para cada atributo y el atributo con la mayor ganancia de información se convierte en el nodo raíz. El resto de los valores posibles se denotan por ramas. El crecimiento se detiene cuando todas las observaciones pertenecen a un solo valor de la característica objetivo (todas las observaciones estén bien clasificadas) o cuando la mejor ganancia de información no es

mayor que cero. ID3 no aplica ningún procedimiento de poda ni maneja atributos numéricos o valores faltantes.

Ventajas de ID3

- Los datos de entrenamiento se utilizan para crear reglas de predicción comprensibles.
- Construye el árbol más rápidos y más cortos.
- ID3 busca en todo el conjunto de datos para crear todo el árbol.

Desventajas de ID3

- Para una muestra pequeña, los datos pueden estar sobreajustados o sobreclasificados.
- Para tomar una decisión, solo se prueba un atributo a la vez, lo que consume mucho tiempo.

ii. C4.5

C4.5 es un algoritmo utilizado para generar un árbol de decisión que también fue desarrollado por Ross Quinlan. Es una extensión del algoritmo ID3 de Quinlan. C4.5 genera árboles de decisión que se pueden utilizar para la clasificación y, por lo tanto, C4.5 a menudo se denomina clasificador estadístico, utiliza la proporción de ganancia de información como criterio de división. Es mejor que el algoritmo ID3 porque trata con atributos continuos, discretos y también con los valores faltantes; después de la construcción del árbol C4.5 se convierte en un conjunto de reglas para podarlo (post poda) buscando mejorar el rendimiento (disminuir la tasa de mala clasificación o la suma de cuadrados del error) y de paso obtener un árbol más corto.

Para manejar valores continuos, genera un umbral o punto de corte dividiendo los atributos con valores superiores al umbral y valores iguales o inferiores al umbral. Puede inducirse a partir de un conjunto de entrenamiento que incorpora valores perdidos mediante el uso de criterios de proporción de ganancia corregida (los valores de atributos faltantes simplemente no se usan en los cálculos de ganancia y entropía).

Ventajas de C4.5

- C4.5 es fácil de implementar.
- C4.5 crea modelos que se pueden interpretar fácilmente.
- Puede manejar valores categóricos y continuos.
- Puede lidiar con el ruido y con los atributos de valor perdido.
- Al utilizar la proporción de ganancia de información como criterio de división, puede seleccionar atributos.

Desventajas de C4.5

- Una pequeña variación en los datos puede conducir a diferentes árboles de decisión cuando se usa C4.5.
- Para un conjunto de entrenamiento pequeño, C4.5 no funciona muy bien.
- Los métodos para generar el conjunto de reglas de C4.5 son lentos y requieren mucha memoria.

iii. C5.0

Según Kuhn et al. (2013), el algoritmo C5.0 es una versión más avanzada del modelo de clasificación C4.5 de Quinlan, tienen varias mejoras básicas como generar árboles más pequeños de manera mucho más rápida, adicionalmente el algoritmo permite el Boosting, que no tiene contrapartida en C4.5, donde se combinan múltiples clasificadores para mejorar la precisión predictiva. C5.0 incorpora varias instalaciones nuevas, como los costos variables de clasificación errónea. En C4.5, todos los errores se tratan como iguales, pero en aplicaciones prácticas algunos errores de clasificación son más graves que otros. C5.0 permite definir un costo por separado para cada par de clases previsto / real, esto se almacena en una matriz de costos CxC, donde C es el número de clases (los elementos diagonales se ignoran), las columnas deben corresponder a las clases verdaderas y las filas son las clases predicha, si se utiliza esta opción C5.0 construye clasificadores para minimizar los costos de clasificación errónea esperados en lugar de las tasas de error.

Adicionalmente, Patel et al. (2012) señalaron que el algoritmo C5.0 lleva a cabo un procedimiento de poda global al final, debido a que los árboles de decisión a pesar de generar

resultados precisos y eficientes, a menudo proporcionan árboles muy grandes que los hacen incomprensibles para los expertos y crecen más allá de cierto nivel de complejidad, lo cual conduce a un sobreajuste dado que el algoritmo de aprendizaje continúa desarrollando hipótesis que reducen el error del conjunto de entrenamiento a costa de un aumento de los errores en el conjunto de prueba. Para superar este problema de sobreajuste, es necesaria la poda. La poda de un árbol de decisión es un paso fundamental para optimizar la eficiencia computacional y la precisión de la clasificación. La poda generalmente resulta en la reducción del tamaño del árbol, evita la complejidad innecesaria y evita el sobreajuste de los conjuntos de datos al clasificar datos nuevos. El sobreajuste puede conducir a un número excesivamente grande de reglas, muchas de las cuales tienen muy poco valor predictivo para datos no vistos. Hay dos técnicas para podar:

A. Poda previa

La poda previa también se llama poda hacia adelante o poda en línea. La poda previa evita la generación de ramas no significativas. La poda previa de un árbol de decisión implica el uso de una "condición de terminación" para decidir cuándo es conveniente terminar algunas de las ramas prematuramente a medida que se genera el árbol. Al construir el árbol, se pueden usar algunas medidas significativas para evaluar la bondad de una división. Si particionar un nodo da como resultado una división que cae por debajo de un umbral preestablecido, entonces la partición adicional del subconjunto dado se detiene de lo contrario se expande. El umbral alto da como resultado árboles demasiado simplificados, mientras que el umbral bajo da como resultado muy poca simplicidad. El árbol deja de crecer cuando cumple cualquiera de estos criterios de poda previa o descubre las clases puras, algunos de ellos son:

- Profundida máxima (maxdepth): Este parámetro se utiliza para establecer la profundidad máxima de un árbol y donde se dejará de dividir los nodos. La profundidad es la longitud del camino más largo desde un nodo raíz hasta un nodo hoja; el nodo raíz es contado como profundidad 0. Establecer este parámetro dejará de crecer el árbol cuando la profundidad sea igual al valor establecido.
- Cantidad mínima de corte (minsplit): Es el número mínimo de registros que deben existir en un nodo para que se produzca o intente una división.

- Cantidad mínima en el nodo terminal (minCases): Es el número mínimo de registros que pueden estar presentes en un nodo hoja. Si se establece en un valor demasiado pequeño, como 1, podemos correr el riesgo de sobreajustar el modelo.

B. Post poda

Una alternativa para evitar el sobreajuste consiste en generar árboles grandes, sin condiciones de parada más allá de las necesarias por las limitaciones computacionales, para después podarlos, manteniendo únicamente la estructura robusta que consigue un test error bajo.

La selección del sub-árbol óptimo puede hacerse mediante cross-validation, sin embargo, dado que los árboles se crecen lo máximo posible (tienen muchos nodos terminales) no suele ser viable estimar el test error de todas las posibles sub-estructuras que se pueden generar. Frente a ello se utiliza la poda posterior, también se conoce como poda hacia atrás. En esto, primero se genera completamente el árbol de decisión y luego se eliminan las ramas no significativas, con el objetivo de mejorar la precisión de la clasificación, el parámetro más utilizado para lograr esto es el Costo de Complejidad o Parámetro de Complejidad (CP), el cual se usa para controlar el tamaño del árbol de decisión y para seleccionar el tamaño óptimo del árbol. Si el costo de agregar otra variable al árbol de decisión desde el nodo actual está por encima del valor de CP, entonces la construcción del árbol no continúa. También se puede decir que el parámetro de complejidad es un método de penalización, que busca el sub-árbol T que minimiza la ecuación 17:

$$\sum_{j=1}^{|T|} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 + \alpha\,|T| \qquad \ldots\ldots\ldots\ldots\ldots\ldots (17)$$

donde |T| es el número de nodos terminales del árbol.

El primer término de la ecuación se corresponde con el sumatorio total de los residuos cuadrados para un árbol de regresión o a la tasa de mala clasificación para uno de clasificación. Por definición, cuantos más nodos terminales tenga el modelo menor será esta parte de la ecuación. El segundo término es la restricción, que penaliza al modelo en función del número de nodos terminales (a mayor número, mayor penalización). El grado de

penalización se determina mediante el tunning parameter α. Cuando α=0, la penalización es nula y el árbol resultante es equivalente al árbol original. A medida que se incrementa α la penalización es mayor y como consecuencia, los árboles resultantes son de menor tamaño. El valor óptimo de α puede identificarse mediante cross validation.

Adicionalmente, C5.0 mide la importancia del predictor al determinar la ganancia de información o el primer porcentaje de muestra del conjunto de entrenamiento que cae en un nodo antes de una división; el predictor en la primera división tiene automáticamente una medida de importancia del 100% ya que todas las muestras se ven afectadas por esta división. Otros predictores pueden usarse con frecuencia en divisiones, pero si los nodos terminales cubren solo un puñado de muestras de conjuntos de entrenamiento, los puntajes de importancia pueden ser cercanos a cero. La misma estrategia se aplica a los modelos basados en reglas y a las versiones mejoradas del modelo.

C5.0 también tiene una opción para seleccionar o eliminar predictores: un algoritmo inicial descubre qué predictores tienen una relación con la variable objetivo, y el modelo final se crea solo a partir de predictores importantes. Para hacer esto, el conjunto de entrenamiento se divide aleatoriamente por la mitad y se crea un árbol con el fin de evaluar la utilidad de los predictores (llame a esto el " winnowing tree"). Dos procedimientos caracterizan la importancia de cada predictor para el modelo:

1. Los predictores no se consideran importantes si no se encuentran en ningún nodo del "winnowing tree".
2. La mitad de las muestras del conjunto de entrenamiento no incluidas en el paso 1 se usan para estimar la tasa de error del árbol. La tasa de error también se estima sin cada predictor y se compara con la tasa de error cuando se utilizan todos los predictores. Si la tasa de error mejora sin el predictor, se considera irrelevante y se elimina provisionalmente.

Una vez que se establece la lista tentativa de predictores no informativos, C5.0 recrea el árbol. Si la tasa de error ha empeorado, el proceso de eliminación se desactiva y no se excluyen los predictores. Una vez establecidos los predictores importantes (si los hay), el

proceso de entrenamiento convencional C5.0 se usa con el conjunto completo de entrenamiento, pero solo con los predictores que se seleccionaron en el "winnowing tree".

Según Pandya et al. (2015), el algoritmo C4.5 sigue las reglas del algoritmo ID3. Del mismo modo, el algoritmo C5.0 sigue las reglas del algoritmo de C4.5. El algoritmo C5.0 tiene muchas características como:

- El gran árbol de decisiones se expresa como un conjunto de reglas fáciles de interpretar.
- El algoritmo C5.0 puede lidiar con el ruido y con los atributos de valor perdido.
- El problema del ajuste excesivo y la poda de errores se resuelve con el algoritmo C5.0
- El clasificador C5.0 puede anticipar qué atributos son relevantes y cuáles no son relevantes en la clasificación.
- El algoritmo aprovecha la capacidad de procesamiento de datos para acelerar el análisis, ya que puede ser programado para que procese de manera paralela en varios núcleos de una computadora o servidor.

iv. Random Forest

Según Breiman (2001), Random Forest son un algoritmo de ensamble, es decir, en lugar de ajustar un único árbol, se ajustan muchos de ellos en paralelo formando un "bosque".
En cada nueva predicción, todos los árboles que forman el "bosque" participan aportando su predicción. Como valor final, se toma la media de todas las predicciones (variables continuas) o la clase más frecuente (variables cualitativas); lo cual permite lograr un equilibrio entre el error de predicción y sobreajuste a los datos de entrenamiento.

A comparación de otros algoritmos de ensamble, Random Forest trabaja con una combinación de árboles decorrelacionados, es decir, cada árbol selecciona un subconjunto aleatorio de predictores antes de evaluar cada nodo, lo cual permite que un predictor muy influyente que destaca sobre el resto de variables no sea usado repetitivamente en todos los árboles en la primera ramificación, dando la posibilidad que otros predictores sean

seleccionados, produciendo que los arboles no sean similares y las predicciones entre ellos no estén altamente correlacionadas; consiguiendo de esta manera la reducción sustancial de la varianza con respecto a un solo árbol.

El algoritmo construye un número de árboles de decisión a partir de pseudo-muestras generadas del mismo tamaño que la muestra real mediante muestreo con reemplazo (bootstrapping), en cada árbol, antes de cada división se seleccionan aleatoriamente m predictores como candidatos, donde un valor recomendado del hiperparámetro es $m = \sqrt{p}$, siendo p el número de predictores totales.

En resumen, cada árbol se construye:

1. Dado que el número de casos en el conjunto de datos es N, se selecciona la tercera parte de este como data de validación o test y del resto se toma una muestra aleatoriamente CON REEMPLAZO de tamaño N.
2. Con el conjunto de datos de entrenamiento se construye el árbol i y se estima el error de este, utilizando la data de validación.
3. Si existen M variables de entrada, un número $m < M$ se especifica tal que para cada nodo, m variables se seleccionan aleatoriamente de M. La mejor división de estos m atributos es usado para ramificar el árbol. El valor m se mantiene constante durante la generación de todo el bosque.
4. Cada árbol crece hasta su máxima extensión posible y NO hay proceso de poda.
5. Nuevas instancias se predicen a partir de la agregación de las predicciones de los x árboles (i.e., mayoría de votos para clasificación, promedio para regresión).

Adicionalmente, Random Forest posee una opción para medir la importancia de los predictores y con ella hacer una selección de variables, para esto se basa en distintas medidas tales como:

1. Importancia de la permutación o disminución media de la precisión

Cuando se construye cada árbol, este tiene su propia muestra de datos que no se utilizó durante el entrenamiento. Esta muestra se usa para medir la fuerza de predicción de cada variable; primero, se registra la precisión de predicción en cada árbol con los datos que no se utilizó durante el entrenamiento; luego, los valores para la i-ésima variable en cada muestra test se permutan aleatoriamente y la precisión se calcula nuevamente. Las disminuciones de las precisiones como resultado de esta permutación son promediadas y se normaliza por la desviación estándar de las diferencias, esta se usa como una medida de la importancia de la variable i. La escala es irrelevante: solo importan los valores relativos.

$$\frac{promedio(Disminuciones\ en\ la\ precisión\ de\ los\ árboles)}{desv\_estand(Disminuciones\ en\ la\ precisión\ de\ los\ árboles)} \quad \text{.....(18)}$$

La aleatorización anula efectivamente el efecto de una variable, esto no mide el efecto sobre la predicción si esta variable no estuviera disponible, porque si el modelo se reajustara sin la variable, otras variables podrían usarse como sustitutos.

2. Importancia de Gini / disminución media de la impureza

Cuando se construye un árbol, la decisión sobre qué variable dividir en cada nodo utiliza un cálculo de la impureza de Gini (disminución total de la impureza del nodo), el cual mide las divergencias entre las distribuciones de probabilidad de los valores del atributo objetivo, si disminuye el valor de Gini aumenta la homogeneidad, es decir se prefiere las variables que tengan un Gini *mínimo*, este se define como:

$$G_m = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad \text{..... (19)}$$

Donde $\hat{p}_{mk}$ representa la proporción de observaciones del nodo m que pertenecen a la clase k. Para cada variable, se calcula la suma de la disminución del Gini en cada árbol del bosque ponderada por la probabilidad de llegar a ese nodo (la proporción de muestras que llegan a ese nodo) se acumula cada vez que se elige esa variable para dividir un nodo. La suma

acumulada se divide por el número de árboles en el bosque para dar un promedio. La escala es irrelevante: solo importan los valores relativos.

$$\frac{\sum(disminución\ del\ Gini * probabilidad)}{cantidad\ de\ árboles} \quad \text{..... (20)}$$

donde $probabilidad = \ la\ proporción\ de\ muestras\ que\ llegan\ a\ ese\ nodo$.

v. Algoritmo de selección de variables Boruta

Según B. Kursa et al. (2010), Boruta es un algoritmo de selección de características relevante, capaz de trabajar con cualquier método de clasificación que genere una medida de importancia; de forma predeterminada, Boruta usa Random Forest. El método realiza una búsqueda de arriba hacia abajo para las características relevantes comparando la importancia de los atributos originales con la importancia que se puede obtener al azar, estimada usando sus copias permutadas y eliminando progresivamente las características irrelevantes.

Para medir la importancia de las variables, se necesita alguna referencia externa para decidir si la importancia de un atributo dado es significativa, es decir, si es discernible de la importancia que puede surgir de fluctuaciones aleatorias. Con este fin, se amplía el sistema de información con atributos que son aleatorios por diseño. Para cada atributo se crea un atributo de "sombra" correspondiente, cuyos valores se obtienen barajando los valores del atributo original entre los objetos. Luego se realiza una clasificación utilizando todos los atributos de este sistema extendido y se calcula la importancia de todos los atributos. La importancia de un atributo de sombra puede ser distinto de cero solo debido a fluctuaciones aleatorias. Por lo tanto, el conjunto de importancia de los atributos de sombra se utiliza como referencia para decidir qué atributos son realmente importantes.

En resumen, Boruta se basa en la misma idea que forma la base de random forest aleatorio, a saber, que al agregar aleatoriedad al sistema y recolectar resultados del conjunto de muestras aleatorias se puede reducir el impacto engañoso de las fluctuaciones y

correlaciones aleatorias. Aquí, esta aleatoriedad adicional proporciona una visión más clara de qué atributos son realmente importantes.

A continuación, se muestra el funcionamiento del algoritmo Boruta:

a) Extender el sistema de información agregando copias aleatorias de todas las variables, llamadas “sombras” (el sistema de información siempre se extiende por al menos 5 atributos de sombra, incluso si el número de atributos en el conjunto original es inferior a 5).
b) Se mezcla los atributos sombra a los originales.
c) Se ejecute un algoritmo Random Forest, con el sistema de información extendido y reúna los puntajes de importancia calculados.
d) Se encuentra el puntaje de importancia máximo entre los atributos de sombra (maximum Z score among shadow attributes o MZSA) y en cada árbol se asigna un flat de éxito a cada atributo que obtuvo un puntaje mejor que MZSA. Al finalizar este proceso se calcula la proporción de veces que el atributo se marcó como éxito; de no ser igual a 1, esta variable puede ser rechaza y eliminada de la matriz original.
e) Se considera que los atributos que tienen una importancia mediana menor que MZSA son "sin importancia" y elimínelos permanentemente del sistema de información.
f) Se considera, también, que los atributos que tienen una importancia mediana mayor que MZSA son "importantes".
g) Finalmente, se eliminar todos los atributos de sombra.

III. METODOLOGÍA

3.1. Formulación de hipótesis

3.1.1. Hipótesis general

Se puede caracterizar el perfil de los ingresantes de una universidad pública respecto a sus variables sociodemográficas (edad, sexo, procedencia, periodo transcurrido entre colegio y universidad), económicas (aporte semestral) y de rendimiento académico (notas en los cursos del último año de colegio, notas en el examen de admisión, modalidad de ingreso, tercio superior, elección de opción de la carrera a la que ingresó y especialidad) utilizando algoritmos de segmentación.

3.1.2. Hipótesis específicas

- Es posible determinar el algoritmo de segmentación más adecuado para el estudio de caso, utilizando indicadores de validación interna clustering.
- Se puede encontrar el número de conglomerados óptimo para el estudio de caso, utilizando indicadores de validación interna clustering.
- Es posible identificar las variables más importantes para caracterizar el perfil de los ingresantes de una universidad pública.

3.2. Datos

La investigación fue aplicada a los alumnos ingresantes de la Universidad Nacional Agraria La Molina (UNALM) de los semestres 2015-I y 2015-II, y la información necesaria fue obtenida a partir de la vinculación entre las bases de datos de la Oficina de Estudios y Registros Académicos, del Centro de Admisión y Promoción y la Oficina de Bienestar Universitario y Asuntos Estudiantiles.

3.3. Población

La población investigada fueron todos los alumnos ingresantes de la UNALM de las modalidades: Concurso Ordinario y Dos Primeros Puestos de Colegios de Educación Secundaria de los semestres 2015-I y 2015-II con un total de 690 estudiantes.

3.4. Identificación de variables

3.4.1. Variables para la segmentación

Las variables identificadas en la aplicación de ambas técnicas fueron:

i. Variables socio-demográficas:

ID	Variable	Tipo de Variable	Descripción
X1:	Años_Colegio_ Admisión	CUANTITATIVA (Continua)	Tiempo transcurrido desde que terminó el 5to año de secundario e ingresó a la universidad
X2:	Edad_Admisión	CUANTITATIVA (Continua)	Edad del ingresante al momento del examen de admisión
X3:	Dept_Colegio	CUALITATIVA (Nominal)	Ubicación del colegio donde cursó el 5to año de secundaria (Lima o Provincia)
X4:	Sexo	CUALITATIVA (Nominal)	Sexo del ingresante

ii. Variables socio-educativas:

ID	Variable	Tipo de Variable	Descripción
X5:	Tipo_Colegio	CUALITATIVA (Nominal)	Tipo de institución de procedencia (Privada o Pública)

iii. Variables socio-económicas:

ID	Variable	Tipo de Variable	Descripción
X6:	Aporte_Semestral	CUANTITATIVA (Continua)	Aporte semestral asignado al ingresante

iv. Variables de rendimiento en las áreas del conocimiento en la secundaria:

ID	Variable	Tipo de Variable	Descripción
X7:	CTA_Colegio	CUANTITATIVA (Continua)	Nota obtenida en el 5to año de secundaria en el área de Ciencia tecnología y Ambiente
X8:	COM_Colegio	CUANTITATIVA (Continua)	Nota obtenida en el 5to año de secundaria en el área de Comunicación
X9:	MAT_Colegio	CUANTITATIVA (Continua)	Nota obtenida en el 5to año de secundaria en el área de Matemática
X10:	Nota_Colegio	CUANTITATIVA (Continua)	Nota promedio del último año de estudios

v. Variables de rendimiento en el examen de Admisión:

ID	Variable	Tipo de Variable	Descripción
X11:	RM_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el curso de RM en el examen de admisión
X12:	RV_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el curso de RV en el examen de admisión
X13:	MAT_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el área de Matemática en el examen de admisión

X14:	FIS_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el curso de Física en el examen de admisión
X15:	QUI_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el curso de Química en el examen de admisión
X16:	BIO_Admisión	CUANTITATIVA (Continua)	Nota obtenida en el curso de Biología en el examen de admisión
X17:	Nota_Admisión	CUANTITATIVA (Continua)	Nota general obtenida en el examen de admisión
X18:	Tercio_Superior_ESP	CUALITATIVA (Ordinal)	Si el alumno pertenece o no al tercio superior en la especialidad a la que ingresó

vi. Variables de elección en el ingreso a una carrera:

ID	Variable	Tipo de Variable	Descripción
X19:	Modalidad	CUALITATIVA (Nominal)	Modalidad de ingreso a la universidad
X20:	Especialidad	CUALITATIVA (Nominal)	Especialidad a la que ingresó
X21:	Elección_ESP_Ingreso	CUALITATIVA (Ordinal)	Orden de elección que tuvo la carrera a la cual ingresó (PRIMERA, SEGUNDA, TERCERA opción)

3.4.2. Variable pasiva

La variable utilizada para la validación fué:

ID	Variable	Tipo de Variable	Descripción
Z1:	PROM_Ponderado_ Acumulado	CUANTITATIVA (Continua)	Promedio ponderado acumulado del alumno obtenido al finalizar el primer año de estudios universitarios.

3.5. Tipo de investigación

El tipo de investigación fue de carácter descriptivo, se identificó y determinó el perfil del ingresante de la UNALM a través de la descripción de sus variables socio-demográficas (X1,X2,X3,X4), socio-educativas (X5), socio-económicas (X6), de rendimiento en las áreas del conocimiento en la secundaria (X7,X8,X9.X10), de rendimiento en el examen de admisión (X11,X12,X13,X14,X15,X16,X17,X18) y de elección en el ingreso a una carrera (X19,X20,X21).

3.6. Diseño de investigación

El diseño de investigación fue de carácter no experimental-transversal, ya que se contó con un conjunto de datos de estudiantes, los cuales se recolectó de los archivos personales tomados en la Ficha Socio-Económica administradas por las asistentas sociales de la OBUAE, también se tuvo los registros del rendimiento en las áreas del conocimiento en la secundaria y las notas de los exámenes de admisión, los cuales son depositados oficialmente en el Centro de Admisión y Promoción.

3.7. Instrumento de colecta de datos

3.7.1. Examen de admisión

Uno de los instrumentos empleados en la obtención de los datos requeridos para la investigación fueron el examen de admisión elaborado por el Comité Permanente de Admisión. Este instrumento tuvo un tiempo de aplicación de aproximadamente tres horas, 100 preguntas con cinco alterativas, donde sólo hay una respuesta correcta. Las preguntas están distribuidas en nueve cursos de la siguiente forma: Razonamiento Matemático (l4), Razonamiento Verbal (20), Aritmética (8), Álgebra (6), Geometría (6), Trigonometría (4), Física (l4), Química (14) y Biología (l4). Cada pregunta bien contestada tuvo un valor de 1.00 punto, sin contestar 0.00 y mal contestada - 0.25.

3.7.2. Ficha del ingresante

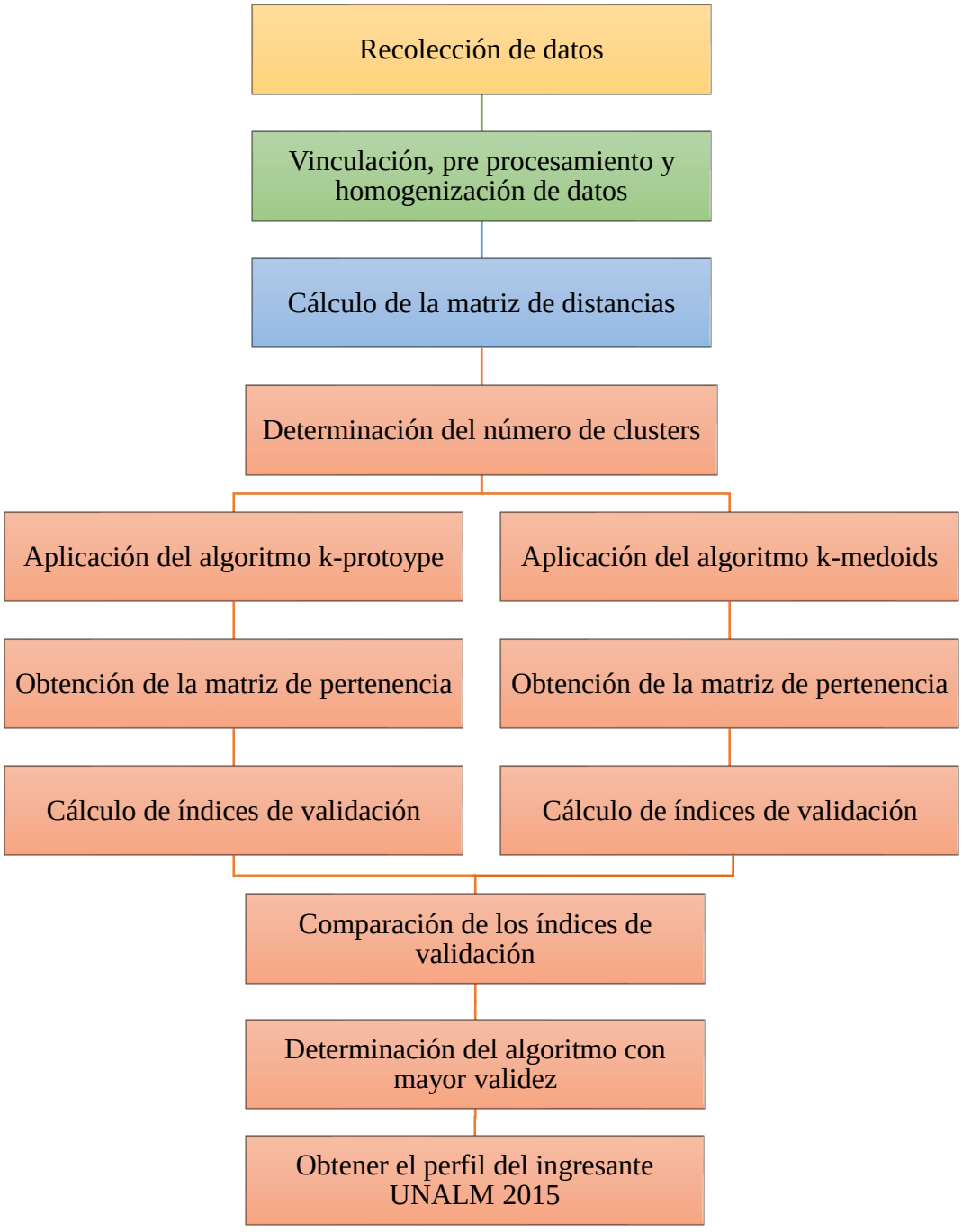
La ficha del ingresante es un documento elaborado por la OBUAE como parte de la información que es manejada por el Departamento de Asuntos Estudiantiles. Este documento fue completado por los ingresantes a la UNALM juntamente con los padres si son menores de edad o dependientes económicamente. Esta base de datos se encuentró en formato físico y contenía la información básica que identifica al ingresante por apellidos y nombres, género, fecha de nacimiento, edad de ingreso, colegio o institución educativa de procedencia, nivel educativo de los padres, condición laboral de los padres, la situación laboral de los padres y del estudiante, la constitución de la familia, el ingreso total familiar.

3.7.3. Certificado de estudio escolar

El certificado de estudio escolar de los ingresantes es un documento que certifica y registra las notas de los cursos en los últimos años de educación secundaria. El documento es almacenado por la OBUAE como parte de la información que es administrada por el Departamento de Asuntos Estudiantiles en cada folder personal de los ingresantes.

