UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN

SISTEMA DE APOYO A LA TOMA DE DECISIONES PARA LA REALIZACIÓN DE CORONARIOGRAFÍAS UTILIZANDO TÉCNICAS DE MINERÍA DE DATOS

Tesis presentada en opción al título académico de Máster en Ciencia de la Computación

Autora: Lic. Beyda González Camacho

SANTA CLARA, 2016

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN

SISTEMA DE APOYO A LA TOMA DE DECISIONES PARA LA REALIZACIÓN DE CORONARIOGRAFÍAS UTILIZANDO TÉCNICAS DE MINERÍA DE DATOS

Tesis presentada en opción al título académico de Máster en Ciencia de la Computación

Autora: Lic. Beyda González Camacho

Tutores: Dr.C Dánel Sánchez Tarragó

MS.c Mabel González Castellanos

Consultante: MS.c Dr. Francisco L. Moreno Martínez

SANTA CLARA, 2016

PENSAMIENTO

“En prevenir está todo el arte de salvar”

José Martí

DEDICATORIA

A mis padres, por el esfuerzo constante, amor y dedicación.

AGRADECIMIENTOS

A mi familia, a mi novio, a mis tutores, a los especialistas del Cardiocentro “Ernesto Che Guevara” de Santa Clara.

RESUMEN

La Minería de Datos es muy utilizada en casi todas las ramas de la ciencia, en especial en la medicina. Los datos recopilados en el Salón de Hemodinámica del Cardiocentro “Ernesto Che Guevara” pueden utilizarse en una aplicación de Minería de Datos. La aplicación de técnicas de preprocesamiento a los datos debe disminuir la complejidad del conjunto de datos. De las metodologías del Aprendizaje Automático Supervisado, la Clasificación es la que se utiliza cuando los datos están estructurados por clases y las medidas de exactitud ayudan a elegir el mejor modelo de clasificación. Se realizaron, manualmente, las tareas de limpieza de datos y normalización. Para imputar los valores perdidos se compararon tres algoritmos, de ellos el que mejor comportamiento mostró fue *KMI*. Para la selección de atributos se compararon los resultados de dos algoritmos: *ConsistencySubsetEval* y *ReliefFAttributeEval*, este último con dos variantes. Aunque los tres resultados disminuyeron la complejidad del conjunto de datos, ninguno fue un ganador global. En la selección de instancias se confrontaron los resultados de tres métodos, de ellos el que mejor desempeño tuvo fue *NCNEdit*. Para seleccionar el mejor clasificador simple se compararon los algoritmos que mayor sensibilidad presentaban, el que mejor comportamiento mostró fue *IBk*. Se compararon tres metaclasificadores, el que mejor valores de exactitud mostró fue *Stacking.* Se realizó un experimento añadiéndole instancias artificiales al conjunto de entrenamiento para lograr un modelo más interpretable logrando con el algoritmo *LMT* altos valores de sensibilidad y especificidad. Se tuvieron en cuenta los costos al seleccionar el modelo final, el elegido fue *CostSensitiveClassifier* usando como algoritmo de base a *Stacking*. Se evaluó el modelo en la práctica médica y se compararon sus resultados con la clasificación realizada por un grupo de especialistas de alto nivel. Esta comparación arrojó que el sistema cometió casi la mitad de errores menos que los especialistas, a pesar de haber sido probado con un conjunto de casos valorado por los especialistas como difícil de clasificar.

SUMMARY

Data mining is widely used in almost all branches of science, especially in medicine. Data collected in the Salón de Hemodinámica del Cardiocentro "Ernesto Che Guevara" can be used in a data mining application. The application of preprocessing techniques to data must decrease the complexity of the data set. From all the methodologies of Supervised Machine Learning, the Classification is used when data are structured by classes, and accuracy measures help choose the best classification model. The data cleansing and normalization tasks were performed manually. To impute missing values three algoritms were compared, who showed the best performance was KMI. For the selection of attributes the results of two algorithms were compared: ConsistencySubsetEval and ReliefFAttributeEval, the last one with two variants. Although the three results reduce complexity of the data set, none was a global winner. In selecting instances the results of three methods were compared, of which who had the best performance was NCNEdit. To select the best single classifier algorithms the algorithms with higher sensitivity were compared, the one who showed better performance was IBK. Three meta-classifiers were compared, who had better accuracy values was Stacking. An experiment was performed adding artificial instances to the training set to achieve a more interpretable model, the algorithm LMT achieve high values of sensitivity and specificity. Costs were taken into account when selecting the final model, the chosen was CostSensitiveClassifier using Stacking as base algorithm. The model was evaluated in medical practice and their results were compared with the classification made by a group of high-level specialists. This comparison showed that the system incurred in almost half of errors less than specialists, despite having been tested with a set of cases considered by specialists as difficult to classify.

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

Las técnicas de Minería de Datos dentro de la Ciencia de la Computación pueden ser usadas para resolver numerosos problemas inherentes a distintos campos de la ciencia y la tecnología, entre estas destacan las Ciencias Médicas y en particular las vinculadas con las enfermedades cardiovasculares.

La Minería de Datos aborda el estudio y tratamiento de grandes conjuntos de datos para extraer conclusiones e información relevante de ellos. Puede ser útil en prácticamente todas las facetas de la actividad humana. Es una disciplina que se está desarrollando cada vez con mayores capacidades gracias al avance de la tecnología y a la cada vez más alta capacidad de la computación de los ordenadores.

La cardiopatía isquémica constituye la primera causa de muerte en el mundo desarrollado y también en Cuba. Esta enfermedad se produce por la coexistencia de diversos factores de riesgo como la hipertensión, la diabetes, el colesterol alto, el hábito de fumar, la herencia, entre otros, que favorecen la obstrucción (total o parcial) de las arterias del corazón.

Múltiples son los complementarios de rutina que se utilizan en la práctica médica para el diagnóstico de esta enfermedad (electrocardiograma, pruebas de esfuerzo, ecocardiograma, tomografía), pero es la coronariografía la que se considera como la “prueba de oro”.

La coronariografía es la parte del cateterismo cardíaco cuyo objetivo es estudiar las arterias coronarias, mediante la inyección de contraste yodado.

Permite diagnosticar con precisión la enfermedad de las arterias del corazón y proceder, en muchos casos, a su tratamiento. Además de analizar el estado de las arterias coronarias, el cateterismo cardíaco permite estudiar y, en ocasiones, tratar las válvulas del corazón, así como diversas malformaciones [1], [2] . Se realiza para confirmar la sospecha de una enfermedad cardíaca de cualquier tipo, pues constituye la “prueba de oro” en Cardiología y cuando el diagnóstico es evidente (angina de pecho, infarto agudo de miocardio, enfermedad de las válvulas o del músculo

cardíaco), el cateterismo sirve para determinar la extensión y gravedad de la enfermedad [2].

La información que aporta esta prueba es fundamental para comprender la importancia de la enfermedad en cuestión y permite decidir el tratamiento más adecuado para cada enfermo. Así, en algunos pacientes, será suficiente el tratamiento medicamentoso y en otros, será necesario actuar directamente sobre las placas de ateroma de las arterias coronarias, las válvulas o las comunicaciones anormalmente existentes entre cavidades y vasos[1]–[3].

Dado el elevado coste del cateterismo cardíaco y conociendo los riesgos que entraña, esta prueba no se practica de forma sistemática cada vez que se diagnostica o se sospecha una cardiopatía. Se recomienda solo cuando es necesario confirmar la presencia de una posible enfermedad, definir su trascendencia anatómica y fisiológica, y determinar si se acompaña de otro proceso importante. Por ejemplo, cuando existen síntomas incapacitantes o progresivos de disfunción cardíaca o isquemia miocárdica, o datos objetivos (prueba de esfuerzo o ecocardiografía) que indiquen que el paciente corre un gran riesgo de sufrir un deterioro funcional rápido, un infarto de miocardio u otros acontecimientos adversos. En estas circunstancias, el cateterismo a menudo constituye el preludio del tratamiento mediante cirugía cardíaca o intervención con catéter. El cateterismo cardíaco constituye un instrumento clínico fundamental para evaluar la anatomía, fisiología y vascularización del corazón[4], [5].

La coronariografía es un procedimiento mínimamente invasivo y presenta una baja incidencia de complicaciones; sin embargo, cuando ocurren, pueden ser fatales.

Los riesgos se incrementan con la edad y fundamentalmente con la gravedad de la enfermedad de base. La edad mayor de 75 años, la diabetes, la insuficiencia renal y la enfermedad coronaria o valvular graves, incrementan los riesgos[6].

Como en cualquier intervención, se pueden producir una serie de efectos adversos:

• La complicación más grave es la muerte, cuya probabilidad es de uno a dos por cada 1000 procedimientos, asociada a pacientes con enfermedad coronaria grave o con múltiples patologías.

• Alergia a algunas de las sustancias empleadas. Las reacciones alérgicas, generalmente secundarias al contraste radiológico utilizado son infrecuentes y, la mayor parte de las veces, imprevisibles, salvo si se ha presentado reacción previa a la misma sustancia. En general, se trata de reacciones leves y de fácil manejo. Las reacciones graves son muy raras.

• Hipotensión y enlentecimiento del ritmo cardíaco (habitualmente pasajeros).

• Complicaciones vasculares locales, consistentes en hematoma o sangrado excesivo en el lugar de la punción. En esta zona puede quedar una leve molestia dolorosa, pero nunca dolor al caminar o en reposo, o algún bulto pulsátil doloroso. En esos casos es aconsejable consultarlo inmediatamente con el médico que realizó el procedimiento.

• Otras complicaciones más raras son: arritmias graves (que requieren choque eléctrico o marcapasos), accidentes cerebrovasculares (embolia cerebral), infarto agudo de miocardio o alteración del funcionamiento de los riñones.

En sus inicios este procedimiento era muy costoso, a veces imposible de afrontar por cualquier sistema de salud con bajos recursos económicos, debido al uso imprescindible de varios dispositivos como catéteres, guías y trócares[1], que tienen elevados precios en el mercado. Sin embargo, con el avance científico-tecnológico y el indiscutible beneficio de la prueba estos costos se han reducido, aunque siguen siendo elevados.

Una coronariografía, sin consecución de algún procedimiento terapéutico como lo es la angioplastia, cuesta como promedio en España entre 2500-3000 euros[7], en Estados Unidos de 7000-9000 dólares[8], en Inglaterra de 2000-3500 libras esterlinas[9], y en Cuba 875 CUC[2], incluyendo el costo de la habitación donde se ingresa al paciente.

Los pacientes que son remitidos a esta prueba son aquellos que más la necesitan, donde el beneficio, teóricamente, supera el riesgo; sin embargo, aún son muchos los pacientes que se someten a una coronariografía por sospecha de cardiopatía

[1] Trócar: Instrumento de cirugía que consiste en un punzón revestido de una cánula.
[2] CUC: Pesos cubanos convertibles

isquémica y el resultado es negativo. Es decir, se exponen a los pacientes a riesgos y se gastan recursos innecesarios.

En el Cardiocentro “Ernesto Che Guevara” de Santa Clara desde marzo del año 2003 (que se inició esta actividad de forma rutinaria) hasta septiembre de 2016, de todas las coronariografías realizadas sólo el 54,8% han sido positivas. Esto evidencia que casi la mitad de los pacientes se sometieron a un procedimiento que no necesitaban.

Desde hace varias décadas, se viene investigando en busca de alternativas que permitan predecir, con cierto grado de exactitud, el resultado que podría tener un determinado paciente. Hasta el momento existen muchas investigaciones publicadas donde se exploran las potencialidades de la tomografía axial computarizada (TAC) y la resonancia magnética nuclear (RMN); pero en la literatura no se ha encontrado ninguna propuesta basada en un modelo computacional que logre predecir el resultado de esta costosa y riesgosa prueba (a partir de los factores de riesgo, los síntomas y signos del paciente).

En el Cardiocentro, desde el año 2014, se viene instrumentando un Sistema de Apoyo a la toma de decisiones que –utilizando técnicas de Minería de Datos y Aprendizaje Automático– sea capaz de predecir el resultado de la coronariografía. La investigación desarrollada por Álvarez y Gutiérrez[10] tuvo un resultado satisfactorio pero no logró índices de exactitud como los que requiere una prueba diagnóstica de este tipo.

Atendiendo a lo anterior se identifica el siguiente **problema científico**: la coronariografía es una prueba con riesgo para la vida del paciente y costosa. Cerca de la mitad de los pacientes a los que se les realiza en el Cardiocentro “Ernesto Che Guevara” no la necesitan. Los especialistas de los Centro Diagnósticos no cuentan con los medios suficientes para determinar si un paciente realmente necesita esta prueba.

Objetivo general: Desarrollar un sistema de apoyo a la toma de decisiones para realizar coronariografías a partir de los datos de factores de riesgo, síntomas y signos de los pacientes que permita la reducción de los riesgos para la vida y costos, maximizando la probabilidad de recomendar a quien requiera esta prueba.

Interrogantes científicas:

1. ¿Qué elementos de la Minería de Datos hacen que sea apropiada para el diagnóstico de la cardiopatía isquémica?
2. ¿Qué técnicas de preprocesamiento de datos aplicar?
3. ¿Cuál es el modelo computacional de Minería de Datos que mejor discrimina entre pacientes con coronariografía positiva o negativa?
4. ¿Qué resultados se obtienen de la aplicación del modelo computacional en la práctica médica?

Objetivos específicos:

1. Fundamentar teórica y metodológicamente la Minería de Datos aplicada al diagnóstico de la cardiopatía isquémica.
2. Aplicar técnicas de preprocesamiento a los datos recolectados.
3. Seleccionar el modelo computacional de Minería de Datos que mejor discrimine entre pacientes con coronariografía positiva o negativa.
4. Evaluar la propuesta en la práctica médica.

Hipótesis: La utilización de las técnicas de Minería de Datos contribuirá a decidir sobre la conveniencia de realizar la coronariografía a los pacientes.

El informe consta de 3 capítulos. El primero presenta el marco teórico conceptual en el que se desarrolla la investigación. El segundo describe cómo se aplicaron las técnicas de preprocesamiento de los datos, guiados por las medidas de complejidad. El tercero muestra el trabajo de selección del modelo de clasificación que mejor se ajusta a los datos basado en medidas de exactitud y la evaluación del sistema en la práctica. Se ofrecen además conclusiones, recomendaciones, bibliografía y anexos.

CAPÍTULO I: ELEMENTOS DE MINERÍA DE DATOS

El Cardiocentro es un hospital regional de atención terciaria, lo que significa que los pacientes que son atendidos allí, fueron remitidos desde alguno de los ocho Centros de Diagnóstico de la Región Central que pertenecen a él. En este Centro existe un Servicio de Hemodinámica y Cardiología Intervencionista, en cuyo Salón se realiza la coronariografía de manera rutinaria.

En laboratorios de hemodinámica de los países desarrollados el porcentaje de positividad de la coronariografía es cercano al 70 %. La media en el mundo es de aproximadamente 60 %, pero en el Cardiocentro es menor; pues desde marzo del año 2003 (que se inició esta actividad) hasta septiembre de 2016, se han realizado en este centro 14050 coronariografías y solo 7695 han resultado positivas, lo que representa el 54,8 %.

En dicho Salón, se lleva un registro informático de todos los pacientes a los que se les ha hecho la prueba, a partir del año 2013, allí se guardan datos como: la edad, el sexo, las enfermedades que padecen, el resultado de los análisis complementarios que se les realizan, los antecedentes por los que se les realiza la prueba y su resultado.

Estos datos almacenados son sensibles a contener información valiosa para el dominio de procedencia, los pacientes están clasificados de acuerdo al resultado de su coronariografía, por lo que se puede crear una base de casos y, utilizando técnicas de Minería de Datos y Aprendizaje Automático, construir un sistema de apoyo para la toma de decisiones en este sentido.

En este capítulo se tratarán los referentes teóricos y metodológicos de la Minería de Datos, la Clasificación como técnica del Aprendizaje Automático Supervisado, las medidas de complejidad y de exactitud.

1.1. Minería de Datos

La Minería de Datos trata acerca de la solución de problemas mediante el análisis de los datos presentes en bases de datos reales. Es calificada hoy en día como la ciencia y tecnología para explorar datos y descubrir patrones desconocidos

presentes en ellos. En la literatura algunos distinguen a la Minería de Datos como sinónimo de Descubrimiento de Conocimiento en Bases de Datos (*KDD* por sus siglas en inglés), mientras que otros ven a la Minería de Datos como la etapa principal del *KDD*[11]–[13].

Existen varias definiciones de *KDD*. Por ejemplo, Fayyad, *et al*.[14] lo definen como "el proceso trivial de identificación de patrones válidos, nuevos, útiles y en última instancia comprensibles, en los datos", Friedman[15] considera el proceso de *KDD* como un análisis exploratorio automático de grandes bases de datos. Un aspecto clave que caracteriza al proceso de *KDD* es la forma en que se divide en etapas según el criterio de varios importantes investigadores del tema. Existen varios métodos disponibles para hacer esta división, cada uno con ventajas y desventajas[11].

1.1.1. Etapas del *KDD*

El proceso completo de extracción de conocimiento a partir de bases de datos, conocido como *KDD*, consta de varias etapas que transcurren desde la preparación de los datos hasta la presentación de los resultados obtenidos[16]. Dentro de estas etapas se encuentra la Minería de Datos, y actualmente engloba todas las etapas del proceso de *KDD* y sus técnicas se clasifican en dos grandes categorías: supervisadas[17], y no supervisadas[18]. Las técnicas supervisadas trabajan con conjuntos de datos en los que se conoce a *priori* la clase a la que pertenece cada ejemplo, en el caso de las no supervisadas los datos no se encuentran agrupados por clases.

