UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
Instituto de Investigación en Matemáticas Aplicadas y en Sistemas

Inteligencia Artificial

Metodología de preprocesamiento de datos estructurados para el uso de técnicas de aprendizaje de máquina.

TESIS

PARA OPTAR POR EL GRADO DE
MAESTRO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

PRESENTA:
CARLOS ANTONIO HERNÁNDEZ DÍAZ

Director de Tesis
Dr. Ángel Fernando Kuri-Morales
Instituto Tecnológico Autónomo de México.

Ciudad de México, Cd. Mx. **Noviembre 2018**

Dedicatoria

A mis padres

Agradecimientos

- A Daisy y Carlos, mis padres, que con su exigencia, comprensión, sabios consejos y su ejemplo me han conducido por el camino de la vida.
- A mi familia, fuente de motivación en mis empeños.
- A todos los amigos, que me alentaron en el camino. Tanto aquellos que desde la distancia lo hicieron, como los que apenas conociendo me apoyaron.
- A México y a su gente por abrirme las puertas.
- A Conacyt que con su apoyo económico me permitió llegar hasta aquí.
- A mi tutor, Dr. Ángel Fernando Kuri-Morales, por su cooperación y consejos.
- A maestros y profesores que a lo largo de la vida han contribuido a mi formación.
- A Dios, por haberme permitido llegar hasta aquí.

Resumen

Actualmente las empresas están siendo capaces de generar grandes volúmenes de datos, y necesitan obtener información de estos para impulsar su desarrollo. Los algoritmos de aprendizaje de máquina permiten llevar a cabo esta labor, pero en muchas ocasiones los datos no están condicionados para obtener resultados de utilidad en las organizaciones. En este contexto se plantea como objetivo proponer una metodología para el preprocesamiento de datos estructurados que permita obtener resultados coherentes al aplicar técnicas de aprendizaje de máquina, particularmente enfocadas a los algoritmos numéricos que subyacen el aprendizaje de máquina. Para esto se estudiaron las diferentes etapas que comprenden el preprocesamiento de datos, y los problemas que se presentan en cada una de estas. Habiendo realizado esta tarea se procedió a identificar los algoritmos que muestra mejores resultados resolviendo los problemas de cada una de las etapas del preprocesamiento. Teniendo en cuenta los algoritmos identificados, se definió una metodología y para su evaluación se calculó la Exactitud y el Área Bajo la Curva al aplicar un grupo de métodos seleccionados sobre los conjuntos de datos que se escogieron. Estos métodos fueron Regresión Logística, Arboles de Decisión, Redes Neuronales Artificiales, Máquinas de Soporte Vectorial, *Random Forest* y *Gradient Boosting*. Posteriormente se obtuvieron las medidas de evaluación al aplicar los métodos seleccionados usando validación cruzada de 10 particiones en los datos sin preprocesar, los datos preprocesados usando la metodología. También se tuvieron en cuenta los mejores resultados reportados en las investigaciones revisadas que usaron los conjuntos de datos seleccionados para incluirlo en la comparación y lograr evaluar la metodología. Al comparar los resultados se puede notar que de manera general

nuestros resultados mejoran sustancialmente, aunque en algunos algoritmos los resultados son igualados en el peor de los casos. Se puede señalar que estos peores casos fueron al aplicar los métodos de naturaleza categóricas en las bases de datos obtenidas al aplicar la metodología de preprocesamiento, la cual está enfocada a los algoritmos numéricos.

Abstract

Nowadays, companies are generating large volumes of data. To get useful information from them has become a very relevant issue for boosting the companies' development. Machine learning algorithms allow us to carry out this work. In many cases, however, data are not properly conditioned so as to obtain useful results. In this context, the object of this work is to propose a methodology for the preprocessing of structured data that allows us to obtain coherent results when machine learning techniques are applied. It is particularly focused on the numerical algorithms that underlie machine learning. The different stages that comprise the preprocessing of data and the problems that occur in each of the different cases were studied. We then proceeded to identify the algorithms that show better results, solving each of the preprocessing stages. Considering the selected algorithms, a methodology was defined and for its evaluation the Accuracy and the Cumulative Distribution Function were approximated by applying a set of methods on the chosen data. These methods were: Logistic Regression, Decision Trees, Artificial Neural Networks, Vector Support Machines, Random Forest and Gradient Boosting. Subsequently, the evaluation measures were obtained by applying the selected methods using cross-validation of 10 partitions in the non-preprocessed data. Then the data were pre-processed using the proposed methodology. The previous best results reported in the reviewed works were also taken into account. We used the same data sets to establish a comparison and evaluate the proposed methodology. Once we compared the results from our and theirs reported results, we noticed that, in general, our results are better. In some cases, all algorithms led to similar results. It can be pointed out that the cases when our results were significantly better occurred when data included categorical

attributes and in which our proposed methodology was applied. This is so because most of the analyzed algorithms are focused on numerical algorithms. In such cases results are better when a special preprocessing scheme to map categorical into numerical instances was implemented.

Índice

Índice de Tablas x

Índice de Figuras xii

Capítulo I. Introducción 1

1.1. Antecedentes 2

1.2. Situación Problemática 2

1.3. Objetivo General 3

1.4. Objetivos Específicos 3

1.5. Hipótesis 3

1.6. Contribuciones 4

Capítulo II. Estado del Arte y de la práctica 6

2.1. Introducción 6

2.2. Limpieza de Datos 10

2.3. Integración de Datos 13

2.4. Reducción de Datos 15

2.5. Transformación de Datos 16

Capítulo III. Descripción de las rutinas 19

3.1. Limpieza de Datos 19

3.2. Integración de Datos 24

3.3. Reducción de datos 27

3.4. Transformación de Datos 29

Capítulo IV. Casos de estudio 36

4.1. Introducción 36

4.2. Conjuntos de datos 36

4.3. Análisis de resultados obtenidos sobre los conjuntos de datos 38
4.4. Descripción de los métodos 40
4.5. Medidas para medir el rendimiento 44
4.6. Implementación de los algoritmos 46
4.7. Descripción de la metodología 46
4.8. Resultados 48
Capítulo V. Conclusiones 63
Bibliografía 64
Anexos

Índice de Tablas

Tabla 2.1 Técnicas empleadas para la limpieza de datos durante el preprocesamiento..... 6

Tabla 2.2 Técnicas empleadas para la integración de datos durante el preprocesamiento..... 7

Tabla 2.3 Técnicas empleadas para la reducción de datos durante el preprocesamiento..... 8

Tabla 2.4 Técnicas empleadas para la transformación de datos durante el preprocesamiento..... 8

Tabla 4.1 Resumen de características de los conjuntos de datos seleccionados.....38

Tabla 4.2 Resumen de resultados sobre el conjunto de datos *Census Icome*. 39

Tabla 4.3 Hiperparámetros definidos para usar la regresión logística. 41

Tabla 4.4 Hiperparámetros definidos para usar el árbol de decisión. 42

Tabla 4.5 Hiperparámetros definidos para usar la Red Neuronal.....43

Tabla 4.6 Hiperparámetros definidos para usar la SVM. 44

Tabla 4.7 Resultados obtenidos con los datos de *Census Income* sin preprocesar. 49

Tabla 4.8 Resultados obtenidos con los datos de Hepatitis sin preprocesar. 50

Tabla 4.9 Resultados obtenidos con los datos de *Breast Cancer Wisconsin* sin preprocesar. 50

Tabla 4.10 Resultados obtenidos con los datos de *Car Evaluation* sin preprocesar. 51

Tabla 4.11 Resultados obtenidos con los datos de *Census Income* preprocesados. 52

Tabla 4.12 Resultados obtenidos con los datos de Hepatitis preprocesados.....52

Tabla 4.13 Resultados obtenidos con los datos de Breast Cancer Wisconsin preprocesados. 53

Tabla 4.14 Resultados obtenidos con los datos de *Car Evaluation* preprocesados. 53

Índice de Figuras

Figura 4.1 Gráfica del AUC usando validación cruzada de 10 particiones en los datos preprocesados (GB). 55

Figura 4.2 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Census Income*. 55

Figura 4.4 Gráfica del AUC usando validación cruzada de 10 particiones en los datos preprocesados (SVM). 57

Figura 4.6 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de Hepatitis. 58

Figura 4.7 Gráfica del Área Bajo la Curva (AUC) usando validación cruzada de 10 particiones en los datos preprocesados (RF). 59

Figura 4.8 Gráfica del Área Bajo la Curva (AUC) usando validación cruzada de 10 particiones en los datos preprocesados (GB)....59

Figura 4.9 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Breast Cancer Wisconsin.*....60

Figura 4.10 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de *Breast Cancer Wisconsin.* 60

Figura 4.11 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Car Evaluation.* 61

Figura 4.12 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de *Breast Cancer Wisconsin.* 62

Capítulo I.

Introducción

Actualmente se está viendo cómo las empresas se enfrentan a un reto que cada día es mayor, y que está determinado por la gran cantidad de datos que se generará. Aunque el análisis de datos tiene más de medio siglo de haberse definido, en los últimos años ha presentado un crecimiento vertiginoso a causa de la necesidad de poder obtener información de éstos, para impulsar el desarrollo en las empresas y organizaciones tomando decisiones oportunas y adecuadas.

Los algoritmos de aprendizaje de máquina juegan un papel fundamental en el análisis de datos, pero muchas veces los encargados de implementar estas soluciones se centran más en escoger un algoritmo apropiado que en estudiar los datos en sí. Es útil ver si los mismos cuenta con ruidos, valores perdidos o conocer qué características deben tener los datos para que el algoritmo seleccionado devuelva resultados coherentes. Muchos de estos algoritmos asumen que los valores de entrada son numéricos, también algunos como las Redes de Perceptrones asumen que los valores de salida se encuentran entre 0 y 1. El no satisfacer las condiciones de entrada hacen que los resultados sean poco precisos y, en ocasiones, claramente malos. Por estas razones, se pretende desarrollar una metodología que facilite la aplicación de técnicas de aprendizaje de máquina de manera eficiente y evite cometer errores que son muy frecuentes.

1.1. Antecedentes

Para realizar el análisis relacionado con el preprocesamiento de datos fue necesario consultar diversos libros y trabajos relacionados con el tema de estudio, que constituyen los antecedentes de esta investigación como fue (Han, Pei, & Kamber, 2011) el cual cuenta con un capítulo donde se hace una introducción a técnicas para el preprocesamiento de datos. Este libro se refiere a los conceptos de calidad de los datos y discute métodos para su limpieza, integración, reducción, transformación y discretización. Además se revisó (García, Luengo, & Herrera, 2015) que tiene en cuenta las mismas etapas de (Han, Pei, & Kamber, 2011) pero describe métodos muchos más efectivos para cada una de las fases del preprocesamiento.

También se encontraron publicaciones que hacen análisis comparativos de algunos métodos, que se usan para llevar a cabo tareas específicas de preprocesamiento. Todas estas investigaciones realizan un análisis de diferentes métodos que se utilizan para realiza tarea de preprocesamiento de datos.

1.2. Situación Problemática

Se manifiesta como la necesidad de realizar la tarea de preprocesamiento de datos cuando se utilicen técnicas de aprendizaje de máquina, teniendo en cuenta todos los problemas que pueden tener los datos. En la literatura revisada no se encontró un procedimiento que abarque todas las etapas del preprocesamiento de datos de una manera integrada, y que permita realizar esta tarea lo más adecuadamente posible. Es relevante destacar que en la mayoría de los casos revisados no se documentan las tareas de preprocesamiento que se realizan. Teniendo en cuenta la gran cantidad de tareas para enfrentar cada uno de los problemas que los datos pueden tener, no existe un proceso sistematizado que permita afrontar el preprocesamiento de datos

estructurados de manera que personas con conocimientos limitados en el área puedan mejorar los resultados al aplicar técnicas de aprendizaje de máquina.

1.3. Objetivo General

Proponer una metodología para el preprocesamiento de datos estructurados que permita obtener resultados coherentes al aplicar técnicas de aprendizaje de máquina, particularmente enfocadas a los algoritmos numéricos que subyacen el aprendizaje de máquina.

1.4. Objetivos Específicos

1. Describir las principales etapas y tareas para el preprocesamiento de datos estructurados cuando se aplican técnicas de aprendizaje de máquina.
2. Identificar los métodos más adecuados para la realización de cada una de las tareas que componen el preprocesamiento de los datos.
3. Realizar pruebas con los diferentes métodos que conformarán la metodología para el preprocesamiento de datos.
4. Validar la metodología elaborada realizando comparaciones con resultados obtenidos sobre los conjuntos de datos seleccionados sin ser preprocesados.

1.5. Hipótesis

Si se propone una metodología para realizar el preprocesamiento de conjunto de datos estructurados entonces personas con pocos conocimientos en el área podrían mejorar los resultados cuando aplican técnicas de aprendizaje de máquina.

1.6. Contribuciones

- Una metodología de preprocesamiento para base de datos estructuradas que permita que personas con pocos conocimientos en el área puedan realizar esta tarea de manera guiada y obtener resultados coherentes al aplicar técnicas de aprendizaje de máquina.
- Una herramienta de código abierto para realizar las tareas de preprocesamiento que propone la metodología, de manera que sea muy intuitivo aplicar la metodología por personas con conocimientos mínimos.

El presente trabajo está compuesto por 4 capítulos, conclusiones, bibliografía y anexos.

Capítulo II. Estado del Arte y de la Práctica.

Aborda los principales conceptos asociados al problema, describe las diferentes etapas del preprocesamiento de los datos, así como los métodos más comúnmente usado. Se realiza un estudio de estos métodos con el fin de seleccionar aquellos con mejores resultados prácticos por cada una de las etapas que fueron identificadas.

Capítulo III. Descripción de las rutinas.

Se detallan desde el punto de vista teórico cada uno de los algoritmos seleccionados como más convenientes para realizar las tareas que corresponden a cada una de las etapas. Se determinan todos aquellos aspectos para entenderlos y poder implementarlos, con el fin de hacer un uso adecuado al momento de proponer la metodología.

Capítulo IV. Casos de Estudio

Se seleccionan los conjuntos de datos que se usarán para realizar las pruebas que permitirán comparar los resultados cuando se realice el preprocesamiento. También se revisarán las investigaciones que estudian estos conjuntos de datos seleccionados con el fin de ver los mejores resultados obtenidos y establecer un punto de referencia para comparar. Se definen los métodos y métricas de rendimiento para realizar la comparación con los datos luego de preprocesar usando la metodología que este se propone. Además, se muestran los resultados de los métodos al usar los datos preprocesados y se comparan con los resultados obtenidos en las investigaciones examinadas.

Capítulo II.

Estado del Arte y de la práctica

2.1. Introducción

Las etapas para realizar el prepocesamiento de datos pueden ser diversas. Si nos basamos en las investigaciones de (García et al., 2015) y (Han et al., 2011) se pueden resumir de la siguiente forma:

- Limpieza de Datos.
- Integración de Datos.
- Reducción de Datos.
- Transformación de Datos.

En cada una de estas etapas se realizan un grupo de tareas para enfrentar los problemas que tienen los datos, y que pueden afectar los resultados cuando se aplican técnicas de aprendizaje de máquina. Existe gran diversidad de formas para resolver los problemas que pueden tener los datos, la cuales vienen de muy diversos enfoques. A continuación, se resumen las etapas del preprocesamiento, las tareas a realizar en cada una de las etapas y las técnicas reportadas con más frecuencia en la literatura consultada.

Tabla 2.1 Técnicas empleadas para la limpieza de datos durante el preprocesamiento.

Limpieza de Datos
Imputación de valores perdidos

Eliminar las tuplas con valores perdidos
Imputación usando Medidas de Tendencia Central
Imputación por regresión
Imputación usando Esperanza Maximización
Imputación no estocástica
Imputación usando algoritmos de agrupamientos
Imputación usando *Spline* Natural
Detección de valores atípicos
Modelos de mezclas Gaussianas
Estimadores de densidad por *Kernel*
Análisis de componentes principales probabilísticos
Detección de valores atípicos por Mínimos Cuadrados
Distancia de *Mahalanobis*
Análisis de valores atípicos locales
Divergencia Kullback-Leibler
Máquina de Soporte Vectorial de una clase
Isolation forest

Tabla 2.2 Técnicas empleadas para la integración de datos durante el preprocesamiento.

Integración de Datos
Redundancia en los datos
Análisis de Correlación de Chi-Cuadrado

Análisis de Correlación de Pearson
Análisis de Covarianza

Tabla 2.3 Técnicas empleadas para la reducción de datos durante el preprocesamiento.

Reducción de Datos
Reducción de dimensionalidad
Transformada de Wavelet
Análisis de Componentes Principales
Selección de variables
Reducción de Numerosidad
Histogramas
Agrupamiento
Muestreo
Agregación de Cubo de Datos

Tabla 2.4 Técnicas empleadas para la transformación de datos durante el preprocesamiento.

Transformación de Datos
Construcción de características
Agregación
Normalización
Escalamiento

Estabilización
Generation of concept hierarchies for nominal data
Codificación estadística de variables categóricas
Codificación binaria de variables categóricas
Conversión de fecha gregoriana a fecha juliana

En el análisis anterior se puede ver que existe una gran variedad técnicas que se realizan en las diferentes etapas del preprocesamiento de datos. Esto demuestra que existen una gran variedad de algoritmos para intentar afrontar los problemas que pueden tener los datos, incluso para afrontar el mismo problema las variantes pueden ser diversas.

En las revisiones se puede notar que en muchas investigaciones no se reportan las técnicas empleadas en el preprocesamiento, aunque en algunas si se menciona que se tiene en cuenta. En otras como (P. C. González, 2018) se lleva a cabo la etapa de preprocesamiento para lograr la integridad de los datos. Se enfocan en no tener datos nulos y no numéricos solamente, y para ello proponen la asignación de -1. También proponen eliminar los atributos que no tienen ningún dato. En (B. V. González & Mora, 2018) solo se tiene en cuenta la etapa de Reducción de variables usando el Análisis de Componentes Principales para descubrir la verdadera dimensionalidad de los datos. Se pudo constatar que en (Pareja, 2017) se tiene en cuenta una etapa de selección de variables en el procesamiento, para estimar la asociación que existen entre las variables. Para esto realizan las siguientes pruebas: *Pearson's Chi-squared Test*, *Sperman Rank Correlation coefficient*, *Kendall rank correlation coefficient* y *Goodman-Kruskal Gamma for ordered tables.*

También se puede encontrar algunas aplicaciones comerciales de preprocesamiento de datos. Estas herramientas como *Tableau Prep*, *Data Prep*, *Datawatch Monarch* por mencionar algunas, incorporan un grupo de técnicas para hacer el preprocesamiento, pero no definen un proceso o metodología para realizarlo. Las mismas traen un grupo de técnicas para enfrentar los problemas que pueden tener los datos de manera que cada quién pueda seleccionar la técnica y aplicarla de manera intuitiva.

Se puede notar que hay gran variedad de técnicas y que no existe sistematicidad a la hora de emplearlo para mejorar la calidad de los datos. En cada una de estas investigaciones se tienen en cuenta etapas diferentes del preprocesamiento y en ocasiones hacen uso de técnicas poco adecuadas para enfrentar el problema. Durante este capítulo se revisarán cada una de las etapas para el preprocesamiento de datos y se harán análisis comparativos entre las técnicas para tratar de identificar las más pertinentes a la hora de resolver cada uno de los problemas en los datos.

2.2. Limpieza de Datos

La limpieza de datos consiste en la corrección de problemas causados principalmente en el proceso de recolección de datos. Algunos de estos problemas que se encuentra son los valores perdidos, las inconsistencias en los datos, valores con ruido y valores atípicos. En esta sección se analizan y se recomiendan métodos para la imputación de valores perdidos y la detección de valores atípicos que son algunos de los problemas más frecuentes en los datos obtenidos del mundo real.

Valores perdidos

Cuando procedemos a realizar el análisis de una base de datos, puede suceder que falten valores, es decir, que las variables para algunas o muchas tuplas no tengan sus datos correspondientes. Lo ideal sería que los datos con los que se cuenta sean los suficientes para arrojar buenos resultados y que eliminar las tuplas que contengan valores faltantes no sea un problema con respecto a la perdida de información. Siempre que se usan métodos de imputación existe la posibilidad de introducir información errónea.

En caso de que se proceda con la imputación de datos, existen diversas estrategias como la imputación usando medidas de tendencia central (Schafer & Graham, 2002). Estas pueden ser adecuadas para lograr predicciones más precisas. Pero tiene implicaciones negativas en la varianza del estimador e introduce distorsiones en el patrón de correlación de los datos. También se utilizan imputaciones por regresión que, aunque son superiores a la imputación por media, se puede afirmar que también tienen sesgos predecibles (Enders, 2010). El algoritmo de Esperanza Maximización (EM) es un método que busca la máxima verosimilitud y también es empleado en la imputación (Roth, 1994), demostrando ser superior a la eliminación, a la imputación no estocástica y métodos de regresión por imputación estocástica.

Existen además estrategias que se basan en algoritmos de agrupamientos los cuales tiene como objetivo dividir el conjunto de datos en grupos. Cada dato pertenece al grupo cuyo valor medio es más cercano. Aunque se pueden encontrar variaciones de estos algoritmos, estudios como (Li, Deogun, Spaulding, & Shuart, 2004) demuestran que la versión difusa del *K-Means* logra mejores resultados.

La imputación con *splines* es otra de las formas usadas para realizar esta tarea. En (Kuri-Morales, 2015) se concluye que usando el *Spline* Natural se puede obtener una expresión que es simple de calcular, económica [en que un *spline*, para n tuplas, necesita solamente n-2 valores (los valores de las segundas derivadas) para ser definido completamente] y estable [la curvatura del *spline* es mínima]. Se dice que un *spline* Natural es la mejor manera de interpolar un conjunto de datos dado (ver sección 2.2) cuando no se sabe nada más sobre tales datos, siendo esta la forma que se utilizará aquí para llevar a cabo la imputación de valores perdidos.

Valores atípicos

Los valores atípicos son patrones en los datos que no coinciden con una noción bien definida de acciones normales, o ni con una noción bien definida de comportamiento aislado (Subramanlan K, 2011). También se pueden encontrar otras definiciones en (Hawkins, 1980) y (Johnson & Wichern, 1992).

La naturaleza y el tipo de valores atípicos es un importante aspecto para estudiar los enfoques que permiten su detección. En (Subramanlan K, 2011) se describen tres tipos de valores atípicos y los clasifican en valor atípico de Tipo I, valor atípico de Tipo II y valor atípico de Tipo III con sus particularidades, aunque en (Sadawarti, Ieee, & Kalra, 2014) se define otra clasificación.

También existen diferentes enfoques para la detección de valores atípicos como la que proponen en (Sadawarti et al., 2014) y se resumen en: métodos estadísticos, métodos paramétricos y no paramétricos, métodos basados en proximidad, métodos basados en distancia, métodos basados en densidad, métodos de agrupamiento y, finalmente, técnicas de redes neuronales para detectar y analizar valores atípicos. En

(Kaur & Garg, 2016), (Aggarwal, 2015) proponen otras formas para estructurar los enfoques.

En (Sadawarti et al., 2014) se categorizan y comparan una gran cantidad de enfoques para la detección de valores atípicos, y se hace hincapié que no existe ningún enfoque aceptado universalmente. (Kaur & Garg, 2016) concluye que no existe un enfoque universalmente aplicable e identifica que los algoritmos de agrupamiento son comparativamente mejores, mientras que (Bakar, Mohemad, Ahmad, & Deris, 2006) determina que la distancia de Manhattan es superior a otras técnicas del enfoque basado en distancia y enfoque basado en estadística.

Teniendo en cuenta lo antes mencionado y la diversidad de resultados se decidió usar el método *Isolation Forest* (Liu, Ting, & Zhou, 2008) que demostró tener excelentes resultados en una investigación realizada por (Domingues, Filippone, Michiardi, & Zouaoui, 2018) donde se realiza un análisis comparativo de 14 algoritmos de diferentes enfoques en conjuntos de datos sintéticos y reales. Además de los excelentes resultados obtenidos muestra escalabilidad en grandes conjuntos de datos y un uso aceptable de memoria.

2.3. Integración de Datos

La integración de datos es el proceso mediante el cual se combinan los datos de diferentes fuentes y de esta manera se obtiene un único conjunto de datos unificado. Realizar este proceso de manera exhaustiva y cuidadosa nos permite disminuir y, en el mejor de los casos, evitar completamente las redundancias e inconsistencias que pudiera traer consigo esta tarea. Obtener un conjunto de datos, resultados de una buena integración, ayudaría a mejorar la precisión y la velocidad cuando se apliquen

técnicas de aprendizaje de máquina. Esta etapa se centra en describir algunos métodos para analizar los datos después de que fueron integrados y de esta manera evitar los problemas que pudieran existir.

Redundancia en los datos

La redundancia se refiere al almacenamiento de los mismos datos varias veces de manera inútil. Lo cual trae consigo un aumento en el tamaño del conjunto de datos y una demora para obtener los resultados cuando se aplican algoritmos de aprendizaje de máquina. Podemos considerar que una variable es redundante cuando puede ser derivada de otra u otras variables. El análisis de correlación de Pearson nos permite identificar la relación lineal entre dos variables cuantitativas por lo que se utiliza para determinar la redundancia de las variables numéricas. En el caso de las variables nominales se usará la prueba de Chi-Cuadrada que es una prueba de hipótesis que compara la distribución observada de los datos con una distribución esperada de los datos. Existe la Prueba de bondad de ajuste de Chi-Cuadrada, que permite probar qué tan bien una muestra de datos categóricos se ajusta a una distribución teórica y las Pruebas de Chi-Cuadrada de independencia y asociación. La Prueba de independencia permite determinar si el valor observado de una variable depende del valor observado de otra variable, y la Prueba de asociación permite determinar si una variable está asociada a otra variable y será utilizada para determinar la redundancia entre variables nominales.

2.4. Reducción de Datos

En esta etapa se incluirán las estrategias de reducción de dimensionalidad y reducción de numerosidad. La reducción de numerosidad es el proceso mediante el cual se eliminan un número de atributos basado en un criterio seleccionado. La reducción de dimensionalidad consiste en disminuir el número de tuplas.

Reducción de la dimensionalidad

La reducción de la dimensionalidad es un proceso que se realiza para evitar el efecto Hughes (Oommen et al., 2008), fenómeno que surge al analizar y organizar datos de espacios de múltiples dimensiones. Al aumentar la dimensionalidad, el volumen del espacio aumenta geométricamente haciendo que los datos disponibles se dispersen.

La reducción de dimensionalidad se puede enfrentar usando el Análisis de Componentes Principales (PCA) y las Transformadas de Wavelets según (Han et al., 2011).

En (Van Der Maaten, Postma, & den Herik, 2009) realizan una revisión y comparación de algunas técnicas no lineales propuestas con el objetivo de abordar las limitaciones de las técnicas tradicionales, concluyendo que las técnicas no lineales para la reducción de dimensionalidad son a menudo incapaces de superar las técnicas tradicionales lineales, como PCA. Esta conclusión nos motiva a trabajar con PCA para llevar a cabo la tarea de reducción de dimensionalidad.

Reducción de la numerosidad

Para la reducción de la numerosidad en los datos existen varios métodos que pueden usarse según (Han et al., 2011), los histogramas, agregación, agrupamiento y muestreo son alguno de ellos.

En este trabajo se usará el muestreo, que es una la técnica para la selección de una muestra a partir de una población. Por lo que puede usarse como una técnica de reducción de datos porque permite que un conjunto grande de datos sea representado como un subconjunto (muestra) de datos más pequeña.

Existen varias técnicas de muestreo según (Bencardino, 2003) y (Cochran, 1980). El muestreo aleatorio simple es una de estas técnicas y consiste en seleccionar una muestra a partir de una población, en donde cada elemento tiene igualdad de probabilidad de ser seleccionado. Estos conjuntos de técnicas son aconsejables y permiten una alta representatividad de la población. Estas técnicas a su vez se dividen en diferentes tipos, de los cuales se le prestará atención al muestreo aleatorio simple sin reposición, el cual impide que un elemento que haya sido extraído para la muestra vuelva a ser extraído en las siguientes etapas del muestreo.

