Universidad Católica de Santa María

Facultad de Ciencia e Ingenierías Físicas y Formales

Escuela Profesional de Ingeniería de Sistemas

ANÁLISIS DE LA INTERPRETABILIDAD DE LA SALUD ÓSEA CON ÁRBOLES DE DECISIÓN DIFUSA

Tesis presentada por el Bachiller:

Zegarra Vilca, Steven Herman

Para optar el Título Profesional de:

Ingeniero de Sistemas

Especialidad en Ingeniería de Software

Asesor:

Dr. Sulla Torres, José Alfredo

Arequipa – Perú

2019

FACULTAD DE CIENCIAS E INGENIERIAS FISICAS Y FORMALES

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS

INFORME DICTAMEN DE BORRADOR TESIS

VISTO

El Borrador de TESIS titulado:

ANALISIS DE LA INTERPRETABILIDAD DE LA SALUD O'SEA CON ARBOLES DE DECISIÓN DIFUSA

Presentado por (el) (la) (los) Bachiller (es):

STEVEN HERMAN ZEGARRA VILCA

Nuestro dictamen es:

Favorable

OBSERVACIONES: Diseñar esquemas gráficos faltantes

Arequipa, 19 de Junio de 2019

1631

1635

PRESENTACIÓN

Sr. Director de la Escuela Profesional de Ingeniería de Sistemas.

Sres. Miembros del Jurado.

De conformidad con las disposiciones del Reglamento de Grados y Títulos de la Escuela Profesional de Ingeniería de Sistemas, pongo a vuestra consideración el siguiente trabajo de investigación titulado: **"ANÁLISIS DE LA INTERPRETABILIDAD DE LA SALUD ÓSEA CON ÁRBOLES DE DECISIÓN DIFUSA",** el cual de ser aprobado me permitirá optar por el Título Profesional de Ingeniero de Sistemas.

ZEGARRA VILCA STEVEN HERMAN.

DEDICATORIA

El presente trabajo va dedicado a mi familia y amigos, los cuales son personas importantes, sin las que no hubiera podido llegar a este punto de mi vida, y los que me brindaron su apoyo sin interés alguno en todo momento. En primer lugar, a mi madre, la persona más importante en mi vida y la que me ha enseñado todo lo primordial que una persona debe poseer. Mi padre, el cual siempre me guía en todo momento. A mis hermanos, los cuales me ayudan a seguir adelante. A mis amigos, los cuales me ayudan a ser una mejor persona. También a las personas con las que me cruce en la vida con poca interacción, aportaron un gramo en mi vida para poder crecer y aprender de la vida.

AGRADECIMIENTOS

Agradezco al Ingeniero José Sulla Torres, al Dr. Marco Cossío y al grupo de personas que participaron en este proyecto de investigación sobre la salud ósea, mis más sinceros agradecimientos, por toda la ayuda brindada, confianza y paciencia durante todo el proceso de este trabajo.

Mi profundo reconocimiento y gratitud hacia ustedes.

INTRODUCCIÓN

Actualmente, el uso de sistemas informáticos para el apoyo en la medicina está en un gran auge, como se manejan gran cantidad de datos, esta puede digitalizarse y obtener la información que se desea y poder utilizarla de una forma eficaz y eficiente.

A través de los años existen herramientas más poderosas de recolección de datos, herramientas que son utilizadas por empresas, que no son explotadas en su máximo potencial. La minería de datos es una poderosa tecnología con el gran potencial de ayudar a empresas, compañías u organizaciones a centrarse en la parte más importante de los datos recolectados.

Algoritmos de minería de datos, ya sean de clasificación como el algoritmo J48, de asociación como el algoritmo A priori o de clusterización como el algoritmo K-Means, nos permiten analizar correctamente bajo diferentes enfoques, aquí surgen preguntas sobre la minería de datos, ¿El uso de técnicas de minerías de datos para el análisis de repositorios brinda un conocimientos significativo que otro tipo de análisis brindaría?, ¿Emplear diferentes algoritmos de minería de datos que pertenecen a una misma categoría, afectan de manera significativa a los resultados obtenidos?, ¿Efectivamente la lógica difusa ayudara en obtener mejores resultados?, ¿Por qué es importante el análisis de la interpretabilidad ?

Además, en nuestro país no existe un sistema que se adapta a nuestra característica y/o a nuestros datos, en la mayoría de estos sistemas están dirigidos para los países como EE. UU y las zonas de Europa, por lo cual, al ingresar datos de la región, no se obtendrá una información que ayude a tomar las decisiones correctas.

En el presente trabajo de investigación, en la primera etapa de experimentación, se va a utilizar herramientas de minería de datos, una de la más utilizada y poderosa herramienta de minería de datos, porque, es un software libre bajo la licencia pública general de GNU, esta implementado en java, es multiplataforma, es decir, es capaz de correr en casi cualquier plataforma, contiene una de las más extensas colecciones de técnicas para el procesamiento y modelado de los datos, para finalizar, es fácil de utilizar por su interfaz gráfica amigable con el usuario. En esta herramienta utilizaremos los datos recolectados de los colegios de niños y adolescentes de los colegios Inmaculada Concepción y Jorge Basadre de la provincia de Arequipa.

Ahora, en la segunda etapa de experimentación, después de haber comprobado los algoritmos de minería de datos, utilizaremos el más eficaz, ha este algoritmo de clasificación

le añadiremos lógica difusa, dado que, la lógica difusa se adapta mejor al mundo real en el que vivimos. Este sistema de lógica difusa nos dará reglas de resultado. Para finalizar, estas reglas, la idea principal es analizar e interpretarlas y obtener una mejor información, de una manera eficaz, para obtener una mejor toma de decisiones.

RESUMEN

Hoy en día, la salud ósea es un campo de la medicina el cual ha tomado mucho interés, porque cada vez son más comunes las enfermedades relacionadas a los huesos.

El objetivo principal de este proyecto es el análisis de la interpretabilidad de la salud ósea, con la ayuda de un programa de escritorio, el cual tiene por modelo un árbol de decisión difusa y este nos proporciona las reglas. Se han recolectado datos antropométricos y usado fórmulas de regresión para calcular la densidad ósea de escolares entre 12 y 18 años de los colegios Inmaculada Concepción y Jorge Basadre del sector de Arequipa Perú, clasificados según edad y sexo. Se tomo los datos de escolares, porque, los huesos se desarrollan en edades tempranas, y diferenciamos entre géneros, porque existe una notoria diferencia entre hombre y mujer al producir densidad ósea. Al analizar en la herramienta Weka los 5 algoritmos, se ha determinado que el mejor algoritmo entre estos es el árbol de injerto (graft tree) con un porcentaje de clasificación del 92.42% según le conjunto de datos obtenidos. Para el análisis de la interpretabilidad se tomó en cuenta el número de reglas, las variables lingüísticas y la simplicidad de las reglas.

Palabras Clave

Minería de Datos, Lógica Difusa, Interpretabilidad, Salud Ósea

ABSTRACT

Nowadays, bone health is a field of medicine which has taken a lot of interest, because diseases related to bones are becoming more common.

The main objective of this project is the analysis of the interpretability of bone health, with the help of a desktop program, which is modeled on a diffuse decision tree and this provides us with the rules. Anthropometric data were collected and regression formulas were used to calculate the bone density of schoolchildren between 12 and 20 years old from the schools of Inmaculada Concepción and Jorge Basadre in the Arequipa sector, Peru, classified according to age and sex. We took data from schoolchildren, because, bones develop at an early age, and we differentiate between genders, because there is a noticeable difference between men and women when producing bone density. When analyzing the 5 algorithms in the Weka tool, it has been determined that the best algorithm among these is the graft tree with a classification percentage of 92.42% according to the set of data obtained.

Keywords

Data Mining, Fuzzy Logic, Interpretability, Bone Health

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO I

1. PLANTEAMIENTO TEÓRICO

1.1. Planteamiento de la Investigación

1.1.1. Planteamiento del Problema

La lógica difusa es reconocida a nivel mundial, por su capacidad para modelar conceptos lingüísticos, como también su utilización para extracción y representación de conocimiento en el modelado de sistemas (Alonso J. M., 2009). La computación con palabras (CWW) se basa en la representación lingüística del conocimiento que se procesa operando en el nivel semántico, el problema impórtate es cuando los modelos basados en reglas difusas se adquieren datos mediante alguna forma de aprendizaje empírico, a menudo se solicita que estos modelos muestren la interpretabilidad que normalmente se evalúa en términos de características estructurales, tales como la complejidad de reglas, propiedades de conjuntos difusos, particiones, etc. (Mencar, Interpretability assessment of fuzzy knowledge bases: A cointension based approach, 2011).

La interpretabilidad representa la fuerza más importante detrás de la implementación de sistemas basados en lógica difusa (Cannone, A study on interpretability conditions for fuzzy rule-based classifiers, 2009), por su capacidad para modelar conceptos lingüísticos, así como por su utilización para extracción y representación de conocimiento en el modelado de sistemas.

Con este proyecto se buscar realizar una evaluación de 5 algoritmos de minería de datos y utilizar el más eficaz, para poder analizar la interpretabilidad del algoritmo con lógica difusa.

1.1.2. Objetivos de la Investigación

El objetivo principal de la investigación es analizar la interpretabilidad del resultado del algoritmo de minería de datos árbol de decisión con lógica difusa, Los objetivos específicos son:

a) Preparación de datos en los colegios Inmaculada Concepción y Jorge Basadre del sector de Arequipa de la provincia de Arequipa.
b) Comparación de algoritmos de clasificación o supervisados utilizando herramientas de minería de datos.
c) Probar un sistema con lógica difusa, en el cual se utilizará el algoritmo de minería de datos más eficaz.
d) Evaluar los resultados, con las reglas difusas y se podrán analizar para su interpretabilidad en un índice aproximado para su evaluación.

1.1.3. Preguntas de investigación

a) ¿Identificar y evaluar algoritmos de minería de datos ayudaría a una mejor toma de decisiones?
b) ¿Efectivamente la lógica difusa ayudará en obtener mejores resultados?
c) ¿Por qué es importante el análisis de la interpretabilidad?
d) ¿La interpretabilidad y la precisión, estos valores como deben aproximar?

1.1.4. Línea y Sub-Línea de Investigación

a) **Línea:** Inteligencia Artificial
b) **Sub-Línea:** Lógica difusa

1.1.5. Palabras Clave

a) Minería de datos
b) Lógica Difusa
c) Interpretabilidad
d) Salud Ósea

1.1.6. Solución Propuesta

a) Justificación e Importancia

Los resultados obtenidos a partir de esta investigación permitirán a estudiantes comprender mejor el uso técnicas y herramientas de minería de datos, aplicando datos reales, y entender la importancia de este campo para el manejo de información de cualquier institución o empresa.

La información obtenida de la investigación también contribuirá al desarrollo de aplicaciones, software o técnicas que ayuden a instituciones y empresas desarrollarse y tomar decisiones que contribuyan en su crecimiento empresarial. También el uso de la lógica difusa para adaptarse más al mundo real y la interpretabilidad, su análisis que ayudara en las reglas obtenidas.

b) Descripción de la solución

Durante la investigación se realizará una evaluación comparativa de algoritmos de clasificación o supervisados, determinando de esta manera, teórica y experimental, cuales trabajarían de manera más eficaz al ser aplicados con los datos recolectados de niños y jóvenes de la provincia de Arequipa. Después de obtener el algoritmo más eficaz, se utilizará lógica difusa para obtener las reglas, así poder analizar la interpretabilidad. Así mismo, durante dicha investigación se estudiará aspectos relevantes a tener en consideración al momento de aplicar los algoritmos.

1.2. FUNDAMENTOS TEORICOS

1.2.1. Estado del Arte

El uso de conjuntos de datos médicos ha atraído la atención de investigadores de todo el mundo. Las técnicas de minería de datos se han estado utilizado ampliamente en el desarrollo de sistemas de apoyo a la toma de decisiones para la predicción de enfermedades a través de un conjunto de datos médicos, proponen un nuevo sistema basado en el conocimiento para poder predecir las enfermedades mediante la agrupación, eliminación de ruido y técnicas de predicción. Los sistemas basados en el conocimiento pueden ayudar a los medico en la práctica sanitaria como un método analítico clínico (Nilashi, 2017).

La computación con palabras (Computing With Words) está basada en la representación lingüística del conocimiento que se procesa operando en el nivel semántico definido a través de conjuntos difusos. La representación lingüística del conocimiento es un problema importante cuando los modelos basados en reglas difusas se adquieren de los datos mediante alguna forma de aprendizaje empírico. A menudo se solicita a estos modelos que muestren la interpretabilidad, que normalmente se evalúa en términos de características estructurales, tales como complejidad de reglas, propiedades en conjuntos difusos, particiones, etc. En este documento proponen otra forma de evaluar la interpretabilidad, basado en la cointensión, para eso se mide la semántica explicita, que está definida por los parámetros formales del modelo y la semántica implícita trasmitida por el lector por la representación lingüística del conocimiento, esta semántica exige la representación del conocimiento del usuario. El análisis destaco que, al definir la estrategia para evaluar la interpretabilidad de los clasificadores basados en reglas difusas, estos no coinciden con el conocimiento del usuario, por lo que, su representación lingüística no es apropiada, aunque pueden etiquetarse como interpretables desde un punto de vista estructural (Mencar, Interpretability assessment of fuzzy knowledge bases: A cointension based approach, 2011).

La evolución histórica de los procesos y modelado de sistemas basados en reglas difusas, dichos sistemas se caracterizan por dos propiedades fundamentales: Precisión e Interpretabilidad. Inicialmente se dio prioridad a la interpretabilidad, esta quedó en segundo plano y la atención se centró en la precisión. Hoy en día el principal desafío consiste en encontrar un equilibrio óptimo entre ambos. La

precisión no entraña gran dificultad, en cuando a la interpretabilidad, depende mucho de su evaluación, es muy subjetiva, depende de la persona, de sus conocimientos y su experiencia previa y los criterios de evaluación heurística varían bastante de una persona a otra (Alonso J. M., 2008).

En los últimos años, ha crecido el interés de los investigadores en obtener modelos fuzzy lingüísticos más interpretables. Mientras que las medidas de precisión son claras y conocidas, las medidas de interpretabilidad son difíciles de definir ya que la interpretabilidad tiende a depender de varios factores; principalmente la estructura del modelo, el número de reglas, el número de características, el número de términos lingüísticos, la forma de los conjuntos difusos, etc. Además, debido a la subjetividad del concepto, la elección de medidas de interpretabilidad apropiadas sigue siendo un problema abierto (Gacto, 2011).

La interpretabilidad es una de las propiedades más significativas de los sistemas difusos, que son conocidos ampliamente como cajas grises frente a otras técnicas de soft computing, como las redes neuronales, generalmente consideradas como cajas negras. Es esencial para aplicaciones con alta interacción humana (sistemas de apoyo a la toma de decisiones en medicina, economía, etc.). El objetivo es evaluar los índices más utilizados, se llevó realizo un análisis experimental, en la cual se obtuvieron algunas pistas claves en relación con la interpretabilidad, los resultados obtenidos de la encuesta muestran subjetividad inherente de la medida porque recopilaron una gran diversidad de respuestas (Alonso J. M., 2009).

La interpretabilidad representa la fuerza impulsora más importante detrás de la implementación de sistemas basados en lógica difusa. Está relacionada directamente con la base de conocimiento del sistema, con referencia a la facilidad del usuario humano para leer y entender las piezas de información integradas. En este documento nos habla sobre un enfoque innovador que consiste en analizar los componentes de un clasificador difuso en que cual la inferencia se lleva a cabo respetando las propiedades lógicas, como resultado derivaron algunas condiciones y requisitos básicos para ser analizados con el fin de ser interpretables en el sentido semántico (Cannone, A study on interpretability conditions for fuzzy rule-based classifiers, 2009).

Para comprender mejor la interpretabilidad, se llevara a cabo un procedimiento de selección de características basada en arboles de decisión, generaran fuertes

particiones difusas para todas las entradas seleccionadas, luego se define un conjunto de reglas lingüísticas que combinan las variables lingüísticas generadas previamente, después se aplica un procedimiento de simplificación lingüístico guiado por un nuevo índice de interpretabilidad para obtener un conjunto de reglas más compacto y generar sin perder precisión (Alonso J. M., 2009).

La capacidad de interpretación de la información manejada automáticamente ha sido un problema de ficción, los intentos de extraer conocimiento de cantidades grandes de datos o de interpretar el funcionamiento de sistemas complejos, llevo a cabo investigaciones como las reglas de asociación en base de datos o sistemas expertos. La explosión de datos disponibles en el mundo digital, presionan a los investigadores que obtenga métodos eficientes para extraer y resumir información fácilmente compresible. La interpretabilidad de los resúmenes lingüísticos difusos, tanto a nivel de la oración o como resumen. La interpretabilidad de la oración individual se examina como dependiente tanto de su representatividad medida por un grado de calidad como de su expresión lingüística. También se discuten diferentes propiedades en el nivel de resumen, es decir, su consistencia, su no redundancia y la información que transmiten (Lesot, 2016).

El modelado borroso es una de las técnicas más conocidas para modelar sistemas y procesos. En la mayoría de los casos, estos modelos difusos alcanzan una gran precisión, pero muestran un rendimiento deficiente en complejidad de interpretabilidad, que son aspectos clave de la lógica difusa. Existen varios enfoques en la literatura para tratar los desafíos de complejidad e interpretación para los sistemas basados en reglas difusas, se propone un enfoque de posprocesamiento mediante una selección de reglas genéticas basadas en la relevancia de cada regla (usan Transformaciones Ortogonales OT) y también la conocida regla entre exactitud e interpretabilidad, lo principal es verifica la importancia, los inconvenientes y las ventajas de la selección de reglas basadas en OT (Isabel Rey, 2013).

Los trastornos óseos pueden ocurrir de dos maneras por accidente y carecen de vitaminas. Tanto la Osteoporosis como la Osteopenia son las fracturas óseas minúsculas que se pueden detectar a través de varios procesos y métodos. Especialmente estos dos trastornos óseos se deben a la deficiencia de vitamina (D3). Aquí la detección de los trastornos óseos se realiza con la ayuda del

densitómetro óseo. El densitómetro óseo utiliza una técnica para medir la densidad ósea en términos de T-score. Los valores derivados se trazaron con la ayuda de valores de desviación estándar (Ramkumar, 2018).

La agrupación de datos es un proceso en el cual consiste en poner datos similares en grupos, un algoritmo de agrupamiento lo divide en varios grupos de forma que la similitud de un grupo es mayor a otros grupos. Este articulo nos muestra seis formas de agrupamiento DBScan, Agrupamiento basado en densidad, óptica, EM, entre otros. También estas técnicas de agrupamiento son analizadas utilizando la herramienta WEKA, en la cual se usaron datos bancarios relacionados a la información del cliente (Manish Verma, 2012).

La clasificación es una técnica de minería de datos utilizada para predecir la pertenencia de un grupo de datos. Presentan la comparación de diferentes técnicas de clasificación de minería de datos de código abierto, que consiste en un árbol de decisión y aprendizaje automático. Los métodos probados son: J48-injerto y árbol LAD, red de función radial y la máquina de vectores de soporte. Los resultados obtenidos indican que, mediante el uso de vectores de soporte, es más eficiente que los demás que se compararon (Wisaeng, 2013).

Como afirma Bhandari et al (2015), el algoritmo a priori, el cual es uno de los más importantes que se utiliza para extraer conjunto de elementos frecuentes de grandes bases de datos y obtener la regla de asociación para obtener conocimiento. Lo que necesita este algoritmo, son 2 cosas importantes: Apoyo mínimo y confianza mínima, lo primero es verificar los artículos, si son mayores o iguales al soporte mínimo después se encuentran los conjuntos de elementos frecuentes. Luego la restricción de confianza mínima la cual se usa para formar reglas de asociación.

La revolución digital, hoy en día ha hecho posible capturar información digital muy fácil, también, es frecuente escuchar el uso de minería de datos en las empresas, la minería de datos incrementa la satisfacción del cliente o la competencia para ganar cuota de mercado. El incremento del uso de minería de datos indica que esta será adoptada por la sociedad con el mismo peso que tiene la estadística, algunas aplicaciones de la minería de datos son: comercio y banca, medicina y farmacia, seguridad y detección de fraudes, recuperación de información no numérica, astronomía, geología, minería, agricultura y pesca,

ciencias ambientales, ciencias sociales, entre otros. Esto, nos indica que la minería de datos está en un crecimiento y aceptado por la sociedad (Riquelme, 2006).

Como indica Vanesa Berlanga (2013), un árbol de decisión es una de una forma gráfica y analítica de representar todos los sucesos que pueden surgir al momento de tomar una decisión en cierto momento, ayuda a tomar una decisión acertada, visto desde un punto probabilístico. Los arboles de decisión son una técnica estadística para la segmentación, la estratificación, la predicción, la reducción de datos y el filtrado de variables, la identificación de interacciones, la fusión de categorías y la discretización de variables continuas. La función en SPSS es crear arboles de clasificación y de decisión, así identificar grupos, descubrir las relaciones entre estos y poder predecir eventos a futuro.

El descubrimiento de conocimiento y la minería de datos, es un área multidisciplinaria la cual usa metodología para extraer conocimiento útil de los datos, existen varias herramientas para extraer el conocimiento. Una de las aplicaciones de este conocimiento es para aumentar la calidad de la educación, la minería de datos se puede usar para la toma de decisiones en el sistema educativo, este documento propone usar arboles de decisión en la minería de datos educativos, estos algoritmos se aplicaran en los datos pasados de los estudiantes y así este puede predecir el rendimiento de los estudiantes, ayudara a identificar a los que abandonan la escuela y los estudiantes que necesitan atención especial, también para que el maestro brinde consejería y/o asesoramiento apropiado (Yadav & Brijesh, 2012).

Como afirma Wisaeng (2013), la técnica de clasificación en minería de datos que se utiliza para predecir la pertenencia a un grupo para una instancia de datos, en su documento presenta la comparación de diferentes técnicas de clasificación de minería de datos de código abierto las cuales consisten en árbol de decisiones y aprendizaje automático. Los métodos de árbol de decisión probados son J48-injerto y árbol LAD, mientras que los de aprendizaje automático probado son la red de función radial y la máquina de vectores de soporte. Los resultados de este experimento señalan una clasificación de sensibilidad, especificidad, precisión, error absoluto medio y error cuadrático medio.

Como afirma Martínez (2011), la minería de datos es una tecnología de gran importancia que permite la integración de un conjunto de áreas (estadística,

inteligencia artificial, matemáticas, biología y medicina) y también ayuda a identificar información oculta significativa que se encuentran en los grandes volúmenes de datos. En esta investigación se muestra empíricamente que es posible diagnosticar la administración de fármacos en pacientes con síntomas de enfermedad cardiovascular, usando la variable de presión arterial, índice de colesterol, azúcar en la sangre, alergias a antibióticos y otras alergias.