Según García, *et al*.[19] las etapas del *KDD* son:

a) Especificación del problema: Diseño y organización del dominio de aplicación, el conocimiento previo relevante obtenido por expertos y los objetivos finales perseguidos por el usuario final.
b) Entendimiento del problema: Incluye la comprensión de los datos seleccionados y del conocimiento experto asociado con el objetivo de obtener un alto grado de fiabilidad.

c) Preprocesamiento de los datos: Esta etapa incluye operaciones para la limpieza de los datos (tales como el tratamiento y eliminación de ruido y de datos inconsistentes), integración (cuando múltiples fuentes de datos pueden ser combinadas en una sola), transformación (cuando los datos son transformados y consolidados a formas que son apropiadas para tareas de Minería de Datos específicas o para operaciones de agregación) y reducción, incluyendo la selección y extracción de atributos e instancias en la base de conocimientos.
d) Minería de Datos: Es el proceso en el que los métodos son usados para extraer patrones válidos a partir de los datos. Este paso incluye la elección de la técnica de Aprendizaje Automático (clasificación, regresión, clusterización o asociación), la elección del algoritmo de aprendizaje perteneciente a una de las familias anteriores. Finalmente el empleo y "acomodo" del algoritmo seleccionado al problema, mediante el ajuste de parámetros esenciales y proceso de validación.
e) Evaluación: Estimación e interpretación de los patrones encontrados basados en medidas de interés.
f) Explotación del resultado: En esta última etapa se puede usar directamente el conocimiento, incorporarlo a otro sistema para crear nuevos procesos o simplemente reportar el conocimiento descubierto a través de herramientas de visualización.

De las etapas antes expuestas la presente investigación se centró en la c, d y e puesto que las etapas a y b fueron abordadas en un trabajo precedente[10]. Se escogieron estas etapas porque, sabiendo que el problema está bien modelado y que se tiene un buen entendimiento de este, son las que pueden marcar una diferencia a la hora de mejorar la exactitud de una prueba diagnóstica.

1.1.2. Preprocesamiento de los datos

El preprocesamiento de datos puede llevar la mayor parte del tiempo del trabajo en una aplicación de Minería de Datos. Entre otras técnicas, el procesamiento de datos incluye la limpieza, transformación y reducción de los datos

Según Dorian Pyle, “El propósito fundamental de la preparación de los datos es la manipulación y transformación de los datos sin refinar para que la información contenida en el conjunto de datos pueda ser descubierta o estar accesible de forma más fácil”[20].

Esta etapa engloba a todas aquellas técnicas de análisis de datos que permiten mejorar la calidad de un conjunto de estos de modo que las técnicas de extracción de conocimiento puedan obtener mayor y mejor información.

Puede generar un conjunto de entrenamiento más pequeño que el original, lo cual mejoraría la eficiencia del proceso de Minería de Datos. Genera “datos de calidad”, los cuales pueden conducir a “patrones/reglas de calidad”.

1.2. Técnicas de preprocesamiento

Las técnicas de preprocesamiento, según García *et al*.[19] se dividen en dos grandes grupos: Preparación de los datos y Reducción de los datos.

1.2.1. Preparación de los datos

Se llama Preparación de los datos al conjunto de técnicas que inicializan los datos adecuadamente para servir como entrada para un determinado algoritmo. Es un paso obligatorio, convierte datos que previamente eran inútiles en nuevos datos. Si los datos no están preparados, el algoritmo de Minería de Datos puede no aceptarlos o reportar errores en tiempo de ejecución. En el mejor de los casos, el algoritmo puede funcionar, pero los resultados no tendrán sentido o no podrán ser considerados como conocimiento correcto.

Los procesos que pertenecen a esta familia de técnicas son:

- Limpieza de los datos.
- Transformación de los datos.
- Integración de los datos.
- Normalización de los datos.
- Imputación de valores perdidos.

- Identificación de ruido.

De estos, en el presente trabajo se utilizaron la Limpieza de Datos, porque se detectaron valores fuera de rango (*outliers*); la Normalización, porque uno de los atributos numéricos se había medido por dos medidas de unidad diferentes, y la Imputación de Valores Perdidos debido a la considerable presencia de estos en los datos.

1.2.1.1. Limpieza de los datos

La Limpieza de datos[21] incluye operaciones para corregir datos erróneos, filtrar los incorrectos fuera del conjunto de datos y reducir el detalle innecesario de estos. Es un concepto general que comprende o solapa otras conocidas técnicas de preparación de los datos. Contiene al tratamiento de valores perdidos y ruidosos.

Otras tareas de limpieza de datos involucran la detección de discrepancias y datos "sucios" (fragmentos de los datos originales que no tienen sentido). Estas últimas son tareas más relacionadas con la comprensión de los datos originales y por lo general requieren auditoría humana.

Las fuentes de datos "sucios" pueden ser errores en la entrada su entrada, en la actualización de los datos, en la transmisión e incluso mal funcionamiento del sistema de procesamiento. Como resultado, los datos "sucios" normalmente se presentan en dos formas: datos perdidos y datos erróneos.

1.2.1.2. Normalización de los datos

La normalización pretende darle a todos los atributos igual peso y es muy útil en métodos estadísticos de aprendizaje. La medida de unidad utilizada puede afectar el análisis de los datos. Todos los atributos deben estar expresados en la misma unidad de medida y deben usar una escala común o rango.

1.2.1.3. Imputación de valores perdidos

La Imputación de valores perdidos[22] es una forma de limpieza de datos cuyo propósito es rellenar las variables que contienen valores perdidos con datos intuitivos. En la mayoría de los casos adicionar un valor razonablemente estimado es mejor que dejar el espacio en blanco[19].

Muchos conjuntos de datos, tanto industriales como de investigación, contienen valores perdidos. Son introducidos debido a varias razones como son: procedimientos de entrada manual de datos, errores de equipamiento o mediciones incorrectas. Es usual encontrarlos en la mayoría de las fuentes de información usadas.

Los valores perdidos dificultan la realización de análisis en los datos. Tres tipos de problemas se asocian usualmente a valores perdidos[23]:

1. Pérdida de eficiencia.
2. Complicaciones en el manejo y análisis de los datos.
3. Sesgos resultantes de diferencias entre valores perdidos y completos.

Farhangfar *et al*.[24] resumen las 3 estrategias principales para manejar valores perdidos. La manera más simple es descartar las instancias que los contienen, no obstante, este método es práctico sólo cuando los datos portan una cantidad relativamente pequeña de instancias con valores perdidos y cuando el análisis de las instancias completas no cause grandes sesgos durante el proceso de inferencia. Otra estrategia es convertir los valores perdidos en un nuevo valor (codificarlos como un valor numérico), pero este método tan simple demostró causar serios problemas de inferencia. Por otro lado, si un número significativo de instancias contienen valores perdidos para un número relativamente pequeño de atributos, puede ser beneficioso realizar una imputación de los valores perdidos, este método consiste en rellenarlos con valores estimados a través de la identificación de relaciones entre los atributos[19].

1.2.2. Reducción de los datos

La Reducción de los datos comprende el conjunto de técnicas que de una manera u otra obtienen una representación reducida de los datos originales. Los datos producidos generalmente mantienen la estructura esencial y la integridad de los datos originales, pero la cantidad de datos disminuye. Se conoce que todo algoritmo tiene una complejidad en términos de tiempo, que depende de varios parámetros. En el aprendizaje automático, uno de esos parámetros es directamente proporcional al tamaño del conjunto de entrenamiento; por tanto, el uso de la reducción de datos es

crucial. En cuanto a otros factores tales como la disminución de la complejidad y la mejora de la calidad de los modelos resultantes, el rol de la reducción de datos es, otra vez, decisiva.

Técnicas de reducción de los datos:

- Selección de atributos.
- Selección de instancias.
- Discretización.
- Extracción de atributos y/o generación de instancias.

1.2.2.1. Selección de atributos

La Selección de atributos (*FS* por sus siglas en inglés)[25], [26] realiza la reducción del conjunto de datos eliminando atributos irrelevantes o redundantes. El objetivo de la *FS* es encontrar un conjunto de atributos mínimo de forma tal que la distribución de probabilidad resultante de las clases sea lo más cercana posible a la distribución original obtenida usando todos los atributos. Facilita la comprensión del patrón obtenido e incrementa la velocidad de la etapa de aprendizaje.

La *FS* puede ser considerada como un problema de búsqueda en el que cada estado del espacio de búsqueda se corresponde con un conjunto de atributos seleccionado. Se debe especificar una dirección de búsqueda, adoptarse diferentes estrategias para obtener subconjuntos óptimos y elegir un criterio de selección de los atributos.

Las técnicas de selección de atributos pueden ser categorizadas de acuerdo a varios criterios. Una categorización muy conocida y empleada utiliza los términos *filter* y *wrapper* para describir la naturaleza de la medida usada para evaluar el valor de los atributos[27]. Los métodos *wrapper* evalúan los atributos usando estimados de exactitud obtenidos a partir de un algoritmo de aprendizaje determinado. Por su parte, los *filter* usan características generales de los datos para evaluar atributos y trabajan independiente de cualquier algoritmo de aprendizaje. Otra taxonomía útil es la que divide a los algoritmos en aquellos que evalúan atributos individualmente y los que evalúan subconjuntos de atributos. Este último grupo puede subdividirse sobre la

base de la técnica de búsqueda usada con cada método para explorar el espacio de subconjuntos de atributos[28].

1.2.2.2. Selección de instancias

La Selección de instancias[19], [29] consiste en elegir un subconjunto del total de datos disponibles para lograr el propósito del sistema de aprendizaje automático, como si se hubiese usado el conjunto completo. Conforma la familia de métodos que realizan de forma inteligente la elección del mejor subconjunto posible, a partir de los datos originales usando algunas reglas y/o heurísticas. La selección aleatoria de las instancias es conocida como muestreo y está presente en un gran número de modelos Minería de Datos para llevar a cabo la validación interna y evitar el ajuste excesivo (*overfitting*).

Los métodos de selección de prototipos, son métodos de selección de instancias cuyo objetivo es encontrar conjuntos de entrenamiento que ofrezcan una mejoría en la exactitud de la clasificación. Usan clasificadores basados en instancias que toman como criterio una cierta similitud o medida de distancia.

En la literatura se distinguen dos tipos de algoritmos dentro de la Selección de Instancias: los de Selección de Prototipos y los de Selección del Conjunto de Entrenamiento[30].

Propiedades comunes en los métodos de Selección de Prototipos:

- Dirección de la Búsqueda: Puede ser Incremental, Decremental, por Bloques y Mixta.
- Tipo de selección: Condensación, Edición e Híbrida.
- Evaluación de la búsqueda: *Filter, Wrapper*.

1.3. Medidas de complejidad

Las medidas de complejidad analizan cómo las características intrínsecas de los datos afectan a los sistemas de aprendizaje. Debido a que el funcionamiento de estos sistemas depende de la distribución de los datos y de la representación del conocimiento usado, se ha prestado especial atención a los datos y a la estimación

cuantitativa de las diferentes fuentes de dificultad del problema para investigar su influencia en el funcionamiento de los algoritmos de aprendizaje[31].

La complejidad de los problemas de clasificación[31] puede ser atribuida a tres fuentes de aplicación: ambigüedad de las clases, límite de complejidad, y poca diversidad de ejemplos y la dimensión del espacio de características.

Medidas de superposición en los valores de las características de clases diferentes

Estas métricas evalúan el poder de atributos individuales para discriminar entre clases. Para cada atributo se examina el límite y se extienden los valores de las instancias de las diferentes clases, además se chequea el poder discriminativo de un atributo simple o de una combinación de ellos. Dentro de esta categoría se encuentran: La razón del discriminante máximo de Fisher (F1), la superposición de límites por clases (F2) y la máxima eficiencia de un atributo (F3).

Medidas de separabilidad de clases

Esta familia de métricas estiman cuan separables son las clases, examinando la distancia y la linealidad del límite de ellas. Dentro de esta familia están: la suma minimizada de la distancia de error de un clasificador lineal (L1), el error de entrenamiento de un clasificador lineal (L2), la fracción de puntos en el límite de las clases (N1), la razón de distancia promedio al vecino más cercano intra/inter clases (N2) y la tasa de error de sacar un clasificador de un vecino cercano (N3).

Medidas de geometría, topología, y densidad de las colecciones

Evalúan la superposición entre clases y cómo estas están distribuidas como hiperesferas en el espacio de atributos. Esta categoría comprende las siguientes medidas: la no linealidad de los clasificadores lineales (L3), la no linealidad del primer clasificador más cercano (N4), la fracción del máximo dominio de cobertura (T1), y el número promedio de puntos por dimensión (T2).

En la presente investigación se utiliza la Metodología Propuesta por Bernadó-Mansilla y Macià-Antolínez[32] para transformar un problema reduciendo su complejidad, para que los modelos de clasificación extraídos sean más precisos.

Para analizar si cada transformación es apropiada se analizan medidas que aproximan la complejidad geométrica del conjunto de datos. Utilizando estas medidas se puede estimar la complejidad intrínseca del conjunto de datos sin estar atados a ningún clasificador en particular.

Del conjunto de métricas propuestas por Ho y Basu[31] las que se utilizan en esta metodología son:

- La razón del discriminante máximo de Fisher (F1): para cada atributo la se calcula como: $f = (\mu_1 - \mu_2)^2 / (\sigma_{12} + \sigma_{22})$, donde μ_1, μ_2 y σ_{12}, σ_{22} son la media y la varianza del atributo para cada una de las dos clases respectivamente. Con esta métrica el atributo más discriminante es el que tiene el valor máximo de Fisher.

- La superposición de límites por clases (F2): la región de superposición de un atributo es calculada como el rango de superposición dividido por el rango total de ese atributo. F2 es el producto de las regiones de superposición de cada atributo.

- La máxima eficiencia de un atributo (F3): describe cuánto contribuye cada atributo a la separación entre clases. Consiste en eliminar las instancias ambiguas para cada atributo. La eficiencia de cada atributo es la razón entre los puntos restantes no superposicionados y el número total de puntos. La eficiencia de atributo más alta es tomada como el valor de F3.

- La fracción de puntos en la frontera de la clase (N1): esta métrica está basada en la prueba propuesta por Friedman y Rafsky en 1979[33]. Primeramente se genera el árbol de extensión mínimo conectando todos los ejemplos de entrenamiento usando la distancia euclidiana entre cada par de puntos. Luego se calcula la fracción de puntos que unen a clases diferentes entre el número total de puntos. Esta medida es sensible a la separabilidad entre clases y a la tendencia a clusterización de puntos pertenecientes a la misma clase.

- La razón de distancia promedio al vecino más cercano intra/inter clases (N2): para cada punto se calcula la distancia a su vecino más cercano perteneciente a su misma clase y al vecino más cercano que pertenezca a la clase contraria. Finalmente

se calcula la razón de la suma de las distancias dentro de una clase a la suma de las distancias entre clases para cada punto.

- La no linealidad del primer clasificador más cercano (N4): esta medida, propuesta por Hoekstra y Duin en 1996[34], una vez determinado el conjunto de datos de entrenamiento, crea un conjunto de prueba por interpolación lineal con un coeficiente aleatorio con pares de instancias aleatoriamente seleccionadas de la misma clase. Retorna el error de prueba del clasificador 1-NN después de usar el conjunto de datos original como entrenamiento.

N4 se extiende en el intervalo [0,1]. Valores elevados de esta medida expresan una alta interpolación entre clases.

Terminado el preprocesamiento se pasa a la etapa donde tiene lugar el aprendizaje, como se apreció anteriormente de las distintas técnicas de Aprendizaje Automático Supervisado, se utilizará la clasificación porque los datos obtenidos se encuentran etiquetados en dos clases: positiva y negativa.

1.4. Clasificación

El objetivo de la clasificación es construir un modelo de la distribución de las clases a partir de los atributos. El modelo resultante se utiliza para asignar valores de la clase a una base de datos en la que se conocen los valores de los atributos, pero el valor de la clase es desconocido. La clasificación tiene una amplia gama de aplicaciones, incluyendo experimentos científicos, diagnóstico médico, detección de fraudes y aprobación de créditos[35]. En la literatura se han propuesto muchos modelos de clasificación: redes neuronales[36], [37], algoritmos genéticos[38], métodos bayesianos[39], modelos lineales-logarítmicos y otros métodos estadísticos[40], [41], tablas de decisión[42], y modelos estructurados, también llamados árboles de clasificación[43], [44].

Para usar un algoritmo de clasificación, primeramente hay que entrenarlo, para esto se utiliza un conjunto de ejemplos de entrenamiento, que es conocido como: conjunto de entrenamiento. En los algoritmos de aprendizaje automático el objetivo es predecir la variable objetivo, en la clasificación esta variable adquiere un valor

nominal y su valor es conocido. El aprendizaje consiste en buscar una relación entre los atributos y la variable objetivo.

En los problemas de clasificación, a la variable objetivo se le llama clase, y el número de clases debe ser finito.