2.5. Transformación de Datos

En la transformación de datos, los datos se transforman o consolidan en formas apropiadas para la minería. Las estrategias para la transformación de datos incluyen lo siguiente:

Escalamiento: Cuando se refiere a escalamiento no debe confundirse con normalización. Escalamiento es el proceso de aplicar una transformación lineal en los

datos originales con el objetivo de agruparlos en un intervalo previamente definido, que en la mayoría de los casos suele ser [0,1] o [-1,1].

Estabilización: Cuando se aplican algunas técnicas de aprendizaje de máquina, pueden existir situaciones en la que los algoritmos presenten inestabilidad matemática debido a la naturaleza de los algoritmos. Para evitar este tipo de comportamiento se estabilizan los datos, cuyo proceso consisten en realizar una perturbación a partir de un grado decimal N.

Codificación de las variables categóricas: Cuando se trabaja con base de datos mixtas aparecen variables categóricas. En estos casos los algoritmos de aprendizaje de máquina basados en métricas no son aplicables. En la práctica muchos especialistas suelen asignar valores numéricos para diferenciar las categorías, lo cual está mal porque en caso de aplicar esta técnica estamos introduciendo una relación que no necesariamente existe y que un algoritmo puede percibir y obtener resultados inadecuados. Existen algunos otros métodos que nos permiten evitar este tipo de problemas como son:

> ***Codificación binaria para variables categóricas:*** El algoritmo para la codificación binaria de las variables categóricas a cada categoría de una variable le crea una nueva variable y asigna valores de 0 si no corresponde con el valor de la categoría o 1 en caso contrario. Este proceso hace que aumente la dimensionalidad de los datos considerablemente, aunque es un método comúnmente utilizado en las investigaciones.
>
> ***Codificación estadística para variables categóricas:*** El objetivo del algoritmo para la codificación estadística de variables categóricas es sustituir

las categorías por valores numéricos, de manera que estos maximicen la correlación con las demás variables y convierte el problema de no métrico a métrico. Los códigos obtenidos se pueden utilizar solamente en el propio problema, no pueden ser extendidos a otros conjuntos de datos.

Conversión de fecha gregoriana a fecha juliana: El calendario gregoriano es utilizado de manera oficial en casi todo el mundo y fue el que sustituyó al calendario Juliano en 1582. La fecha gregoriana que es la forma en que actualmente se almacenan las fechas en la mayoría de las bases de datos, contiene día, mes y año. Esta fecha puede ser representada de diferentes maneras dependiendo de la región geográfica, aunque la representación completa del formato extendido definido por ISO (ISO 8601) intenta solucionar ese problema y especifica la siguiente estructura para las fechas: AAAA-MM-DD. Esta norma ISO también define las especificaciones para representar la hora del día, los intervalos y además para la fecha y horas conjuntas. Teniendo en cuenta las estructuras antes descritas los algoritmos de aprendizaje podrían hacer interpretaciones equívocas de la información. Debido a esto sería adecuado convertir las fechas gregorianas a fechas juliana para obtener un único valor numérico que contenga la información, preserve las relaciones y pueda ser útil para los algoritmos de aprendizaje de máquina.

Capítulo III.

Descripción de las rutinas

3.1. Limpieza de Datos

Imputación de valores perdidos

Cuando se tienen valores perdidos en una base de datos se pueden tomar dos caminos que son muy evidentes. Elimino las tuplas con valores perdidos o se imputan estos valores perdidos. Para evitar que si se eliminan las tuplas se pierda demasiada información en los datos, se utilizará el cálculo de Entropía. Este proporciona una forma de medir el comportamiento esperado de la fuente ψ, donde P_i es la probabilidad de un símbolo y se defina entropía (que usualmente denotamos por "H") como:

$$H(\psi) = -\sum_{i=1}^{n} P_i \log_2 P_i$$

Si se conoce la diferencia de información que existe entre los datos que le fueron eliminadas las tuplas que tenían valores perdidos y los datos con los valores perdidos imputados se podría determinar qué es más conveniente. En caso de que la diferencia de información sea demasiado grande (determinada por un umbral) se puede identificar con cual conjunto de datos continuar trabajando.

Si se determina que la cantidad de información de los datos resultantes de la eliminación de tuplas con valores perdidos no son significativas se continúa

trabajando con la misma y así se evita introducir ruidos en los datos durante el proceso de imputación. En caso contrario se procede a imputar usando el

spline Natural y para ello comenzamos mencionando que la aproximación polinomial de la forma $y = \sum_{i=0}^{n} c_i x^i$, requiere de *n+1* punto de datos y el grado más alto es *n*. Sin embargo, a medida que *n* crece, aumenta el número de ecuaciones y esto puede llevar a que el error del interpolador pueda aumentar, en el sentido que los valores interpolados sean muy diferentes a los observados.

Para evitar estos casos insatisfactorios, se puede olvidar la idea de colocar todos en un polinomio y, en su lugar, intentar colocar los datos en varios polinomios, que estén bien definidos y para ello deben cumplir lo siguiente:

1. Tener n elementos (para los *n+1* puntos de los datos) en el conjunto de polinomio el cual denotamos como *S(x).* El i-ésimo elemento del conjunto esté representado por *Si(x).*
5. El conjunto de polinomios requiere que cumpla con $S(x) = y_i$ para toda *i*, es decir, este tiene que colocar las observaciones.
6. También requiere que el conjunto sea "suave" entre los vecinos de los elementos. Nosotros garantizamos esto si $S_i'(x_i) = S_{i+1}'(x_i)$ para *i=1, 2, …, n-2.*
7. Además, se requiere que la concavidad de S(x) no cambie de manera abrupta, es decir, $S_i''(x_i) = S_{i+1}''(x_i)$ para *i=1, 2, …, n-2.*
8. Por último, queremos que la curvatura de *S(x)* sea tal que las oscilaciones entre los datos observados se mantengan lo más pequeña posible (dado que se cumpla la condición 1).

Un *spline* Natural es un conjunto de polinomios cúbicos *S(x)* de la forma:

$$S(x) = \sum_{i=0}^{n-1} (a_{i,i+1} + b_{i,i+1}x + c_{i,i+1}x^2 + d_{i,i+1}x^3)\delta(x)$$

$$\delta(x) = \begin{cases} 1, & x_i \leq x \leq x_i \\ 0, & otros\ casos \end{cases}$$

Los elementos del *spline* son de grado 3 porque éste es el grado más pequeño, que cumple con todas las condiciones anteriores.

Detección y eliminación de valores atípicos

El método seleccionado para detectar valores atípicos es *Isolation Forest* o *iForest* (iF), el cual crea un conjunto de *iTrees* para un grupo de datos determinados. Luego los valores atípicos son aquellas instancias que tienen longitudes de rutas promedio cortas en los *iTrees*. Solo hay dos variables en este método: la cantidad de árboles para construir y el tamaño del submuestreo. Mostramos que el rendimiento de detección de iF converge rápidamente con una cantidad muy pequeña de árboles, y solo requiere un pequeño tamaño de submuestreo para lograr un alto rendimiento de detección con una alta eficiencia.

El iF se distingue de otros enfoques existentes basados en modelos, basados en distancia y basados en la densidad en los siguientes aspectos (Liu et al., 2008):

- La característica de aislamiento de *iTrees* les permite construir modelos parciales y explotar el submuestreo a una medida en que no es factible en otros métodos existentes. Una gran parte de un *iTree* que aísla puntos normales no es necesaria para la detección de anomalías; no necesita ser

construido. Un tamaño de muestra pequeño produce mejores *iTrees* porque se reducen los efectos de enmascaramiento.

- iF no utiliza medidas de distancia o densidad para detectar los valores atípicos. Esto elimina los cálculos necesarios en los métodos basados en distancia y los métodos basados en densidad, logrando disminuir un gran costo computacional.
- Tiene una complejidad de tiempo lineal con una baja constante y un bajo requerimiento de memoria.
- iF tiene la capacidad de ampliarse para manejar un tamaño de datos extremadamente grande y problemas de gran dimensión con una gran cantidad de atributos irrelevantes.

Se entiende por *Isolation* (aislamiento) separar una instancia del resto de las instancias. En un árbol aleatorio inducido por datos, la partición de instancias se repite recursivamente hasta que todas las instancias se aíslan. Esta partición aleatoria produce trayectorias y cuando se tiene una longitud de camino más corta para puntos específicos es más probable que sea una anomalía.

En (Liu et al., 2008) se describe el proceso de detección de anomalías con iF en dos etapas:

1. Etapa de entrenamiento:

Se construyen recursivamente particiones del conjunto de entrenamiento hasta que las instancias se aíslen o hasta que se alcance una altura específica en el árbol. La altura del árbol se establece por el tamaño del submuestreo, que es aproximadamente la altura promedio del árbol. Esto se debe a que solo nos interesan los puntos de datos que tienen longitudes de ruta más corta que el promedio, ya que son los más

propensos a ser anomalías. Los parámetros de entrada de este algoritmo son el tamaño del submuestreo (ψ) y el número de árboles.

9. Etapa de evaluación.

Una puntuación de anomalía es obtenida de la longitud de camino esperada *E(h(x))*, para cada instancia de prueba. *E(h(x))* se derivan pasando instancias a través de cada *iTree* en un *iForest*. Se calcula la longitud de camino (*h(x)*) contando el número de aristas desde el nodo raíz hasta el nodo de terminación. Cuando se obtiene *h(x)* para cada árbol del conjunto, se obtiene una puntuación de anomalía calculando *s (x, ψ)*.

Cálculo de la puntuación de anomalía (Liu & Ting, 2012).

Dado que *iTrees* tiene una estructura equivalente a *Binary Search Tree* o *BST*, la estimación de *h(x)* promedio para las terminaciones de nodos externos es la misma como el de las búsquedas ineficaz en *BST*. Tomamos prestado el análisis de *BST* para estimar la longitud de camino promedio de *iTree*. Dado un conjunto de muestra de ψ instancias, se proporciona la duración promedio de las búsquedas ineficaz con *BST* como:

$$f(x) = \begin{cases} 2H(\psi - 1) - \dfrac{2(\psi - 1)}{n} & para\ \psi > 2, \\ 1 & para\ \psi = 2, \\ 0 & otros\ casos. \end{cases}$$

Donde *H(i)* es el número armónico y puede ser estimado por *ln(i) + Euler*. Como c(ψ) es el promedio de *h(x)* dado ψ, se usa para normalizar *h(x)*. La puntuación de anomalía de una instancia *x* está definida como:

$$s(x, \psi) = 2^{-\frac{E(h(x))}{c(\psi)}}$$

Donde *E(h(x))* es el promedio de *h(x)* de una colección de *iTree*. A continuación, se presentan las condiciones para tres valores especiales de puntuaciones de anomalía:

cuando *E(h(x)) -> 0, s->1*;

cuando *E(h(x))->*ψ*-1, s->0*; y

cuando *E(h(x)) -> c(*ψ*), s->0.5*.

Usando las puntuaciones de anomalía, podemos hacer la siguiente evaluación:

(i) si las instancias devuelven s muy cerca de 1, entonces definitivamente son anomalías,

(ii) si los casos tienen s mucho más pequeño que 0.5, entonces son bastante seguros para ser considerados como instancias normales, y

(iii) si todas las instancias devuelven $s \approx 0.5$, entonces la muestra completa en realidad no tiene ninguna anomalía distintiva.

3.2. Integración de Datos

Prueba de chi-cuadrada

Para un conjunto de datos nominales, se puede obtener la asociación de dos variables A y B mediante la prueba de chi-cuadrada. Supongamos que A tiene v valores distintos, *a* saber, $a_1, a_2, \ldots a_v$. B tiene w valores distintos, es decir, $b_1, b_2, \ldots b_w$. Las

tuplas de datos descritas por A y B pueden mostrarse como una tabla de contingencia, con los valores v de A que componen las columnas y los valores de w de B que componen las filas. Sea (A_i, B_j) el evento conjunto que el atributo A toma en el valor a_i y el atributo B toma el valor b_j, es decir, donde $(A = a_i, B = b_j)$. Todos y cada uno de los posibles eventos conjuntos (A_i, B_j) tienen su propia celda en la tabla. El valor $\chi 2$ (también conocido como la estadística Pearson $\chi 2$) se calcula como:

$$\chi 2 = \sum_{i=1}^{v} \sum_{j=1}^{w} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \qquad \text{Ec. 3.1}$$

donde o_{ij} es la frecuencia observada del evento conjunto (A_i, B_j) y e_{ij} es la frecuencia esperada de (A_i, B_j) calculada como:

$$e_{ij} = \frac{\text{cantidad}(A=a_i) * \text{cantidad}(B=b_j)}{m} \qquad \text{Ec. 3.2}$$

Donde m es el número de instancias en el conjunto de datos, cantidad $(A=a_i)$ es el número de instancias con el valor a_i para el atributo A y la cantidad $(B=b_j)$ es el número de instancias que tienen el valor de b_j para el atributo B.

La prueba $\chi 2$ comprueba la hipótesis de que A y B son independientes, con $(r-1) * (c-1)$ grados de libertad. La estadística obtenida en la Ecuación 3.1 se compara con la tabla de $\chi 2$ usando los grados de libertad adecuados. Si el nivel de significación de dicha tabla es inferior al establecido (o el valor estadístico obtenido es superior al necesario en la tabla), podemos decir que la hipótesis nula es rechazada y, por lo tanto, no pude afirmarse que A y B están estadísticamente correlacionados.

Correlación de Pearson

El coeficiente de correlación de Pearson(r) es el más utilizado y mide lo bien que se ajustan los puntos a una línea recta ideal. Es un método estadístico paramétrico, ya que utiliza la media, la varianza, etc. Y, por tanto, requiere de criterios de normalidad para las variables analizadas. El valor de correlación r será grande cuando los puntos estén muy concentrados en torno a la recta y será pequeño cuando los puntos en el gráfico estén muy dispersos respecto a la recta imaginaria que define la relación. A veces se puede concebir la correlación como la "fuerza de la asociación lineal" entre dos variables y el rango de valores en los que se mueve este coeficiente [-1, +1] es una cuantificación del grado en que se asocian dos variables, independientemente de cuales sean las unidades de medida de los valores.

Los valores de r pueden interpretarse de la siguiente manera según (Martínez-González, Sánchez-Villegas, & Faulín, 2014):

Para realizar el cálculo de r para A y B la fórmula es:

$$r_{AB} = \frac{n\sum a_i b_i - \sum a_i \sum b_i}{\sqrt{n\sum a_i^2 - (\sum a_i)^2}\sqrt{n\sum b_i^2 - (\sum b_i)^2}}$$

Donde n es el tamaño de muestra y a_i, b_i son las muestras individuales indexadas con i.

3.3. Reducción de datos

Análisis de Componentes Principales

El análisis de componentes principales (PCA) es un procedimiento estadístico que utiliza una transformación ortogonal para convertir un conjunto de observaciones de variables posiblemente correlacionadas en un conjunto de valores de variables linealmente no correlacionadas llamadas componentes principales. El número de componentes principales distintos es igual al menor entre el número de variables originales o el número de observaciones menos uno. Esta transformación se define de tal manera que el primer componente principal tiene la mayor varianza posible (es decir, representa la mayor variabilidad posible en los datos), y cada componente sucesivo a su vez tiene la mayor varianza posible bajo la restricción de ser ortogonal a los componentes anteriores. Los vectores resultantes son un conjunto de bases ortogonales no correlacionadas. PCA es sensible a la escala relativa de las variables originales.

El procedimiento es el siguiente y se describe detalladamente en (Han et al., 2011):

1. Los datos de entrada no se normalizan, sino que entran en esa distribución. Este paso ayuda a asegurar que los atributos con dominios ensanchados no dominen los atributos con dominios más pequeños.
2. PCA calcula k vectores ortonormales que proporcionan una base para los datos de entrada normalizados. Estos son vectores unitarios que apuntan en una dirección perpendicular a los demás. Estos vectores se conocen como los componentes principales. Los datos de entrada son una combinación lineal de los componentes principales.

3. Los componentes principales se ordenan en orden de "significación" decreciente o fuerza. Los componentes principales sirven esencialmente como un nuevo conjunto de ejes para los datos, proporcionando información importante sobre la varianza. Es decir, los ejes ordenados son tales que el primer eje muestra la mayor varianza entre los datos, el segundo eje muestra la siguiente varianza más alta, y así sucesivamente. Esta información ayuda a identificar grupos o patrones dentro de los datos.
4. Debido a que los componentes se clasifican en un orden decreciente o de "importancia", el tamaño de la información puede reducirse eliminando los componentes más débiles, es decir, aquellos con baja varianza. Usando los componentes principales más fuertes, debería ser posible reconstruir una buena aproximación de los datos originales.

PCA se puede aplicar a atributos ordenados y desordenados, y puede manejar datos dispersos y sesgados. Los datos multidimensionales de más de dos dimensiones se pueden manejar reduciendo el problema a dos dimensiones. Los componentes principales se pueden usar como entradas para la regresión múltiple y el análisis de agrupamiento.

Muestreo Aleatorio Uniforme Sin Reposición

El muestreo es el proceso de obtención de una muestra de una población, el término uniforme nos indica que todos los elementos tienen la misma probabilidad de pertenecer a la muestra y que sea sin reposición garantiza que ninguno de los

elementos que ya han sido muestreados van a volver a ser seleccionados para formar parte de la muestra, lo que indica que solo aparecerán una vez en la muestra.

3.4. Transformación de Datos

Escalamiento

Para el proceso se cuenta con los valores mínimo (min) y máximo (max) de la variable a la que se le quiere aplicar el escalamiento, además del nuevo valor mínimo (nuevo_min) y el nuevo valor máximo (nuevo_max) que determinan el nuevo intervalo de los valores para esta variable. Para obtener los v_i' de cada uno de los v_i se define la Ecuación 3.3, que se muestra a continuación:

$$v_i' = \frac{v_i - \min}{\max_A + \min_A}(\text{nuevo}_{max} - \text{nuevo}_{min}) + \text{nuevo}_{min} \qquad \text{Ec. 3.3}$$

La necesidad del escalamiento se debe a que los datos son recopilados de diferentes fuentes y esto puede traer asociado inconsistencias. Puede darse el caso de usar diferentes unidades de medidas en el momento de capturar la información. Por ejemplo, puede ser la estatura que en algunos casos la tengamos en centímetros y en metro. O el caso del peso, el cual puede ser en libras o en kilogramos.

También en algunos algoritmos de aprendizaje de máquina como las Redes de Perceptrones todas las entradas deben tener la misma ponderación. Si las entradas de dos neuronas se encuentran en rangos diferentes, entonces la neurona con la escala absoluta más grande será favorecida durante el entrenamiento. También asume que los valores de salida se encuentran entre 0 y 1, estos son algunos de los motivos por lo que es necesario llevar a cabo el escalamiento.

Si los datos utilizados con una red neuronal no se escalan a un rango apropiado, entonces la red no convergerá en el entrenamiento o no producirá resultados significativos.

Estabilización

El proceso de estabilización de un conjunto de datos D, nos permite obtener un conjunto de datos D^* y consiste en realizar una perturbación a partir de un grado decimal N. Lo cual podría verse como:

$$f(x) = \begin{cases} 10^{-N} * \in & a=0 \\ a+\left(10^{-N} * \in\right) & a\neq 0 \end{cases}$$

Donde N determina a partir de qué grado decimal se quiere realizar la perturbación y $\in$ es el valor de la perturbación aleatoria.

Cuando se realiza este procedimiento se está modificando los datos originales D y se obtiene un nuevo problema que difiere al original. En (Kuri-Morales & Galaviz Casas, 2002) se demuestra que este proceso de condicionamiento lleva a una diferencia relativa entre la función evaluada en D^* y la función evaluada en D del orden de 10^{-N}. Por otra parte, la diferencia de los errores minimax D^* y D, es también, del orden de 10^{-N}. Debido a que el valor 10^{-N} es pequeño, el aproximante que se obtiene del conjunto perturbado aleatoriamente es muy cercano al aproximante original obtenida de D.

Codificación binaria para variables categóricas

Cuando se analizan bases de datos se encuentran en muchos casos valores que representan categorías (ej. Estado Civil) el cual puede ser sustituido por valores

numéricos continuos (ej. valor 1 para el caso de Casado, valor 2 para el caso de Soltero, valor 3 para el caso de Viudo, etc.). Cuando se realiza este tipo de transformación se está estableciendo una relación entre los diferentes valores de esta variable la cual es analizada e interpretada de manera incorrecta por los algoritmos de aprendizaje de máquina. Para codificar los datos de una manera adecuada y de esta manera evitar que los algoritmos arrojen información inconsistente, se puede definir una variable nueva por cada una de las categorías que tenga la variable en cuestión. En caso de que correspondiera con esta categoría tomaría el valor de 1 y en caso contrario el valor de 0. Para el ejemplo de la variable Estado Civil (Z) que antes se mencionaba, la cual tenía 3 categorías (Soltero (1), Casado (2) y Viudo (3)) ahora se definen 3 variables (Estado Civil Soltero (Z1), Estado Civil Casado (Z2), Estado Civil Viudo (Z3)) y para cada caso se le asignarán los valores de 0 o 1 como antes se mencionaba. El símbolo # en la Tabla 3.1 es equivalente a un número cualquiera.

Tabla 3.1. Ejemplo de aplicar Codificación binaria A) Es una manera no adecuada de representar los datos que son variables categóricas. B) Es la forma de representar las variables categóricas usando codificación binaria.

A)

Variables			
W	X	Y	Z
#	#	#	2
#	#	#	1

B)

Variables					
W	X	Y	Z1	Z2	Z3
#	#	#	0	1	0
#	#	#	1	0	0

#	#	#	3			#	#	#	0	0	1
#	#	#	1			#	#	#	1	0	0
#	#	#	1			#	#	#	1	0	0

Codificación estadística para variables categóricas

La codificación estadística para variables categóricas, tiene su fundamento matemático en el Teorema del Límite Central (Dawson-Saunders & Trapp, 1997), el cual indica que si $S_n$es la suma de n variables aleatorias independientes y de varianza no nula pero finita, entonces los promedios de la función de distribución de S_n se aproxima a una distribución normal. La secuencia de valores que se van a probar para determinar los códigos va a depender del generador de números pseudo aleatorios. Para llevar a cabo la asignación de valores a las categorías se debe generar una extensa secuencia de números pseudo aleatorios.

En el cálculo del promedio es necesario especificar cuántas muestras se deben efectuar, con el fin de asegurar que las medias muestrales se aproximen a una normal. Existe un criterio que plantea que debe ser alrededor de 30, aunque nunca nadie ha probado que sea suficiente (Toby Mordkoff, 2000). Este método realiza la prueba de Chi Cuadrada, que consiste en dividir el espacio de observaciones en rangos. El número de rangos típicos es 10 (deciles), aunque se pueden elegir rangos de cualquier cantidad. Para satisfacer el criterio de normalidad de las variables cuya suma origina la distribución Chi-Cuadrada, se estipula que haya 5 o más observaciones en cada intervalo, aunque se pude adicionar un factor de seguridad que permitirá garantizar

una mejor aproximación a la normal. Teniendo en cuenta lo anterior, el procedimiento que permite asegurar una aproximación a una distribución normal consiste en el producto de:

a. El número de objetos por medias.
b. El número de intervalos.
c. El valor 5 para aproximar valores normales a cada una de las variables en cada uno de los intervalos.
d. El Factor de seguridad para garantizar la aproximación normal y se incorpora con el fin de evitar realizar la prueba de Chi-Cuadrada para comprobar la Gausianidad en cada caso. Este factor adiciona un número de pruebas de las que se suponen necesarias para lograr que los promedios se distribuyan normalmente.

Para seleccionar los códigos es necesario establecer cuál es la medida de bondad de un código. En nuestro caso usaremos el coeficiente de correlación de Pearson (r) para determinar la relación linealmente entre dos variables A y B en caso de que se quiera encontrar relaciones de mayor orden se tendría que usar otro método. Se explica de manera detallada el coeficiente de correlación de Pearson en la sección 3.2.

En resumen, se asigna valores pseudo aleatorias a las variables categóricas y calculamos la correlación de Pearson, un número de veces que garantice la aproximación normal y los valores resultantes para las categorías es aquel conjunto de código que maximiza el valor promedio de las correlaciones.

Conversión de fecha gregoriana a fecha juliana

Si se cuenta con un conjunto de datos que contiene fechas como comúnmente suele suceder, se puede obtener el valor en fecha juliana el cual es un valor numérico. Este valor es el número de días y fracción transcurridos desde el mediodía del 1ero de enero del año 4713 a. C, es utilizado principalmente por astrónomos.

Debido a que la fecha que constituye el punto de partida para el cálculo de la fecha juliana es de hace mucho tiempo, los números pueden ser demasiado grandes y engorrosos, por lo que ha surgido algunas modificaciones que determinar el punto de partida con una fecha más recientes. Existen varias fórmulas para convertir la fecha del calendario gregoriana al número del día juliana, en este caso se utilizará la Ecuación 3.4, que es válida para todas las fechas del calendario gregoriano proléptico después del 23 de noviembre de 4713 ac o desde el 1 de enero de 4713 hacen el calendario juliano proléptico.

JDN = (1461 × (Y + 4800 + (M - 14) / 12)) / 4 + (367 × (M - 2 - 12 × ((M - 14) / 12))) / 12 - (3 × ((Y + 4900 + (M - 14) / 12) / 100)) / 4 + D – 32075

$$JDN = 1461 * A + B - C + D - 32075 \qquad \text{(Ec. 3.4)}$$

$$A = \frac{\left(Y+4800+\frac{(M-14)}{12}\right)}{4}$$

$$B= \frac{\left(367*\left(M\text{-}2\text{-}12*\ \left(\frac{(M\text{-}14)}{12}\right)\right)\right)}{12}$$

$$C= \frac{\left(3*\left(\frac{\left(Y+4900+\frac{(M\text{-}14)}{12}\right)}{100}\right)\right)}{4}$$

Donde la Y representa el año, la M el mes y la D, el día en el calendario gregoriano

Capítulo IV.

Casos de estudio

4.1. Introducción

Este capítulo describe diferentes aspectos para poder desarrollar los casos de estudios. Primero se seleccionarán los conjuntos de datos con las que se procederá a realizar las comparaciones, así como los estudios realizados con estos conjuntos de datos para ver los mejores resultados obtenidos. Teniendo en cuenta las investigaciones se van a definir los métodos y las métricas para medir el rendimiento y en función de estas establecer las comparaciones. También se obtendrá la metodología, y se mostrarán los detalles de la implementación de los algoritmos. Además, se concluirá con los resultados obtenidos usando los conjuntos de datos preprocesados usando la metodología.

4.2. Conjuntos de datos

En nuestro análisis se usaron 4 bases de datos obtenidas del repositorio de aprendizaje de máquina UCI. Los conjuntos de datos cuentan entre 10 y 20 atributos incluyendo categóricos y numéricos, las muestras son desde 155 hasta 48842 tuplas.

- El conjunto de datos de *Census Income (CI)*[1] fue extraído de la base de datos de la oficina de censo de EE. UU. y corresponden al censo de 1994. La tarea es predecir si una persona tiene un salario mayor de 50 mil dólares al año.

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names

- El conjunto de datos de Hepatitis[2] está relacionado con la enfermedad que le da nombre y la tarea es identificar si los pacientes están vivos o no, la distribución de clases es *DIE* (20.6%) y *LIVE* (79.4%). Aproximadamente en el conjunto de datos el 48.30% incluye valores faltantes.
- El conjunto de datos de *Breast Cancer Wisconsin*[3] fue recopilado por la Universidad de Wisconsin. Las muestras fueron obtenidas periódicamente como el Dr. Wolberg informó sus casos clínicos, el período en que fueron tomadas comprende desde enero de 1989 a noviembre de 1991. La tarea es clasificar si el cáncer es benigno o maligno basándose en 10 atributos y la distribución de la clase para benigno es de 65.5% y para maligno de 34.5%.
- El conjunto de datos de *Car Evaluation*[4] se derivó de un modelo de decisión jerárquico simple para la demostración de DEX (*Decision EXpert*). La tarea es predecir el nivel de aceptación del cliente por un auto, las clases son inaceptable(*unacc*), aceptable(*acc*), bueno (*good*) y muy bueno (*v-good*). La distribución de clases es para *unacc* (70.023%), *acc* (22.222%), *good* (3.993%) y *v-good* (3.762%).