3.8. Procedimiento de análisis de datos

Esquematización del procedimiento:



IV. RESULTADOS Y DISCUSIÓN

En la presente investigación se aplicaron algoritmos clustering K-medoids y K- prototype para obtener el perfil de los ingresantes de la UNALM según sus variables socio-demográficas, socio-educativas, socio-económicas, áreas de rendimiento académico en el último año de educación secundaria, en el examen de admisión y la opción de elección de la carrera a la cual ingresó; en las modalidades: Concurso Ordinario y Dos Primeros Puestos de Colegios de Educación Secundaria de los semestres 2015-I y 2015-II. Antes de aplicar los algoritmos se procedió a realizar un análisis de las variables en estudio.

4.1. Análisis estadístico univariado

Tabla 1: Resumen para variables cuantitativas

	VARIABLES	MEDIANA	MEDIA	DESVIACIÓN ESTANDAR	RECUENTO
X1	Años_Colegio_Admisión	2.2	2.2	1.7	690.0
X2	Edad_Admisión	18.6	18.9	1.7	690.0
X6	Aporte_Semestral	200.0	204.2	177.9	690.0
X7	CTA_Colegio	15.0	15.4	2.1	690.0
X8	COM_Colegio	15.0	15.3	2.0	690.0
X9	MAT_Colegio	15.0	15.3	2.1	690.0
X10	NOTA_Colegio	15.6	15.6	1.6	690.0
X11	RM_Admisión	13.2	13.0	3.0	690.0
X12	RV_Admisión	9.3	9.2	3.1	690.0
X13	MAT_Admisión	12.1	11.8	2.9	690.0
X14	FIS_Admisión	12.9	12.5	3.5	690.0
X15	QUI_Admisión	15.4	15.0	3.1	690.0
X16	BIO_Admisión	10.0	9.5	3.9	690.0
X17	NOTA_Admisión	11.8	11.8	1.7	690.0

En el Tabla N° 1 se observó que en promedio los ingresantes 2015 tienen una edad promedio de 19 años, terminaron el colegio con una nota media de 16 y la nota promedio alcanzada en el examen de admisión fue de 12, finalmente las variables con mayor variabilidad fueron

Aporte_Semestral, BIO_Admisión, FIS_Admisión, QUI_Admisión, RV_Admisión y RM_Admisión.

Tabla 2: Resumen para variables cualitativas

VARIABLES		ATRIBUTOS	RECUENTO	% DEL N COLUMNA
X3	Dept_Colegio	Lima	635.0	92.0%
		Provincia	55.0	8.0%
		Total	690.0	
X4	Sexo	Femenino	346.0	50.1%
		Masculino	344.0	49.9%
		Total	690.0	
X5	Tipo_Colegio	Privada	422.0	61.2%
		Pública	268.0	38.8%
		Total	690.0	
X18	Tercio_Superior_ESP	No	482.0	69.9%
		Si	208.0	30.1%
		Total	690.0	
X19	Modalidad	Dos Primeros Puestos de Colegios de Educación Secundaria	69.0	10.0%
		Concurso Ordinario	621.0	90.0%
		Total	690.0	
X20	Especialidad	Agronomía	128.0	18.6%
		Biología	44.0	6.4%
		Ciencias Forestales	45.0	6.5%
		Economía	44.0	6.4%
		Estadística Informática	40.0	5.8%
		Gestión Empresarial	52.0	7.5%
		Industrias Alimentarias	68.0	9.9%
		Ingeniería Agrícola	62.0	9.0%
		Ingeniería Ambiental	39.0	5.7%
		Meteorología	23.0	3.3%
		Pesquería	60.0	8.7%
		Zootecnia	85.0	12.3%
		Total	690.0	
X21	Elección_ESP_Ingreso	Primera opción	269.0	39.0%
		Segunda opción	270.0	39.1%
		Tercera opción	151.0	21.9%
		Total	690.0	

En el Tabla 2 se aprecia que la mayoría de los ingresantes 2015 provienen de colegios ubicados en Lima (92%), privados (61%) y más de la tercera parte ingresó a la carrera que marcaron como primera opción (39%).

4.2. Pre procesamiento de datos

Antes de realizar el análisis clustering, se procedió al manejo de datos atípicos para las variables cuantitativas, para ello se utilizó diagramas de cajas con lo que se identificó los valores outliers extremos superiores e inferiores, los cuales fueron reemplazaron por el Q99-Q97 o Q1-Q3 respectivamente, teniendo en consideración no perder la variabilidad de cada atributo, por lo que se reemplazó como máximo el 3% de los datos. Los gráficos elaborados se encuentran en el anexo 1.

4.3. Transformación de variables

En la presente investigación las variables cuantitativas que son utilizadas para formar los clusters están en escalas diferentes, frente a ello, será necesario estandarizarlas y llevarlo a una escala entre cero y uno, con la finalidad de eliminar sus unidades de medida que pueden influenciar en la métrica de segmentación, es decir:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)} \qquad \text{..... (21)}$$

4.4. Aplicación del algoritmo K-medoids

A continuación, se desarrolla el proceso para la aplicación algoritmos K-medoids

A. Seleccionar la medida de distancia

Como anteriormente se señaló en el punto 5.3 la medida de disimilitud utilizada combina la medida de distancia euclidiana al cuadrado en los atributos numéricos y una medida de disimilitud de coincidencia simple en los atributos categóricos.

B. Estimación del número óptimo de clusters (*k*) y semilla inicial

Para aplicar el algoritmo K-medoids es necesario conocer a priori el número de clusters (*k*) a formarse, aspecto muy importante ya que una mala elección de *k* puede dar lugar a realizar agrupaciones de datos muy heterogéneos (pocos clusters); o datos, que siendo muy similares unos a otros se agrupen en clusters diferentes (muchos clusters).

En este caso, se utilizó el índice de validación interna de Davies Bouldin, índice de Dunn y el índice de Calinski Harabasz calculándolos de manera iterativa cambiando el número de cluster y el valor de semilla inicial, el valor de *k* seleccionado fue aquel que permitió obtener el índice de validación interna óptimo.

Tabla 3: Determinar el número de clusters con el índice de Davies-Bouldin

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
10	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
20	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
30	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
40	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
50	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
60	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
70	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
80	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
90	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37
100	2.32	1.80	1.86	1.90	2.12	1.83	2.07	1.93	2.03	2.25	2.21	2.37

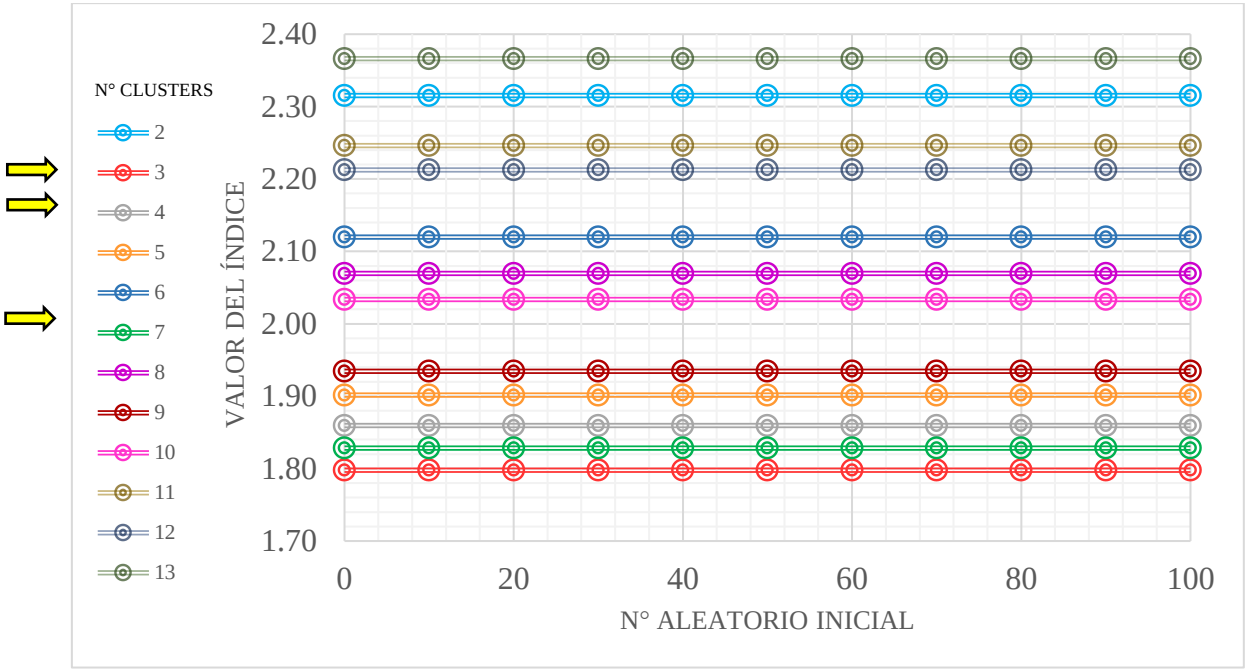


Figura 11: Determinar el número de clusters con el índice de Davies-Bouldin

Se observa en el Tabla N° 3 y la Figura N° 11 que al aplicar el algoritmo clustering K-medoids los valores del índice de validación interna de Davies-Bouldin óptimos fueron 1.80, 1.83 y 1.86, por lo que el número clusters posibles fueron K=3, K=7 o K=4, sin importar la semilla inicial que se considere, ya que el algoritmo no muestra variabilidad en sus resultados y siempre converge en la misma solución.

Tabla 4: Determinar el número de clusters con el índice de Dunn

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
10	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
20	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
30	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
40	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
50	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
60	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
70	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
80	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
90	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08
100	0.10	0.16	0.13	0.13	0.10	0.15	0.11	0.11	0.11	0.08	0.08	0.08

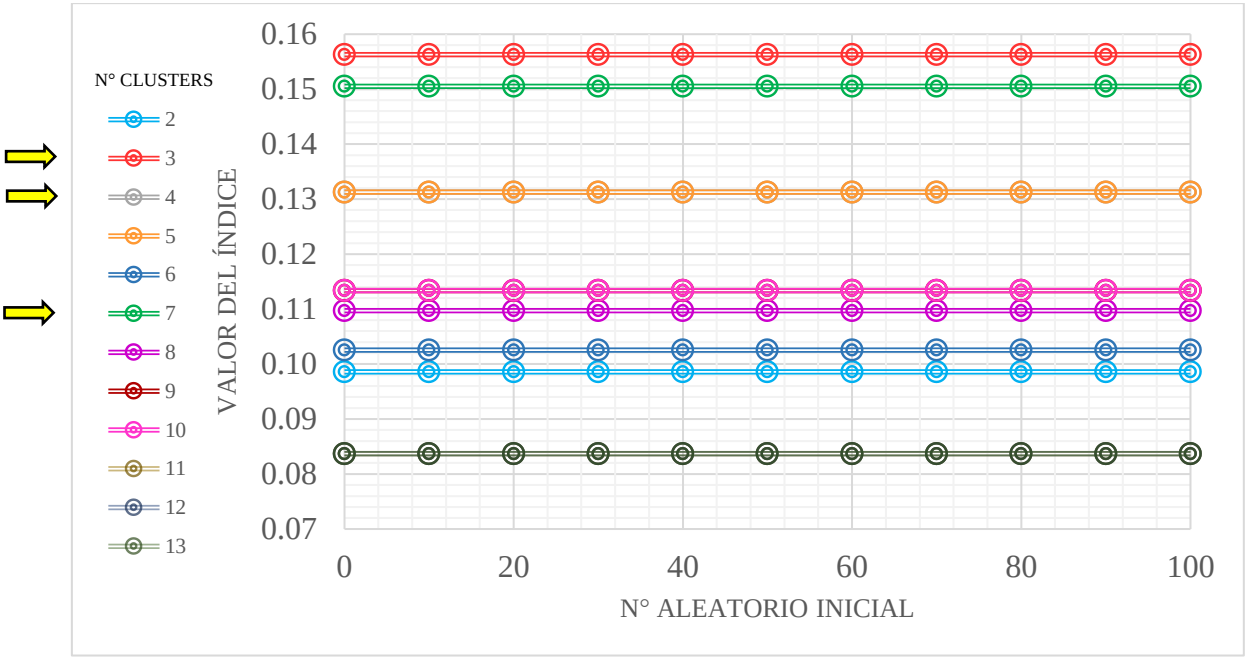


Figura 12: Determinar el número de clusters con el índice de Dunn

Se observa en el Tabla N° 4 y la Figura N° 12 que al aplicar el algoritmo clustering K-medoids los valores del índice de validación interna de Dunn óptimos fueron 0.16, 0.15 y 0.13, por lo que el número clusters posibles fueron K=3, K=7 o K=4, sin importar la semilla inicial que se considere, ya que el algoritmo no muestra variabilidad en sus resultados y siempre converge en la misma solución.

Tabla 5: Determinar el número de clusters con el índice de Calinski Harabasz

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
10	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
20	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
30	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
40	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
50	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
60	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
70	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
80	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
90	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0
100	158.9	240.2	196.3	163.0	137.7	141.0	126.6	141.5	130.2	128.0	119.7	107.0

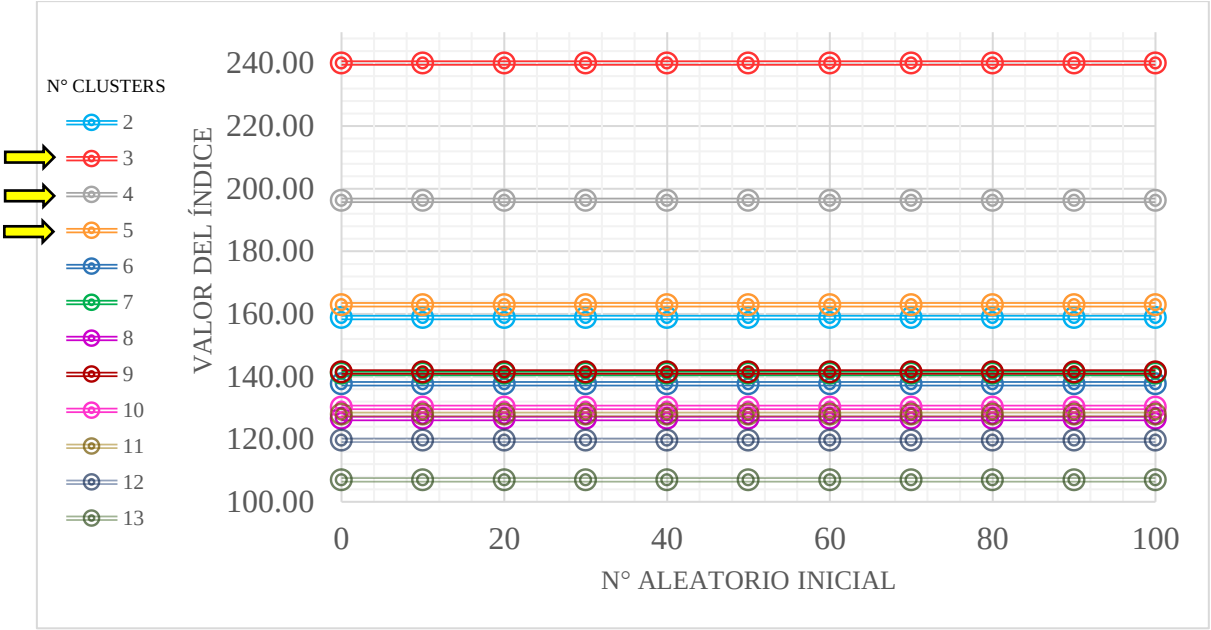


Figura 13: Determinar el número de clusters con el índice de Calinski Harabasz

Se observa en el Tabla N° 5 y la Figura N° 13 que al aplicar el algoritmo clustering K-medoids los valores del índice de validación interna de Calinski Harabasz óptimos fueron 240.2, 196.3 y 163.0, por lo que el número clusters posibles fueron K=3, K=4 o K=5, sin importar la semilla inicial que se considere, ya que el algoritmo no muestra variabilidad en sus resultados y siempre converge en la misma solución.

C. Aplicación del algoritmo K-medoids

Se determinó que el número de cluster óptimo dado los índices de validación interna desarrollados, por mayoría, para el algoritmo clustering K-medoids fue K=3 y la semilla inicial que se considere no afecta los resultados finales ya que el algoritmo internamente hace todas las posibles combinaciones para determinar la mejor segmentación, esto fue desarrollado en el punto 5.2.6.1.

D. Obtención de la matriz de pertenencia

Al aplicar el algoritmo, iterativamente se asignó a cada ingresante un cluster con observaciones similares y se determinó los centros de los conglomerados o medoids, los cuales caracterizan al cluster al cual pertenece.

Tabla 6: Tabla de distribución K medoids

CLUSTER	FRECUENCIA	PORCENTAJE
1	249	36%
2	291	42%
3	150	22%
Total	690	100%

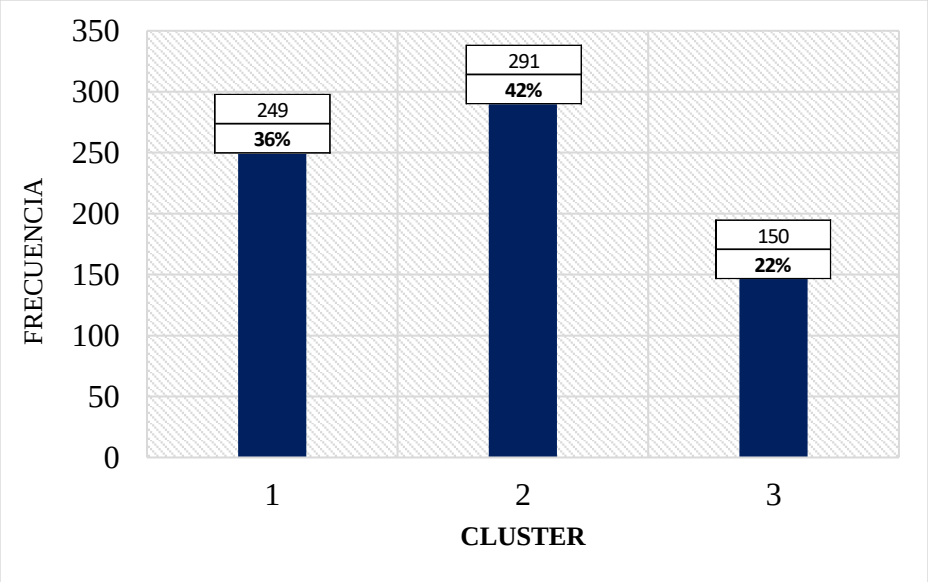


Figura 14: Figura de distribución K medoids

La mayor cantidad de ingresantes fueron agrupados en el cluster 2 (291 individuos) que representa el 42% de los datos, mientras que el cluster 3 tiene la menor cantidad de observaciones (150 individuos) representando el 22% de los datos.

E. Cálculo de índices de validación interna clustering

En resumen, dado el número de clusters (K=3) los índices de validación interna resultaron:

Tabla 7: Índices de validación interna K medoids

ÍNDICE	VALOR DEL ÍNDICE
Davies-Bouldin	1.798
Dunn	0.156
Calinski Harabasz	240.152

4.5. Aplicación del algoritmo K- prototype

A continuación, se desarrolla el proceso para la aplicación algoritmos K- prototype

A. Seleccionar la medida de distancia

Como anteriormente se señaló en el punto 5.3 la medida de disimilitud utilizada combina la medida de distancia euclidiana al cuadrado en los atributos numéricos y una medida de disimilitud de coincidencia simple en los atributos categóricos.

B. Estimación del número óptimo de clusters (*k*) y semilla inicial

Para aplicar el algoritmo K- prototype es necesario conocer a priori el número de clusters (*k*) a formarse. En este caso, se utilizó el índice de validación interna de Davies-Bouldin, índice de Dunn y el índice de Calinski Harabasz, calculándolos de manera iterativa cambiando el número de cluster y el valor de semilla inicial, el valor de *k* seleccionado fue aquel que permitió obtener el índice de validación interna más óptimo.

Tabla 8: Determinar el número de clusters con el índice de Davies-Bouldin

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	1.97	2.15	2.09	2.34	2.16	2.24	2.22	2.36	2.50	2.76	2.98	2.49
10	2.26	2.19	1.94	2.20	2.10	2.07	2.17	2.15	2.22	2.36	2.26	2.70
20	1.96	2.58	2.17	2.02	2.16	2.00	2.19	2.24	2.26	2.31	2.23	2.18
30	2.26	2.20	2.31	2.14	1.97	2.26	2.19	2.10	2.35	2.03	2.28	2.29
40	1.98	2.24	2.09	1.89	1.95	2.36	2.39	2.25	2.23	2.21	2.49	2.41
50	1.97	1.78	1.99	2.10	2.15	2.08	2.10	2.30	2.25	2.37	2.20	2.21
60	2.26	2.83	1.95	1.83	2.10	2.41	2.27	2.79	2.27	2.57	2.58	2.61
70	2.26	1.92	2.78	3.04	2.53	2.43	2.26	2.37	2.79	2.77	2.32	2.28
80	2.26	2.01	1.88	2.00	2.68	2.32	2.17	2.16	2.15	2.63	2.32	2.47
90	2.26	2.24	2.33	1.85	2.23	2.25	2.20	2.17	2.41	2.26	2.28	2.39
100	1.98	2.42	2.14	2.10	2.65	2.31	2.52	2.30	2.36	2.50	2.56	2.39

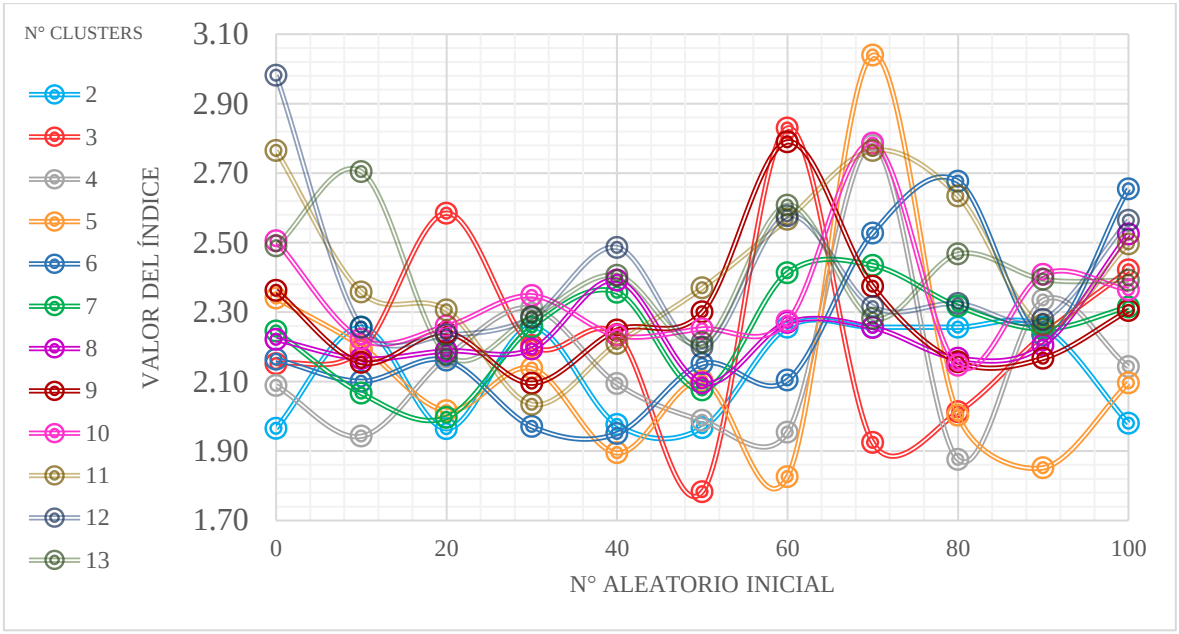


Figura 15: Determinar el número de clusters con el índice de Davies-Bouldin

Se observa en el Tabla N° 8 y la Figura N° 15 que al aplicar el algoritmo clustering K-prototype los valores del índice de validación interna de Davies-Bouldin óptimos fueron 1.78, 1.83 y 1.88 por lo que el número clusters posibles fueron K=3, K=5 o K=4, con la semilla inicial seed=50, seed=60 o seed=80 respectivamente, visualizando la variabilidad en sus resultados por la naturaleza del algoritmo.

Tabla 9: Determinar el número de clusters con el índice de Dunn

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	0.11	0.13	0.11	0.10	0.11	0.11	0.11	0.09	0.09	0.06	0.07	0.08
10	0.09	0.11	0.13	0.10	0.12	0.10	0.11	0.11	0.11	0.08	0.11	0.08
20	0.10	0.10	0.13	0.12	0.13	0.12	0.10	0.10	0.10	0.08	0.11	0.14
30	0.09	0.13	0.14	0.12	0.13	0.08	0.08	0.11	0.08	0.13	0.10	0.11
40	0.10	0.11	0.11	0.13	0.13	0.10	0.08	0.09	0.13	0.12	0.09	0.11
50	0.10	0.17	0.11	0.10	0.12	0.12	0.10	0.08	0.08	0.06	0.09	0.09
60	0.09	0.07	0.11	0.12	0.12	0.09	0.09	0.06	0.12	0.07	0.07	0.09
70	0.09	0.12	0.08	0.07	0.08	0.09	0.09	0.09	0.07	0.07	0.09	0.11
80	0.09	0.12	0.13	0.13	0.08	0.11	0.13	0.13	0.13	0.07	0.09	0.08
90	0.09	0.11	0.11	0.15	0.10	0.10	0.10	0.11	0.09	0.10	0.10	0.10
100	0.10	0.13	0.11	0.10	0.08	0.09	0.06	0.08	0.09	0.08	0.10	0.11

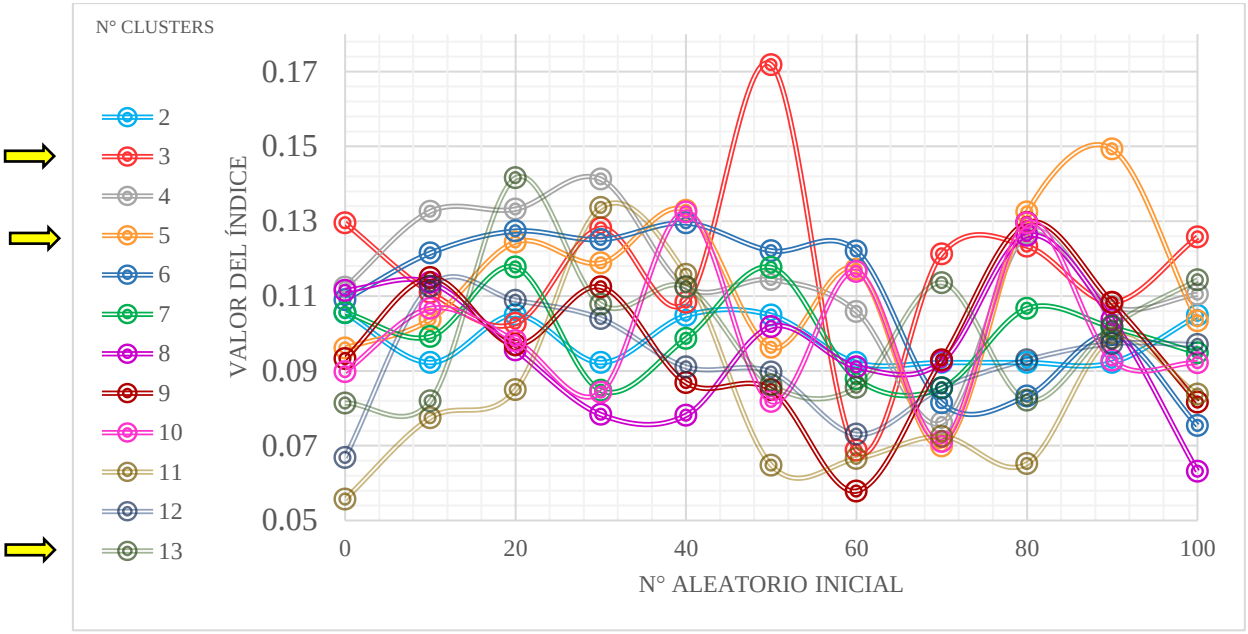


Figura 16: Determinar el número de clusters con el índice de Dunn

Se observa en el Tabla N° 9 y la Figura N° 16 que al aplicar el algoritmo clustering K-prototype los valores del índice de validación interna de Dunn óptimos fueron 0.17, 0.15 y 0.14, por lo que el número clusters posibles fueron K=3, K=5 o K=13, con la semilla inicial seed=50, seed=90 o seed=20 respectivamente, visualizando la variabilidad en sus resultados por la naturaleza del algoritmo.