Para probar la calidad del modelo de clasificación, usualmente se tiene un conjunto de entrenamiento y otro conjunto de datos por separado, llamado conjunto de prueba. Inicialmente, el algoritmo se "alimenta" con los ejemplos de entrenamiento; aquí es donde tiene lugar el aprendizaje automatizado, luego se le proporciona el conjunto de prueba. El valor de la clase para cada ejemplo del conjunto de prueba no se le da al algoritmo y este decide a qué clase debe pertenecer cada ejemplo. Luego se compara la clase a la que pertenece el ejemplo de entrenamiento con el valor predicho para tener una idea de cuán exacto es el algoritmo.

La forma de ver qué aprendió el sistema es examinando la forma de representación del modelo. Algunos algoritmos tienen una forma de representación que es más entendible que otros. Este puede tomar la forma de un conjunto de reglas, una distribución de probabilidad o un ejemplo del conjunto de entrenamiento. En algunos casos el interés puede no ser la construcción de un sistema experto, sino sólo la forma de representación del modelo obtenida tras el entrenamiento de un algoritmo de aprendizaje automático[45].

1.4.1. Modelos de Clasificación

- Redes Neuronales Artificiales (*ANNs* por sus siglas en inglés): Son poderosos modelos matemáticos apropiados para casi todas las tareas de Minería de Datos, en especial las predictivas[36]. Existen diferentes formulaciones de *ANNs*, las más comunes son *Multi-Layer Perceptron, Radial Basis Function Networks* y *Learning Vector Quantization (MLP, RBFNs* y *LQV,* respectivamente, por su siglas en inglés*)*. Las *ANNs* están basadas en la definición de neuronas, en estas cada neurona recibe una serie de entradas a través de interconexiones y emite una salida que se va a transmitir idéntica a múltiples neuronas posteriores[46]. Usualmente tienen mejor desempeño que los demás modelos debido a su compleja estructura, no obstante, la complejidad y la adecuada configuración de estas redes hacen que no sean muy

populares en comparación con otros métodos; se les considera el típico ejemplo de modelos de caja negra. Al igual que en los modelos de regresión, requieren atributos numéricos y no manejan valores perdidos. Por otro lado, si están correctamente configurados, son robustos frente a valores extremos y a datos ruidosos.

- Aprendizaje Bayesiano: Usa la teoría de la probabilidad para tomar decisiones racionales bajo incertidumbre basadas en el Teorema de Bayes[47]. El método bayesiano más aplicado es Naïve Bayes, el cual asume que el efecto del valor de un atributo de una clase determinada es independiente de los valores de otros atributos. Las versiones iniciales de estos algoritmos sólo trabajan con atributos categóricos, porque el cálculo de la probabilidad sólo puede hacerse en dominios discretos. Por otro lado, el hecho de asumir independencia entre los atributos trae como consecuencia que estos métodos sean muy sensibles a la redundancia e inutilidad de alguno de los atributos y ejemplos, a la presencia de valores perdidos, de ruido y también de valores extremos. Además de Naïve Bayes hay modelos complejos basados en estructuras de dependencia como, por ejemplo, las Redes Bayesianas.

- Aprendizaje basado en instancias: En este tipo de aprendizaje, se almacenan los ejemplos de entrenamiento textualmente y cuando se quiere asignar la clase a un nuevo ejemplo, se extraen las instancias más parecidas y se usa su distribución de clases para clasificar al nuevo. En este esquema, el proceso de aprendizaje es trivial y el de clasificación es el que consume el mayor tiempo. Este tipo de aprendizaje también se conoce como *lazy learning*[48] o *memory-based learning*, donde los datos de entrenamiento se procesan solo hasta que se requiere (cuando se requiere contestar alguna pregunta), y la relevancia de los datos se mide en función de una medida de distancia[49]. La diferencia entre ellos radica en la función de distancia utilizada, el número de ejemplos tomados para hacer la predicción, su influencia cuando se usan mecanismos de asignación de pesos o votos y el uso de algoritmos eficientes para encontrar los ejemplos más cercanos, como *KD-Trees* o esquemas *hashing.* Tiene algunas desventajas tales como, requiere gran espacio de almacenamiento, baja eficiencia a la hora de predecir una respuesta y baja tolerancia al ruido.

- Máquinas de Soporte Vectorial (*SVM* por sus siglas en inglés): Estas aprenden la superficie de decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un *kernel* Gaussiano o de otro tipo a un espacio de características en un espacio dimensional más alto, donde se busca la máxima separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada, puede separar los datos en todas las clases distintas, cada una formando un agrupamiento[50]. Están basadas en la teoría del aprendizaje estadístico[51], se asemejan a las *ANNs* en que son usadas para la estimación y tienen buen rendimiento cuando los datos son linealmente separables. Requieren datos numéricos, sin valores perdidos y son robustos frente a ruido y a valores extremos.

- Aprendizaje de reglas: También conocido como divide y vencerás o *covering rule algorithms*[52], este tipo de métodos busca una regla que explique alguna parte de los datos, separa esos ejemplos y recursivamente "vence" los ejemplos restantes. Desde el punto de vista del preprocesamiento de datos, por lo general requiere datos nominales o discretizados (aunque esta tarea está implícita en la mayoría de los algoritmos) y tiene un selector intrínseco de atributos interesantes dentro de los datos. Sin embargo los valores perdidos, extremos y los ejemplos ruidosos, pueden perjudicar el desempeño del modelo final. Algunos ejemplos de este tipo de algoritmos son: *AQ, CN2, RIPPER, PART* y *FURIA*.

- Árboles de decisión: Son modelos predictivos formados por iteraciones del esquema divide y vencerás de decisiones jerárquicas[53]. El procedimiento que usan es intentar dividir los datos en subgrupos homogéneos usando una de las variables independientes. La forma final de árbol puede ser traducida a un conjunto de reglas *If-Then-Else* desde la raíz hasta cada uno de los nodos hoja. Son similares a los métodos de aprendizaje de reglas y tienen las mismas desventajas. Los árboles de decisión más conocidos son: *CART, C4.5* y *PUBLIC*.

- Metaclasificadores: La estrategia de estos algoritmos para tomar decisiones más certeras es combinar la salida de diferentes modelos, creando un "ensamblado" de modelos y usando esta combinación: ejemplos prominentes de estos esquemas son *bagging, boosting* y *stacking*. Estas técnicas generalmente obtienen mejores resultados que los algoritmos pertenecientes a las categorías anteriores.

- Aprendizaje basado en costos: Optimizar la exactitud de la clasificación sin tener en cuenta el costo de los errores generalmente trae como consecuencia extraños resultados. Si no se especifican los costos, el algoritmo asume que son iguales. Existen dos variantes para tener en cuenta los costos, la clasificación sensible a costos, que utiliza una matriz donde se especifican estos y calcula los costos de un modelo de aprendizaje en particular para un conjunto de datos asumiendo los elementos de esta matriz para cada una de las instancias; aquí los costos son ignorados durante el entrenamiento y son tomados en cuenta en el proceso de predicción. La otra es el aprendizaje sensible a costos, esta es todo lo contrario de la anterior, toma en cuenta la matriz durante el proceso de aprendizaje e ignora los costos en el proceso de predicción[13].

Algoritmos pertenecientes a las categorías anteriores fueron los utilizados para encontrar cuál era el modelo que mejor se ajustaba a los datos, según las medidas que se exponen a continuación.

1.5. Medidas para evaluar la calidad del aprendizaje automatizado

Las medidas que se utilizaron para evaluar la calidad del aprendizaje son las que se usan para evaluar la exactitud de una prueba diagnóstica. Algunas de estas medidas son análogas de las del aprendizaje automatizado, pero se nombran de otra manera.

1.5.1. Medidas para evaluar la exactitud de una prueba diagnóstica

Los estudios de exactitud diagnóstica tienen una estructura común básica, en términos generales, pueden corresponder a estudios de tipo caso-control, transversales o de cohorte. En este tipo de estudios los resultados obtenidos con la prueba diagnóstica se comparan con los de un estándar de referencia en un mismo grupo de pacientes. El estándar de referencia, también llamado *gold standard*

(estándar de oro) corresponde a la mejor manera disponible y ampliamente aceptada para establecer la presencia o ausencia de determinada condición. El término exactitud se refiere precisamente a la concordancia entre los resultados de la prueba diagnóstica con el estándar de referencia[54].

La utilidad de las pruebas diagnósticas generalmente se describe y/o cuantifica en términos de su sensibilidad, especificidad, valor predictivo positivo, valor predictivo negativo y *likelihood* ratios (razones de verosimilitud) positivo y negativo[55].

1.5.2. Matriz de confusión

Esta matriz se compone de dos filas que corresponden al resultado dicotómico positivo o negativo (presencia o ausencia) de la enfermedad o condición, según el *gold standard,* y en las columnas según nuestra prueba diagnóstica. Forman cuatro celdas donde cada una de ellas corresponde a verdadero positivo, falso negativo, falso positivo y verdadero negativo, respectivamente. Estos términos se definen como:

- Verdadero positivo: El paciente tiene la enfermedad y la prueba es positiva.
- Falso negativo: El paciente tiene la enfermedad, pero el resultado de la prueba es negativo.
- Verdadero negativo: El paciente no tiene la enfermedad y la prueba es negativa.
- Falso positivo: El paciente no tiene la enfermedad, pero el resultado de la prueba es positivo.

Tabla 1. Matriz de confusión

		CLASE PREDICHA	
		SÍ	**NO**
CLASE REAL	**SÍ**	**Verdaderos positivos**	**Falsos negativos**
	NO	**Falsos positivos**	**Verdaderos negativos**

1.5.3. Sensibilidad y especificidad

Sensibilidad: Corresponde a la proporción de individuos correctamente diagnosticados con la condición o enfermedad por la prueba diagnóstica (Tabla I). En otras palabras la proporción de verdaderos positivos, correctamente identificados por el test, del total de individuos enfermos según el estándar de referencia[56]. Su fórmula de cálculo es la siguiente:

$$\boldsymbol{Sensibilidad} = \frac{\boldsymbol{VP}}{\boldsymbol{VP} + \boldsymbol{FN}}$$

Especificidad: Corresponde a la proporción de individuos correctamente diagnosticados con ausencia de la condición o enfermedad por la prueba diagnóstica en estudio. Es la proporción de verdaderos negativos que fueron correctamente identificados por el test, del total de individuos sanos según el estándar de referencia[56]. De lo anterior podemos inferir que la especificidad es el cociente entre los verdaderos negativos dividido por la suma de verdaderos negativos y falsos positivos. Su fórmula sería:

$$\boldsymbol{Especificidad} = \frac{\boldsymbol{VN}}{\boldsymbol{FP} + \boldsymbol{VN}}$$

Estas proporciones son parámetros inherentes a la prueba diagnóstica, menos dependientes de la prevalencia de la enfermedad[57].

Una buena prueba diagnóstica es la que ofrece alta sensibilidad y especificidad, pero esto no siempre se cumple. En este estudio, al ser en una población enferma, se busca una alta sensibilidad, ya que se pretenden evitar los falsos negativos; es decir, los pacientes que realmente están enfermos a los que la prueba diagnóstica clasifica como no candidatos a coronariografía.

1.6. Herramientas de Minería de Datos

KEEL (*Knowledge Extraction based on Evolutionary Learning*) [58], es un software de código abierto que puede ser usado para un gran número de tareas de *KDD* diferentes. Provee una *GUI (Graphical User Interface)* simple, basada en un flujo de datos para diseñar experimentos con diferentes conjuntos de datos y algoritmos

aprendizaje automático (prestando especial atención a los algoritmos evolutivos) con el objetivo de evaluar el comportamiento de estos. Contiene gran variedad de algoritmos de extracción de conocimientos clásicos, técnicas de preprocesamiento, algoritmos de aprendizaje basados en inteligencia computacional, modelos híbridos, metodologías estadísticas para contrastar experimentos, entre otros.

WEKA (*Waikato Environment for Knowledge Analysis*)[59] es una herramienta de aprendizaje automatizado [5], programada totalmente en Java. Es un ambiente de trabajo para la prueba y validación de algoritmos de la Inteligencia Artificial. Tiene implementada una colección de algoritmos conocidos, varias maneras para preprocesar los archivos de datos a utilizar por dichos algoritmos; así como facilidades para validar los mismos. Posee interfaces gráficas de usuario *(GUI)* y cuenta con herramientas para realizar tareas de regresión, clasificación, agrupamiento, asociación y visualización.

1.7. Conclusiones parciales

- Las técnicas de Minería de Datos son ampliamente utilizadas en múltiples áreas, incluyendo la medicina.
- Los datos recopilados, sobre los pacientes atendidos en Salón de Hemodinámica del Cardiocentro, pueden aportar información valiosa para este dominio a través de la utilización de técnicas de Minería de Datos.
- Aplicar técnicas de preprocesamiento a los datos debe contribuir a la disminución de la complejidad del conjunto de datos.
- Las medidas de exactitud apoyan la elección del mejor modelo de clasificación.

CAPÍTULO II: PREPROCESAMIENTO

Los datos recolectados como parte del proceso de *KDD* usualmente contienen valores erróneos o ausentes debido a múltiples causas, que dificultan el entrenamiento de un algoritmo de aprendizaje automático y por ende su futuro desenvolvimiento. El presente capítulo trata acerca del preprocesamiento, de una de las etapas del *KDD* que aborda estas dificultades.

El preprocesamiento contiene un conjunto de técnicas cuyo objetivo es dejar al conjunto de datos es un estado óptimo para la aplicación de un algoritmo de aprendizaje. Para guiar estas transformaciones se van a utilizar medidas de complejidad de los datos que van a ir indicando cuál es la mejor decisión a tomar en cada paso. En la Figura 3 se muestra el flujo de trabajo de este capítulo.

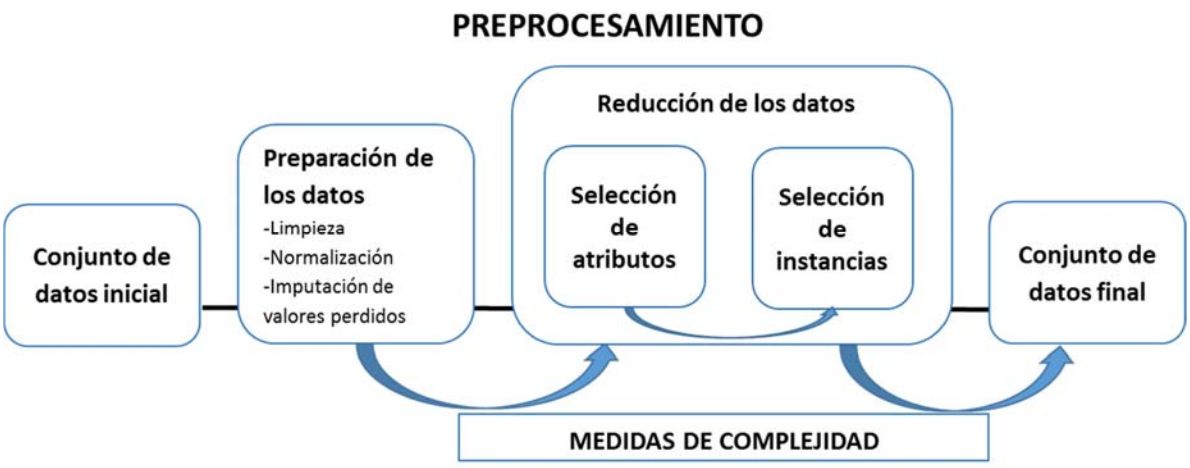

Figura 1. Etapas del preprocesamiento guiadas por las medidas de complejidad.

2.1. Construcción de la base de casos

Se ejecutaron varias tareas para construir la base de conocimientos o conjunto de entrenamiento que se utilizará en las etapas posteriores.

2.1.1 Recolección inicial de los datos

Se tomaron los datos de una aplicación desarrollada en Microsoft Office Access, allí se encontraban almacenados los datos de los pacientes a los que se les había realizado la coronariografía, con resultado positivo o negativo, atendidos de 2013 a 2015.

Estos presentaban muchos valores perdidos y conociendo que toda la información de los pacientes, se archiva en sus historias clínicas, se trabajó directamente con estas para completar dichos valores, pasándolos a un archivo Excel.

2.1.2 Descripción de los atributos recolectados

En la Tabla 2 se presentan los atributos recolectados en su formato original. Estos tuvieron que ser tratados para poder formar con ellos una Base de Casos (BC) coherente y consistente con la que se pudiera trabajar a lo largo del proyecto.

Tabla 2. Descripción de los atributos en su formato original.