En la Tabla 4.1 se muestra un resumen de las características de estos conjuntos de datos antes mencionados.

[2] https://archive.ics.uci.edu/ml/machine-learning-databases/hepatitis/hepatitis.names

[3] https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names

[4] https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.names

Tabla 4.1 Resumen de características de los conjuntos de datos seleccionados.

Conjunto de datos	Atributos			Instancias	Valores Perdidos	Área
	Categóricos	Numérico	Total			
Census Income (CI)	9	6	15	48842	4262	Social
Hepatitis	14	6	20	155	167	Vida
Breast Cancer Wisconsin	2	8	10	569	16	Vida
Car Evaluation	7	0	7	1728	0	N/A

4.3. Análisis de resultados obtenidos sobre los conjuntos de datos

En la literatura podemos encontrar gran número de investigaciones que realizan análisis sobre los conjuntos de datos que van a ser objeto de estudio para demostrar la eficiencia de la metodología. En esta sección mencionaremos algunas de las investigaciones y algunos de los algoritmos, métricas y resultados obtenidos.

Census Income

En la descripción de la base de datos *Census Icome*[5] que se encuentra en el repositorio UCI[6] se puede encontrar los resultados obtenidos al usar diferentes algoritmos sobre este conjunto de datos. En la Tabla 4.2 se muestra los algoritmos que alcanzaron mayor exactitud luego de eliminar los valores perdidos y usando una división original de conjunto de datos en Entrenamiento/Prueba.

[5] https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.names
[6] https://archive.ics.uci.edu/ml/index.php

Tabla 4.2 Resumen de resultados sobre el conjunto de datos *Census Icome*.

Algoritmos	Exactitud
C4.5	84.46 (+/- 0.30)
Naive-Bayes	83.88 (+/- 0.30)
NBTree	85.90 (+/- 0.28)

Hepatitis

En (Pushpalatha & Pandya, 2014) se realiza un análisis comparativo de diferentes modelos que usan los datos relacionados con Hepatitis que se encuentran en el repositorio UCI y mencionan investigaciones como la de (Roslina & Noraziah, 2010), (Sathyadevi, 2011) entre otra y concluyen que los mejores resultados son obtenidos por (Ba-Alwi & Hintaya, 2013) usando un *Naives Bayes* logran una exactitud de 96.52%. En otras investigaciones se logran resultados de 96.25% de exactitud usando Máquinas de Soporte Vectorial y otros resultados más bajos usando árboles de decisión.

Breast Cancer Wisconsin

(Salama, Abdelhalim, & Zeid, 2012) hacen un estudio de los diferentes conjuntos de datos que vienen asociados al *Breast Cancer Wisconsin*. En una primera parte de su estudio realizan una comparación de los resultados de diferentes investigaciones usando diferentes algoritmos sobre las base de datos que se trabajará y concluyen (Aruna, Rajagopalan, & Nandakishore, 2011) que se logra una exactitud de 96.84% usando una Máquina de Soporte Vectorial (SVM) con un *Kernel* RBF. También en

(Zafiropoulos & Maglogiannis, 2006) se utiliza una SVM obteniendo una exactitud de 96.91% . En la parte final de la investigación mejoran esta exactitud a 97.568% usando Análisis de Componentes Principales y combinando J48 y un Red Neuronal Multicapas luego de realizar una selección de atributos.

4.4. Descripción de los métodos

En esta sección se describen los métodos de aprendizaje de máquina que serán usados para establecer una comparación entre los resultados con las bases de datos seleccionadas luego de ser preprocesadas y resultados obtenidos en los artículos recomendados en las fuentes en donde se obtuvieron los datos. Para realizar esta selección se tuvieron en cuenta los algoritmos que logran buenos resultados en los artículos revisados y otros se seleccionaron teniendo en cuenta los resultados alcanzando en investigaciones de la misma área. También cabe resaltar que se encuentran entre las técnicas de aprendizaje de máquinas más utilizadas en la actualidad.

Regresión Logística (LR)

La Regresión Logística es una técnica multivariada que nos permite estudiar la relación entre una variable dependiente, y otras variables denominadas usualmente independientes. Los modelos de regresión van desde modelos lineales a no lineales y paramétricos a no paramétricos. El caso de la regresión logística clasifica en los modelos no lineales de regresión y generalmente se aplica cuando hay una variable dependiente binaria; este caso se conoce como Regresión Logística Binaria. El objetivo de esta técnica es modelar como influyen las variables regresoras en la probabilidad de ocurrencia de un suceso particular.

Tabla 4.3 Hiperparámetros definidos para usar la regresión logística.

Hiperparámetro	Valor	Hiperparámetro	Valor
C	1.0	*n_jobs*	1
class_weight	*None*	*penalty*	l2
dual	*False*	*random_state*	*None*
fit_intercept	*True*	*solver*	*liblinear*
intercept_scaling	1	*tol*	0.0001
max_iter	100	*verbose*	0
multi_class	*ovr*	*warm_start*	*False*

Arboles de Decisión (DT)

Son un método de aprendizaje supervisado no paramétrico utilizado para la clasificación y la regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo mediante el aprendizaje de reglas simples de decisión inferidas a partir de las características de los datos y lo logra dividiendo repetidamente los datos de acuerdo con un criterio, dando como resultado una estructura con forma de árbol. Los árboles de decisión no son modelos de caja negra, y son simples de entender e interpretar. Además, pueden ser visualizados. Esta ventaja hace que sean ampliamente utilizados en muchos dominios de aplicaciones. Estos algoritmos son capaces de usar tanto datos categóricos como numéricos, lo cual es también una ventaja frente a otros algoritmos.

Tabla 4.4 Hiperparámetros definidos para usar el árbol de decisión.

Hiperparámetro	Valor	Hiperparámetro	Valor
class_weight	*None*	*min_samples_leaf*	1
criterion	*gini*	*min_samples_split*	2
max_depth	*None*	*min_weight_fraction_leaf*	0.0
max_features	*None*	*presort*	*False*
max_leaf_nodes	*None*	*random_state*	*None*
min_impurity_decrease	0.0	*splitter*	*best*
min_impurity_split	*None*		

Redes Neuronales Artificiales (ANN)

Las ANN son los modelos de inteligencia artificial más usados y son algoritmos de aprendizaje supervisados. Estas son un conjunto de neuronas con una arquitectura especifica que viene determinada por cómo están relacionadas estas neuronas en las diferentes capas, donde una neurona es una unidad de cálculo que intenta modelar el comportamiento de una neurona "natural". Las Redes Neuronales al igual que otros modelos no generan una relación paramétrica entre las variables dependientes e independientes, pero sí establecen una relación entre los datos de entrada y de salida que la hace capaz de aproximar cualquier función continua con cierto nivel de precisión. La ANN es conocido como un modelo de “caja negra” debido a la incapacidad de identificar las relaciones de las variables de manera explícita.

Tabla 4.5 Hiperparámetros definidos para usar la Red Neuronal.

Hiperparámetro	Valor	Hiperparámetro	Valor
activation	*relu*	*epsilon*	1e-08
alpha	0.0001	*hidden_layer_sizes*	(100,)
batch_size	*auto*	*learning_rate*	*constant*
beta_1	0.9	*learning_rate_init*	0.001
beta_2	0.999	*max_iter*	200
early_stopping	*False*	*momentum*	0.9
nesterovs_momentum	*True*	*power_t*	0.5
random_state	*None*	*shuffle*	*True*
solver	*adam*	*tol*	0.0001
validation_fraction	0.1	*verbose*	*False*
warm_start	*False*		

Máquinas de Soporte Vectorial (SVM)

Estos modelos son implementaciones basadas en la idea de la teoría del aprendizaje estadístico. Algorítmicamente, las máquinas de soporte vectorial construyen límites de separación óptimos entre los conjuntos de datos, resolviendo un problema de optimización cuadrática restringida (Schölkopf & Smola, 2002). La máquina de soporte vectorial (SVM) es un término para un conjunto de métodos de aprendizaje supervisados relacionados que analizan datos y reconocen patrones, utilizados para la clasificación y el análisis de regresión.

En la clasificación la diferencia entre SVM y otros clasificadores es dividir el espacio de decisión de una manera que minimice el riesgo de clasificación. Mediante el uso

de diferentes funciones del *kernel*, se pueden incluir diversos grados de no linealidad y flexibilidad en el modelo. Debido a que pueden derivarse de ideas estadísticas avanzadas, y los límites en el error de generalización se pueden calcular para ellos, las máquinas de soporte vectorial han recibido considerable interés de investigación en los últimos años.

Tabla 4.6 Hiperparámetros definidos para usar la SVM.

Hiperparámetro	Valor	Hiperparámetro	Valor
C	1.0	*kernel*	*rbf*
cache_size	200	*max_iter*	-1
class_weight	*None*	*probability*	*False*
coef0	0.0	*random_state*	*None*
decision_function_shape	*ovr*	*shrinking*	*True*
degree	3	*tol*	0.001
gamma	*auto*	*verbose*	*False*

4.5. Medidas para medir el rendimiento

Para realizar un análisis del comportamiento de los algoritmos mencionados en la sección 4.3 con los datos preprocesados, se realizará validación cruzada de 10 particiones con el fin de demostrar que los resultados de las medidas son independientes de la partición de los datos.

Las medidas usadas para medir los resultados de los algoritmos serán la Precisión, Exactitud y Exhaustividad. A continuación, se explican algunos términos necesarios para comprender la descripción de las medidas seleccionadas.

Verdaderos Positivos (VP): Son los valores positivos correctamente predichos.

Falsos Positivos (FP): Son los valores positivos incorrectamente predichos.

Verdaderos Negativos (VN): Son los valores negativos correctamente predichos.

Falsos Negativos (FN): Son los valores negativos incorrectamente predichos.

Exactitud (Accuracy):

Esta medida es muy intuitiva, y mide la relación entre las predicciones realizadas correctas y el total de las observaciones originales.

$$\text{Exactitud} = \frac{\text{VP}+\text{VN}}{\text{VP}+\text{FP}+\text{VN}+\text{FN}}$$

Precisión (Precision):

La exactitud es la relación entre los valores positivos correctamente predichos y los valores positivos totales predichos.

$$\text{Precisión} = \frac{\text{VP}}{\text{VP}+\text{FP}}$$

Exhaustividad (Recall):

La exhaustividad es la relación entre los valores positivos correctamente predichos y todos los valores en la clase.

$$\text{Exhaustividad} = \frac{\text{VP}}{\text{VP+FN}}$$

4.6. Implementación de los algoritmos.

Los algoritmos fueron implementados usando Python 3.5, haciendo uso de sus múltiples librerías creadas con esta finalidad como son: *Numpy*, *Pandas*, *Scipy* y *Scikit-learn* la cual es una eficiente herramienta para la minería de datos y el análisis de datos.

Con el fin de mejorar la usabilidad y facilidad para realizar la tarea de preprocesamiento de los datos usando los algoritmos que se han propuesto en esta investigación, se implementó una interfaz gráfica que permitirá una interacción más intuitiva para realizar las distintas tareas haciendo uso de PyQt 4.5 el cual es un *binding*[7] de la biblioteca gráfica Qt para el lenguaje Python en el Anexo 5 se muestran más detalles. El código fuente de la aplicación se encuentra en el GitHub[8] y en el Anexo 6 se encuentra un manual para su instalación y uso.

4.7. Descripción de la metodología

En esa sección se detallan los pasos propuesto para llevar a cabo la tarea de preprocesamiento de los datos. Este flujo fue utilizado para obtener las bases de datos que luego de aplicarle las técnicas de aprendizaje de máquina propuestas en la Sección 4.4, van a ser comparados con los resultados obtenidos con los datos sin preprocesar

[7] Es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita.
[8] https://github.com/cantoniohdez2/PREPRODAT

y los resultados presentados en los artículos recomendados por la fuente. El propósito es dejar claro que luego de aplicar estas técnicas de preprocesamiento los resultados son superiores. A continuación de detallas los pasos de la metodología propuesta:

1. Se comprueba si existen variables en fecha gregoriana y en el caso de tener este tipo de valores se procede a convertirla a juliana.
2. Se verifica si se cuenta con variables categóricas y entonces se aplica la codificación estadística de variables categóricas en cada caso.
3. Se analiza si existen variables numéricas con valores perdidos. Si fuese el caso se obtiene una base de datos de eliminar las tuplas (BD1) con valores perdidos y se obtiene otra base de datos con los valores perdidos imputados (DB2) usando el Spline Natural. En caso de no haber valores perdidos se continua en 5.
4. Se calcula la entropía para ambas bases de datos (Base de dato BD1 y Base de dato BD2). Si la diferencia de información de BD2 y BD1 es menor que un umbral definido, entonces se continúa trabajando con BD1 para evitar el riesgo de introducir ruido en los datos. En casos de que la diferencia sea mayor que el umbral establecido se continúa trabajando con BD2 en la que los valores perdidos fueron imputados.
5. Se procede escalando los datos entre un rango definido que puede depender del algoritmo de aprendizaje de máquina que se vaya a utilizar, para uso genérico se seleccionó el rango [0,1].
6. Se estabilizan las variables.
7. Se realiza la detección de valores atípicos usando el algoritmo *iForest* y se procede a eliminar aquellas tuplas que contienen los valores que fueron detectados.

8. Se calcula el coeficiente de correlación de Pearson entre todas las variables para ver la relación lineal entre las mismas. Se define un umbral (0.85) que permite seleccionar aquellas variables que se eliminarán por su fuerte relación.
9. Se realiza el Análisis de Componentes Principales (PCA) para reducir la dimensionalidad (Opcional).
10. Se realiza el Muestreo Aleatorio Simple sin reposición para reducir la numerosidad (Opcional).

4.8. Resultados

Para el análisis de resultados se comparan las medidas propuestas en la Sección 4.5 luego de aplicar los algoritmos de aprendizaje de máquina seleccionados, a los diferentes conjuntos de datos que se mencionan en la Sección 4.1. Esto en una primera etapa se realiza en los datos sin preprocesar, para establecer una comparación con los datos preprocesados. También se realizará una comparación con los mejores resultados teniendo en cuenta las investigaciones que recomienda la fuente de donde se obtuvieron los datos y otras referencias consultadas.

Resultados en los conjuntos de datos sin preprocesar.

En esta sección se muestran los resultados de las medias de las precisiones al aplicar validación cruzada de 10 particiones usando los algoritmos que se describen en la Sección 4.4 (para diferentes conjuntos de datos). Cabe resaltar que se aplicaron algunas técnicas de preprocesamiento en estos datos las cuales se describen en cada uno de los casos.

Census Income.

Esta base de datos contaba con valores perdidos y con variables categórica. En el caso de los valores perdidos se eliminaron y a las variables categóricas les fue asignado un valor numérico para poder aplicar algunas técnicas de aprendizaje de máquinas propuestas. Los resultados se muestran en la Tabla 4.7.

Tabla 4.7 Resultados obtenidos con los datos de *Census Income* sin preprocesar.

Algoritmos	**Exactitud**
Regresión Logística	78% (+/- 0.4)
Árbol de Decisión	78%(+/-0.6)
Redes Neuronales	72%(+/-0.2)
Máquina de Soporte de Vectorial	74% (+/- 0.5)
Random Forest	83% (+/- 0.4)
Gradient Boosting	85% (+/- 0.2)

Hepatitis.

El conjunto de datos correspondiente a Hepatitis contaba con valores perdidos tanto en variables categóricas como numéricas. En el caso de las variables numéricas fue sustituido por el promedio de los sí existentes y las tuplas con valores categóricos faltantes fueron eliminadas debido a que eran una cantidad significativa que fuera afectar la cantidad de información. En la siguiente tabla se muestran los resultados:

Tabla 4.8 Resultados obtenidos con los datos de Hepatitis sin preprocesar.

Algoritmos	Exactitud
Regresión Logística	69% (+/- 2.0)
Árbol de Decisión	57% (+/- 1.6)
Redes Neuronales	57% (+/- 3.0)
Máquina de Soporte de Vectorial	58% (+/- 1.4)
Random Forest	62% (+/- 1.4)
Gradient Boosting	59% (+/- 2.6)

Breast Cancer Wisconsin.

El conjunto de datos tenía valores perdidos por lo que fue necesario eliminar las tuplas correspondientes que como se describe en la Sección 4.2 eran 16. En la Tabla 4.9 se muestran los resultados:

Tabla 4.9 Resultados obtenidos con los datos de *Breast Cancer Wisconsin* sin preprocesar.

Algoritmos	Exactitud
Regresión Logística	65% (+/- 0.1)
Árbol de Decisión	95% (+/- 0.3)
Redes Neuronales	53% (+/- 2.9)
Máquina de Soporte de Vectorial	66% (+/- 0.3)
Random Forest	96% (+/- 0.5)
Gradient Boosting	96% (+/- 0.4)

Car Evaluation.

Este conjunto de datos no contaba con valores perdidos, pero las variables categóricas fueron remplazadas por valores numéricos. Los resultados se muestran a continuación:

Tabla 4.10 Resultados obtenidos con los datos de *Car Evaluation* sin preprocesar.

Algoritmos	Exactitud
Regresión Logística	77% (+/- 1.8)
Árbol de Decisión	88% (+/- 1.3)
Redes Neuronales	88% (+/- 1.9)
Máquina de Soporte de Vectorial	90% (+/- 0.5)
Random Forest	86% (+/- 1.4)
Gradient Boosting	90% (+/- 1.1)

Resultados de los conjuntos de datos preprocesados.

Census Income

En el Anexo 1 se describe detalladamente la metodología. Aplicando los métodos seleccionados sobre los datos preprocesada los resultados obtenidos se muestran en la Tabla 4.11 para las métricas seleccionadas usando validación cruzada de 10 particiones son:

Tabla 4.11 Resultados obtenidos con los datos de *Census Income* preprocesados.

Algoritmos	Exactitud
Regresión Logística	83% (+/- 0.4)
Árbol de Decisión	80%(+/-0.6)
Redes Neuronales	85%(+/-0.2)
Máquina de Soporte de Vectorial	83% (+/- 0.5)
Random Forest	82% (+/- 0.4)
Gradient Boosting	87% (+/- 0.2)

Hepatitis

Los resultados obtenidos luego de aplicar la metodología que se detalla en el Anexo 2 sobre el conjunto de datos de Hepatitis, se muestran en la Tabla 4.12.

Tabla 4.12 Resultados obtenidos con los datos de Hepatitis preprocesados.

Algoritmos	Exactitud
Regresión Logística	96% (+/- 0.7)
Árbol de Decisión	95%(+/- 0.5)
Redes Neuronales	96% (+/- 0.7)
Máquina de Soporte de Vectorial	97% (+/- 0.5)
Random Forest	97% (+/- 0.6)
Gradient Boosting	97% (+/- 0.6)

Breast Cancer Wisconsin

En el Anexo 3 se describe detalladamente la metodología. En el análisis realizado luego de preprocesar los datos usando los algoritmos propuesto en la Sección 4.4 se obtuvieron los resultados que se muestran en la Tabla 4.13.

Tabla 4.13 Resultados obtenidos con los datos de Breast Cancer Wisconsin preprocesados.

Algoritmos	Exactitud
Regresión Logística	95% (+/- 0.8)
Árbol de Decisión	93% (+/- 0.6)
Redes Neuronales	96% (+/- 0.6)
Máquina de Soporte de Vectorial	97% (+/- 0.4)
Random Forest	96% (+/- 0.4)
Gradient Boosting	96% (+/- 0.6)

Car Evaluation.

Luego que los datos fueron preprocesados usando la metodología propuesta y que se describe detalladamente en el Anexo 4, los resultados obtenidos se encuentran en la Tabla 4.14 que se muestra a continuación:

Tabla 4.14 Resultados obtenidos con los datos de *Car Evaluation* preprocesados.

Algoritmos	Exactitud
Regresión Logística	86% (+/- 0.4)
Árbol de Decisión	99% (+/- 0.2)

Redes Neuronales	92% (+/- 0.2)
Máquina de Soporte de Vectorial	90% (+/- 0.3)
Random Forest	98% (+/- 0.3)
Gradient Boosting	99% (+/- 0.2)

Análisis de los resultados.

En esta sección se comparan los resultados de los datos sin preprocesar (NP), los datos preprocesados (SP) y los mejores resultados (en general) obtenidos en las investigaciones revisadas (G).

Census Income

Como se puede notar el algoritmo *Gradient Boosting* (GB) obtuvo una exactitud de 87% (SP) comparable con el mejor resultado alcanzado por el *NBTree* (G) de un 85.9% que es el mejor resultado registrado en las investigaciones revisadas. El mejor resultado registrado en los datos sin preprocesar (NP) es de 85%.

Cuando se analiza el Área Bajo la Curva (ó AUC) al aplicar GB se obtienen los resultados que se muestran a continuación en la Figura 4.1. En la Figura 4.2 se muestran una gráfica que resume los resultados obtenidos al usar los diferentes métodos sobre esta base de datos.

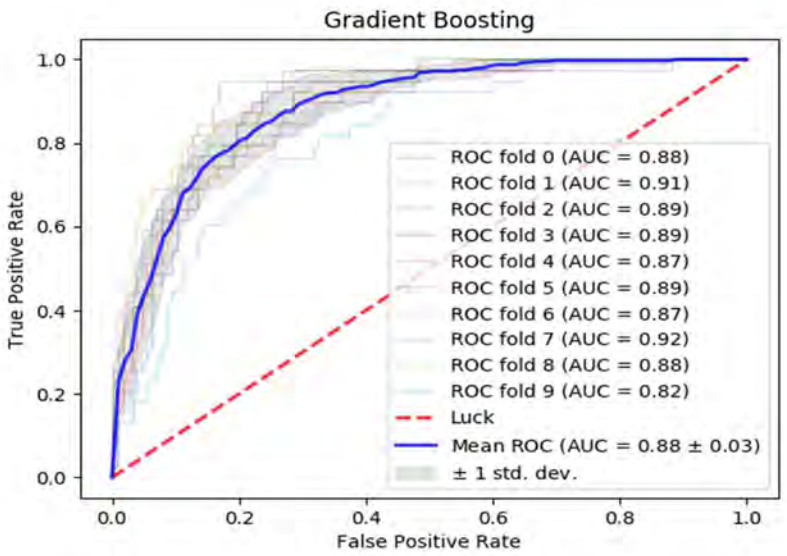


Figura 4.1 Gráfica del AUC usando validación cruzada de 10 particiones en los datos preprocesados (GB).

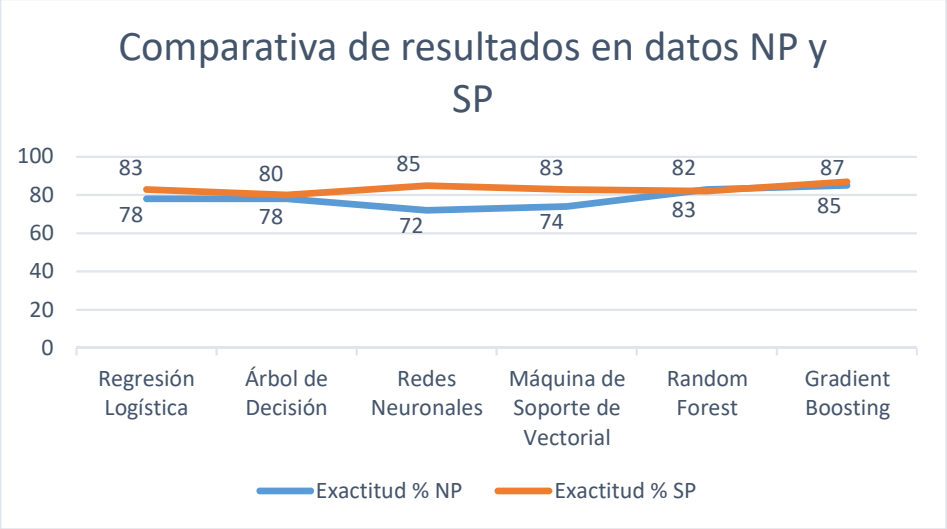


Figura 4.2 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Census Income*.

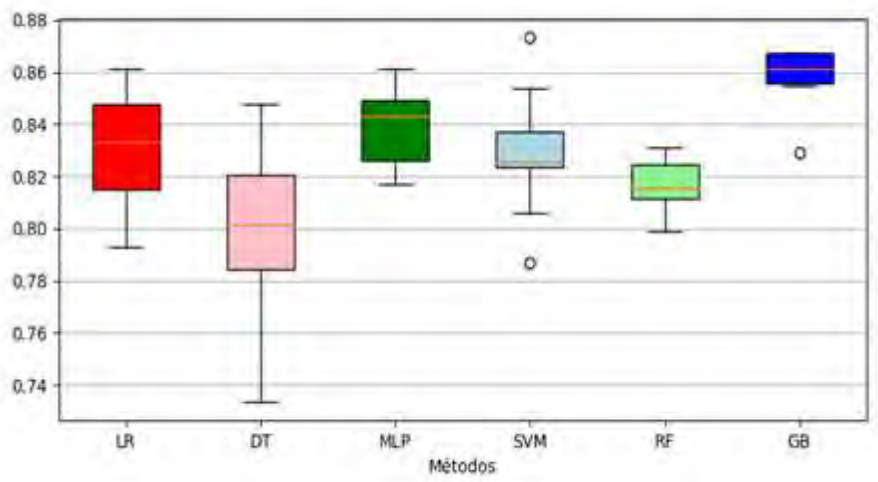


Figura 4.3 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de Census Income.

Hepatitis

El mejor resultado del que se tenía constancia (G) es del algoritmo *Naives Bayes* que logró una exactitud de 96.52%. Éste ha sido mejorado (SP) por varios de los algoritmos mencionados anteriormente: SVM (97%), RF (97%) y GB (97%).

El siguiente resultado (SP) no es bueno y fue obtenido usando Regresión Logística con un 69% de exactitud.

Al realizar un análisis del AUC para tener una mejor certeza en los resultados se puede notar que la Máquina de Soporte Vectorial alcanzan una media del AUC de 70% como se muestra en la Figura 4.4. En la Figura 4.5 se muestran una gráfica que resume los resultados obtenidos al usar los diferentes métodos sobre esta base de datos.

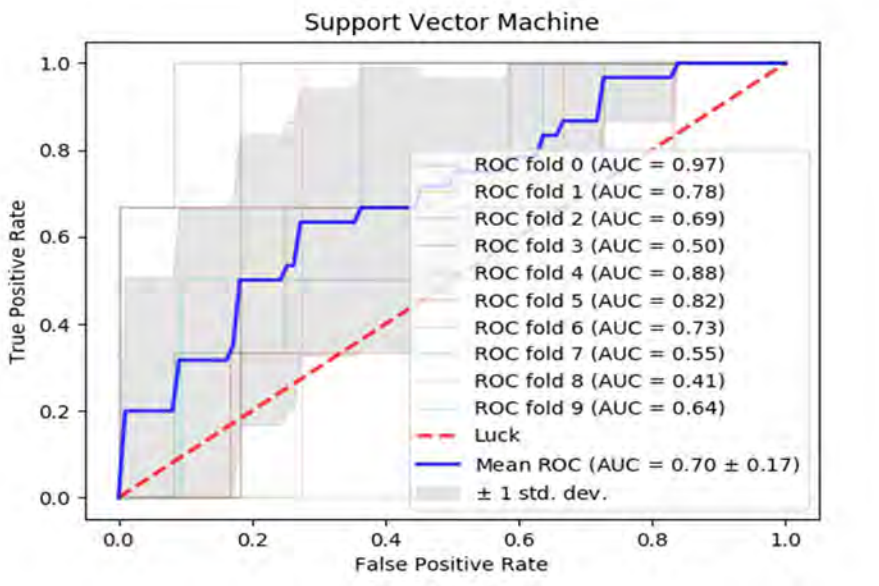


Figura 4.4 Gráfica del AUC usando validación cruzada de 10 particiones en los datos preprocesados (SVM).