Jaqueline (2015) nos indica, como las universidades han tenido gran interés sobre el rendimiento académico de los estudiantes y en determinar qué factores influyen, para esto se analizó los datos académicos, personales, socioeconómicos e institucionales correspondientes. Posteriormente se realizó estudios de metodologías de minería de datos y aplicando las técnicas de clasificación, que son algoritmos como RIDOR, ID3, C4.5, JRIP y PART, las cuales se utilizaron en RAPIDMINER, lo cual permitió llevar los procesos y obtener resultados del modelo, los cuales se evaluaron a través de los datos reales para determinar los factores que influyen en el rendimiento académico.

Carlos Denegri (2006), indica que para el proceso mental que acerca al diagnóstico de un paciente es complejo, dado que, en la medicina se hallan con este planteo: el paciente presenta un conjunto de signos y de síntomas ¿Qué enfermedad tiene?, esta pregunta tiene un grado de incertidumbre implícito. De hecho, la secuencia de eventos y los datos disponibles pueden no ser conocidos con exactitud, el paciente puede administrar datos equivocados, no está segura de tal o cual afirmación o evocación, puede haber ausencia de información, errores y subjetividad. Para medir la incertidumbre se puede partir de un conjunto grande que incluya las posibilidades diagnosticas, construir un subconjunto y asignar un número real que mida el grado de incertidumbre, para esto es necesario recurrir a medidas de probabilidad, ahí es donde se utiliza la lógica difusa, la cual es una técnica de la inteligencia computacional que permite trabajar con información con alto grado de imprecisión.

Como nos afirma José Garcia (2010), con el uso de minería de datos, se puede encontrar información que hasta el momento había sido desconocida, también indica que esta información ayuda en la toma de decisiones o el desarrollo de algún proceso a nivel empresarial.

Como nos indica Hayashi (2015), el objetivo de este estudio fue extraer reglas de clasificación altamente precisas, concisas e interpretables para el diagnóstico utilizando el algoritmo Re-RX con el árbol de decisión J48graft, para el diagnóstico de cáncer de mama, comparando con otros algoritmos de extracción de reglas, dio como resultado número promedio menor de reglas para diagnosticar, lo cual es una ventaja sustancial, también un promedio más bajo de antecedentes por regla.

La salud ósea en niños y adolescentes es muy importante, ya que en esta edad es cuando se puede prevenir las enfermedades a futuro como osteopenia y osteoporosis, con una serie de ecuaciones de regresión se puede predecir algunas enfermedades de la salud ósea y teniendo en cuenta algunos indicadores antropométricos para proponer valores de referencia basados en edad y sexo. En este proyecto generó cuatro modelos de regresión para calcular la salud ósea para los hombres DMO = (R^2 = 0.79) y BMC = (R^2=0. 84), para las mujeres DMO = (R^2 = 0.76) y BMC = (R^2 = 0.83), se desarrollaron los percentiles usando el método LMS (p3, p5, p15, p25, p50, p75, p85, p90 y p97), (Gómez-Campos, Arruda, & Cossio-Bolaños, 2017).

1.3. MARCO METODOLÓGICO

1.3.1. Alcances y Limitaciones

Esta investigación toma los siguientes puntos de alcance y limitaciones:

- Se tomarán datos antropométricos de escolares entre 12 y 18 años de los colegios de Inmaculada Concepción y Jorge Basadre del sector de Arequipa, Perú.
- Se realizará esta tesis en un ambiente normal, en el cual se tendrá una estación de trabajo con las funciones básicas y que pueda soportar las herramientas de Weka, Matlab y NetBeans.
- El tiempo tomado para realizar esta investigación es de 1 año aproximadamente, en el cual se realizará todo el proceso de la extracción del conocimiento.
- Esta tesis pertenece a un proyecto de la Universidad Católica de Santa María, se verá financiado por esta.

1.3.2. Aporte

Una vez culminado el proyecto de investigación se obtendrá un análisis corroborado de los algoritmos seleccionados para el procesamiento y obtención de información relevante en repositorios de datos, también teniendo en cuenta el análisis de la interpretabilidad que se podrá obtener, además al final se contara con diversas conclusiones que incluirán desde aquellas que tenga correlación directa con el resultado final esperado hasta los hallazgos realizados durante el proceso.

1.3.3. Tipo y Nivel de Investigación

a) **Tipo:** Aplicada

La presente investigación se realizará para determinar la eficiencia de algoritmos de minería de datos de clasificación o supervisados, luego de obtener el más eficaz para poder aplicarlo con lógica difusa y así analizar la interpretabilidad de los resultados obtenidos.

b) **Nivel:** Comparativa / Experimental o Evaluatoria

Con esta investigación se pretende realizar una evaluación teórica y experimental de algoritmos de minería de datos con el fin de obtener el más eficaz con los datos recolectados de niños y adolescentes, para poder así utilizar con lógica difusa y analizar la interpretabilidad.

1.3.4. Población y Muestra o Universo

a) Niños y adolescentes de los colegios Inmaculada Concepción y Jorge Basadre de la Región de Arequipa

Conjunto de datos recolectados de los colegios de la provincia de Arequipa de los colegios nacionales los cuales son Inmaculada Concepción y Jorge Basadre, se obtuvo 660 registros de niños y adolescentes entre 12 y 18 años de los cuales se les tomo en cuenta su peso, edad, genero, estatura, entre otros datos.

1.3.5. Métodos, Técnicas e Instrumentos de Recolección de Datos

La tabla mostrará los resultados del análisis realizado, mostrando a través de ella una comparación entre los algoritmos estudiados. Los datos que se emplearán en la etapa de experimentación se obtendrán a partir de la recolección de adolescentes entre 12 y 18 años de los colegios de la provincia de Arequipa, los cuales son Inmaculada Concepción y Jorge Basadre.

CAPITULO II

2. Marco Teórico

En este capítulo se verá el marco teórico, el marco teórico que fundamenta esta investigación proporcionará al lector una idea más clara acerca del tema "Análisis de la interpretabilidad de la salud ósea con árbol de decisión difusa", así como una breve introducción de "algoritmos de minería de datos", "lógica difusa". Se encontrarán conceptos básicos, complementarios y específicos.

2.1. DEFINICIONES DE MINERÍA DE DATOS

2.1.1. Minería de Datos

Para comprender que es lo que hace la minería de datos, tenemos que tener en cuenta lo que nos dice Riquelme (2006) La revolución digital ha hecho posible que la información digitalizada sea fácil de capturar, procesar, almacenar, distribuir, y transmitir. Algunas características halladas fueron las siguientes:

i. La minería de datos nos ayuda a descubrir conocimiento de estos enormes volúmenes de datos y transformarla en información.

ii. Buscarle sentido a la explosión de información que actualmente puede ser almacenada.

iii. La habilidad para extraer información útil la toma de decisiones o la exploración, y la comprensión del fenómeno gobernante en la fuente de datos.

2.1.2. Patrones Frecuentes

Los patrones frecuentes, como su nombre lo sugieren, son patrones que ocurren con frecuencia en los datos. Hay muchos tipos de elementos frecuentes, subsecuencias frecuentes (también conocidas como patrones secuenciales) y subestructuras frecuentes. Un elemento frecuente suele referirse a un conjunto de elementos que a menudo aparecen juntos en un conjunto de datos transaccionales, por ejemplo, leche y pan, que muchos clientes compran juntos en las tiendas de comestibles. Una subsecuencia que se produce con frecuencia, como el patrón que los clientes tienden a comprar en una computadora portátil, seguida de una cámara digital y luego una tarjeta de memoria, es un patrón secuencial (frecuente). Una subestructura puede referirse a diferentes formas estructurales, gráficos, árboles,

etc. Se puede combinar con conjuntos de elementos o subsecuencias. Si una subestructura ocurre con frecuencia, se denomina patrón estructurado frecuente. La minería de patrones frecuentes conduce al descubrimiento de asociaciones interesantes y la correlación dentro de los datos (Jiawei Han, 2006).

2.1.3. Clasificación y Predicción

Es el proceso de encontrar un modelo (o función) que describe y distingue clases de datos o conceptos de estos. El modelo se deriva en base al análisis de un conjunto de datos de entrenamiento, el modelo se usa para predecir la etiqueta de clase de objetos para los cuales la etiqueta de la clase es desconocida, el modelo puede representarse en diversas formas, como reglas de clasificación, árboles de decisión, fórmulas matemáticas o redes neuronales. Los modelos se utilizan para predecir la ausencia o se refiere a la predicción numérica categórica (Jiawei Han, 2006).

2.1.4. Análisis de Clúster

A diferencia de la clasificación y la regresión, que analizan conjuntos de datos etiquetados (entrenamientos), la agrupación analiza objetos de datos sin consultar las etiquetas de clase. En muchos casos, los datos etiquetados como clase simplemente no pueden existir al comienzo. La agrupación en clúster puede usarse para generar etiquetas de clase para un grupo de datos. Los objetos se agrupan o se basan en el principio de maximizar la similitud intraclase y minimizar la similitud interclase. Es decir, se forman grupos de objetos para que los objetos dentro de un grupo tengan una gran similitud en comparación unos con otros, pero son más bien similares a los objetos en otros grupos. Cada clúster así formado se puede ver como una clase de objetos, de la cual se pueden derivar reglas. La agrupación también puede facilitar la formación de taxonomía, es decir, la organización de observaciones en una jerarquía de clases que agrupan sucesos similares en conjunto (Jiawei Han, 2006).

2.1.5. Eficiencia y Escalabilidad

La eficiencia y la escalabilidad siempre se consideran al comparar algoritmos de minería de datos. A medida que las cantidades de datos continúan multiplicándose, estos dos factores son especialmente críticos. Los algoritmos de minería de datos deben ser eficientes y escalables para extraer de manera efectiva

la información de enormes cantidades de datos en muchos repositorios de datos o en flujos de datos dinámicos. En otras palabras, el tiempo de ejecución de un algoritmo de minería de datos debe ser predecible, breve y aceptable para las aplicaciones. La eficiencia, la escalabilidad, el rendimiento, la optimización y la capacidad de ejecución en tiempo real son criterios clave que impulsan el desarrollo de muchos algoritmos nuevos de minería de datos (Jiawei Han, 2006).

2.1.6. Reglas de Asociación

Es la exploración de los datos con el propósito de identificar relaciones entre los datos, dentro de una fuente o base de datos. Son utilizadas cuando el objetivo es realizar análisis exploratorios, buscando relaciones dentro del conjunto de datos. Las asociaciones identificadas pueden usarse para predecir comportamientos, y permiten descubrir correlaciones y coocurrencias de eventos (Encarnación, 2015).

2.1.7. Discretización

El atributo es transformado de valores numéricos en valor valores categóricos, de esta forma se reduce el número de posibles valores. La Discretización suaviza el efecto del ruido y permite modelos más simples (Miguel Cárdenas Montes, 2016).

2.1.8. Derivación

La derivación permite crear nuevos atributos partiendo de otros anteriores, esto se realiza a través de alguna operación matemática: por ejemplo, agrupamiento de valores de tiempo en unidades de orden superior (segundos en minutos), agrupamiento de valores (meses en trimestres), reemplazar valores por medias (suavización), etc. (Miguel Cárdenas Montes, 2016).

2.1.9. Normalización

El atributo es escalado a un rango específico, normalmente de -1 a 1 o de 0 a 1, donde 1 representa al caso más general. La normalización es empleada cuando se tienen atributos con órdenes de magnitud muy diferentes. Gracias a la normalización se evita que atributos con valores más altos ganen un peso significativamente más importante en el modelo final que aquellos valores más bajos (Miguel Cárdenas Montes, 2016).

2.1.10. Transformación de los datos y reducción

Los datos pueden ser transformados por normalización, particularmente cuando se utilizan métodos que implican mediciones de distancia en la etapa de aprendizaje (Jiawei Han y Micheline Kamber, 2006). En este paso se construyen nuevos atributos a partir de los atributos originales, facilitando una mejor interpretación de la información (Miguel Cárdenas Montes, 2016).

2.1.11. Descubrimiento de Clústeres de forma Arbitraria

Muchos algoritmos determinan los clústeres basados en las medidas de distancia Euclideana o de Manhattan. Algoritmos basados en este tipo de medidas de distancia tienden a encontrar clústeres esféricos con un tamaño 40 y densidad similar, sin embargo, un grupo podría ser de cualquier forma. Es importante desarrollar algoritmos que puedan detectar grupos de forma arbitraria (Jiawei Han y Micheline Kamber, 2006).

2.1.12. Ruido

La mayoría de las bases de datos del mundo real contienen valores atípicos, faltantes, desconocidos o erróneos. Algunos algoritmos de clustering son sensibles a estos datos y pueden dan lugar a clústeres de mala calidad (Jiawei Han y Micheline Kamber, 2006).

2.1.13. Interpretabilidad y Facilidad de uso

Los usuarios esperan que los resultados del clustering sean interpretables, comprensibles y utilizables; es decir, el clustering puede necesitar estar atado a interpretaciones semánticas específicas y aplicaciones (Jiawei Han y Micheline Kamber, 2006).

2.2. DEFINICION BASE DE DATOS

2.2.1. Sistema de base de datos

Un sistema de base de datos, también llamado sistema de administración de bases de datos (DBMS), consiste en una colección de datos interrelacionados, conocida como base de datos, y un conjunto de programas de software para administrar y acceder a los datos. Los programas de software proporcionan mecanismos para definir estructuras de bases de datos y almacenamiento de datos; para especificar y gestionar el acceso a datos concurrente, compartido o distribuido; y para

garantizar el cosensec y la seguridad de la información almacenada a pesar de que el sistema se bloquee o intente acceder sin autorización.

2.2.2. Base de datos relacional

Una base de datos relacional es una colección de tablas, cada una de las cuáles es asignada con un nombre único. Cada tabla consiste en un conjunto de atributos (columnas o campos), y por lo general almacena un gran conjunto de tuplas (registros o filas). Cada tupla en una tabla relacional representa a un objeto identificado por una clave única y es descrita por un conjunto de valores atributo. Un modelo de datos semántico, tal como un modelo entidad – relación (ER), a menudo se construye para las bases de datos relacionales. Un modelo de datos ER representa a la base de datos como un conjunto de entidades y sus relaciones (Jiawei Han y Micheline Kamber, 2006).

2.2.3. Atributos categóricos y numéricos

2.2.3.1. Atributos Categóricos (Cualitativos)

Representan categorías más que números. Operaciones como la suma o la resta no tienen sentido. Se dividen a su vez en:

Nominales: no tienen orden significativo. Podemos realizar operaciones de igualdad o desigualdad.

Ordinales: tienen orden definido. Se puede realizar igualdades, desigualdades, mayor y menor que.

Hay que tener cuidado ya que existen atributos que pueden parecer numéricos, pero son categóricos, como un código postal o un número de teléfono, hay que notar que no tiene sentido que dos códigos postales sean sumados o sacar el promedio de varios números de teléfono, por eso son categóricos (Wikiversity, 2016).

2.2.3.2. Atributos Numéricos (Cuantitativos)

Son atributos que son número y pueden ser tratados como tal. Se dividen a su vez en:

Intervalo: no existe un 'cero', la división no tiene sentido. Se pueden hacer operaciones de igualdad, desigualdad, de orden, sumas y restas.

Radio: el cero existe, la división tiene sentido. Podemos realizar operaciones que tienen intervalo y además multiplicación y división (Wikiversity, 2016).

2.2.4. Atributos Discretos y Continuos

Un atributo discreto tiene un número finito o contable de valores. En general se representa como números enteros. Atributos binarios son un caso especial de ellos. Los atributos categóricos o cualitativos siempre son discretos (Wikiversity, 2016).

Un atributo continuo tiene un número infinito de valores posibles. Es representado por números reales o de punto flotante. Se pueden obtener tan precisos como sea el instrumento de medición. Los atributos numéricos pueden ser continuos o discretos (Wikiversity, 2016).

2.2.5. Base de Datos Transaccional

En general, cada registro en una base de datos transaccional captura una transacción, como la compra de un cliente, una reserva de vuelo o los clics de un usuario en una página web. Una transacción generalmente incluye un número de identidad de transacción único (ID) y una lista de los elementos que componen la transacción, como los artículos comprados en la transacción. Una base de datos transaccional puede tener tablas adicionales, que contienen otra información sobre el vendedor o la sucursal, y así sucesivamente (Jiawei Han, 2006).

2.3. DEFINICIÓN DE LÓGICA DIFUSA

2.3.1. Lógica Difusa

La lógica difusa es una metodología que proporciona una manera simple y elegante de obtener una conclusión a partir de información de entrada vaga, ambigua, imprecisa, con ruido o incompleta. En general la lógica difusa imita como una persona toma decisiones basada en información con las características mencionadas. Una de las ventajas de la lógica difusa es la posibilidad de implementar sistemas basados en ella tanto en hardware como en software o en combinación de ambos (Carlos Denegri, 2006).

La lógica difusa es una técnica de la inteligencia computacional que permite trabajar con información con alto grado de imprecisión, en esto se diferencia de la lógica convencional que trabaja con información bien definida y precisa. Es una lógica multivaluada que permite valores intermedios para poder definir

evaluaciones entre sí, verdadero/falso, negro/blanco, caliente/frio, pequeño/grande, cerca/lejos, pocos/muchos, etc. (Carlos Denegri, 2006).

El concepto de lógica difusa fue concebido por Lofti A. Zadeh, profesor del a Universidad de California en Berkeley, quien disconforme con los conjuntos clásicos que solo permiten dos opciones, la pertenencia o no de un elemento a dicho conjunto, la presentó como una forma de procesar información permitiendo pertenecías parciales a unos conjuntos, que en contraposición a los clásicos denomino Conjuntos Difusos (fuzzy sets), (Carlos Denegri, 2006).

En la conocida teoría de conjuntos, un elemento pertenece o no a un conjunto. En un conjunto difuso su frontera no está precisamente definida, y el grado de pertenencia entrega un valor entre 0 y 1. El concepto grado de pertenecía reemplaza al blanco y negro, es subjetivo y dependiente del dominio. El concepto de conjunto difuso fue expuesto por Zadeh en un artículo del año 1965, hoy clásico en la literatura de la lógica difusa, titulado “Fuzzy Sets” (Carlos Denegri, 2006).

Como Zadeh nos dice: “La lógica difusa trata de copiar la forma en que los humanos toman decisiones. Lo curioso es que, aunque baraja información imprecisa, esta lógica es en cierto modo muy precisa: se puede aparcar un coche en muy poco espacio sin darle al de atrás. Suena paradoja, pero es así” (Carlos Denegri, 2006).

La expresividad semántica de la lógica difusa, mediante el uso de reglas y variables lingüísticas, es muy próxima al lenguaje natural empleado por los humanos. Esto favorece la interpretabilidad del modelo final construido, lo que se considera la principal virtud de los sistemas basados en reglas difusas frente a otras técnicas de soft computing (Alonso J. M., 2008).

La primera definición formal de Interpretabilidad que se trata de una definición matemática en el contexto de la lógica clásica con la que Tarski pretendía sentar las bases para identificar teorías interpretables (Alonso J. M., 2008).

Bodenhofer y Bauer realizaron una definición axiomática más formal: La Interpretabilidad de un sistema es la posibilidad de estimar su comportamiento a partir de la lectura y comprensión de la descripción de su base de reglas. Puesto que la legibilidad de las reglas depende a su vez de la legibilidad de las expresiones lingüísticas que estas incluyen, Bodenhofer considera que el análisis se debe

realizar a nivel de particiones garantizando que estas respeten las relaciones de orden e inclusión entre conjuntos difusos. En conclusión, la Interpretabilidad de las particiones es un prerrequisito para que un SBRD sea interpretable (Alonso J. M., 2008).

2.3.1.1. Interpretabilidad de variables

Aunque el uso de variables lingüísticas favorece la interpretabilidad, no la garantiza. Por ello es necesario imponer ciertas restricciones en la fase de diseño. Este es un tema que ha sido estudiado en profundidad por muchos autores, y por suerte existe un tipo especial de particiones difusas, las denominadas particiones difusas fuertes que satisfacen todas las restricciones exigidas para que el sistema resultante sea interpretable (Alonso J. M., 2008):

- Definir un numero moderado de etiquetas, términos lingüísticos, por variable.
- Asociar de forma clara un término lingüístico con cada función de pertenencia, de manera que los conjuntos difusos se distingan sin ambigüedad.
- Utilizar funciones de pertenencia que cubran todo el universo de discurso en que se define la variable, y que cumplan la propiedad de normalidad.

Las particiones fuertes garantizan la interpretabilidad a nivel de variable, sin embargo, supone una gran restricción desde el punto de vista de la precisión, cuanto más se degrada la partición mayor es la precisión que se puede alcanzar, pero la interpretabilidad disminuye drásticamente (Alonso J. M., 2008).

2.3.1.2. Interpretabilidad de reglas

Hay que destacar que la legibilidad de la base de reglas es mayor si se define una semántica global previa a la generación de reglas, es decir, si todas las reglas comparten los mismos términos lingüísticos definidos por los mismos conjuntos difusos, lo que permite comparar las reglas directamente a nivel lingüístico. Luego de estudiar la interpretabilidad de las variables queda discutir la interpretabilidad de las reglas (Alonso J. M., 2008). Existen dos tipos de reglas difusas:

1. Mamdani: La conclusión es un conjunto difuso, las premisas de las reglas están formadas por tuplas, además el uso de modificadores lingüísticos (más, menos, entre, ligeramente, etc.) lleva a aumentar la precisión del modelo final, sin embargo, podrían empeorar la legibilidad y dificultar la compresión del sistema. El uso de reglas tipo Mamdani favorece la Interpretabilidad porque son reglas lingüísticas de la forma (Alonso J. M., 2008):

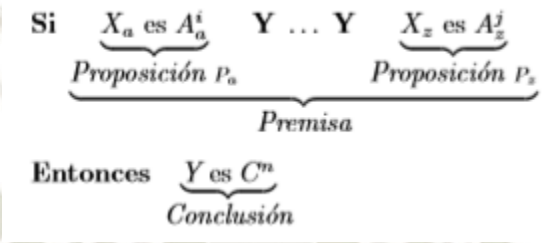


Figura Nº 1: Reglas de Mamdani

Fuente: (Alonso J. M., 2008)

2. Takagi-Sugeno: La conclusión es un valor numérico obtenido como una combinación de los valores de entrada.

2.3.2. Evaluación de la Interpretabilidad

Como nos dice Alonso J. M., los sistemas difusos más interpretables son aquellos que consideran reglas difusas lingüísticas puras (sin pesos ni excepciones), con semánticas globales y particiones difusas fuertes. No obstante, no todos los sistemas alcanzan el mismo grado de Interpretabilidad. Esta se puede evaluar según dos puntos de vista:

1. **Descripción global:** Para comprender un Sistemas Basados en Reglas Difusas (SBRD) en su conjunto, éste debe tener unas dimensiones reducidas. El análisis se debe hacer a dos niveles. A nivel de variable, si se trabaja con particiones fuertes, el único parámetro que se debe considerar es el número de etiquetas. A nivel de reglas, hay que tener en cuenta el número de reglas, pero también su complejidad, es decir, el número de variables utilizadas por regla y la complejidad de las proposiciones lingüísticas que forman las premisas (Alonso J. M., 2008).