Atributo	Descripción	Tipo
HC	Número de la historia clínica del paciente que ingresa al Cardiocentro.	Número entero
Edad	Edad del paciente que ingresa al Cardiocentro.	Número entero
Sexo	Sexo del paciente que ingresa al Cardiocentro (Masculino o Femenino).	Categoría (M, F)
Provincia	Provincia a la que pertenece el paciente que ingresa al Cardiocentro.	Categoría (una para cada provincia)
HTA	Hipertensión arterial. Rasgo que guarda si el paciente que ingresa al Cardiocentro es hipertenso (Y o N).	Categoría (Sí, No)
DM	Diabetes mellitus. Rasgo que guarda si el paciente que ingresa al Cardiocentro es diabético (Y o N).	Categoría (Sí, No)
Tabaco	Rasgo que guarda si el paciente que ingresa al Cardiocentro es fumador (Y o N).	Categoría (Sí, No)
Dislipidemia	Rasgo que guarda si el paciente que ingresa al Cardiocentro presenta	Categoría (Sí, No)

	alteración en el metabolismo de los lípidos (Y o N).	
ERC	Enfermedad renal crónica. Rasgo que guarda si el paciente que ingresa al Cardiocentro presenta una enfermedad renal (Y o N).	Categoría (Sí, No)
SCA	Síndrome coronario agudo. Rasgo que guarda si el paciente que ingresa al Cardiocentro presenta algún síndrome coronario.	Categoría (descrita más adelante)
Topografía	Rasgo que guarda la topografía del infarto del paciente que ingresa al Cardiocentro.	Categoría (descrita más adelante)
Ainest	Angina Inestable. Rasgo que guarda si el paciente que ingresa al Cardiocentro presenta una angina inestable (Y o N).	Categoría (Si, No)
Tipo AI	Tipo de angina inestable que presenta el paciente que ingresa al Cardiocentro.	Categoría ordinal (descrita más adelante)
Aestable	Tipo de angina estable que presenta el paciente que ingresa al Cardiocentro.	Categoría ordinal (descrita más adelante)
TAS	Tensión arterial sistólica del paciente que ingresa al Cardiocentro.	Número continuo
TAD	Tensión arterial diastólica del paciente que ingresa al Cardiocentro.	Número continuo
FC	Frecuencia del paciente que ingresa al Cardiocentro.	Número continuo
Peso	Peso del paciente que ingresa al Cardiocentro.	Número continuo
Talla	Talla del paciente que ingresa al Cardiocentro.	Número entero
HB	Hemoglobina del paciente que ingresa al Cardiocentro.	Número entero
Tcoag	Tiempo de coagulación del paciente que ingresa al Cardiocentro.	Número continuo
Tsang	Tiempo de sangramiento del paciente que ingresa al Cardiocentro.	Número continuo

Plaq	Conteo de plaquetas del paciente que ingresa al Cardiocentro.	Número entero
Creat	Valor de la creatinina del paciente que ingresa al Cardiocentro.	Número continuo
Glic	Valor de la glicemia del paciente que ingresa al Cardiocentro.	Número continuo
Clase	Valor que determina si la coronariografía del paciente que ingresa al Cardiocentro es positiva o negativa.	Categoría (Positivo, Negativo)

2.2. Limpieza de los datos

Para aumentar la calidad de los atributos, se analizó la base de casos en busca de datos con ruido o irrelevantes. En este trabajo esta operación se realizó manualmente teniendo en cuenta la opinión de los especialistas.

2.2.1. Atributos irrelevantes

Debido a que algunos atributos son de carácter irrelevante para la aplicación de la Minería de Datos, se procedió a descartar estos de los atributos finales:

- HC
- Provincia
- Ainest
- TAS
- TAD
- FC

Se descartó el atributo HC porque es un valor que sirve de identificación para el paciente y no aporta ninguna información. Con respecto a la Provincia, la distribución de pacientes por provincia no es proporcional, pues debido a roturas o falta de algún insumo, en ocasiones se priorizan los pacientes de Villa Clara o de provincias más cercanas por la inmediatez con la que pueden presentarse en el Cardiocentro a ingresar, por tanto si se tuviera en cuenta podría confundir al sistema. La información del atributo Ainest se recoge en TipoAI, por lo que se decidió descartarlo. Los valores de la TAS, TAD y FC, por sus características tienen más importancia cuando se le

evalúa a través del tiempo, este único valor que es resultado de la toma de la presión arterial y de la frecuencia cardíaca al paciente cuando se le remite a la prueba, no aportan información relevante al sistema.

Teniendo en cuenta que el atributo ERC permanecía constante para todos los pacientes y no aportaban información al proceso de Minería de Datos, se decidió descartarlo.

Luego del proceso de selección de los atributos relevantes, se obtuvo la siguiente lista de 19 atributos a tener en cuenta para la Minería de Datos (Tabla 3):

Tabla 3. Atributos seleccionados para la Minería de Datos.

Atributo
Edad
Sexo
HTA
DM
Tabaco
Dislipidemia
SCA
Topografía
TipoAI
Aestable
Peso
Talla
HB

Tcoag
Tsang
Plaq
Creat
Glic
Clase

2.2.2. Atributos mal categorizados

Como resultado del análisis se detectaron atributos mal categorizados en la aplicación de la que se obtuvieron los datos.

Atributo SCA:

- SCACEST (Síndrome coronario agudo con elevación del ST)
- SCASEST (Síndrome coronario agudo sin elevación del ST)
- Angina Inestable
- AI con CE (Angina Inestable con Cambios Eléctricos)
- AI sin CE (Angina Inestable sin Cambios Eléctricos)
- BRIHH Agudo
- Trombosis Intrastent
- Angina Vasoespástica

Para este se definieron solo dos categorías (SCACEST y SCASEST), eliminando las instancias (14 en total) que presentaran BRIHH Agudo y Trombosis Intrastent que realmente no pertenecían a esta categoría, las demás caracterizaciones se tomaron como valores perdidos y en caso de que no hubiera ningún valor seleccionado se colocó el valor de No Presenta, quedando de la siguiente forma:

SCA:

- SCACEST

- SCASEST
- No presenta

El atributo TipoAI presentaba las siguientes categorías:

- Mixta
- De reciente comienzo
- De empeoramiento progresivo
- Post Infarto
- De reposo

Según el criterio de los expertos médicos la angina Post Infarto es un tipo de angina en reposo, por lo que estos casos se convirtieron en De reposo. La angina mixta desde hace varios años no es un término aceptado por lo que se decidió que en los casos donde esta se contemplaba (57) se pusieran valores perdidos y en caso de que no hubiera ningún valor seleccionado se colocó el valor de No Presenta, finalmente quedó de esta forma:

TipoAI:

- No presenta
- De reciente comienzo
- De empeoramiento progresivo
- De reposo

Los valores del atributo TipoAI tienen un orden creciente, de 0 a 3, lo que denota la gravedad de este padecimiento, estos valores se corresponden con las siguientes categorías:

- No presenta -> 0
- De reciente comienzo -> 1
- De empeoramiento progresivo -> 2
- De reposo -> 3

Atributo Aestable:

El atributo aestable tiene las mismas características del anterior, pero en este caso las categorías son:

- No presenta -> 0
- Grado I -> 1
- Grado II -> 2
- Grado III -> 3
- Grado IV -> 4

Atributo Topografía:

Este atributo tiene las categorías que se muestran a continuación:

- InferoLateral
- InferiorVD
- Inferior
- AnteriorExtenso
- AnteroSeptal
- Anterior
- No presenta

Atributo clase: Según el criterio de los especialistas, la coronariografía de un paciente se considera positiva, cuando al menos en una de las arterias del corazón existe una obstrucción del más del 50 por ciento, en el caso de la arteria TCI (Tronco coronario izquierdo) debe ser más de un 30 por ciento. En caso contrario esta sería negativa, por lo que la creación de la clase se basa en este criterio, teniendo dos categorizaciones: positiva o negativa.

2.2.3. Valores fuera de rango

Se establecieron rangos de valores normales[60] para cada atributo numérico y de acuerdo a este se eliminaron errores cometidos a la hora de entrar los datos. Se

decidió trabajar de esta forma porque por ejemplo, puede existir un valor que una técnica computacional detecte como *outlier,* pero en realidad sea un valor normal para un paciente que se encuentra bajo algún tipo de tratamiento o que tenga algún parámetro “descompensado” a la hora de realizarle los análisis complementarios.

Se consideraron los valores fuera de rango como valores perdidos. La Tabla 4 muestra, para cada atributo, la cantidad de ejemplos en el conjunto de entrenamiento que tienen valores perdidos debido a valores erróneos encontrados. Existen, en el conjunto de entrenamiento, valores perdidos por otras causas.

Tabla 4. Valores erróneos por cada atributo.

Atributo	Cantidad
Glicemia	3
Creatinina	5
Plaquetas	2

Este trabajo de limpieza fue muy importante porque los valores eliminados eran *outliers,* podían confundir al modelo a la hora de aprender, además muchos de los algoritmos de aprendizaje son sensibles a valores de este tipo.

En total son 772 instancias, 467 positivas y 305 negativas.

2.3. Normalización de los datos

En el conjunto de datos se presentó el caso de que el valor de la hemoglobina en algunos casos se medía en g/dL y en otros en g/L. Hace varios años que el formato utilizado en Cuba es este último, pero algunos médicos siguen usando la primera unidad de medida, por lo que se procedió a normalizar todos estos datos. Esta tarea también, por sus características, se realizó de forma manual.

De un total de 772 pacientes, 609 tenían los valores de hemoglobina según el formato antiguo, lo cual creaba un sesgo en el conjunto de datos. La proporción de estos valores se muestra en el Gráfico 1.

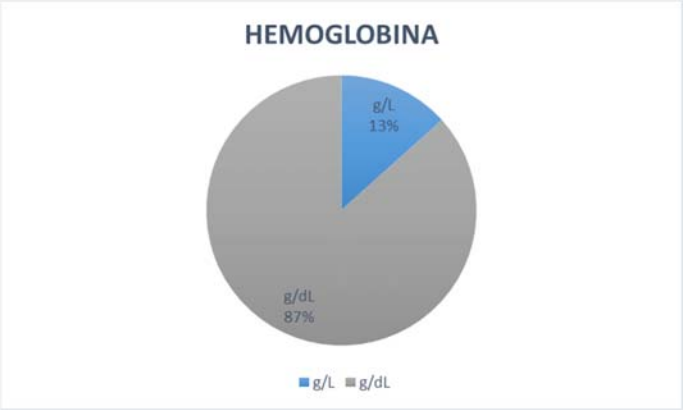

Gráfico 1. Proporción de valores con unidad de medida errónea.

Hasta este punto se contaba con un conjunto de datos cuyas medidas de complejidad se comportaban como se muestra en el Gráfico 2.

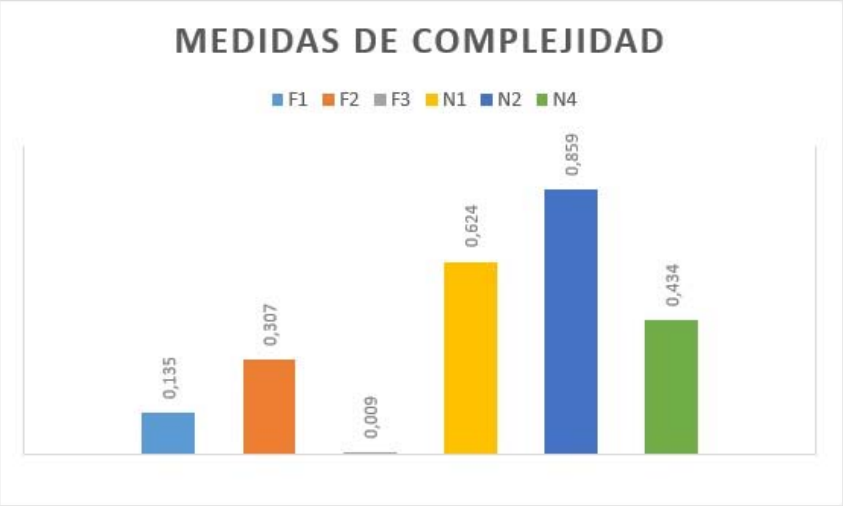

Gráfico 2. Medidas de complejidad del conjunto de entrenamiento luego de realizadas la Limpieza y Normalización de los datos.

2.4. Tratamiento de valores perdidos

Para el tratamiento de los valores perdidos en el conjunto de datos se utilizó la estrategia de imputación y se realizó con los algoritmos que provee la herramienta KEEL. Una ventaja fundamental de este método es que el manejo de los valores perdidos es independiente del algoritmo de aprendizaje utilizado. Esta brinda 14 algoritmos para tratar este problema:

1. *Do Not Impute (DNI):* no se reemplaza ningún valor perdido, por lo que el algoritmo de aprendizaje debe usar sus propias estrategias para lidiar con ellos[61].

2. Case Deletion or Ignore Missing (IM): todas las instancias con al menos un valor perdido son eliminadas del conjunto de datos.

3. *Global Most Common Attribute Value for Symbolic Attributes, and Global Average Value for Numerical Attributes (MC):* para atributos nominales los valores perdidos son reemplazados con el valor más común para ese atributo, y los valores numéricos son reemplazados con el promedio de todos los valores del atributo correspondiente[62].

4. *Concept Most Common Attribute Value for Symbolic Attributes, and Concept Average Value for Numerical Attributes (CMC):* como en el caso anterior, se reemplazan los valores perdidos por el más repetido si los atributos son nominales o el valor promedio si son numéricos, pero considerando sólo las instancias con la misma clase que la instancia de referencia[62].

5. *Imputation with K-Nearest Neighbour (KNNI):* es un algoritmo basado en instancias, cada vez que encuentra un valor perdido en un instancia, calcula sus *k* vecinos más cercanos e imputa un valor a partir de ellos. El valor más común para atributos nominales y el promedio para los numéricos. Como medida de proximidad usa la distancia euclidiana[63].

6. *Weighted Imputation with K-Nearest Neighbour (WKNNI)*: selecciona las instancias con valores similares (en términos de distancia) a una instancia determinada, para poder imputar como lo hace *KNNI.* No obstante, el valor estimado ahora toma en cuenta las diferentes distancias a los vecinos, usando una mediana "pesada" o el valor más repetido de acuerdo a la distancia[64].

7. *K-Means Clustering Imputation (KMI):* En este algoritmo la diferencia intra-clúster es medida como la adición de las distancias entre los objetos y el centroide del clúster al que están asignadas. El centroide de un clúster representa el valor medio de los objetos en el clúster. Una vez que converjan los clústeres, el proceso final es rellenar todos los atributos sin referencia para cada objeto incompleto basándose en la información del clúster. Los objetos que pertenecen al mismo clúster son tomados

como vecinos más cercanos entre si y se aplica el algoritmos KNN para reemplazar los datos perdidos[73].

8. *Imputation with Fuzzy K-Means Clustering (FKMI):* en este método cada objeto x_i tiene una función de pertenencia que describe el grado en el que un objeto pertenece a determinado clúster v_k. En el proceso de actualizar las funciones de pertenencia y los centroides, sólo se tienen en cuenta atributos completos. Finalmente se imputan los valores perdidos de cada objeto incompleto x_i basándose en la información de los grados de pertenencia y los valores del centroide del clúster[65].

9. *Support Vector Machines Imputation (SVMI):* intercambia los atributos de decisión (clases) y los atributos condicionales (atributos de entrada) y usa la regresión de las *SVM* para predecir los valores de los atributos de decisión[66].

10. *Event Covering (EC): s*e discretizan los valores continuos usando un criterio de pérdida mínima de información y se aproxima un modelo de probabilidad mixto con uno discreto. Con el método de inferencia desarrollado se estiman los valores perdidos que existan en los datos[67].

11. *Regularized Expectation-Maximization (EM):* este algoritmo maximiza la probabilidad logarítmica de los datos incompletos explotando la relación entre los datos completos y los datos incompletos[68].

12. *Singular Value Decomposition Imputation (SVDI)*: este método emplea *singular value decomposition* para obtener un conjunto de patrones que puedan ser combinados linealmente para aproximar los valores de todos los atributos en el conjunto de datos[64].

13. *Bayesian Principal Component Analysis (BPCA)*: este método estima los valores perdidos basándose en el análisis bayesiando de componentes principales. Consta de tres procesos elementales: regresión de componentes principales, estimación bayesiana y *EM* como algoritmo repetitivo[69].

14. *Local Least Squares Imputation (LLSI)*: en este algoritmo la instancia con valores perdidos es representada como una combinación lineal de instancias similares. El

primer paso es seleccionar *k* instancias por la norma L2 y el segundo es de regresión y estimación[70].

Los algoritmos *DNI* e *IM* se descartaron porque en el conjunto de datos existía una cantidad de valores perdidos considerable (2316 en total), por tanto era imposible ignorarlos o eliminar todos los casos que tuvieran al menos un valor perdido.

CMC y *SVMI* son algoritmos de aprendizaje supervisado, sus resultados tienen en cuenta la clase a la que pertenece la instancia con valores perdidos y, previendo que probablemente alguno de los pacientes que se quiera diagnosticar no tenga el valor de algún análisis complementario, es necesario pensar en una estrategia que no tenga en cuenta la clase.

EC no se puede aplicar al conjunto de datos porque no admite valores numéricos enteros o continuos.

Finalmente se aplicaron a los datos los algoritmos restantes, *EM*, *SVDI*, *BPCA* y *LLSI* generan valores muy cercanos a los límites de cada atributo o valores prácticamente imposibles en la vida real. Por ejemplo *EM y SVDI* a un paciente que mide 147cm le imputan el peso con 178kg.

Por las características del problema que nos ocupa, decidimos decantarnos por los algoritmos que imputan valores perdidos usando el método *KNN* (*KNNI*, *WKNNI* y *KMI)* pues estos garantizan imputar con un valor que esté dentro de los valores normales del conjunto de datos.