Tabla 10: Determinar el número de clusters con el índice de Calinski Harabasz

N° ALEATORIO INICIAL	N° CLUSTERS											
	2	3	4	5	6	7	8	9	10	11	12	13
0	87.6	68.1	79.4	80.9	82.3	81.4	77.7	72.9	67.3	66.1	60.0	60.3
10	55.2	77.2	111.9	78.2	75.1	89.8	81.1	76.3	71.4	71.8	76.2	69.7
20	70.9	68.3	77.0	72.8	85.9	86.8	83.1	71.8	68.0	66.0	66.7	61.7
30	55.2	79.7	67.9	95.5	101.1	92.2	81.3	79.4	76.8	75.2	71.2	59.6
40	70.2	61.0	68.7	93.7	86.3	73.8	71.9	73.5	73.4	70.8	66.7	63.5
50	70.8	83.6	100.3	81.4	94.9	84.1	79.4	73.9	78.5	73.2	68.3	67.9
60	55.2	53.9	84.7	95.3	80.4	77.0	78.1	80.5	74.5	69.2	62.7	61.3
70	55.2	66.7	78.4	65.9	82.0	78.9	85.8	69.8	63.9	62.7	66.8	57.1
80	55.2	71.7	90.2	92.0	81.3	81.9	81.8	86.2	78.7	63.6	68.4	63.8
90	55.2	61.0	86.9	99.1	87.2	79.4	79.1	67.8	61.2	58.1	61.6	54.1
100	87.2	61.5	97.5	100.6	85.7	82.5	82.3	76.4	77.5	72.8	72.1	68.4

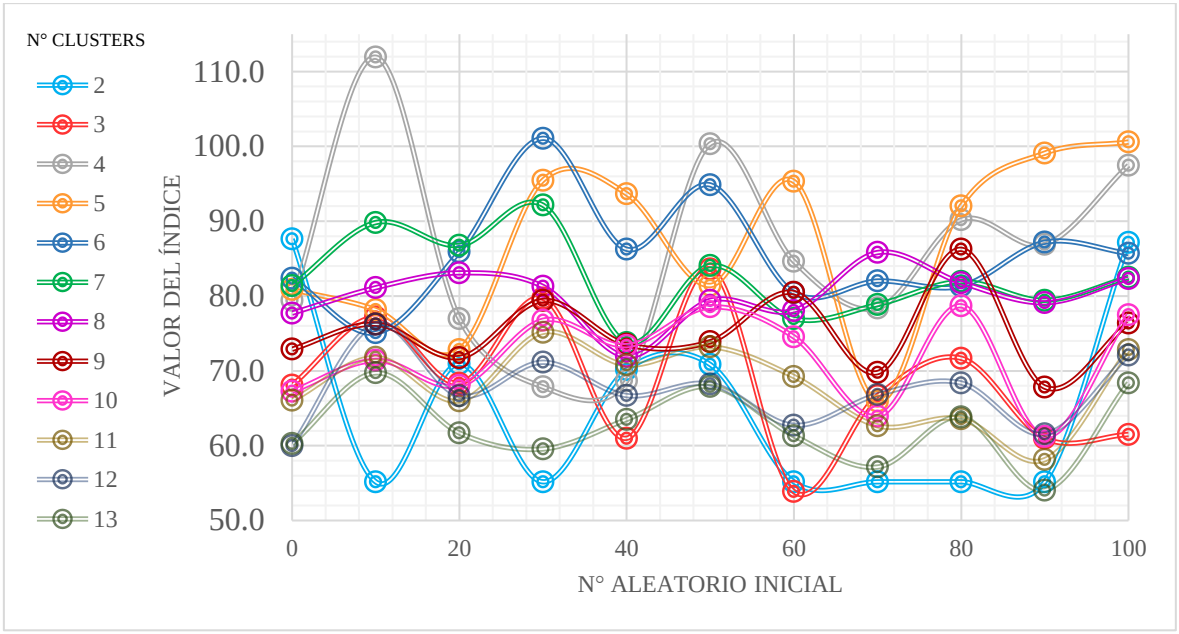


Figura 17: Determinar el número de clusters con el índice de Calinski Harabaz

Se observa en el Tabla N° 10 y la Figura N° 17 que al aplicar el algoritmo clustering K-prototype los valores del índice de validación interna de Calinski Harabasz óptimos fueron 111.9, 101.1 y 100.6, por lo que el número clusters posibles fueron K=4, K=6 o K=5, con la semilla inicial seed=10, seed=30 o seed=100 respectivamente, visualizando la variabilidad en sus resultados por la naturaleza del algoritmo.

C. Aplicación del algoritmo K- prototype

Se determinó que el número de cluster óptimo dado los índices de validación interna desarrollados, por mayoría, para el algoritmo clustering K- prototype fue K=5 y con la semilla inicial seed=90.

D. Obtención de la matriz de pertenencia

Al aplicar el algoritmo, iterativamente se asignó a cada ingresante un cluster con observaciones similares y se determinó los centros de los conglomerados o prototypes, los cuales caracterizan al cluster al cual pertenece.

Tabla 11: Tabla de distribución K prototypes

CLUSTER	FRECUENCIA	PORCENTAJE
1	175	25%
2	140	20%
3	71	10%
4	190	28%
5	114	17%
Total	690	100%

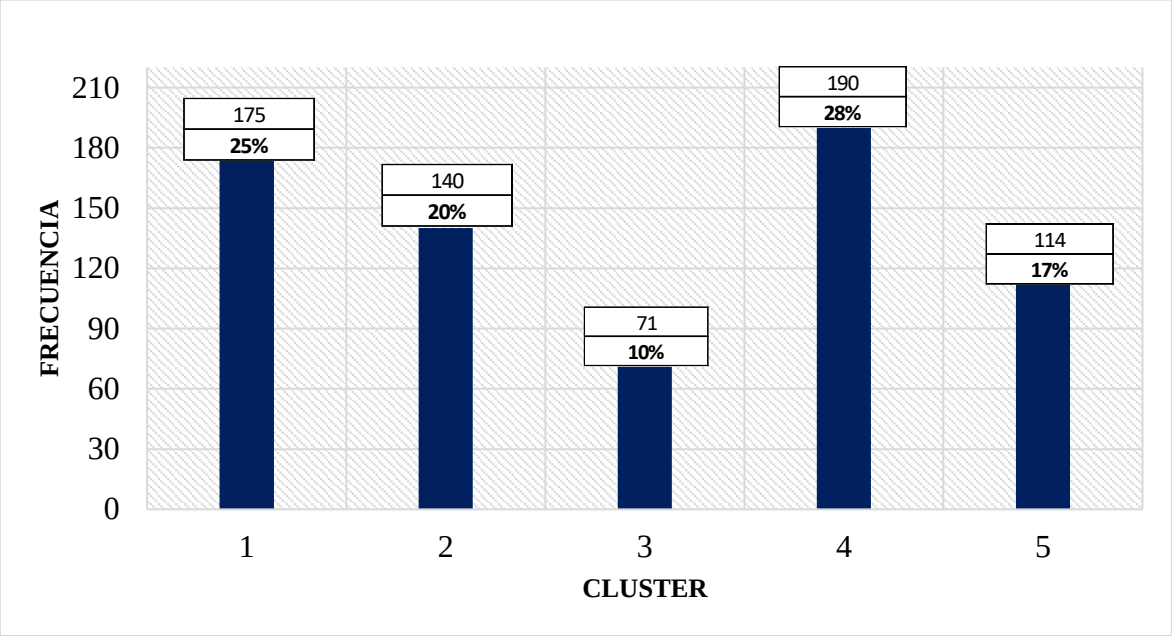


Figura 18: Figura de distribución K prototypes

La mayor cantidad de ingresantes fueron agrupados en el cluster 4 (190 individuos) que representa el 28% de los datos, mientras que el cluster 3 tiene la menor cantidad de observaciones (71 individuos) representando el 10% de los datos.

E. Cálculo de índices de validación interna clustering

En resumen, dado el número de clusters (K=5) los índices de validación interna resultaron:

Tabla 12: Índices de validación interna K prototypes

ÍNDICE	VALOR DEL ÍNDICE
Davies-Bouldin	1.826
Dunn	0.149
Calinski Harabasz	100.587

4.6. Comparación de algoritmos

Tabla 13: Comparación de índice de validación interna

ALGORITMO	K	DAVIES BOULDIN	DUNN	CALINSKI HARABASZ
K medoids	3	1.798	0.156	240.152
K prototype	5	1.826	0.149	100.587

Dado los resultados obtenidos, el algoritmo que permite tener la mejor agrupación de los ingresantes 2015, es el K medoids, ya que tuvo un índice de Davies Boudin menor e índice de Dunn y de Calinski Harabasz mayores que el del algoritmo K prototype, a partir de ello se procedió a determinar los 3 perfiles de los ingresantes.

4.7. Centros de los clusters formados

Tabla 14: Valores de los centros

CLUSTER	Años_Colegio_ Admisión	Edad_ Admisión	Aporte_ Semestral	CTA_Colegio	COM_Colegio
1	1.16	18.31	340	14	16
2	2.16	18.69	60	17	17
3	4.17	20.01	60	12	13

CLUSTER	MAT_ Colegio	Nota_ Colegio	RM_ Admisión	RV_ Admisión	MAT_ Admisión	FIS_ Admisión
1	15	15.27	13.92	10.00	12.7	14.64
2	15	16.45	12.14	10.75	12.08	11.07
3	14	13.64	8.92	8.00	12.91	13.21

CLUSTER	QUI_ Admisión	BIO_ Admisión	Nota_ Admisión	DEPT_ Colegio	Sexo	Tipo_ Colegio
1	16.42	10.00	12.95	Lima	M	Privada
2	15.35	9.28	11.78	Lima	F	Pública
3	13.57	8.21	10.80	Lima	M	Pública

CLUSTER	Tercio_ Superior_ ESP	Modalidad	ESPECIALIDAD	ELECCION_ESP_ INGRESO
1	Si	Concurso Ordinario	Agronomía	Primera
2	No	Concurso Ordinario	Agronomía	Tercera
3	No	Concurso Ordinario	Ingeniería Agrícola	Segunda

Estos son los centros que caracterizan a cada segmento formado, se puede observar que el cluster 1 agrupa a los estudiantes que pertenecen al tercio superior de sus especialidades a las que ingresaron, más detalles de la caracterización de cada segmento se dará más adelante, después de revisar con detalle las variables seleccionadas frente a la agrupación obtenida.

4.8. Validación del agrupamiento

1. Enfoque Univariado

Con la finalidad de tener un análisis univariado previo a la validación de los segmentos de los ingresantes, se propuso usar dos procedimientos: el análisis de variancia o ANOVA (para las variables cuantitativas) considerando como factor la clasificación de los clusters (los 3 clusters pasan a ser los niveles del factor) y prueba Chi cuadrado (para las variables cualitativas). Los resultados de cada una de las pruebas se encuentran en el anexo 2.

En resumen:

Tabla 15: ANOVA para variables cuantitativas

VARIABLE	P VALUE	SIG
Años_Colegio_Admisión	6.05E-63	***
Edad_Admisión	8.53E-62	***
Aporte_Semestral	1.53E-41	***
CTA_Colegio	6.80E-69	***
COM_Colegio	1.95E-68	***
MAT_Colegio	2.07E-60	***
Nota_Colegio	5.21E-88	***
RM_Admisión	1.01E-19	***
RV_Admisión	7.40E-05	***
MAT_Admisión	4.02E-13	***
FIS_Admisión	1.41E-32	***
QUI_Admisión	2.03E-14	***
BIO_Admisión	4.08E-06	***
Nota_Admisión	6.22E-42	***

Las pruebas resultaron altamente significativa, es decir, existe evidencia estadística para rechazar la hipótesis de la igualdad de medias, lo que indicó que la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que todas las variables cuantitativas aportan significativamente de manera univariada para obtener el perfil del ingresante de la UNALM.

Tabla 16: Prueba Chi cuadrada entre las variables cualitativas y los clusters

VARIABLE	PROB	SIG
Dept_Colegio	7.40E-01	
Sexo	1.99E-29	***
Tipo_Colegio	4.01E-33	***
Tercio_Superior_ESP	4.39E-25	***
Modalidad	2.90E-19	***
Especialidad	1.35E-10	***
Elección_ESP_Ingreso	2.74E-13	***

Las pruebas resultaron altamente significativas para las variables: Sexo, Tipo_Colegio, Tercio_Superior_ESP, Modalidad, Especialidad y Elección_ESP_Ingreso, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas (clusters) y cada una de las variables cualitativas mencionadas existe una relación; por lo que estas aportan significativamente para obtener el perfil del ingresante de la UNALM.

2. Enfoque Multivariado

Con la finalidad de validar el agrupamiento de los ingresantes de manera multivariada, se propuso usar algoritmos de clasificación de aprendizaje supervisado, para ello se consideró como variable dependiente a los clusters, se aplicaron el algoritmo Boruta para la selección de variables y el árbol C5.0 con poda para la caracterización de cada perfil. Los resultados de cada una de las pruebas se encuentran en los anexos 3.

a. Ajuste de parámetros

Como paso previo para la selección de variables se determinó el valor ajustado del número de variables (*mtry*) que se muestrean aleatoriamente como candidatos en cada división o nodo. Para ello, para cada valor de *mtry* se entrenó un modelo con el algoritmo Random Forest, combinando 500 árboles de decisión y se consideró una submuestra de atributos aleatorios candidatos en cada ramificación de los árboles; para establecer el valor óptimo de este parámetro, se probó diferentes valores iterativamente en el modelo y se calculó la capacidad de predicción de cada modelo con validación cruzada (propio del algoritmo).

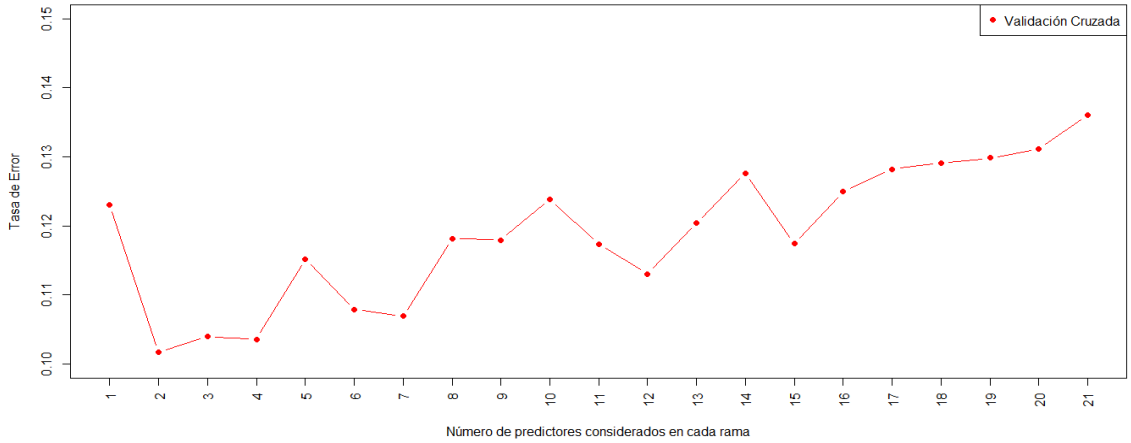


Figura 19: Cantidad de predictores por rama con todas las variables

Como se aprecia gráficamente en la Figura 19, a medida que aumenta la cantidad de predictores considerados en cada ramificación de los árboles, el error estimado obtenido crece por validación cruzada, frente a ello se consideró 2 como valor óptimo, ya que a partir de este valor se observa un valor pequeño en la tasa de error por validación cruzada.

b. Selección de variables

El algoritmo Boruta permite seleccionar los predictores más importantes, en este caso se utilizó como criterio: la importancia de la permutación o disminución media de la precisión, combinando 500 árboles de decisión una submuestra de 2 atributos aleatorios candidatos en cada ramificación de los árboles.

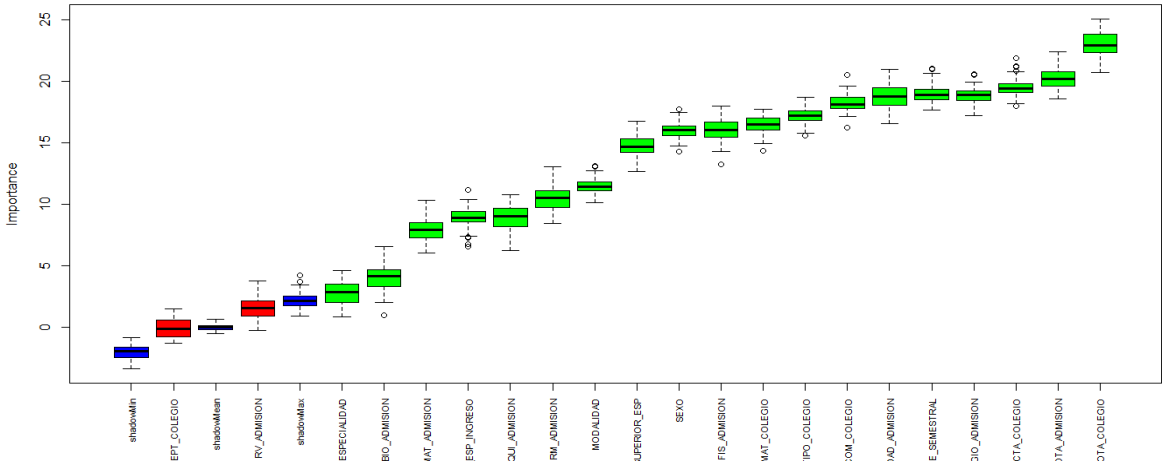


Figura 20: Importancia de variables

Tabla 17: Importancia de variables

VARIABLE	MEAN IMP	MEDIAN IMP	MIN IMP	MAX IMP	NORM HITS	DECISION
Nota_Colegio	23.03	22.96	20.74	25.10	1.00	Confirmed
Nota_Admisión	20.22	20.20	18.59	22.42	1.00	Confirmed
CTA_Colegio	19.51	19.43	17.98	21.88	1.00	Confirmed
Años_Colegio_Admisión	18.85	18.93	17.21	20.62	1.00	Confirmed
Aporte_Semestral	18.98	18.88	17.67	21.05	1.00	Confirmed
Edad_Admisión	18.80	18.76	16.59	20.99	1.00	Confirmed
COM_Colegio	18.25	18.12	16.26	20.51	1.00	Confirmed
Tipo_Colegio	17.19	17.19	15.60	18.71	1.00	Confirmed
MAT_Colegio	16.46	16.51	14.38	17.72	1.00	Confirmed
FIS_Admisión	16.08	16.07	13.25	17.98	1.00	Confirmed
Sexo	16.05	16.03	14.32	17.75	1.00	Confirmed
Tercio_Superior_ESP	14.75	14.70	12.70	16.75	1.00	Confirmed
Modalidad	11.47	11.46	10.15	13.10	1.00	Confirmed
RM_Admisión	10.46	10.50	8.42	13.03	1.00	Confirmed
QUI_Admisión	8.88	9.01	6.26	10.81	1.00	Confirmed
Elección_ESP_Ingreso	8.86	8.89	6.54	11.16	1.00	Confirmed
MAT_Admisión	7.85	7.91	6.05	10.36	1.00	Confirmed
BIO_Admisión	4.03	4.15	0.97	6.53	0.90	Confirmed
Especialidad	2.75	2.84	0.83	4.62	0.70	Confirmed
RV_Admisión	1.55	1.55	-0.26	3.77	0.25	Rejected
Dept_Colegio	-0.05	-0.13	-1.31	1.50	0.01	Rejected

En la Figura 20 y la tabla 17, se observa, de 21 atributos, 2 de ellos son rechazados (cajas rojas) y 19 confirmados (cajas azules), ya que el valor mediano de su importancia no supera a la mediana de las importancias máximas obtenido por los atributos sombra.

c. Selección de variables

Las variables que ingresaron al modelo final fueron: Nota_Colegio, Nota_Admisión, CTA_Colegio, Años_Colegio_Admisón, Aporte_Semestral, Edad_Admisión, COM_Colegio, Tipo_Colegio, MAT_Colegio, FIS_Admisión, Sexo, Tercio_Superior_ESP, Modalidad, RM_Admisión, QUI_Admisión, Elección_ESP_Ingreso, MAT_Admisión, BIO_Admisión y Especialidad. No ingresaron al modelo RV_Admisión ni Dept_Colegio.

d. Evaluación del rendimiento del modelo

Finalmente, con las variables seleccionadas, se calcula el error de validación cruzada en un algoritmo de Random Forest actualizando el valor ajustado de parámetro *mtry* y con ello estimar el rendimiento del modelo.

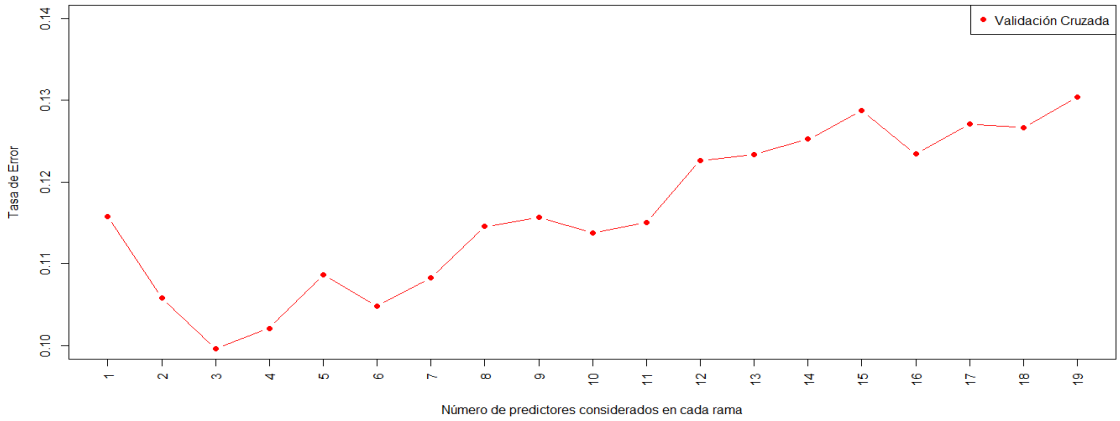


Figura 21: Cantidad de predictores por rama con las variables importantes

Tabla 18: Matriz de confusión considerando a los cluster como variable dependiente

OBSERVADO	PRONÓSTICO			
	CLUSTER 1	CLUSTER 2	CLUSTER 3	% TOTAL
Cluster 1	227	13	9	36.09%
Cluster 2	17	267	7	42.17%
Cluster 3	7	8	135	21.74%
% Acierto	90.44%	92.71%	89.40%	

OOB estimate of error rate:	**9.96%**

En la tabla 18 se presenta la matriz de confusión, resultado de aplicar el modelo Random Forest a los datos, se aprecia que el clasificador obtiene un 91.16% de correcta clasificación, sin embargo, esta medida nos puede dar una idea errónea del nivel de predicción, ya que con estos mismos datos se entrenó el modelo, frente a ello se utilizó el error de validación cruzada que el algoritmo internamente calculó, este fue de 9.96%, esto nos indica que el modelo tiene un nivel de acierto estimado de 90.04%, lo cual indica que los clusters obtenidos son consistentes con los datos analizados y las variables independientes permiten describir las agrupaciones obtenidas (clusters).

e. Obtención de las reglas de clasificación

Dado que el algoritmo Random Forest no facilita la interpretación de sus resultados por la cantidad de árboles que genera internamente, se optó por entrenar un modelo, usando las variables previamente seleccionadas; C5.0 basado en reglas, el cual descompone la estructura de árbol en un conjunto de reglas mutuamente excluyentes. Estas reglas se podaron considerando el número mínimo de registros que deben existir en un nodo y el parámetro de complejidad CP. Para estables estos parámetros, se probó diferentes valores iterativamente en el árbol C5.0 y se probó la capacidad de predicción en la data test (10% de los datos), con minCases: 2, 5, 10, 15, 20 o 25 y CP: 0.10, 0.25, 0.40, 0.55, 0.70 o 0.85; generándose 36 combinaciones y se seleccionó aquellos valores que permitieron obtener el menor error de predicción.

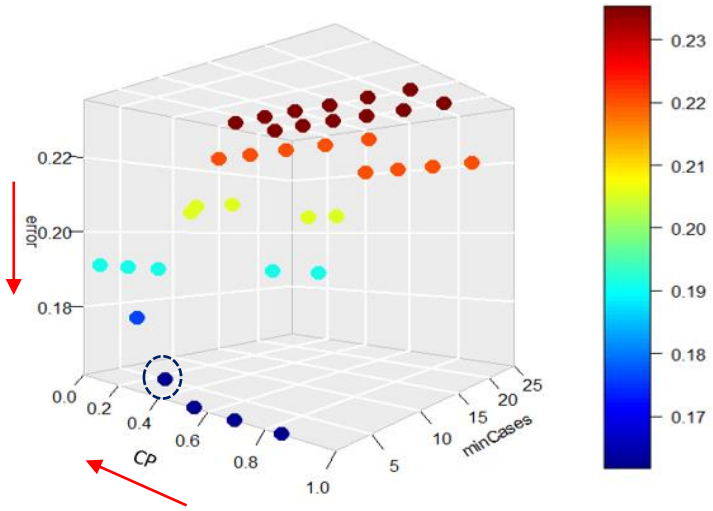


Figura 22: Diagrama de dispersión de los modelos generados

Finalmente, los parámetros para el modelo fueron: minCases = 5 y CP =0.25, dados los parámetros ajustados, se procedió a entrenar un modelo de árbol C5.0 con el total de datos y obtener las reglas que permitan perfilar al ingresante.

4.9. Caracterización del perfil del ingresante

El propósito final de la presente investigación, fue caracterizar o tipificar los segmentos de ingresantes 2015 de la UNALM, se identificó las características que diferencian un cluster de otro y las similitudes de los ingresantes que pertenecen al mismo segmento, con esta información, se busca que los profesores consejeros tengan conocimiento del perfil del alumno que tiene a su cargo y empezar con ello diversas estrategias educativas, tales como las asesorías personalizadas y el seguimiento académico, los cuales sirven para la orientación constante de los estudiantes sobre los cursos ofrecidos en cada semestre, las regulaciones en los diversos aspectos de los procedimientos universitarios, la utilización de los diversos recursos de la universidad e inclusive fortalecer las debilidades y dificultades que pueden estar enfrentando en su proceso de formación universitaria desde que ingresa.

1. Resumen Univariado

Analizando los resultados obtenidos en 4.7, así como, la información que surge del cruce entre cada variable y los clusters obtenidos (anexo 3) se realizó una tabla de resumen general para caracterizar cada cluster por variable de los ingresantes 2015 de la UNALM, esto se puede visualizar en la siguiente tabla:

Tabla 19: Perfil del ingresante UNALM 2015

VARIABLE	CLUSTER		
	1	2	3
DISTRIBUCIÓN %	36.3%	42.1%	21.6%
Años_Colegio_Admisión	▼	▬	▲
Edad_Admisión	▼	▬	▲
Aporte_Semestral	▲	▬	▼
CTA_Colegio	▬	▲	▼
COM_Colegio	▬	▲	▼
MAT_Colegio	▬	▲	▼
Nota_Colegio	▬	▲	▼
RM_Admisión	▲	▬	▼
RV_Admisión	▲	▬	▼
MAT_Admisión	▲	▼	▬
FIS_Admisión	▲	▼	▬
QUI_Admisión	▲	▼	▬
BIO_Admisión	▲	▼	▬
Nota_Admisión	▲	▼	▬
DEPT_Colegio	Lima y Provincias	Lima y Provincias	Lima
Sexo	Masculino	Femenino	Masculino
Tipo_Colegio	Privada	Privada y Pública	Pública
Tercio_Superior_ESP	Si	No	No
Modalidad	Concurso Ordinario	Concurso Ordinario y Dos Primeros Puestos de Colegios de Educación	Concurso Ordinario
Especialidad	Agronomía Biología Cencias Forestales Gestión Empresarial Industrias Alimentarias Ingeniería Agrícola Ingeniería Ambiental Meteorología	Agronomía Economía Estadística Informática Gestión Empresarial Industrias Alimentarias Ingeniería Agrícola Pesquería Zootecnia	Economía Estadística Informática Ingeniería Agrícola Pesquería Zootecnia
Elección_ESP_Ingreso	Primera Opción	Segunda y Tercera Opción	Segunda y Tercera Opción

En la tabla N° 19, se observó que el cluster con mayor porcentaje de alumnos ingresantes fue el 2 con 42.1%, los cuales venían con alto rendimiento académico del colegio, pero tuvieron un rendimiento bajo en el examen de admisión, se encontró un gran porcentaje de los alumnos que ingresaron por modalidad Dos Primeros Puestos de Colegios de Educación Secundaria (94.2%) y estudiantes que ingresaron a la carrera que escogen como segunda o tercera opción, por otro lado los alumnos agrupados en el cluster 1 fueron alumnos que tenían

un regular desempeño académico en el colegio, pero en el examen de admisión mostraron un alto rendimiento, hay un gran porcentaje de los alumnos que pertenecieron al tercio superior de su especialidad (64.5%) e ingresaron a la carrera que escogieron como primera opción; finalmente el cluster 3 posee el menor porcentaje de ingresantes (21.6%), estos alumnos por lo general tenían un bajo desempeño académico en el colegio y mostraron un rendimiento bajo en algunas áreas en el examen de admisión como RM y RV y notas regulares en las demás.