2. **Explicación puntual:** Para entender el comportamiento de un SBRD ante una situación concreta hay que analizar los operadores difusos utilizados, así como el número de reglas que pueden activarse simultáneamente dado un vector de entrada. Entender la salida de un SBRD implica comprender los mecanismos de inferencia, agregación y defuzzificación (Alonso J. M., 2008).

Como podemos observar, estos dos puntos son contradictorios, dado que, el primero fomenta la utilización de reglas generales y compactas que sólo utilizan subconjuntos de las variables de entrada. El segundo lleva a preferir reglas completas, muchos más específicas, que utilizan todas las entradas porque cuanto más generales sean las reglas, mayor número de ellas se pueden activar a la vez.

2.4. HERRAMIENTAS DE MINERIA DE DATOS

Se verá acerca de las herramientas que se utilizaran para comparar la eficiencia de los algoritmos de minería de datos con la base de datos obtenidos de los colegios de la provincia de Arequipa. Estas herramientas sirven para extraer conocimientos desde base de datos que contienen grandes cantidades de información.

2.4.1. WEKA

Es una herramienta que permite la experimentación de análisis de datos mediante la aplicación, análisis y evaluación de las técnicas más relevantes de análisis de minería de datos (Encarnación, 2015).

Ventajas:

- Es un software desarrollado bajo licencia GNU – GLP
- Contiene una extensa colección de técnicas para reprocesamiento y modelado de datos.
- Soporta varias tareas de minería de datos, especialmente preprocesamiento, agrupación, clasificación, regresión, visualización y selección.
- Proporciona acceso a Bases de datos usando SQL, gracias a la conexión “Java Data base Connectivity” (JDBC).

2.4.1.1. Entornos de trabajo de WEKA

Define 4 entornos de trabajo:

- **Explorer:** Permite visualizar y aplicar distintos algoritmos de aprendizaje a un conjunto de datos. También posee 6 sub- entornos de ejecución (Encarnación, 2015):
  - Preprocess: Incluye las herramientas y filtros para cargar y manipular los datos.
  - Classify: Acceso a las técnicas de clasificación y regresión.
  - Clúster: Integra varios métodos de agrupación.
  - Associate: Incluye técnicas de reglas de asociación.
  - Select Attributes: Permite aplicar diversas técnicas para la reducción del número de atributos.
  - Visualize: En esta pestaña podemos estudiar el comportamiento de los datos mediante técnicas de visualización.
- **Experimenter:** Entorno centrado en la automatización de tareas de manera que se facilite la realización de experimentos a gran escala.
  Es un entorno grafico que permite al usuario crear, ejecutar, modificar y analizar experimentos sobre tareas de clasificación de un modelo ágil y eficaz (Encarnación, 2015).
- **KnowledgeFlow:** Permite generar proyectos de minería de datos mediante generación de flujos de información (Encarnación, 2015).
- **Simple CLI:** La interfaz "Command-Line Interfaz" es simplemente una ventana de comandos java para ejecutar las clases de WEKA (Encarnación, 2015).

2.4.2. RAPID MINER

Es un entorno de código abierto para aprendizaje automático y minería de datos. Permite realizar todos los procesos que intervienen en un proyecto: la adquisición de datos, la transformación de los datos, la selección de datos, la selección de atributos, la transformación de los atributos, el aprendizaje/modelización y la validación (Encarnación, 2015).

Rapid Miner apoya el proceso flexible de acuerdo con que permite buscar el mejor esquema de aprendizaje de preprocesamiento de los datos y de las tareas de aprendizaje que están a la mano de acuerdo con el problema planteado (José Antonio Garcia Bermúdez, 2010).

Además, permite el desarrollo de procesos de análisis de datos mediante el encadenamiento de operadores a través de un entorno gráfico, que hace posible aumentar la productividad a través de modelos que solucionan los problemas de predicción, clasificación y segmentación de la información (Encarnación, 2015).

Ventajas (Encarnación, 2015):

- Está desarrollado en java.
- Es multiplataforma.
- Representación interna de los procesos de análisis de datos en ficheros XML.
- Contiene más de 500 técnicas de preprocesamiento de datos, modelación predictiva y descriptiva.
- Permite a los experimentos componerse de un gran número de operadores anidables arbitrariamente, que se detallan en archivos XML.
- Puede usarse de diversas maneras:
    - A través de un GUI.
    - En líneas de comando.
    - En batch (lotes)
    - Desde otros programas, a través de llamadas a sus bibliotecas.
- Incluyen gráficos y herramientas de visualización de datos.
- Software de código abierto.

2.4.3. R

R es un entorno de software libre para el cálculo estadístico y gráfico. Se proporciona una amplia variedad de técnicas estadísticas y gráficas. R puede ser extendido fácilmente a través de paquetes. Hay alrededor de 4000 paquetes disponibles en el repositorio de paquetes CRAN (Encarnación, 2015).

Ventajas (Encarnación, 2015):

- Un conjunto integrado de herramientas de análisis de datos.
- Está disponible de manera gratuita para múltiples plataformas.

- Posee funciones estadísticas que son útiles para la minería de datos.
- Funciones gráficas para análisis y visualización de los datos.
- Un buen gestor de datos.
- Un conjunto de operadores para cálculos en arrays (vectores de gran tamaño).

2.4.4. MATLAB

Matlab es un ambiente de computadora científica, que permite la interacción con el usuario a través de una ventana denominada "ventada de comando", donde los comandos deben ser proporcionados por los usuarios mediando el lenguaje FORTRAN, se muestre los cálculos y los resultados.

En Matlab, se puede usar de distintas formas, estos usos son facilitados mediante los Toolbox, cada uno de estos tiene una colección de archivos orientados a tratar un tipo de problema científico. Entre estos se encuentra el Toolbox de "Fuzzy Logical Toolbox" la cual, se explicará adelante.

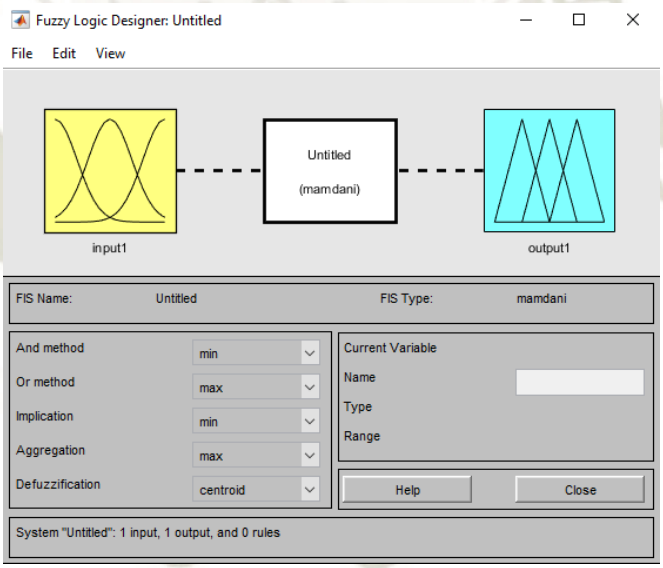


Figura Nº 2: Fuzzy Logic Designer

Fuente: Matlab

Al momento de abrir Fuzzy Logical Toolbox, aparece la imagen de la figura N° 2, como se puede observar, en esta figura, destaca el nombre de "MAMDANI", es el método que hemos utilizado.

El método Mamdani, es un sistema fuzzy que coincide con cada entrada difusa y salida difusa, este método, espera que cada entrada crips (un número real o n

números reales) coincidan con una salida crips, esto quiere decir, es una función de Rn en R, construida de una manera específica, los módulos que se usaron para esta función son:

1. **Módulo de fuzzificación:** Es el que se encargar de modelar matemáticamente la información de las variables de entrada mediante los conjuntos fuzzy, es una parte importante para el proceso que se va a analizar.
2. **Módulo de la base de reglas:** Es el núcleo del sistema, este módulo se encarga de guardar las variables y sus clasificaciones lingüísticas.
3. **Módulo de inferencia:** en este módulo se definen los conectores lógicos que se usaran para estableces la relación fuzzy para poder modelar la base de reglas. De este módulo depende el éxito del sistema fuzzy.
4. **Módulo de defuzzificación:** el módulo que se encarga de traducir el estado de la variable de salida fuzzy a un valor numérico.

2.5. SALUD ÓSEA

La salud ósea se crea durante la infancia y la adolescencia, es decir, para obtener el mejor desarrollo de contenido mineral óseo durante y la maduración es la clave para la salud del esqueleto (Gómez-Campos, Arruda, & Cossio-Bolaños, 2017).

Una persona normal se ve afectando por la deficiencia de minerales, entonces puede ser afectado por una enfermedad de salud ósea. La pérdida de densidad ósea puede ser un factor clave para la fractura por fragilidad. (Ramkumar, 2018). Ciertos factores influyen en la pérdida del deterioro del esqueleto, como la obesidad o algún tratamiento médico y el consumo inadecuado de vitaminas o calcio (Gómez-Campos, Arruda, & Cossio-Bolaños, 2017).

Se ha demostrado que la evaluación de salud ósea en niños permite identificar los bajos niveles de densidad mineral ósea (DMO o por sus siglas en ingles bone mineral density BMD) y esto trae como consecuencia algunas enfermedades como la osteopenia y la osteoporosis (Gómez-Campos, Arruda, & Cossio-Bolaños, 2017).

Para poder calcular la densidad de mineral ósea se han usado ecuaciones de regresión propuestas, estas ecuaciones tienen un poder explicativo del 76 al 84%, la fórmula es la siguiente:

Tabla N° 1

Ecuación de Regresión para estimar la densidad mineral ósea basada en la maduración biológica e indicadores antropométricos en hombres.

Ecuaciones	VIF	R	R^2	SEE	p
Hombres					
DMO = 0.605+0.056*APHV+0.008*LongitudAntebrazo+0.022*DiámetroFémur					
APHV	4.034	0.89	0.79	0.10	0.000
LongitudAntebrazo	4.099				
DiámetroFémur	1.867				

Fuente: (Gómez-Campos, Andruske, Arruda, Urra Albornoz & Cossio-Bolaños, 2017).

Tabla N° 2

Ecuación de Regresión para estimar la densidad mineral ósea basada en la maduración biológica e indicadores antropométricos en mujeres.

Ecuaciones	VIF	R	R^2	SEE	p
Mujeres					
DMO = 0.469+0.027* Longitud Antebrazo+0.019*DiámetroFémur					
APHV	3.15	0.87	0.76	0.08	0.000
LongitudAntebrazo	2.963				
DiámetroFémur	1.781				

Fuente: (Gómez-Campos, Andruske, Arruda, Urra Albornoz & Cossio-Bolaños, 2017).

También se utilizó los puntos de corte para poder clasificar el conjunto de datos obtenidos mediante los percentiles divididos entre hombres y mujeres, a su vez ordenados por edades, lo cual nos indica en qué condición está cada persona evaluada con respecto a los resultados obtenidos de su densidad mineral ósea como se puede apreciar en los siguientes cuadros

Tabla N° 3

Valores y distribución de percentiles en base a la densidad mineral ósea del total de niños y adolescentes hombres basados en su edad.

Edad	L	M	S	P3	P5	P15	P25	P50	P75	P85	P95	P97
Hombres												
4.0–4.9	-0.0074	0.4999	0.00065	0.44	0.45	0.47	0.48	0.50	0.52	0.54	0.56	0.57
5.0–5.9	-0.0089	0.5430	0.00065	0.48	0.49	0.51	0.52	0.54	0.57	0.58	0.61	0.62
6.0–6.9	-0.0102	0.5865	0.00065	0.52	0.53	0.55	0.56	0.59	0.61	0.63	0.66	0.67
7.0–7.9	-0.0110	0.6306	0.00065	0.56	0.57	0.59	0.60	0.63	0.66	0.68	0.71	0.72
8.0–8.9	-0.0109	0.6758	0.00066	0.60	0.61	0.63	0.65	0.68	0.71	0.73	0.76	0.77
9.0–9.9	-0.0098	0.7241	0.00065	0.64	0.65	0.68	0.69	0,.72	0.76	0.78	0.81	0.83
10.0–10.9	-0.0074	0.7764	0.00065	0.69	0.70	0.73	0.74	0.78	0.81	0.83	0.87	0.88
11.0–11.9	-0.0027	0.8316	0.00064	0.74	0.75	0,78	0.80	0.83	0.87	0.89	0.92	0.94
12.0–12.9	0.0033	0.8886	0.00062	0.79	0.80	0.83	0.85	0.89	0.93	0.95	0.98	1.00
13.0–13.9	0.0090	0.9461	0.00059	0.84	0.86	0.89	0.91	0.95	0.98	1.00	1.04	1.05
14.0–14.9	0.0132	10.023	0.00055	0.90	0.91	0.94	0.96	1.00	1.04	1.06	1.09	1.11
15.0–15.9	0.0150	10.527	0.00052	0.95	0.96	1.00	1.02	1.05	1.09	1.11	1.14	1.15
16.0–16.9	0.0144	10.962	0.00048	1.00	1.01	1.04	1.06	1.10	1.13	1.15	1.18	1.19
17.0–17.9	0.0124	11.327	0.00044	1.04	1.05	1.08	1.10	1.13	1.17	1.18	1.21	1.23
18.0–18.9	0.0101	11.657	0.0004	1.08	1.09	1.12	1.13	1.17	1.20	1.21	1.24	1.25

Fuente: (Gómez-Campos, Andruske, Arruda, Urra Albornoz & Cossio-Bolaños, 2017).

Tabla N° 4

Valores y distribución de percentiles en base a la densidad mineral ósea del total de niñas y adolescentes mujeres basados en su edad.

Edad	L	M	S	P3	P5	P15	P25	P50	P75	P85	P95	P97
Mujeres												
4.0–4.9	0.0209	0.5280	0.00044	0.48	0.49	0.50	0.51	0.53	0.54	0.55	0.57	0.57
5.0–5.9	0.0195	0.5611	0.00044	0.51	0.52	0.53	0.54	0.56	0.58	0.59	0.60	0.61
6.0–6.9	0.0179	0.5956	0.00045	0.54	0.55	0.57	0.58	0.60	0.61	0.62	0.64	0.64
7.0–7.9	0.0164	0.6322	0.00045	0.58	0.58	0.60	0.61	0.63	0.65	0.66	0.68	0.68
8.0–8.9	0.0152	0.6710	0.00045	0.61	0.62	0.64	0.65	0.67	0.69	0.70	0.72	0.73
9.0–9.9	0.0142	0.7118	0.00045	0.65	0.66	0.68	0.69	0.71	0.73	0.75	0.76	0.77
10.0–10.9	0.0138	0.7539	0.00045	0.69	0.70	0.72	0.73	0.75	0.78	0.79	0.81	0.82
11.0–11.9	0.0139	0.7962	0.00045	0.73	0.74	0.76	0.77	0.80	0.82	0.83	0.85	0.86
12.0–12.9	0.0144	0.8361	0.00045	0.76	0.77	0.80	0.81	0.84	0.86	0.87	0.90	0.91
13.0–13.9	0.0146	0.8710	0.00044	0.80	0.81	0.83	0.84	0.87	0.90	0.91	0.93	0.94
14.0–14.9	0.0134	0.9012	0.00044	0.83	0.84	0.86	0.87	0.90	0.93	0.94	0.97	0.98
15.0–15.9	0.0101	0.9281	0.00044	0.85	0.86	0.89	0.90	0,.93	0.96	0.97	0.99	1.00
16.0–16.9	0.0051	0.9524	0.00043	0.88	0.89	0.91	0.92	0.95	0.98	1.00	1.02	1.03
17.0–17.9	-0.000	0.9744	0.00043	0.90	0.91	0.93	0.95	0.97	1.00	1.02	1.05	1.06
18.0–18.9	-0.006	0.9951	0.00042	0.92	0.93	0.95	0.97	1.00	1.02	1.04	1.07	1.08

Fuente: (Gómez-Campos, Andruske, Arruda, Urra Albornoz & Cossio-Bolaños, 2017).

CAPITULO III

3. PROCESO DE EXTRACIÓN DE CONOCIMIENTO

En este capítulo se verá el análisis de los 5 algoritmos de minería de datos de clasificación; el análisis y posterior cuadro comparativo se realizó en base a la eficacia que tendrían los algoritmos aplicados a la base de datos obtenidas de los colegios Inmaculada Concepción y Jorge Basadre del sector de Arequipa.

3.1. ALGORITMOS

3.1.1. J48 – GRAFT TREE

El algoritmo J48 – Graft genera un árbol de decisión injertado, la técnica de injerto es un proceso inductivo que añade a los nodos a árboles de decisión inferidos con el propósito de reducir los errores de predicción. El árbol J48 – Graft clasifica la región del espacio multidimensional de atributos no ocupados por los ejemplos de entrenamiento, este proceso frecuentemente mejora la predicción. Algunos resultados que muestra este algoritmo de injerto J48, logra una sensibilidad de 76.50%, especificidad de 78.60%, precisión de 76.52%, error absoluto medio de 0.32+, error cuadrático medio de 0.42+ y error absoluto relativo de 71.33+, respectivamente (Wisaeng, 2013).

El concepto de árbol injertado se basa en el deseo de descartar el método de “lo simple es lo mejor” para la selección de un buen árbol. En contraste, en el árbol injertado, la atención se centra en el hecho de que los objetos similares tienden a tener la más alta probabilidad de pertenencia a la misma clase; en otras palabras, si el resultado final es un mejor modelo de clasificación, la necesidad de producir árboles más complejos se elimina. El injerto es un postproceso que se puede aplicar fácilmente a árboles de decisión. Su principal objetivo es la reclasificación de regiones de un espacio de instancia donde no existan datos de entrenamiento o donde solo haya datos mal clasificados. El injerto identifica los cortes más adecuados de las regiones hoja actuales y luego hace el proceso de ramificación con el fin de crear nuevas hojas con clasificaciones las cuales difieren de las originales. En este proceso el árbol se hace más complejo naturalmente; sin embargo, sólo la ramificación que no introduce errores de clasificación en datos que ya hayan sido clasificados es considerada, asegurando que el nuevo árbol reduce errores (Hayashi, 2015).

3.1.2. DECISION TABLES

Las tablas de decisión son una reprensión tabular del conocimiento en la que el estado de un conjunto de condiciones determina conjuntamente uno o más resultados, si se usan de modelos de clasificación, las condiciones toman el valor de los atributos y cada conjunto de condiciones está asociado con una predicción de clase. Veremos las tablas de un solo hit, con reglas mutuamente excluyentes (celdas), ya que no tienen redundancia y facilitan la interpretación de las tablas de decisiones por parte del usuario. Las tablas de decisión se pueden inducir a partir de datos de diferentes maneras, por ejemplo, pueden inducirse realizando una selección de atributos con un envoltorio o un enfoque de filtro. El algoritmo se puede dividir en dos partes: construcción de un gráfico de dependencia etiquetado y producción de la tabla de decisiones del gráfico etiquetado. Comenzamos con una tabla de reglas. Se requiere una estructura de datos auxiliar: un recuento de alternativas, que contendrá para cada aserción el número de reglas que tienen esa aserción como consecuencia. Las dimensiones del problema serán: (Colomb & Chung, 2000).

- r el número de reglas
- un promedio de afirmaciones por regla
- d la máxima profundidad de razonamiento

Para construir el gráfico de dependencia, primero contamos el número de alternativas para cada aserción. Esto requiere un paso por regla, por lo tanto, es O (r). Los hechos serán tomados como teniendo cero alternativas. Luego procedemos a identificar las reglas con la etiqueta I, que son aquellas reglas cuyos antecedentes tienen cero alternativas (Freitas, 2010). Cuando una regla está etiquetada, decrementamos el número de alternativas para su aseveración consecuente, y también registramos un puntero a la regla en una estructura de datos asociada con la aserción. Este paso requiere el examen de cada antecedente en cada regla, y por lo tanto es O (ar). Al final del paso, las aserciones adicionales tienen un recuento alternativo de cero (se sigue del lema 2). El gráfico se puede construir y etiquetar completamente en un solo paso para cada etiqueta posible, limitada por la máxima profundidad de razonamiento (Freitas, 2010).

3.1.3. C4.5 o J48 Tree

Este algoritmo es un sucesor de ID3 desarrollado por Quinlan Ross. También está basado en el algoritmo de Hunt. C4.5 maneja los atributos categóricos y continuos para construir un árbol de decisión. Para manejar los atributos continuos, C4.5 divide los valores de los atributos en dos particiones basadas en el umbral seleccionado de tal manera que todos los valores por encima del umbral como un niño y el resto como otro niño. También maneja los valores de atributo faltantes. C4.5 usa la razón de ganancia como una medida de selección de atributo para construir un árbol de decisión. Elimina la parcialidad de la ganancia de información cuando hay muchos valores de resultado de un atributo. Al principio, calcule la relación de ganancia de cada atributo. El nodo raíz será el atributo cuya relación de ganancia sea máxima. C4.5 utiliza la poda pesimista para eliminar ramas innecesarias en el árbol de decisión para mejorar la precisión de la clasificación. (Yadav & Brijesh, 2012), también obtuvo los siguientes resultados aplicado a un dataset real orientado al campo educacional universitario: en precisión alcanzó un 67,78% para instancias correctamente clasificadas y un 32,22% para instancias incorrectamente clasificadas, su tiempo de ejecución fue de 0.003 segundos.

Algunas mejoras del algoritmo C4.5 (Jaqueline, 2015):

- Se manejan atributos continuos.
- Se mejora la eficiencia computacional.
- Se lleva un control de qué tan profundo va a ser el tamaño del árbol de decisión construido.
- Reducir errores en la poda.
- Se evita el sobreajuste (overfitting) de datos.
- Manejo de atributos con diferentes valores.
- Manejo de datos de entrenamiento con valores desconocidos.