A los conjuntos de datos resultantes de la aplicación de estos tres algoritmos se les calcularon las medidas de complejidad y en la Tabla 5 se muestra la diferencia entre el valor de cada medida antes de aplicar los algoritmos y después de aplicados. La diferencia se calcula de la siguiente forma:

$$Diferencia = Valor\ antes\ del\ algoritmo - Valor\ después\ del\ algoritmo$$

Tabla 5. Diferencia entre el valor de las medidas antes de aplicar cada algoritmo de imputación de valores perdidos y después de aplicado. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / Medidas	***KNNI***	***WNNNI***	***KMI***
F1	0,000	0,000	0,000
F2	0,218	0,218	0,218
F3	-0,012	-0,012	-0,012
N1	-0,006	-0,015	**0,004**
N2	0,011	0,003	**0,032**
N4	0,002	**0,018**	-0,003

Como se puede apreciar en la tabla anterior, las medidas de superposición en los valores de los atributos de clases diferentes, para los tres algoritmos tienen la misma complejidad, por tanto no las podemos usar como patrón de comparación. Sin embargo en las medidas de separabilidad de clases y de geometría se pueden apreciar cambios. Por las características de N1, entre más pequeño sea su valor, menor va a ser la complejidad del conjunto de datos, por lo que *KMI* es el que más disminuye la complejidad según N1, de hecho es el único que tiene una diferencia positiva. Con N2 sucede igual, valores pequeños muestran menor complejidad y *KMI* es el que mejor funciona en este aspecto. En el caso de N4 la filosofía es la misma y *WKNNI* es el que mejor se comporta, con una diferencia relativamente alta con respecto a los demás algoritmos.

Luego de analizar los resultados se decidió elegir el conjunto de datos resultantes de aplicar *KMI*, porque de manera general fue el que más medidas demostraron que disminuyó la complejidad de los datos.

En el gráfico 3 se muestra una comparación de los valores de las medidas de complejidad del conjunto antes y después de aplicado *KMI.* Se puede apreciar que la técnica aplicada mejora de manera general la complejidad de los datos. F1, por su definición, mientras más grande sea su valor, menor complejidad tendrán los datos, en este caso se mantiene constante. Entre más pequeño sea el valor de F2 menor es la complejidad y como se observa en el gráfico, el valor de esta métrica disminuye considerablemente. Los valores de F3 deben aumentar para que la complejidad del conjunto disminuya, en este caso se cumplió, el valor de esta medida aumentó en 0,012. N1 disminuyó en 0,004; N2 en 0,032 y N4 aumentó en 0,003.

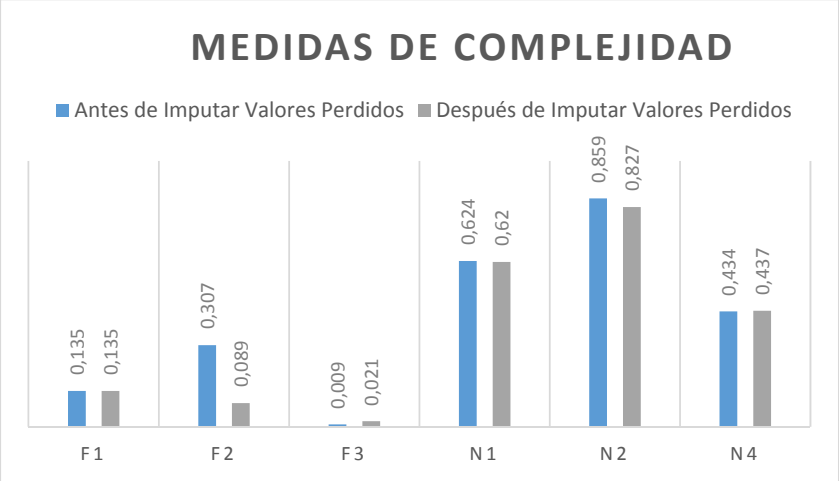

Grafico 3. Medidas de complejidad antes y después de imputar los valores perdidos mediante el algoritmo *KMI*. Las medidas sugieren una menor complejidad cuando los valores de F1 y F3 son mayores y los valores de F2, N1, N2 y N4 son menores.

2.5. Selección de atributos

Para el trabajo se utilizó la herramienta WEKA y se aplicaron algoritmos representativos de cada una de las categorías expuestas anteriormente.

Filter:

- *CfSubsetEval*: Evalúa un subconjunto de atributos considerando la habilidad predictiva individual de cada variable, así como el grado de redundancia entre ellas. Se prefieren los subconjuntos de atributos que estén altamente correlacionados con la clase y tengan baja intercorrelación[71].

- *ConsistencySubsetEval*: Evalúa un subconjunto de atributos por el nivel de consistencia en los valores de la clase al proyectar las instancias de entrenamiento sobre el subconjunto de atributos[72].

***Wrapper*:**

- *ClassifierSubsetEval:* Evalúa los subconjuntos de atributos en los datos de entrenamiento o en un conjunto de prueba independiente, utilizando un clasificador. Utilizamos el método J48 (árbol de decisión C4.5)
- *WrapperSubsetEval:* Evalúa los subconjuntos de atributos utilizando un clasificador (también el J48). Emplea validación cruzada para estimar la exactitud del esquema de aprendizaje en cada conjunto[27].

Evaluadores de Atributos Individuales:

- *ChiSquaredAttributeEval:* Calcula el valor estadístico Chi-cuadrado de cada atributo con respecto a la clase y así obtiene el nivel de correlación entre la clase y cada atributo.
- *GainRatioAttributeEval:* Evalúa cada atributo midiendo su razón de beneficio con respecto a la clase.
- *InfoGainAttributeEval:* Evalúa los atributos midiendo la ganancia de información de cada uno con respecto a la clase. Anteriormente discretiza los atributos numéricos[73].
- *ReliefFAttributeEval:* Evalúa el valor de un atributo mediante el muestreo repetido de una instancia y teniendo en cuenta el valor del atributo dado para la instancia más cercana dentro de la misma clase y fuera de la clase. Puede trabajar con datos discretos y continuos.

Estos algoritmos le fueron aplicados al conjunto de datos (con valores perdidos ya sustituidos usando *K-Means*), en los casos de algoritmos *Filter* y *Wrapper* se aplicaron todas las estrategias de búsqueda.

Los resultados les fueron mostrados a los especialistas para que dieran su opinión. Puede que algún algoritmo eliminara un atributo importante desde el punto de vista médico.

Según sus recomendaciones se descartaron los resultados de los algoritmos:

CfSubsetEval y *WrapperSubsetEval*, porque eliminaban demasiados atributos que son claves desde el punto de vista médico, eliminaban como mínimo 12 y 11 atributos respectivamente.

ClassifierSubsetEval, aunque no elimina tantos atributos como el anterior, los que elimina no se corresponden con el criterio médico.

ChiSquaredAttributeEval, GainRatioAttributeEval, InfoGainAttributeEval, aunque varían el orden de los atributos, eliminan los mismos (8 atributos). Entre ellos atributos como la hemoglobina, la presencia o no de una angina estable y el grado de esta, la glicemia, entre otros, que los especialistas opinan que son importantes a la hora de tomar una decisión de este tipo

Finalmente se eligieron las selecciones de atributos realizadas por los algoritmos *ConsistencySubsetEval*, con estrategia de búsqueda *RandomSearch* (14 atributos) y la selección de *ReliefFAttributeEval+Ranker*, con dos variantes, una eliminando sólo el atributo con valor de "*rankeo*" negativo (17 atributos) y otra eliminando la misma cantidad de atributos que *ConsistencySubsetEval*. Todo esto se realizó siguiendo el criterio de los especialistas.

Los atributos de cada conjunto de datos serían (Figura 2):

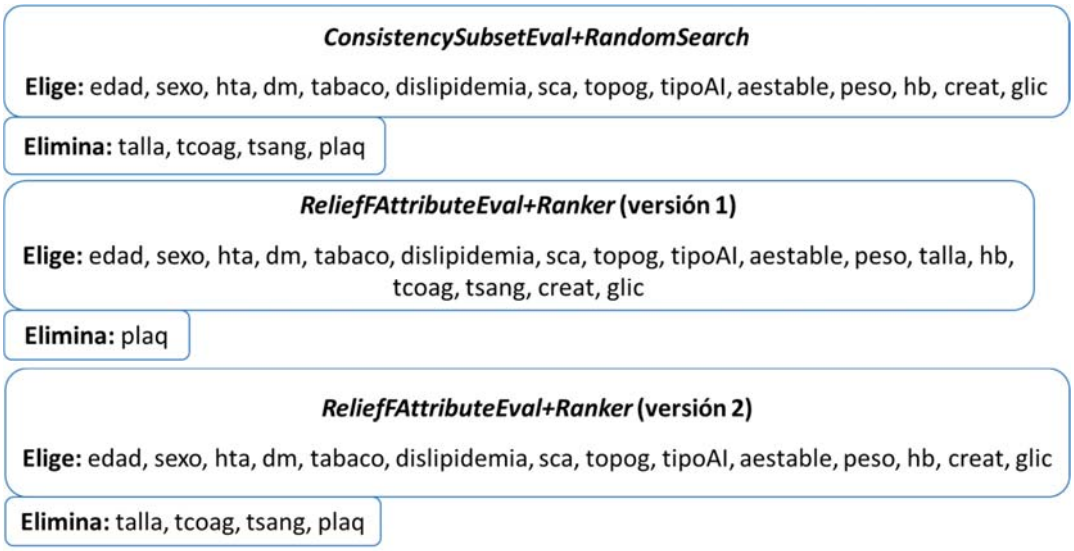

Figura 2. Atributos elegidos y eliminados por cada algoritmo de selección.

A los conjuntos de datos obtenidos se les aplicaron las medidas de complejidad, las diferencias entre estas medidas antes y después de aplicar los algoritmos se muestran en la Tabla 6.

Tabla 6. Diferencia entre el valor de las medidas antes de aplicar cada algoritmo de selección de atributos y después de aplicado. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / **Medidas**	***ConsistencySubsetEval***	***ReliefF* v1**	***ReliefF* v2**
F1	0,000	0,000	0,000
F2	-0,124	-0,008	-0,135
F3	0,000	0,000	0,000
N1	0,156	0,161	**0,169**
N2	**0,166**	0,129	0,158
N4	0,098	**0,113**	0,110

En la tabla se aprecia que el valor de F1 y F3 sigue invariable y continua siendo igual para todos los casos. F2 varía para cada conjunto y el menor valor lo tiene *ReliefF* v1, aunque ninguno disminuye la complejidad según esta medida.

En cuanto a las medidas de separabilidad de clases el valor óptimo de N1 lo tiene *ReliefF* v2, mientras que para N2 el mejor valor lo tiene *ConsistencySubsetEval* con una disminución importante con respecto a los demás.

En el caso de N4 el de mejor valor es el de *ReliefF* v1.

Se puede notar que ningún conjunto de datos es totalmente óptimo en cuanto a las medidas de complejidad analizadas, por lo que se decidió continuar con los tres para el siguiente paso, Selección de instancias, y luego evaluar cuál obtenía los mejores resultados.

Es importante antes de terminar con esta técnica, comparar sus resultados con los obtenidos hasta el paso anterior, pues la comparación realizada anteriormente es sólo entre los 3 conjuntos de datos resultantes de la selección de atributos.

En los gráficos 4, 5 y 6 se muestra una comparación de cada conjunto de datos con respecto al conjunto que les dio origen.

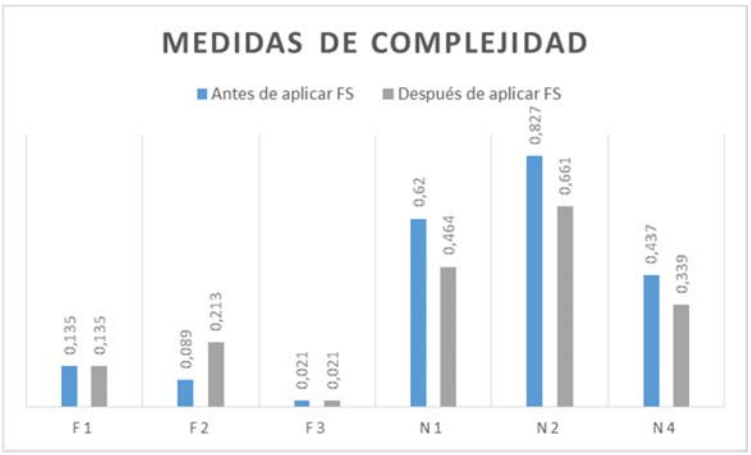

Grafico 4. Medidas de complejidad antes y después de seleccionar atributos mediante el algoritmo *ConsistencySubsetEval*. Las medidas sugieren una menor complejidad cuando los valores de F1 y F3 son mayores y los valores de F2, N1, N2 y N4 son menores.

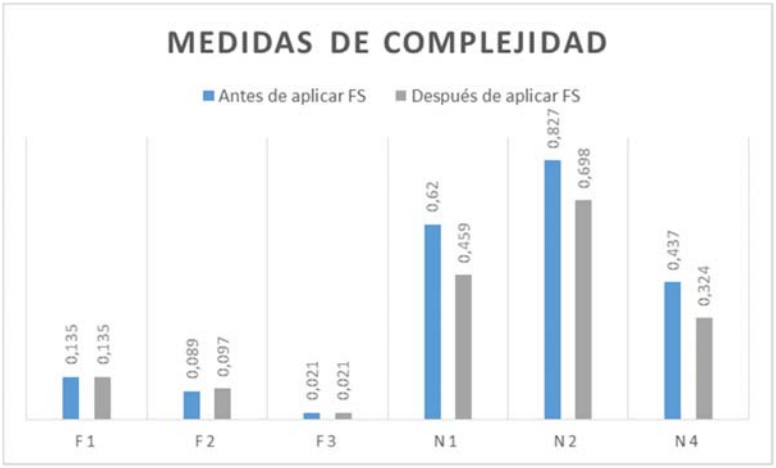

Grafico 5. Medidas de complejidad antes y después de seleccionar atributos mediante el algoritmo *ReliefF* v1. Las medidas sugieren una menor complejidad cuando los valores de F1 y F3 son mayores y los valores de F2, N1, N2 y N4 son menores.

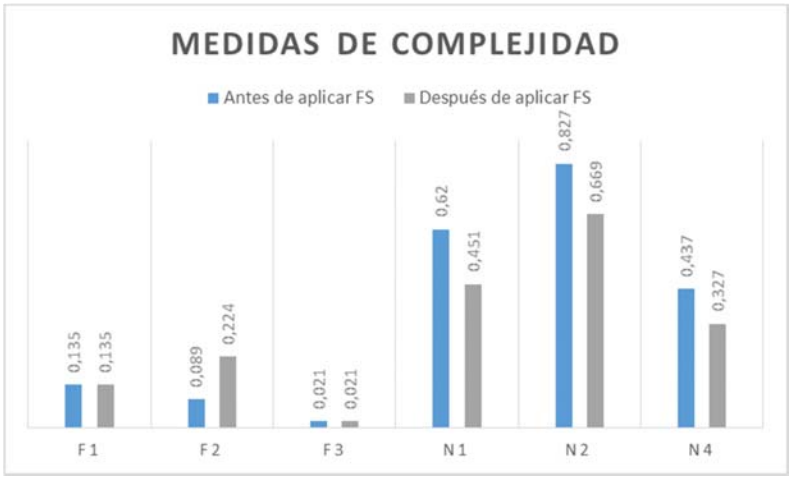

Grafico 6. Medidas de complejidad antes y después de seleccionar atributos mediante el algoritmo *ReliefF* v2. Las medidas sugieren una menor complejidad cuando los valores de F1 y F3 son mayores y los valores de F2, N1, N2 y N4 son menores.

Se observa en todos los casos que el valor de F2 aumenta, lo que denota que la superposición de límites por clases es mayor, sin embargo las medidas de separabilidad de clases y de geometría disminuyen en todos los casos.

2.6. Selección de instancias (prototipos)

En la investigación se usó la selección de prototipos y para esta tarea se usó la herramienta KEEL[34] que contiene varios algoritmos con este fin.

El criterio de selección que se siguió fue el de la proporción entre las clases, es decir, que luego de la selección la razón entre la clase positiva y la negativa no pasara del doble, para que las clases no estuvieran muy desbalanceadas. Se utilizó la siguiente fórmula:

$$\frac{Número\ de\ pacientes\ con\ clase\ positiva}{Número\ de\ pacientes\ con\ clase\ negativa} < 2$$

La selección de prototipos se realizó en la herramienta KEEL, para ello se seleccionaron 8 algoritmos con diferentes propiedades, pues según la literatura[19] son los que mejor rendimiento tienen.

Edición Por Bloques:

- *AllKNN:* All KNN es una extensión de *ENN.* El algoritmo, para *i*=0 hasta *k* marca como “malas” cualquier instancia incorrectamente clasificada por su *i*-ésimo vecino más cercano. Cuando el ciclo se completa *k* veces, elimina las instancias marcadas como “malas”[74].
- *ModelCS (Model Class Selection)*: Reduce el tamaño del conjunto de entrenamiento llevando la cuenta de cuántas veces cada instancia fue uno de los k vecinos más cercanos de otra, y cuándo su clase coincidió con la de la instancia que estaba siendo clasificada. Si el número de veces que estuvo incorrecto es más grande que el número de veces que fue correcto, entonces se elimina[75].