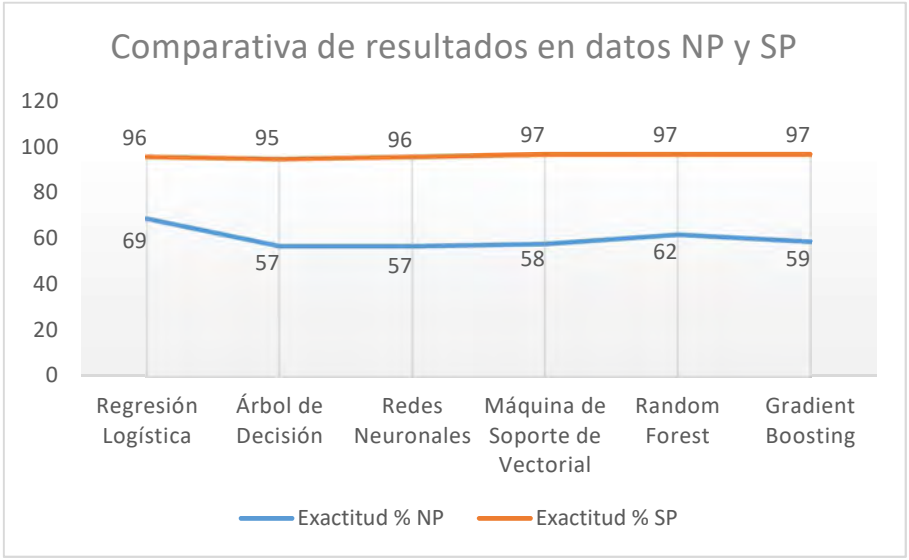


Figura 4.5 Gráfica de resultados en los datos NP y SP para el conjunto de datos de Hepatitis.

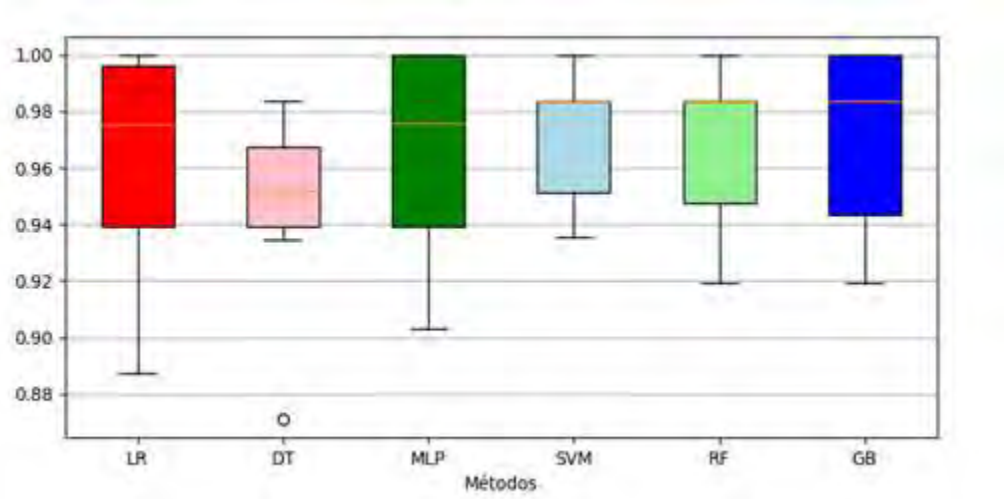


Figura 4.6 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de Hepatitis.

Breast Cancer Wisconsin

El mejor resultado (G) (combinación de J48 y un Red Neuronal Multicapas) es de 97.56%. En nuestro análisis el mejor resultado de los datos sin preprocesar varía desde 53% logrado por la MLP hasta un 96% que logra RF y GB. En el caso de los datos preprocesados los resultados se encuentran entre 93% con el valor más bajo de exactitud logrado por los DT y un 97% logrado por las SVM. Este resultado de las SVM es comparable con el mejor resultado reportado pues este análisis se realiza usando validación cruzada y en varias etapas alcanza picos de 98.4%.

También se realizó un análisis del Área Bajo la Curva (AUC) y los mejores resultados obtenidos se muestran en la Figura 4.7 y Figura 4.8. En la Figura 4.9 se muestran una gráfica que resume los resultados obtenidos al usar los diferentes métodos sobre esta base de datos.

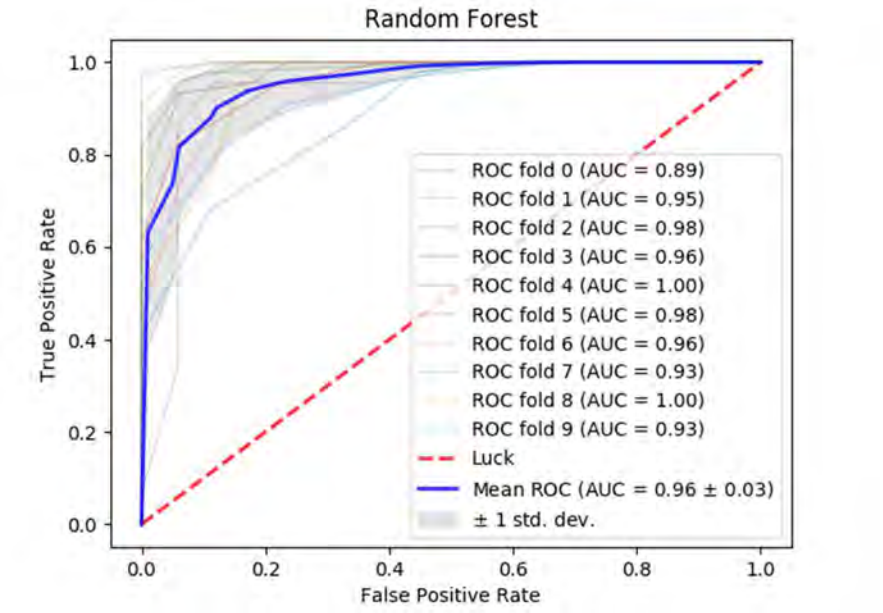


Figura 4.7 Gráfica del Área Bajo la Curva (AUC) usando validación cruzada de 10 particiones en los datos preprocesados (RF).

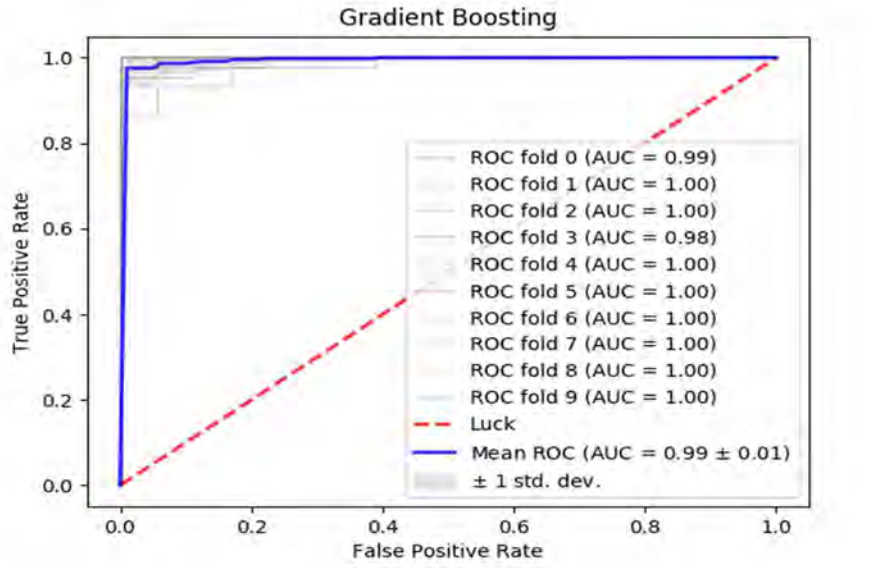


Figura 4.8 Gráfica del Área Bajo la Curva (AUC) usando validación cruzada de 10 particiones en los datos preprocesados (GB).

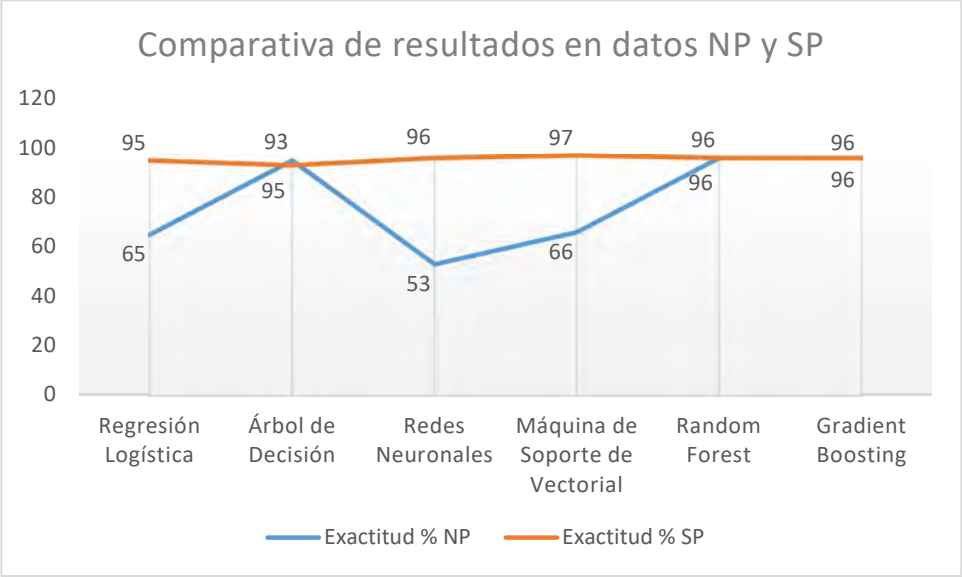


Figura 4.9 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Breast Cancer Wisconsin*.

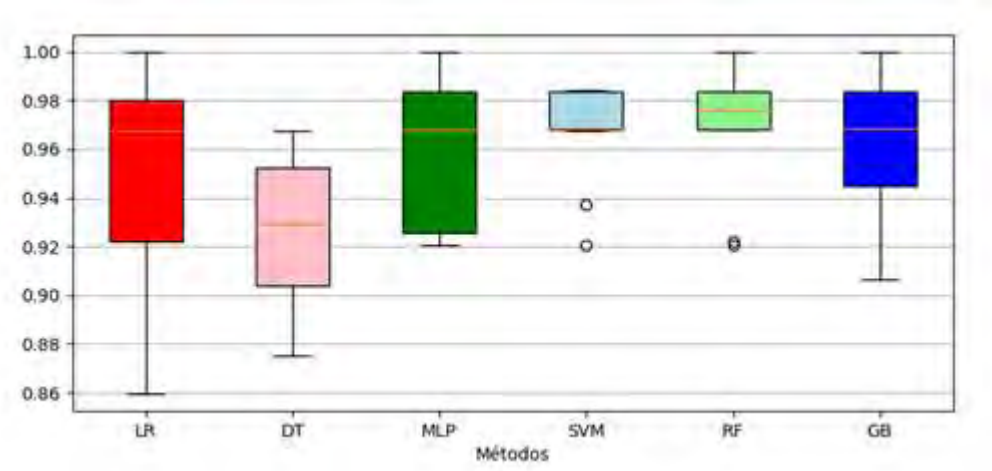


Figura 4.10 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de *Breast Cancer Wisconsin*.

Car Evaluation.

Los mejores resultados (NP) obtenidos se logran por la Máquina de Soporte Vectorial (SVM) y GB con un 90% y el menor resultado es de un 77% obtenido por la Regresión Logística. Esto resultados son superados (SP) significativamente (aproximadamente 10%) en todos los casos antes mencionados excepto en la SVM en la cual los resultados son igualados.

En la Figura 4.12 se muestran una gráfica que resume los resultados obtenidos al usar los diferentes métodos sobre esta base de datos.

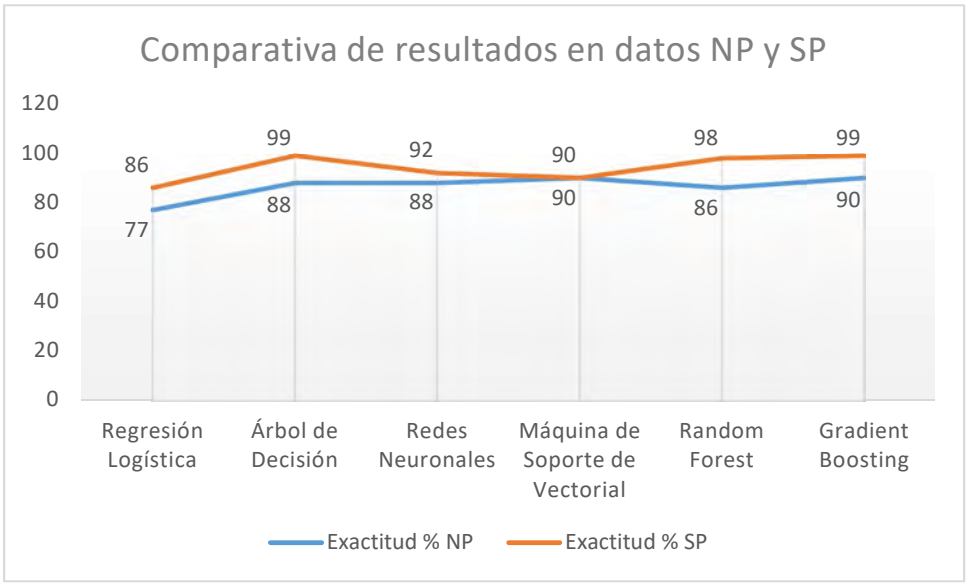


Figura 4.11 Gráfica de resultados en los datos NP y SP para el conjunto de datos de *Car Evaluation.*

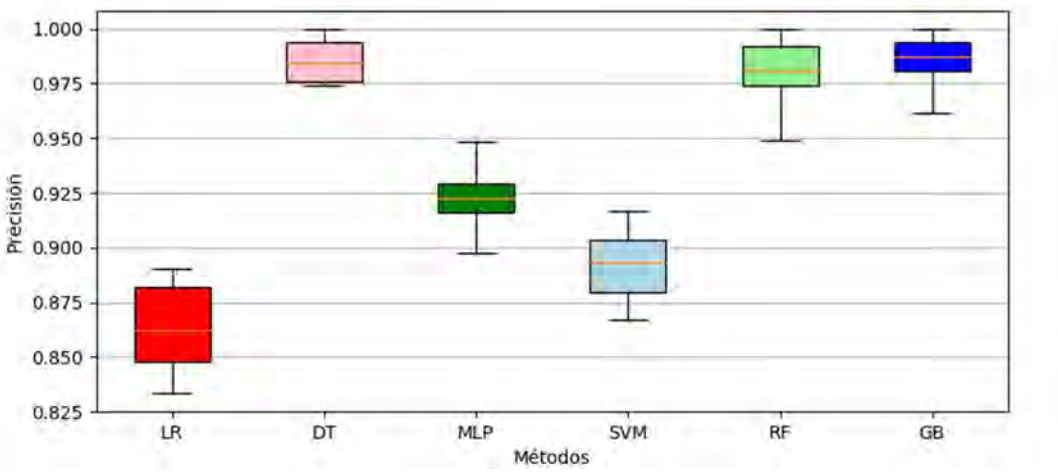


Figura 4.12 Diagrama de caja de los resultados en los datos SP para el conjunto de datos de *Breast Cancer Wisconsin*.

Capítulo V.

Conclusiones

1. La descripción de las principales etapas y tareas que se deben tener en cuenta para el preprocesamiento de los datos permitieron obtener los conocimientos teóricos necesarios para su estudio.
2. La identificación adecuada de los métodos para realizar las tareas de preprocesamiento permitirá garantizar los resultados cuando se realice alguna de las tareas necesarias para el preprocesamiento.
3. El realizar pruebas con los métodos seleccionados permitió elaborar una propuesta para la metodología, de manera que se tuvieran en cuenta todos los problemas que pueden tener los datos y su mejor solución.
4. Se validó la metodología elaborada estableciendo una comparación entre los mejores resultados de exactitud reportados en las investigaciones revisadas, los resultados de exactitud obtenidos en los datos sin preprocesar y la exactitud en los datos preprocesados.

Bibliografía

Aggarwal, C. C. (2015). *Data Mining* (1st ed.). Springer International Publishing. https://doi.org/10.1007/978-3-319-14142-8

Aruna, S., Rajagopalan, S. P., & Nandakishore, L. V. (2011). Knowledge based analysis of various statistical tools in detecting breast cancer. *Computer Science & Information Technology*, *2*, 37–45.

Ba-Alwi, F. M., & Hintaya, H. M. (2013). Comparative study for analysis the prognostic in hepatitis data: data mining approach. *Spinal Cord*, *11*, 12.

Bakar, Z. A., Mohemad, R., Ahmad, A., & Deris, M. M. (2006). A Comparative Study for Outlier Detection Techniques in Data Mining. *IEEE Conference on Cybernetics and Intelligent Systems*, 1–6. https://doi.org/10.1109/ICCIS.2006.252287

Domingues, R., Filippone, M., Michiardi, P., & Zouaoui, J. (2018). A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, *74*, 406–421. https://doi.org/10.1016/J.PATCOG.2017.09.037

Enders, C. K. (2010). *Applied missing data analysis. Methodology in the social sciences*. Guilford Press. https://doi.org/10.1017/CBO9781107415324.004

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining. Springer* (Vol. 72). Springer. https://doi.org/10.1007/978-3-319-10247-4

González, B. V., & Mora, A. H. (2018). *Adquisición de datos, preprocesamiento y reconocimiento de patrones para matrices de sensores SAW-UV*. Universidad Nacional Autónoma de Mexico.

González, P. C. (2018). *Sistema de Redes Neuronales para la predicción de ozono en la CDMX y el área metropolitana.* UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques* (Third Edit). Elsevier.

Hawkins, D. M. (1980). Identification of Outliers. *London: Chapman and Hall*,

Volumen 11. https://doi.org/10.1007/978-94-015-3994-4

Johnson, R., & Wichern, D. (1992). Applied multivariate statistical methods. *Prentice Hall, Englewood Cliffs, NJ.*

Kaur, K., & Garg, A. (2016). Comparative Study of Outlier Detection Algorithms. *International Journal of Computer Applications.*

Kuri-Morales, Á. (2015). Natural Splines.

Kuri-Morales, Á., & Galaviz Casas, J. (2002). *Algoritmos Genéticos* (Segunda Ed). Sociedad Mexicana de Inteligencia Artificial.

Li, D., Deogun, J., Spaulding, W., & Shuart, B. (2004). Towards Missing Data Imputation: A Study of Fuzzy K-means Clustering Method. In *Rough sets and current trends in computing* (pp. 573–579). Springer. https://doi.org/10.1007/978-3-540-25929-9_70

Liu, F. T., & Ting, K. M. (2012). Isolation-based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *V*, 1–44.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Oommen, T., Misra, D., Twarakavi, N. K. C., Prakash, A., Sahoo, B., & Bandopadhyay, S. (2008). An objective analysis of support vector machine based classification for remote sensing. *Mathematical Geosciences*, *40*(4), 409–424.

Pareja, M. S. L. (2017). *Análisis estadístico del estilo de vida en pacientes diabéticos mediante un modelo de Regresión Logística Ordinal.* Universidad Nacional Autónoma de México.

Pushpalatha, S., & Pandya, J. (2014). Data model comparison for Hepatitis diagnosis, *9359*(7), 138–141.

Roslina, A. H., & Noraziah, A. (2010). Prediction of hepatitis prognosis using Support Vector Machines and Wrapper Method. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on* (Vol. 5, pp. 2209–2211).

Roth, P. L. (1994). Missing data: A conceptual review for applied psychologists. *Personnel Psychology*, *47*(3), 537–560. https://doi.org/10.1111/J.1744-6570.1994.TB01736.X

Sadawarti, K. M. H., Ieee, M., & Kalra, G. S. (2014). Comparative Analysis of Outlier Detection Techniques. *International Journal of Computer Applications*, *97*(8), 12–21. https://doi.org/10.5120/17026-7318

Salama, G. I., Abdelhalim, M., & Zeid, M. A. (2012). Breast cancer diagnosis on three different datasets using multi-classifiers. *Breast Cancer (WDBC)*, *32*(569), 2.

Sathyadevi, G. (2011). Application of CART algorithm in hepatitis disease diagnosis. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on* (pp. 1283–1287).

Schafer, J. L., & Graham, J. W. (2002). Multiple imputation: Our view of the state of the art. *Psychological Methods*, *7*(2), 147–177.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels : support vector machines, regularization, optimization, and beyond*. MIT Press. Retrieved from https://dl.acm.org/citation.cfm?id=559923

Subramanlan K, R. E. (2011). Outlier detection: A review. *International Journal of Advances in Embedeed System Research*, *1*, 55–57.

Toby Mordkoff, J. (2000). The Assumption(s) of Normality.

Van Der Maaten, L., Postma, E., & den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, *10*, 66–71.

Zafiropoulos, E., & Maglogiannis, I. (2006). A Support Vector Machine Approach to Breast Cancer Diagnosis and Prognosis. *IFIP International Federation for Information Processing*, *204*, 500–507. https://doi.org/10.1007/0-387-34224-9_58

Anexos

Anexo 1.

Descripción de las etapas de preprocesamiento del conjunto de datos de Censo de EEUU.

Información

La tarea de predicción en este problema es determinar si una persona gana más de 50,000 (>50K) en un año. A continuación, se describen los datos con los que se cuenta:

Descripción de las Variables:

Variable	**Valores**
age	continuous
workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt	continuous
education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num	continuous
marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces

relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
sex	Female, Male
capital-gain	continuous
capital-loss	continuous
hours-per-week	continuous
native-country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

Descripción de la Clase:

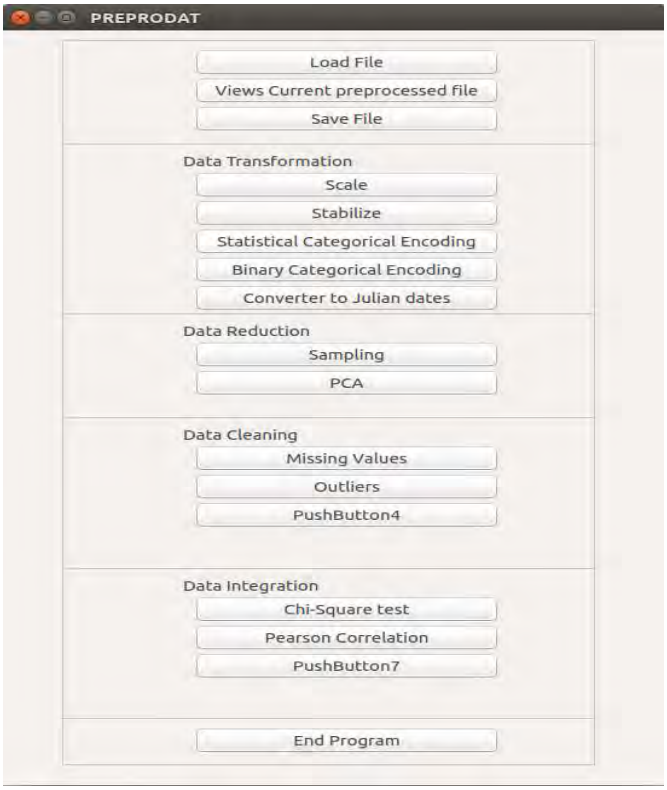
Class	**Número**	**Número (%)**	**Descripción**
>50K,	11688	23.93%	Personas con salarios mayores a 50 mil al año.
<=50K	37154	76.07%	Personas con salarios menores o iguales a 50 mil al año.

Los resultados que se obtienen usando diferentes algoritmos con este dataset se muestran en la siguiente tabla. Todos estos resultados se logran luego de eliminar los valores desconocidos y dividiendo el dataset original en train/test:

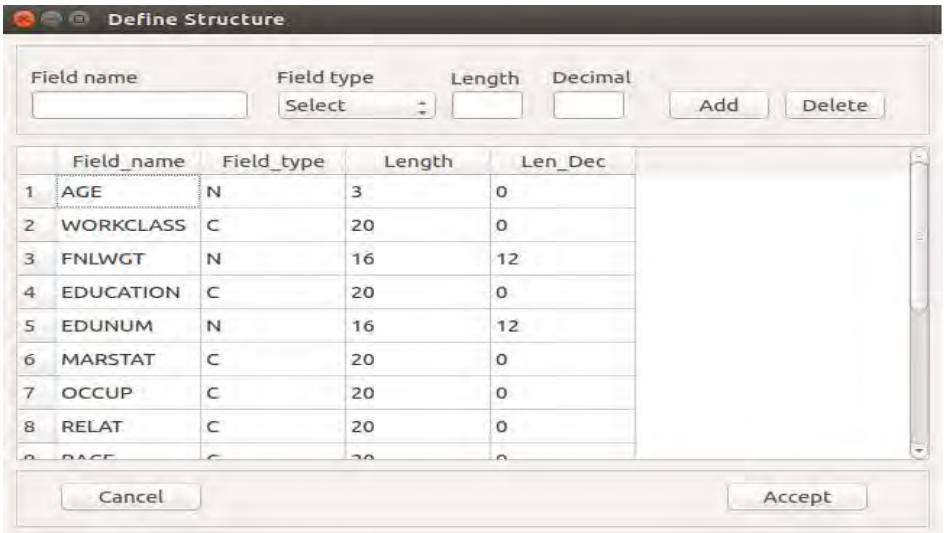
Algoritmos	Error
C4.5	15.54
C4.5-auto	14.46
C4.5 rules	14.94
Voted ID3 (0.6)	15.64
Voted ID3 (0.8)	16.47
T2	16.84
1R	19.54
NBTree	14.10
CN2	16.00
HOODG	14.82
FSS Naive Bayes	14.05
IDTM(Decision table)	14.46
Naive-Bayes	16.12
Nearest- neighbor(1)	21.41
Nearest- neighbor(3)	20.35
OC1	15.04
Pebls	Crashed. Unknown why (bounds WERE increased)

Descripción del preprocesamiento realizado:

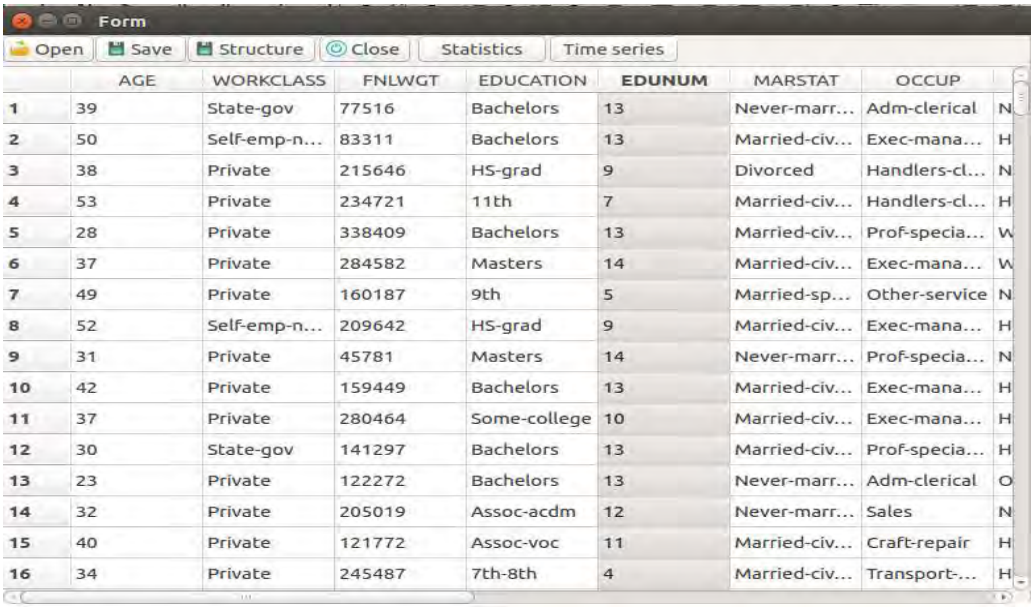
1. Se procede a cargar el dataset y definir la estructura de los datos (**Load File**).