3.1.4. PART

Este algoritmo genera una lista de decisión sin restricciones usando el procedimiento de divide y vencerás, evita el paso de optimización global que se usa en las reglas del C4,5. Además construye un árbol de decisión parcial para obtener una regla y así poder podar una rama (una regla) es necesario que todas

sus implicaciones sean conocidas. El PART evita la generalización precipitada y usa los mismos mecanismos que el C4.5 para generar un árbol. La hoja con máxima cobertura se convierte en una regla y los valores ausentes de los atributos se tratan como en el C4.5, la instancia se divide en piezas (Jaqueline, 2015).

Respecto al tiempo máximo para generar una regla, ese el mismo que para construir un árbol podado, esto ocurre cuando los datos tienen ruido. En el mejor de los casos el tiempo necesario es el mismo que para generar una regla sencilla y mayormente se da cuando los datos no presentan ruido (Jaqueline, 2015).

3.1.5. BAYESNET

Una red bayesiana es un gráfico acíclico dirigido anotado que codifica una distribución de probabilidad conjunta sobre un conjunto de variables aleatorias U. Formalmente, una red bayesiana para U es un par B = hG, Oi. El primer componente, G, es un gráfico acíclico dirigido cuyos vértices corresponden a las variables aleatorias X1,...., Xn y cuyos bordes representan dependencias directas entre las variables. El grafico G codifica las suposiciones de independencia: cada variable Xi es independiente de sus no descendientes dados sus padres en G. El segundo componente del par representa el conjunto de parámetros que cuantifica la red.

Clasificador bayesiano, este enfoque se justifica por la corrección asintótica del procedimiento de aprendizaje bayesiano. Dado un gran conjunto de datos, la red aprendida será una aproximación cercana para la distribución de probabilidad que gobierna el dominio (asumiendo que los casos se muestrean de forma independiente de una distribución fija).

CAPITULO IV

4. DESARROLLO DE LA PROPUESTA

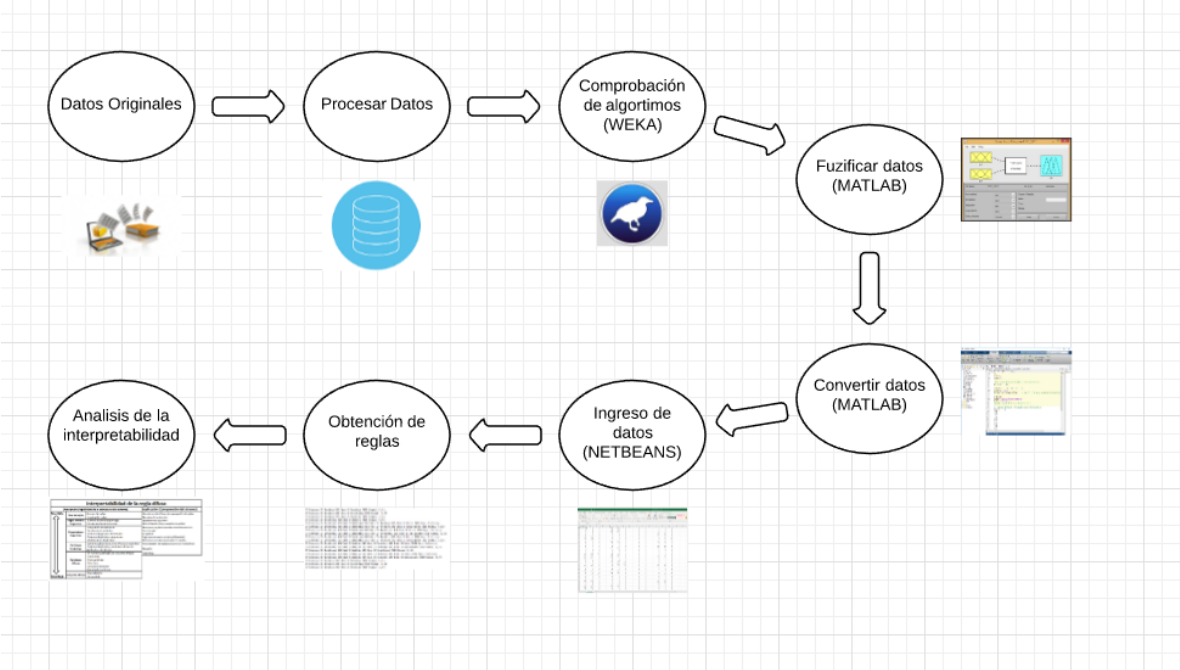


Figura Nº 3: Proceso de desarrollo.

Fuente: Elaboración propia.

Como se muestra en la figura N° 3, todo el proceso que se usara para llegar al resultado deseado, lo primero es el procesamiento de datos mediante la eliminación de vacíos, luego de haber hecho la limpieza de datos, estos se ingresaran a la herramienta WEKA, con el fin de analizar algoritmos que nos permitan encontrar el más eficaz. Después de haber analizado los algoritmos y obtener el más eficaz, se utilizará la herramienta de MATLAB, para hacer la fuzzificación de datos, particiones y los datos de entrada y salida. Luego con la misma herramienta, defuzzificamos los datos y aseguramos la integridad de estos. Después, los datos son ingresados al programa de apoyo, se utilizó la herramienta de NETBEANS y la cual nos muestra las reglas difusas, para finalizar se analizará la interpretabilidad de las reglas difusas obtenidas.

En este capítulo veremos el análisis experimental realizado a los 5 algoritmos, se comprobará el algoritmo más eficaz.

4.1. REPOSITORIO DE DATOS

4.1.1. DATOS OBTENIDOS

Los datos obtenidos, fueron recopilados de los alumnos de los colegios Inmaculada Concepción y Jorge Basadre del sector de Arequipa de la provincia de Arequipa.

4.1.2. DICCIONARIO DE DATOS

Tabla N° 5 Diccionario de datos.

ID	Atributo	Tipo	Descripción
0	Nombre	Categórico	Nombres y Apellidos del alumno
1	Fecha_Nacimiento	Fecha	Fecha de nacimiento del alumno
2	Genero	Binario	Genero del alumno
3	Practica_deporte	Binario	Practica algún deporte en la semana
4	Fuma	Binario	Fuma
5	Desayuna_todos_los_dias	Binario	Toma desayuno todos los días
6	Estatura	Numérico	Estatura en centímetros
7	Peso	Numérico	Peso en kilos
8	Estatura_parado	Numérico	Estatura parado en centímetros
9	Estatura_sentado	Numérico	Estatura sentado en centímetros
10	Circunferencia_abdominal	Numérico	Circunferencia abdominal en centímetros
11	Codo	Numérico	Medida del codo en centímetros
12	Muñeca	Numérico	Medida de la muñeca en centímetros
13	Rodilla	Numérico	Medida de la rodilla en centímetros
14	Tobillo	Numérico	Medida del tobillo en centímetros

15	Antebrazo_derecho	Numérico	Medida del antebrazo derecho en centímetros
16	Pierna	Numérico	Medida del grueso de la pierna en centímetros
17	Saturación_O2	Numérico	Cantidad de oxígeno en la sangre
18	Frecuencia_cardiaca	Numérico	Latidos por minuto
19	Flujo_respiratorio	Numérico	Índice aceptado como medida independiente de la función pulmonar.
20	Edad	Numérico	Edad del alumno
21	PHV	Numérico	peak height velocity (velocidad de altura pico)
22	BMD	Numérico	Bone mineral density (Densidad de mineral óseo)
23	Resultado	Categórico	El resultado obtenido, que pueden ser tres: Normal, Osteopenia y Osteoporosis.

4.1.3. PREPROCESAMIENTO

Se ha preprocesado los datos, como algunos que son impuros, esto quiere decir, los datos incompletos, datos con ruidos y/o datos inconsistentes. El preprocesamiento de datos genera un conjunto de datos más pequeño que el original, lo cual puede mejorar la eficiencia del proceso de minería de datos, se utilizó la eliminación de vacíos.

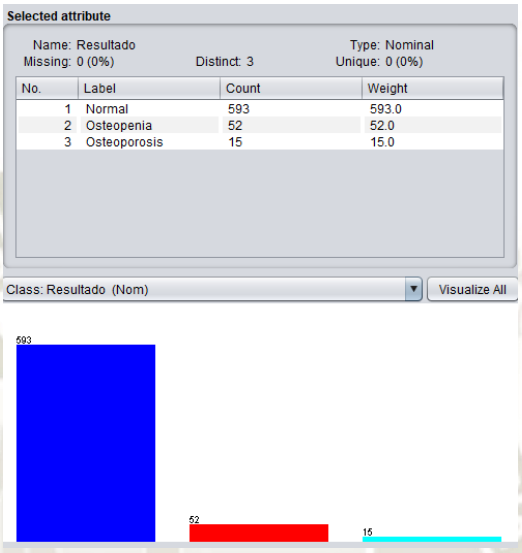


Figura Nº 4: Cuadro de resultado.

Fuente: WEKA

4.1.4. APLICACIÓN DE LOS ALGORITMOS

El objetivo es predecir el número de alumnos que puedan sufrir alguna enfermedad ósea como osteopenia u osteoporosis a partir de los datos obtenidos de los colegios Inmaculada Concepción y Jorge Basadre del sector de Arequipa.

4.1.4.1. ALGORITMO J48 – GRAFT TREE

```
Time taken to build model: 0.05 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances         610               92.4242 %
Incorrectly Classified Instances        50                7.5758 %
Kappa statistic                          0.4032
Mean absolute error                      0.0869
Root mean squared error                  0.2084
Relative absolute error                 69.2708 %
Root relative squared error             83.7121 %
Total Number of Instances              660

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,701    0,927      1,000    0,962      0,526    0,784     0,954     Normal
                 0,192    0,005    0,769      0,192    0,308      0,363    0,763     0,309     Osteopenia
                 0,467    0,000    1,000      0,467    0,636      0,679    0,899     0,623     Osteoporosis
Weighted Avg.    0,924    0,631    0,916      0,924    0,903      0,517    0,785     0,895

=== Confusion Matrix ===

   a   b   c   <-- classified as
 593   0   0 |   a = Normal
  42  10   0 |   b = Osteopenia
   5   3   7 |   c = Osteoporosis
```

Figura Nº 5: Resultados del algoritmo J48- Graft Tree, eficiencia.

Fuente WEKA.

Como se puede observar en la figura N° 5, que un 92.4242% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es relativamente alta para las 3 clases del atributo “Resultado”.

También se puede ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que se puede decir que la clasificación es confiable.

```
J48graft pruned tree
------------------

Estatura Sentado <= 80
|   Estatura Sentado <= 66: Osteoporosis  (5.0)
|   Estatura Sentado > 66
|   |   Antebrazo Derecho (CM) <= 21.5
|   |   |   Estatura Sentado <= 78.5
|   |   |   |   Edad <= 15: Osteopenia  (7.0/1.0)
|   |   |   |   Edad > 15: Osteoporosis  (2.0)
|   |   |   Estatura Sentado > 78.5: Normal (2.0)
|   |   Antebrazo Derecho (CM) > 21.5: Normal (101.0/24.0)
Estatura Sentado > 80
|   BMD <= 0.87
|   |   Edad <= 14: Normal (26.0/1.0)
|   |   Edad > 14
|   |   |   Rodilla (CM) Femur <= 13.55
|   |   |   |   Peso <= 69.85
|   |   |   |   |   Estatura Sentado <= 86.05
|   |   |   |   |   |   Pierna (CM) <= 48.25
|   |   |   |   |   |   |   Muñeca (CM) <= 5.35
|   |   |   |   |   |   |   |   Tobillo (CM) <= 6.25
|   |   |   |   |   |   |   |   |   Estatura Parado(m) <= 1.725
|   |   |   |   |   |   |   |   |   |   Estatura (cm) <= 172.5
|   |   |   |   |   |   |   |   |   |   |   Hace Deporte? = B: Normal (0.0|88.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   Hace Deporte? != B
|   |   |   |   |   |   |   |   |   |   |   |   PHV <= 1.405
|   |   |   |   |   |   |   |   |   |   |   |   |   Circunferencia Abdominal (CM) <= 87.5
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Codo (CM) <= 6.05
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Antebrazo Derecho (CM) <= 26.05
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Flujo Espiratorio <= 430
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Desayuna todos los dias? = B: Normal (0.0|20.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Desayuna todos los dias? != B
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Género = M: Normal (0.0|174.0/8.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Género != M
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Frecuencia Cardiaca <= 98.5
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Edad <= 17.5
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Saturación O2 (%) <= 117.5
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   PHV <= -0.95: Normal (0.0|16.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   PHV > -0.95
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Rodilla (CM) Femur <= 8.95: Normal (0.0|76.0/8.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Rodilla (CM) Femur > 8.95
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Frecuencia Cardiaca <= 67.5: Normal (0.0|69.0/8.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Frecuencia Cardiaca > 67.5
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Flujo Espiratorio <= 175: Normal (0.0|19.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Flujo Espiratorio > 175

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura Parado(m) <= 1.515: Normal (0.0|23.0/4.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura Parado(m) > 1.515
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura (cm) <= 151.5: Normal (0.0|23.0/4.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura (cm) > 151.5: Osteopenia  (6.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Saturación O2 (%) > 117.5: Normal (0.0|9.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Edad > 17.5: Normal (0.0|12.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Frecuencia Cardiaca > 98.5: Normal (0.0|17.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Flujo Espiratorio > 430: Normal (0.0|52.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Antebrazo Derecho (CM) > 26.05: Normal (0.0|56.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Codo (CM) > 6.05: Normal (0.0|93.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Circunferencia Abdominal (CM) > 87.5: Normal (0.0|39.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   PHV > 1.405: Normal (0.0|121.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura (cm) > 172.5: Normal (0.0|43.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Estatura Parado(m) > 1.725: Normal (0.0|43.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Tobillo (CM) > 6.25: Normal (0.0|135.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Muñeca (CM) > 5.35: Normal (0.0|90.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   Pierna (CM) > 48.25: Normal (0.0|45.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   Estatura Sentado > 86.05: Normal (0.0|140.0/2.0)
|   |   |   |   |   |   |   |   |   |   |   |   Peso > 69.85: Normal (0.0|49.0)
|   |   |   |   |   |   |   |   |   |   |   Rodilla (CM) Femur > 13.55: Normal (0.0|86.0)
|   BMD > 0.87: Normal (511.0/22.0)

Number of Leaves  :     33

Size of the tree :      65
```

Figura Nº 6: Fragmento del árbol de clasificación masculino generado, análisis del conocimiento J48 – graft tree.

Fuente WEKA

En este fragmento podemos interpretar, por ejemplo, el algoritmo lo divide en 2 partes generales, es decir divide en dos grupos, que cuando el alumno tiene una estatura sentado de menor o igual a 80 y el otro es mayor a 80, el primer conjunto, toma como base los atributos de edad y antebrazo derecho para poder predecir si tiene osteopenia, osteoporosis o si esta normal. En el segundo grupo se puede apreciar que toma los demás valores, comenzando

por el BMD y así va usando los demás atributos. También se puede observar en este fragmento, el algoritmo J48 – graft tree emplea todos los atributos que se han recolectado de los estudiantes y mostrar la relación entre ellos y el atributo objetivo; es decir, las distintas ramas del árbol muestran como ciertos valores para cada atributo influyen, sea de manera positiva o negativa, al atributo resultado.

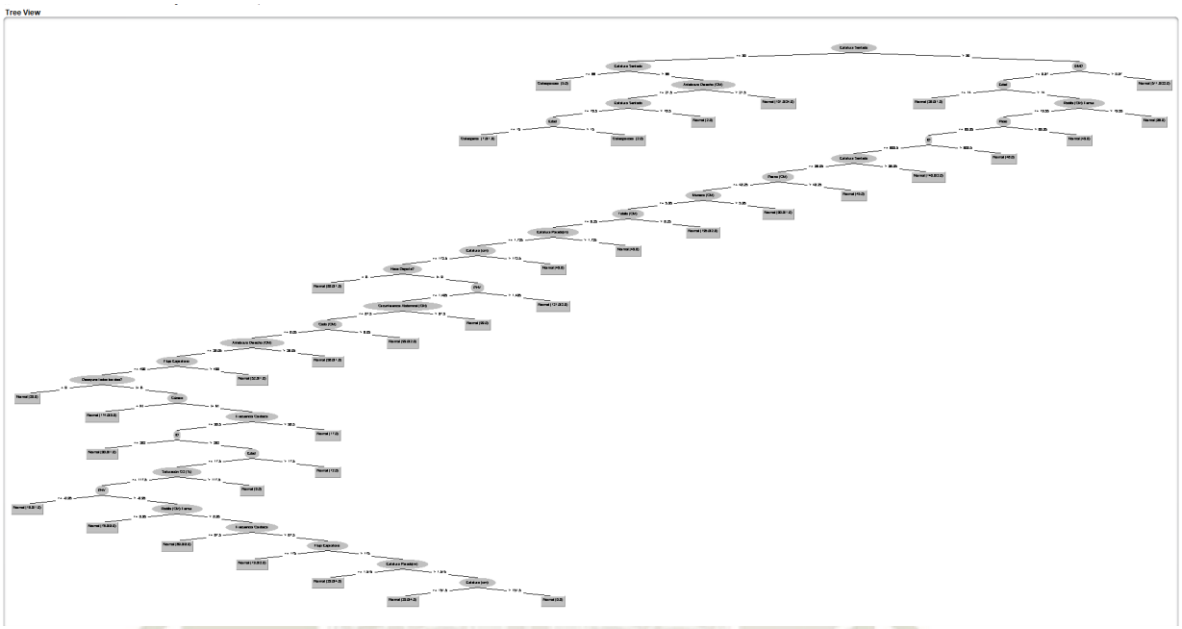


Figura Nº 7: Visualización del árbol del algoritmo Graft Tree.

Fuente Weka

Como la figura indica, se puede apreciar la visualización del árbol obtenido por el algoritmo graft tree.

4.1.4.2. DECISION TABLES

```
Time taken to build model: 0.2 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances         603               91.3636 %
Incorrectly Classified Instances        57                8.6364 %
Kappa statistic                          0.2681
Mean absolute error                      0.1182
Root mean squared error                  0.2217
Relative absolute error                 94.2029 %
Root relative squared error             89.0303 %
Total Number of Instances              660

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,998    0,821    0,915      0,998    0,955      0,386    0,834     0,971     Normal
                 0,135    0,003    0,778      0,135    0,230      0,305    0,819     0,350     Osteopenia
                 0,267    0,000    1,000      0,267    0,421      0,512    0,856     0,419     Osteoporosis
Weighted Avg.    0,914    0,738    0,906      0,914    0,886      0,382    0,833     0,909

=== Confusion Matrix ===

   a   b   c   <-- classified as
 592   1   0 |   a = Normal
  45   7   0 |   b = Osteopenia
  10   1   4 |   c = Osteoporosis
```

Figura Nº 8: Resultados del algoritmo Decision Tables, eficiencia.

Fuente WEKA.

Podemos ver que un 91.3636% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es relativamente alta para las 3 clases del atributo “Resultado”.

También se puede ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que se puede decir que la clasificación es confiable.

```
=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 660
Number of Rules : 24
Non matches covered by Majority class.
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 174
        Merit of best subset found:   91.364
Evaluation (for feature selection): CV (leave one out)
Feature set: 3,4,10,13,24
```

Figura Nº 9: Clasificación del modelo Decision Tables.

Fuente WEKA.

En esta figura, se puede deducir que, se han evaluado 660 instancias en las cuales se obtuvo 24 reglas con las que el algoritmo llego al resultado esperado, la búsqueda de dirección fue hacia adelante, la búsqueda paso el 5to nodo de expansión, el total de subconjuntos evaluados fue de 174, el mejor subconjunto encontrado fue de 91.364, la evaluación para la selección del conjunto fue la de dejar uno a fuera (leave one out).

Para concluir, el algoritmo Decision Tables mostro muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo “Resultado”. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivo fue excesivamente alta para las clases del atributo y el tiempo que le tomo al algoritmo construir el modelo fue muy bueno, siendo tan solo 0.01 segundos.

4.1.4.3. ALGORITMO C4.5 O J48 TREE

```
Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         610               92.4242 %
Incorrectly Classified Instances        50                7.5758 %
Kappa statistic                          0.4032
Mean absolute error                      0.0869
Root mean squared error                  0.2084
Relative absolute error                 69.2708 %
Root relative squared error             83.7121 %
Total Number of Instances              660

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1,000    0,701    0,927      1,000    0,962      0,526    0,784     0,954     Normal
                 0,192    0,005    0,769      0,192    0,308      0,363    0,763     0,309     Osteopenia
                 0,467    0,000    1,000      0,467    0,636      0,679    0,899     0,623     Osteoporosis
Weighted Avg.    0,924    0,631    0,916      0,924    0,903      0,517    0,785     0,895

=== Confusion Matrix ===

   a   b   c   <-- classified as
 593   0   0 |   a = Normal
  42  10   0 |   b = Osteopenia
   5   3   7 |   c = Osteoporosis
```

Figura Nº 10: Resultados del algoritmo J48, eficiencia.

Fuente WEKA.

Podemos ver que un 92.4242% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es moderadamente alta para las 3 clases del atributo “resultado”.

También se puede ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que se puede decir que la clasificación es confiable.

```
=== Classifier model (full training set) ===

J48 pruned tree
------------------

Estatura Sentado <= 80
|   Estatura Sentado <= 66: Osteoporosis  (5.0)
|   Estatura Sentado > 66
|   |   Antebrazo Derecho (CM) <= 21.5
|   |   |   Estatura Sentado <= 78.5
|   |   |   |   Edad <= 15: Osteopenia  (7.0/1.0)
|   |   |   |   Edad > 15: Osteoporosis  (2.0)
|   |   |   Estatura Sentado > 78.5: Normal (2.0)
|   |   Antebrazo Derecho (CM) > 21.5: Normal (101.0/24.0)
Estatura Sentado > 80
|   BMD <= 0.87
|   |   Edad <= 14: Normal (26.0/1.0)
|   |   Edad > 14: Osteopenia  (6.0/2.0)
|   BMD > 0.87: Normal (511.0/22.0)

Number of Leaves  :     8

Size of the tree :      15
```

Figura Nº 11: Fragmento del árbol de clasificación del algoritmo J48, análisis del conocimiento.

Fuente WEKA.

En este fragmento podemos interpretar, por ejemplo, toma la estatura sentado como base para dividirlo en dos grupos, la estatura sentado menor o igual a 80 es el primer grupo y el segundo grupo es mayor a 80. En el primer grupo, utiliza algunos datos aparte de la estatura sentado, usa antebrazo derecho y edad, con estos valores el algoritmo ya puede predecir si los alumnos van a sufrir alguna enfermedad ósea como osteoporosis, osteopenia o si el alumno se encuentra normal.