Edición Decremental:

- *ENN (Edited Nearest Neighbor):* Este algoritmo comienza con *S*=conjunto de entrenamiento y luego cada instancia de *S* es eliminada si no concuerda con la mayoría de sus *k* vecinos más cercanos[76].
- *ENNTh (Edited Nearest Neighbor Estimating):* Para cada clase en *T*, ENNTh determina la probabilidad de que *x* pertenezca a la misma. Para obtener un estimado de este valor, se usan los *k* vecinos más cercanos x_i de *x.* Entre más pequeña sea la distancia $d(x,x_i)$, mayor será la contribución de x_i a la estimación de la probabilidad de que *x* pertenezca a la clase $l(x_i)$. La probabilidad de que un elemento *x* pertenezca a su propia clase tiene que ser suficientemente alta, si no *x* es descartado, incluso cuando la predicción sea correcta[77].
- *MENN (Modified Edited Nearest Neighbor):* Similar a *ENN,* comienza con *S*=conjunto de entrenamiento y luego cada instancia X_i en S se elimina si no coincide con todos sus *k* + *l* vecinos más cercanos, donde *l* son todas las instancias en S que están a la misma distancia que el último vecino de X_i[78].
- *NCNEdit (Nearest Centroid Neighbor Edition):* Aplica la regla de clasificación *NCN* para realizar un proceso de edición sobre el conjunto de entrenamiento. Definido el esquema *NCN,* el proceso de edición consiste en asignarle a *S*=conjunto de entrenamiento y eliminar de *S* cada prototipo mal clasificado por la regla *NCN*.
- *RNG (Relative Neighborhood Graph Editing):* Este método utiliza un gráfico de proximidad *G=(T, E)*, cuando hace las predicciones de las clases. G es un grafo

indirecto, con un conjunto de nodos *T* y un conjunto de arcos *E*. Geométricamente, x e y son vecinos en *G* si y solo si la intersección de las dos hiperesferas con centros x e y y radios *d(x,y)* no contiene ningún otro de los elementos de *T*. La clase de un elemento se predice como a la que la mayoría de sus nodos adyacentes en *G* pertenecen. La instancia es eliminada cuando la clase actual y la predicha no coinciden[77].

Condensación Por Bloques

- *POP (Patterns by Ordered Projections):* Elimina los ejemplos que no están dentro de los límites de la región a la que pertenecen. Para ello, cada atributo se estudia por separado, con un valor llamado *weakness*, asociado a cada una de las instancias, si no está dentro del límite. Las instancias con un valor de *weakness* igual al número de atributos son eliminadas[79].

Estos algoritmos se aplicaron a los 3 conjuntos de datos obtenidos del paso anterior, resultando en 24 conjuntos de datos (3 conjuntos de datos x 8 algoritmos).

A cada conjunto de datos se le calculó la razón entre la clase positiva y la negativa. En los que la razón fue mayor o igual que dos, se descartaron.

Quedando las siguientes alternativas (Tablas 7, 8 y 9):

Tabla 7. Selección de algoritmos para el primer grupo.

ConsistencySubsetEval		
Algoritmo	**Número de instancias**	**Razón**
ModelCS	643	1,74
NCNEdit	551	1,92
POP	761	1,50

Tabla 8. Selección de algoritmos para el segundo grupo.

ReliefF v1		
Algoritmo	**Número de instancias**	**Razón**
ModelCS	654	1,70
POP	762	1,50

Tabla 9. Selección de algoritmos para el tercer grupo.

ReliefF v2		
Algoritmo	**Número de instancias**	**Razón**
ModelCS	654	1,75
NCNEdit	547	1,99
POP	754	1,47

Seguidamente se aplicaron las medidas de complejidad a cada uno de los conjuntos anteriores y se compararon para elegir el mejor representante de cada grupo (en lo adelante Grupo 1: resultados provenientes de *ConsistencySubsetEval*, Grupo 2: de *ReliefF* v1, Grupo 3: de *ReliefF* v2).

En la Tabla 10 se muestra la comparación para el primer grupo. Se puede ver que para todas las medidas de complejidad analizadas *NCNEdit* es el que da mejores resultados y con diferencias importantes con respecto a los otros algoritmos.

Tabla 10. Diferencia entre el valor de las medidas antes de aplicar cada algoritmo de selección de prototipos al primer grupo y después de aplicado. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / Medidas	*ModelCS*	*NCNEdit*	*POP*
F1	-0,041	**-0,222**	0,001
F2	-0,072	**0,067**	0,000
F3	-0,004	**-0,026**	0,000
N1	0,156	**0,293**	0,003
N2	0,155	**0,333**	0,005
N4	0,066	**0,113**	0,014

La Tabla 11 muestra la comparación del segundo grupo. Observamos que *ModelCS* fue el algoritmo que mejores valores de complejidad obtuvo en todos los casos

Tabla 11. Diferencia entre el valor de las medidas antes de aplicar cada algoritmo de selección de prototipos al segundo grupo y después de aplicado. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / Medidas	***ModelCS***	***POP***
F1	**-0,067**	0,001
F2	**0,007**	0,000
F3	**-0,003**	0,000
N1	**0,144**	0,002
N2	**0,141**	0,005
N4	**0,046**	0,002

La Tabla 12 muestra los resultados para el tercer grupo, aquí sucede igual que en el primer grupo, el método *NCNEdit* es el que obtiene los mejores valores de complejidad con diferencias significativas con respecto a sus dos contrincantes.

Tabla 12. Diferencia entre el valor de las medidas antes de aplicar cada algoritmo de selección de prototipos al tercer grupo y después de aplicado. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / Medidas	*ModelCS*	*NCNEdit*	*POP*
F1	-0,073	**-0,224**	0,000
F2	0,000	**0,073**	0,000
F3	-0,003	**-0,030**	0,000
N1	0,160	**0,275**	0,003
N2	0,157	**0,331**	0,004
N4	0,061	**0,121**	-0,002

Concluida la comparación dentro de cada grupo, pasamos a comparar a los ganadores por cada grupo, el resultado se muestra en la Tabla 13.

Tabla 13. Comparación de los conjuntos de datos seleccionados por cada grupo. Se prefiere el algoritmo con mayor diferencia positiva en las medidas F2, N1, N2 y N4 y con mayor diferencia negativa en F1 y F3.

Algoritmos / Medidas	Grupo 1 *NCNEdit*	Grupo 2 *ModelCS*	Grupo 3 *NCNEdit*
F1	-0,222	-0,067	**-0,224**
F2	0,067	0,007	**0,073**
F3	-0,026	-0,003	**-0,030**
N1	**0,293**	0,144	0,275
N2	**0,333**	0,141	0,331
N4	**0,113**	0,046	0,101

Se puede observar que en cuanto a las medidas de superposición en los valores de los atributos de clases diferentes, los mejores valores los obtiene el candidato del Grupo 3.

Pasando a las medidas de separabilidad, el mejor valor de N1 lo tiene el candidato del Grupo 1, y para N2 también, este último con diferencias importantes en comparación con sus contrincantes.

En cuanto a N4, también el mejor valor lo tiene el candidato del Grupo 1.

Según Bernadó-Mansilla y Macià-Antolínez[32], las medidas que calculan la superposición en los valores de los atributos de clases diferentes sugieren, pero que por sí mismas no son decisivas, para estimar la complejidad de la clasificación ya que no están tan fuertemente correlacionadas con la complejidad . Se deben revisar el resto de las métricas para para completar la decisión. Además aseguran que las métricas enfocadas en la distribución de las clases deben proveer más información acerca de la complejidad.

Por lo tanto el elegido fue el del Grupo 1, pues a pesar de que no obtiene los mejores valores de complejidad para las medidas F1, F2 y F3, tampoco es el peor y sí es el ganador absoluto en las métricas de separabilidad y geometría de las clases.

En el Gráfico 7 se muestran las medidas de complejidad del conjunto elegido comparados con el que le dio origen en el paso anterior.

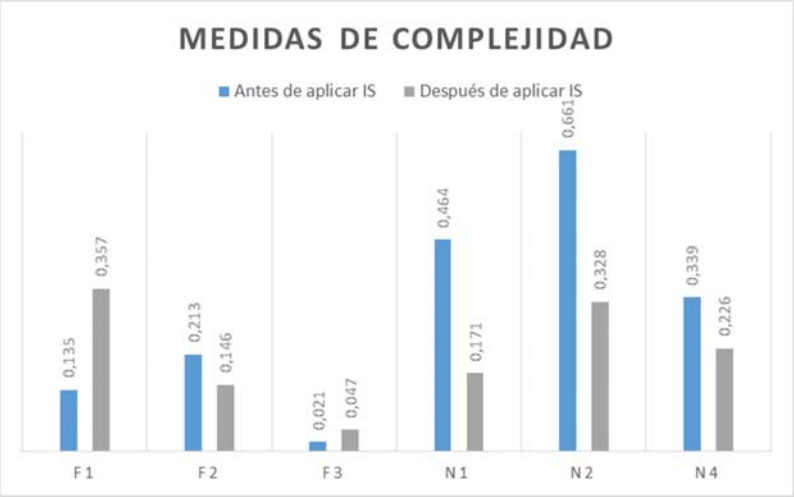

Grafico 7. Comparación de las medidas de complejidad antes y después la selección de instancias con *NCNEdit*. Las medidas sugieren una menor complejidad cuando los valores de F1 y F3 son mayores y los valores de F2, N1, N2 y N4 son menores.

2.7. Descripción del conjunto de datos final

Terminada la etapa de preprocesamiento (Limpieza de los Datos, Normalización, Imputación de Valores Perdidos, Selección de Atributos y Selección de Instancias) el conjunto obtenido tiene las siguientes características: 15 atributos (incluida la clase), se eliminaron:

- Talla
- Tiempo de coagulación
- Tiempo de sangrado
- Plaquetas

Con 551 instancias cuya distribución por clases se muestra en el Gráfico 8.

Gráfico 8. Proporción de las clases en el conjunto de entrenamiento.

Esta ligera desproporción entre las clases es beneficiosa pues una de las alternativas del aprendizaje sensible a costos es tener mayor cantidad de instancias de la clase más costosa, que es la positiva, pues así el algoritmo aprende más y cometer menos errores a la hora de predecirla. En la literatura[13] se propone una desproporción mucho más amplia, pero en nuestro caso, aunque los costos de no hacerle la prueba a un paciente enfermo son mucho más altos o más graves que hacérsela a uno que no la requiere, nos interesa que el algoritmo a la vez que clasifique correctamente la mayor cantidad posible de positivos, también tenga un rendimiento adecuado con la clase negativa, porque en estos casos se somete al paciente a un riesgo innecesario y además se consumen insumos y dispositivos médicos de alto valor.

A continuación se muestra cómo se comportaron las medidas de complejidad a través de cada paso del preprocesamiento (Gráfico 9). Obsérvese cómo fueron disminuyendo los valores de N1, N2, N4 y F2 así como aumentado los valores de F1 y F2 en cada etapa del preprocesamiento. Esto evidencia que la complejidad de los datos finales es menor que la de los datos iniciales.

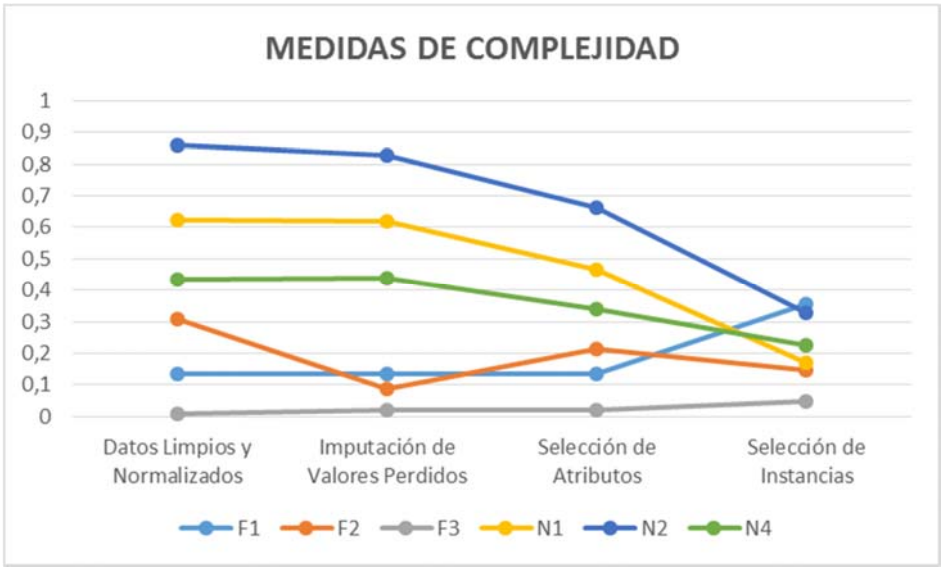

Gráfico 9. Evolución de las medidas de complejidad con las tareas de preprocesamiento.

2.8. Conclusiones parciales

- Para la imputación de valores perdidos se compararon los algoritmos *KNNI, WKNNI* y *KMI*, de ellos el que mejor comportamiento mostró fue *KMI*, realizando una mayor disminución de la complejidad del conjunto de datos.
- Para la selección de atributos se compararon los resultados de dos algoritmos: *ConsistencySubsetEval, ReliefFAttributeEval*, este último con dos variantes. Aunque los tres resultados disminuyeron la complejidad del conjunto de datos, ninguno fue un ganador global.
- En la selección de instancias se compararon los resultados de tres algoritmos: *NCNEdit, ModelCS* y *POP.* De ellos el que mejor se comportó fue *NCNEdit,* siendo el que más reducía la complejidad del conjunto de datos.

CAPÍTULO 3 MODELO DE CLASIFICACIÓN

En este capítulo se muestra el proceso de selección del modelo, para ello se prueban algoritmos de tres categorías diferentes: clasificadores simples, metaclasificadores y basados en el costo del error, todos ellos disponibles en WEKA. Además se pone en práctica un experimento basado en la creación de instancias artificiales para lograr un modelo que sin perder precisión, sea más interpretable. Se le aplican costos a los errores para tener una evaluación más real de la exactitud del modelo. Al resultado de cada uno de los modelos se le aplica medidas de exactitud y en base a eso se elige el mejor candidato de cada una de las categorías. Luego, de entre estos, se selecciona el modelo final. En todos los casos se utiliza la técnica de evaluación validación cruzada de 10 particiones. Para finalizar se utiliza un grupo de casos nuevos para comparar el comportamiento del sistema con el de un grupo de especialistas en la materia.

3.1. Selección del modelo simple

Para la selección del modelo simple se eligieron algoritmos representativos de cada una de las categorías enumeradas en el Capítulo 1 (Redes Neuronales Artificiales, Aprendizaje Bayesiano, Aprendizaje Basado en Instancias, Máquinas de Soporte Vectorial, Aprendizaje Basado en Reglas y Árboles de Decisión) y estos se le aplicaron al conjunto de entrenamiento. Algunos de estos algoritmos brindan la posibilidad de variarle parámetros para mejorar el ajuste del modelo a los datos. En las Tablas (14-18) se muestran las variaciones que se hicieron con los parámetros de algunos algoritmos, estos están agrupados por la categoría a la que pertenecen. Los parámetros que no se muestran en las tablas permanecieron con los valores que trae WEKA por defecto. La configuración elegida en cada caso es resaltada. La selección del mejor modelo estuvo sujeta a los resultados de las medidas de exactitud y como medida de referencia para la selección se tomó a la sensibilidad. En el caso de que los valores de sensibilidad de algunas pruebas fueran muy similares o iguales, se eligió el caso que mejor especificidad tuviera.

Tabla 14. Prueba de parámetros para Redes Neuronales Artificiales. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

MLP	**Sensibilidad (%)**	**Especificidad (%)**
hiddenLayers=a	93,92	85,19
hiddenLayers=i	93,65	87,83
hiddenLayers=o	92,82	74,07
hiddenLayers=t	94,20	87,30
hiddenLayers=t & -L=0,1	92,27	85,19
hiddenLayers=t & -L=0,5	94,20	84,66
hiddenLayers=t & -L=1	93,92	84,13
hiddenLayers=t & -L=0,3 & -M=0,5	94,20	84,13
hiddenLayers=t & -L=0,3 & -M=0,6	95,03	83,07
hiddenLayers=t & -L=0,3 & -M=0,7	92,82	84,13
hiddenLayers=t & -L=0,3 & -M=0,8 & triningTime= 1000	95,03	83,60
hiddenLayers=t & -L=0,3 & -M=0,8 & triningTime= 1500	**95,03**	**84,66**

Tabla 15. Prueba de parámetros para Máquinas de Soporte Vectorial. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

SMO	**Sensibilidad (%)**	**Especificidad (%)**
(por defecto WEKA)[3]	92,54	72,49
C=10 & Polykernel	90,06	74,07
C=1 & Puk	96,13	81,48
C=1 & Puk omega=0,2	97,79	77,25
C=1 & Puk omega=0,3	97,51	80,42
C=1 & Puk omega=0,5	96,69	80,95
C=1 & Puk omega=0,2 sigma=0,5	98,62	79,89
C=1 & Puk omega=0,2 sigma=0,6	98,62	80,95
C=10 & Puk omega=0,2 sigma=0,6	**98,34**	**89,95**
C=1 & RBFKernel g=0,01	94,75	57,67
C=1 & RBFKernel g=0,03	96,69	60,85
C=1 & RBFKernel g=0,05	96,13	60,85
C=1 & RBFKernel g=0,07	96,13	62,96
SPegasos		
(por defecto WEKA)	90,61	72,49
lambda=0,005	94,48	67,72
lambda=0,008	95,03	63,49
lambda=0,01	**96,13**	**61,38**

[3] Quiere decir que se utilizan los valores de parámetros que trae WEKA cada algoritmo por defecto.