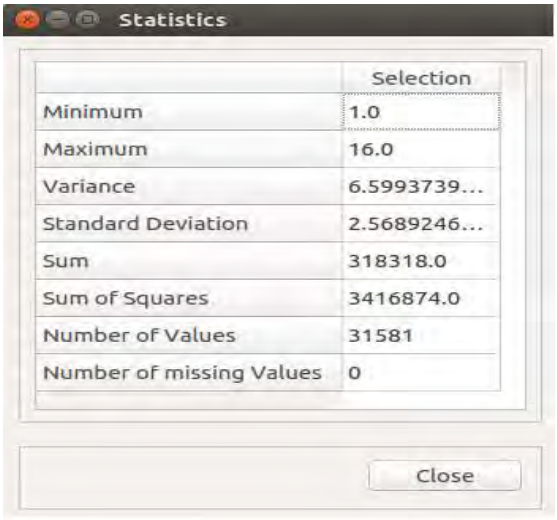
2. Se define la estructura de los datos a preprocesar para conocer qué manejo hacerle dependiendo si son numéricos, categóricos o fechas.



3. Se cargan los datos desde un archivo de texto delimitado por comas(.csv) o un archivo de texto delimitado por Tabuladores (.txt) que se encuentre en un directorio local (**Open**) y se puede obtener algunos valores estadísticos de los datos, para tener un mayor acercamiento a los mismos (S**tatistics**).

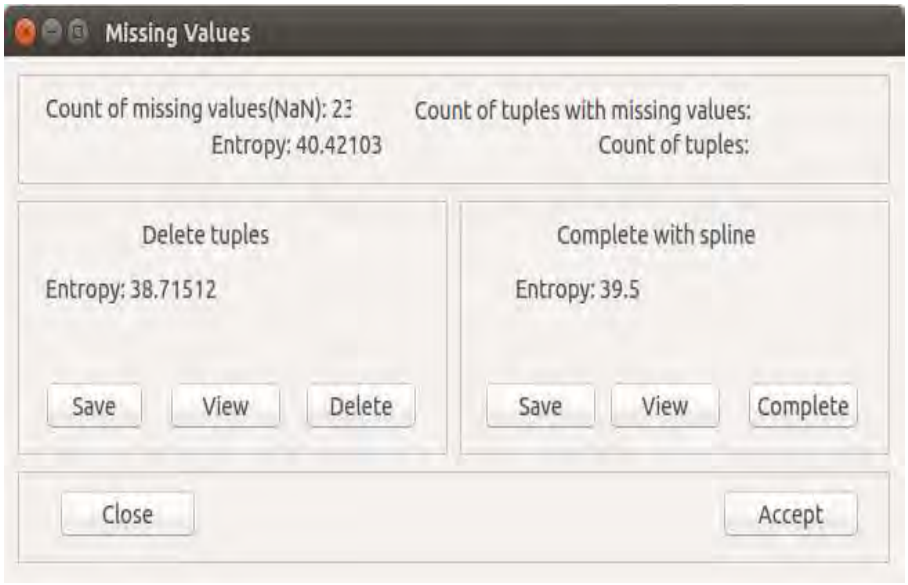


	AGE	WORKCLASS	FNLWGT	EDUCATION	EDUNUM	MARSTAT	OCCUP	
1	39	State-gov	77516	Bachelors	13	Never-marr...	Adm-clerical	N
2	50	Self-emp-n...	83311	Bachelors	13	Married-civ...	Exec-mana...	H
3	38	Private	215646	HS-grad	9	Divorced	Handlers-cl...	N
4	53	Private	234721	11th	7	Married-civ...	Handlers-cl...	H
5	28	Private	338409	Bachelors	13	Married-civ...	Prof-specia...	W
6	37	Private	284582	Masters	14	Married-civ...	Exec-mana...	W
7	49	Private	160187	9th	5	Married-sp...	Other-service	N
8	52	Self-emp-n...	209642	HS-grad	9	Married-civ...	Exec-mana...	H
9	31	Private	45781	Masters	14	Never-marr...	Prof-specia...	N
10	42	Private	159449	Bachelors	13	Married-civ...	Exec-mana...	H
11	37	Private	280464	Some-college	10	Married-civ...	Exec-mana...	H
12	30	State-gov	141297	Bachelors	13	Married-civ...	Prof-specia...	H
13	23	Private	122272	Bachelors	13	Never-marr...	Adm-clerical	O
14	32	Private	205019	Assoc-acdm	12	Never-marr...	Sales	N
15	40	Private	121772	Assoc-voc	11	Married-civ...	Craft-repair	H
16	34	Private	245487	7th-8th	4	Married-civ...	Transport-...	H



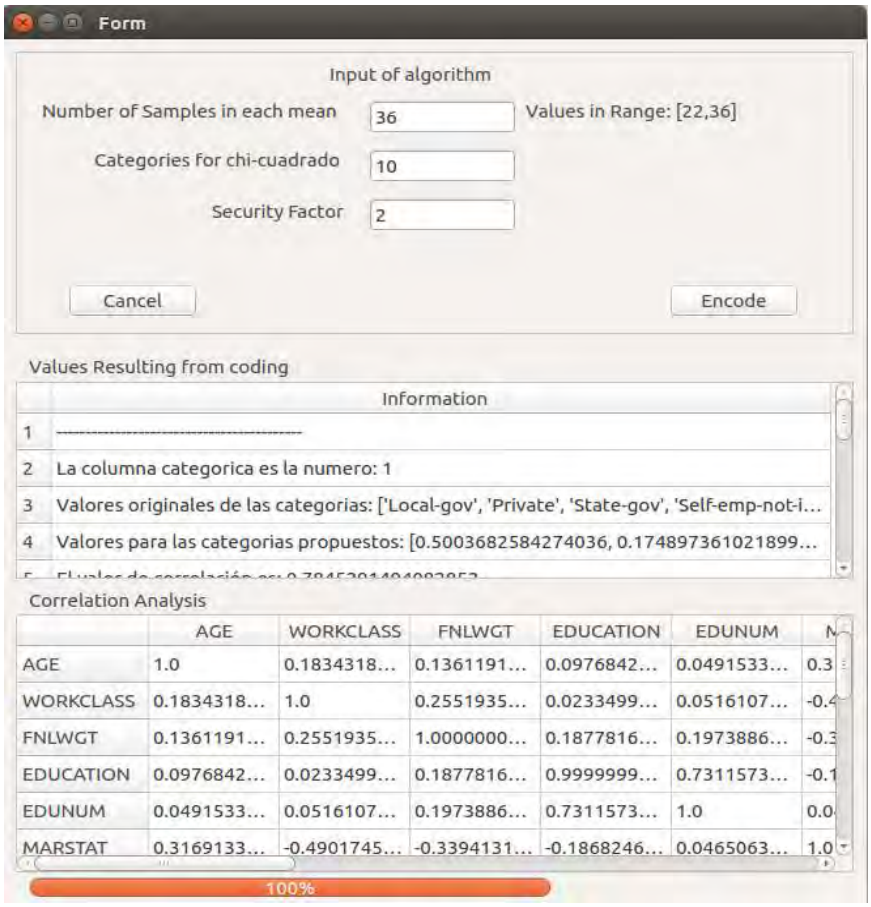
4. La cantidad de instancias es de 48,842. Por ello se procede a realizar un muestreo Aleatorio Simple sin repetición para trabajar una menor cantidad de instancias en este caso con 1,652.

5. Los datos muestreados todavía cuentan con valores perdidos por lo que se procederán a imputar (usando *splines*) estas instancias de las variables numéricas. En el caso de haber valores perdidos en las variables categóricas, cuando se procesada con la codificación estadística este algoritmo asumirá que es una categoría más (NaN) y le asignará un valor que se puede interpretar como una categoría desconocida.

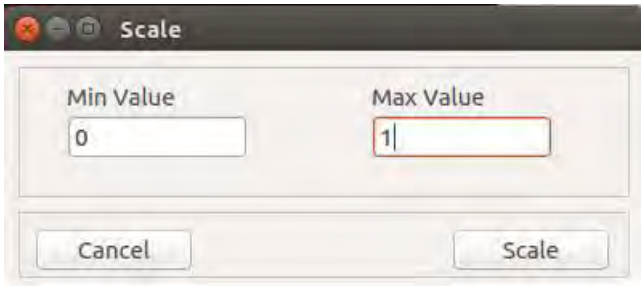


En la ventana anterior se puede notar que se realiza el cálculo de entropía tanto para la opción "Delete Tuples" como para la opción "Complete with *Spline*". Esto se debe a que, si la cantidad de información no difiere a un valor umbral definido, se puede trabajar con los datos luego de eliminar las instancias. Lo que significa que no hubo pérdida de información y así se evita introducir ruido al realizar la imputación de los valores.

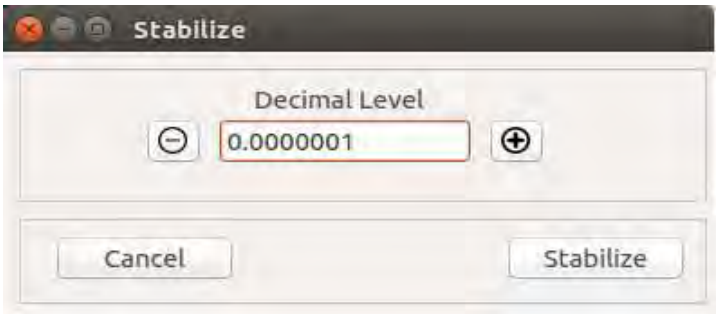
6. Ahora se procede a codificar las variables categóricas usando la opción del Menú Principal (Statistical Categorical Encoding). Con este método se obtiene un valor numérico para cada una de las categorías de las variables categóricas con un alto valor de correlación con las demás.



7. Se procede a escalar los datos entre [0, 1].



8. Esta etapa consiste en estabilizar los datos para evitar inestabilidad matemática en el momento de aplicar alguna técnica de aprendizaje de máquina.



9. En el último paso se exporta el archivo obtenido a un archivo de texto delimitado por comas (.csv)

Resultados obtenidos luego de aplicar el preprocesamiento:

Para obtener estos resultados luego de aplicar el preprocesamiento de datos, se realizó validación cruzada de 10 particiones para disminuir la incertidumbre con respecto a la exactitud de los algoritmos analizados frente al dataset.

Algoritmo	Accuracy
Regresión Logística	83% (+/- 0.4)
Árbol de Decisión	80%(+/-0.6)
Redes Neuronales	85%(+/-0.2)
Máquina de Soporte de Vectorial	83% (+/- 0.5)
Random Forest	82% (+/- 0.4)
Gradient Boosting	87% (+/- 0.2)

Anexo 2.

Descripción de las etapas de preprocesamiento del conjunto de datos de Hepatitis.

Información

La tarea de predicción en este problema es determinar si una persona Vivió o Murió.

A continuación, se describen los datos con los que se cuenta:

Descripción de las Variables:

Variable	**Valores**
AGE	10, 20, 30, 40, 50, 60, 70, 80
SEX	male, female
STEROID	no, yes
ANTIVIRALS	no, yes
FATIGUE	no, yes
MALAISE	no, yes
ANOREXIA	no, yes
LIVER BIG	no, yes
LIVER FIRM	no, yes
SPLEEN PALPABLE	no, yes
SPIDERS	no, yes
ASCITES	no, yes
VARICES	no, yes
BILIRUBIN	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
ALK PHOSPHATE	33, 80, 120, 160, 200, 250
SGOT	13, 100, 200, 300, 400, 500
ALBUMIN	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
PROTIME	10, 20, 30, 40, 50, 60, 70, 80, 90
HISTOLOGY	no, yes

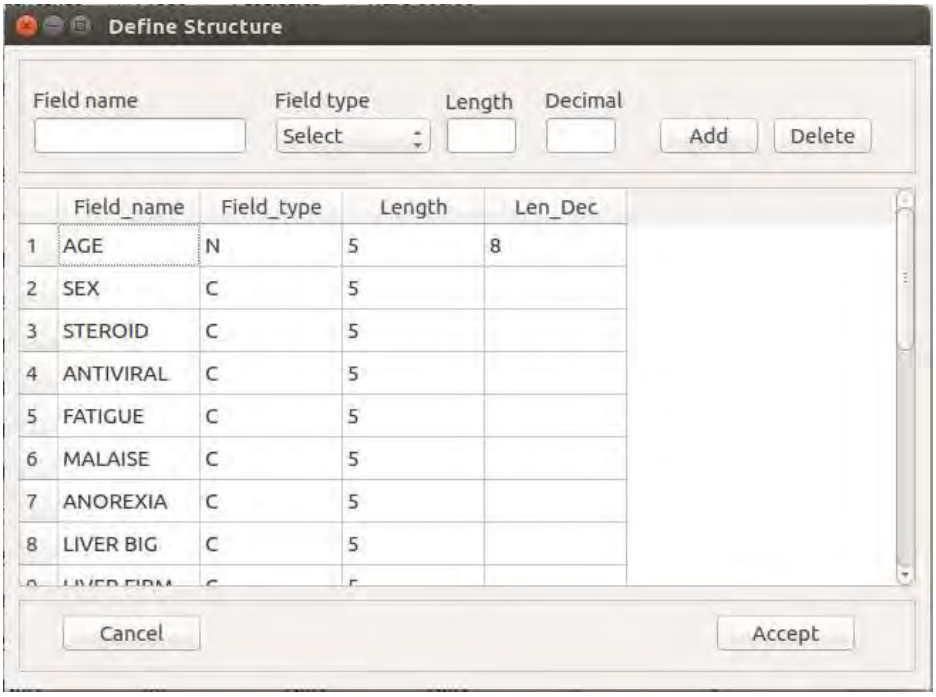
Descripción de la Clase

Class	Número	Número (%)	Descripción
DIE	32	21.8%	Si la persona murió.
LIVE	123	84.2%	Si la persona vivió.

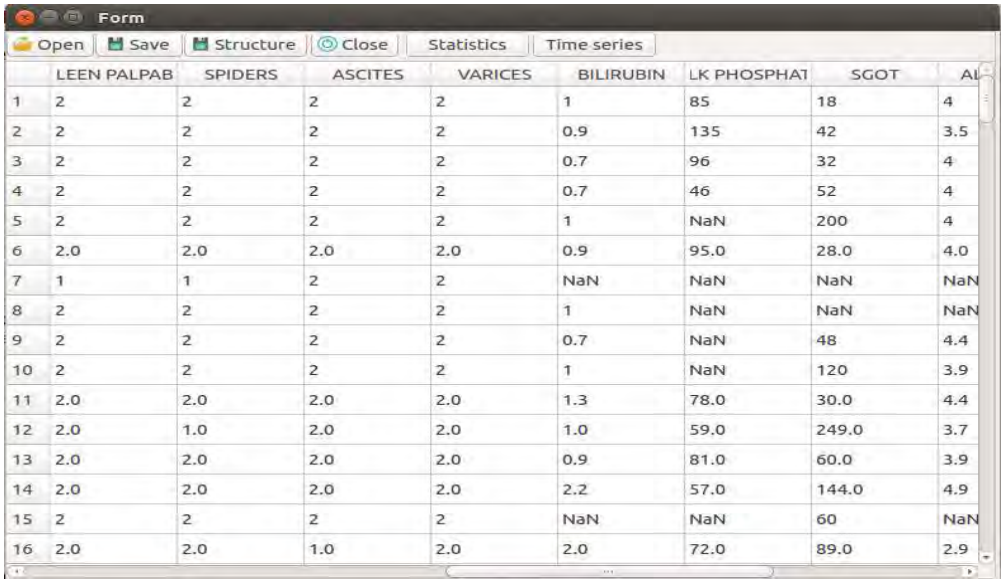
Nota: El conjunto de datos cuenta con un total de 167 valores perdidos**.**

Descripción del preprocesamiento realizado:

1. Se procede a cargar el dataset y definir la estructura de los datos, se debe tener en cuenta que la base de datos originales tenía la clase en la primera columna. Ésta fue movida hacia la última columna. (NOTA: En nuestra implementación la variable dependiente deberá encontrarse siempre en la última columna) (**Load File**).

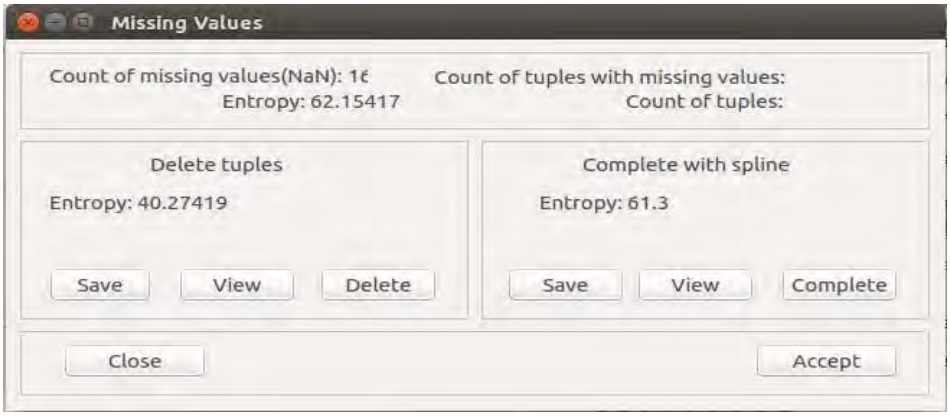


2. Se cargan los datos desde un archivo de texto delimitado por comas(.csv) o un archivo de texto delimitado por Tabuladores (.txt) que se encuentre en un directorio local (**Open**) y se puede obtener algunos valores estadísticos de los datos, para tener un mayor acercamiento a los mismos (S**tatistics**).

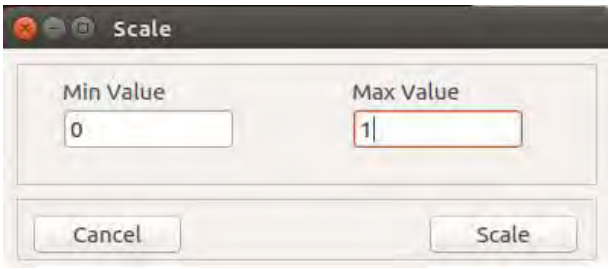


	LEEN PALPAB	SPIDERS	ASCITES	VARICES	BILIRUBIN	LK PHOSPHAT	SGOT	AL
1	2	2	2	2	1	85	18	4
2	2	2	2	2	0.9	135	42	3.5
3	2	2	2	2	0.7	96	32	4
4	2	2	2	2	0.7	46	52	4
5	2	2	2	2	1	NaN	200	4
6	2.0	2.0	2.0	2.0	0.9	95.0	28.0	4.0
7	1	1	2	2	NaN	NaN	NaN	NaN
8	2	2	2	2	1	NaN	NaN	NaN
9	2	2	2	2	0.7	NaN	48	4.4
10	2	2	2	2	1	NaN	120	3.9
11	2.0	2.0	2.0	2.0	1.3	78.0	30.0	4.4
12	2.0	1.0	2.0	2.0	1.0	59.0	249.0	3.7
13	2.0	2.0	2.0	2.0	0.9	81.0	60.0	3.9
14	2.0	2.0	2.0	2.0	2.2	57.0	144.0	4.9
15	2	2	2	2	NaN	NaN	60	NaN
16	2.0	2.0	1.0	2.0	2.0	72.0	89.0	2.9

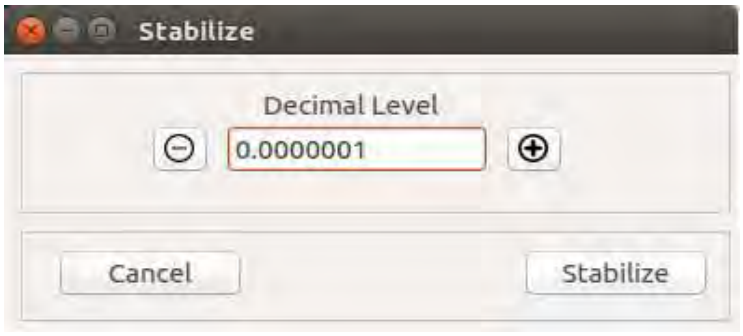
3. Los datos cuentan con valores perdidos por lo que se procederán a imputar usando *splines* estas instancias de las variables numéricas. En el caso de haber valores perdidos en las variables categóricas, cuando se proceda con la codificación estadística, este algoritmo asumirá que es una categoría más (NaN) y le asignará un valor que se puede interpretar como una categoría desconocida. Para este caso se calculó la entropía de los datos con las variables eliminadas y los datos con las variables imputadas. Como se puede notar, la diferencia de la entropía es de aproximadamente +21. Se ve, entonces, que es convenientes imputar los valores porque la pérdida de información (de no hacerse así) sería alta. Cabe resaltar que cuando los valores perdidos se encuentran en las variables categóricas este procedimiento no es aplicable.



4. Ahora se procede a escalar los datos entre [0, 1].

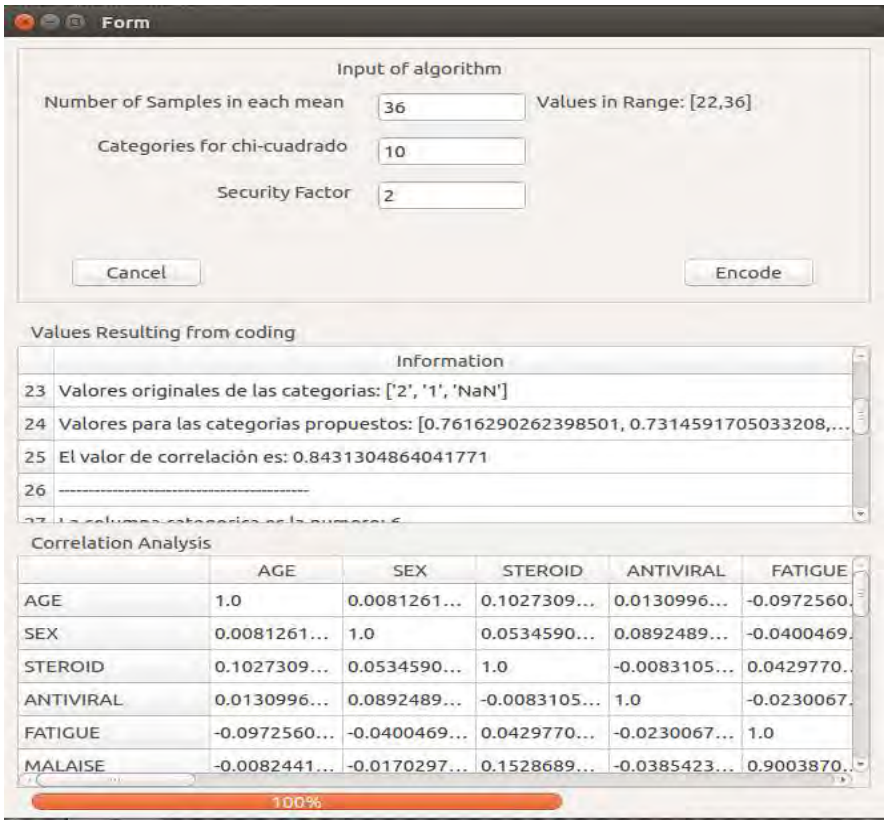


5. La siguiente etapa consiste en estabilizar los datos para evitar inestabilidad matemática en el momento de aplicar alguna técnica de aprendizaje de máquina.

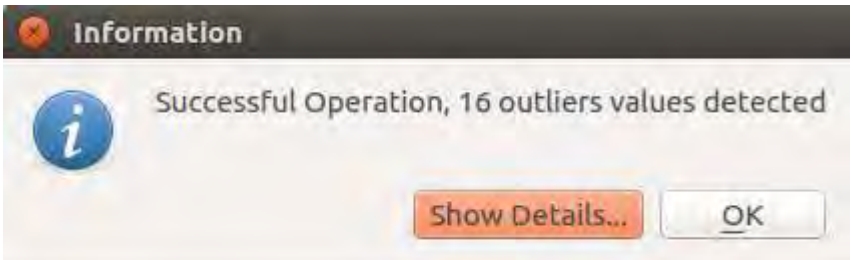


6. Ahora se procede a codificar las variables categóricas usando la opción del Menú Principal (Statistical Categorical Encoding). Con este método se obtiene un valor numérico para cada una de las categorías de las variables categóricas con

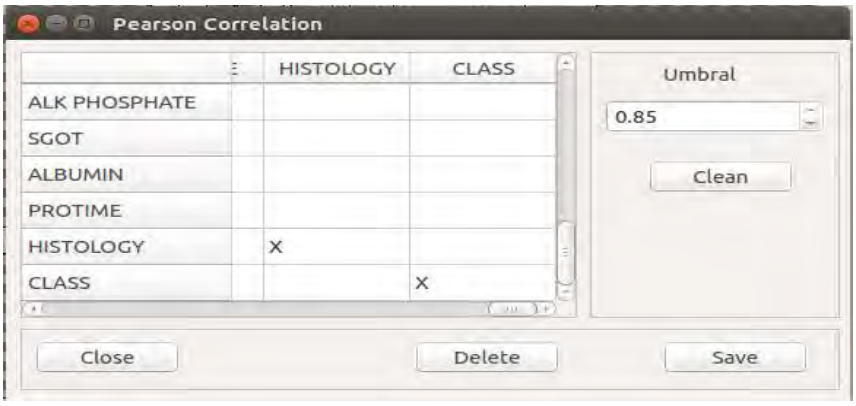
un alto valor de correlación con las demás, en el casoque existan variables categóricas con valores faltantes el algoritmo a asumir que es una categoría desconocida, le va a asignar un valor.



7. Se procedió a detectar valores atípicos y fueron detectados 16 tuplas.



8. Se realiza el análisis de correlación de Pearson. No se encontraron correlaciones mayores 0.85. Por lo que no fue necesario proceder con la eliminación de variables.

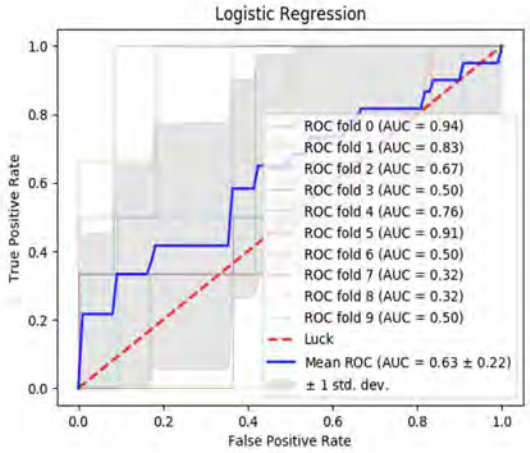


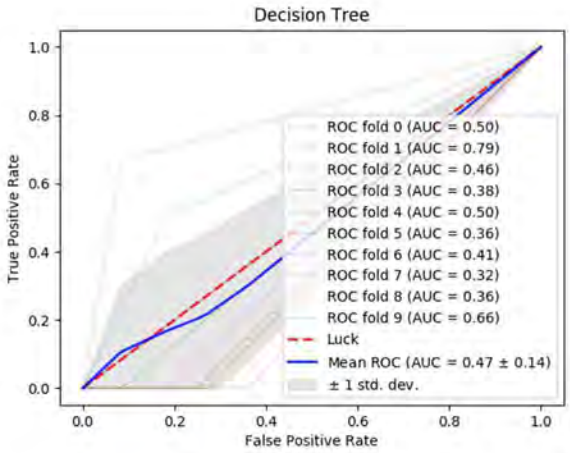
9. En el último paso, se exportó el archivo obtenido a un archivo de texto delimitado por comas (.csv)

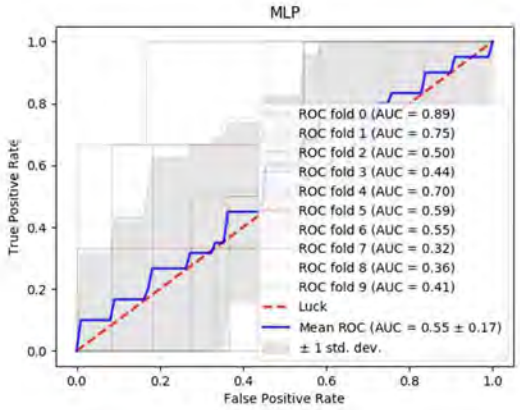
Resultados obtenidos luego de aplicar el preprocesamiento:

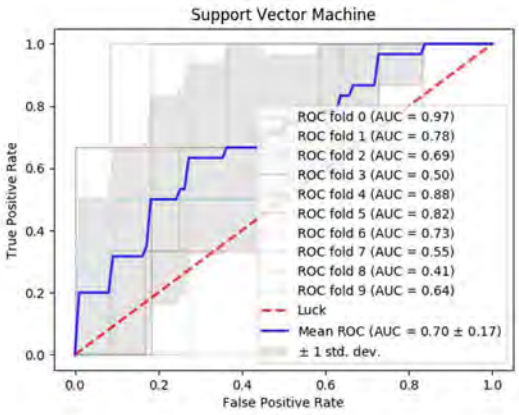
Para obtener estos resultados luego de aplicar el preprocesamiento de datos, se realizó validación cruzada de 10 particiones para disminuir la incertidumbre con respecto a la exactitud de los algoritmos analizados frente al dataset.