En el segundo grupo, se puede interpretar, que aparte de usar la estatura sentado, para clasificar utiliza el BMD y la edad, con esto ya puede predecir si un alumno sufrirá alguna enfermedad ósea como osteoporosis, osteopenia o si esta normal.

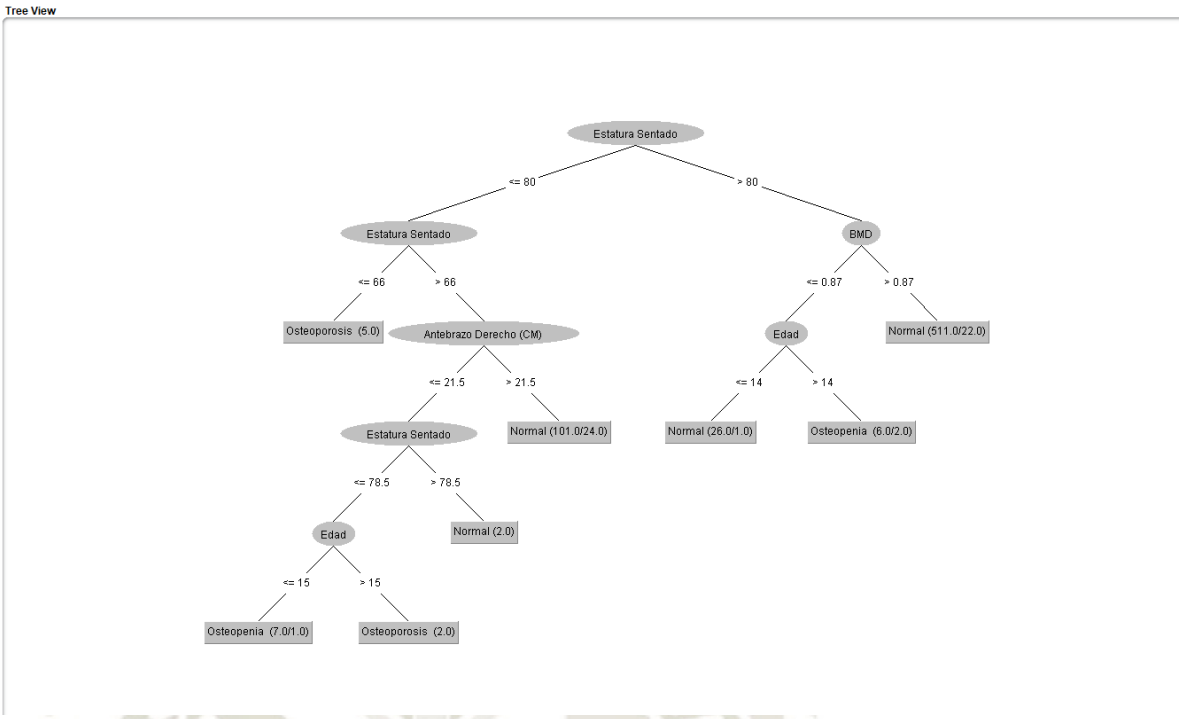


Figura Nº 12: Visualización del árbol generado del algoritmo J48.

Fuente WEKA.

Como se muestra en la figura N° 12, se puede visualizar el árbol obtenido por el algoritmo J48.

Para concluir, el algoritmo J48 mostro muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo "Resultado". En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivo fue excesivamente alta para las clases del atributo y el tiempo que le tomo al algoritmo construir el modelo fue muy bueno, siendo tan solo 0 segundos.

4.1.4.4. ALGORITMO PART

```
Time taken to build model: 0.02 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances         608               92.1212 %
Incorrectly Classified Instances        52                7.8788 %
Kappa statistic                          0.3615
Mean absolute error                      0.0853
Root mean squared error                  0.2065
Relative absolute error                 68.0123 %
Root relative squared error             82.9481 %
Total Number of Instances              660

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,998    0,761    0,921      0,998    0,958      0,452    0,874     0,974     Normal
                 0,135    0,002    0,875      0,135    0,233      0,327    0,855     0,303     Osteopenia
                 0,600    0,000    1,000      0,600    0,750      0,771    0,930     0,625     Osteoporosis
Weighted Avg.    0,921    0,684    0,919      0,921    0,896      0,449    0,874     0,913

=== Confusion Matrix ===

   a   b   c   <-- classified as
 592   1   0 |   a = Normal
  45   7   0 |   b = Osteopenia
   6   0   9 |   c = Osteoporosis
```

Figura Nº 13: Resultados del algoritmo PART, eficiencia.

Fuente WEKA.

Podemos ver que un 92.1212% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es moderadamente alta para las 3 clases del atributo “Resultado”.

También se puede ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que se puede decir que la clasificación es confiable.

```
=== Classifier model (full training set) ===

PART decision list
------------------

Estatura Sentado > 80 AND
Antebrazo Derecho (CM) > 23.8 AND
BMD > 0.87: Normal (407.0/1.0)

Antebrazo Derecho (CM) > 20.7 AND
Estatura Sentado > 71: Normal (233.0/50.0)

Tobillo (CM) <= 5: Osteopenia  (6.0/1.0)

Codo (CM) > 5.2 AND
Flujo Espiratorio <= 285: Osteoporosis  (5.0)

Pierna (CM) > 47.5: Osteoporosis  (4.0)

Codo (CM) > 5.2: Normal (3.0)

: Osteopenia  (2.0)

Number of Rules  :      7
```

Figura Nº 14: Fragmento de clasificación del algoritmo PART, análisis del conocimiento.

Fuente WEKA.

En este caso se tienen en cuenta, nos muestra que se obtuvieron 7 reglas, las cuales son fáciles de interpretar. Se puede observar que comienza con la estatura sentado, un conector y luego usa el antebrazo derecho, después de esto usa otro conecto y utiliza el atributo BMD, lo cual nos muestra unos datos en los cuales se obtiene información valiosa para poder predecir si el alumno está en una condición normal.

En la segunda regla, nos muestra el atributo de antebrazo derecho, con un conector “and”, para poder usar conjuntamente la estatura sentado, viendo estos datos que muestra la figura 9, se puede llegar a predecir que el alumno está en una condición normal. Y así sucesivamente se puede hacer con las demás reglas obtenidas en este algoritmo.

Para concluir, el algoritmo PART mostro muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo “Resultado”; el conocimiento generado ha demostrado ser relevante para la toma de decisiones, por ejemplo, determinar que tanto influye la estatura

sentado o el antebrazo derecho, entre otros atributos. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivo fue ligeramente alta para las clases del atributo y el tiempo que le tomo al algoritmo construir el modelo fue muy bueno, siendo tan solo 0 segundos.

4.1.4.5. BAYESNET

```
Time taken to build model: 0.03 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0.01 seconds

=== Summary ===

Correctly Classified Instances         600               90.9091 %
Incorrectly Classified Instances        60                9.0909 %
Kappa statistic                          0.6242
Mean absolute error                      0.0694
Root mean squared error                  0.211
Relative absolute error                 55.2836 %
Root relative squared error             84.7176 %
Total Number of Instances              660

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,912    0,104    0,987      0,912    0,948      0,650    0,976     0,997     Normal
                 0,865    0,086    0,464      0,865    0,604      0,593    0,974     0,860     Osteopenia
                 0,933    0,002    0,933      0,933    0,933      0,932    1,000     0,992     Osteoporosis
Weighted Avg.    0,909    0,101    0,945      0,909    0,921      0,652    0,976     0,986

=== Confusion Matrix ===

   a   b   c   <-- classified as
 541  51   1 |   a = Normal
   7  45   0 |   b = Osteopenia
   0   1  14 |   c = Osteoporosis
```

Figura Nº 15: Resultados del algoritmo BAYESNET, eficiencia.

Fuente WEKA.

Podemos ver que un 90.9091% de las instancias fueron correctamente clasificadas y que la tasa de verdaderos positivos es moderadamente alta para las 3 clases del atributo “Resultado”.

También se puede ver que la diagonal central de la matriz de confusión tiene los valores más altos, por lo que se puede decir que la clasificación es confiable.

Probability Distribution Table For Género

'Resultado '	M	F
Normal	0,457	0,543
'Osteopenia '	0,33	0,67
'Osteoporosis '	0,469	0,531

Figura Nº 16: Fragmento del algoritmo BAYESNET, probabilidad según el género.

Fuente WEKA.

La probabilidad de resultado según el género, como se puede observar, en el cuadro "M", tiene una probabilidad de 0.457 que sea normal, el segundo tiene una probabilidad de 0.33 de que sufra osteopenia y por último tiene una probabilidad de 0.469 de que sufra osteoporosis. En la otra parte del cuadro, la del género "F", se puede observar que tiene una probabilidad de 0.543 de que sea normal, en el segundo, tiene una probabilidad de 0.67 que sufra osteopenia y, por último, tiene una probabilidad de 0.531 que sufra osteoporosis.

Como se puede observar del cuadro, el género femenino tiene una alta probabilidad de sufrir enfermedades óseas, en el otro cuadro con el género masculino, se puede observar que tienen muy alta probabilidad de estar normal, y tienen muy bajas probabilidades de sufrir enfermedades óseas.

Para concluir, el algoritmo BAYESNET mostro muy buenos resultados al momento de clasificar todas las instancias teniendo en cuenta el atributo objetivo "Resultado"; el conocimiento generado ha demostrado ser relevante para la toma de decisiones, por ejemplo, determinar la probabilidad que puedan tener los alumnos de la región de Arequipa, según su generó, que porcentaje tiene de tener una enfermedad ósea. En cuanto a la eficiencia del algoritmo, la tasa de verdaderos positivo fue moderadamente alta para las clases del atributo y el tiempo que le tomo al algoritmo construir el modelo fue muy bueno, siendo tan solo 0.01 segundos.

4.1.5. CUADRO COMPARATIVO DE LOS ALGORITMOS

Tabla N° 6 Cuadro Comparativo de los algoritmos.

Algoritmo	Instancias Correctamente Clasificadas	Instancias Incorrectas Clasificadas	Tasa de Verdaderos Positivos	Tasa de Falsos Positivos	Precisión
J 48 – Graf Tree	92.4242	7.5758	0.924	0.631	0.916
Decision Tables	91.3636	8.6364	0.914	0.738	0.906
C4.5	92.4242	7.5758	0.924	0.631	0.916
PART	92.1212	7.8788	0.921	0.684	0.919
BayesNet	90.9091	9.0909	0.909	0.101	0.945

Fuente. Elaboración propia

En la tabla N° 6 como se puede observar, se han colocado los 5 algoritmos que se han analizado y los valores que se han tomado en cuenta para identificar el algoritmo que se usará, se han puesto los valores más relevantes que se pueden obtener de la herramienta WEKA y como resultado se optó por usar el algoritmo **J48 Graf Tree**, ya que demostró tener una ligera ventaja a comparación de los demás.

4.1.6. CONCLUSIÓN DE LOS ALGORTIMOS

Obteniendo los resultados de los algoritmos evaluados en WEKA y con la recomendación del experto, se llegó a la conclusión, el primer valor a evaluar fue las instancias correctamente clasificadas, el segundo valor a tener en consideración es la tasa de verdaderos positivos, el tercero es la precisión y por último la cantidad de atributos que el algoritmo llega a utilizar. Teniendo esto en cuenta se puede llegar a la conclusión que el algoritmo con mejor resultado fue le algoritmo J48 graft tree, teniendo un porcentaje de 92.4242 de instancias correctamente clasificadas, una tasa de verdaderos positivos de 0.924 y una

precisión de 0.916, también teniendo en cuenta el uso de la mayoría de los atributos de los datos procesados, viendo también la forma en que agrupo y formo el árbol, es por eso por lo que se utilizará como modelo.

4.2. MATLAB

Usamos esta herramienta, ya que contiene muchas opciones que nos ayudan al momento de fuzzificar y defuzzificarlos los datos.

4.2.1. FUZZIFICACIÓN DE DATOS

Para fuzzificar los datos, se utilizó Toolbox de Lógica fuzzy, la cual nos permite crear entradas y salida de los datos mediante el método de Mamdani.

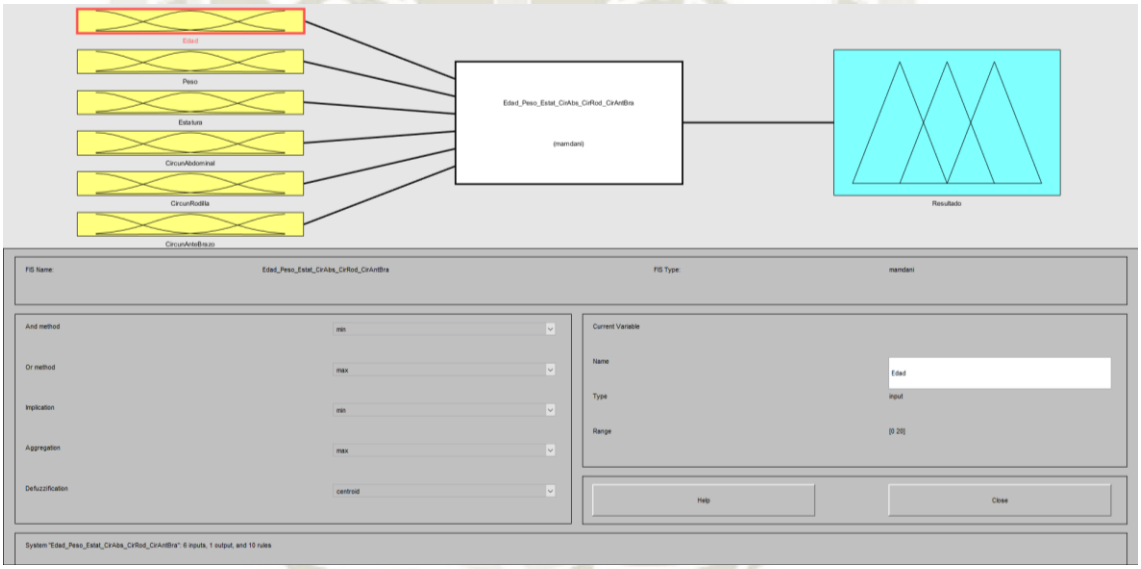


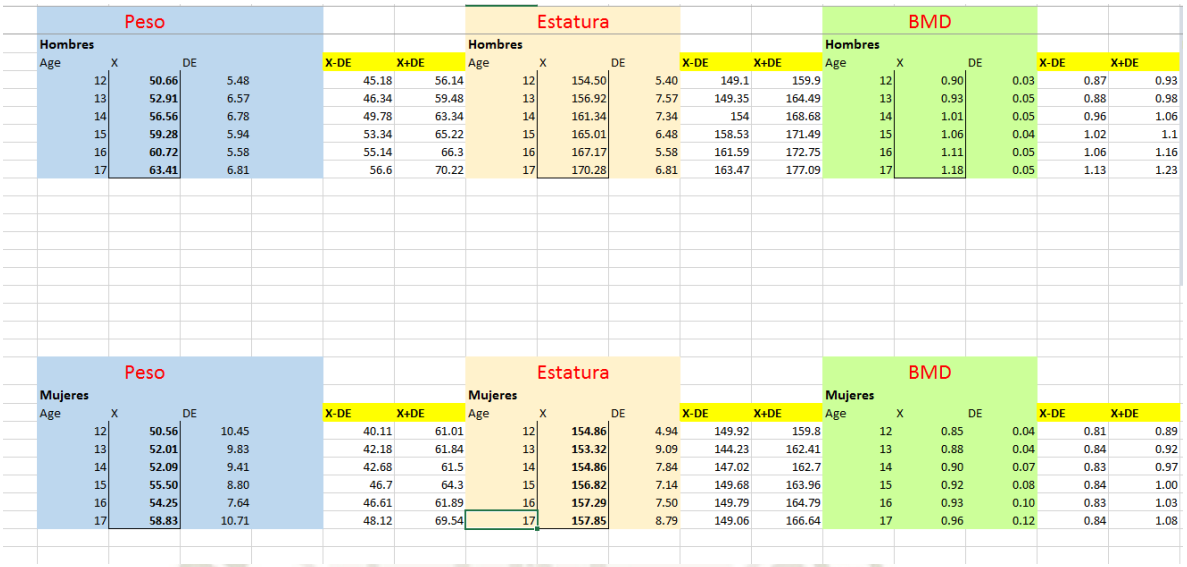
Figura Nº 17: Método de Mamdani, entrada y salida de datos.

Fuente Matlab.

4.2.2. DATOS DE ENTRADA

Con el apoyo del asesor y el experto en el área de medicina, para el proceso de diseño, se consideraron para los datos de entrada lo siguiente: Edad, peso, estatura, circunferencia abdominal, circunferencia de la rodilla (fémur) y circunferencia del antebrazo derecho, estos datos se utilizaron porque, son los que se utilizan en las ecuaciones de regresión para saber la densidad mineral ósea, los cuales están con el tipo de función trapezoidal y cada uno tiene un rango en el que se le considera en bajo, normal y alto, excepto edad, en edad se consideró el grado académico (primaria y secundaria porque nos ayuda a tener particiones fuertes), se tomó estos

tipos de particiones (bajo, normal y alto) por los percentiles, los cuales comienzan con el valor mínimo de densidad ósea y va en aumento a lo normal y alto. Para definir los datos de entrada se utilizaron los puntos de corte, que se obtuvieron de los datos de los estudiantes.



Peso					Estatura					BMD				
Hombres					Hombres					Hombres				
Age	X	DE	X-DE	X+DE	Age	X	DE	X-DE	X+DE	Age	X	DE	X-DE	X+DE
12	50.66	5.48	45.18	56.14	12	154.50	5.40	149.1	159.9	12	0.90	0.03	0.87	0.93
13	52.91	6.57	46.34	59.48	13	156.92	7.57	149.35	164.49	13	0.93	0.05	0.88	0.98
14	56.56	6.78	49.78	63.34	14	161.34	7.34	154	168.68	14	1.01	0.05	0.96	1.06
15	59.28	5.94	53.34	65.22	15	165.01	6.48	158.53	171.49	15	1.06	0.04	1.02	1.1
16	60.72	5.58	55.14	66.3	16	167.17	5.58	161.59	172.75	16	1.11	0.05	1.06	1.16
17	63.41	6.81	56.6	70.22	17	170.28	6.81	163.47	177.09	17	1.18	0.05	1.13	1.23

Peso					Estatura					BMD				
Mujeres					Mujeres					Mujeres				
Age	X	DE	X-DE	X+DE	Age	X	DE	X-DE	X+DE	Age	X	DE	X-DE	X+DE
12	50.56	10.45	40.11	61.01	12	154.86	4.94	149.92	159.8	12	0.85	0.04	0.81	0.89
13	52.01	9.83	42.18	61.84	13	153.32	9.09	144.23	162.41	13	0.88	0.04	0.84	0.92
14	52.09	9.41	42.68	61.5	14	154.86	7.84	147.02	162.7	14	0.90	0.07	0.83	0.97
15	55.50	8.80	46.7	64.3	15	156.82	7.14	149.68	163.96	15	0.92	0.08	0.84	1.00
16	54.25	7.64	46.61	61.89	16	157.29	7.50	149.79	164.79	16	0.93	0.10	0.83	1.03
17	58.83	10.71	48.12	69.54	17	157.85	8.79	149.06	166.64	17	0.96	0.12	0.84	1.08

Figura Nº 18: Puntos de corte.

Fuente Elaboración propia.

Estos puntos de cortes se obtuvieron de los datos de los estudiantes, con ayuda de unas funciones de Excel (promedio y desvesta) se pudo obtener la desviación estándar y así poder clasificarlos por edad los cuales se usarán en las funciones de Matlab.

Después de haber obtenido los puntos de corte, se pudo manejar los datos de entrada con la recomendación del experto, la cual garantiza que las particiones sean fuertes, se pueden hacer de la siguiente forma que se muestran en las siguientes figuras.

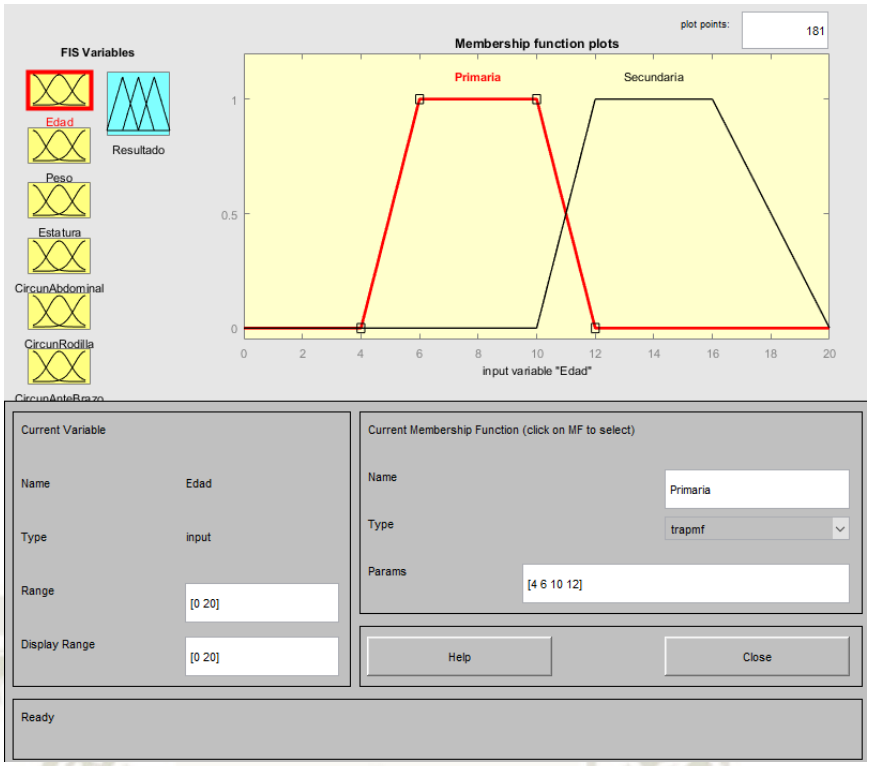


Figura Nº 19: Ejemplo de dato de entrada de edad.

Fuente Matlab.