2. Resumen Multivariado

Dado los resultados obtenidos en 7.8 se realizó una tabla de resumen con las reglas de decisión específicas obtenidas del algoritmo de clasificación C5.0 con poda, que caracterizan a cada cluster de ingresantes 2015 de la UNALM, esto se puede ver en la siguiente tabla:

Tabla 20: Reglas de decisión del algoritmo de clasificación C5.0 con poda

N°	CLUSTER		
	1	2	3
1	Nota_Colegio > 15.36 Nota_Colegio <= 16.1 Nota_Admisión > 10.88 Años_Colegio_Admisión <= 3.17 Tipo_Colegio = Privada Sexo = M	Nota_Colegio > 15.36 Nota_Admisión <= 12.38 Años_Colegio_Admisión <= 3.17 Sexo = F Tercio_Superior_ESP = No	Nota_Colegio <= 15.36 Nota_Admisión <= 11.65 CTA_Colegio <= 13 Aporte_Semestral <= 200
2	Nota_Admisión > 12.38 Años_Colegio_ Admisión <= 3.17 Tipo_Colegio = Privada FIS_Admisión > 15.71	Nota_Colegio > 16.55 FIS_Admisión <= 15.71 Tercio_Superior_ESP = NO	Nota_Colegio <= 14.59 Edad_Admisión > 19.75
3	Nota_Colegio > 15.36 CTA_Colegio <= 17 Años_Colegio_Admisión <= 3.17 Tipo_Colegio = Privada Tercio_Superior_ESP = Si	Nota_Colegio > 16.1 Nota_Admisión <= 12.38 Sexo = M QUI_Admisión <= 15	Nota_Colegio <= 13.64 Nota_Admisión <= 12.26
4	Nota_Colegio > 13.64 Nota_Colegio <= 15.36 Aporte_Semestral > 200 Edad_Admisión <= 19.75	Nota_Colegio > 13.64 Aporte_Semestral <= 200 Edad_Admisión <= 19.75 FIS_Admisión <= 12.14	CTA_Colegio <= 17 Años_Colegio_Admisión > 3.17 Tercio_Superior_ESP = NO MAT_Admisión > 12.5
5	Nota_Colegio > 15.36 Nota_Admisión > 12.38 Tipo_Colegio = PRIVADA Tercio_Superior_ESP = SI	Nota_Colegio > 14.59 Edad_Admisión > 19.75 Edad_Admisión <= 21.09 Sexo = F	Nota_Colegio <= 15.36
6	Nota_Colegio <= 15.36 Nota_Admisión > 12.26 Edad_Admisión <= 19.75	Nota_Colegio > 15.36	
7	Nota_Colegio > 14.59 Nota_Colegio <= 15.36 Edad_Admisión <= 21.09 Sexo = M RM_Admisión > 12.86		
8	Nota_Admisión > 10.88 CTA_Colegio <= 17 Años_Colegio_Admision <= 3.17 Tipo_Colegio = Privada		

En la tabla N° 20, se observó que el cluster 1 agrupa a los ingresantes que tuvieron una mayor nota en el examen de admisión, por lo general mayor a 12.3, en el cluster 2 se encuentran los ingresantes que obtuvieron notas menores a 12.3 y en el clúster 3 están los ingresantes con notas muy bajas menores a 11.7, esto nos indicó que los clusters con estudiantes que tuvieron rendimientos bajo en el examen de admisión se encontraría en el cluster 2 y 3.
Respecto a las notas promedio que obtuvieron en sus respectivos colegios, los ingresantes del cluster 1 agrupó aquellos que tuvieron notas regulares entre 14.6 y 16.1, en el cluster 2 se encuentran los ingresantes con notas altas, por lo general mayores a 16.5; sin embargo, en el cluster 3 tiene alumnos que en su colegio obtuvieron notas promedio menores a 13.6.

En cuanto al tiempo que pasó entre el termino de sus estudios en colegio y el ingreso a la universidad, los cluster 1 y 2 contienen estudiantes que ingresaron a la universidad poco o regular tiempo después de terminar la secundaria aproximadamente menos de 3 años, mientras que el cluster 3 contiene los estudiantes que tomaron más tiempo. Por lo general los clusters 1 y 3 tiene estudiantes varones, muestras que el cluster 2 contiene mujeres. Finalmente, el cluster 1 agrupó a los ingresantes que pertenecieron al tercio superior, frente a los cluster 2 y 3 cuyos alumnos no pertenecen al tercio superior.

En resumen, la caracterización de los perfiles de los ingresantes se detalla a continuación:

CLUSTER	DESIGNACIÓN	CARACTERIZACIÓN
1 (36.1%)	Ingresante previsto	Estos estudiantes se caracterizan por evidenciar conocimientos previstos o esperados al ingresar a la universidad ya que en su mayoría mostraron tener un alto rendimiento en el examen de admisión con desempeño académico medio en el colegio, adicionalmente a esto, los estudiantes que tienen este perfil en su mayoría ocuparon el tercio superior en su carrera e ingresaron a la especialidad que eligieron como su primera opción por la modalidad Concurso Ordinario. Por lo general son varones que terminaron sus estudios en un colegio privado e ingresaron poco tiempo después de terminar la secundaria (< 3.2 años), se les fue asignado un aporte semestral mayor al promedio dada su situación socioeconómica. Las carreras a las que ingresaron

		fueron: Agronomía, Biología, Ciencias Forestales, Gestión Empresarial, Industrias Alimentarias, Ingeniería Agrícola, Ingeniería Ambiental y Meteorología.
2 (42.2%)	Ingresante en proceso	Estos estudiantes se caracterizan por estar en camino a lograr conocimientos previstos o esperados al ingresar a la universidad por lo cual requieren acompañamiento durante un tiempo razonable para alcanzarlo, en su mayoría mostraron tener un desempeño académico muy bueno en el colegio pero no suficiente para afrontar el examen de admisión ya que alcanzaron un rendimiento entre regular y bajo en este, adicionalmente a esto, los estudiantes que tienen este perfil en su mayoría no ocuparon el tercio superior en su carrera e ingresaron a la especialidad que eligieron como su segunda o tercera opción por la modalidad Concurso Ordinario y Dos Primeros Puestos de Colegios de Educación Secundaria. Por lo general son mujeres que terminaron sus estudios en un colegio privado o público e ingresaron después de un periodo regular de tiempo de terminar la secundaria (< 3.2 años), se les fue asignado un aporte semestral igual al promedio dada su situación socioeconómica. Las carreras a las que ingresaron fueron: Agronomía, Economía, Estadística Informática, Gestión Empresarial, Industrias Alimentarias, Ingeniería Agrícola, Pesquería y Zootecnia.
3 (21.7%)	Ingresante en inicio	Estos estudiantes se caracterizan por estar empezando a desarrollar conocimientos previstos o esperados al ingresar a la universidad por lo cual necesita mayor tiempo de acompañamiento e intervención del consejero de acuerdo con su ritmo y estilo de aprendizaje para alcanzarlo, en su mayoría mostraron tener un desempeño académico malo en el colegio y alcanzaron un rendimiento entre regular y bajo en el examen de admisión, adicionalmente a esto, los estudiantes que tienen este perfil en su mayoría no ocuparon el tercio superior en su carrera e ingresaron a la especialidad que eligieron como su

	segunda o tercera opción por la modalidad Concurso Ordinario. Por lo general son varones que terminaron sus estudios en un colegio público e ingresaron después de un periodo largo de tiempo de terminar la secundaria (> 3.2 años), se les fue asignado un aporte semestral menor al promedio dada su situación socioeconómica. Las carreras a las que ingresaron fueron: Economía, Estadística Informática, Ingeniería Agrícola, Pesquería y Zootecnia.

4.10. Análisis de resultados

Frente a los resultados obtenidos del análisis univariado y multivariado, se puedo observar que los ingresantes que fueron agrupados en los clusters 2 y 3 son los que tienen características importantes para resaltar tales como: ambos segmentos agrupan alumnos que no pertenecieron al tercio superior al momento de dar su examen de admisión e ingresaron a su segunda o tercera opción de carrera a la que postularon, por lo que a futuro podrían recurrir a un traslado de carrera, en ambos segmentos se encontraron estudiantes que tuvieron regular o bajo rendimiento en el examen de admisión, en especial en el cluster 3, estos últimos obtuvieron bajas notas en su colegio; finalmente ambos segmentos contienen alumnos que tienen asignado aporte semestral bajo dada su condición socioeconómica.

Estas características permiten identificar y entender cuáles son los perfiles de ingresantes que deben ser atendidos por autoridades pertinentes dentro de la institución, a través de diversas estrategias educativas, apoyo económico y orientación con el fin de que a futuro no tengan bajo rendimiento académico, retraso en sus estudios, dilatación del tiempo de estudio, deserción, entre otros.

Con el fin de validar los resultados obtenidos de la segmentación se cruzó esta información con el promedio ponderado acumulado de los alumnos que obtuvieron al término de su primer año de estudios superiores, ya que en este periodo los universitarios llevan cursos generales que buscan reforzar sus conocimientos adquiridos antes de ingresar a la universidad, tales como: Matemática Básica, Química General, Lengua, Biología General, Física General, entre otros.

Para el análisis se clasificó el promedio ponderado acumulado como:

- EXCELENTE: notas ente 16.5 y 20
- BUENO: notas entre 12,5 y 16.5
- REGULAR: notas entre 10,5 y 12.5
- MALO: notas entre 0 y 10.5

Tabla 21: Cruce clusters frente al promedio ponderado acumulado

CLUSTER	PROMEDIO PONDERADO ACUMULADO			
	EXCELENTE	BUENO	REGULAR	MALO
1	57%	39%	35%	20%
2	43%	48%	41%	28%
3	0%	13%	23%	52%
Total	100%	100%	100%	100%

En la tabla 21, se observó que más de la mitad de los ingresantes que tuvieron un promedio ponderado acumulado en su primer año de estudios EXCELENTE se encuentran en el cluster 1, los alumnos BUENOS y REGULARES se encuentran en su mayoría en el cluster 2, mientras que los alumnos MALOS se encuentran agrupados en el cluster 3. Validando así lo mencionado anteriormente.

Adicionalmente se realizó un cruce de la modalidad de ingreso, promedio ponderado acumulado en su primer año de estudios y los cluster, mostrados en la tabla 22.

Tabla 22: Cruce clusters frente al promedio ponderado acumulado y la modalidad de ingreso

CLUSTER	MODALIDAD DE INGRESO							
	CONCURSO ORDINARIO (90%)				DOS PRIMEROS PUESTOS DE COLEGIOS DE EDUCACIÓN SECUNDARIA (10%)			
	EXCELENTE	BUENO	REGULAR	MALO	EXCELENTE	BUENO	REGULAR	MALO
1	75%	44%	38%	21%	33%	8%	0%	0%
2	25%	42%	36%	23%	67%	92%	100%	100%
3	0%	15%	25%	56%	0%	0%	0%	0%
Total	100%	100%	100%	100%	100%	100%	100%	100%

Se observó en cuanto a los ingresantes del Concurso Ordinario que obtuvieron en su primer año de educación superior notas MALAS, en su mayoría se encuentran en el cluster 3; mientras que los alumnos con promedios REGULARES y MALOS que ingresaron por modalidad Dos Primeros Puestos de Colegios de Educación Secundaria se agruparon en el cluster 2.

Todo esto permite entender que los ingresantes que deben ser atendidos con prioridad, son aquellos que pertenecen a los clusters 2 y 3, en especial a los alumnos del cluster 2 que ingresan por primeros puestos y estudiantes del cluster 3 que ingresaron por modalidad de Concurso Ordinario.

V. CONCLUSIONES

1. Se logró caracterizar el perfil de los ingresantes de una universidad pública respecto a sus variables socioeconómicas, demográficas y de rendimiento educativo utilizando algoritmos de segmentación.

2. El algoritmo de agrupamiento K-medoids generó segmentos con mayor validez dado que obtuvo un índice de validación interna clustering de Davies Bouldin menor y un índice de Dunn y Calinski Harabasz mayor frente al algoritmo K-prototype en el estudio de caso, permitiendo segmentar adecuadamente a los ingresantes de una universidad pública.

3. Se logró determinar que los ingresantes 2015 de la UNALM se ajustan a 3 perfiles y fueron designados como: Ingresante en inicio, Ingresante en proceso e Ingresante previsto, cada uno con características diferentes, los cuales han sido obtenidos aplicando el algoritmo de agrupamiento K- medoids; adicionalmente se identificó que las variables *ubicación del colegio donde cursó el 5to año de secundaria* (Dept_Colegio) y la *nota obtenida en el curso de RV en el examen de admisión* (RV_Admisión) no permitían diferenciar los segmentos planteados, dado los resultados obtenidos en el análisis univariado y multivariado; finalmente al aplicar el algoritmo de Random Forest, teniendo en consideración a los cluster como variable dependiente se obtuvo un 90.04% de correcta clasificación, demostrando así la validez de los segmentos propuestos.

4. Las variables: Nota promedio del último año de estudios (Nota_Colegio), nota general obtenida en el examen de admisión (Nota_Admisión), nota obtenida en el 5to año de secundaria en el área de Ciencia tecnología y Ambiente (CTA_Colegio), tiempo transcurrido desde que terminó el 5to año de secundario e ingresó a la universidad (Años_Colegio_Admisión), aporte semestral asignado al ingresante (Aporte_Semestral), edad del ingresante al momento del examen de admisión

(Edad_Admisión), nota obtenida en el 5to año de secundaria en el área de Comunicación (COM_Colegio), tipo de institución de procedencia (Tipo_Colegio), nota obtenida en el 5to año de secundaria en el área de Matemática (MAT_Colegio), nota obtenida en el curso de Física en el examen de admisión (FIS_Admisión), sexo del ingresante (Sexo), si el alumno pertenece o no al tercio superior en la especialidad a la que ingresó (Tercio_Superior_ESP), modalidad de ingreso a la universidad (Modalidad), nota obtenida en el curso de RM en el examen de admisión (RM_Admisión), nota obtenida en el curso de Química en el examen de admisión (QUI_Admisión), orden de elección que tuvo la carrera a la cual ingresó (Elección_ESP_Ingreso), nota obtenida en el área de Matemática en el examen de admisión (MAT_Admisión), nota obtenida en el curso de Biología en el examen de admisión (BIO_Admisión) y especialidad a la que ingresó (Especialidad), son importantes para determinar el perfil del ingresante, esto se pudo determinar al aplicar el algoritmo Boruta.

5. Los perfiles obtenidos con el algoritmo clustering son válidos, ya que al cruzarlos con la variable pasiva: *promedio ponderado acumulado del primer año en la universidad de los ingresantes en estudio*, se comprobó que los estudiantes que alcanzaron un promedio ponderado malo en su mayoría pertenecen al cluster 3 denominados Ingresantes en inicio, además los ingresantes que alcanzaron una nota entre regular y buena en su mayoría pertenecen al cluster 2 denominados Ingresantes en proceso, finalmente los estudiantes que lograron un promedio ponderado acumulado excelente están agrupados en el cluster 1 denominados Ingresantes previstos, el cual puede ser considerado el perfil de ingreso deseado.

6. La especialidad que tuvo mayor diversidad de perfiles de ingresantes fue: Ingeniería Agrícola, mientras que las carreras con menor variabilidad de perfiles fueron: Biología, Ciencias Forestales, Ingeniería Ambiental y Meteorología.

7. Las carreras que tienen al menos el 25% de sus ingresantes segmentados como Ingresantes en inicio son: Ingeniería Agrícola, Economía, Zootecnia, Pesquería e Ingeniería Estadística Informática, por lo cual, estas deben ser las primeras

especialidades en promover políticas educativas como el emprendimiento del acompañamiento especializado, sistemático e integral con sus estudiantes.

VI. RECOMENDACIONES

1. Desarrollar e implementar estrategias de asesoramiento personalizado, según la caracterización encontrada para cada uno de los 3 perfiles.

2. Los alumnos que se ajustan al perfil Ingresante en inicio, son alumnos que deben tener un seguimiento continuo y ser asesorados periódicamente ya que este perfil se caracteriza por tener ingresantes con bajo rendimiento académico en el colegio y en el examen de admisión, adicionalmente podrían recibir apoyos socio económicos tales como becas de alimentos, estudios y deportes que los incentive a mejor su rendimiento académico y evitar a futuro retraso en sus estudios, dilatación del tiempo de estudio, deserción, entre otros.

3. Esta investigación se enfocó en obtener el perfil del ingresante del año 2015, no obstante, esto también puede desarrollarse para ingresantes de años venideros, sin embargo, para lograr ello de manera eficiente es necesario que los datos se encuentren almacenados en bases de datos relacionales donde se puedan extraer de manera rápida y sencilla conectando información de distintos orígenes, esto se debe a que actualmente cada oficina dentro de la universidad tiene su propio formato y fuente de almacenamiento e incluso aún se registra información importante de los estudiantes en fuentes físicas (papel) que se deterioran al pasar el tiempo.

4. El análisis realizado puede ser utilizado para próximas promociones de estudiantes que ingresan y lograr clasificarlos para determinar el perfil que tengan y a partir de ello recibir asesoramiento personalizado desde su ingreso a la universidad, esto se puede lograr dado que se entrenó un modelo de clasificación (Random Forest).

5. Variables como: número de veces que postuló anteriormente o si ha estudiado previamente en otra institución superior, entre otras; pueden ser variables que permitan obtener mejores perfiles con mayores características relevantes.

VII. BIBLIOGRAFÍA

Amat, J. 2017. Árboles de predicción: bagging, random forest, boosting y C5.0 (en línea, sitio web). s. l. Disponible en https://bit.ly/2QCqbP1

Aquino, B. 2017. Educación universitaria: hay 30 % de deserción por falta de orientación y de recursos (en línea, sitio web). Andina. Lima, Perú. Disponible en https://bit.ly/3a1UZjQ.

Arias, J. 2015. El perfil de ingreso en el rendimiento académico inicial de los estudiantes de la carrera de Agronomía de la Universidad Nacional Agraria La Molina, años 2011 a 2012. Tesis Dr. Ciencias de la Educación. Lima, Perú (en línea). Universidad Nacional de Educación Enrique Guzmán y Valle. Disponible en https://bit.ly/2tD67CV.

Arora, P; Virmani, D; Varshney, S. 2016. Analysis of K-Means and K-Medoids Algorithm for Big Data. Nagpur, India (en línea). Procedia Computer Science 78: 507-512. Disponible en https://bit.ly/2s5X9xy.

B. Kursa, M; R. Rudnicki, W. 2010. Feature Selection with the Boruta Package. Varsovia, Polonia (en línea). Journal of Statistical Software 36(11): 1-11. Disponible en https://bit.ly/2FBA7la.

Beca, S. 2007. Clustering difuso con selección de atributos. Tesis Mg. Gestión de Operaciones. Santiago de Chile, Chile (en línea). Universidad de Chile. Disponible en https://bit.ly/303eAeI.

Bedalli, E; Ninka, I. 2015. Exploring an Educational System's Data through Fuzzy Cluster Analysis (en línea). Athens Journal of Sciences 2(1): 33-44. s. l. Disponible en https://bit.ly/2R5szge.

Bhat, A. 2014. K-medoids clustering using partitioning around medoids for performing face recognition (en línea). International Journal of Soft Computing, Mathematics and Control 3(3): 1-11. s. l. Disponible en https://bit.ly/36QMfer

Breiman, L. 2001. Random Forest (en línea). California, Estados Unidos. Universidad de California. Disponible en https://bit.ly/373dSAI

Calinski, T; Harabasz, J. 1974. A dendrite method for cluster analysis. Poznan, Polonia (en línea). Communications in Statistics - Theory and Methods 3(1): 1-27. Disponible en https://bit.ly/2ssrbvB.

Chun, HL. 2012. Diseño e Implementación de algoritmos aproximados de clustering balanceado en PSO. Tesis Mg. Ciencias en Computación. Santiago de Chile, Chile (en línea). Universidad de Chile. Disponible en https://bit.ly/35M1y6x.

Eckert, K; Suénaga, R. 2013. Aplicación de técnicas de Minería de datos al análisis de situación y comportamiento académico de alumnos de la UGD (en línea). *In* XV Worshop de Investigadores en Ciencias de la Computación. Misiones, Argentina. Universidad Gastón Dachary. Disponible en https://bit.ly/2QSvppC.

Gupta, B; Rawat, A; Jain, A; Arora, A; Dhami, N. 2017. Analysis of Various Decision Tree Algorithms for Classification in Data Mining. Uttarakhand, India (en línea). International Journal of Computer Applications 163(8): 15-19. Disponible en https://bit.ly/2u0ldSY.

Hartigan, J; Wong, M. 1979. Algorithm AS 136: A k-means clustering algorithm. Ota, Nigeria (en línea). Journal of the Royal Statistical Society 28(1): 100-108. Disponible en https://bit.ly/30jLpV1.

Huang, Z. 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Melbourne, Australia (en línea). Data Mining and Knowledge Discovery 2: 283 - 304. Disponible en https://bit.ly/2FMUgoH.

INEI (Instituto Nacional de Estadística e Informática). 2011. II Censo Nacional universitario 2010 (en línea). Lima, Perú. Disponible en https://bit.ly/2ThDGFv (minedu.gob.pe).

Kaur, A; Kaur, N. 2013. Survey Paper on Clustering Techniques. Punjab, India (en línea). International Journal of Science, Engineering and Technology Research (IJSETR) 2(4): 803-806. Disponible en https://bit.ly/3a3Z72B.

Kuhn, M; Johnson, K. 2013. Applied Predictive Modeling. 1 ed (en línea). New York, Estados Unidos. p. 392-399. Disponible en https://bit.ly/2RfVLBe

Maimon, O; Rokach, L. 2010. Data mining and knowledge discovery handbook. 2 ed. (en línea). New York, Estados Unidos. p. 149-170. Disponible en https://bit.ly/2tZYdUi.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. 1 ed (en línea). Los Ángeles, Estados Unidos. p. 281-297. Disponible en https://bit.ly/384ZY1g.

Ochoa, L; Leticia, M. 2016. Estudio comparativo de Técnicas de supervisadas de Minería de Datos para Segmentación de Alumnos. Tesis Especialidad en Sistemas y Tecnologías de la Información. Arequipa, Perú (en línea). Universidad Católica de Santa María. Disponible en https://bit.ly/2RiZeif.

Oyelade, O; Oladipupo, O; Obagbuwa, I. 2010. Application of k-Means Clustering algorithm for prediction of Students' Academic Performance. Ota, Nigeria (en línea). International Journal of Computer Science and Information Security 7(1): 292-295. Disponible en https://bit.ly/3873sjJ.

Pandya, R; Pandya, J. 2015. C5. 0 Algorithm to Improved Decision Tree with Feature Selection and Reduced Error Pruning. s.l. (en línea). International Journal of Computer Applications 117(16): 18-21. Disponible en https://bit.ly/35PBp6R.

Patel, N; Upadhyay, S. 2012. Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA. Gujrat, India (en línea). International Journal of Computer Applications 60(12): 20-25. Disponible en https://bit.ly/2RfCBeA.

R Patel, B; K Rana, K. 2014. A Survey on Decision Tree Algorithm for Classification. Modasa, India (en línea). International Journal of Engineering Development and Research 2(1): 1-5. Disponible en https://bit.ly/30mWowJ.

Rai, P; Singh, S. 2010. A Survey of Clustering Techniques (en línea). International Journal of Computer Applications 7(12): 1-5. Disponible en https://bit.ly/30m6AWp.

Raulji, G. 2014. A Review on Fuzzy C-Mean Clustering Algorithm. s.l. (en línea). International Journal of Modern Trends in Engineering and Research 2(2): 751-754. Disponible en https://bit.ly/2FSxewM.

Singh, I; Sabitha, A; Bansal, A. 2016. Student performance analysis using clustering algorithm. In 6th International Conference - Cloud System and Big Data Engineering 6: 294-299. Noida, India.

Soni, N; Ganatra, A. 2012. Categorization of Several Clustering Algorithms from Different Perspective: A Review (en línea). International Journal of Advanced Research in Computer Science and Software Engineering 2(8): 63-68. Disponible en https://bit.ly/3ab5IbH.

UNALM (Universidad Nacional Agraria La Molina). 2017. Modelo Educativo UNALM (en línea). Lima, Perú. p. 26-80. Disponible en https://bit.ly/2TZdX55.

Velmurugan, T; Santhanam, T. 2011. A comparative analysis between K-medoids and fuzzy C-means clustering algorithms for statistically distributed data points (en línea). Journal of Theoretical and Applied Information Technology 27: 19-29. Disponible en https://bit.ly/3867V6o.

Wang, W; Zhang, Y. 2007. On fuzzy cluster validity indices. Fuzzy Sets and Systems. Fuzzy Sets and Systems 158(19): 2095-2117.

Zaragoza, J. 2019. Educación universitaria: hay 30% de deserción por falta de orientación y de recursos (en línea, sitio web). Andina. Lima, Perú. Disponible en https://bit.ly/36XF8AW.

VIII. ANEXOS

Anexo 1: Ejemplos de aplicación de algoritmos

Ejemplo de aplicación del algoritmo K-means

Para la ejemplificación de la aplicación del algoritmo k-means se utilizan los siguientes datos de 6 personas con 2 variables:

- PESO: peso medido en Kg de la persona
- ALTURA: dimensión vertical de la persona

	PESO	ALTURA
O1	64.3	1.68
O2	87.6	1.73
O3	59.5	1.61
O4	81.4	1.78
O5	64.9	1.57
O6	88.3	1.81

PRIMERO: transformación de variables, ya que ambas variables tienen escalas diferentes se procede a transformarlas, para ello se utilizó:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$y_{PESO} = \frac{x_{PESO} - 64.3}{88.3 - 64.3} \qquad y_{ALTURA} = \frac{x_{ALTURA} - 1.57}{1.81 - 1.57}$$

Resultando:

	PESO	ALTURA
O1	0.17	0.46
O2	0.98	0.67
O3	0.00	0.17
O4	0.76	0.88
O5	0.19	0.00
O6	1.00	1.00

SEGUNDO: definir la cantidad de agrupaciones

$$k = 2$$

TERCERO: seleccionar aleatoriamente *k* objetos del conjunto de datos, los cuales son los centroides iniciales para los grupos, en este caso se eligieron las observaciones O1 y O2.

CUARTO: Calcular la distancia de cada punto a los centroides, para ello se calculará la distancia euclidiana al cuadrado, donde:

$$d_E(P, Q) = \sum_{i=1}^{n} (p_i - q_i)^2$$

Es decir: O3 frente a O1 y O2

$$d_E(O3, O1) = (0 - 0.17)^2 + (0.17 - 0.46)^2 = 0.11$$
$$d_E(O3, O2) = (0 - 0.98)^2 + (0.17 - 0.67)^2 = 1.20$$
…

Así se forma la matriz de distancias:

	O1	O2
O3	0.11	1.20
O4	0.53	0.09
O5	0.21	1,07
O6	0.99	0.11

QUINTO**:** cada uno de los objetos restantes se asigna a su centroide más cercano, en el ejemplo los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

SEXTO: se calcula el nuevo centroide en cada grupo, el algoritmo k-means los calcula en base al promedio por variable de las observaciones dentro del cluster, es decir:

CLUSTER 1		
	PESO	ALTURA
O1	0.17	0.46
O3	0.00	0.17
O5	0.19	0.00
CENTROIDE1	0.12	0.21

CLUSTER 2		
	PESO	ALTURA
O2	0.98	0.67
O4	0.76	0.88
O6	1.00	1.00
CENTROIDE2	0.91	0.85

SEPTIMO: Probar la disimilitud de los objetos con los centroides actuales. Si se encuentra un elemento tal que su centro más cercano pertenece a otro cluster en lugar de su actual, se reasignará el elemento al otro cluster. Es decir: O1 frente a CENTROIDE1 y CENTROIDE2

$$d_E(O1, C1) = (0.17 - 0.12)^2 + (0.46 - 0.21)^2 = 0.06$$
$$d_E(O1, C2) = (0.17 - 0.91)^2 + (0.46 - 0.85)^2 = 0.71$$
…

Así se forma la matriz de distancias:

	C1	C2
O1	0.06	0.71
O2	0.95	0.04
O3	0.02	1.29
O4	0.86	0.02
O5	0.05	1.24
O6	1.40	0.03

OCTAVO**:** cada uno de los objetos restantes se asigna a su centroide más cercano, en el ejemplo, los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

En el ejemplo, la asignación de las observaciones no cambió, por lo que se termina la agrupación, si no hubiera pasado esto, se regresa al SEXTO paso, hasta lograr de manera iterativa la convergencia.

Ejemplo de aplicación del algoritmo K-modes

Para la ejemplificación de la aplicación del algoritmo K- Modes se utilizan los siguientes datos de 6 personas con 3 variables:

- SEXO: Masculino o Femenino
- ESTADO CIVIL: soltero(a) o casado(a)
- TIENE HIJOS: si o no

	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	F	S	SI
O2	M	C	NO
O3	F	C	SI
O4	M	C	NO
O5	F	S	SI
O6	M	S	NO

PRIMERO: definir la cantidad de agrupaciones

$$k = 2$$

SEGUNDO: seleccionar aleatoriamente *k* objetos del conjunto de datos, los cuales son los centroides iniciales para los grupos, en este caso se eligieron las observaciones O1 y O2.

TERCERO: Calcular la distancia de cada punto a los centroides, para ello se calculará la medida de disimilitud de coincidencia simple.

$$d_D(P, Q) = \sum_{i=1}^{m} \delta(p, q)$$

Donde:

$$\delta(p, q) = 0 \; para \; p = q \; y \; \delta(p, q) = 1 \; para \; p \neq q$$

Es decir: O3 frente a O1 y O2

$$d_D(O3, O1) = 0 + 1 + 0 = 1$$
$$d_D(O3, O2) = 1 + 0 + 1 = 2$$
$$\dots$$

Así se forma la matriz de distancias:

	O1	O2
O3	1	2
O4	3	0
O5	0	3
O6	2	1

CUARTO**:** cada uno de los objetos restantes se asigna a su centroide más cercano, en el ejemplo los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

QUINTO: se calcula el nuevo centroide en cada grupo, el algoritmo k modes los calcula en base a la moda por variable de las observaciones dentro del cluster, es decir:

CLUSTER 1			
	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	F	S	SI
O3	F	C	SI
O5	F	S	SI
C1	F	S	SI

CLUSTER 2			
	SEXO	ESTADO CIVIL	TIENE HIJOS
O2	M	C	NO
O4	M	C	NO
O6	M	S	NO
C2	M	C	NO

SEXTO: Probar la disimilitud de los objetos con los centroides actuales. Si se encuentra un elemento tal que su centro más cercano pertenece a otro cluster en lugar de su actual, se reasignará el elemento al otro cluster.