Tabla 16. Prueba de parámetros de Algoritmos Basados en Instancias. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

IBk	Sensibilidad (%)	Especificidad (%)
IBk (K=1)	**96,98**	**95,19**
IBk (K=3)	92,54	76,19
IBk (K=5)	93,37	72,49
KStar		
(por defecto WEKA)	82,60	74,60
globalBlend=10	81,22	71,43
globalBlend=50	92,27	77,25
globalBlend=70	**96,96**	**68,25**

Tabla 17. Prueba de parámetros para Algoritmos Basados en Reglas. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

ConjunctiveRule	Sensibilidad (%)	Especificidad (%)
(por defecto WEKA)	94,75	54,50
numAntds=2	**97,51**	**53,97**
JRip		
(por defecto WEKA)	88,40	72,49
folds=10	**91,16**	**76,72**
NNge		
(por defecto WEKA)	**92,82**	**81,48**

numFoldersMIOption=3	91,71	80,42
numFoldersMIOption=8	92,82	79,37
numFoldersMIOption=10	91,99	80,42
PART		
(por defecto WEKA)	**90,61**	**82,01**
confidenceFactor=0,5	90,88	80,95
minNumObj=10	91,44	69,84
Ridor		
(por defecto WEKA)	91,16	66,67
folds=2	94,20	61,38
folds=2 & majorityClass=True	**95,03**	**60,85**

Tabla 18. Prueba de parámetros para Árboles de Decisión. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

ADTree	**Sensibilidad (%)**	**Especificidad (%)**
(por defecto WEKA)	87,85	76,72
numOfBoostingIterations=5	91,16	73,54
numOfBoostingIterations=5 & searchPath=2	**95,30**	**59,79**
BFTree		
(por defecto WEKA)	92,27	75,66
minNumObj=3	93,09	74,07
minNumObj=5	92,82	73,02

minNumObj=3 & numFoldsPruning=10	92,27	75,66
minNumObj=3 & numFoldsPruning=3 & UNPRUNED	91,99	75,66
minNumObj=3 & numFoldsPruning=3 & PREPRUNED	**96,13**	**59,26**
FT		
(por defecto WEKA)	90,88	78,31
modelType=1	90,88	84,13
modelType=2	91,71	77,25
modelType=2 & numBoostingIterations=10	**93,37**	**73,54**
modelType=2 & numBoostingIterations=30	92,54	72,49
J48		
J48 (por defecto WEKA)	92,54	78,31
confidenceFactor=0,10	93,09	77,25
confidenceFactor=0,15	92,82	76,72
confidenceFactor=0,20	**93,65**	**76,19**
RandomForest		
(por defecto WEKA)	95,03	80,42
numFeatures=5	**95,86**	**78,31**
REPTree		
(por defecto WEKA)	92,27	67,72
numFolds=10	**93,09**	**68,25**
SimpleCart		

(por defecto WEKA)	92,27	74,07
numFoldsPruning=10	**92,54**	**75,13**

Una vez elegida la mejor configuración para cada algoritmo, se compararon todos los métodos (Tabla 19).

Tabla 19. Valores de las medidas de exactitud para los algoritmos simples. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

Algoritmo	Sensibilidad (%)	Especificidad (%)
Naïve Bayes	91,44	69,84
Naïve Bayes Simple	91,71	69,31
Multi Layer Perceptron (MLP)	95,03	84,66
SMO	**98,34**	89,95
SPegasos	96,13	61,38
IBk	**96,98**	**95,19**
LWL	74,31	75,13
KStar	96,96	68,25
Conjunctive Rule	**97,51**	53,97
JRip	91,16	76,72
NNge	92,82	81,48
PART	90,61	82,01
Ridor	95,03	60,85
ADTree	95,30	59,79

BFTree	96,13	59,26
FT	93,37	73,54
J48	93,65	76,19
LADTree	93,37	80,42
LMT	92,82	86,77
RandomForest	95,86	78,31
REPTree	93,09	68,25
SimpleCart	92,54	75,13

Analizando los resultados mostrados en la tabla se puede apreciar que los algoritmos *SMO, IBk* y *ConjunctiveRule* son los que tienen mayor sensibilidad. Si bien *SMO* es el que tiene mayor valor de sensibilidad (98,34 %), si se revisan los valores de especificidad, de todos los algoritmos analizados el único que logra una sensibilidad por encima del 90 % es *IBk* con 95,19; por tanto de forma global es el mejor algoritmo.

3.2. Selección usando metaclasificadores

Después de analizar los resultados obtenidos con los clasificadores simples, se probó con los metaclasificadores. Este tipo de clasificadores son una serie de métodos que permiten crear una colmena de algoritmos que trabajan unidos para realizar la clasificación final. La razón de su utilización es que los diferentes algoritmos existentes, por sí solos, se adecuan mejor a unas situaciones concretas que a otras. Sin embargo, al utilizarlos de forma conjunta se consigue una mayor robustez en las etapas de clasificación.

Los diversos métodos de metaclasificación permiten, por una parte, la generación de lo que se conoce como “ensambles”, la unión de múltiples clasificadores débiles del mismo tipo para la creación de uno más preciso; o, por otra parte, la combinación de diferentes métodos que dotan al metaclasificador de las características híbridas que todos los métodos simples utilizan.

Existen 3 estrategias representativas de este esquema:

- *Bagging (Bootstrap Aggregating)* genera clasificadores de varias muestras de los ejemplos, funciona especialmente para algoritmos de aprendizaje inestables (cambian mucho sus estructuras al cambiar un poco los ejemplos), por ejemplo, los árboles de decisión;

- *Boosting* y su variante más usada *AdaBoost (Adapting Boosting)* genera igual un conjunto de clasificadores, sin embargo, *Adaboost* los genera secuencialmente (*Bagging* los puede generar en paralelo);

- *Stacking* que construye un conjunto de modelos usando diferentes algoritmos de aprendizaje.

Se seleccionaron 3 algoritmos de WEKA representativos de las estrategias anteriores: *AdaBoostM1, Bagging*, *Stacking* y se realizaron pruebas con ellos, que se muestran en la Tabla 20. Los dos primeros se probaron usando como clasificador base el *J48* variándoles el *confidenceFactor,* este factor es el que regula la poda del algoritmo, menores valores de este parámetro implican más poda. Según la literatura[13] se obtienen mejores resultados de *AdaBoostM1* y *Bagging* cuando se usa con un algoritmo de base inestable, *J48* entre menor sea el *confidenceFactor* más inestable se vuelve, por eso se utilizó para este experimento. El último, *Stacking,* se probó con un grupo de 5 clasificadores pertenecientes a diferentes metodologías (*J48*, *SMO, IBk, LWL, ConjunctiveRule*).

Se seleccionó la mejor configuración para cada algoritmo (resaltada en negrita) y en la Tabla 21 se muestra la comparación entre los 3 métodos.

Tabla 20. Prueba de parámetros para metaclasificadores. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

AdaBoostM1	**Sensibilidad (%)**	**Especificidad (%)**
***J48* C=0,0025**	93,09	84,66
***J48* C=0,0020**	93,65	81,48
***J48* C=0,0015**	92,54	81,48
***J48* C=0,0005**	93,09	79,89
***J48* C=0,00045**	93,37	78,84
***J48* C=0,00040**	94,75	79,37
***J48* C=0,000035**	95,58	80,42
***J48* C=0,000035 & numIterations=80**	**95,58**	**87,30**
Bagging		
***J48* C=0,000050**	95,30	69,84
***J48* C=0,000045**	95,03	67,72
***J48* C=0,0000050**	95,86	60,85
***J48* C=0,0000045**	95,86	61,90
***J48* C=0,0000040**	95,58	60,85
***J48* C=0,0000025**	95,86	61,38
***J48* C=0,0000045 & binarySplit= True**	97,24	54,50
***J48* C=0,0000045 & binarySplit= True & numIterations=20**	97,24	55,03
***J48* C=0,0000045 & binarySplit= True & numIterations=25**	**97,51**	**53,44**

Stacking		
Con J48 + SMO	98,34	88,36
Con J48 + SMO + IBk	97,79	90,48
Con J48 + SMO + IBk + LWL	97,79	91,53
Con J48 (binarySplit=T) + SMO + IBk + LWL	97,79	92,59
Con J48 (binarySplit=T) + SMO + IBk + LWL + ConjunctiveRule	**98,34**	**91,53**

Tabla 21. Valores de las medidas de exactitud para los metaclasificadores. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

Algoritmo	Sensibilidad (%)	Especificidad (%)
AdaBoostM1	95,58	87,30
Bagging	97,51	53,44
Stacking	**98,34**	**91,53**

Tras un análisis de la tabla anterior, es evidente que el metaclasificador que mejor funciona para este conjunto de datos es *Stacking*, con una sensibilidad del 98,34 % y la mejor especificidad de los tres algoritmos usados en el experimento (91,53 %).

3.3. Experimento con instancias artificiales

Los algoritmos probados anteriormente tienen un inconveniente, es difícil interpretar el modelo que brindan. En la literatura se aborda este tema, Witten y Frank[13] proponen una estrategia que consiste en crear instancias artificiales a partir de las reales y clasificarlas con el algoritmo robusto pero complicado de interpretar, luego añadir estas nuevas instancias al conjunto de datos original y usar este nuevo conjunto para entrenar a un algoritmo más interpretable como son los árboles de decisión o las reglas.

Para seguir esta estrategia se utilizó la herramienta WEKA, en ella se ofrece un algoritmo cuyo fin no es este precisamente, pero por su filosofía de trabajo pudo utilizarse para los fines deseados, este algoritmo es el *SMOTE* (*Synthetic Minority Over-sampling Technique*), que sobre muestrea la clase minoritaria por medio de la creación de ejemplos sintéticos, y que normalmente se usa en problemas de clases no balanceadas. Este algoritmo tiene varios parámetros, uno define la clase a la que se le va a aplicar la técnica, otro el número de vecinos más cercanos a usar, un tercero para definir el porciento de instancias a crear con SMOTE y el último para introducir la semilla que utilizará el método para hacer el sobre muestreo aleatorio. Se configuró para que creara instancias artificiales para ambas clases, con *K*=1 y para que creara el total de instancias artificiales posibles (100 %). Estas instancias fueron revisadas por los especialistas para descartar casos artificiales que tuvieran errores y luego se clasificaron usando el método Stacking, que fue, de los dos métodos elegidos anteriormente, el que mejor funcionó de forma global. Luego fueron incorporadas al conjunto de datos original y con este nuevo conjunto se entrenó al algoritmo *J48* que es un clásico entre los árboles de decisión y que es muy fácil de interpretar.

Finalmente quedó un conjunto de entrenamiento con 1112 instancias con la distribución de clases que se muestra en el Gráfico 10.

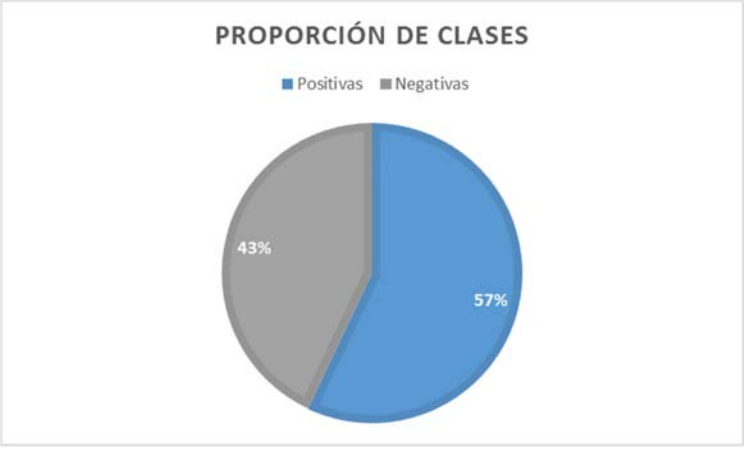

Gráfico 10. Proporción de las clases en el nuevo conjunto de entrenamiento.

Si hicieron varias pruebas con este algoritmo variándole algunos parámetros, los resultados se muestran en la Tabla 22.

Tabla 22. Pruebas realizadas con algoritmos basados en reglas y árboles de decisión. Cuando los valores de sensibilidad son iguales o similares, se prefiere en algoritmo con mejor especificidad.

J48	**Sensibilidad (%)**	**Especificidad (%)**
***J48* (por defecto WEKA)**	94,48	89,33
binarySplit=T	92,9	89,33
binarySplit=F & subtreeRaising=F	**94,79**	89,54
ConjunctiveRule	**97,63**	57,95
***NNge* numAttemptsOfGeneOption=25**	94,64	88,28
PART	93,38	89,33
LMT	**95,11**	**94,14**
REPTree unpruned	94,01	88,70

Se probaron algoritmos basados en reglas y árboles de decisión. Para *J48* además de estos parámetros que se muestran, se probaron diferentes valores de *confidenceFactor*, pero ninguno mejoró el rendimiento del valor que trae el algoritmo por defecto en WEKA (C=0,25), se probó también no podando el árbol pero tampoco se obtuvieron resultados superiores.

El método que mayor sensibilidad tiene es *ConjunctiveRule* pero tiene una especificidad muy baja (57,95 %). Mientras que *LMT* con una sensibilidad del 95,11% y 94,14 % de especificidad, es el segundo mejor en sensibilidad y tiene mejor balance entre las dos medidas.

Vale aclarar que a este nuevo conjunto se le aplicaron las medidas de complejidad, para comprobar que este grupo de instancias que se habían añadido no lo complejizaran más. Los resultados se muestran en la Tabla 23 comparados con los valores de las medidas de complejidad del conjunto de datos antes de sumarle los casos artificiales. Se puede ver que lejos de aumentar, la complejidad disminuye

para todas las medidas que, en el Capítulo 2, concluimos que eran las más importantes en este tipo de problemas.

Tabla 23. Comparación entre las medidas de complejidad del conjunto de datos de entrenamiento preprocesado y del nuevo que contiene a las instancias artificiales.

Conjuntos de datos / Medidas	Original	Nuevo
N1	0,171	0,095
N2	0,328	0,203
N4	0,226	0,204

3.4. Costos

La medición de la precisión predictiva a menudo se considera el evaluador clave del éxito de una aplicación. Si se trata de una organización de marketing, y se envía un correo masivo, el interés será tener la tasa de respuesta más alta. En esta aplicación la mayoría de los usuarios no responden, por lo que incluso pequeños cambios en la probabilidad pueden ser extremadamente valiosos. En el diagnóstico médico es todo lo contrario, lo que se espera es una precisión muy alta. La posibilidad de error no es la única preocupación, los errores tienen costos, el riesgo de fallar en el diagnóstico debe ser más alto que el costo de realizarle más pruebas al paciente[80].

Para tener en cuenta el costo existen dos estrategias: ignorar los costos en el proceso de aprendizaje y tenerlos en cuenta en el proceso de evaluación; y tenerlos en cuenta en el aprendizaje e ignorarlos en la evaluación.

La herramienta WEKA brinda dos metaclasificadores para este tipo de aprendizaje, la matriz de costos puede suministrarse como un parámetro o cargarla desde un archivo con la opción *onDemandDirectory*.

- *CostSensitiveClassifier* provee dos opciones, la primera redistribuye el peso de las instancias de entrenamiento de acuerdo al costo total asignado a cada clase

(aprendizaje sensible a costos), la segunda predice la clase con el menor error de clasificación en lugar de la más probable (clasificación sensible a costos).

- *MetaCost* genera un clasificador sensible a costos a partir de un algoritmo de base.

Estos metaclasificadores se probaron para los dos algoritmos “ganadores” de las dos primeras comparaciones (*IBk* y *Stacking*) y para el *J48* obtenido del experimento. La matriz de costos que se utilizó, según el criterio de los especialistas, fue (Tabla 24):

Tabla 24. Matriz de costos.

		CLASE PREDICHA	
		SÍ	**NO**
CLASE REAL	**SÍ**	0	30
	NO	1	0

Los resultados de aplicar costos a estos algoritmos se muestran en la Tabla 25. El algoritmo *IBk* no muestra ningún cambio al combinarlo con *CostSensitiveClassifier*.

Tabla 25. Pruebas de costos realizadas a los algoritmos *IBk, Stacking* y *J48*.

IBk	**Sensibilidad (%)**	**Especificidad (%)**
MetaCost	99,72	60,85
Stacking		
***CostSensitiveClassifier* (minimizeExpectedCost=F)**	99,45	28,04
***CostSensitiveClassifier* (minimizeExpectedCost=T)**	**99,72**	**66,14**
MetaCost	99,72	51,85
LMT		

***CostSensitiveClassifier* (minimizeExpectedCost=F)**	99,05	49,16
***CostSensitiveClassifier* (minimizeExpectedCost=T)**	97,48	85,56
MetaCost	97,79	72,18

La combinación de algoritmos que mejor funciona es la de *Stacking* con *CostSensitiveClassifier,* con una sensibilidad del 99,72 % y la variante de aprendizaje que mejor funciona es la de clasificación basada en costos, es decir, el sistema toma en cuenta la matriz de costos a la hora de clasificar las instancias y la ignora durante el proceso de entrenamiento.

Finalmente este es el modelo seleccionado para construir el sistema de apoyo a la toma de decisiones, su configuración se muestra en la Figura 3.