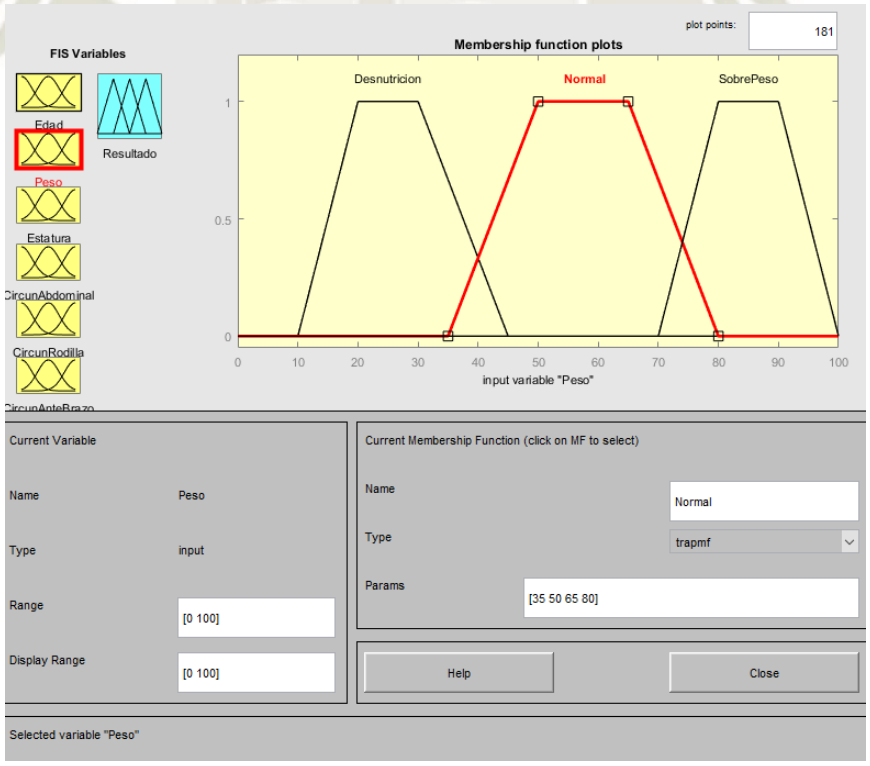


Figura Nº 20: Ejemplo de dato de entrada de Peso.

Fuente Matlab.

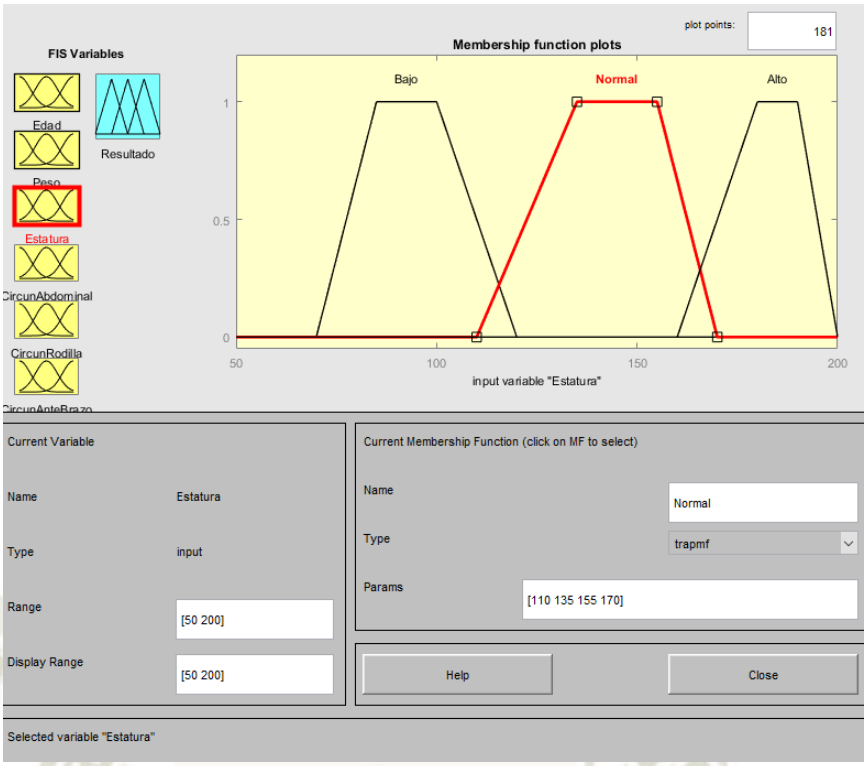


Figura Nº 21: Ejemplo de dato de entrada de Estatura

Fuente Matlab.

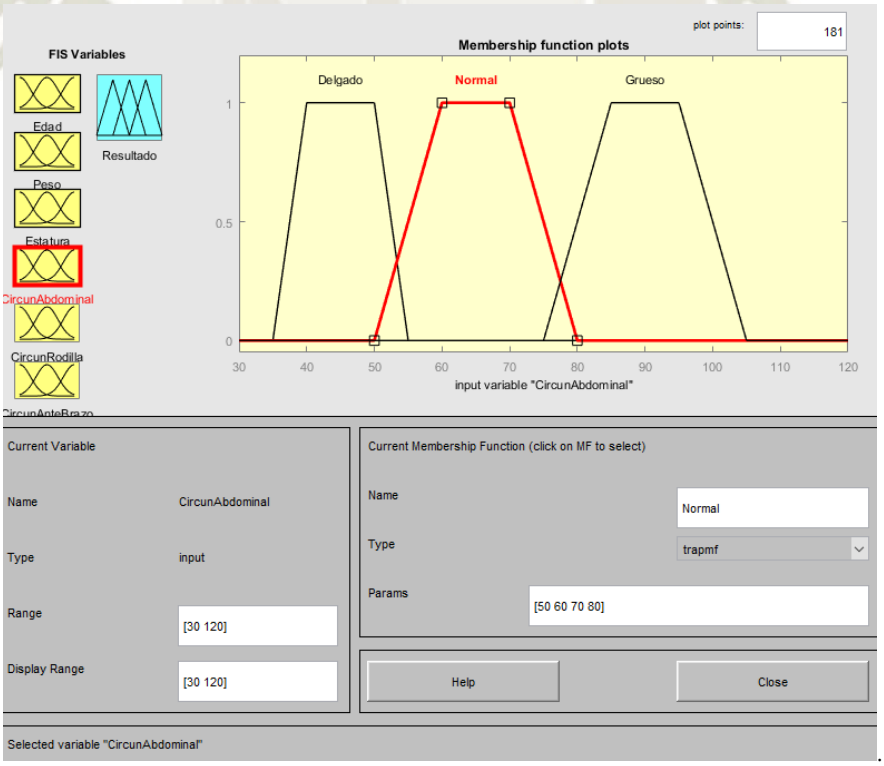


Figura Nº 22: Ejemplo de dato de entrada de Circunferencia Abdominal.

Fuente Matlab.

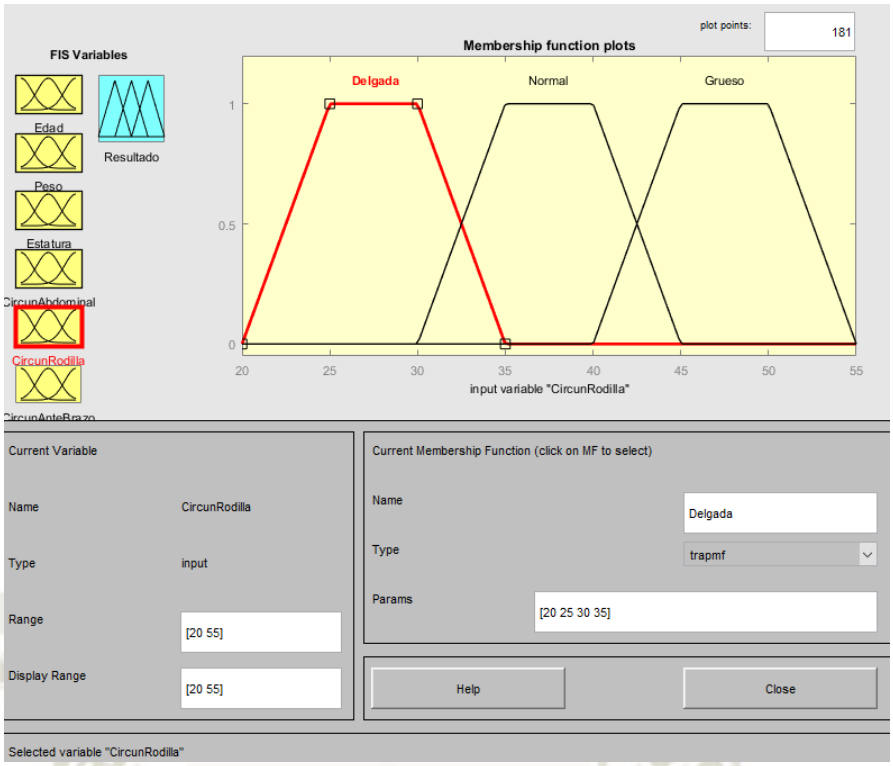


Figura Nº 23: Ejemplo de dato de entrada de Circunferencia de Rodilla.

Fuente Matlab.

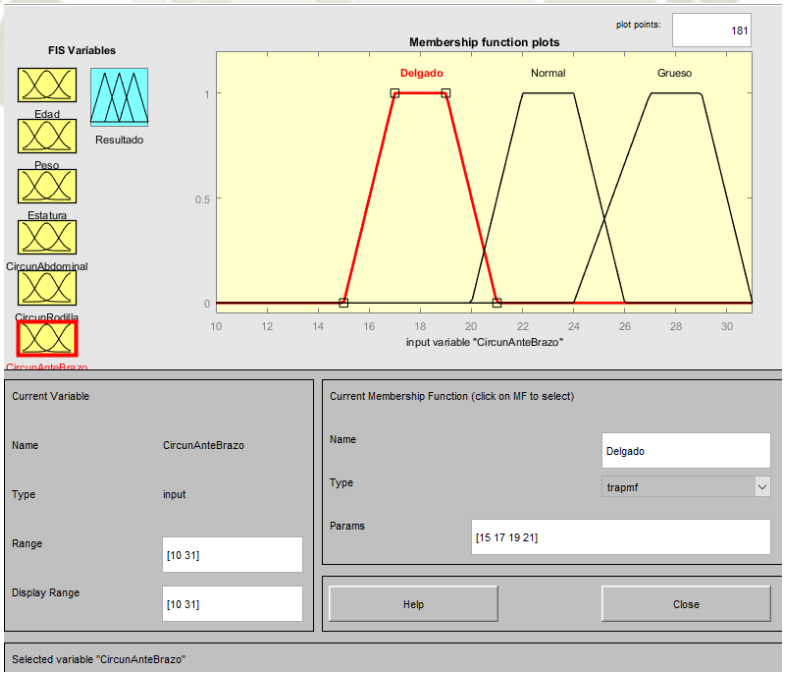


Figura Nº 24: Ejemplo de dato de entrada de Circunferencia de Antebrazo Derecho.

Fuente Matlab.

4.2.3. DATOS DE SALIDA

La salida de datos tiene el nombre de resultado, en la que se muestra 3 trapecios, los cuales son: osteoporosis, osteopenia y normal.

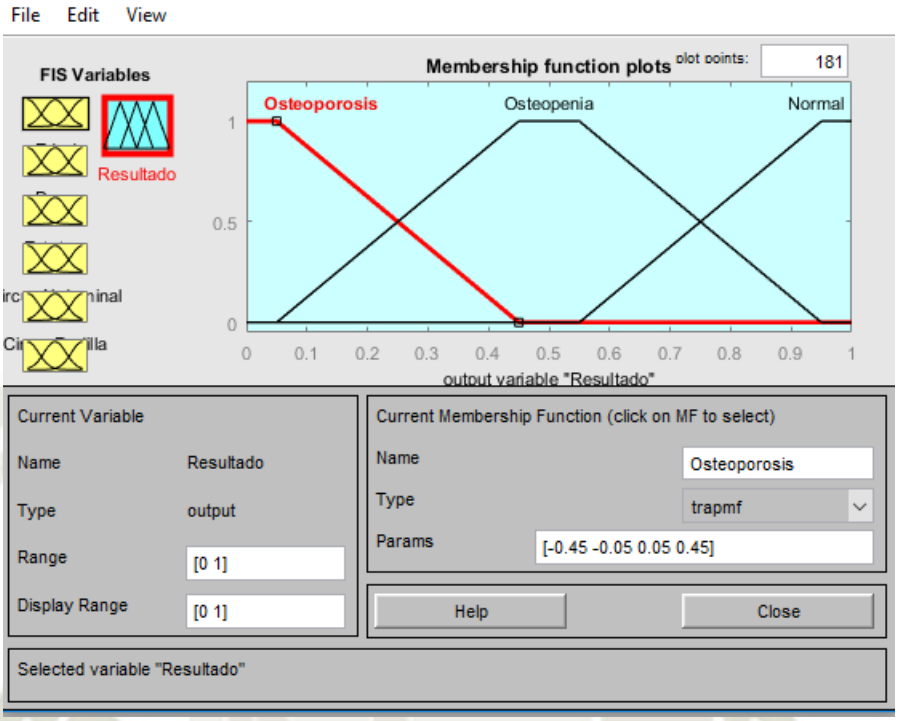


Figura Nº 25: Salida de datos.

Fuente Matlab.

4.2.4. SURFACE

El grafico Surface refleja el intervalo completo de las bases del conjunto de salida en todo el conjunto del intervalo de entrada.

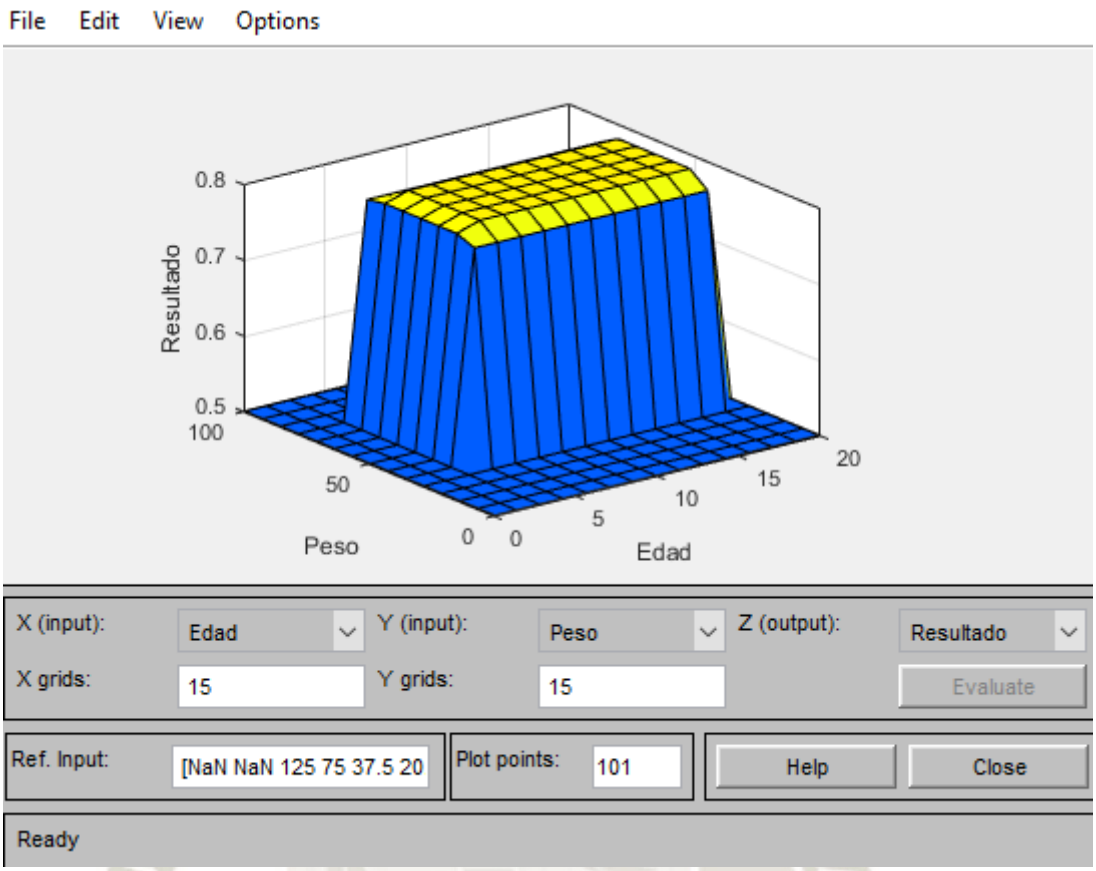


Figura Nº 26: Grafico Surface.

Fuente Matlab.

En la figura anterior, se muestra en el grafico los datos del resultado, peso y edad, se muestra la relación que se encuentra en el conjunto de datos de entrada con el conjunto de datos de salida.

4.2.5. REGLAS

Las variables biológicas, se diferencias por sus características no lineales que están mejor representadas por un intervalo que por un proceso binario, esto hace que la lógica fuzzy sea un método potencial y útil. El modelo presenta un enfoque prometedor para predecir enfermedades óseas.

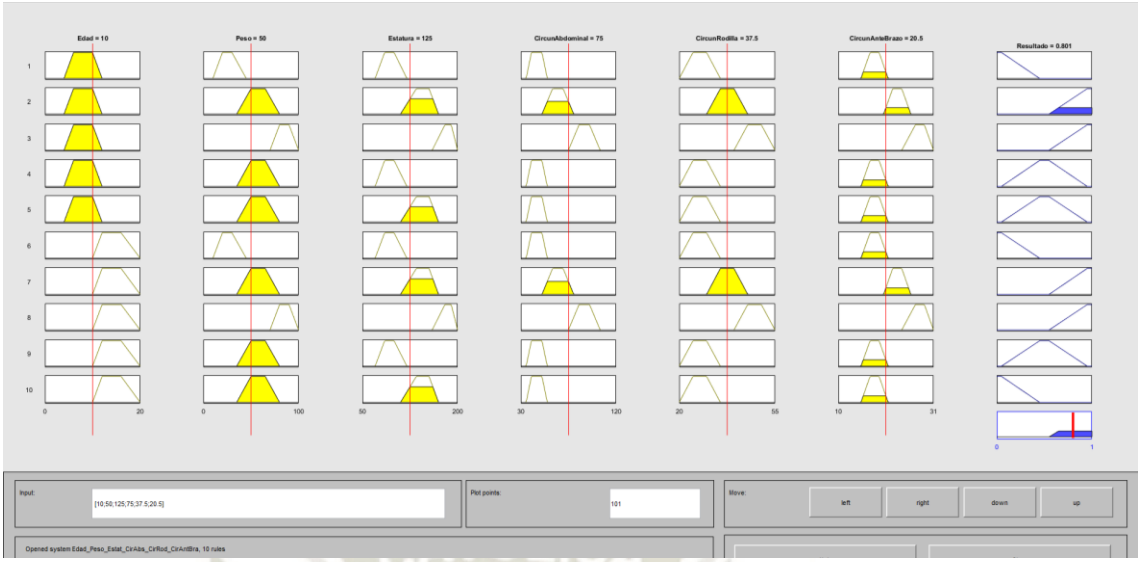


Figura Nº 27: Reglas en Matlab.

Fuente Matlab.

4.2.6. FUNCIONES

Las funciones fueron utilizadas para poder fuzzificar los datos (peso, estatura y densidad ósea), en los cuales se iban clasificando según su edad y su género. Estas funciones como resultado nos mostraban una gráfica trapezoidal en la cual se podía ver los puntos de corte y obtener los datos fuzzificados entre 0 y 1.

Se explicará la función de peso de niñas de 12 años, cual comenzara en la línea 6 como se mostrará en la figura, "X" viene hacer el rango que se mostrara en la gráfica trapezoidal y de cuanto en cuanto va a ir incrementándose, es decir, tiene un inicio en 35 hasta 70, con un aumento de 0.5.

En la línea 8, son los puntos de corte que se dará a la función trapezoidal, en este caso se necesitan 4 puntos de corte como se muestra en la imagen.

En la línea 9, se especifica el tipo de función de pertenencia, en este caso se utilizó la "trapmf" que calcula los valores de pertenencia difusa.

En la línea 10, se evalúa el grado de pertenecía del valor del eje "Y".

En la línea 12, la función plot adopta las diferentes formas según los argumentos de entrada, si se especifican dos vectores como argumentos, plot (X, Y) produce un gráfico de "Y" contra "X".

Por último, en la línea 16, se ingresan los datos de los niños y/o adolescentes que se van a evaluar para fuzzificar.

A continuación, se mostrará un ejemplo de las funciones usadas para obtener los datos difusos.

```
clc
clear all
close all
 % -------------------------
 %Peso niñas: 12 años
x=35:0.5:70;

mfparams = [40.11 50.56 61.01 70.04];
mftype = 'trapmf';
y=evalmf(x,mfparams,mftype);   % evalmf : evalua el grado de pertenencia (valor del eje y)

plot(x,y)
xlabel('Peso')

%%  Calculo de datos  de salida (grado_de_pertencia)
datos = [35
38.1
43.8
48
52
62
75
92.4
]
%grado_de_pertenencia = evalmf(datos, mfparams, mftype)
grado_de_pertenencia = fprintf(1,'%2.1f\n',evalmf(datos, mfparams, mftype))
```

Figura Nº 28: Función de peso de niñas de 12 años.

Fuente Matlab.

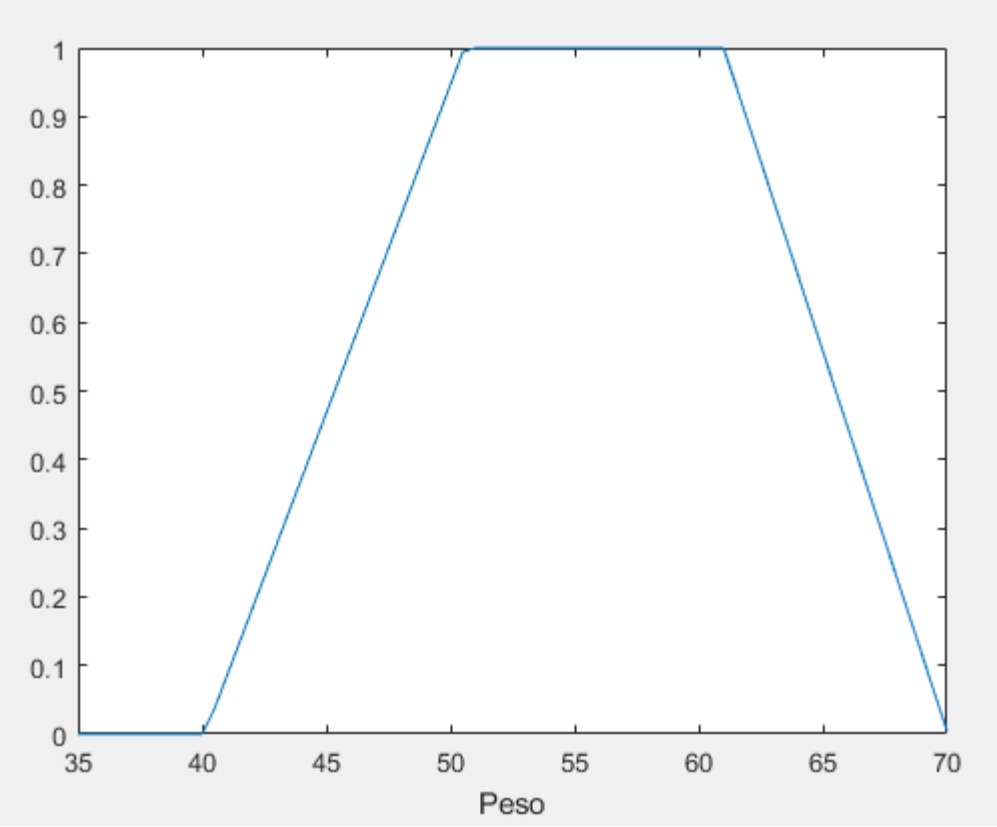


Figura Nº 29: Grafica Trapezoidal obtenida de la función peso de niñas de 12 años.