Es decir: O1 frente a CENTROIDE1 y CENTROIDE2

$$d_D(O1, C1) = 0 + 0 + 0 = 0$$
$$d_D(O1, C2) = 1 + 1 + 1 = 3$$
$$\ldots$$

Así se forma la matriz de distancias:

	C1	C2
O1	0	3
O2	3	0
O3	1	2
O4	3	0
O5	0	3
O6	2	1

SEPTIMO**:** cada uno de los objetos restantes se asigna a su centroide más cercano, en el ejemplo, los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

En el ejemplo, la asignación de las observaciones no cambió, por lo que se termina la agrupación, si no hubiera pasado esto, se regresa al SEXTO paso, hasta lograr de manera iterativa la convergencia.

Ejemplo de aplicación del algoritmo K-prototype

Para la ejemplificación de la aplicación del algoritmo K- prototype se utilizan los siguientes datos de 6 personas con 5 variables:

- PESO: peso medido den Kg de la persona
- ALTURA: dimensión vertical de la persona
- SEXO: Masculino o Femenino
- ESTADO CIVIL: soltero(a) o casado(a)
- TIENE HIJOS: si o no

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	64.3	1.68	F	S	SI
O2	87.6	1.73	M	C	NO
O3	59.5	1.61	F	C	SI
O4	81.4	1.78	M	C	NO
O5	64.9	1.57	F	S	SI
O6	88.3	1.81	M	S	NO

PRIMERO: transformación de variables cuantitativas, ya que ambas variables tienen escalas diferentes se procese a transformarlas, para ello se utilizará:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$y_{PESO} = \frac{x_{PESO} - 64.3}{88.3 - 64.3} \qquad y_{ALTURA} = \frac{x_{ALTURA} - 1.57}{1.81 - 1.57}$$

Resultando:

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	0.17	0.46	F	S	SI
O2	0.98	0.67	M	C	NO
O3	0.00	0.17	F	C	SI
O4	0.76	0.88	M	C	NO
O5	0.19	0.00	F	S	SI
O6	1.00	1.00	M	S	NO

SEGUNDO: definir la cantidad de agrupaciones

$$k = 2$$

TERCERO: seleccionar aleatoriamente k objetos del conjunto de datos, los cuales son los prototypes iniciales para los grupos, en este caso se eligieron las observaciones O1 y O2.

CUARTO: Calcular la distancia de cada punto a los prototypes, para ello se calculará la distancia personalizada:

$$d_P(P,Q) = \sum_{i=1}^{m_r}(p_i - q_i)^2 + \gamma \sum_{i=1}^{m_c}\delta(p,q)$$

Donde: $\delta(p,q) = 0\ para\ p = q\ y\ \delta(p,q) = 1\ para\ p \neq q$ y

$$\gamma = \frac{promedio\ (\sigma^2_{PESO}, \sigma^2_{ALTURA})}{promedio\ (1 - \sum_i^2 {p_{SEXO}}^2, 1 - \sum_i^2 {p_{ESTADO\ CIVIL}}^2, 1 - \sum_i^2 {p_{TIENE\ HIJOS}}^2)}$$

$$\gamma = \frac{promedio\ (0.2002667, 0.1556800)}{promedio\ (1 - \sum_i^2 {p_{SEXO}}^2, 1 - \sum_i^2 {p_{ESTADO\ CIVIL}}, 1 - \sum_i^2 {p_{TIENE\ HIJOS}})}$$

$$\gamma = \frac{0.1779733}{promedio\ (1 - (0.5^2 + 0.5^2), 1 - (0.5^2 + 0.5^2), 1 - (0.5^2 + 0.5^2))}$$

$$\gamma = \frac{0.1779733}{0.5} = 0.3559467$$

Es decir: O3 frente a O1 y O2

$$d_E(O3, O1) = (0 - 0.17)^2 + (0.17 - 0.46)^2 + 0.3559467(0 + 1 + 0) = 0.4689467$$
$$d_E(O3, O2) = (0 - 0.98)^2 + (0.17 - 0.67)^2 + 0.3559467(1 + 0 + 1) = 1.9222933$$

…

Así se forma la matriz de distancias:

	O1	O2
O3	0.47	1.92
O4	1.59	0.45
O5	0.21	2.14
O6	1.69	0.82

QUINTO**:** cada uno de los objetos restantes se asigna a su prototype más cercano, en el ejemplo los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

SEXTO: se calcula el nuevo prototypes en cada grupo, el algoritmo k prototypes los calcula en base al promedio por variable en el caso de las variables cuantitativas y la moda en las variables cualitativas, es decir:

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	0.17	0.46	F	S	SI
O3	0.00	0.17	F	C	SI
O5	0.19	0.00	F	S	SI
PROTOTYPE1	0.12	0.21	F	S	SI

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O2	0.98	0.67	M	C	NO
O4	0.76	0.88	M	C	NO
O6	1.00	1.00	M	S	NO
PROTOTYPE2	0.91	0.85	M	C	NO

SÉPTIMO: Probar la disimilitud de los objetos con los prototypes actuales. Si se encuentra un elemento tal que su centro más cercano pertenece a otro cluster en lugar del actual, se reasignará el elemento al otro cluster.

Es decir: O1 frente a PROTOTYPE1 y PROTOTYPE2

$$d_P(O1, C1) = (0.17 - 0.12)^2 + (0.46 - 0.21)^2 + 0.3559467(0 + 0 + 0) = 0.07$$
$$d_P(O1, C2) = (0.17 - 0.91)^2 + (0.46 - 0.85)^2 + 0.3559467(1 + 1 + 1) = 1.77$$
$$\dots$$

Así se forma la matriz de distancias:

	P1	P2
O1	0.07	1.77
O2	2.02	0.04
O3	0.37	2.00
O4	1.93	0.02
O5	0.05	2.31
O6	2.11	0.39

OCTAVO**:** finalmente, cada uno de los objetos restantes se asigna a su prototypes más cercano, en el ejemplo, los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

En el ejemplo, la asignación de las observaciones no cambió, por lo que se termina la agrupación, si no hubiera pasado esto, se regresa al SEXTO paso, hasta lograr de manera iterativa la convergencia.

Ejemplo de aplicación del algoritmo K-medoids

Para la ejemplificación de la aplicación del algoritmo K- medoids se utilizan los siguientes datos de 6 personas con 5 variables:

- PESO: peso medido den Kg de la persona
- ALTURA: dimensión vertical de la persona
- SEXO: Masculino o Femenino
- ESTADO CIVIL: soltero(a) o casado(a)
- TIENE HIJOS: si o no

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	64.3	1.68	F	S	SI
O2	87.6	1.73	M	C	NO
O3	59.5	1.61	F	C	SI
O4	81.4	1.78	M	C	NO
O5	64.9	1.57	F	S	SI
O6	88.3	1.81	M	S	NO

PRIMERO: transformación de variables cuantitativas, ya que ambas variables tienen escalas diferentes se procese a transformarlas, para ello se utilizará:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

$$y_{PESO} = \frac{x_{PESO} - 64.3}{88.3 - 64.3} \qquad y_{ALTURA} = \frac{x_{ALTURA} - 1.57}{1.81 - 1.57}$$

Resultando:

	PESO	ALTURA	SEXO	ESTADO CIVIL	TIENE HIJOS
O1	0.17	0.46	F	S	SI
O2	0.98	0.67	M	C	NO
O3	0.00	0.17	F	C	SI
O4	0.76	0.88	M	C	NO
O5	0.19	0.00	F	S	SI
O6	1.00	1.00	M	S	NO

SEGUNDO: definir la cantidad de agrupaciones

$$k = 2$$

TERCERO: seleccionar aleatoriamente k objetos del conjunto de datos, los cuales son los medoids iniciales para los grupos, en este caso se eligieron las observaciones O1 y O2.

CUARTO: Calcular la distancia de cada punto a los medoids, para ello se calculará la distancia personalizada.

$$d_P(P,Q) = \sum_{i=1}^{m_r} (p_i - q_i)^2 + \gamma \sum_{i=1}^{m_c} \delta(p,q)$$

Donde: $\delta(p,q) = 0\ para\ p = q\ y\ \delta(p,q) = 1\ para\ p \neq q$ y

$$\gamma = \frac{promedio\ (\sigma^2_{PESO}, \sigma^2_{ALTURA})}{promedio\ (1 - \sum_i^2 p_{SEXO}{}^2\ , 1 - \sum_i^2 p_{ESTADO\ CIVIL}{}^2\ , 1 - \sum_i^2 p_{TIENE\ HIJOS}{}^2)}$$

$$\gamma = \frac{promedio\ (0.2002667, 0.1556800)}{promedio\ (1 - \sum_i^2 p_{SEXO}{}^2\ , 1 - \sum_i^2 p_{ESTADO\ CIVIL}\ , 1 - \sum_i^2 p_{TIENE\ HIJOS})}$$

$$\gamma = \frac{0.1779733}{promedio\ (1-(0.5^2+0.5^2), 1-(0.5^2+0.5^2), 1-(0.5^2+0.5^2))}$$

$$\gamma = \frac{0.1779733}{0.5} = 0.3559467$$

Es decir: O3 frente a O1 y O2

$$d_E(O3, O1) = (0-0.17)^2 + (0.17-0.46)^2 + 0.3559467(0+1+0) = 0.4689467$$
$$d_E(O3, O2) = (0-0.98)^2 + (0.17-0.67)^2 + 0.3559467(1+0+1) = 1.9222933$$
…

Así se forma la matriz de distancias:

	O1	O2
O3	0.47	1.92
O4	1.59	0.09
O5	0.21	2.14
O6	1.69	0.47

QUINTO**:** Calcular el costo total de cambio TC_{ki}

$$TC_{ki} = costo(x, c) = \sum_{i=1}^{k} d^2_{(x_k, c_i)}$$

$$TC_{2i} = 0.47^2 + 0.09^2 + 0.21^2 + 0.47^2 = \mathbf{0.49}$$

SEXTO**:** Selecciona una de las combinaciones no medoids, se asume O3 y O5 y con estos se calcula el costo total de cambio:

	O3	O5
O1	0.47	0.21
O2	1.92	2.14
O4	1.79	2.17
O6	2.76	2.37

$$TC_{2i} = 0.21^2 + 1.92^2 + 1.79^2 + 2.37^2 = \mathbf{12.56}$$

Siendo el costo del intercambio de medoids de O1|O2 a O3|O5:

$$S = \ Costo\ total\ \ actual - Costo\ total\ anterior$$

$$S = 12.56 - 0.49 = 12.07 > 0$$

De manera que cambiar a O3|O5 sería una idea mala, así que la elección anterior sería mejor.

SÉPTIMO**:** este proceso iterativo se calcula el costo total de todas las posibles $\frac{n!}{k!(n-k)!}$ combinaciones de medoids.

$$\frac{6!}{2!\,(6-2)!} = 15\ posibles\ medoides$$

De manera iterativa y aleatoria se obtiene aquel que genere el mínimo costo total (en este ejemplo se hará de manera ordenada).

$TC(O1, O3) = 8.57 \rightarrow S = 8.57 - 0.49 > 0$ ✗

$TC(O1, O4) = 0.46 \rightarrow S = 0.46 - 0.49 < 0$ ✓ → Los nuevos medoids son: O1 y O4

OCTAVO**:** asignación de las observaciones a los medoids más cercano determinados en el paso anterior:

	O1	O4
O2	1.77	0.09
O3	0.47	1.79
O5	0.21	2.17
O6	1.69	0.43

$$TC_{2i} = 0.\mathbf{46}$$

NOVENO**:** Probar si alguna de las combinaciones no medoids, disminuye el costo total de cambio, sin considerar aquellas combinaciones calculadas previamente.

$TC(O1, O5) = 8.70 \rightarrow S = 8.70 - 0.46 > 0$ ✗

$TC(O1, O6) = 0.66 \rightarrow S = 0.66 - 0.46 > 0$ ✗

$TC(O2, O3) = 0.62 \rightarrow S = 0.62 - 0.46 > 0$ ✗

$TC(O2, O4) = 6.15 \rightarrow S = 6.15 - 0.46 > 0$ ✗

$TC(O2, O5) = 0.45 \rightarrow S = 0.45 - 0.46 < 0$ ✓ → Los nuevos medoids son: O2 y O5

asignación de las observaciones a los medoids más cercano determinados en el paso anterior:

	O2	O5
O1	1.77	0.21
O3	1.92	0.42
O4	0.09	2.17
O6	0.47	2.37

$$TC_{2i} = 0.\mathbf{45}$$

Probar si alguna de las combinaciones no medoids, disminuye el costo total de cambio, sin considerar aquellas combinaciones calculadas previamente.

$TC(O2, O6) = 11.15 \rightarrow S = 11.15 - 0.45 > 0$ ✗

$TC(O3, O4) = 0.59 \rightarrow S = 0.59 - 0.45 > 0$ ✗

$TC(O3, O6) = 0.80 \rightarrow S = 0.80 - 0.45 > 0$ ✗

$TC(O4, O5) = 0.41 \rightarrow S = 0.41 - 0.45 < 0$ ✓ → Los nuevos medoids son: O4 y O5

asignación de las observaciones a los medoids más cercano determinados en el paso anterior:

	O4	O5
O1	1.59	0.21
O2	0.09	2.14
O3	1.79	0.42
O6	0.43	2.37

$$TC_{2i} = 0.\mathbf{41}$$

Probar si alguna de las combinaciones no medoids, disminuye el costo total de cambio, sin considerar aquellas combinaciones calculadas previamente.

$TC(O4, O6) = 10.46 \rightarrow S = 10.46 - 0.41 > 0$ ✗

$TC(O5, O6) = 0.62 \rightarrow S = 0.62 - 0.41 > 0$ ✗

Finalmente, los medoids que permitieron reducir el costo total fueron O4 y O5

OCTAVO**:** asignación de las observaciones a los medoids más cercano determinados en el paso anterior:

	O4	O5
O1	1.59	0.21
O2	0.09	2.14
O3	1.79	0.42
O6	0.43	2.37

En el ejemplo los clusters quedarían de la siguiente forma:

CLUSTER 1	CLUSTER 2
O1	O2
O3	O4
O5	O6

Anexo 2: Preprocesamiento de variables cuantitativas

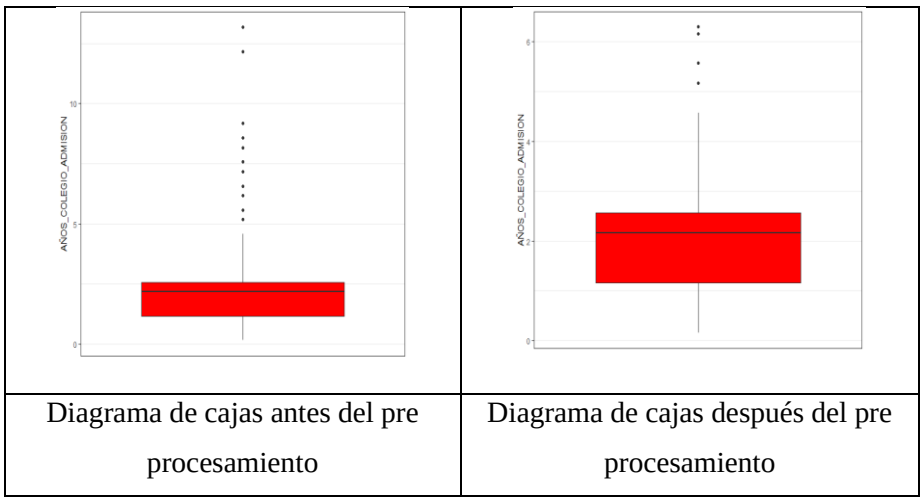


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 23: Diagrama de cajas para la variable Años_Colegio_Admisión

Se observó que la variable Años_Colegio_Admisión presenta outliers superiores, se reemplazaron aquellos que eran superiores al Q97 y fueron reemplazados por el mismo; no

se reemplazaron todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering.

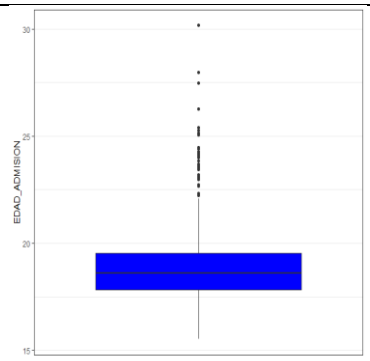


Diagrama de cajas antes del pre procesamiento

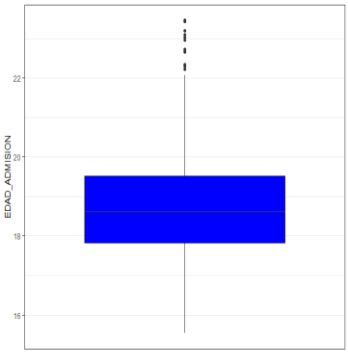


Diagrama de cajas después del pre procesamiento

Figura 24: Diagrama de cajas para la variable Edad_Admisión

Se observó que la variable Edad_Admisión presenta outliers superiores, se reemplazaron aquellos que eran superiores al Q97 y fueron reemplazados por el mismo; no se reemplazaron todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering.

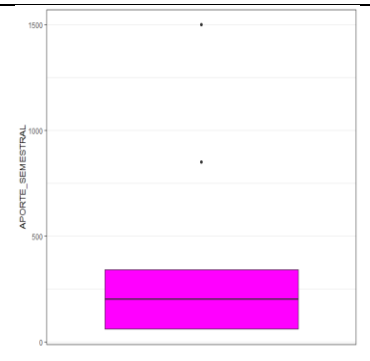


Diagrama de cajas antes del pre procesamiento

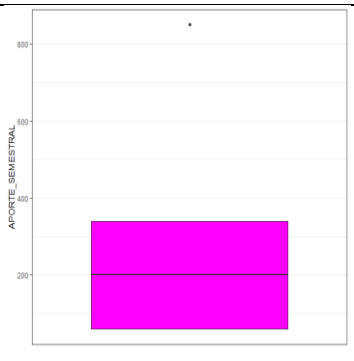


Diagrama de cajas después del pre procesamiento

Figura 25: Diagrama de cajas para la variable Aporte_Semestral

Se observó que la variable Aporte_Semestral presenta outliers superiores, se reemplazaron aquellos que eran superiores al Q97 y fueron reemplazados por el mismo; no se reemplazaron

todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering.

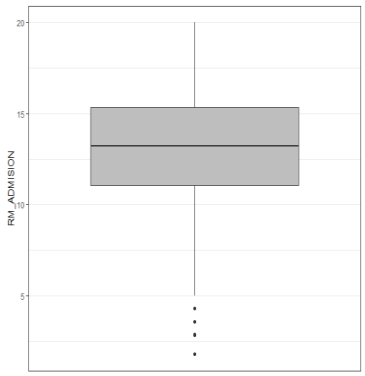


Diagrama de cajas antes del pre procesamiento

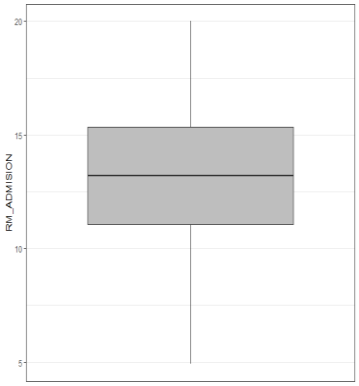


Diagrama de cajas después del pre procesamiento

Figura 26: Diagrama de cajas para la variable RM_Admisión

Se observó que la variable RM_Admisión presenta outliers inferiores, se reemplazaron aquellos que eran superiores al Q1 y fueron reemplazados por el mismo.

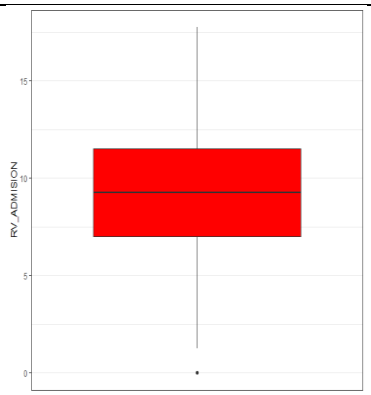


Diagrama de cajas antes del pre procesamiento

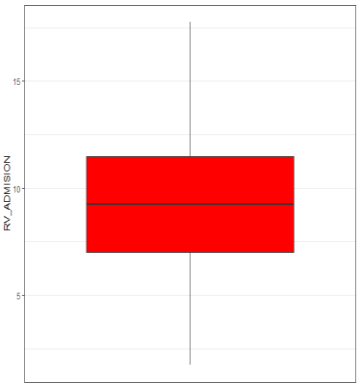


Diagrama de cajas después del pre procesamiento

Figura 27: Diagrama de cajas para la variable RV_Admisión

Se observó que la variable RV_Admisión presenta outliers inferiores, se reemplazaron aquellos que eran superiores al Q1 y fueron reemplazados por el mismo.

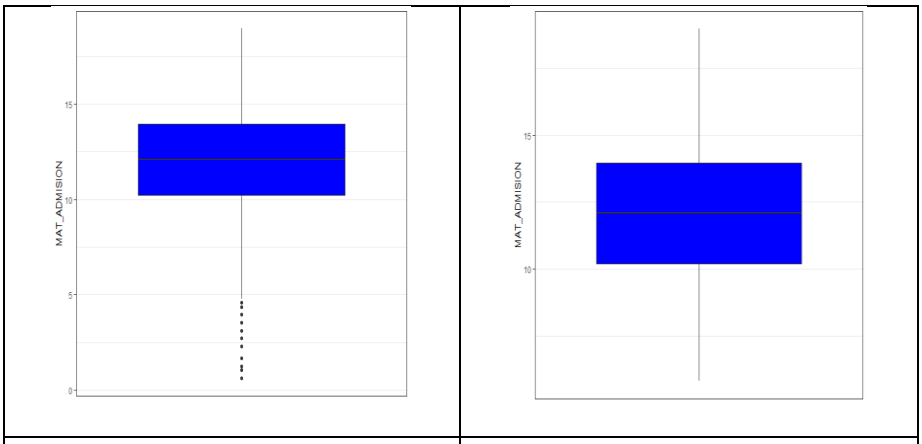


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 28: Diagrama de cajas para la variable MAT_Admisión

Se observó que la variable MAT_Admisión presenta outliers inferiores, se reemplazaron aquellos que eran superiores al Q3 y fueron reemplazados por el mismo.

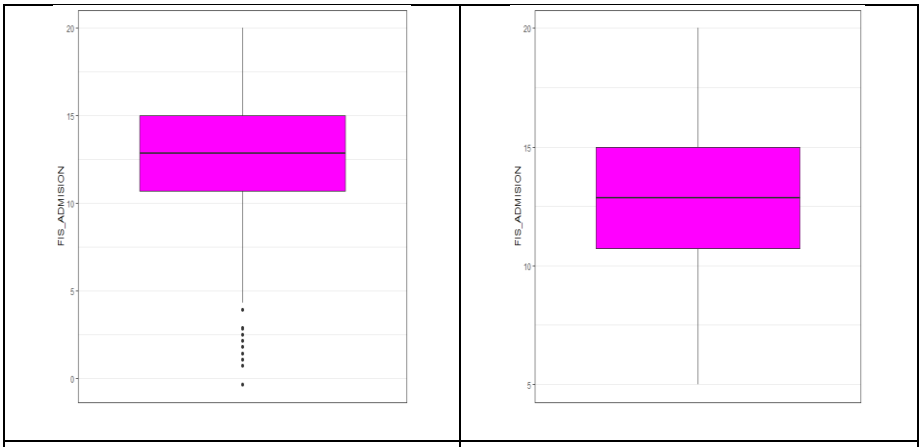
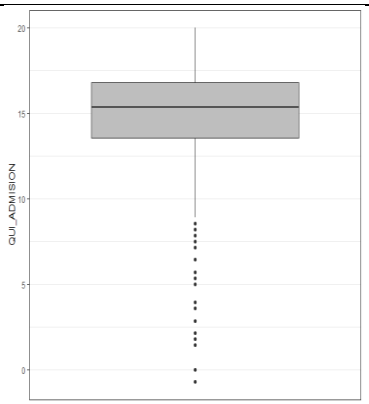


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 29: Diagrama de cajas para la variable FIS_Admisión

Se observó que la variable FIS_Admisión presenta outliers inferiores, se reemplazaron aquellos que eran superiores al Q3 y fueron reemplazados por el mismo.



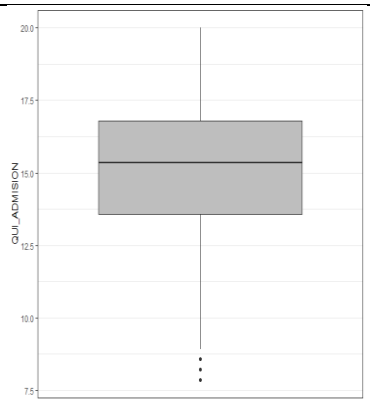
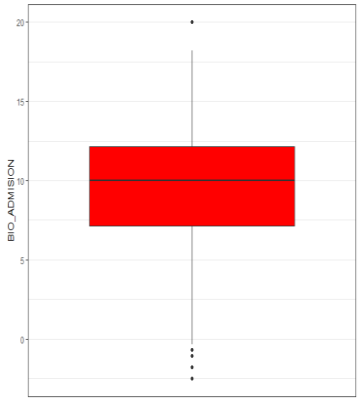


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 30: Diagrama de cajas para la variable QUI_Admisión

Se observó que la variable QUI_Admisión presenta outliers inferiores, se reemplazaron aquellos que eran superiores al Q3 y fueron reemplazados por el mismo; no se reemplazaron todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering



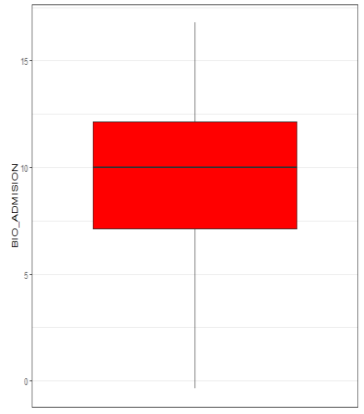


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 31: Diagrama de cajas para la variable BIO_Admisión

Se observó que la variable BIO_Admisión presenta outliers superiores e inferiores, se reemplazaron aquellos que eran superiores al Q99 e inferiores al Q1 fueron reemplazados por los mismos; no se reemplazaron todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering

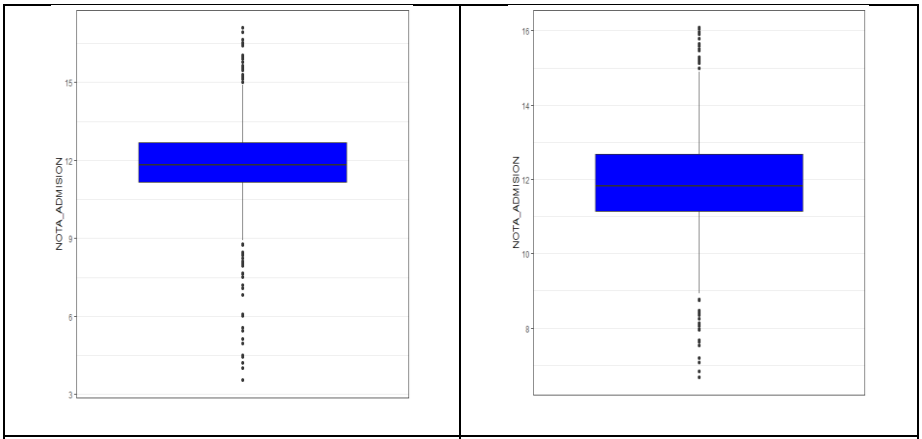


Diagrama de cajas antes del pre procesamiento	Diagrama de cajas después del pre procesamiento

Figura 32: Diagrama de cajas para la variable Nota_Admisión

Se observó que la variable Nota_Admisión presenta outliers superiores e inferiores, se reemplazaron aquellos que eran superiores al Q99 e inferiores al Q2 y fueron reemplazados por los mismos; no se reemplazaron todos los puntos extremos, con el fin de no quitar variabilidad y probar el desempeño de los algoritmos clustering.

Anexo 3: Validación del agrupamiento

A. Análisis de varianza con las variables cuantitativa

Tabla 23: ANVA para la variable Años_Colegio_ Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
Años_Colegio_ Admisión	Entre grupos	489.635	2	244.818	177.762	6.05E-63	***
	Dentro de grupos	946.151	687	1.377			
	Total	1435.787	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable Años_Colegio_ Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una

heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

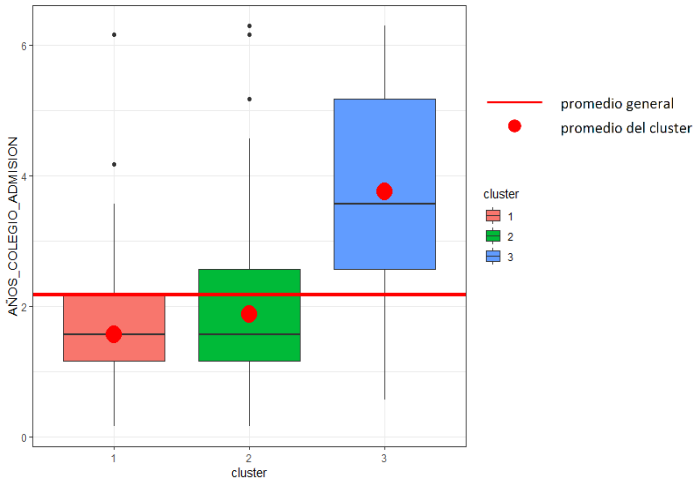


Figura 33: Diagrama de cajas por cluster según la variable Años_Colegio_ Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 3 tienen valores mayores en la variable Años_Colegio_ Admisión frente al promedio general; es decir estos alumnos lograron ingresar a la universidad en mayor tiempo. Por otro lado, los ingresantes pertenecientes al clusters 2 poseen valores de la variable Años_Colegio_ Admisión muy parecidos al promedio general; es decir a estos alumnos les tomó regular tiempo ingresar a la universidad, finalmente el clusters 1 agrupa a los ingresantes que en promedio les tomó menos tiempo ingresar a la universidad.