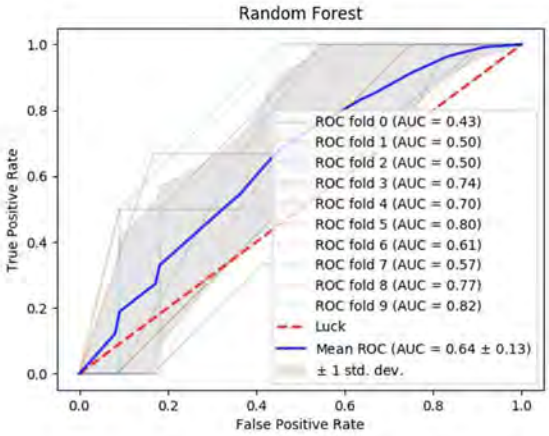
Algoritmo	Accuracy
Regresión Logística	0.96 (+/- 0.07)
Árbol de Decisión	0.95 (+/- 0.05)
Redes Neuronales	0.96 (+/- 0.07)
Máquina de Soporte de Vectorial	0.97 (+/- 0.05)
Random Forest	0.97 (+/- 0.06)
Gradient Boosting	0.97 (+/- 0.06)

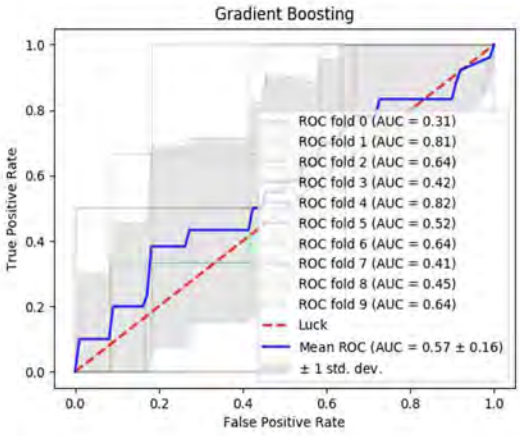

Logistic Regression
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.94)
ROC fold 1 (AUC = 0.83)
ROC fold 2 (AUC = 0.67)
ROC fold 3 (AUC = 0.50)
ROC fold 4 (AUC = 0.76)
ROC fold 5 (AUC = 0.91)
ROC fold 6 (AUC = 0.50)
ROC fold 7 (AUC = 0.32)
ROC fold 8 (AUC = 0.32)
ROC fold 9 (AUC = 0.50)
Luck
Mean ROC (AUC = 0.63 ± 0.22)
± 1 std. dev.


Decision Tree
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.50)
ROC fold 1 (AUC = 0.79)
ROC fold 2 (AUC = 0.46)
ROC fold 3 (AUC = 0.38)
ROC fold 4 (AUC = 0.50)
ROC fold 5 (AUC = 0.36)
ROC fold 6 (AUC = 0.41)
ROC fold 7 (AUC = 0.32)
ROC fold 8 (AUC = 0.36)
ROC fold 9 (AUC = 0.66)
Luck
Mean ROC (AUC = 0.47 ± 0.14)
± 1 std. dev.


MLP
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.89)
ROC fold 1 (AUC = 0.75)
ROC fold 2 (AUC = 0.50)
ROC fold 3 (AUC = 0.44)
ROC fold 4 (AUC = 0.70)
ROC fold 5 (AUC = 0.59)
ROC fold 6 (AUC = 0.55)
ROC fold 7 (AUC = 0.32)
ROC fold 8 (AUC = 0.36)
ROC fold 9 (AUC = 0.41)
Luck
Mean ROC (AUC = 0.55 ± 0.17)
± 1 std. dev.


Support Vector Machine
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.97)
ROC fold 1 (AUC = 0.78)
ROC fold 2 (AUC = 0.69)
ROC fold 3 (AUC = 0.50)
ROC fold 4 (AUC = 0.88)
ROC fold 5 (AUC = 0.82)
ROC fold 6 (AUC = 0.73)
ROC fold 7 (AUC = 0.55)
ROC fold 8 (AUC = 0.41)
ROC fold 9 (AUC = 0.64)
Luck
Mean ROC (AUC = 0.70 ± 0.17)
± 1 std. dev.


Random Forest
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.43)
ROC fold 1 (AUC = 0.50)
ROC fold 2 (AUC = 0.50)
ROC fold 3 (AUC = 0.74)
ROC fold 4 (AUC = 0.70)
ROC fold 5 (AUC = 0.80)
ROC fold 6 (AUC = 0.61)
ROC fold 7 (AUC = 0.57)
ROC fold 8 (AUC = 0.77)
ROC fold 9 (AUC = 0.82)
Luck
Mean ROC (AUC = 0.64 ± 0.13)
± 1 std. dev.


Gradient Boosting
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.31)
ROC fold 1 (AUC = 0.81)
ROC fold 2 (AUC = 0.64)
ROC fold 3 (AUC = 0.42)
ROC fold 4 (AUC = 0.82)
ROC fold 5 (AUC = 0.52)
ROC fold 6 (AUC = 0.64)
ROC fold 7 (AUC = 0.41)
ROC fold 8 (AUC = 0.45)
ROC fold 9 (AUC = 0.64)
Luck
Mean ROC (AUC = 0.57 ± 0.16)
± 1 std. dev.

Anexo 3.

Descripción de las etapas de preprocesamiento del conjunto de datos de *Breast Cancer Wisconsin*.

Información

La tarea de predicción en este problema es determinar si un cáncer es benigno o maligno. A continuación, se describen los datos con los que se cuenta:

Descripción de las Variables:

Variable	**Valores**
Sample code number	1-10
Clump Thickness	1-10
Uniformity of Cell Size	1-10
Uniformity of Cell Shape	1-10
Marginal Adhesion	1-10
Single Epithelial Cell Size	1-10
Bare Nuclei	1-10
Bland Chromatin	1-10
Normal Nucleoli	1-10
Mitoses	1-10

Descripción de la Clase

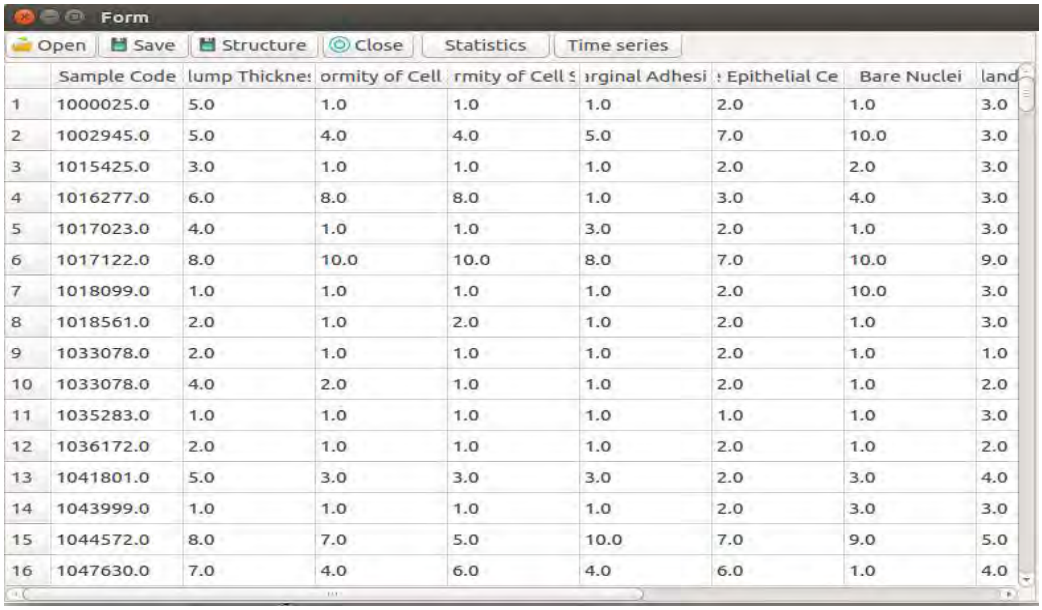
Class	**Número**	**Número (%)**
2	458	65.5%
4	241	34.5%

Nota: El conjunto de datos cuenta con un total de 16 valores perdidos.

Descripción del preprocesamiento realizado:

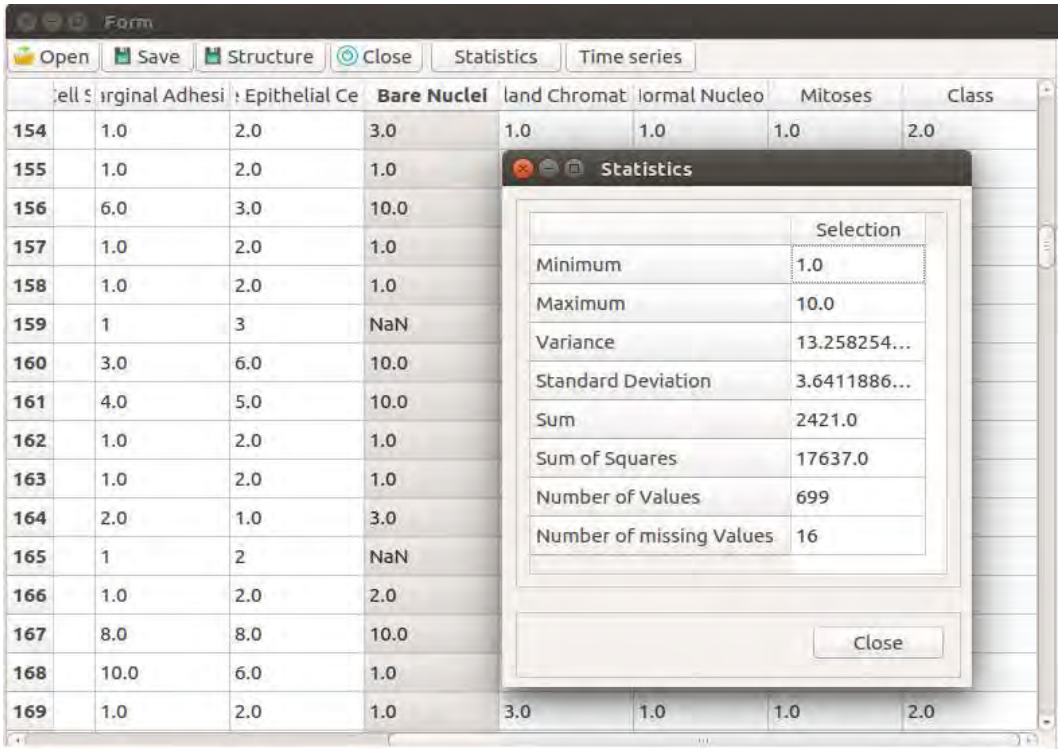
1. Se procede a cargar el dataset y definir la estructura de los datos (**Load File**)

2. Se define la estructura de los datos a preprocesar para conocer que manejo hacerle dependiendo si son numéricos, categóricos o fechas, en esta etapa para esta base de datos cabe destacar que la columna Sample Code Number fue eliminado ya que es solamente un código asociada a cada una de las muestras tomada.



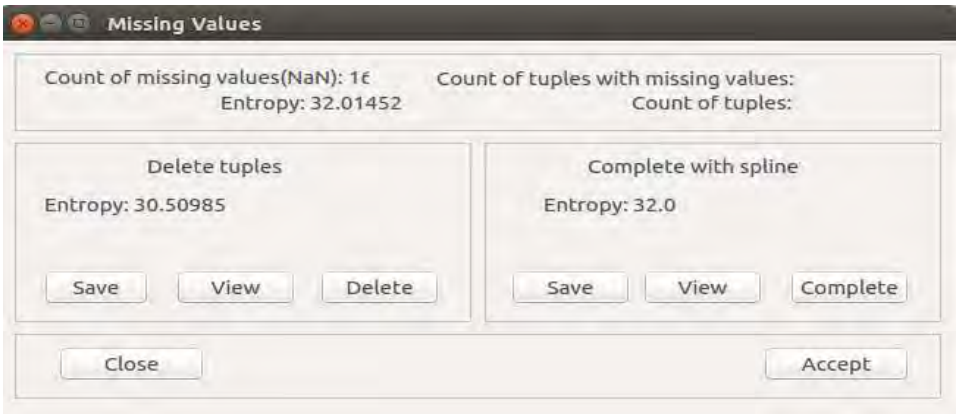
	Sample Code	lump Thickne:	ormity of Cell	rmity of Cell S	ırginal Adhesi	: Epithelial Ce	Bare Nuclei	land
1	1000025.0	5.0	1.0	1.0	1.0	2.0	1.0	3.0
2	1002945.0	5.0	4.0	4.0	5.0	7.0	10.0	3.0
3	1015425.0	3.0	1.0	1.0	1.0	2.0	2.0	3.0
4	1016277.0	6.0	8.0	8.0	1.0	3.0	4.0	3.0
5	1017023.0	4.0	1.0	1.0	3.0	2.0	1.0	3.0
6	1017122.0	8.0	10.0	10.0	8.0	7.0	10.0	9.0
7	1018099.0	1.0	1.0	1.0	1.0	2.0	10.0	3.0
8	1018561.0	2.0	1.0	2.0	1.0	2.0	1.0	3.0
9	1033078.0	2.0	1.0	1.0	1.0	2.0	1.0	1.0
10	1033078.0	4.0	2.0	1.0	1.0	2.0	1.0	2.0
11	1035283.0	1.0	1.0	1.0	1.0	1.0	1.0	3.0
12	1036172.0	2.0	1.0	1.0	1.0	2.0	1.0	2.0
13	1041801.0	5.0	3.0	3.0	3.0	2.0	3.0	4.0
14	1043999.0	1.0	1.0	1.0	1.0	2.0	3.0	3.0
15	1044572.0	8.0	7.0	5.0	10.0	7.0	9.0	5.0
16	1047630.0	7.0	4.0	6.0	4.0	6.0	1.0	4.0

3. Se cargan los datos desde un archivo de texto delimitado por comas(.csv) o un archivo de texto delimitado por Tabuladores (.txt) que se encuentre en un directorio local (**Open**) y se puede obtener algunos valores estadísticos de los datos, para tener un mayor acercamiento a los mismos (S**tatistics**).

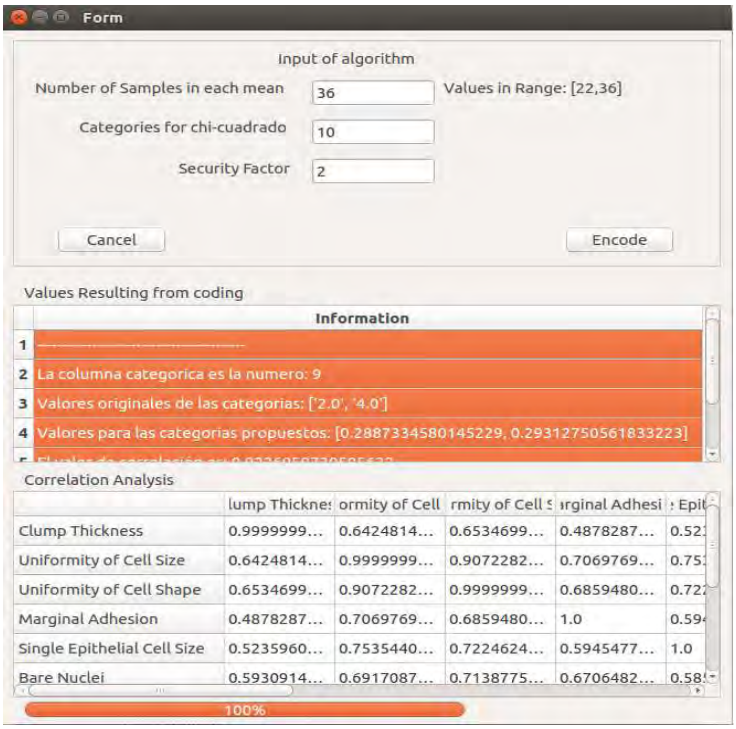


4. Los datos cuentan con valores perdidos por lo que se procederán a imputar usando *splines*. Estas instancias de las variables numéricas. En el caso de haber valores perdidos en las variables categóricas, cuando se procesada con la codificación estadística este algoritmo asumirá que es una categoría más (NaN) y le asignará un valor el cual se puede interpretar con una categoría desconocida. Para este caso se calculó la entropía de los datos con las variables eliminadas y

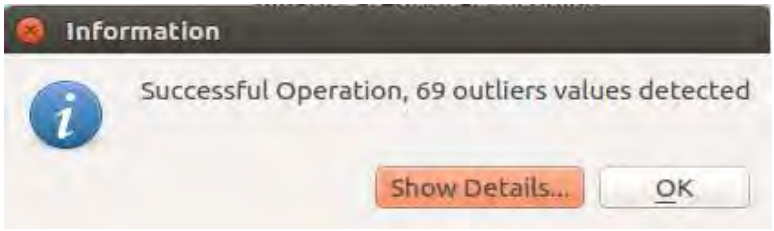
los datos con las variables imputadas y se decidió eliminar los valores faltantes porque la diferencia de información era mínima.



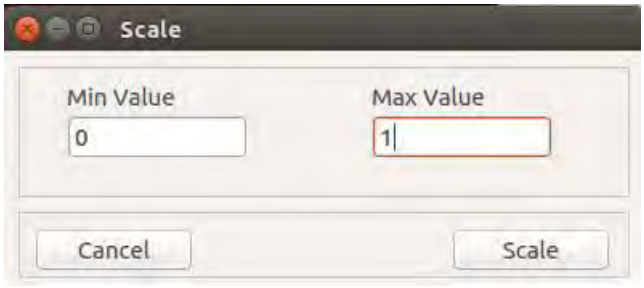
5. Ahora se procede a codificar las variables categóricas usando la opción del Menú Principal (Statistical Categorical Encoding). Con este método se obtiene un valor numérico para cada una de las categorías de las variables categóricas con un alto valor de correlación con las demás.



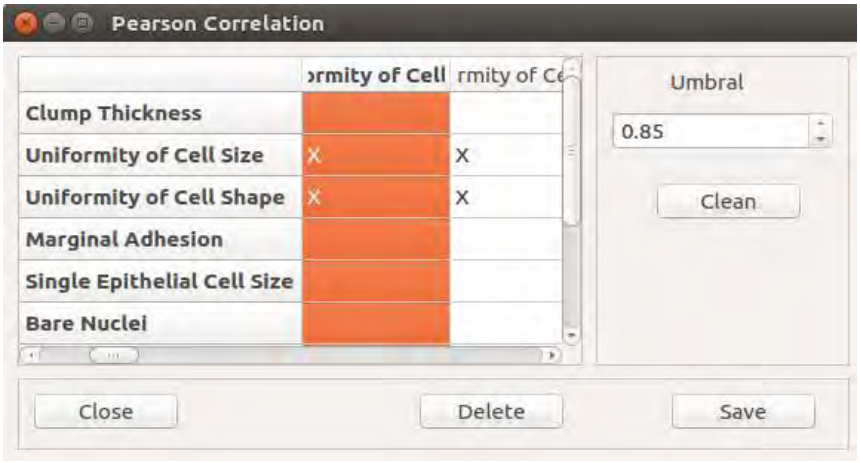
6. Se procedió a detectar valores atípicos y fueron detectadas 69 tuplas.



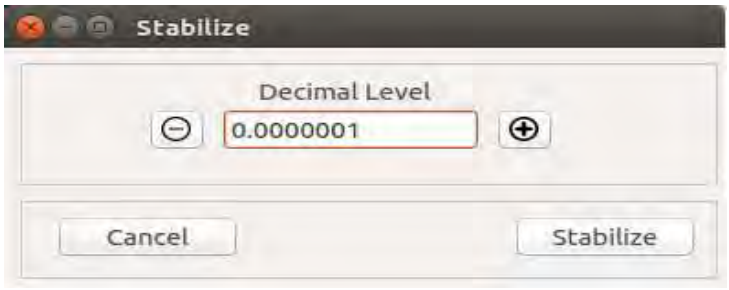
7. Ahora se procede a escalar los datos entre [0, 1].



8. Se realiza el análisis de correlación de Pearson y se encuentra una correlación mayor 0.85 entre las variables *Uniformity of Cell Size* y *Uniformity of Cell Shape*, por lo que se procede a eliminar *Uniformity of Cell Size*.



9. Esta etapa consiste en estabilizar los datos para evitar inestabilidad matemática en el momento de aplicar alguna técnica de aprendizaje de máquina.



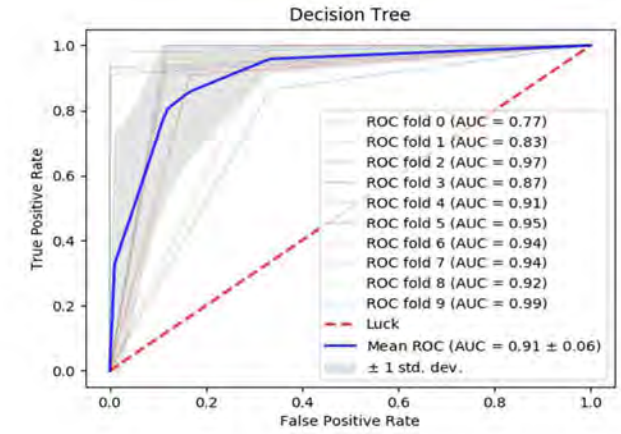
10. En el último paso se exporta el archivo obtenido a un archivo de texto delimitado por comas (.csv)

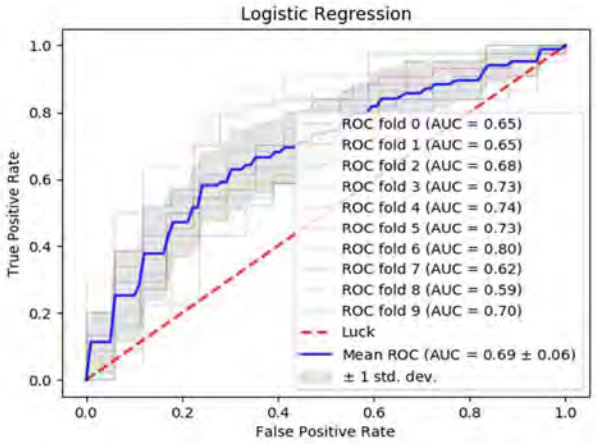
Resultados obtenidos luego de aplicar el preprocesamiento:

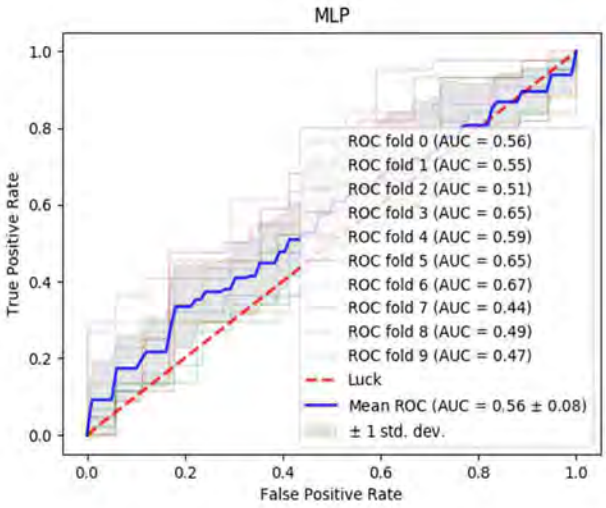
Para obtener estos resultados luego de aplicar el preprocesamiento de datos, se realizó validación cruzada de 10 particiones para disminuir la incertidumbre con respecto a la exactitud de los algoritmos analizados frente al dataset.

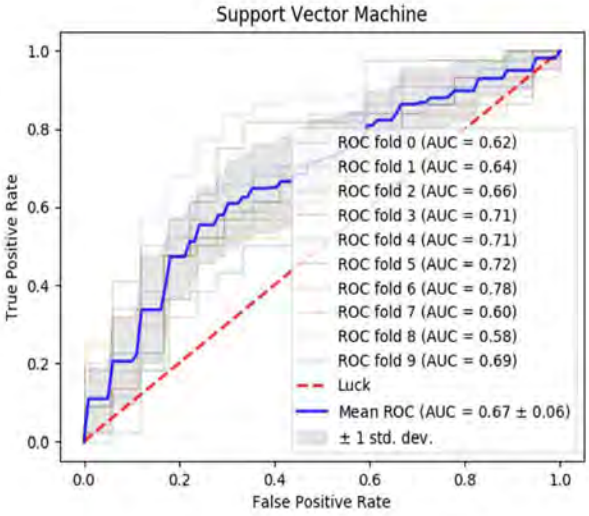
Algoritmo	**Accuracy**
Regresión Logística	0.96 (+/- 0.07)
Árbol de Decisión	0.95 (+/- 0.06)
Redes Neuronales	0.96 (+/- 0.07)
Máquina de Soporte de Vectorial	0.97 (+/- 0.05)
Random Forest	0.98 (+/- 0.04)
Gradient Boosting	0.97 (+/- 0.06)

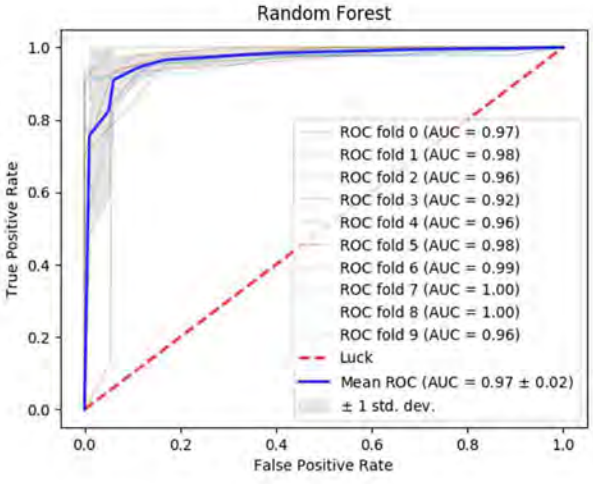
Posteriormente se procedió a calcular el área bajo la Curva para cada uno de estos algoritmos. En las gráficas siguientes se muestran los resultados obtenidos

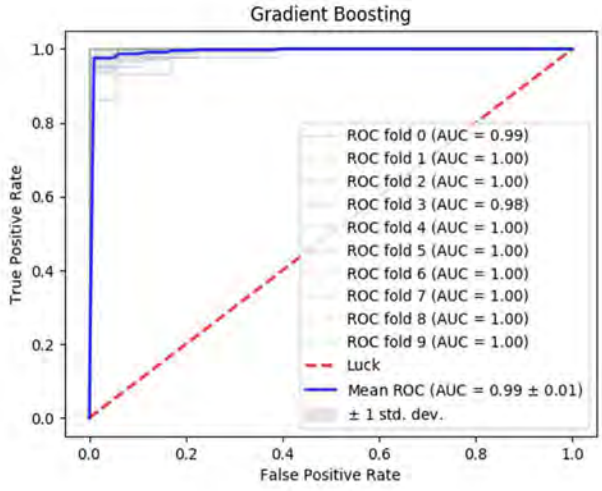

Decision Tree
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.77)
ROC fold 1 (AUC = 0.83)
ROC fold 2 (AUC = 0.97)
ROC fold 3 (AUC = 0.87)
ROC fold 4 (AUC = 0.91)
ROC fold 5 (AUC = 0.95)
ROC fold 6 (AUC = 0.94)
ROC fold 7 (AUC = 0.94)
ROC fold 8 (AUC = 0.92)
ROC fold 9 (AUC = 0.99)
Luck
Mean ROC (AUC = 0.91 ± 0.06)
± 1 std. dev.