Fuente Matlab.

```
clc
clear all
close all
 % -------------------------
 %Peso segun Estatura 17 años: 1.50-1.60-1.70-1.80
x=1.40:0.1:1.80;

mfparams = [1.50 1.58 1.67 1.76];
mftype = 'trapmf';
y=evalmf(x,mfparams,mftype);   % evalmf : evalua el grado de pertenencia (valor del eje y)

plot(x,y)
ylabel('función de pertenencia')
xlabel('Estatura')
%xlabel('trimf, P=[1.50-1.60-1.70-1.80]')

%%  Calculo de datos  de salida (grado_de_pertencia)
datos = [1.48
1.49
1.50
1.51
1.52
1.53
1.54
```

Figura Nº 30: Función de Estatura de adolescentes mujeres de 17 años.

Fuente Matlab.

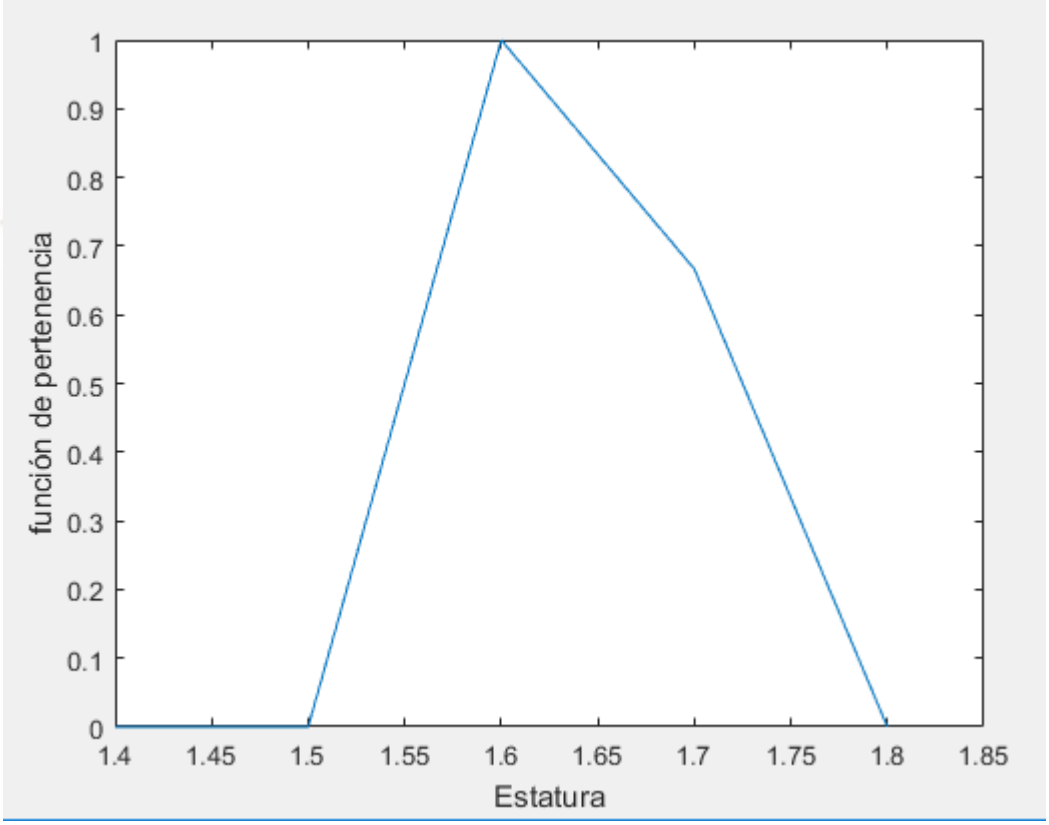


Figura Nº 31: Grafico Trapezoidal obtenida de la función estatura de adolescentes mujeres de 17 años.

Fuente Matlab.

```
clear all
close all
 % ------------------------
 %BMD niñas: 15 años
x=0.8:0.1:1.10;

mfparams = [0.84 0.92 1.00 1.08];
mftype = 'trapmf';
y=evalmf(x,mfparams,mftype);   % evalmf : evalua el grado de pertenencia

plot(x,y)
xlabel('BMD')

%%  Calculo de datos  de salida (grado_de_pertencia)
datos = [0.85
0.86
0.87
0.88
0.89
0.90
```

Figura Nº 32: Función de BMD de adolescentes mujeres de 15 años.

Fuente Matlab.

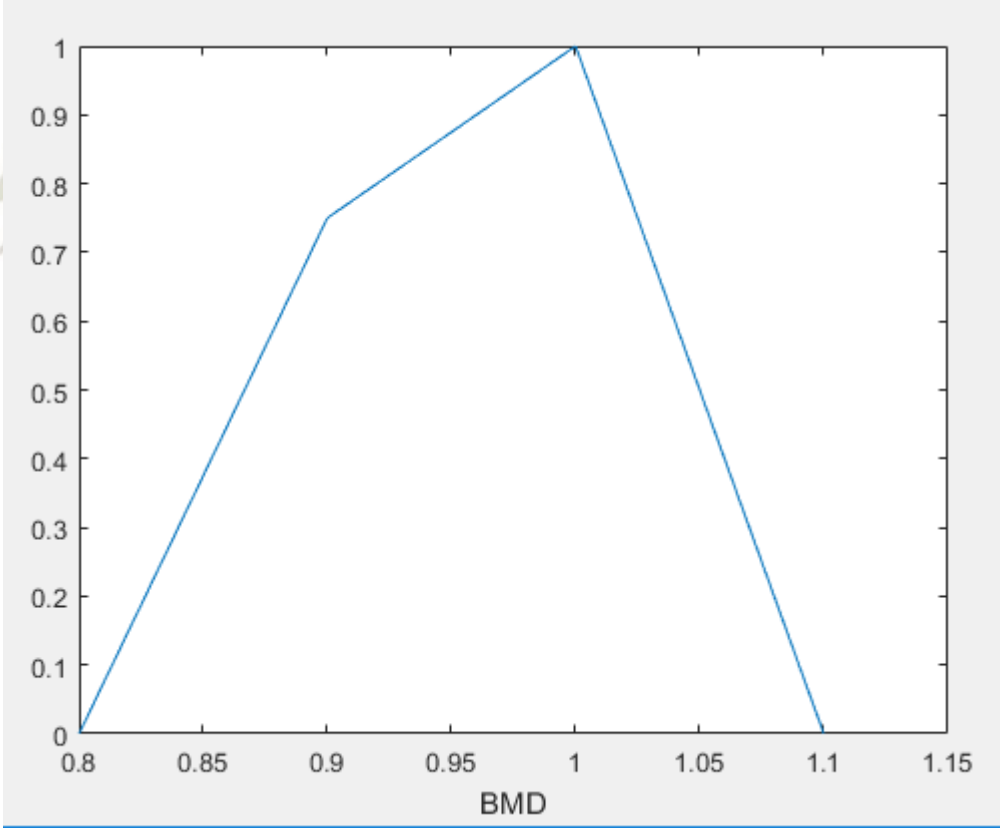


Figura Nº 33: Grafico trapezoidal de la función de BMD de adolescentes mujeres de 15 años.

Fuente Matlab.

4.3. APLICACION DE ESCRITORIO DE APOYO

La aplicación de escritorio fue dada por el Dr. José Alfredo Sulla Torres, el cual tenía la finalidad de obtener el otro valor difuso, se hicieron ciertas modificaciones al programa, adaptando el programa a nuestras necesidades la cual tiene por finalidad obtener el BMD difuso y que nos muestren las reglas difusas y en cierto caso hacer que estas reglas difusas sean lo más interpretables posible, la finalidad de este proyecto es la interpretación de las reglas difusas que se obtendrán, siguiendo el modelo de datos del árbol de injerto. Se ha extraído todos los resultados de todo este conjunto de datos y se ha obtenido una muestra la cual se presenta como anexos.

4.4. INTERPRETACION DE LAS REGLAS DIFUSAS

Para poder interpretar las reglas difusas se tiene en cuenta los requerimientos contradictorios que son precisión e interpretabilidad.

- Precisión: Es la capacidad de representar a pie de letra el sistema real, esta debería ser mejor ya que abría una similitud mayor entre el modelo difuso y el sistema real, para obtener la precisión en modelos de regresión o clasificación, son los porcentajes de patrones clasificados correctamente a partir del conjunto de datos.
- Interpretabilidad: Es la capacidad de expresar el comportamiento del sistema real de una forma comprensible, es subjetivo, es decir, depende de la persona que está realizando la evaluación. Tiene varios factores relacionados, principalmente del modelo, el número de variables de entrada, el número de reglas difusas, el número de términos lingüísticos, la forma de los conjuntos difusos. No existe una medida estándar para poder evaluar qué tan buena es la interpretabilidad.

Se han encontrado muchos estudios sobre la interpretabilidad y la precisión, para muchos investigadores están de acuerdo que la interpretabilidad involucre los siguientes aspectos: La cantidad de reglas debe ser lo más pequeño posible, las premisas de reglas se entienden fácilmente en términos de estructura y contienen solo pocas variables de entrada, términos lingüísticos que son intuitivamente compresibles.

Teniendo en cuenta los puntos anteriores de los investigadores, analizaremos la interpretabilidad de las reglas obtenidas.

```
IF Estatura IS EstaBajo AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaBajo AND Peso IS PesoNormal THEN Normal (0,95)
IF Estatura IS EstaBajo AND Peso IS PesoAlto THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdBajo THEN Normal (0,96)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,93)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Adolescente THEN Normal (0,88)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,89)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,70)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoNormal THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,87)
IF Estatura IS EstaAlto AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaAlto AND Peso IS PesoNormal THEN Normal (0,92)
IF Estatura IS EstaAlto AND Peso IS PesoAlto THEN Normal (1,00)
```

Figura Nº 34: Reglas obtenidas.

Fuente Matlab.

4.5. ANÁLISIS DE LA INTERPRETABILIDAD

Según Luis Anderos Souza, Feagri Carvalho Barros, Alonso José M. Magdalena, Luis Gonzales – Rodríguez Gil:

Tabla N° 7 Cuadro de Análisis de la Interpretabilidad.

Interpretabilidad de la regla difusa			
Descripción (legibilidad de la estructura del sistema)			Explicación (Comprensión del sistema)
Nivel Alto	Base de reglas	-Numero de reglas - Longitud de reglas	Mecanismos de inferencia y agregación de reglas. Operador de conjunción
	Reglas borrosas linguisticas	- Número de entradas por regla - Consecuencias de alto orden	Operador de disyuncion Defuzzificación (interpretación de salida)
	Proposiciones linguisticas	- Longitud de la descripcion - Correlación de atributos - Terminos lingüisticos elementales - Términos lingüísticos compuestos - Modificadores lingüísticos	Número de reglas disparadas simultáneamente. Consistencia Integridad Regla estructura de variables (Mamdani) Número de variables (entradas * / salidas
	Particiones lingüisticas	- Semántica global (conjuntos difusos compartidos) - Términos lingüísticos y conjuntos difusos de significados coincidentes	Interpretación de reglas (conjuntiva / implicativa) Negación
	Particiones Difusas	- Un número justificable de conjuntos difusos - Ordenando - Distinguibilidad - Cobertura - Complementariedad - Granulación uniforme	Antónimos
Nivel Bajo	Conjuntos difusos	- Normalización - Convexidad	

Fuente Elaboración propia

Comenzaremos por la parte izquierda de la tabla anterior, como se puede observar en la tabla anterior, el nivel más bajo corresponde de lo menos interpretable (mayor precisión) a lo más interpretable (menor precisión), mientras vamos subiendo de nivel, se va teniendo en cuenta el uso de variables lingüísticas que esto favorece la legibilidad; debido a la limitada memoria humana a corto plazo y la capacidad de cálculo es mejor trabajar con un pequeño número de términos.

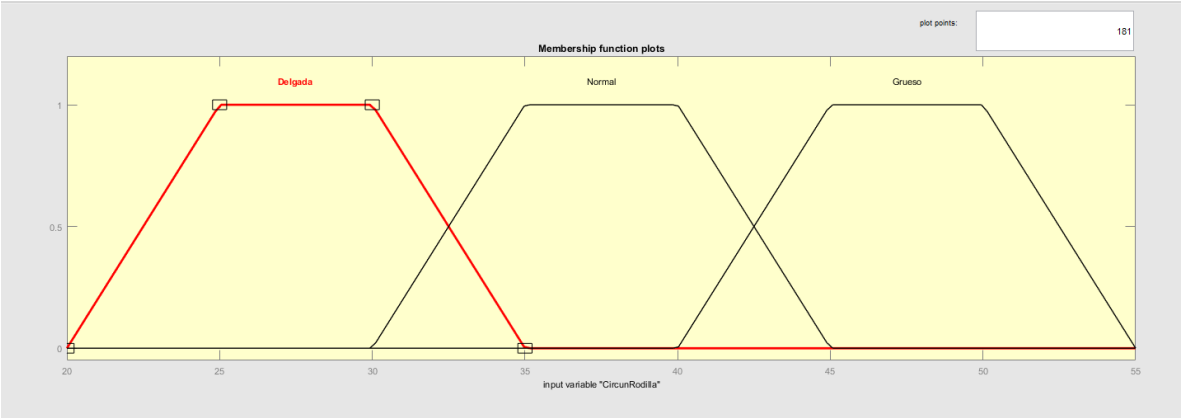


Figura Nº 35: Ejemplo de Particiones.

Fuente Matlab.

Como se muestra en la figura N° 35, las particiones las hemos definido mediante percentiles, con lo cual llegamos a obtener particiones de acorde a los datos.

El uso de particiones fuertes genera particiones borrosas interpretables en el sentido de mantener estructuras claras y transparentes, para tener una partición totalmente significativa, los términos lingüísticos se deben seleccionar de acuerdo con el contexto del problema.

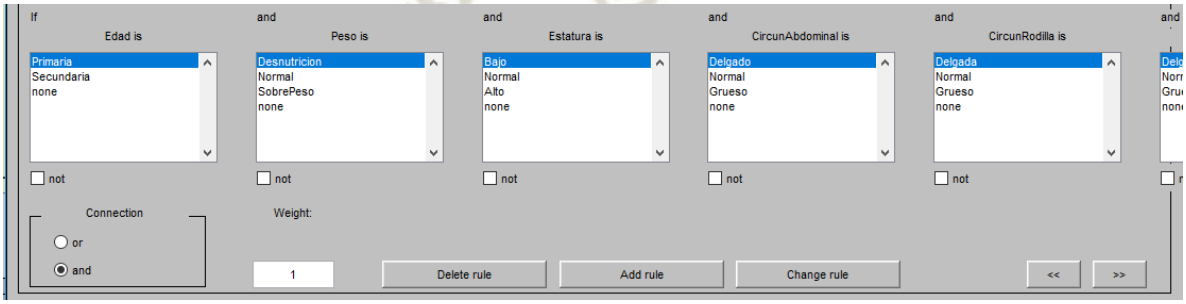


Figura Nº 36: Definiendo el conjunto de términos lingüísticos.

Fuente Matlab.

Después de haber definido el conjunto de términos lingüísticos se puede utilizar para expresar proposiciones lingüísticas, luego de tener varias proposiciones lingüísticas se combinan para formar reglas difusas. Se debe analizar cada regla de manera individual y luego analizar varias reglas para así obtener un nivel de abstracción más alto, solo si las reglas usan los mismos términos lingüísticos, también tener en cuenta que mientras más grande sea el sistema, se complica más la tarea del análisis.

Para finalizar, el cumplimiento de todas las restricciones enumeradas en la parte izquierda de la tabla N° 7 garantiza la interpretabilidad desde el punto de vista estructural, pero en la practicas es difícil cumplir con todas las restricciones, ya que generaría un conjunto de reglas con muy poca precisión.

Ahora explicaremos la parte derecha de la tabla, la explicación, para comprender la descripción lingüística es una tarea muy difícil, porque, se necesita entrar en detalle con respecto a la implementación del mecanismo de inferencia.

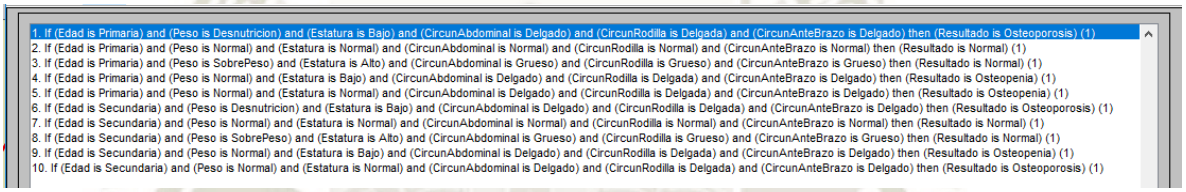


Figura Nº 37: Reglas difusas.

Fuente Matlab.

En realidad, existen una gran diversidad de reglas difusas (reglas graduales, de certeza, de posibilidad, etc.) y todas estas tienen comportamientos de inferencia especifico, por lo tanto, se debe seleccionar el tipo correcto de reglas ya que influye en la precisión final y en la semántica de la regla.

Dentro del ámbito de comprensibilidad, se debe considerar la interpretación correcta de las reglas, en algunas aplicaciones es mejor usar reglas implícitas graduales de las cuales se pueden obtener mejor interpretación que reglas sencillas e intuitivas.

El nivel de inferencia incluye los operadores difusos para la conjunción, disyunción, agregación y defuzzificación, se debe tener en cuenta que toda regla debe ser coherente y cubrir la mayoría de las situaciones posibles.

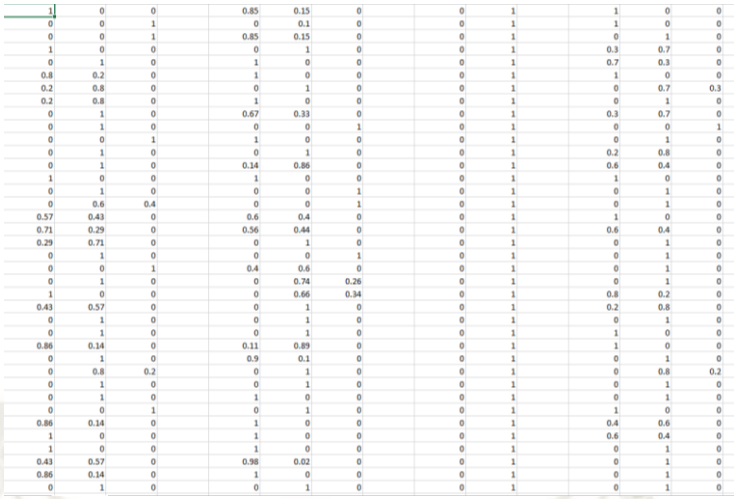


Figura Nº 38: Datos difusos, asegurando la integridad.

Fuente Matlab.

La integridad de las reglas depende del conjunto de datos, mientras sea una cantidad de datos mayor con casos variados, mayor será el número de situaciones gestionadas. Los requerimientos de integridad varían según su aplicación.

La comprensibilidad depende a gran medida del número de reglas, mientras sea menor el número, mayor será la comprensibilidad, un caso práctico, es un sistema que dispara 10 reglas es más comprensible que uno que dispara 100 reglas.

Con respecto a la estructura de reglas, las reglas de Mamdani son generalmente la más interpretables, porque son reglas lingüísticas de la siguiente forma:

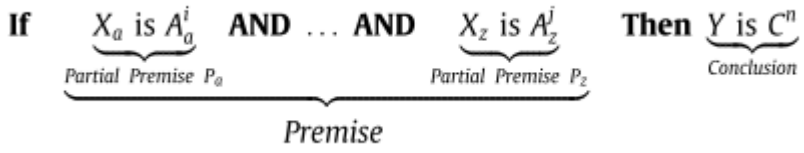


Figura Nº 39: Estructura de reglas de Mamdani.

Fuente. (Alonso J. M., 2009)

Como se puede observar, las premisas de las reglas están formadas por tuplas (variable de entrada y termino lingüístico) donde X_a es el nombre de la variable de entrada a, mientras que A_a^i la etiqueta i de dicha variable. Además, el uso de términos lingüísticos produce reglas más compactas.

Una vez que se analizaron los principales aspectos relacionados con la legibilidad y comprensibilidad, se va a la evaluación de la interpretabilidad. Luego de haber identificado los elementos involucrados, tratar de combinarlos y obtener un buen índice, la mayoría de los índices se centran en la legibilidad de las partes difusas.

Para finalizar, para obtener reglas interpretables se tiene que seguir los indicadores de la tabla N ° 7, desde el nivel bajo hacia el nivel alto, tratando de cumplir con lo que se menciona en cada sección, para así poder no solo tener resultados interpretables sino también legibles y compresivos como se muestra en la figura N ° 34, sin dejar de lado la precisión de las reglas.

CONCLUSIONES

1. Para la preparación de datos se utilizó la eliminación de vacío. Los datos de los escolares se analizaron y prepararon para el correcto uso de la extracción del conocimiento usando técnicas de minería de datos.

2. Se hallo el mejor algoritmo de aprendizaje automático de los 5 analizados el cual es el Árbol de injerto, teniendo una ligera ventaja sobre los demás algoritmos que fueron comparados, con un 92.42% de instancias correctamente clasificadas y utilizando la mayoría de los atributos al generar el árbol.

3. Se modifico correctamente la herramienta de apoyo en base a las necesidades del proyecto actual, la cual, se inserta un conjunto de datos, obteniendo las reglas difusas que serán interpretadas para obtener un mejor resultado.

4. El análisis de la interpretación de las reglas difusas beneficiara a futuro para tener en cuenta el uso de variables, términos lingüísticos, particiones difusas, operadores, premisas difusas entre otros aspectos, para tener un mejor resultado en la interpretación de reglas, no solo es ver la interpretabilidad y la precisión, también es necesario tener en cuenta la legibilidad y la comprensión de las reglas difusas.