Tabla 24: ANVA para la variable Edad_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
Edad_Admisión	Entre grupos	550.981	2	275.491	173.763	8.53E-62	***
	Dentro de grupos	1089.194	687	1.585			
	Total	1640.175	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable Edad_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre

grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

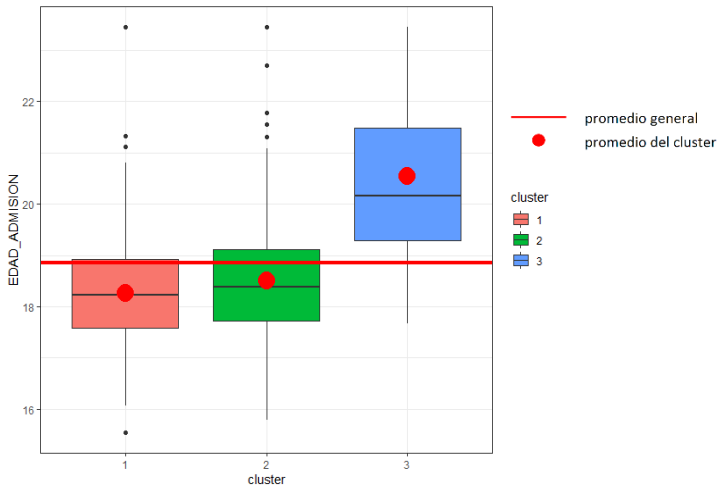


Figura 34: Diagrama de cajas por cluster según la la variable Edad_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 3 tienen valores mayores en la variable Edad_Admisión frente al promedio general; es decir estos alumnos ingresaron a la universidad con una edad mayor. Por otro lado, los ingresantes pertenecientes al cluster 2 poseen valores de la variable Edad_Admisión muy parecidos al promedio general; finalmente el clusters 1 agrupa alumnos que ingresaron a la universidad con una edad menor al promedio general.

Tabla 25: ANVA para la variable Aporte_Semestral

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
Aporte_Semestral	Entre grupos	4314997.055	2	2157498.528	108.098	1.53E-41	***
	Dentro de grupos	13711594.97	687	19958.654			
	Total	18026592.03	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable Aporte_Semestral, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir

que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

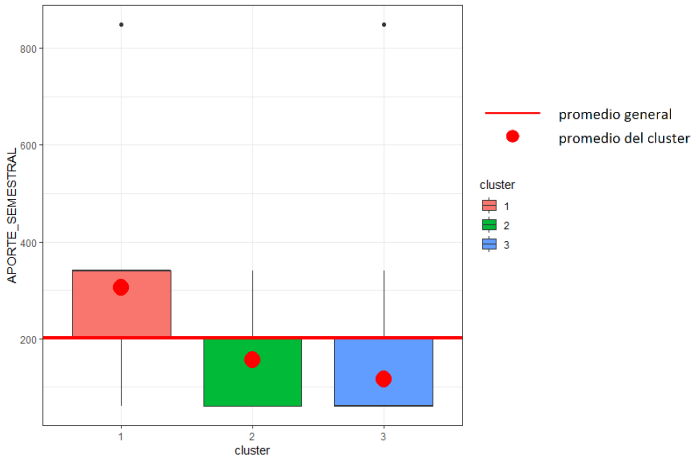


Figura 35: Diagrama de cajas por cluster según la variable Aporte_Semestral

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tienen valores mayores en la variable Aporte_Semestral frente al promedio general; es decir estos alumnos tienen designado un aporte semestral mayor del promedio. Por otro lado, los ingresantes pertenecientes al cluster 2 poseen valores de la variable Aporte_Semestral muy parecidos al promedio general; finalmente el clusters 3 agrupa a los ingresantes que en promedio semestralmente pagan un menor aporte al momento de su matrícula, dada su situación socio económica.

Tabla 26: ANVA para la variable CTA_Colegio

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
CTA_ Colegio	Entre grupos	1141.667	2	570.833	198.971	6.80E-69	***
	Dentro de grupos	1970.955	687	2.869			
	Total	3112.622	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable CTA_Colegio, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre

grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

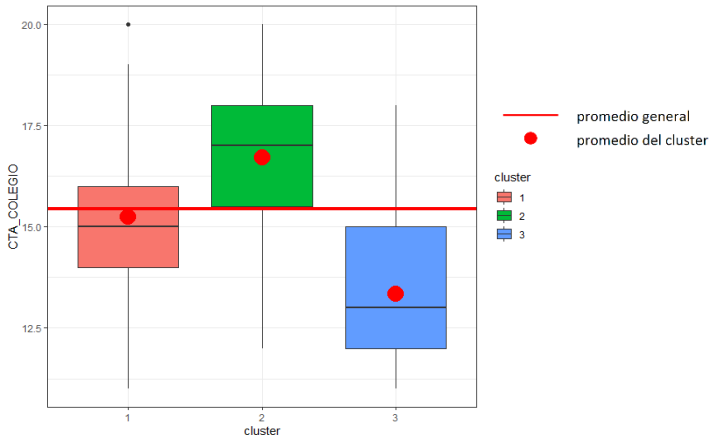


Figura 36: Diagrama de cajas por cluster según la variable CTA_Colegio

Se observa en promedio, que los ingresantes pertenecientes al cluster 2 tienen valores mayores en la variable CTA_Colegio frente al promedio general; es decir estos alumnos lograron obtener en el área de Ciencia, Tecnología y Ambiente (C.T.A.) en el último año de nivel secundaria, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 1 poseen valores muy parecidos al promedio general de la variable CTA_Colegio, en otras palabras, estos estudiantes alcanzaron notas regulares en esta área; finalmente el clusters 3 agrupa a los ingresantes que en promedio obtuvieron en el área de C.T.A. en el último año de nivel secundaria notas bajas.

Tabla 27: ANVA para la variable COM_Colegio

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
COM_ Colegio	Entre grupos	1028.908	2	514.454	197.313	1.95E-68	***
	Dentro de grupos	1791.214	687	2.607			
	Total	2820.122	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable COM_Colegio, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre

grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

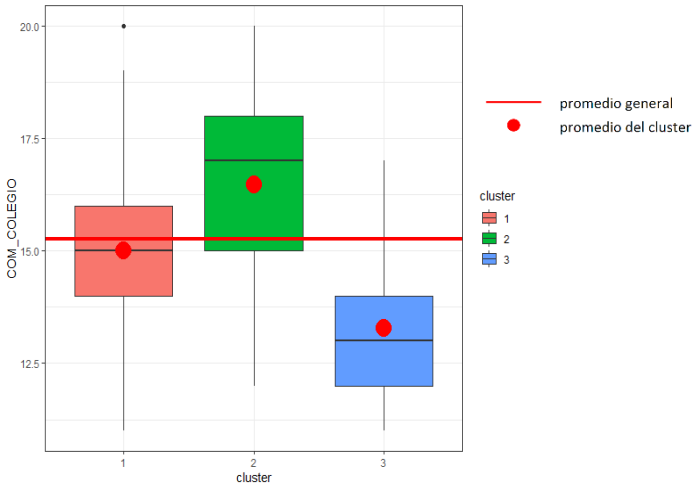


Figura 37: Diagrama de cajas por cluster según la variable COM_Colegio

Se observa en promedio, que los ingresantes pertenecientes al cluster 2 tienen valores mayores en la variable COM_Colegio frente al promedio general; es decir estos alumnos lograron obtener en el área de Comunicación en el último año de nivel secundaria, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 1 poseen valores muy parecidos al promedio general de la variable COM_Colegio, en otras palabras, estos estudiantes alcanzaron notas regulares en esta área; finalmente los ingresantes pertenecientes al cluster 3 poseen valores menores frente al promedio de la variable COM_Colegio; es decir, agrupan a los ingresantes que en promedio obtuvieron en el área de Comunicación en el último año de nivel secundaria notas bajas.

Tabla 28: ANVA para la variable MAT_Colegio

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
MAT_ Colegio	Entre grupos	996.017	2	498.009	168.984	2.07E-60	***
	Dentro de grupos	2024.643	687	2.947			
	Total	3020.661	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable MAT_Colegio, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que

se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

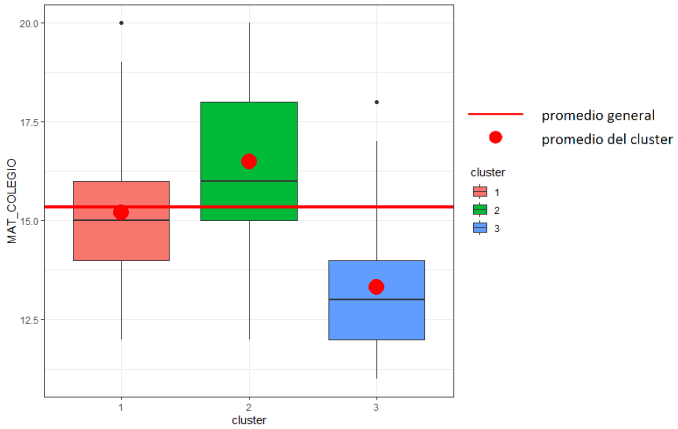


Figura 38: Diagrama de cajas por cluster según la variable MAT_Colegio

Se observa en promedio, que los ingresantes pertenecientes a los cluster 2 tienen valores mayores en la variable MAT_Colegio frente al promedio general; es decir estos alumnos lograron obtener en el área de Matemática en el último año de nivel secundaria, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 1 poseen valores muy parecidos al promedio general de la variable MAT_Colegio, en otras palabras, estos estudiantes alcanzaron notas regulares en esta área; finalmente el cluster 3 agrupa a los ingresantes que en promedio obtuvieron en el área de Matemática en el último año de nivel secundaria notas bajas.

Tabla 29: ANVA para la variable Nota_Colegio

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
Nota_ Colegio	Entre grupos	766.364	2	383.182	273.133	5.21E-88	***
	Dentro de grupos	963.802	687	1.403			
	Total	1730.165	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable Nota_Colegio, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que

se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

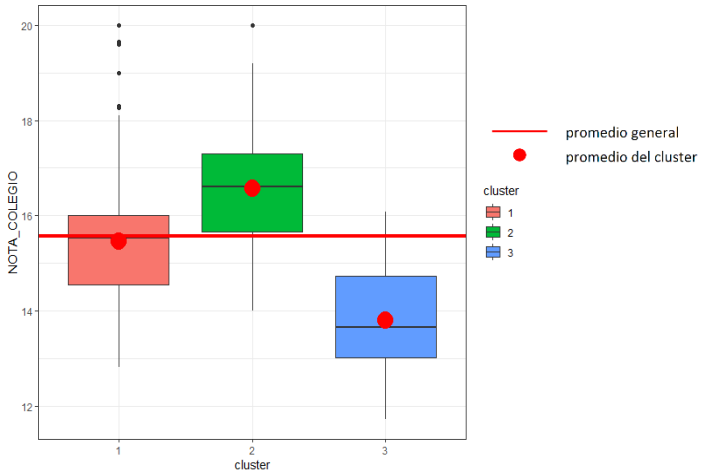


Figura 39: Diagrama de cajas por cluster según la variable Nota_Colegio

Se observa en promedio, que los ingresantes pertenecientes al cluster 2 tienen valores mayores en la variable Nota_Colegio frente al promedio general; es decir estos alumnos en promedio obtuvieron en el examen de admisión 2015 notas altas. Por otro lado, los ingresantes pertenecientes al cluster 1 poseen valores muy parecidos al promedio general de la variable Nota_Colegio, en otras palabras, estos estudiantes lograron obtener en el examen de admisión 2015 notas regulares; finalmente el cluster 3 agrupa a los ingresantes que obtuvieron en el examen de admisión 2015, notas bajas.

Tabla 30: ANVA para la variable RM_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
RM_ Admisión	Entre grupos	712.597	2	356.299	46.644	1.01E-19	***
	Dentro de grupos	5247.802	687	7.639			
	Total	5960.399	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable RM_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que

se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

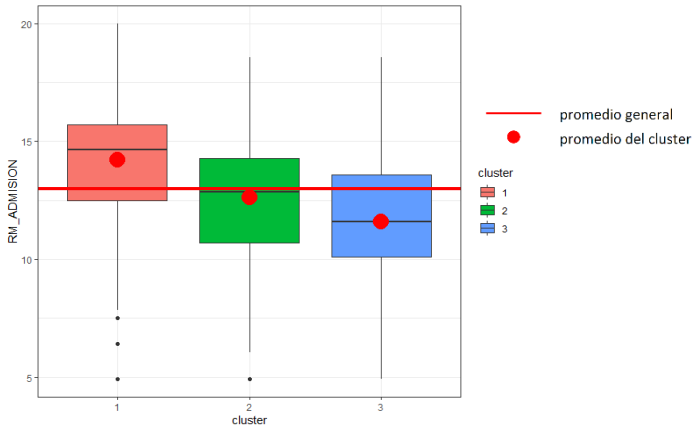


Figura 40: Diagrama de cajas por cluster según la variable RM_Admisión

Se observa en promedio, que los ingresantes pertenecientes el cluster 1 tiene valores mayores en la variable RM_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el curso de Razonamiento Matemático en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 2 poseen valores muy parecidos al promedio general de la variable RM_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, en el curso de Razonamiento Matemático notas regulares; finalmente el cluster 3 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el curso de Razonamiento Matemático notas bajas.

Tabla 31: ANVA para la variable RV_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
RV_ Admisión	Entre grupos	173.647	2	86.823	9.644	7.40E-05	***
	Dentro de grupos	6185.155	687	9.003			
	Total	6358.802	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable RV_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que

se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

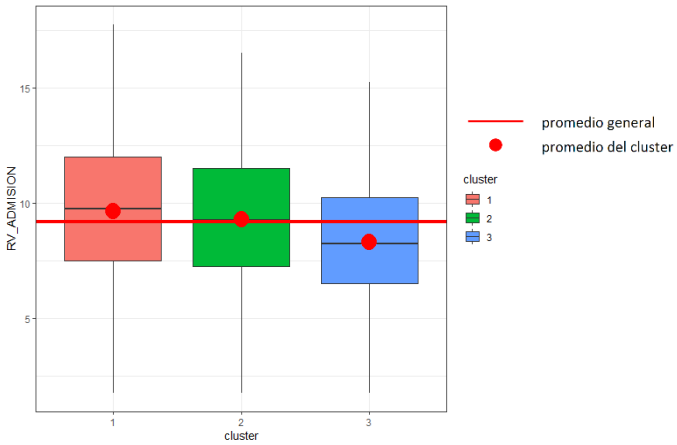


Figura 41: Diagrama de cajas por cluster según la variable RV_Admisión

Se observa en promedio, que los ingresantes pertenecientes el cluster 1 tiene valores mayores en la variable RV_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el curso de Razonamiento Verbal en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 2 poseen valores muy parecidos al promedio general de la variable RV_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, en el curso de Razonamiento Verbal notas regulares; finalmente el clusters 3 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el curso de Razonamiento Verbal notas bajas.

Tabla 32: ANVA para la variable MAT_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
MAT_ Admisión	Entre grupos	404.943	2	202.472	29.76	4.02E-13	***
	Dentro de grupos	4673.918	687	6.803			
	Total	5078.861	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable MAT_Admisión,

la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

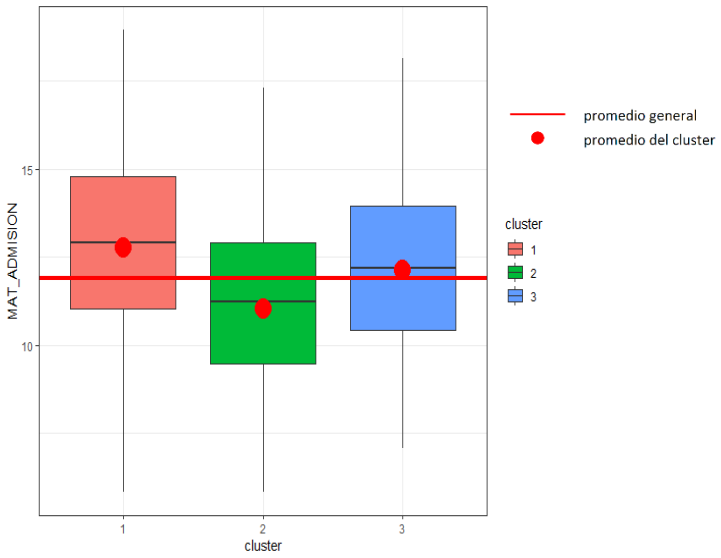


Figura 42: Diagrama de cajas por cluster según la variable MAT_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tienen valores mayores en la variable MAT_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el área de Matemática en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 3 poseen valores muy parecidos al promedio general de la variable MAT_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, en el área de Matemática notas regulares; finalmente el clusters 2 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el área de Matemática notas bajas.

Tabla 33: ANVA para la variable FIS_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
FIS_ Admisión	Entre grupos	1432.92	2	716.46	81.757	1.41E-32	***
	Dentro de grupos	6020.357	687	8.763			
	Total	7453.276	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable FIS_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

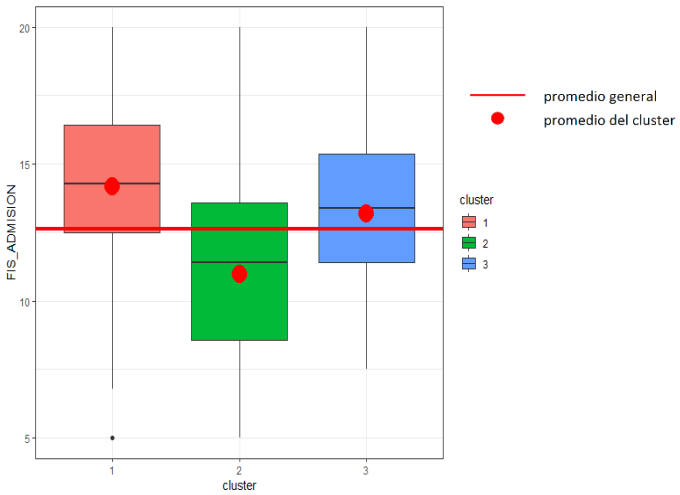


Figura 43: Diagrama de cajas por cluster según la variable FIS_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tienen valores mayores en la variable FIS_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el curso de Física en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 3 poseen valores muy parecidos al promedio general de la variable FIS_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, en el curso de Física notas regulares; finalmente el cluster 2 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el curso de Física notas bajas.

Tabla 34: ANVA para la variable QUI_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
QUI_ Admisión	Entre grupos	462.424	2	231.212	33.023	2.03E-14	***
	Dentro de grupos	4810.121	687	7.002			
	Total	5272.545	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable QUI_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

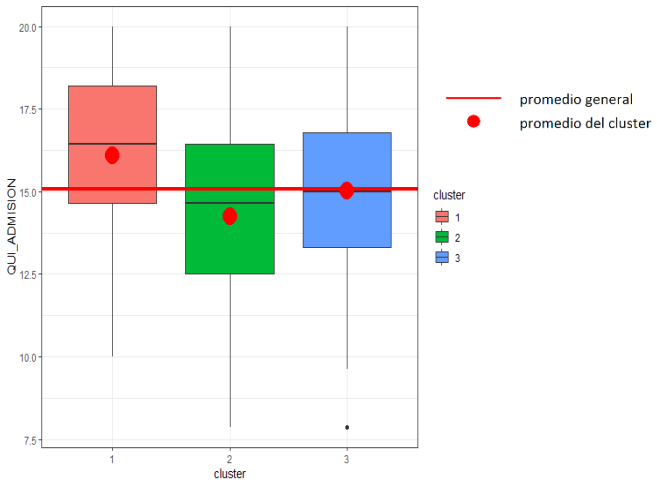


Figura 44: Diagrama de cajas por cluster según la variable QUI_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tienen valores mayores en la variable QUI_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el curso de Química en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 3 poseen valores muy parecidos al promedio general de la variable QUI_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, en el curso de Química notas regulares; finalmente el clusters 2 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el curso de Química notas bajas.

Tabla 35: ANVA para la variable BIO_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
BIO_ Admisión	Entre grupos	364.099	2	182.05	12.635	4.08E-06	***
	Dentro de grupos	9898.448	687	14.408			
	Total	10262.547	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable BIO_Admisión, la variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

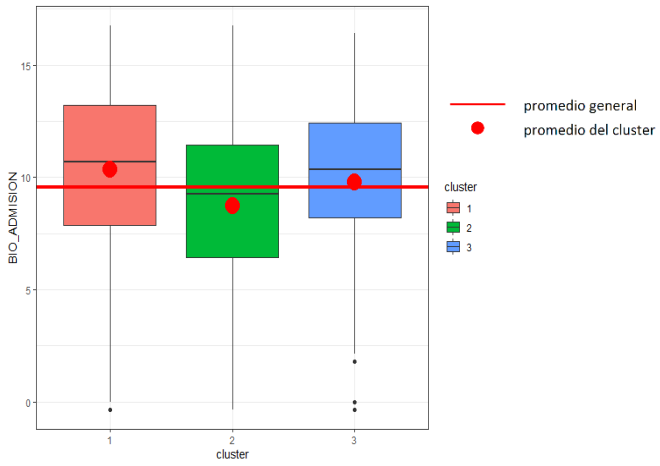


Figura 45: Diagrama de cajas por cluster según la variable BIO_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tiene valores mayores en la variable BIO_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el curso de Biología en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes a los clusters 2 y 3 agrupan a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, en el curso de Biología notas regulares y bajas.

Tabla 36: ANVA para la variable Nota_Admisión

VARIABLE	VARIANZA	SUMA DE CUADRADOS	GL	MEDIA CUADRÁTICA	F VALUE	PROB	SIG
Nota_ Admisión	Entre grupos	412.446	2	206.223	109.28	6.22E-42	***
	Dentro de grupos	1296.439	687	1.887			
	Total	1708.884	689				

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de la igualdad de medias, lo que indicaría que para la variable Nota_Admisión, la

variabilidad entre los grupos es mayor que la variabilidad dentro de los grupos; es decir que se puede afirmar, en términos del análisis clustering, que existe una heterogeneidad entre grupos y homogeneidad dentro de grupos; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

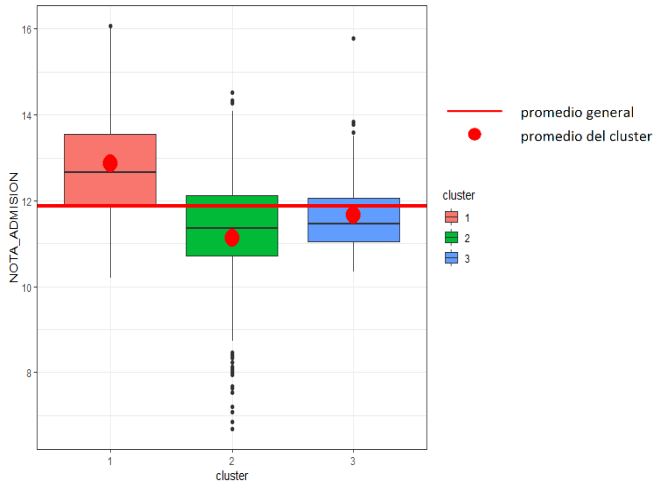


Figura 46: Diagrama de cajas por cluster según la variable Nota_Admisión

Se observa en promedio, que los ingresantes pertenecientes al cluster 1 tiene valores mayores en la variable Nota_Admisión frente al promedio general; es decir estos alumnos lograron obtener en el examen de admisión 2015, notas altas. Por otro lado, los ingresantes pertenecientes al cluster 3 poseen valores muy parecidos al promedio general de la variable Nota_Admisión, en otras palabras, estos alumnos lograron obtener en el examen de admisión 2015, notas regulares; finalmente el cluster 2 agrupa a los ingresantes que en promedio obtuvieron en el examen de admisión 2015, notas bajas.

B. Prueba Chi cuadrado para variables cualitativas

Tabla 37: Prueba Chi cuadrada entre la variable Dept_Colegio y clusters

VARIABLE	PRUEBA	VALOR	GL	PROB	SIG
Dept_Colegio	Chi-cuadrado de Pearson	0.603	2	7.40E-01	

La prueba resultó no significativa, es decir, no existe evidencia estadística para rechazar la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Dept_Colegio no existe una asociación; por lo que dicha variable no aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 38: Distribución de la variable Dept_Colegio según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Dept_Colegio	Lima	35.7%	42.2%	22.0%	100.0%
	Provincia	40.0%	41.8%	18.2%	100.0%

Se observa que los ingresantes provenientes de colegios de Lima y provincias fueron distribuidos de manera similar en los clusters.

Tabla 39: Prueba Chi cuadrada entre la variable Sexo y clusters

VARIABLE	PRUEBA	VALOR	GL	PROB	SIG
Sexo	Chi-cuadrado de Pearson	132.178	2	1.99E-29	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Sexo existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 40: Distribución de la variable Sexo según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Sexo	F	24.6%	63.6%	11.8%	100.0%
	M	47.7%	20.6%	31.7%	100.0%

Se observa que la mayoría de las mujeres ingresantes en el 2015 fueron agrupadas en el cluster 2; mientras que los ingresantes varones fueron agrupados en los clusters 1 y 3.

Tabla 41: Prueba Chi cuadrada entre la variable Tipo_Colegio y clusters

VARIABLE	PRUEBA	VALOR	GL	PROB	SIG
Tipo_Colegio	Chi-cuadrado de Pearson	149.191	2	4.01E-33	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Tipo_Colegio existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 42: Distribución de la variable Tipo_Colegio según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Tipo_Colegio	Privada	52.6%	36.5%	10.9%	100.0%
	Pública	10.1%	51.1%	38.8%	100.0%

Se observa que la mayoría de los ingresantes provenientes de colegios privados fueron agrupadas en los clusters 1 y 2; mientras que los ingresantes provenientes de colegios públicos fueron agrupados en los clusters 2 y 3.

Tabla 43: Prueba Chi cuadrada entre la variable Tercio_Superior_ESP y clusters

Variable	Prueba	Valor	gl	Prob	Sig
Tercio_Superior_ESP	Chi-cuadrado de Pearson	112.169	2	4.39E-25	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones

formadas y la variable Tercio_Superior_ESP existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 44: Distribución de la variable Tercio_Superior_ESP según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Tercio_Superior_ESP	No	23.7%	52.5%	23.9%	100.0%
	Si	64.9%	18.3%	16.8%	100.0%

Se observa que la mayoría de los ingresantes que pertenecieron al tercio superior de la carrera que ingresaron en el examen de admisión fueron agrupados en el cluster 1; mientras que los ingresantes que no pertenecieron fueron agrupados en el cluster 2.

Tabla 45: Prueba Chi cuadrada entre la variable Modalidad y clusters

Variable	Prueba	Valor	gl	Prob	Sig
Modalidad	Chi-cuadrado de Pearson	85.368	2	2.90E-19	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Modalidad existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 46: Distribución de la variable Modalidad según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Modalidad	Concurso Ordinario	39.5%	36.4%	24.2%	100.0%
	Dos Primeros Puestos de Colegios de Educación Secundaria	5.8%	94.2%	0.0%	100.0%

Se observa que la mayoría de los ingresantes que ingresaron por la modalidad Dos Primeros Puestos de Colegios de Educación Secundaria fueron agrupados en el cluster 2; mientras que los ingresantes por Concurso Ordinario fueron agrupados en el cluster 1.

Tabla 47: Prueba Chi cuadrada entre la variable Especialidad y clusters

VARIABLE	PRUEBA	VALOR	GL	PROB	SIG
Especialidad	Chi-cuadrado de Pearson	92.378	22	1.35E-10	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Especialidad existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 48: Distribución de la variable Especialidad según clusters

		CLUSTERS			
% DENTRO DE	ATRIBUTOS	1	2	3	TOTAL
Especialidad	Agronomía	29.7%	51.6%	18.8%	100.0%
	Biología	56.8%	34.1%	9.1%	100.0%
	Ciencias Forestales	51.1%	28.9%	20.0%	100.0%
	Economía	27.3%	40.9%	31.8%	100.0%
	Estadística Informática	22.5%	52.5%	25.0%	100.0%
	Gestión Empresarial	48.1%	34.6%	17.3%	100.0%
	Industrias Alimentarias	38.2%	50.0%	11.8%	100.0%
	Ingeniería Agrícola	27.4%	33.9%	38.7%	100.0%
	Ingeniería Ambiental	76.9%	23.1%	0.0%	100.0%
	Meteorología	56.5%	26.1%	17.4%	100.0%
	Pesquería	20.0%	51.7%	28.3%	100.0%
	Zootecnia	22.4%	45.9%	31.8%	100.0%

Se observa que la mayoría de los alumnos ingresantes a la carrera de Agronomía fueron agrupados en el cluster 2, de Biología en el cluster 1, Ciencias Forestales en el clusters 1, Economía en los clusters 2 y 3; Estadística Informática en el cluster 2, Gestión Empresarial en los clusters 1 y 2, Industrias Alimentarias en el cluster 2, Ingeniería Agrícola en los clusters 2 y 3, Ingeniería Ambiental en el cluster 1, Ingeniería en, Meteorología en el cluster 1, Pesquería en el cluster 2; finalmente Zootecnia en el cluster 2.