Logistic Regression
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.65)
ROC fold 1 (AUC = 0.65)
ROC fold 2 (AUC = 0.68)
ROC fold 3 (AUC = 0.73)
ROC fold 4 (AUC = 0.74)
ROC fold 5 (AUC = 0.73)
ROC fold 6 (AUC = 0.80)
ROC fold 7 (AUC = 0.62)
ROC fold 8 (AUC = 0.59)
ROC fold 9 (AUC = 0.70)
Luck
Mean ROC (AUC = 0.69 ± 0.06)
± 1 std. dev.


MLP
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.56)
ROC fold 1 (AUC = 0.55)
ROC fold 2 (AUC = 0.51)
ROC fold 3 (AUC = 0.65)
ROC fold 4 (AUC = 0.59)
ROC fold 5 (AUC = 0.65)
ROC fold 6 (AUC = 0.67)
ROC fold 7 (AUC = 0.44)
ROC fold 8 (AUC = 0.49)
ROC fold 9 (AUC = 0.47)
Luck
Mean ROC (AUC = 0.56 ± 0.08)
± 1 std. dev.


Support Vector Machine
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.62)
ROC fold 1 (AUC = 0.64)
ROC fold 2 (AUC = 0.66)
ROC fold 3 (AUC = 0.71)
ROC fold 4 (AUC = 0.71)
ROC fold 5 (AUC = 0.72)
ROC fold 6 (AUC = 0.78)
ROC fold 7 (AUC = 0.60)
ROC fold 8 (AUC = 0.58)
ROC fold 9 (AUC = 0.69)
Luck
Mean ROC (AUC = 0.67 ± 0.06)
± 1 std. dev.


Random Forest
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.97)
ROC fold 1 (AUC = 0.98)
ROC fold 2 (AUC = 0.96)
ROC fold 3 (AUC = 0.92)
ROC fold 4 (AUC = 0.96)
ROC fold 5 (AUC = 0.98)
ROC fold 6 (AUC = 0.99)
ROC fold 7 (AUC = 1.00)
ROC fold 8 (AUC = 1.00)
ROC fold 9 (AUC = 0.96)
Luck
Mean ROC (AUC = 0.97 ± 0.02)
± 1 std. dev.


Gradient Boosting
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.99)
ROC fold 1 (AUC = 1.00)
ROC fold 2 (AUC = 1.00)
ROC fold 3 (AUC = 0.98)
ROC fold 4 (AUC = 1.00)
ROC fold 5 (AUC = 1.00)
ROC fold 6 (AUC = 1.00)
ROC fold 7 (AUC = 1.00)
ROC fold 8 (AUC = 1.00)
ROC fold 9 (AUC = 1.00)
Luck
Mean ROC (AUC = 0.99 ± 0.01)
± 1 std. dev.

Anexo 4.

Descripción de las etapas de preprocesamiento del conjunto de datos de Evaluación de Autos.

Información

La tarea de clasificar teniendo en cuenta ciertas características proporcionadas si un auto es aceptable(acc), no aceptable (unacc), bueno (good) o muy bueno (v-good). A continuación, se describen los datos con los que se cuenta:

Descripción de las Variables:

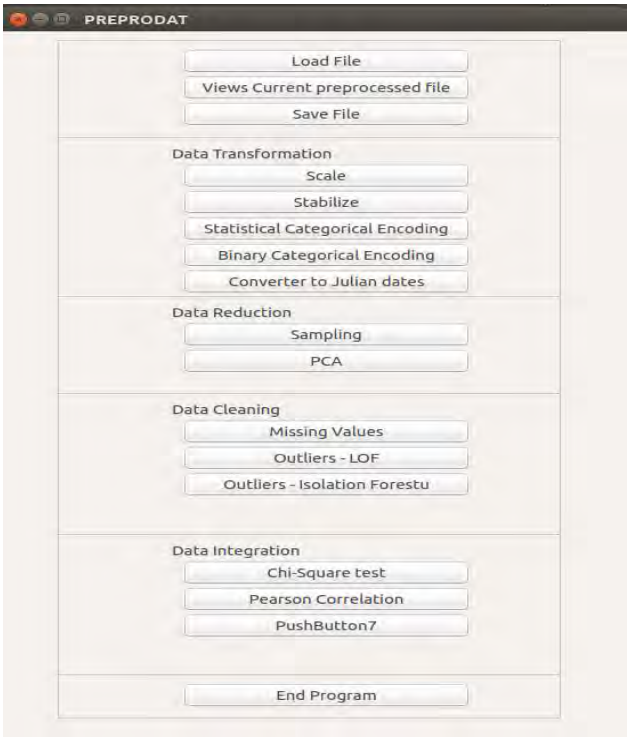
Variable	**Valores**
buying	v-high, high, med, low
maint	v-high, high, med, low
doors	2, 3, 4, 5-more
persons	2, 4, more
lug_boot	small, med, big
safety	low, med, high

Descripción de la Clase

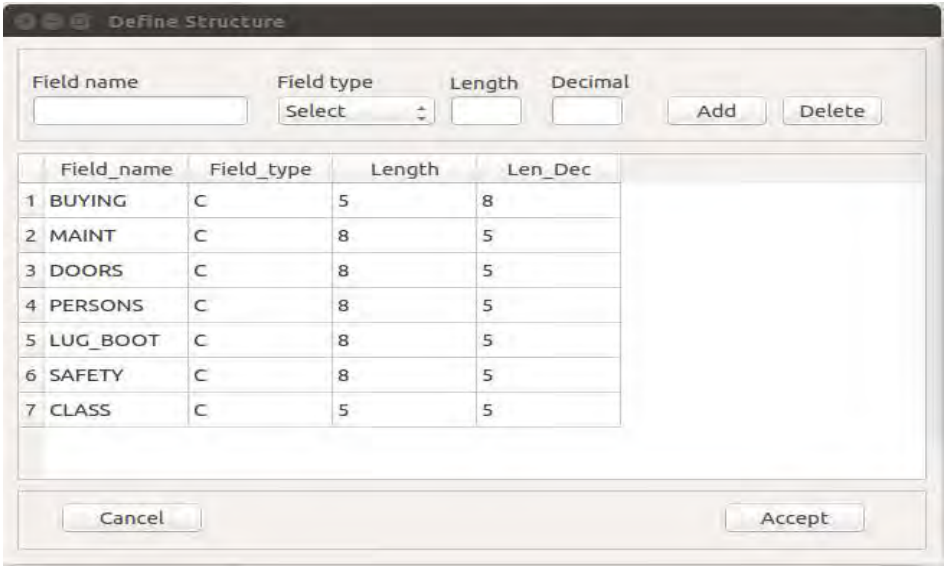
Class	**Número**	**Número (%)**
unacc	1210	70.023%
acc	384	22.222%
good	69	3.993%
v-good	65	3.762%

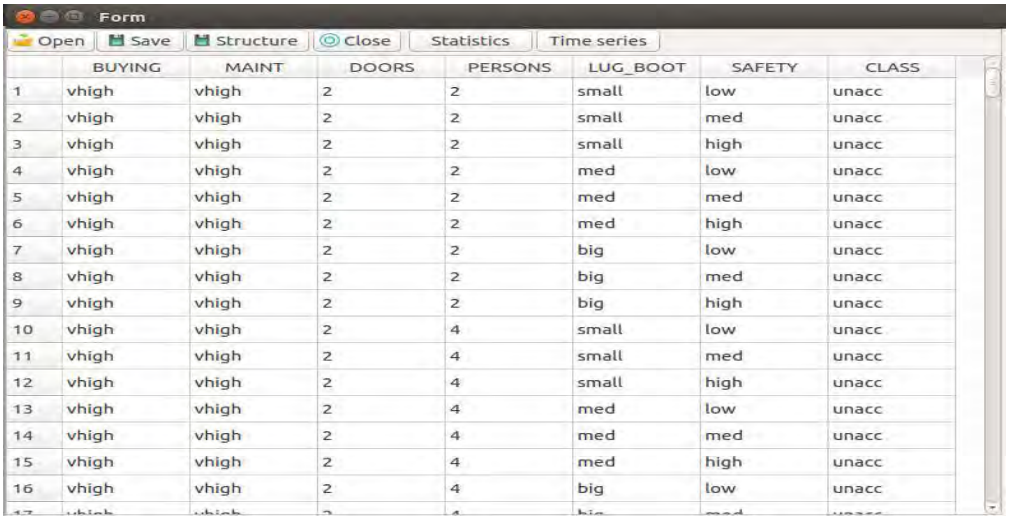
Descripción del preprocesamiento realizado:

1. Al abrir la aplicación nos encontramos con el menú que se muestra a continuación para cargar y definir la estructura de datos entramos al menú (Load File).



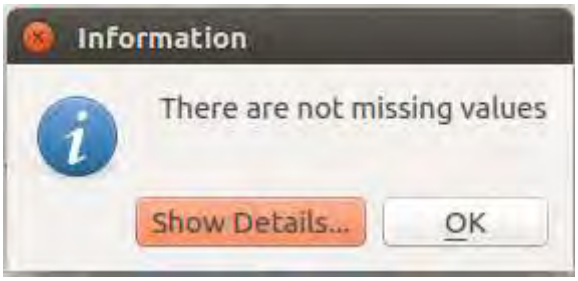
2. Se procede a definir la estructura de los datos para conocer que manejo hacerle dependiendo si son numéricos, categóricos o fechas y posteriormente se carga el conjunto de datos correspondientes.



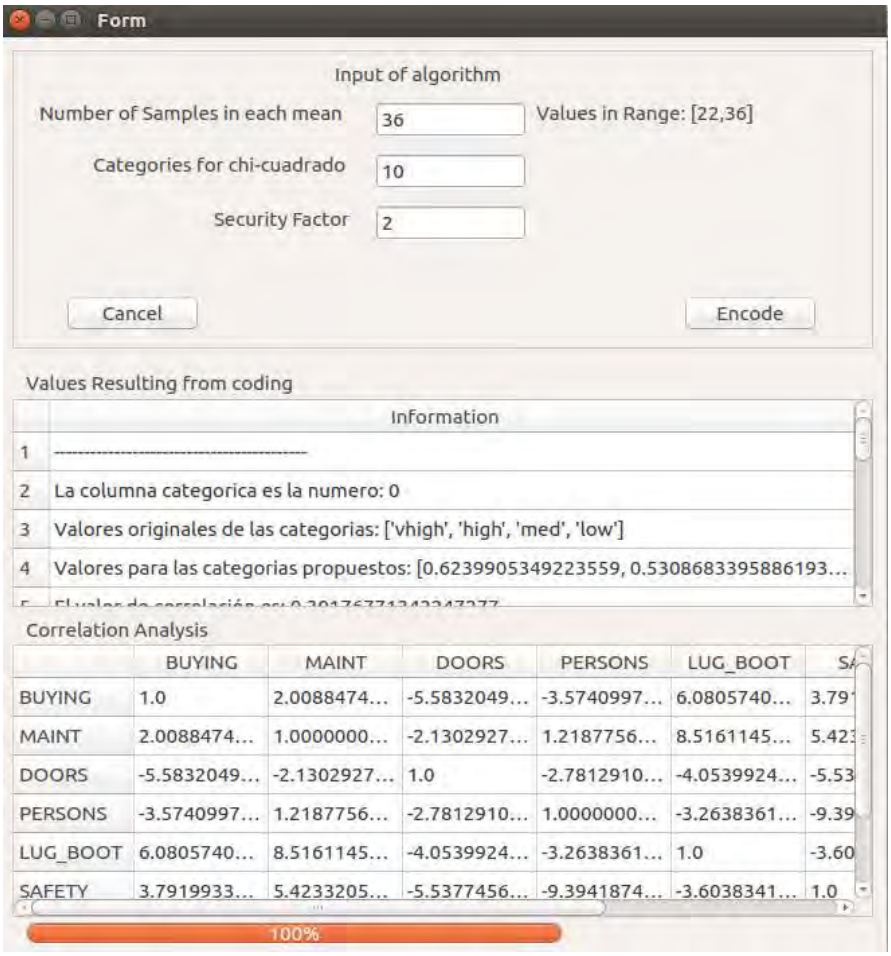


	BUYING	MAINT	DOORS	PERSONS	LUG_BOOT	SAFETY	CLASS
1	vhigh	vhigh	2	2	small	low	unacc
2	vhigh	vhigh	2	2	small	med	unacc
3	vhigh	vhigh	2	2	small	high	unacc
4	vhigh	vhigh	2	2	med	low	unacc
5	vhigh	vhigh	2	2	med	med	unacc
6	vhigh	vhigh	2	2	med	high	unacc
7	vhigh	vhigh	2	2	big	low	unacc
8	vhigh	vhigh	2	2	big	med	unacc
9	vhigh	vhigh	2	2	big	high	unacc
10	vhigh	vhigh	2	4	small	low	unacc
11	vhigh	vhigh	2	4	small	med	unacc
12	vhigh	vhigh	2	4	small	high	unacc
13	vhigh	vhigh	2	4	med	low	unacc
14	vhigh	vhigh	2	4	med	med	unacc
15	vhigh	vhigh	2	4	med	high	unacc
16	vhigh	vhigh	2	4	big	low	unacc

3. El conjunto de datos que se está analizando no cuenta con valores perdidos por lo que en esta etapa no fue necesario hacer la tarea de imputación o eliminación de valores perdidos.



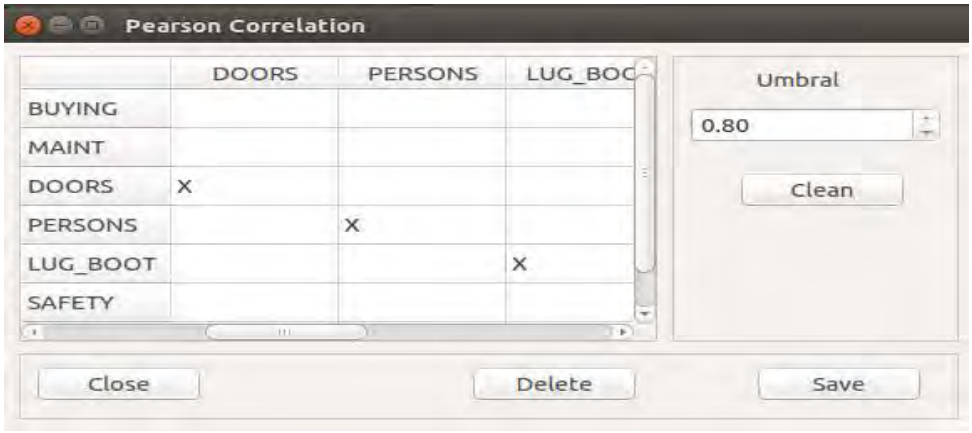
4. Las variables de la base de datos son categóricas por lo que se procederá con la codificación estadística de variables categóricas. Este proceso permite obtener un valor numérico para cada categoría.



5. Se procedió a detectar valores atípicos y fueron detectados 173 tuplas.

6. Como todas las variables eran categóricas y se obtuvieron los valores usando la codificación estadística de variables categóricas, ya estos se encuentra escalados y no hay riesgo de encontrar el problema que motiva a estabilizar los datos.

7. Se realiza el análisis de correlación de Pearson y no se encuentra una correlación mayor 0.85, por lo que no fue necesario eliminar ninguna de las variables.



8. En el último paso, exportó el archivo obtenido a un archivo de texto delimitado por comas(.csv)

Resultados obtenidos con el conjunto de datos preprocesados:

Para obtener estos resultados luego de aplicar el preprocesamiento de datos, se realizó validación cruzada de 10 particiones para disminuir la incertidumbre con respecto a la exactitud de los algoritmos analizados frente al dataset.

Algoritmo	Accuracy
Regresión Logística	0.86 (+/- 0.04)
Árbol de Decisión	0.99 (+/- 0.02)
Redes Neuronales	0.92 (+/- 0.02)
Máquina de Soporte de Vectorial	0.90 (+/- 0.03)
Random Forest	0.98 (+/- 0.03)
Gradient Boosting	0.99 (+/- 0.02)

Anexo 5.

Descripción de las tecnologías y herramientas usadas para la implementación del software para el preprocesamiento de datos.

Herramientas

1. PyCharm

PyCharm es un entorno de desarrollo integrado, específicamente para el lenguaje Python y creado por JetBrains. Este IDE fue usado para la implementación de la aplicación debido a que integra algunos paquetes científicos que fueron usado como son: Matplotlib y NumPy. PyCharm cuenta con algunas características que facilitan el desarrollo como es el completado de código, también ayuda a generar código limpio y sin duplicado producto de copiado/pegado mediante un detector de código duplicado.

2. DataGrip

DataGrip es un entorno de base de datos de motores múltiples. Es compatible con MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, DB2, HyperSQL, Apache Derby, H2 y SQLite que es para el cual se hará uso de esta herramienta. Incluye un editor que proporciona asistencia de codificación inteligente para editar código SQL, como autocompletado, análisis y navegación. También presenta una consola de consulta para ejecutar y crear perfiles de consultas.

3. QtDesigner

Qt Designer es la herramienta de Qt para diseñar y construir interfaces gráficas de usuario (GUI) a partir de componentes Qt. Puede componer y personalizar sus widgets o cuadros de diálogo de una manera WYSIWYG ("lo que ves es lo que se

obtiene") y probarlos con diferentes estilos y resoluciones lo cual fue de mucha ayuda para el diseño y generación de código de las interfaces gráficas de la aplicación.

Tecnologías

1. Python

Python es un lenguaje de programación interpretado, orientado a objeto y de alto nivel con semántica dinámica. Este cuenta con una sintaxis simple y fácil, su legibilidad es alta y, por lo tanto, reduce los costos de mantenimiento. La versión de Python 3.5 es la que se utiliza para la implementación y cuenta con un gran número de bibliotecas que son de gran utilidad para las labores que se requieren como son: scikit-learn, numpy, pandas, matplotlib y scipy.

2. PyQt

PyQt 4.11 fue la versión utilizada de framework multiplataforma orientado a objetos que es ampliamente usado para desarrollar programas que utilicen interfaz gráfica de usuario. Qt utiliza el lenguaje de programación C++ de forma nativa, adicionalmente puede ser utilizado en varios otros lenguajes de programación a través de bindings.

PyQt es un binding (es una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita) de la biblioteca gráfica Qt para el lenguaje de programación Python que permitirá desarrollar la interfaz gráfica y lograr la tarea de preprocesamiento de una manera más amigable y intuitiva para el usuario.

3. SQLite

SQLite es un sistema de gestión de base de datos relacional contenida en una biblioteca escrita en C. El código para SQLite es de dominio público y, por lo tanto, es gratuito para cualquier uso, comercial o privado. SQLite es la base de datos más implementada del mundo con más aplicaciones de las que podemos contar, incluidos varios proyectos de alto perfil.

SQLite es un motor de base de datos SQL incorporado. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso de servidor por separado lo cual permite mantener la portabilidad de la aplicación que es una de las características que se pretenden lograr. SQLite lee y escribe directamente en archivos de disco ordinarios. Una base de datos SQL completa con múltiples tablas, índices, disparadores y vistas, está contenida en un solo archivo de disco.

Anexo 6.

Manual de usuario del software implementado para realizar el preprocesamiento de datos.

Requisitos del Programa

Los requisitos previos de software instalado para poder ejecutar el programa serían:

- Python 3.5
- PyQt4.11

Los requisitos mínimos de hardware serían los siguientes:

- Sistema Operativo: Linux y Windows.
- CPU: Core 2 Duo or Athlon X2 at 2.4 GHz.
- Memoria RAM: 2GB.
- Disco Duro: 2 GB libres.

Pasos para la instalación en Windows y Linux

1. Download Anaconda 5.0.1 para la versión de Python 3.6 e instalar (https://www.anaconda.com/download/)
9. Abrir consola >conda install python=3.5
10. Download PyQt4-4.11.4-cp35-cp35m-win_amd64.whl (https://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt4)
11. Usando la consola, ir al directorio donde se encuentra el archivo antes descargado y ejecutar para instalar pyqt4 > pip install PyQt4-4.11.4-cp35-cp35m-win_amd64.whl

Menú de Opciones del Software

Las diferentes opciones disponibles, así como el funcionamiento que ofrecen cada una de ellas se lista en la siguiente Tabla 1:

Pantalla Inicial

#	Nombre	Descripción
1	Load File	Permite acceder a una ventana para proceder a cargar el conjunto de datos, definir la estructura, visualizar propiedades de los datos y gráficas de estos.
2	View Current preprocessed file	Permite acceder a una ventana para visualizar los datos luego de realizar alguna de las etapas del preprocesamiento de los datos.
3	Save File	Permite almacenar el último conjunto de datos que se obtuvo de realizar cualquiera de las tareas de preprocesamiento.
Data Transformation		
4	Scale	Permite acceder a una ventana para definir un rango y proceder a escalar los datos.
5	Stabilize	Permite acceder a una ventana para definir el grado decimal y proceder a estabilizar los datos.
6	Statistical Categorical Encoding	Permite acceder a una ventana para definir los parámetros y ver los resultados de realizar la codificación estadística a variables categóricas
7	Binary Categorical Encoding	Realiza la codificación binaria de variables categóricas.
8	Converter to Julian Date	Permite convertir las fechas gregorianas a juliana.
Data Reduction		
9	Sampling	Permite acceder a una ventana para definir la forma para determinar el tamaño de la muestra y el tipo de muestreo, para realizar el muestro.

10	PCA	Permite acceder a una ventana para definir el método para realizar el análisis de componentes principales, así como poder almacenar el resultado.
Data Cleaning		
11	Imputation of missing values	Permite acceder a una ventana donde se puede ver la entropía de los datos eliminando los valores perdidos y imputando los mismos. Y da la opción de imputar o eliminar los valores perdidos y guardar cada uno de los conjuntos de datos resultantes.
12	Detection of outliers values	Detecta los valores atípicos y los elimina usando el método de Isolation Forest.
Data Integration		
13	Chi- Square Test	Permite acceder a una ventana para realizar las pruebas de chi-cuadrado y ver los resultados de la misma. También provee la opción de eliminar alguna y poder almacenar el conjunto de datos.
14	Pearson Correlation	Permite acceder a una ventana donde se muestran el cálculo de la correlación de Pearson. Permite definir un umbral para analizar las mismas y eliminar algunas que estén muy correlacionadas.
15	End Program	Finaliza la aplicación.

Tabla 1. Opciones de la ventana principal de PREPRODATA

Ventana Principal.

Esta es la ventana principal de la aplicación PREPRODATA que se utiliza para el preprocesamiento de datos.

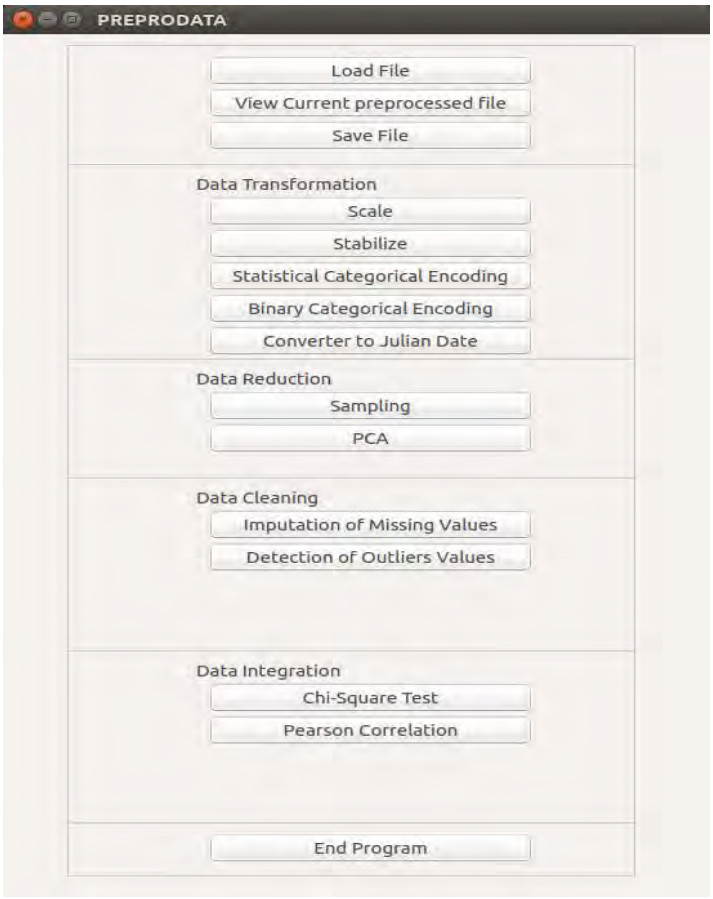
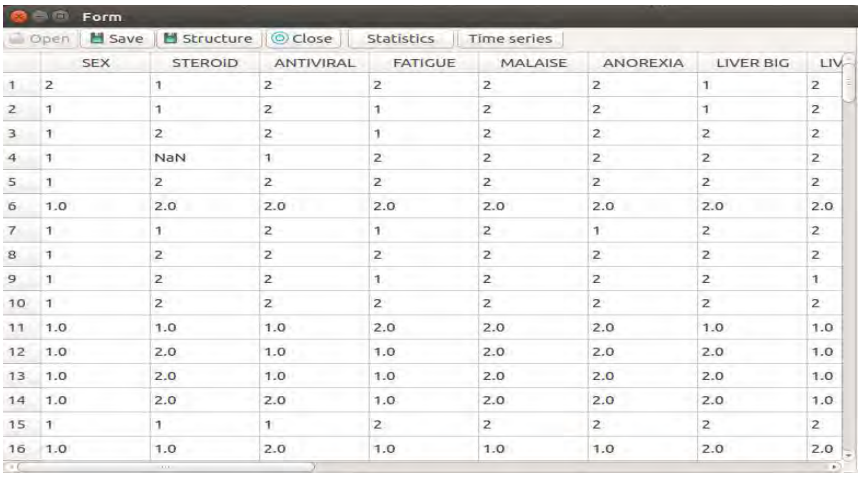


Figura 1. Ventana Principal de la aplicación PREPRODATA.

Ventana para cargar y visualizar los datos.

A esta ventana se accede desde la opción Load File o View Current preprocessed file desde la ventana principal de la aplicación. Desde esta se accede a definir la estructura de los datos en la opción Structure (Ventana en Figura 3), y posteriormente en la Opción (Open) a seleccionar el conjunto de datos correspondientes. También permite visualizar estadísticas de los datos (Ventana en la Figura 4), gráficas (Ventana en la Figura 5), y guardar el conjunto de datos (Ventana en la Figura 6).



	SEX	STEROID	ANTIVIRAL	FATIGUE	MALAISE	ANOREXIA	LIVER BIG	LIV
1	2	1	2	2	2	2	1	2
2	1	1	2	1	2	2	1	2
3	1	2	2	1	2	2	2	2
4	1	NaN	1	2	2	2	2	2
5	1	2	2	2	2	2	2	2
6	1.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0
7	1	1	2	1	2	1	2	2
8	1	2	2	2	2	2	2	2
9	1	2	2	1	2	2	2	1
10	1	2	2	2	2	2	2	2
11	1.0	1.0	1.0	2.0	2.0	2.0	1.0	1.0
12	1.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0
13	1.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0
14	1.0	2.0	2.0	1.0	2.0	2.0	2.0	1.0
15	1	1	1	2	2	2	2	2
16	1.0	1.0	2.0	1.0	1.0	1.0	2.0	2.0

Figura 2. Ventana correspondiente a la opción 1 (Load File)

La ventana para definir la estructura permite poner las variables que contiene la base de datos antes de ser cargada en la aplicación. En la misma se define el nombre de las variables, así como el tipo de dato (Numérico, Categórico y Fecha) para identificar de manera más eficiente que opción del menú están disponibles. Esto se debe a que dependiendo del tipo de datos con que cuenta la base de datos se puede acceder a diferentes opciones de menú.