RECOMENDACIONES

1. Se recomienda analizar con una cantidad de datos mayor, para poder obtener más reglas difusas que las obtenidas en el proyecto.
2. El proyecto fue realizado en base a la de la densidad de la salud ósea, analizando la interpretación de las reglas difusas y teniendo en cuenta los percentiles, sin embargo, se puede utilizar para otras áreas de interés médico como traumatología, reumatología, nutrición y pediatría.
3. Los resultados obtenidos en este proyecto sugieren hacer otro tipo de investigación con nuevos algoritmos que vayan saliendo en un futuro y puedan llegar a ser mejores de los que se presentaron actualmente.
4. En base al estudio realizado, se sugiere hacer otro tipo de análisis con otro percentiles de datos que traten sobre la densidad ósea normal y demasiada densidad ósea.

REFERENCIAS BIBLIOGRÁFICAS

Alonso, J. M. (2008). Equilibrio entre Interpretabilidad y Precision en Sistemas Basados en Reglas Difusas : Nuevos retos. *XIV Spanish conference for Fuzzy Logic and Technology (ESTYLF)*, 501-508.

Alonso, J. M. (2009). An interpretability-guided modeling process for learning comprehensible fuzzy rule-based classifiers. *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, 432-437.

Bhandari, A., Gupta, A., & Das, D. (2015). Improvised apriori algorithm using frequent pattern tree for real time applications in data mining. *Procedia Computer Science*, 644-651.

Cannone, R. C. (2009). A study on interpretability conditions for fuzzy rule-based classifiers. *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, 438-443.

Cannone, R. C. (2009). A study on interpretability conditions for fuzzy rule-based classifiers. *ISDA 2009 - 9th International Conference on Intelligent Systems Design and Applications*, 438-443.

Carlos Denegri, E. D. (2006). Introducción al razonamiento aproximado: lógica difusa. *Revista Argentina de Medicina Respiratoria*, 126-136.

Colomb, R. M., & Chung, C. Y. (2000). Very Fast Decision Table Execution of Propositional Expert Systems. 671-676.

Encarnación, D. A. (2015). Estudio del Rendimiento Académico Aplicando.

Freitas, A. A. (2010). Freitas, Alex A. 76-86.

G. Martínez, J. M. (2011). Árboles De Decisiones En El Diagnóstico De Enfermedades Cardiovasculares. *Scientia Et Technica*, 104-109.

Gacto, M. J. (2011). Interpretability of linguistic fuzzy rule-based systems: An overview of interpretability measures. *Information Sciences*, 4340-4360.

Godora, S., & Verma, A. (2013). Analysis of Various Clustering Algorithms. 186-189.

Gómez-Campos, A., Arruda, U. A., & Cossio-Bolaños. (2017). Proposed equations and reference values for calculating bone health in children and adolescent based on age and sex.

Hayashi, Y. N. (2015). Use of a Recursive-Rule eXtraction algorithm with J48graft to achieve highly accurate and concise rule extraction from a large breast cancer dataset. *Informatics in Medicine Unlocked*, 9-16.

Isabel Rey, M. G. (2013). Selection of rules by orthogonal transformations and genetic algorithms to improve the interpretability in fuzzy rule based systems. *IEEE International Conference on Fuzzy Systems*.

Jaqueline, V. C. (2015). Estudio del Rendimiento Académico Aplicando Técnicas de Minería de Datos. Loja, Ecuador.

Jiawei Han, M. K. (2006). *Data Mining Concepts and Techniques.*

José Antonio Garcia Bermúdez, Á. M. (2010). Análisis para predicción de ventas utilizando minería de datos en almacenes de ventas de grandes superficies.

Lesot, M.-j. M.-m.-j.-m. (2016). Interpretability of fuzzy linguistic summaries. *Fuzzy Sets and Systems*, 307-317.

Manish Verma, M. S. (2012). A Comparative Study of Various Clustering Algorithms in Data Mining. *International Journal of Engineering Research and Applications*, 1379-1384.

Mencar, C. C. (2011). Interpretability assessment of fuzzy knowledge bases: A cointension based approach. *International Journal of Approximate Reasoning*, 501-518.

Mencar, C. C. (2011). Interpretability assessment of fuzzy knowledge bases: A cointension based approach. *International Journal of Approximate Reasoning*, 501-518.

Nilashi, M. I. (2017). An analytical method for diseases prediction using machine learning techniques. *Computers and Chemical Engineering*, 212-223.

Ramkumar, S. M. (2018). Detection of osteoporosis and osteopenia using bone densitometer - Simulation study. *Materials Today: Proceedings*, 1024-1036.

Riquelme, J. C. (2006). Minería de datos: Conceptos y tendencias. *Inteligencia Artificial*, 11-18.

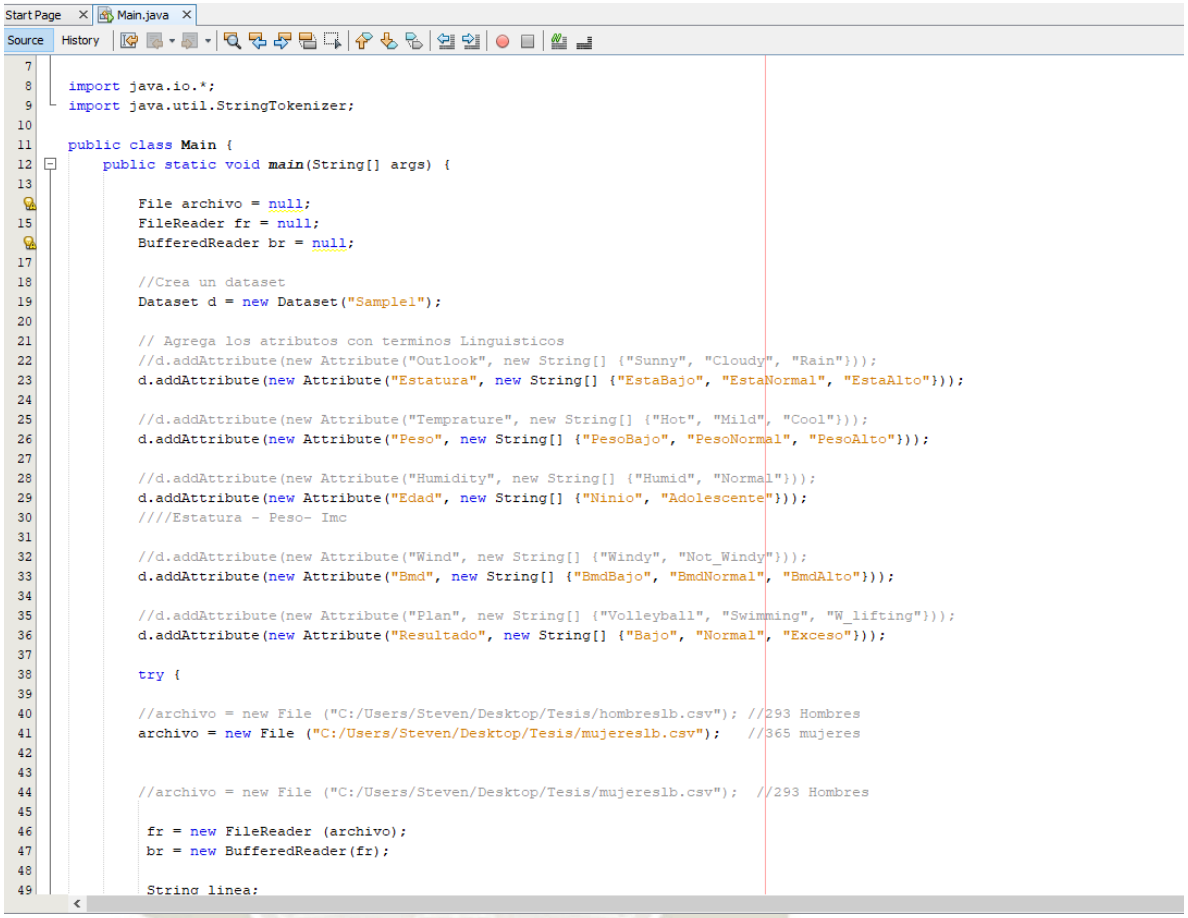
Vanesa Berlanga Silvente, M. J. (2013). Cómo aplicar árboles de decisión en SPSS. *REIRE. Revista d'Innovació i Recerca en Educació*, 62-79.

Wisaeng, K. (2013). A Comparison of Different Classification Techniques for Bank Direct Marketing. *International Journal of Soft Computing and Engineering (IJSCE)*, 116-119.

Yadav, S., & Brijesh, P. S. (2012). Data mining applications: A comparative study for predicting student's performance. *International Journal of Innovative Technology & Creative Engineering*, 13-19.

ANEXOS

Anexo A. Muestra los cambios realizados a la aplicación desarrollada en NetBeans.



```
import java.io.*;
import java.util.StringTokenizer;

public class Main {
    public static void main(String[] args) {

        File archivo = null;
        FileReader fr = null;
        BufferedReader br = null;

        //Crea un dataset
        Dataset d = new Dataset("Sample1");

        // Agrega los atributos con terminos Linguisticos
        //d.addAttribute(new Attribute("Outlook", new String[] {"Sunny", "Cloudy", "Rain"}));
        d.addAttribute(new Attribute("Estatura", new String[] {"EstaBajo", "EstaNormal", "EstaAlto"}));

        //d.addAttribute(new Attribute("Temprature", new String[] {"Hot", "Mild", "Cool"}));
        d.addAttribute(new Attribute("Peso", new String[] {"PesoBajo", "PesoNormal", "PesoAlto"}));

        //d.addAttribute(new Attribute("Humidity", new String[] {"Humid", "Normal"}));
        d.addAttribute(new Attribute("Edad", new String[] {"Ninio", "Adolescente"}));
        ////Estatura - Peso- Imc

        //d.addAttribute(new Attribute("Wind", new String[] {"Windy", "Not_Windy"}));
        d.addAttribute(new Attribute("Bmd", new String[] {"BmdBajo", "BmdNormal", "BmdAlto"}));

        //d.addAttribute(new Attribute("Plan", new String[] {"Volleyball", "Swimming", "W_lifting"}));
        d.addAttribute(new Attribute("Resultado", new String[] {"Bajo", "Normal", "Exceso"}));

        try {

        //archivo = new File ("C:/Users/Steven/Desktop/Tesis/hombreslb.csv"); //293 Hombres
        archivo = new File ("C:/Users/Steven/Desktop/Tesis/mujereslb.csv");   //365 mujeres


        //archivo = new File ("C:/Users/Steven/Desktop/Tesis/mujereslb.csv");  //293 Hombres

         fr = new FileReader (archivo);
         br = new BufferedReader(fr);

         String linea;
```

```
        try {

        //archivo = new File ("C:/Users/Steven/Desktop/Tesis/hombreslb.csv"); //293 Hombres
        archivo = new File ("C:/Users/Steven/Desktop/Tesis/mujereslb.csv");   //365 mujeres


        //archivo = new File ("C:/Users/Steven/Desktop/Tesis/mujereslb.csv");  //293 Hombres

         fr = new FileReader (archivo);
         br = new BufferedReader(fr);

         String linea;
         double [] valores = new double[14];

         //double [] valores = new double[12];
         int i;
         while((linea=br.readLine())!=null)
         {
            //System.out.println(linea);
            StringTokenizer st = new StringTokenizer(linea, ";");
            i=0;
            while(st.hasMoreTokens()) {
                //System.out.println(st.nextToken());
                valores[i] = Double.parseDouble(st.nextToken());
                //System.out.println("valores["+i+"]" + valores[i] );
                i++;
            }

            d.addRow(new Row(new Object[]{"Dummy", "Dummy", "Dummy", "Dummy", "Dummy"}, new double[][] {{ valores[0],valores[1],valores[2]},
                {valores[3],valores[4],valores[5]}, {valores[6],valores[7]}, {valores[8],valores[9],valores[10]}, {valores[11],valores[12],valores[13] }}))
```

```
84
85      //public FuzzyDecisionTree(double truthLevel, double significantLevel)
86      //FuzzyDecisionTree fdt = new FuzzyDecisionTree(0.82, 0.0); //Hombres (0.82)
87      FuzzyDecisionTree fdt = new FuzzyDecisionTree(0.96, 0.0); //Mujeres
88
89      fdt.setSignificantLevel(0.0);
90      System.out.println("-------------------------------------------------------------------------");
91      System.out.println("Con alfa (nivel significante) = 0.0");
92
93      System.out.println(String.format("G(Estatura) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura")));
94      System.out.println(String.format("G(Peso) = %.2f", fdt.getAmbiguity(d,"Resultado", "Peso")));
95      System.out.println(String.format("G(Edad) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad")));
96      System.out.println(String.format("G(Bmd) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd")));
97
98      System.out.println(String.format("G(BmdAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", new String[] {"Bmd", "BmdAlto"})));
99      System.out.println(String.format("G(BmdNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", new String[] {"Bmd", "BmdNormal"})));
100     System.out.println(String.format("G(BmdBajo) = %.2f", fdt.getAmbiguity(d, "Resultado", new String[] {"Bmd", "BmdBajo"})));
101
102     System.out.println(String.format("G(Estatura | BmdAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Bmd", "BmdAlto"})));
103     System.out.println(String.format("G(Edad | BmdAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Bmd", "BmdAlto"})));
104     System.out.println(String.format("G(Peso | BmdAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Peso" , new String[] {"Bmd", "BmdAlto"})));
105
106     System.out.println(String.format("G(Estatura | BmdNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Bmd", "BmdNormal"})
107     System.out.println(String.format("G(Edad | BmdNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Bmd", "BmdNormal"})));
108     System.out.println(String.format("G(Peso | BmdNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Peso" , new String[] {"Bmd", "BmdNormal"})));
109
110
111     System.out.println(String.format("G(Estatura | PesoAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Peso", "PesoAlto"}))
112     System.out.println(String.format("G(Edad | PesoAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Peso", "PesoAlto"})));
113     System.out.println(String.format("G(Bmd | PesoAlto) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd" , new String[] {"Peso", "PesoAlto"})));
114
115     System.out.println(String.format("G(Estatura | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Peso", "PesoNormal
116     //System.out.println(String.format("G(Edad | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Peso", "PesoNormal"})));
117     System.out.println(String.format("G(Bmd | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd" , new String[] {"Peso", "PesoNormal"})));
118
119     TreeNode root = fdt.buildTree(d, "Resultado");
120
```

```
126
127     System.out.println("");
128     System.out.println("Simplificando reglas:");
129     for(String rule : rules) {
130         System.out.println(fdt.simplifyRule(d, rule, "Resultado"));
131     }
132     System.out.println();
133
134     ////////////////////////////////////////////////////////////////////////
135     fdt.setSignificantLevel(0.5);
136     System.out.println("-------------------------------------------------------------------------");
137     System.out.println("Con alfa (nivel significante) = 0.5");
138     //fdt.getAmbiguity(d, "Plan", "Outlook");
139     System.out.println(String.format("G(Estatura) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura")));
140     System.out.println(String.format("G(Peso) = %.2f", fdt.getAmbiguity(d,"Resultado", "Peso")));
141     System.out.println(String.format("G(Edad) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad")));
142     System.out.println(String.format("G(Bmd) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd")));
143
144     System.out.println(String.format("G(PesoBajo) = %.2f", fdt.getAmbiguity(d, "Resultado", new String[] {"Peso", "PesoBajo"})));
145
146     System.out.println(String.format("G(Estatura | PesoBajo) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Peso", "PesoBajo"}))
147     System.out.println(String.format("G(Edad | PesoBajo) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Peso", "PesoBajo"})));
148     System.out.println(String.format("G(Bmd | PesoBajo) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd" , new String[] {"Peso", "PesoBajo"})));
149
150     System.out.println(String.format("G(Estatura | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Estatura" , new String[] {"Peso", "PesoNormal
151     System.out.println(String.format("G(Edad | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Edad" , new String[] {"Peso", "PesoNormal"})));
152     System.out.println(String.format("G(Bmd | PesoNormal) = %.2f", fdt.getAmbiguity(d, "Resultado", "Bmd" , new String[] {"Peso", "PesoNormal"})));
153
154     System.out.println("");
155     System.out.println("");
156
157     root = fdt.buildTree(d, "Resultado");
158
159     fdt.printTree(root, "");
160
161     rules = fdt.generateRules(root);
162     for(String rule : rules) {
163         System.out.println(rule);
164     }
165
```

```
            root = fdt.buildTree(d, "Resultado");

            fdt.printTree(root, "");

            rules = fdt.generateRules(root);
            for(String rule : rules) {
                System.out.println(rule);
            }

            System.out.println("");
            System.out.println("Simplificando reglas:");
            for(String rule : rules) {
                System.out.println(fdt.simplifyRule(d, rule, "Resultado"));
            }

            for(int i = 0; i < rules.length;i++) {
                rules[i] = fdt.simplifyRule(d, rules[i], "Resultado");
            }

            System.out.println("Prediccion del conjunto de entrenamiento:");

            //for(int j = 0; j < 293; j++) { //hombres
            for(int j = 0; j < 365; j++) { //mujeres
               double[] cVals = fdt.classify(j, d, "Resultado", rules);
                for(int i = 0; i < cVals.length; i++) {
                    System.out.print(String.format("%.2f     ", cVals[i]));
                }
                System.out.println("");
            }
            System.out.println("");

            //System.out.println("Training set Prediction with a out information about wind:");
            System.out.println("Conjunto de entrenamiento de prediccion con una salida de information sobre edad:");
            d.getRows().clear();
            ////////////////////
        try {
            //archivo = new File ("C:/Users/Steven/Desktop/Tesis/hombreslb.csv");  //293 Hombres
            archivo = new File ("C:/Users/Steven/Desktop/Tesis/mujereslb.csv");  //364 mujeres
            //archivo = new File ("C:/Users/Steven/Desktop/Tesis/hombreslb.csv");  //293 Hombres
            fr = new FileReader (archivo);
            br = new BufferedReader(fr);

            String linea;
            double [] valores = new double[14];

            int i;
            while((linea=br.readLine())!=null)
            {
                //System.out.println(linea);
                StringTokenizer st = new StringTokenizer(linea, ";");
                i=0;
                while(st.hasMoreTokens()) {
                    valores[i] = Double.parseDouble(st.nextToken());
                    //System.out.println("valores["+i+"]" + valores[i] );
                    i++;
                }

                d.addRow(new Row(new Object[]{"Dummy", "Dummy", "Dummy", "Dummy", "Dummy"}, new double[][] {{ valores[0],valores[1],valores[2]},
                    {valores[3],valores[4],valores[5]}, {valores[6],valores[7]}, {valores[8],valores[9],valores[10]}, {valores[11],valores[12],valores[13]

            }
        }
            }
            catch(Exception e){
                e.printStackTrace();
            }finally{
             // En el finally cerramos el fichero,
                try{
                    if( null != fr ){
                        fr.close();
                    }
                }catch (Exception e2){
                e2.printStackTrace();
                }
            }
            ////////////////////

            //for(int j = 0; j < 293; j++) { //hombres
            for(int j = 0; j < 365; j++) { //mujeres
                double[] cVals = fdt.classify(j, d, "Resultado", rules);
                for(int i = 0; i < cVals.length; i++) {
                    System.out.print(String.format("%.2f     ", cVals[i]));
                }
                System.out.println("");
            }
        }
    }

```

Anexo B. Modelo de datos utilizado por el programa.

```
|Estatura|
        <EstaBajo>
                |Peso|
                        <PesoBajo>
                                        [Normal](0,91)
                        <PesoNormal>
                                        [Normal](0,95)
                        <PesoAlto>
                                        [Normal](1,00)
        <EstaNormal>
                |Bmd|
                        <BmdBajo>
                                        [Normal](0,96)
                        <BmdNormal>
                                |Peso|
                                        <PesoBajo>
                                                |Edad|
                                                        <Ninio>
                                                                        [Bajo](Infinity)
                                                        <Adolescente>
                                                                        [Normal](0,93)
                                        <PesoNormal>
                                                |Edad|
                                                        <Ninio>
                                                                        [Bajo](Infinity)
                                                        <Adolescente>
                                                                        [Normal](0,88)
                                        <PesoAlto>
                                                |Edad|
                                                        <Ninio>
                                                                        [Bajo](Infinity)
                                                        <Adolescente>
                                                                        [Normal](0,89)
                        <BmdAlto>
                                |Peso|
                                        <PesoBajo>
                                                |Edad|
                                                        <Ninio>
                                                                        [Bajo](Infinity)
                                                        <Adolescente>
                                                                        [Normal](0,70)
                                        <PesoNormal>
                                                        [Normal](1,00)
                                        <PesoAlto>
                                                |Edad|
                                                        <Ninio>
                                                                        [Bajo](Infinity)
                                                        <Adolescente>
                                                                        [Normal](0,87)
        <EstaAlto>
                |Peso|
                        <PesoBajo>
                                        [Normal](0,91)
                        <PesoNormal>
                                        [Normal](0,92)
                        <PesoAlto>
                                        [Normal](1,00)
```

Anexo C. Resultados del programa mostrando las reglas difusas que se van a interpretar.

```
IF Estatura IS EstaBajo AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaBajo AND Peso IS PesoNormal THEN Normal (0,95)
IF Estatura IS EstaBajo AND Peso IS PesoAlto THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdBajo THEN Normal (0,96)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,93)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Adolescente THEN Normal (0,88)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,89)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,70)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoNormal THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,87)
IF Estatura IS EstaAlto AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaAlto AND Peso IS PesoNormal THEN Normal (0,92)
IF Estatura IS EstaAlto AND Peso IS PesoAlto THEN Normal (1,00)

Simplificando reglas:
IF Estatura IS EstaBajo AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaBajo AND Peso IS PesoNormal THEN Normal (0,95)
IF Estatura IS EstaBajo AND Peso IS PesoAlto THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdBajo THEN Normal (0,96)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,93)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoNormal AND Edad IS Adolescente THEN Normal (0,88)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdNormal AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,89)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoBajo AND Edad IS Adolescente THEN Normal (0,70)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoNormal THEN Normal (1,00)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Ninio THEN Bajo (Infinity)
IF Estatura IS EstaNormal AND Bmd IS BmdAlto AND Peso IS PesoAlto AND Edad IS Adolescente THEN Normal (0,87)
IF Estatura IS EstaAlto AND Peso IS PesoBajo THEN Normal (0,91)
IF Estatura IS EstaAlto AND Peso IS PesoNormal THEN Normal (0,92)
IF Estatura IS EstaAlto AND Peso IS PesoAlto THEN Normal (1,00)
```