Tabla 49: Prueba Chi cuadrada entre la variable Elección_ESP_Ingreso y clusters

VARIABLE	PRUEBA	VALOR	GL	PROB	SIG
Elección_ESP_Ingreso	Chi-cuadrado de Pearson	64.867	4	2.74E-13	***

La prueba resultó altamente significativa, es decir, existe evidencia estadística para rechaza la hipótesis de independencia de variables, lo que indicaría que entre las agrupaciones formadas y la variable Elección_ESP_Ingreso existe una asociación; por lo que dicha variable aporta significativamente para obtener el perfil del ingresante de la UNALM.

Tabla 50: Distribución de la variable Elección_ESP_Ingreso según clusters

		CLUSTERS			TOTAL
% DENTRO DE	ATRIBUTOS	1	2	3	
Elección_ESP_Ingreso	Primera	53.2%	32.0%	14.9%	100.0%
	Segunda	30.4%	45.2%	24.4%	100.0%
	Tercera	15.9%	55.0%	29.1%	100.0%

Se observa que la mayoría de los alumnos que ingresaron a la carrera que eligieron como primera opción al postular, fueron agrupados en el cluster 1, mientras que los alumnos que ingresaron a la carrera que eligieron como segunda opción al postular, fueron agrupados en los clusters 1 y 2, finalmente los alumnos que ingresaron a la carrera que eligieron como última opción al momento de postular, fueron agrupados en los clusters 2 y 3.