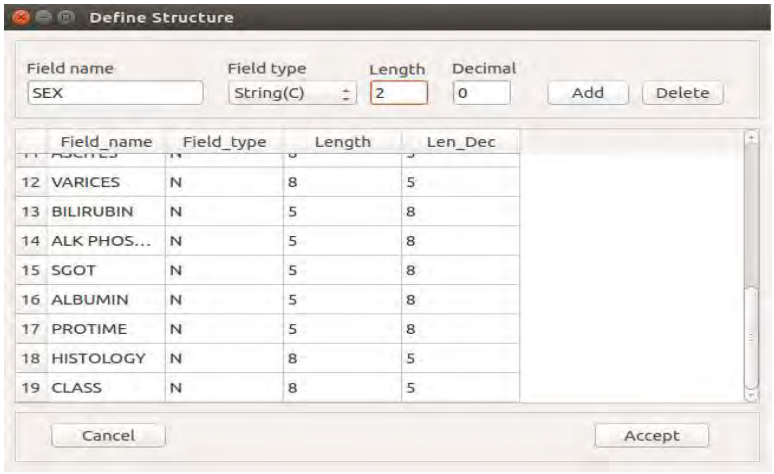


Figura 3. Ventana para definir la estructura del conjunto de datos a cargar (Structure).

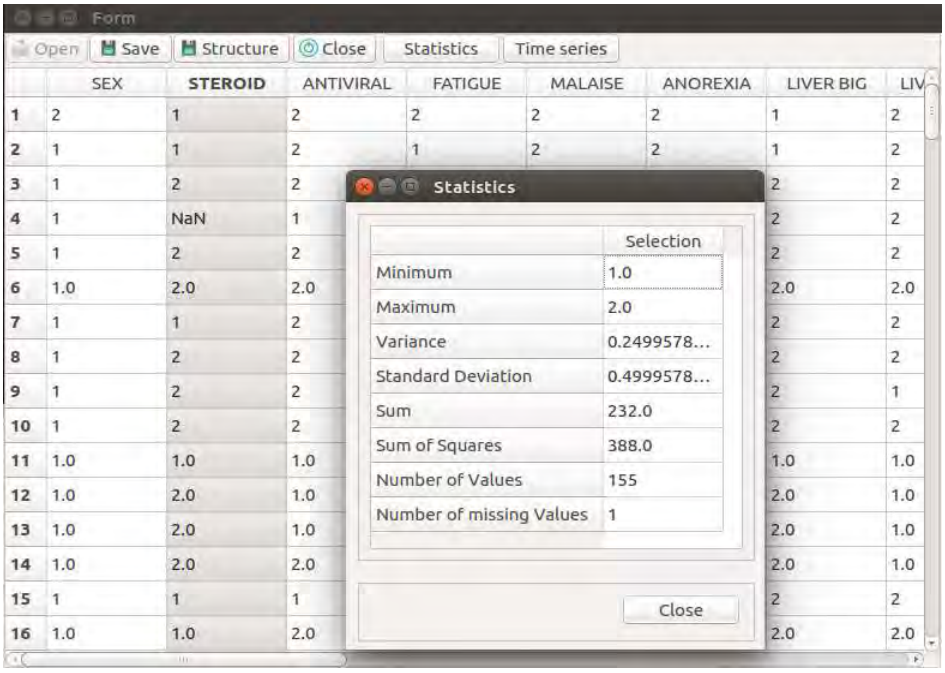


Figura 4. Ventana para ver la estadística de una columna seleccionada (Statistics).

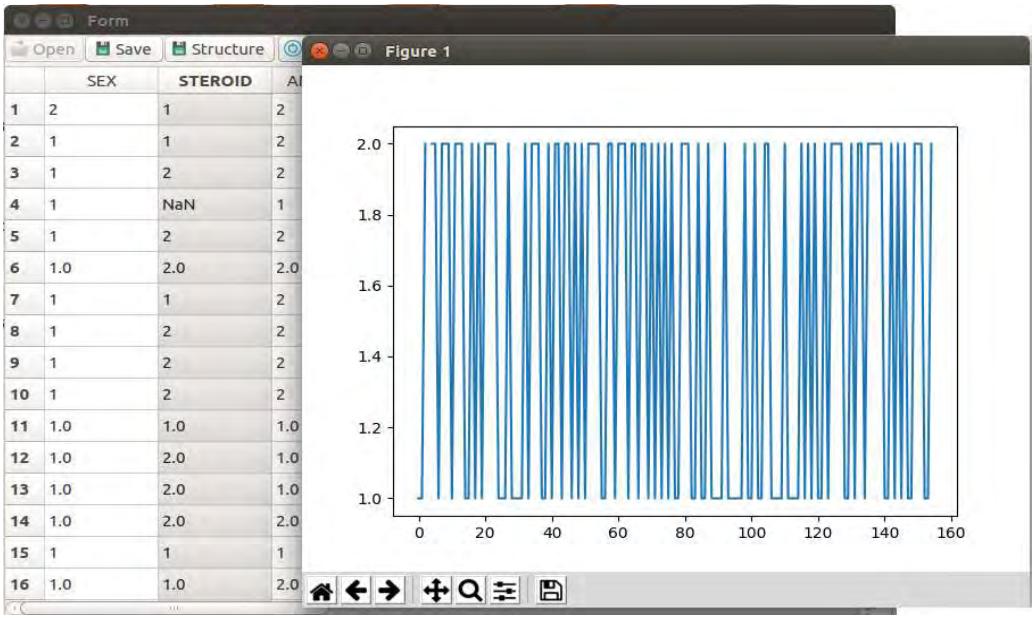


Figura 5. Ventana para ver gráfica de una columna seleccionada (Time Series).

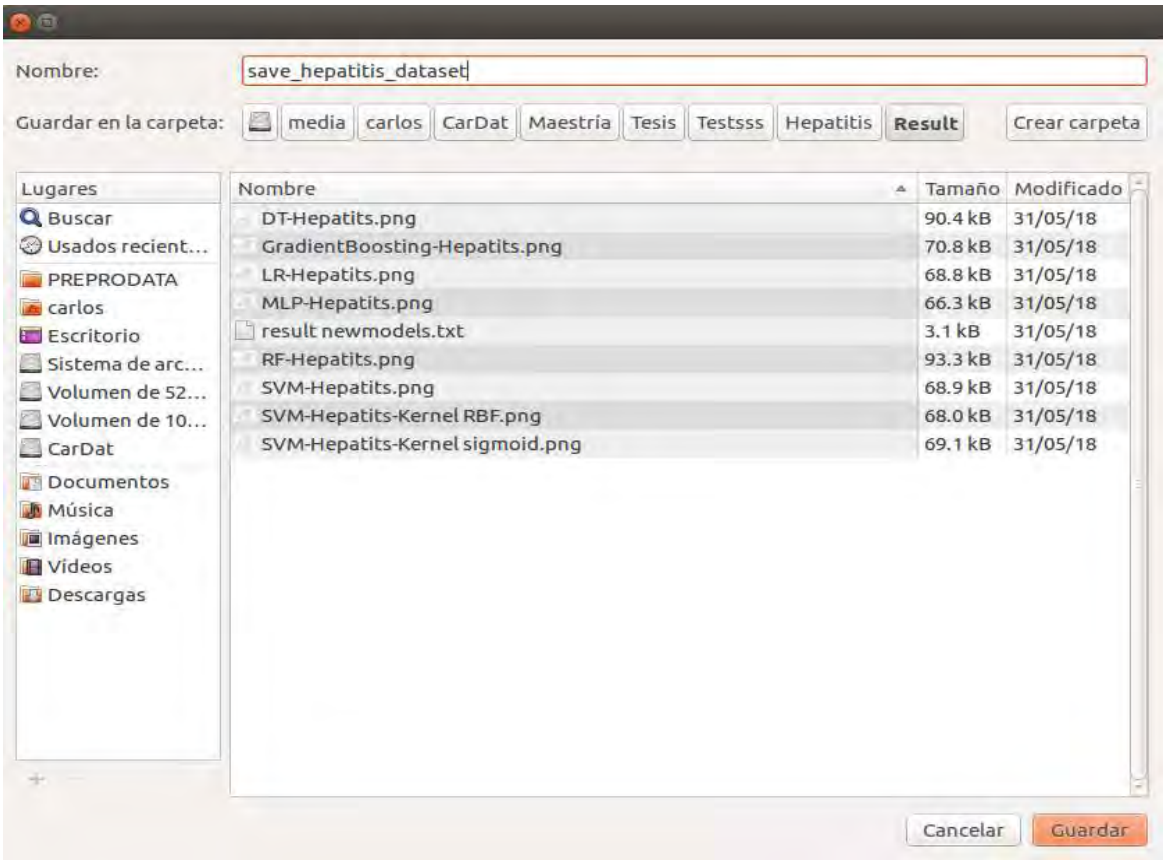


Figura 6. Ventana para seleccionar el destino del archivo a almacenar.

Opción para escalar los datos.

En la ventana principal se accede en la Opción Scale y se muestra la ventana para definir el rango para proceder a escalamiento (Ventana en la Figura 7). Definir el valor mínimo y valor máximo es el primer paso, normalmente se usa [0,1] o [-1,1]. Luego de definir estos datos se procede a Escalar (Scale).

Figura 7. Ventana para escalar los datos.

Opción para estabilizar los datos.

En la ventana principal se accede en la Opción Stabilize y se muestra la ventana para definir el grado decimal de la estabilización (Ventana en la Figura 8). Luego de definir este dato se procede a Estabiliza (Stabilize).

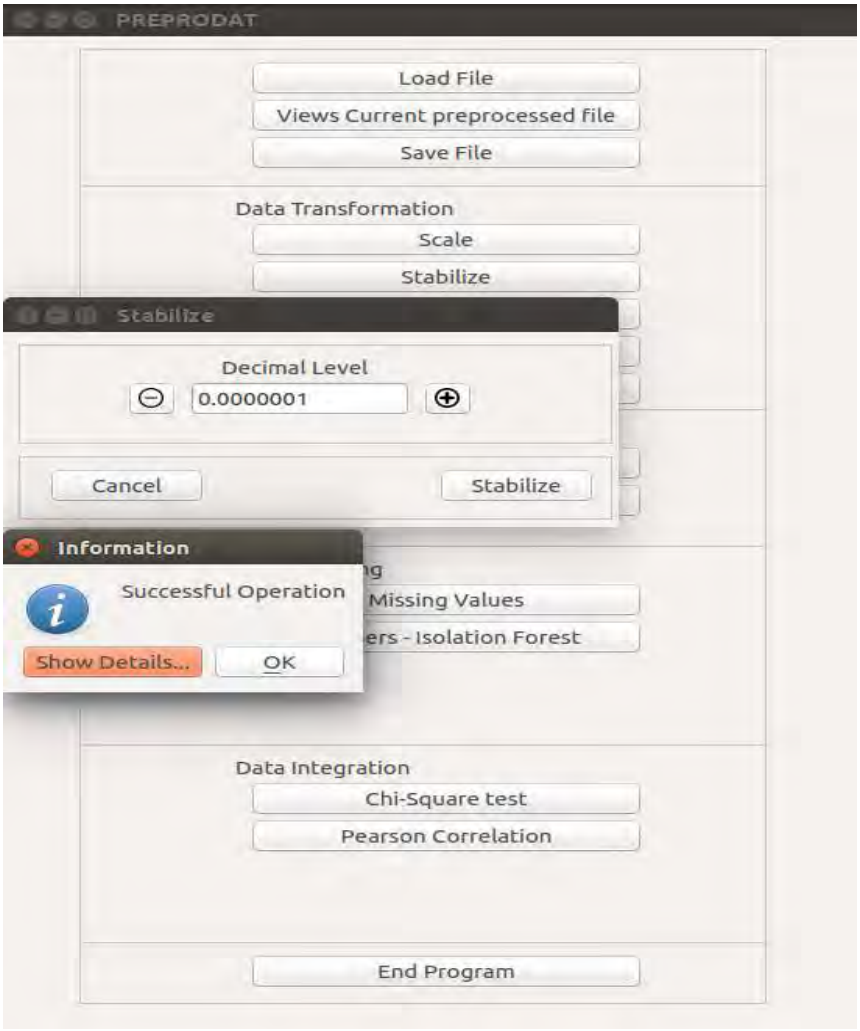


Figura 8. Ventana para estabilizar los datos.

Opción para codificar las variables categóricas usando codificación estadística.

En la ventana principal se accede a la opción Statistical Categorical Encoding y se muestra la ventana para establecer los parámetros para realizar la codificación estadística de las variables categóricas (Ventana Figura 9). A esta ventana solo se accede en caso de que existan variables categóricas definidas.

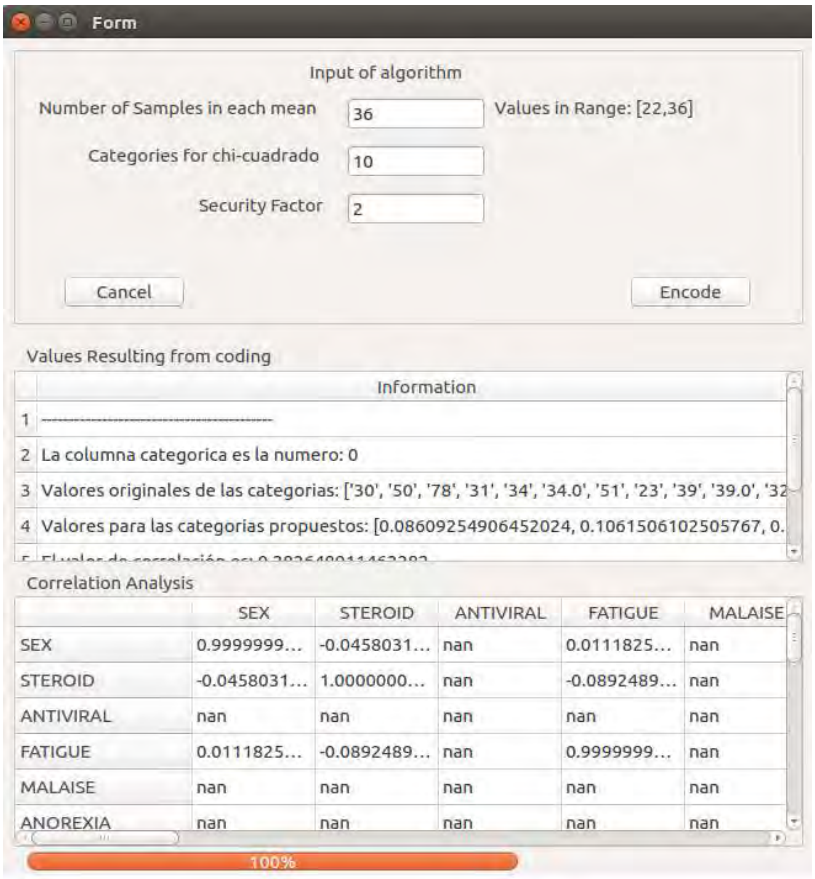


Figura 9. Ventana para configurar los paramentos y ver los resultados de la codificación estadística de variables categóricas.

Opción para codificar las variables categóricas usando codificación binaria.

En la ventana principal se accede a la opción Statistical Categorical Binary y se realiza la codificación binaria de las variables categóricas. Se muestra la opción de operación realizada satisfactoriamente (Ventana Figura 10). Para ver los datos obtenidos se puede acceder a la opción View Current preprocessed file.

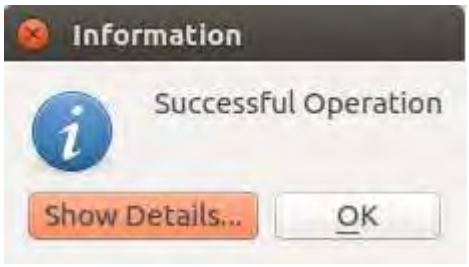


Figura 10. Ventana de notificación de que la operación se realizó satisfactoriamente.

Opción para convertir a fecha juliana las fechas gregorianas

En la ventana principal se accede a la opción Converter to Julian Date. A la ventana de esta opción se accede cuando están definidas fechas gregorianas y permite seleccionar con que estructura está definida la fecha en el conjunto de datos.

Figura 11. Ventana para seleccionar la forma en la que está definida la fecha gregoriana.

Opción para muestrear los datos.

En la ventana principal se accede a la opción Sampling y se muestra una ventana donde aparecen las opciones para realizar el muestreo (Figura 12). En esta ventana se selecciona el método para determinar el tamaño de la muestra y se define si el muestreo será aleatorio simple con o sin reposición. El tamaño de la muestra puede ser calculado usando la entropía (Figura 13) o definirlo personalmente (Figura 12).

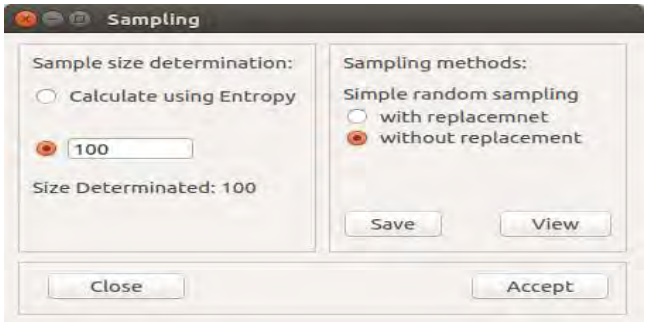


Figura 12. Ventana para definir los parámetros del muestro (Tamaño de muestra definido).

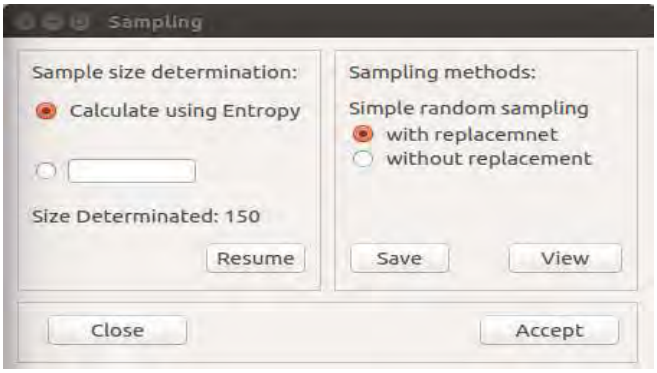


Figura 13. Ventana para definir los parámetros del muestro (Tamaño de muestra calculado).

Para el caso de que el tamaño de la muestra sea calculado usando la entropía, también se pueden acceder a otros valores del cálculo mediante la opción Resume que se muestra en la Figura 13 y se visualiza una ventana como se muestra en la Figura 14.

Figura 14. Ventana de la opción Resume cuando se utiliza el cálculo de entropía para calcular el tamaño de la muestra.

Opción para el análisis de componentes principales (PCA).

En la ventana principal se accede a la opción PCA y se muestra una ventana (Figura 15) donde se puede seleccionar la forma en que se va a determinar el número de componentes principales y luego de ejecutar se visualizan los datos de precisión y covarianza del cálculo. También se cuenta con la opción de guardar (Save) los datos obtenidos del análisis de componentes principales (PCA).

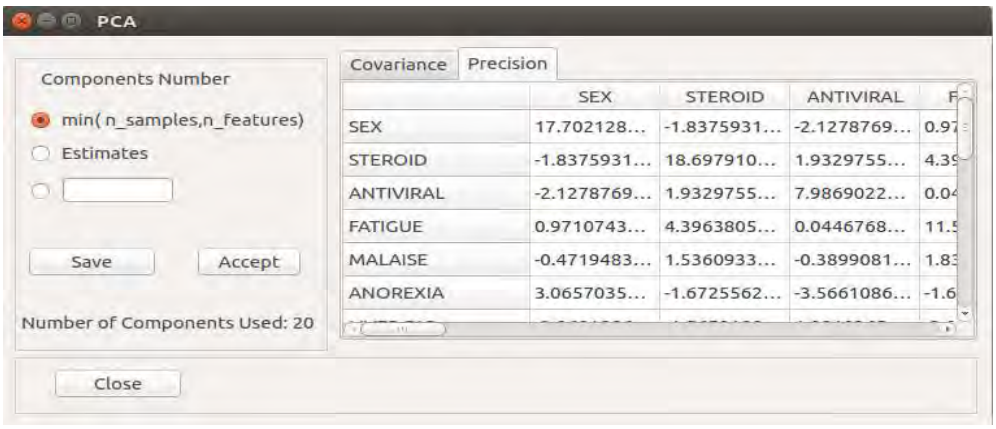


Figura 15. Ventana para definir la forma de determinar el número de componentes y realizar el PCA.

Opción de imputación de valores perdidos.

En la ventana principal se accede a la opción imputación de valores perdidos (*Imputation of Missing Values*) y en caso de existir valores perdidos en las variables numéricas se procede a acceder a la ventana de *Missing Values* (Ventana Figura 16). En esta ventana se puede ver la cantidad de valores perdidos y la entropía de estos. También es posible calcular la entropía luego de eliminarlas tuplas con valores perdidos y luego de la imputación para poder determinar si la diferencia no es significativa. La implementación define que si la diferencia es mayor de 10 entonces se queda con los datos imputados usando Spline Natural. En esta ventana se puede

proceder a almacenar cualquiera de los archivos resultantes tanto de la eliminación como de la imputación.

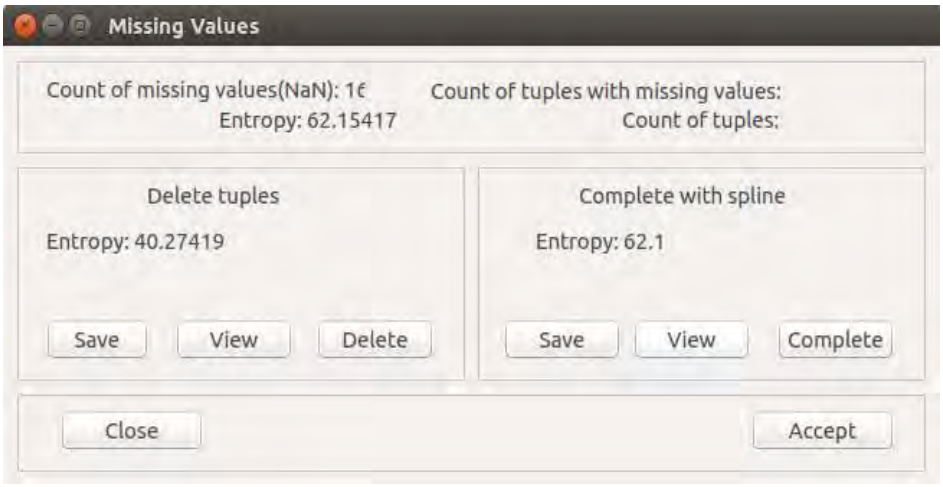


Figura 16. Ventana para definir la forma de determinar el número de componentes y realizar el PCA.

Opción para la detección y eliminación de valores atípicos.

En la ventana principal se accede a la opción detección de valores atípicos (*Detection of outliers values*). Esta procede a detectar usando el algoritmo *Isolation Forest* los valores atípicos y posteriormente se eliminan mostrado una ventana con la cantidad de valores atípicos detectados como se muestra en la Figura 17.

Figura 17. Ventana de mensaje cuando se procede a detectar valores atípicos.

Opción para realizar la Prueba de Chi Cuadrado.

En la ventana principal se accede a la opción de prueba de Chi- Cuadrado (Chi-Square Test) y se muestra una ventana para poder realizar el análisis entre las variables categóricas definidas (Ventana Figura 18) y poder proceder a analizar aquellas que sean consideradas. También basado en los resultados es posible eliminarlas (Delete), guardar (Save) o Aceptar (Accept) para continuar trabajando con el conjunto de datos resultantes luego de aplicar el procedimiento.

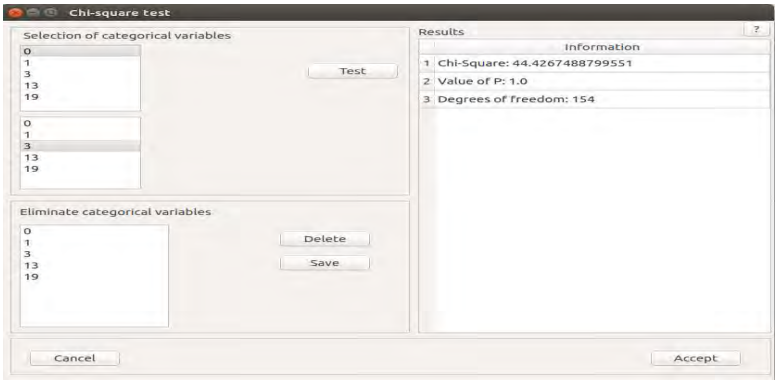


Figura 18. Ventana para realizar las pruebas de chi-cuadrado.

Opción para realizar el cálculo y análisis de la correlación de Pearson.

En la ventana principal se accede a la opción de Correlación de Pearson (Pearson Correlation) y se muestra una ventana con el resultado de la correlación de pearson entre las variables (Ventana Figura 19). En esta ventana es posible determinar un umbral y ver las variables que tiene una correlación mayor (Clean) que el umbral definido (Ventana Figura 20). También es posible basado en los resultados que se muestran eliminar (Delete) aquellas que tengan una correlación lineal fuerte y guardar (Save) el conjunto de datos resultantes para continuar con el preprocesamiento.

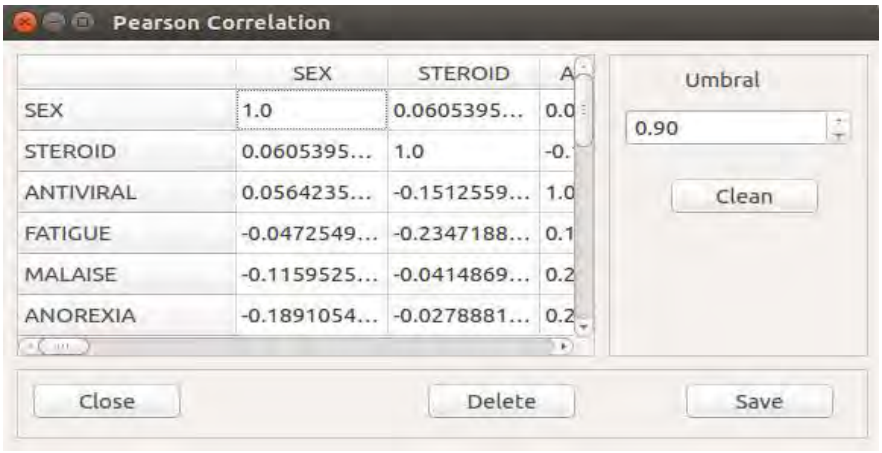


Figura 19. Ventana con el resultado del cálculo de la correlación de Pearson.

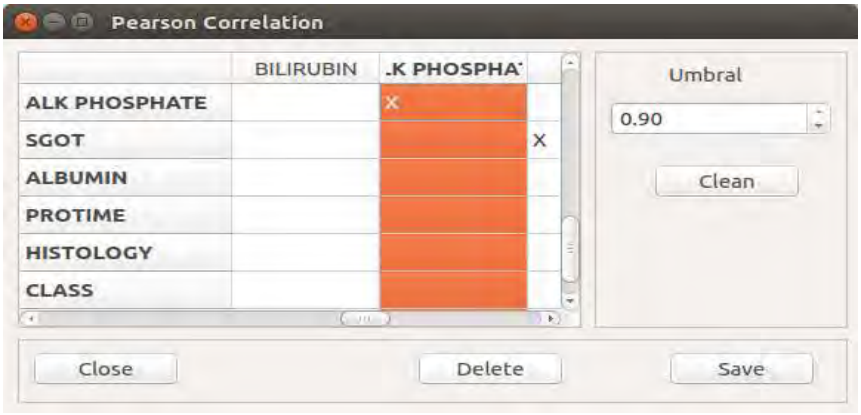


Figura 20. Ventana con el resultado del cálculo de la correlación de Pearson luego.

Opción para finalizar el programa.

En la ventana principal se accede a la opción de Finalizar Programa (End Program) y se procede cerrando el programa con todos los datos cargados.