Anexo 4: Códigos utilizados para el procesamiento de datos

```
################
### PAQUETES ###
################

library("mlbench")
library("clustMixType")
library("cluster")
library("ggplot2")
library("factoextra")
library("plotrix")
library("C50")
library("caret")
library("partykit")
library("plot3D")
library("gtools")
library("randomForest")
library("Boruta")
library("desc")
library("dplyr")

#################################
### PRE PROCESAMIENTO DE DATOS ###
#################################

######### IMPORTAR LOS DATOS
datos_completos<-read.delim("clipboard",header=T)


######### SELECCIONAR LAS VARIABLES CON LAS QUE SE TRABAJARÁ
datos<-datos_completos[,-c(1:5)]
summary(datos)
str(datos)


#########FUNCIÓN QUE PERMITE DETERMINAR EL TIPO DE VARIABLE
tipo_de_variable<-function(datos)
  {
  variable<-c()
  tipo<-c()

  for(i in 1:ncol(datos))
  {
    variable[i]<-colnames(datos)[i]
    tipo[i]<-class(datos[,i])
  }
  tmp<-data.frame(orden=1:ncol(datos),variable,tipo)
  return(tmp)
}

tipo_de_variable(datos)



######### SEPARAR VARIABLES CUALITATIVAS Y CUANTITATIVAS PARA DAR FORMATO

cualis_datos<-datos[,c(15:ncol(datos))]
cuantis_datos<-datos[,-c(15:ncol(datos))]

######### PARÁMETROS PARA LA VISUALIZACIÓN DE DATOS

x<-rep(factor(0),nrow(cuantis_datos))
colores<-seq(10,10*14,10)
nombres<-colnames(cuantis_datos)
```

```
######### FUNCIÓN PARA REEMPLAR OUTLIER DE UNA COLUMNA

REEMPLAZAR_SUPERIOR<- function(columna,p)
       {
       Q<-quantile(columna,p)
       temp1<-data.frame(datos=columna,condicion=columna> Q)
       temp2<-as.numeric(rownames(subset(temp1,condicion=='TRUE')))
       temp1[temp2,1]<-Q
       return(data.frame(NUEVA_VARIABLE=temp1[,1]))
       }

REEMPLAZAR_INFERIOR<- function(columna,p)
       {
       Q<-quantile(columna,p)
       temp1<-data.frame(datos=columna,condicion=columna< Q)
       temp2<-as.numeric(rownames(subset(temp1,condicion=='TRUE')))
       temp1[temp2,1]<-Q
       return(data.frame(NUEVA_VARIABLE=temp1[,1]))
       }

######### ANÁLISIS DE CADA VARIABLE
#------------------------------------------------- 1

######### SEPARANDO LA VARIABLE A ANALIZAR
columna<-cuantis_datos[,1]

######### QUANTIL SUPERIOR O INFERIOR POR EL QUE SE REEMPLAZARÁ LOS OUTLIERS (MAX
3%)
p<-0.97

######### APLICANDO LA FUNCIÓN
datos_sin_outlier1<-REEMPLAZAR_SUPERIOR(columna,p)
salida<-datos_sin_outlier1
######### VISUALIZACIÓN DEL PRE PROCESAMIENTO DE DATOS

i<-1

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()


######### REPETIR EL PROCESAMIENTO PARA LAS VARIBLES CON PRESENCIA DE OUTLIERS

#------------------------------------------------- 2

columna<-cuantis_datos[,2]
p<-0.97
datos_sin_outlier2<-REEMPLAZAR_SUPERIOR(columna,p)
salida<-datos_sin_outlier2


i<-2

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
```

```
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()


#------------------------------------------------- 3

columna<-cuantis_datos[,3]
p<-0.97
datos_sin_outlier3<-REEMPLAZAR_SUPERIOR(columna,p)
salida<-datos_sin_outlier3

i<-3

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()
#------------------------------------------------- 4

columna<-cuantis_datos[,8]
p<-0.01
datos_sin_outlier8<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier8

i<-8

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()


#------------------------------------------------- 5

columna<-cuantis_datos[,9]
p<-0.01
datos_sin_outlier9<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier9

i<-9

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
```

```
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

#------------------------------------------------- 6

columna<-cuantis_datos[,10]
p<-0.03
datos_sin_outlier10<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier10
i<-10

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()
#------------------------------------------------- 7

columna<-cuantis_datos[,11]
p<-0.03
datos_sin_outlier11<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier11

i<-11

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

#------------------------------------------------- 8

columna<-cuantis_datos[,12]
p<-0.03
datos_sin_outlier12<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier12

i<-12

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
```

```
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

#------------------------------------------------- 9

columna<-cuantis_datos[,13]
p<-0.01
datos_sin_outlier13_1<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier13_1
p<-0.99
datos_sin_outlier13_2<-REEMPLAZAR_SUPERIOR(datos_sin_outlier13_1[,1],p)
salida<-datos_sin_outlier13_2

i<-13

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

#------------------------------------------------- 10

columna<-cuantis_datos[,14]
p<-0.02
datos_sin_outlier14_1<-REEMPLAZAR_INFERIOR(columna,p)
salida<-datos_sin_outlier14_1

p<-0.99
datos_sin_outlier14_2<-REEMPLAZAR_SUPERIOR(datos_sin_outlier14_1[,1],p)
salida<-datos_sin_outlier14_2

i<-14

ggplot(data=cuantis_datos, aes(x= x , y=columna)) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()

ggplot(data=cuantis_datos, aes(x= x , y=salida[,1])) +
       geom_boxplot(fill=colores[i])+
       xlab('')+
       ylab(colnames(cuantis_datos[i]))+
       scale_x_discrete(breaks=NULL) +
       theme_bw()
```

```
######## JUNTANDO LAS VARIABLES

cuantis_datos<-
data.frame(datos_sin_outlier1,datos_sin_outlier2,datos_sin_outlier3,cuantis_datos
[,c(4:7)],datos_sin_outlier8,

datos_sin_outlier9,datos_sin_outlier10,datos_sin_outlier11,datos_sin_outlier12,da
tos_sin_outlier13_2,
                              datos_sin_outlier14_2)
colnames(cuantis_datos)<-nombres_cuantis

######## DATA FINAL PARA SER EXPORTADA

data_export<-data.frame(datos_completos[,c(1:5)],cuantis_datos,cualis_datos)
write.table(data_export,file="C:/Users/User/Desktop/TESIS/DATOS/data_preprocess.t
xt",row.names=F,sep=",")


#####################################################################
### DETERMINAR EL NÚMERO DE CLUSTERS PARA EL ALGORITMO K-MEDOID ###
#####################################################################

######### SEPARAR NUEVAMENTE LAS VARIABLES CUALITATIVAS Y CUANTITATIVAS DE LA
DATA PRE PROCESADA

datos_completos <- data_export
#datos_completos <-read.delim("clipboard",header=T)
datos<-datos_completos[,-c(1:5)]
cualis_datos<-datos[,c(15:ncol(datos))]
cuantis_datos<-datos[,-c(15:ncol(datos))]


######### ESCALAMIENTO DE VARIABLES CUANTITATIVAS ENTRE UN INTERVALO DE 0 A 1
#y=(x - min(x)) / (max(x)-min(x))


cuantis_datos_norm<-cuantis_datos
for(i in 1:ncol(cuantis_datos_norm))
      {
      cuantis_datos_norm[,i]<-(cuantis_datos_norm[,i]-
as.numeric(min(cuantis_datos_norm[,i])))/as.numeric(max(cuantis_datos_norm[,i])-
min(cuantis_datos_norm[,i]))
      }


######### JUNTANDO LAS VARIABLES REESCALADAS Y LAS CUALITATIVAS
datitos<-data.frame(cuantis_datos_norm,cualis_datos)

######### FUNCIÓN PARA CALCULAR DE LA MATRIZ DE DISTANCIAS MIXTAS
distancias<- function(datos_cuantis,datos_cualis,lambda)
      {
      datos_cuantis<-data.frame(datos_cuantis)
      datos_cualis<-data.frame(datos_cualis)
      lista<-list()
      matriz1<- matrix(0,nrow(datos_cualis),nrow(datos_cualis))
      for(j in 1:ncol(datos_cualis))
            {
            for(i in 1:nrow(datos_cualis))
                  {
#comparando registro i de la variable j
                  matriz1[i,]<-ifelse(datos_cualis[i,j]==datos_cualis[,j],0,1)
                  }
#se genera una lista para cada variable
            lista[[j]]<-matriz1
            }
      matriz3<-matrix(0,nrow(datos_cualis),nrow(datos_cualis))

      for(i in 1:ncol(datos_cualis))
```

```
            {
            matriz2<-matrix(lista[[i]],nrow(datos_cualis),nrow(datos_cualis))
#se suman todas las listas
            matriz3<-matriz3+matriz2
            }

#si hay un sólo elemento en el cluster la distancia debe ser cero
      ifelse(dim(as.matrix(dist(datos_cuantis,method =
"euclidean")^2))[1]==0,distancias_cuantis<-0,distancias_cuantis<-
as.matrix(dist(datos_cuantis,method = "euclidean")^2))

      distancias_cualis<-as.matrix(matriz3)
      return(list(distancias=distancias_cuantis+lambda*distancias_cualis))
      }

######### ÍNDICE DE DAVID BOUDIN PARA EL ALGORITMO K MEDOIDS

DAVIES_BOULDIN_KMEDOID<-
function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
  {

#M1 calcula la distancia promedio entre el centro y las observaciones del cluster
    M1<-c()
    for(i in 1:length(unique(KMEDOID_datos_agrup$cluster)))
    {
      Ci<-subset(KMEDOID_datos_agrup,cluster==i)
      centers<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]

      #le quito la columna cluster
      #junto cada centro con las observaciones de su cluster
      ELEMENTO_CENTRO<-rbind(Ci[,-
which(colnames(Ci)%in%c("cluster"))],centers[i,])

      cuantis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,c(n_cuantis)])
      cualis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,-c(n_cuantis)])

      d<-
distancias(cuantis_ELEMENTO_CENTRO,cualis_ELEMENTO_CENTRO,KPROTO_datos$lambda)
      DISTANC_ELEMENTO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
      M1[i]<-mean(DISTANC_ELEMENTO_CENTRO)
    }

#M2 calcula la distancia entre centros
    M2<-
matrix(0,ncol=length(unique(KMEDOID_datos_agrup$cluster)),nrow=length(unique(KMED
OID_datos_agrup$cluster)))
    for(i in 1:nrow(M2))
    {
      for(j in 1:ncol(M2))
      {
        #juntando centros y le quito la columna cluster
        centers<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]
        CENTRO_CENTRO<-rbind(centers[i,],centers[j,])

        cuantis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,c(n_cuantis)])
        cualis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,-c(n_cuantis)])

        d<-
distancias(cuantis_CENTRO_CENTRO,cualis_CENTRO_CENTRO,KPROTO_datos$lambda)
        DISTANC_CENTRO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
        M2[i,j]<-DISTANC_CENTRO_CENTRO
      }
    }
```

```
#M3 calcula los indices para cada cluster
    M3<-
matrix(0,ncol=length(unique(KMEDOID_datos_agrup$cluster)),nrow=length(unique(KMED
OID_datos_agrup$cluster)))
    for(i in 1:nrow(M3))
    {
      for(j in 1:ncol(M3))
      {
        oi<-M1[i]
        oj<-M1[j]
        d_cicj<-M2[i,j]
        M3[i,j]<-ifelse((oi+oj)/d_cicj==Inf |
(oi+oj)/d_cicj=="NaN",0,(oi+oj)/d_cicj)
      }
    }

    return(mean(apply(M3,1,max)))
  }


######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL ALG. KMEDOIDS Y
OBTENGA LA MEJOR SEMILLA QUE GENERE EL MEJOR ÍNDICE

MEJOR_DAVIES_BOULDIN_KMEDOID<-function(datos_cuantis,datos_cualis,K,SEED)
       {
       indice_DAVIES_BOULDIN_KMEDOID<-list()
       for(i in 1:length(K))
              {
              k<-K[i]
              temp1<-c(0,0,0)
              for(j in 1:length(SEED))
                     {
                     seed<-SEED[j]
                     cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PQ NECESITO EL LAMBDA QUE CALCULA
                     datitos<-cbind(datos_cuantis,datos_cualis)
                     set.seed(seed)
                     KPROTO_datos<-kproto(datitos,k)
#CALCULO LA MATRIZ DE DISTANCIAS
                     d<-distancias(datos_cuantis,datos_cualis,KPROTO_datos$lambda)
#APLICO EL KMEDOID(PAM)
                     set.seed(seed)
                     KMEDOID_datos<-pam(d$distancias,k, diss = TRUE)
#JUNTAR LA TABLA CON LA SEGMENTACIÓN
                     KMEDOID_datos_agrup<-
data.frame(datitos,cluster=KMEDOID_datos$clustering)
                     n_cuantis<-subset(tipo_de_variable(KMEDOID_datos_agrup[,-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE

       indice_DB=DAVIES_BOULDIN_KMEDOID(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_
datos,n_cuantis)
                     temp<-cbind(k,seed,indice_DB)
                     temp1<-rbind(temp1,temp)
                     }
#LE QUITO LA PRIMERA FILA DE MI TEMP
              indice_DAVIES_BOULDIN_KMEDOID[[i]]<-temp1[-1,]
              }
       salida<-do.call("rbind",indice_DAVIES_BOULDIN_KMEDOID)
       minimo<-min(salida[,3])
       mejor<-subset(data.frame(salida),indice_DB==minimo)
       return(list(salida,mejor))
       }


######### SEPARO MIS CUANTIS DE LAS CUALIS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
```

```
K<- 3:13
SEED<-seq(0,100,10)

######### APLICO LA FUNCION
a<-MEJOR_DAVIES_BOULDIN_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]

######### ÍNDICE DE DUNN PARA EL ALGORITMO K MEDOIDS

DUNN_KMEDOID<-function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
  {

#DETERMINAR LA MÁXIMA DISTANCIA EN CADA UNO DE LOS CLUSTERS
    Dist_obs_obs<-function(KMEDOID_datos_agrup,KMEDOID_datos,n_cuantis)
    {
      M1<-c()
      for(i in 1:length(unique(KMEDOID_datos_agrup$cluster)))
      {
        Ci<-subset(KMEDOID_datos_agrup,cluster==i)
        #le quito el cluster
        ELEMENTO_ELEMENTO<-Ci[,-which(colnames(Ci)%in%c("cluster"))]

        cuantis_ELEMENTO_ELEMENTO<- data.frame(ELEMENTO_ELEMENTO[,c(n_cuantis)])
        cualis_ELEMENTO_ELEMENTO<- data.frame(ELEMENTO_ELEMENTO[,-c(n_cuantis)])

        DISTANC_ELEMENTO_ELEMENTO<-
distancias(cuantis_ELEMENTO_ELEMENTO,cualis_ELEMENTO_ELEMENTO,KPROTO_datos$lambda
)
        DISTANC_ELEMENTO_ELEMENTO<-
matrix(do.call("c",DISTANC_ELEMENTO_ELEMENTO),ncol=nrow(ELEMENTO_ELEMENTO),nrow=n
row(ELEMENTO_ELEMENTO))
        M1[i]<-max(apply(as.matrix(DISTANC_ELEMENTO_ELEMENTO),1,max))
      }
      return(max(M1))
    }

    M1<-
Dist_obs_obs(KMEDOID_datos_agrup=KMEDOID_datos_agrup,KMEDOID_datos=KMEDOID_datos,
n_cuantis)

#DETERMINAR LA DISTANCIA ENTRE LOS CENTROS

    Dist_centro_centro<-function(KMEDOID_datos_agrup,KMEDOID_datos,n_cuantis)
    {
      M2<-
matrix(0,ncol=length(unique(KMEDOID_datos_agrup$cluster)),nrow=length(unique(KMED
OID_datos_agrup$cluster)))
      for(i in 1:nrow(M2))
      {
        for(j in 1:ncol(M2))
        {
          #juntando centros
          #le quito la columna cluster

          centers<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]
          CENTRO_CENTRO<-rbind(centers[i,],centers[j,])

          cuantis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,c(n_cuantis)])
          cualis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,-c(n_cuantis)])

          d<-
distancias(cuantis_CENTRO_CENTRO,cualis_CENTRO_CENTRO,KPROTO_datos$lambda)
          DISTANC_CENTRO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
          M2[i,j]<-DISTANC_CENTRO_CENTRO
        }
```

```
      }
      diag(M2)<-rep(Inf,length(diag(M2)))
      return(min(apply(M2,1,min)))
    }

    M2<-
Dist_centro_centro(KMEDOID_datos_agrup=KMEDOID_datos_agrup,KMEDOID_datos=KMEDOID_
datos,n_cuantis)
    DUNN<- function(M1,M2)
    {
      return(M2/M1)
    }
    DUNN<-DUNN(M1,M2)

    return(DUNN)
  }


######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL ALG. KMEDOIDS Y
OBTENGA LA MEJOR SEMILLA QUE DÉ EL MEJOR ÍNDICE

MEJOR_DUNN_KMEDOID<-function(datos_cuantis,datos_cualis,K,SEED)
       {
       indice_DUNN_KMEDOID<-list()
       for(i in 1:length(K))
              {
              k<-K[i]
              temp1<-c(0,0,0)
              for(j in 1:length(SEED))
                     {
                     seed<-SEED[j]
                     cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PQ NECESITO EL LAMBDA QUE CALCULA
                     datitos<-cbind(datos_cuantis,datos_cualis)
                     set.seed(seed)
                     KPROTO_datos<-kproto(datitos,k)
#CALCULO LA MATRIZ DE DISTANCIAS
                     d<-distancias(datos_cuantis,datos_cualis,KPROTO_datos$lambda)
#APLICO EL KMEDOID(PAM)
                     set.seed(seed)
                     KMEDOID_datos<-pam(d$distancias,k, diss = TRUE)
#JUNTAR LA TABLA CON LA SEGMENTACIÓN
                     KMEDOID_datos_agrup<-
data.frame(datitos,cluster=KMEDOID_datos$clustering)
                     n_cuantis<-subset(tipo_de_variable(KMEDOID_datos_agrup[,-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE

       indice_DN=DUNN_KMEDOID(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cu
antis)
                     temp<-cbind(k,seed,indice_DN)
                     temp1<-rbind(temp1,temp)
                     }
#LE QUITO LA PRIMERA FILA DE MI TEMP
              indice_DUNN_KMEDOID[[i]]<-temp1[-1,]
              }
       salida<-do.call("rbind",indice_DUNN_KMEDOID)
       maximo<-max(salida[,3])
       mejor<-subset(data.frame(salida),indice_DN==maximo)
       return(list(salida,mejor))
       }


######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 3:13
SEED<-seq(0,100,10)
```

```
######### APLICO LA FUNCION
a<-MEJOR_DUNN_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]


######### CÁLCULO DE SSW PARA EL ALGORITMO K MEDOIDS

SSW_KMEDOID<-function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
  {
    Dist_obs_centro<-function(KMEDOID_datos_agrup,KMEDOID_datos,n_cuantis)
    {
      M1<-c()
      for(i in 1:length(unique(KMEDOID_datos_agrup$cluster)))
      {

        Ci<-subset(KMEDOID_datos_agrup,cluster==i)

        centers<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]
        ELEMENTO_CENTRO<-rbind(Ci[,-
which(colnames(Ci)%in%c("cluster"))],centers[i,])

        cuantis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,c(n_cuantis)])
        cualis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,-c(n_cuantis)])

        d<-
distancias(cuantis_ELEMENTO_CENTRO,cualis_ELEMENTO_CENTRO,KPROTO_datos$lambda)
        d<-d$distancias^2
        DISTANC_ELEMENTO_CENTRO<-d[nrow(d),-nrow(d)]
        M1[i]<-sum(DISTANC_ELEMENTO_CENTRO)
      }
      return(sum(M1))
    }
    M2<-Dist_obs_centro(KMEDOID_datos_agrup,KMEDOID_datos,n_cuantis)
    return(M2)
  }


######### CÁLCULO DE SSB PARA EL ALGORITMO K KMEDOIDS

SSB_KMEDOID<-function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
  {
    Dist_centro_centraso<-
function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
    {

      set.seed(4)
      cuantis_datos<-KMEDOID_datos_agrup[,c(n_cuantis)]
      cualis_datos<-KMEDOID_datos_agrup[,-
c(n_cuantis,which(colnames(KMEDOID_datos_agrup)%in%c("cluster")))]
      d<-distancias(cuantis_datos,cualis_datos,KPROTO_datos$lambda)
      KMEDOID_datos_1<-pam(d$distancias, 1, diss = TRUE)

      centrazo<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos_1$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]
      centers<-KMEDOID_datos_agrup[as.numeric(KMEDOID_datos$medoids),-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]

      centers_centraso<-rbind(centers,centrazo)

      cuantis_centers_centraso<- data.frame(centers_centraso[,c(n_cuantis)])
      cualis_centers_centraso<- data.frame(centers_centraso[,-c(n_cuantis)])

      d<-
distancias(cuantis_centers_centraso,cualis_centers_centraso,KPROTO_datos$lambda)
      d<-d$distancias^2
```

```
      DISTANC_centers_centraso<-table(KMEDOID_datos$cluster)*d[nrow(d),-nrow(d)]

      M1<-sum(DISTANC_centers_centraso)
      return(M1)

    }

    M2<-
Dist_centro_centraso(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
    return(M2)
  }


######### ÍNDICE DE CALINSHI PARA EL ALGORTIMO K MEDOIDS

CALINSKI_HARABASZ_KMEDOID<-
function(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
       {
       cuantis_datos<-KMEDOID_datos_agrup[,c(n_cuantis)]
       cualis_datos<-KMEDOID_datos_agrup[,-
c(n_cuantis,which(colnames(KMEDOID_datos_agrup)%in%c("cluster")))]

       d<-distancias(cuantis_datos,cualis_datos,KPROTO_datos$lambda)

       ssw_kmedoid<-
SSW_KMEDOID(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)
       ssb_kmedoid<-
SSB_KMEDOID(KMEDOID_datos_agrup,KMEDOID_datos,KPROTO_datos,n_cuantis)

       N<-nrow(KMEDOID_datos_agrup)
       M<-length(unique(KMEDOID_datos_agrup$cluster))
       Calinski_Harabasz_kmedoid<-(ssb_kmedoid/(M-1))/(ssw_kmedoid/(N-M))

       return(list(SSB=ssb_kmedoid,SSW=ssw_kmedoid,indice=Calinski_Harabasz_kmedo
id))
       }


######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL KMEDOIDS Y OBTENGA
LA MEJOR SEMILLA QUE DÉ EL MEJOR INDICE

MEJOR_CALINSKI_HARABASZ_KMEDOID<-function(datos_cuantis,datos_cualis,K,SEED)
       {
       indice_CALINSKI_HARABASZ_KMEDOID<-list()
       for(i in 1:length(K))
              {
              k<-K[i]
              temp1<-c(0,0,0)
              for(j in 1:length(SEED))
                     {
                     seed<-SEED[j]
                     cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PQ NECESITO EL LAMBDA QUE CALCULA
                     datitos<-cbind(datos_cuantis,datos_cualis)
                     set.seed(seed)
                     KPROTO_datos<-kproto(datitos,k)
#CALCULO LA MATRIZ DE DISTANCIAS
                     d<-distancias(datos_cuantis,datos_cualis,KPROTO_datos$lambda)
#APLICO EL KMEDOID(PAM)
                     set.seed(seed)
                     KMEDOID_datos<-pam(d$distancias,k, diss = TRUE)
                     #JUNTAR LA TABLA CON LA SEGMENTACIÓN
                     KMEDOID_datos_agrup<-
data.frame(datitos,cluster=KMEDOID_datos$clustering)
                     n_cuantis<-subset(tipo_de_variable(KMEDOID_datos_agrup[,-
which(colnames(KMEDOID_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE
```

```
       indice_CALINSKI_HARABASZ=CALINSKI_HARABASZ_KMEDOID(KMEDOID_datos_agrup,KME
DOID_datos,KPROTO_datos,n_cuantis)$indice
                    temp<-cbind(k,seed,indice_CALINSKI_HARABASZ)
                    temp1<-rbind(temp1,temp)
                    }
#LE QUITO LA PRIMERA FILA DE TEMP1
              indice_CALINSKI_HARABASZ_KMEDOID[[i]]<-temp1[-1,]
              }
       salida<-do.call("rbind",indice_CALINSKI_HARABASZ_KMEDOID)
       maximo<-max(salida[,3])
       mejor<-subset(data.frame(salida),indice_CALINSKI_HARABASZ==maximo)
       return(list(salida,mejor))
       }


######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 3:13
SEED<-seq(0,100,10)


######### APLICO LA FUNCIÓN
a<-MEJOR_CALINSKI_HARABASZ_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]


######################################################################
### DETERMINAR EL NÚMERO DE CLUSTERS PARA EL ALGORITMO K-PROTOTYPE ###
######################################################################


######### INDICE DE DAVID BOUDIN PARA EL ALGORITMO K PROTOTYPE

DAVIES_BOULDIN_KPROTO<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
  {

#M1 calcula la distancia promedio entre el centro y las observaciones del cluster
    M1<-c()
    for(i in 1:length(unique(KPROTO_datos_agrup$cluster)))
      #i<-1
    {
      Ci<-subset(KPROTO_datos_agrup,cluster==i)

      #le quito la columna cluster
      #junto cada centro con las observaciones de su cluster
      ELEMENTO_CENTRO<-rbind(Ci[,-
which(colnames(Ci)%in%c("cluster"))],KPROTO_datos$centers[i,])

      cuantis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,c(n_cuantis)])
      cualis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,-c(n_cuantis)])

      d<-
distancias(cuantis_ELEMENTO_CENTRO,cualis_ELEMENTO_CENTRO,KPROTO_datos$lambda)
      DISTANC_ELEMENTO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
      M1[i]<-mean(DISTANC_ELEMENTO_CENTRO)
    }

#M2 calcula la distancia entre centros
    M2<-
matrix(0,ncol=length(unique(KPROTO_datos_agrup$cluster)),nrow=length(unique(KPROT
O_datos_agrup$cluster)))
    for(i in 1:nrow(M2))
    {
      for(j in 1:ncol(M2))
```

```
      {
        #juntando centros y le quito la columna cluster
        CENTRO_CENTRO<-rbind(KPROTO_datos$centers[i,],KPROTO_datos$centers[j,])

        cuantis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,c(n_cuantis)])
        #le quito el cluster
        cualis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,-c(n_cuantis)])

        d<-
distancias(cuantis_CENTRO_CENTRO,cualis_CENTRO_CENTRO,KPROTO_datos$lambda)
        DISTANC_CENTRO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
        M2[i,j]<-DISTANC_CENTRO_CENTRO
      }
    }

#M3 calcula los indices para cada cluster
    M3<-
matrix(0,ncol=length(unique(KPROTO_datos_agrup$cluster)),nrow=length(unique(KPROT
O_datos_agrup$cluster)))
    for(i in 1:nrow(M3))
    {
      for(j in 1:ncol(M3))
      {
        oi<-M1[i]
        oj<-M1[j]
        d_cicj<-M2[i,j]
        M3[i,j]<-ifelse((oi+oj)/d_cicj==Inf |
(oi+oj)/d_cicj=="NaN",0,(oi+oj)/d_cicj)
      }
    }

    return(mean(apply(M3,1,max)))
  }

######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL KPROTOTYPE Y OBTENGA
LA MEJOR SEMILLA QUE DÉ EL MEJOR INDICE

MEJOR_DAVIES_BOULDIN_KPROTO<-function(datos_cuantis,datos_cualis,K,SEED)
       {
       indice_DAVIES_BOULDIN_KPROTO<-list()
       for(i in 1:length(K))
              {
              k<-K[i]
              temp1<-c(0,0,0)
              for(j in 1:length(SEED))
                     {
                     seed<-SEED[j]
                     cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PARA CALCULAR EL LAMBDA
                     datitos<-cbind(datos_cuantis,datos_cualis)
                     set.seed(seed)
                     KPROTO_datos<-kproto(datitos,k)
#LAS DISTANCIAS SON CALCULADAS INTERNAMENTE
                     #JUNTAR LA TABLA CON LA SEGMENTACIÓN
                     KPROTO_datos_agrup<-
data.frame(datitos,cluster=KPROTO_datos$cluster)
                     n_cuantis<-subset(tipo_de_variable(KPROTO_datos_agrup[,-
which(colnames(KPROTO_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE

       indice_DB=DAVIES_BOULDIN_KPROTO(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
                     temp<-cbind(k,seed,indice_DB)
                     temp1<-rbind(temp1,temp)
                     }
#LE QUITO LA PRIMERA FILA DE TEMP
              indice_DAVIES_BOULDIN_KPROTO[[i]]<-temp1[-1,]
              }
```

```
       salida<-do.call("rbind",indice_DAVIES_BOULDIN_KPROTO)
       minimo<-min(salida[,3])
       mejor<-subset(data.frame(salida),indice_DB==minimo)
       return(list(salida,mejor))
       }

######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 3:13
SEED<-seq(0,100,10)

######### APLICO LA FUNCIÓN
a<-MEJOR_DAVIES_BOULDIN_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]


######### ÍNDICE DE DUNN PARA EL ALGORITMO K KPROTOTYPE

DUNN_KPROTO<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
  {

#DETERMINAR LA MÁXIMA DISTANCIA EN CADA UNO DE LOS CLUSTERS
    Dist_obs_obs<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
    {
      M1<-c()
      for(i in 1:length(unique(KPROTO_datos_agrup$cluster)))
      {
        Ci<-subset(KPROTO_datos_agrup,cluster==i)
        ELEMENTO_ELEMENTO<-Ci[,-which(colnames(Ci)%in%c("cluster"))]

        cuantis_ELEMENTO_ELEMENTO<- data.frame(ELEMENTO_ELEMENTO[,c(n_cuantis)])
        cualis_ELEMENTO_ELEMENTO<- data.frame(ELEMENTO_ELEMENTO[,-c(n_cuantis)])

        DISTANC_ELEMENTO_ELEMENTO<-
distancias(cuantis_ELEMENTO_ELEMENTO,cualis_ELEMENTO_ELEMENTO,KPROTO_datos$lambda
)
        DISTANC_ELEMENTO_ELEMENTO<-
matrix(do.call("c",DISTANC_ELEMENTO_ELEMENTO),ncol=nrow(ELEMENTO_ELEMENTO),nrow=n
row(ELEMENTO_ELEMENTO))
        M1[i]<-max(apply(as.matrix(DISTANC_ELEMENTO_ELEMENTO),1,max))
      }
       return(max(M1))
    }

    M1<-
Dist_obs_obs(KPROTO_datos_agrup=KPROTO_datos_agrup,KPROTO_datos=KPROTO_datos,n_cu
antis)

#DETERMINAR LA DISTANCIA ENTRE LOS CENTROS

    Dist_centro_centro<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis){
      M2<-
matrix(0,ncol=length(unique(KPROTO_datos_agrup$cluster)),nrow=length(unique(KPROT
O_datos_agrup$cluster)))

      for(i in 1:nrow(M2))
      {
        for(j in 1:ncol(M2))
        {
          CENTRO_CENTRO<-rbind(KPROTO_datos$centers[i,],KPROTO_datos$centers[j,])

          cuantis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,c(n_cuantis)])
          cualis_CENTRO_CENTRO<- data.frame(CENTRO_CENTRO[,-c(n_cuantis)])

          d<-
distancias(cuantis_CENTRO_CENTRO,cualis_CENTRO_CENTRO,KPROTO_datos$lambda)
```

```
          DISTANC_CENTRO_CENTRO<-d$distancias[nrow(d$distancias),-
nrow(d$distancias)]
          M2[i,j]<-DISTANC_CENTRO_CENTRO
        }
      }
      diag(M2)<-rep(Inf,length(diag(M2)))
      return(min(apply(M2,1,min)))
    }

    M2<-
Dist_centro_centro(KPROTO_datos_agrup=KPROTO_datos_agrup,KPROTO_datos=KPROTO_dato
s,n_cuantis)

    DUNN<- function(M1,M2)
    {
      return(M2/M1)
    }
    DUNN<-DUNN(M1,M2)
    return(DUNN)
  }


######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL KPROTOTYPE Y OBTENGA
LA MEJOR SEMILLA QUE DÉ EL MEJOR INDICE

MEJOR_DUNN_KPROTO<-function(datos_cuantis,datos_cualis,K,SEED)
      {
      indice_DUNN_KPROTO<-list()
      for(i in 1:length(K))
            {
            k<-K[i]
            temp1<-c(0,0,0)
            for(j in 1:length(SEED))
                  {
                  seed<-SEED[j]
                  cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PQ NECESITO EL LAMBDA QUE CALCULA
                  datitos<-cbind(datos_cuantis,datos_cualis)
                  set.seed(seed)
                  KPROTO_datos<-kproto(datitos,k)
#LAS DISTANCIAS SON CALCULADAS INTERNAMENTE
                  #JUNTAR LA TABLA CON LA SEGMENTACIÓN
                  KPROTO_datos_agrup<-
data.frame(datitos,cluster=KPROTO_datos$cluster)
                  n_cuantis<-subset(tipo_de_variable(KPROTO_datos_agrup[,-
which(colnames(KPROTO_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE

      indice_DN=DUNN_KPROTO(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
                  temp<-cbind(k,seed,indice_DN)
                  temp1<-rbind(temp1,temp)
                  }
#LE QUITO LA PRIMERA FILA DE MI TEMP
            indice_DUNN_KPROTO[[i]]<-temp1[-1,]
            }
      salida<-do.call("rbind",indice_DUNN_KPROTO)
      maximo<-max(salida[,3])
      mejor<-subset(data.frame(salida),indice_DN==maximo)
      return(list(salida,mejor))
      }


######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 3:13
SEED<-seq(0,100,20)
```

```
######### APLICO LA FUNCIÓN
a<-MEJOR_DUNN_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]


######### CÁLCULO DE SSW PARA EL ALGORITMO K KPROTOTYPE

SSW_KPROTO<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
  {

    Dist_obs_centro<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
    {
      M1<-c()
      for(i in 1:length(unique(KPROTO_datos_agrup$cluster)))
      {
        Ci<-subset(KPROTO_datos_agrup,cluster==i)

        ELEMENTO_CENTRO<-rbind(Ci[,-
which(colnames(Ci)%in%c("cluster"))],KPROTO_datos$centers[i,])

        cuantis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,c(n_cuantis)])
        cualis_ELEMENTO_CENTRO<- data.frame(ELEMENTO_CENTRO[,-c(n_cuantis)])

        d<-
distancias(cuantis_ELEMENTO_CENTRO,cualis_ELEMENTO_CENTRO,KPROTO_datos$lambda)
        d<-d$distancias^2
        DISTANC_ELEMENTO_CENTRO<-d[nrow(d),-nrow(d)]
        M1[i]<-sum(DISTANC_ELEMENTO_CENTRO)
      }
      return(sum(M1))
    }
    M2<-Dist_obs_centro(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
    return(M2)
  }


######### CÁLCULO DE SSB PARA EL ALGORITMO K KPROTOTYPE

SSB_KPROTO<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
  {
    Dist_centro_centraso<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
    {


      set.seed(4)
      KPROTO_datos_1<-kproto(KPROTO_datos_agrup[,-
which(colnames(KPROTO_datos_agrup)%in%c("cluster"))],1)
      centrazo<-KPROTO_datos_1$centers
      centers_centraso<-rbind(KPROTO_datos$centers,centrazo)

      cuantis_centers_centraso<- data.frame(centers_centraso[,c(n_cuantis)])
      cualis_centers_centraso<- data.frame(centers_centraso[,-c(n_cuantis)])

      d<-
distancias(cuantis_centers_centraso,cualis_centers_centraso,KPROTO_datos$lambda)
      d<-d$distancias^2

      DISTANC_centers_centraso<-table(KPROTO_datos$cluster)*d[nrow(d),-nrow(d)]

      M1<-sum(DISTANC_centers_centraso)
      return(M1)

    }

    M2<-Dist_centro_centraso(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
    return(M2)
  }
```

```
######### ÍNDICE DE CALINSHI PARA EL ALGORTIMO K PROTOTYPES

CALINSKI_HARABASZ_KPROTO<-function(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
       {
       cuantis_datos<-KPROTO_datos_agrup[,c(n_cuantis)]
       cualis_datos<-KPROTO_datos_agrup[,-
c(n_cuantis,which(colnames(KPROTO_datos_agrup)%in%c("cluster")))]

       d<-distancias(cuantis_datos,cualis_datos,KPROTO_datos$lambda)

       ssw_kproto<-SSW_KPROTO(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)
       ssb_kproto<-SSB_KPROTO(KPROTO_datos_agrup,KPROTO_datos,n_cuantis)

       N<-nrow(KPROTO_datos_agrup)
       M<-length(unique(KPROTO_datos_agrup$cluster))
       Calinski_Harabasz_kproto<-(ssb_kproto/(M-1))/(ssw_kproto/(N-M))

       return(list(SSB=ssb_kproto,SSW=ssw_kproto,indice=Calinski_Harabasz_kproto)
)
       }

######### SIGUIENTE PASO ES HACER UNA FUNCIÓN QUE APLIQUE EL PROTOTYPES Y OBTENGA
LA MEJOR SEMILLA QUE DÉ EL MEJOR INDICE

MEJOR_CALINSKI_HARABASZ_KPROTO<-function(datos_cuantis,datos_cualis,K,SEED)
       {
       indice_CALINSKI_HARABASZ_KPROTO<-list()
       for(i in 1:length(K))
              {
              k<-K[i]
              temp1<-c(0,0,0)
              for(j in 1:length(SEED))
                     {
                     seed<-SEED[j]
                     cat("iteracion=",i,".",j,"seed=",seed," N=",k,"\n")
#APLICAR EL KPROTO PQ NECESITO EL LAMBDA QUE CALCULA
                     datitos<-cbind(datos_cuantis,datos_cualis)
                     set.seed(seed)
                     KPROTO_datos<-kproto(datitos,k)
#LAS DISTANCIAS SON CALCULADAS INTERNAMENTE
                     #JUNTAR LA TABLA CON LA SEGMENTACIÓN
                     KPROTO_datos_agrup<-
data.frame(datitos,cluster=KPROTO_datos$cluster)
                     n_cuantis<-subset(tipo_de_variable(KPROTO_datos_agrup[,-
which(colnames(KPROTO_datos_agrup)%in%c("cluster"))]),tipo=="numeric")$orden
#CALCULAR EL ÍNDICE

       indice_CALINSKI_HARABASZ=CALINSKI_HARABASZ_KPROTO(KPROTO_datos_agrup,KPROT
O_datos,n_cuantis)$indice
                     temp<-cbind(k,seed,indice_CALINSKI_HARABASZ)
                     temp1<-rbind(temp1,temp)
                     }
#LE QUITO LA PRIMERA FILA DE MI TEMP1
              indice_CALINSKI_HARABASZ_KPROTO[[i]]<-temp1[-1,]
              }
       salida<-do.call("rbind",indice_CALINSKI_HARABASZ_KPROTO)
       maximo<-max(salida[,3])
       mejor<-subset(data.frame(salida),indice_CALINSKI_HARABASZ==maximo)
       return(list(salida,mejor))
       }



######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 3:13
SEED<-seq(0,100,20)
```

```
######### APLICO LA FUNCIÓN
a<-MEJOR_CALINSKI_HARABASZ_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a
a[[1]]



################################################################
### PROCESAMIENTO DE TODOS LOS ÍNDICES DE AMBOS ALGORITMOS ###
################################################################


######### SEPARO LAS VARIABLES CUANTITATIVAS DE LA CUALITATIVAS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]
K<- 2:13
SEED<-seq(0,100,10)

######### APLICO LA FUNCIÓN
a1<-MEJOR_DAVIES_BOULDIN_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a1
######### APLICO LA FUNCION
a2<-MEJOR_DUNN_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a2

######### APLICO LA FUNCIÓN
a4<-MEJOR_CALINSKI_HARABASZ_KMEDOID(datos_cuantis,datos_cualis,K,SEED)
a4

######### APLICO LA FUNCIÓN
a5<-MEJOR_DAVIES_BOULDIN_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a5

######### APLICO LA FUNCIÓN
a6<-MEJOR_DUNN_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a6

######### APLICO LA FUNCIÓN
a8<-MEJOR_CALINSKI_HARABASZ_KPROTO(datos_cuantis,datos_cualis,K,SEED)
a8


b1<-a1[[1]][order(a1[[1]][,3], decreasing = FALSE),]
#MEJOR_DAVIES_BOULDIN_KMEDOID
b2<-a2[[1]][order(a2[[1]][,3], decreasing = TRUE),] #MEJOR_DUNN_KMEDOID
b4<-a4[[1]][order(a4[[1]][,3], decreasing = TRUE),]
#MEJOR_CALINSKI_HARABASZ_KMEDOID
b5<-a5[[1]][order(a5[[1]][,3], decreasing = FALSE),] #MEJOR_DAVIES_BOULDIN_KPROTO
b6<-a6[[1]][order(a6[[1]][,3], decreasing = TRUE),] #MEJOR_DUNN_KPROTO
b8<-a8[[1]][order(a8[[1]][,3], decreasing = TRUE),]
#MEJOR_CALINSKI_HARABASZ_KPROTO



data.frame(b1,b2,b3,b4)
data.frame(b5,b6,b7,b8)


#####################################################################
### SABIENDO CUAL ES EL MEJOR K Y SEED SE APLICAN LOS ALGORITMOS ###
#####################################################################

###################################
### PARA EL ALGORITMO K-MEDOID ###
###################################

######### CANTIDAD DE CLUSTERS (K)
K<-3
```

```
######### SEPARO MIS CUANTIS DE LA CUALIS
datos_cuantis<-datitos[,-c(15:ncol(datitos))]
datos_cualis<-datitos[,c(15:ncol(datitos))]

######### APLICAR EL KPROTO PARA OBTENER EL LAMBDA
set.seed(0)
KPROTO_datos<-kproto(datitos,k=K)

######### CALCULAR LA MATRIZ DE DISTANCIAS
d<-distancias(datos_cuantis,datos_cualis,KPROTO_datos$lambda)

######### APLICO EL KMEDOID(PAM)
set.seed(0)
KMEDOID_datos<-pam(d$distancias,K, diss = TRUE)
KMEDOID_datos$medoids

######### ASIGNO CADA REGISTRO A SU CLUSTERING CON LA DATA REESCALADA (OJO)
KMEDOID_datos_agrup<-data.frame(datos,cluster=KMEDOID_datos$clustering)


#####################################
### PARA EL ALGORITMO K-KPROTOTYPE ###
#####################################

######### CANTIDAD DE CLUSTERS (K)
K<-5

######### APLICO EL KPROTOTYPE
set.seed(90)
KPROTO_datos<-kproto(datitos,k=K)
KPROTO_datos$centers

######### ASIGNO CADA REGISTRO A SU CLUSTERING CON LA DATA REESCALADA (OJO)
KPROTO_datos_agrup<-data.frame(datos,cluster=KPROTO_datos$cluster)



#########################################################
### OBTENER LOS CENTROS PARA EL ALGORITMO  K-MEDOID ###
#########################################################

datos[KMEDOID_datos$medoids,]

############################################################
### OBTENER LOS CENTROS PARA EL ALGORITMO K-KPROTOTYPE ###
############################################################


######### REESCALAMIENTO DE VARIABLES CANTITATIVAS #x= y*(max(x)-min(x)) +min(x))

######### DATOS INICIALES
cualis_datos<-datos[,c(15:ncol(datos))]
cuantis_datos<-datos[,-c(15:ncol(datos))]

######### DATOS ESCALADOS
cualis_datos_desnorm<-KPROTO_datos$centers[,c(15:ncol(datos))]
cuantis_datos_desnorm<-KPROTO_datos$centers[,-c(15:ncol(datos))]
######### REESCALAMIENTO DE LOS CENTROS
for(i in 1:ncol(cuantis_datos_desnorm))
      {
      cuantis_datos_desnorm[,i]<-
cuantis_datos_desnorm[,i]*(as.numeric(max(cuantis_datos[,i])-
min(cuantis_datos[,i])))+min(cuantis_datos[,i])
      }

#########JUNTANDO LAS VARIABLES REESCALADAS Y LAS CUALITATIVAS
centers<-data.frame(cuantis_datos_desnorm,cualis_datos_desnorm)
```

```
#########################################################
### VALIDACIÓN DE RESULTADOS: ANOVA Y CHI_CUADRADO ###
#########################################################

######### FUNCIÓN PARA VERIFICAR SI UNA VARIABLE CUANTITATIVA APORTA O NO

ANOVA<- function(datos_cuantis, cluster)
       {
       resultado<-list()
       for (i in 1:ncol(datos_cuantis))
              {
              variable<-colnames(datos_cuantis)[i]
              modelo<-lm(datos_cuantis[,i] ~ cluster)
              anova<-anova(modelo)
              GL<-anova$'Df'
              SC<-anova$'Sum Sq'
              F<-anova$'F value'
              P<-anova$'Pr(>F)'

              if((P>=0) && (P<=0.001)) {
              signo<-'***'
              } else if((P>=0.001) && (P<=0.01)){
              signo<-'**'
              } else if((P>=0.01) && (P<=0.05)){
              signo<-'.'
              }else{
              signo<-''
              }

              nombre<- c("Entre grupos","Dentro de grupos","Total")

              tabla<-
data.frame(Variable=rep(variable,3),Varianza=nombre,Suma_de_cuadrados=round(c(SC,
sum(SC)),3), 'gl'=c(GL,sum(GL)),

       Media_cuadrática=c(round(SC/GL,3),''),'F_value'=c(round(F[1],3),'',''),"Pr
ob"=c(P[1],'',''),'Sig'=c(signo,'',''))
              resultado[[i]]<-tabla
              }
       salida=do.call("rbind",resultado)
       return(salida)
       }

######### VARIABLES ORIGINALES SEPARADAS POR CUALITATIVAS Y CUANTITATIVAS

######### DATOS INICIALES
cualis_datos<-datos[,c(15:ncol(datos))]
cuantis_datos<-datos[,-c(15:ncol(datos))]

######### CLUSTERS FINALES
cluster<-KMEDOID_datos_agrup$cluster
cluster<-as.factor(cluster)

######### ANOVA POR VARIBLE FRENTE AL CLUSTER
ANOVA(cuantis_datos, cluster)
######### FUNCIÓN PARA VERIFICAR SI UNA VARIABLE CUALITATIVA APORTA O NO

CHI_CUADRADO<- function(datos_cualis, cluster)
       {
       resultado<-list()
       for (i in 1:ncol(datos_cualis))
              {
              variable<-colnames(datos_cualis)[i]
              prueba<-chisq.test(datos_cualis[,i],cluster)
```

```
              X2<-prueba$statistic
              GL<-prueba$parameter
              P<-prueba$p.value

              if((P>=0) && (P<=0.001)) {
              signo<-'***'
              } else if((P>=0.001) && (P<=0.01)){
              signo<-'**'
              } else if((P>=0.01) && (P<=0.05)){
              signo<-'.'
              }else{
              signo<-''
              }

              nombre<- c("Chi-cuadrado de Pearson")

              tabla<-
data.frame(Variable=variable,Prueba=nombre,Valor=round(X2,3),gl=GL,Prob=P,Sig=sig
no)
              resultado[[i]]<-tabla
              }
       salida=do.call("rbind",resultado)
       return(salida)
       }

######### VARIABLES ORIGINALES SEPARADAS POR CUALITATIVAS Y CUANTITATIVAS

######### DATOS INICIALES
cualis_datos<-datos[,c(15:ncol(datos))]
cuantis_datos<-datos[,-c(15:ncol(datos))]

######### CLUSTERS FINALES
cluster<-KMEDOID_datos_agrup$cluster
cluster<-as.factor(cluster)

######### PRUEBA CHI CUADRADO POR VARIBLE FRENTE AL CLUSTER
CHI_CUADRADO(datos_cualis, cluster)


#####################################################
### VALIDACIÓN DE RESULTADOS: ARBOL DE DECISIÓN ###
#####################################################

######### SEPARAR VARIABLES CUALITATIVAS Y CUANTITATIVAS PARA DAR FORMATO
tipo_de_variable(KMEDOID_datos_agrup)
cualis_datos<-KMEDOID_datos_agrup[,c(15:ncol(KMEDOID_datos_agrup))]
cuantis_datos<-KMEDOID_datos_agrup[,-c(15:ncol(KMEDOID_datos_agrup))]

summary(cualis_datos)
summary(cuantis_datos)
tipo_de_variable(cualis_datos)
tipo_de_variable(cuantis_datos)

######### TRANSFORMAR LAS VARIABLES CUALITATIVAS EN FACTOR
for(i in 1: ncol(cualis_datos))
       {
       cualis_datos[,i]<-as.factor(cualis_datos[,i])
       }
######## CONSOLIDADNDO TODOS LOS DATOS
datitos<- data.frame(cuantis_datos,cualis_datos)
tipo_de_variable(datitos)


######## ENCONTRAR EL NÚMERO DE VARIABLES ÓPTIMO A UTILIZAR EN CADA NODO

oob.err=double(21)

##mtry es el número de vartibales en cada split
```

```
for(mtry in 1:21)
       {
       set.seed(27)
       rf=randomForest(cluster~., data = datitos, importance =
TRUE,mtry=mtry,ntree =500)
       oob.err[mtry] = mean(rf$err.rate[,1])

       cat(mtry,"\n") #printing the output to the console
       }
######## GRÁFICO PARA DETERMINAR EL mtry ÓPTIMO

matplot(1:mtry , xaxt="n",ylim=c(0.1,0.15), oob.err, pch=19 ,
col=c("red"),type="b",ylab="Tasa de Error",xlab="Número de predictores
considerados en cada rama")
axis(1, at = seq(1, 21, by = 1), las=2)
legend("topright",legend=c("Validación Cruzada"),pch=19, col=c("red"))



######## ALGORITMO BORUTA
######## PARA LA SELECCIÓN DE VARIABLES

set.seed(27)
boruta.train <- Boruta(cluster~., data = datitos, doTrace = 2,ntree=500,mtry=2)
#doTrace: muestra avisos al ejecutar

final.boruta <- TentativeRoughFix(boruta.train)
print(final.boruta)

plot(final.boruta, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(final.boruta$ImpHistory),function(i)
final.boruta$ImpHistory[is.finite(final.boruta$ImpHistory[,i]),i])
names(lz) <- colnames(final.boruta$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
at = 1:ncol(final.boruta$ImpHistory), cex.axis = 0.7)


#OBTENER LA LISTA DE ATRIBUTOS CONFIRMADOS
getSelectedAttributes(final.boruta, withTentative = F)

#Crearemos un marco de datos del resultado final derivado de Boruta
boruta.df <- attStats(final.boruta)
class(boruta.df)
print(boruta.df)


######## ALGORITMO RANDOM FOREST
######## VALIDACIÓN DE LOS CLUSTERS

##### ENCONTRAR EL NÚMERO DE VARIABLES ÓPTIMO A UTILIZAR EN CADA NODO

oob.err=double(19)

##mtry es el número de vartibales en cada split
for(mtry in 1:19)
       {
       set.seed(27)
       rf=randomForest(cluster~NOTA_COLEGIO+NOTA_ADMISION+CTA_COLEGIO+AÑOS_COLEGI
O_ADMISION+APORTE_SEMESTRAL+EDAD_ADMISION+COM_COLEGIO+TIPO_COLEGIO+MAT_COLEGIO+FI
S_ADMISION+SEXO+TERCIO_SUPERIOR_ESP+MODALIDAD+RM_ADMISION+QUI_ADMISION+ELECCION_E
SP_INGRESO+MAT_ADMISION+BIO_ADMISION+ESPECIALIDAD,
              data = datitos, importance = TRUE,mtry=mtry,ntree =500)
       oob.err[mtry] = mean(rf$err.rate[,1])

       cat(mtry,"\n") #printing the output to the console
       }
```

```
######## GRÁFICO PARA DETERMINAR EL mtry ÓPTIMO

matplot(1:mtry , xaxt="n",ylim=c(0.1,0.14), oob.err, pch=19 ,
col=c("red"),type="b",ylab="Tasa de Error",xlab="Número de predictores
considerados en cada rama")
axis(1, at = seq(1, 19, by = 1), las=2)
legend("topright",legend=c("Validación Cruzada"),pch=19, col=c("red"))


######## MODELO FINAL

set.seed(27)
rf_out <-
randomForest(cluster~NOTA_COLEGIO+NOTA_ADMISION+CTA_COLEGIO+AÑOS_COLEGIO_ADMISION
+APORTE_SEMESTRAL+EDAD_ADMISION+COM_COLEGIO+TIPO_COLEGIO+MAT_COLEGIO+FIS_ADMISION
+SEXO+TERCIO_SUPERIOR_ESP+MODALIDAD+RM_ADMISION+QUI_ADMISION+ELECCION_ESP_INGRESO
+MAT_ADMISION+BIO_ADMISION+ESPECIALIDAD,
                    data = datitos, importance = TRUE,mtry=3,ntree =500)

######## ERROR ESTIMADO POR VALIDACIÓN CRUZADA
yhat=rf_out$predicted
y=datitos$cluster # training data
mean(y != yhat)


######## ALGORITMO C5.0

######## PARA OBTENER REGLAS DE CLASIFICACIÓN

######## SEPARAR EN DATA TEST Y TRAINING PARA DETERMINAR LOS PARÁMETROS ÓPTIMOS

set.seed(27)
intrain<-createDataPartition(y=datitos$cluster,p=0.9,list=FALSE)
training<-datitos[intrain,]
testing<-datitos[-intrain,]


######## PROPORCIÓN DE OBSERVACIONES POR CLUSTER

round(prop.table(table(training$cluster))*100,1)
round(prop.table(table(testing$cluster))*100,1)


####### PARÁMETROS DEL MODELO PODADO

CF_vector <- seq(0.1,0.9,0.15)
minCases_vector <- c(2,5,10,15,20,25)

#CF factor de confianza
#La opción CF afecta la manera en que se estiman las tasas de error y, por lo
tanto, la gravedad de la poda
#los valores más pequeños que el valor predeterminado (25%) hacen que se recorte
más parte del árbol inicial, mientras que los valores más grandes producen menos
poda.



####### PERMUTACIONES DEL 1 AL 5 EN PARES
x <- 1:length(CF_vector)
indice<-permutations(n=length(x),r=2,v=x,repeats.allowed=T)
resultado<- matrix(0,nrow(indice),ncol(indice)+1)

####### ITERACIÓN DE LOS MODELOS PARA ENCONTRAR LOS PARÁMETROS ÓPTIMOS
for(i in 1:nrow(indice))
      {
      modelo<-
C5.0(formula=cluster~NOTA_COLEGIO+NOTA_ADMISION+CTA_COLEGIO+AÑOS_COLEGIO_ADMISION
```

```
+APORTE_SEMESTRAL+EDAD_ADMISION+COM_COLEGIO+TIPO_COLEGIO+MAT_COLEGIO+FIS_ADMISION
+SEXO+TERCIO_SUPERIOR_ESP+MODALIDAD+RM_ADMISION+QUI_ADMISION+ELECCION_ESP_INGRESO
+MAT_ADMISION+BIO_ADMISION+ESPECIALIDAD,
                    data =training, rules = TRUE,
                    control =C5.0Control(subset = FALSE, CF
=CF_vector[indice[i,1]],
                   minCases = minCases_vector[indice[i,2]],earlyStopping = F))
       estimado<-predict(modelo, newdata = testing[,-22])
       tabla<-table(testing[,22],estimado)
       error<-1-sum(diag(tabla))/(sum(tabla))

       resultado[i,1]<- error
       resultado[i,2]<- CF_vector[indice[i,1]]
       resultado[i,3]<- minCases_vector[indice[i,2]]
       }

resultado<-data.frame(resultado)
colnames(resultado)<-c("error","CF ","minCases ")


####### GRAFICANDO LOS RESULTADOS
scatter3D(resultado$CF,resultado$minCases,resultado$error, phi = 0, bty = "g",
        pch = 20, cex = 3, ticktype = "detailed",xlab = "CF", ylab = "minCases",
zlab = "error",
         xlim=c(0,1))

####### LOS VALORES ÓPTIMOS
subset(resultado,error==min(resultado$error))


####### ARBOL DE DECISIÓN CON TODOS LOS DATOS

set.seed(27)
modelo<-
C5.0(cluster~NOTA_COLEGIO+NOTA_ADMISION+CTA_COLEGIO+AÑOS_COLEGIO_ADMISION+APORTE_
SEMESTRAL+EDAD_ADMISION+COM_COLEGIO+TIPO_COLEGIO+MAT_COLEGIO+FIS_ADMISION+SEXO+TE
RCIO_SUPERIOR_ESP+MODALIDAD+RM_ADMISION+QUI_ADMISION+ELECCION_ESP_INGRESO+MAT_ADM
ISION+BIO_ADMISION+ESPECIALIDAD,
                    data = datitos, rules = TRUE,
                    control =C5.0Control(subset = FALSE, CF =0.25,
                       minCases = 5,earlyStopping = F))


summary(modelo)
```