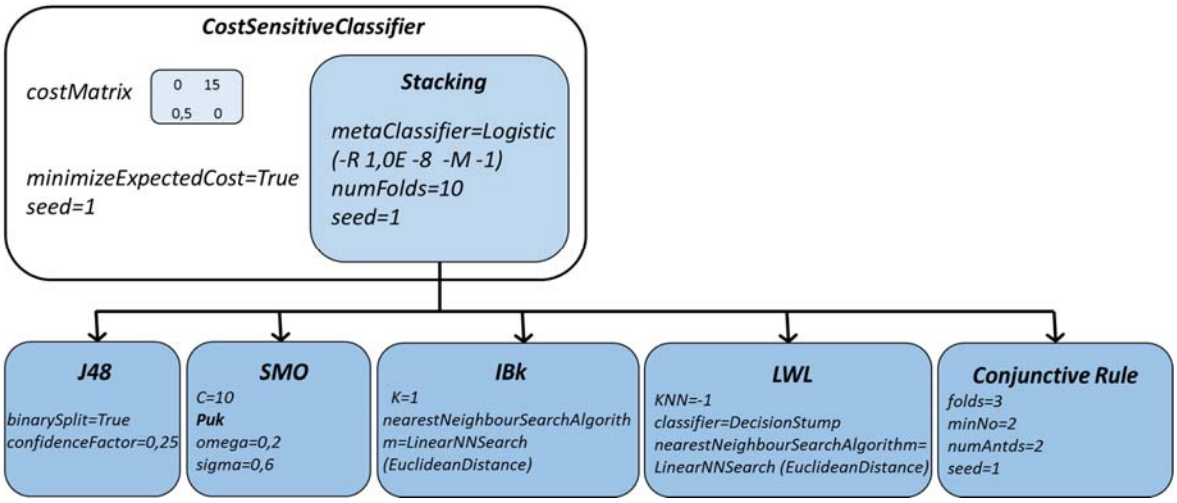

Figura 3. Configuración del modelo elegido.

3.5. Evaluación con el conjunto de prueba y comparación con la clasificación de los especialistas

Para evaluar el sistema en la práctica, se tomó un conjunto de pacientes a los que ya se les había hecho la prueba pero que el sistema no conocía y se clasificaron usando el modelo elegido. Estos casos fueron además clasificados por un grupo de especialistas y se contrastaron los resultados para ver qué tal se comportaba el

sistema. El grupo de especialistas estuvo compuesto por ocho Doctores en Medicina, cinco de ellos especializados en Hemodinámica y Cardiología Intervencionista que realizan la coronariografía a diario y atienden con la misma frecuencia a pacientes con enfermedades del corazón, dos especializados en Electrofisiología y un Cirujano Cardiovascular.

Para confeccionar el conjunto de prueba se recopilaron casos del sistema AngioDat 2.0, utilizado en el Salón de Hemodinámica. El conjunto se compone de 40 instancias elegidas aleatoriamente, sin valores perdidos, con 31 casos positivos y 9 negativos realizados en el año 2016 en dicho Salón. El grupo de casos seleccionados inicialmente era más extenso, pero la mayoría de los casos presentaban valores perdidos, se decidió solicitar al Archivo del Hospital las Historias Clínicas de estos pacientes para revisarlas e intentar rellenar estos valores, se encontraron muchos valores ausentes y finalmente se decidió descartar las instancias que todavía presentaban valores perdidos y utilizar un conjunto de casos que representara fielmente la realidad.

Este grupo de casos se les presentó a los especialistas sin mostrarles la clase real a la que pertenecía cada paciente y se les pidió que los clasificaran de acuerdo a sus conocimientos y experiencia. En la Tabla 26 se muestran los resultados de la clasificación del sistema y en la 27 la de los especialistas.

Tabla 26. Resultados de la clasificación del sistema.

		CLASE PREDICHA	
		SÍ	**NO**
CLASE REAL	**SÍ**	29	2
	NO	6	3

Tabla 27. Resultados de la clasificación del grupo de especialistas.

		CLASE PREDICHA	
		SÍ	**NO**
CLASE REAL	**SÍ**	21	10
	NO	5	4

El análisis demostró que el sistema comete 8 errores, 6 falsos positivos y 2 falsos negativos, para un 93,55 % de sensibilidad; mientras que los especialistas, cometen 15 errores, 10 falsos negativos y 5 falsos positivos con una sensibilidad del 80,77 %. Cinco casos son declarados por los especialistas como dudosos pero dan su valoración y esta coincide con la realidad así que se asumieron como correctas. El sistema y los especialistas coinciden con la realidad en 23 de los pacientes y de los 8 errores que comete el sistema con este conjunto de prueba, en 6 coincide con los especialistas y otro es un caso que los especialistas, aunque lo clasificaron correctamente, lo declaran como dudoso. Se evidencia que para ese conjunto de prueba la sensibilidad del sistema es más alta que la de los especialistas y precisamente ese es el objetivo, lograr un sistema que pueda ofrecer una opinión experta en los Centros de Diagnóstico y que su precisión sea mejor o, comparable al criterio del/los especialista(s).

3.6. Conclusiones parciales

- Durante la selección del mejor clasificador simple se compararon los algoritmos *SMO*, *IBk* y *ConjunctiveRule*, de ellos el que mejor comportamiento mostró fue *IBk*, alcanzando un 96,98 % de sensibilidad y una especificidad de más del 90 %.
- Se compararon tres metaclasificadores: *AdaBoostM1, Bagging* y *Stacking*, el que mejor valores de exactitud mostró fue este último con un 98,34 % de sensibilidad y una especificidad de 91,53 %.

- Mediante la adición de instancias artificiales al conjunto de entrenamiento se logró, con el algoritmo de árboles de decisión *LMT*, una sensibilidad del 95,11 % y una especificidad de más del 90 %.
- Se tuvieron en cuenta los costos al seleccionar el modelo final, para ello se compararon los resultados de dos algoritmos *MetaCost* y *CostSensitiveClassifier.* El que mejor resultados tuvo fue este último usando *Stacking* como algoritmo de base. La sensibilidad lograda es del 99,72 % y la especificidad por encima del 50 %.
- Se evaluó el modelo con un conjunto de prueba y se compararon sus resultados con los de los especialistas para el mismo conjunto. El desempeño del modelo final superó en un 12,78 % el criterio de los especialistas sobre la realización o no de la coronariografía.

CONCLUSIONES

1. La Minería de Datos incluye técnicas de preprocesamiento de datos que contribuyen a la disminución de la complejidad del conjunto de entrenamiento. Las medidas de exactitud apoyan la elección del mejor modelo de clasificación. Las características de los datos recopilados en el Salón de Hemodinámica del Cardiocentro “Ernesto Che Guevara” permiten su utilización en una aplicación de Minería de Datos.
2. En el preprocesamiento se realizaron, manualmente, las tareas de limpieza de datos y normalización, eliminando atributos irrelevantes, mejorando la categorización de algunos, eliminando valores fuera de rango y unificando unidades de medida. Se imputaron valores perdidos, se aplicó selección de atributos y de instancias, utilizando herramientas de Minería de Datos. Se obtuvo un conjunto de entrenamiento con 14 atributos y 551 instancias.
3. El modelo seleccionado es el resultante de la aplicación de *CostSensitiveClassifier*, con una matriz de costos que pondera a los pacientes enfermos que el sistema clasifica como sanos. Este algoritmo usa como método de base al metaclasificador *Stacking* y este a su vez utiliza una combinación de los algoritmos *J48, SMO, IBk, LWL* y *ConjunctiveRule.* El modelo tiene una sensibilidad del 99,72 % y una especificidad de más del 50 %.
4. Se validó el modelo obtenido para un conjunto nuevo de casos y se compararon sus resultados con la clasificación realizada a los mismos casos por un grupo de especialistas. Esta comparación arrojó que el sistema obtuvo resultados superiores en un 12,78 % en comparación con el criterio de los expertos.

RECOMENDACIONES

1. Homogeneizar la recolección de datos para que contengan atributos que pueden ser más discriminantes, esto sería posible gracias a la instalación del sistema AngioDat 2.0.
2. Extenderlo a los Centros de Diagnóstico que pertenecen al Cardiocentro “Ernesto Che Guevara”.

BIBLIOGRAFÍA

[1] «Angiografía | Centro Médico Teknon». [En línea]. Disponible en: http://www.teknon.es/es/servicio-de-diagnosticos/diagnostico-por-la-imagen/angiografia. [Accedido: 29-sep-2016].

[2] «Cateterismo cardiaco y angiografía coronaria diagnósticos | Harrison. Principios de Medicina Interna, 18e | HarrisonMedicina | McGraw-Hill Medical». [En línea]. Disponible en: http://harrisonmedicina.mhmedical.com/Content.aspx?bookId=865&sectionId=68945085. [Accedido: 29-sep-2016].

[3] «socime.net - Sociedad de Cardiología Intervencionista de México». [En línea]. Disponible en: http://www.socime.net/paciente.aspx. [Accedido: 29-sep-2016].

[4] F. L. Moreno Martínez, C. Serrano Poyato, A. Alonso Moreno, y I. Delgado Solís, «Indicaciones y contraindicaciones del cateterismo cardíaco diagnóstico y terapéutico.», en *Manual de enfermería en cardiología intervencionista y hemodinámica. Protocolos unificados*, Madrid: Artes Gráficas Diumaró, 2007, pp. 57-67.

[5] C. J. Davidson y R. O. Bonow, «Cardiac catheterization», en *Braunwald´s Heart Disease. A textbook of cardiovascular medicine.*, 7th ed., Philadelphia: Elsevier Saunders, 2005, pp. 395-422.

[6] A. López Farré y C. Macaya Miguel, *Libro de la Salud Cardiovascular*. Fundación BBVA, 2009.

[7] «¿Cual es el precio aproximado de un cateterismo cardíaco con implante...», *Doctoralia*. [En línea]. Disponible en: http://www.doctoralia.es/pruebamedica/cateterismo+cardiaco+con+angiocardiografia+y+coronariografia-197/pregunta/cual-es-el-precio-aproximado-de-un-cateterismo-cardiaco-con-implante-de-muelle-442112.

[8] «Healthcare Bluebook - Procedure Details». [En línea]. Disponible en: https://healthcarebluebook.com/page_ProcedureDetails.aspx?id=361&dataset=MD.

[9] «How much does a private coronory angiogram cost in the UK? | Private Healthcare UK». [En línea]. Disponible en: http://www.privatehealth.co.uk/conditions-and-treatments/coronary-angiogram/costs/.

[10] D. Gutiérrez Borrás, A. M. Álvarez González, B. González Camacho, y M. González Castellanos, «Módulo para diagnosticar la necesidad de realizar la coronariografía a los pacientes del Cardiocentro “Ernesto Che Guevara”», Diploma, Universidad Central «Marta Abreu» de Las Villas, Santa Clara, 2016.

[11] J. Han, M. Kamber, y J. Pei, *Data Mining Concepts and Techniques*, 3rd ed. USA: Morgan Kaufmann, 2012.

[12] R. Nisbet, G. Miner, y J. Elder, *Handbook of Statistical Analysis and Data Mining Applications*. Boston: Elsevier Science, 2009.

[13] I. H. Witten, E. Frank, y M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011.

[14] U. M. Fayyad , G. Piatetsky-Shapiro, y P. Smyth , «From data mining to knowledge discovery: an overview as chapter», en *Advances in Knowledge Discovery and Data Mining*, San Francisco: American Association for Artificial Intelligence, 1996, pp. 37-54.

[15] J. H. Friedman, «Data mining and Statistics: What’s the connection?», *Comput. Sci. Stat.*, vol. 29, n.º 1, pp. 3-9, 1998.

[16] P. J. Tuya González, J. J. Dolado Cosín, e I. Ramos Román, *Técnicas cuantitativas para la gestión en la ingeniería del software*. Oleiros, La coruña: Netbiblo, 2007.

[17] T. G. Ditterrich, «Machine learning research: four current direction», *Artif. Intell. Magzine*, vol. 4, pp. 97-136, 1997.

[18] S. M. Weiss y N. Indurkhya, *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann, 1998.

[19] S. García, J. Luengo, y F. Herrera, *Data Preprocessing in Data Mining*. Springer, 2014.

[20] D. Pyle, *Data Preparation for Data Mining*. Morgan Kaufmann, 1999.

[21] W. Kim, B.J. Choi, E.K. Hong, S.K. Kim, y D. Lee, «A taxonomy of dirty data», *Data Min. Knowl. Discov.*, vol. 7, n.º 1, pp. 81-99, 2003.

[22] J. Luengo, S. García, y F. Herrera, «On the choice of the best imputation methods for missing values considering three groups of classification methods», *Knowl. Inf. Syst.*, vol. 32, n.º 1, pp. 77-108, 2012.

[23] J. Barnard y X.-L. Meng, «Applications of multiple imputation in medical studies: from AIDS to NHANES», *Stat. Methods Med. Res.*, vol. 8, n.º 1, pp. 17-36, 1999.

[24] A. Farhangfar, L. Kurgan, y J. Dy, «Impact of imputation of missing values on classification error for discrete data», *Pattern Recognit.*, vol. 41, n.º 12, pp. 3692-3705, 2008.

[25] H. Liu y H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*. Springer Science & Business Media, 2012.

[26] H. Liu y H. Motoda, *Computational Methods of Feature Selection*. CRC Press, 2007.

[27] R. Kohavi y G. H. John, «Wrappers for feature subset selection», *Artif. Intell.*, vol. 97, n.º 1, pp. 273-324, 1997.

[28] M. A. Hall y G. Holmes, «Benchmarking attribute selection techniques for data mining», 2000.

[29] H. Liu y H. Motoda, *Instance Selection and Construction for Data Mining*. Springer Science & Business Media, 2013.

[30] T. K. Ho y M. Basu, «Complexity measures of supervised classification problems», *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, n.º 3, pp. 289-300, 2002.

[31] E. Bernadó-Mansilla y N. Macià-Antolínez, «Modeling Problem Transformations based on Data Complexity», en *Proceedings of the 2007 conference on Artificial Intelligence Research and Development*, 2007, pp. 133-140.

[32] J. H. Friedman y L. C. Rafsky, «Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests», *Ann. Stat.*, pp. 697-717, 1979.

[33] A. Hoekstra y R. P. Duin, «On the nonlinearity of pattern classifiers», en *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 1996, vol. 4, pp. 271-275.

[34] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, y R. Uthurusamy, Eds., *Advances in knowledge discovery and data mining*. American Association for Artificial Intelligence, 1996.

[35] C. M. Bishop, *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.

[36] B. D. Ripley, *Pattern recognition and neural networks.* Cambridge University Press Cambridge, UK:, 1996.

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

[38] P. Cheeseman, M. Self, J. Kelly, y J. Stutz, «Bayesian Classification», *Artif. Intell. Magzine*, pp. 607-611, 1996.

[39] A. Agresti, *Categorical Data Analysis*. John Wiley & Sons, 2003.

[40] C. C. Christensen, *Indiana Fixed Station Statistical Analysis 1997*. Indiana Department of Environmental Management, Office of Water Management, Assessment Branch, Surveys Section, 1998.

[41] R. Kohavi, «A study of cross-validation and bootstrap for accuracy estimation and model selection», en *International Joint Conference on Artificial Intelligence*, 1995, vol. 14, pp. 1137-1145.

[42] L. Breiman, J. Friedman, C. J. Stone, y R. A. Olshen, *Classification and Regression Trees*. Taylor & Francis, 1984.

[43] J. R. Quinlan, «Induction of decision trees», *Mach. Learn.*, vol. 1, n.º 1, pp. 81-106, 1986.

[44] P. Harrington, *Machine Learning in Action*. Manning Publications Company, 2011.

[45] M. Gestal Pose, «Introducción a las Redes Neuronales Artificiales».

[46] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[47] D. W. Aha, D. Kibler, y M. K. Albert, «Instance-based learning algorithms», *Mach. Learn.*, vol. 6, n.º 1, pp. 37-66, 1991.

[48] «Aprendizaje Basado en Instancias». [En línea]. Disponible en: https://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/node69.html.

[49] G. A. Betancourt, «Las máquinas de soporte vectorial (SVMs)», *Sci. Tech.*, vol. 1, n.º 27, 2005.

[50] B. Schölkopf y A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

[51] J. Fürnkranz, D. Gamberger, y N. Lavrač, *Foundations of Rule Learning*. Springer Science & Business Media, 2012.

[52] L. Rokach y O. Maimon, *Data Mining with Decision Trees: Theory and Applications*. World Scientific, 2014.

[53] P. M. Bossuyt *et al.*, «The STARD statement for reporting studies of diagnostic accuracy: explanation and elaboration», *Ann. Intern. Med.*, vol. 138, n.º 1, pp. W1-12, 2003.

[54] S. Bravo-Grau y J. P. Cruz, «Estudios de exactitud diagnóstica: Herramientas para su Interpretación», *Rev. Chil. Radiol.*, vol. 21, n.º 4, pp. 158-164, 2015.

[55] D. G. Altman y J. M. Bland, «Diagnostic tests. 1: Sensitivity and specificity.», *BMJ*, vol. 308, n.º 6943, p. 1552, 1994.

[56] J. Cerda y L. Cifuentes, «Uso de tests diagnósticos en la práctica clínica (Parte 1): Análisis de las propiedades de un test diagnóstico», *Rev. Chil. Infectol.*, vol. 27, n.º 3, pp. 205-208, 2010.

[57] «KEEL: A software tool to assess evolutionary algorithms for Data Mining problems (regression, classification, clustering, pattern mining and so on)», 07-oct-2016. [En línea]. Disponible en: http://www.keel.es/.

[58] «Weka 3 - Data Mining with Open Source Machine Learning Software in Java», 07-oct-2016. [En línea]. Disponible en: http://www.cs.waikato.ac.nz/ml/weka/.

[59] «Valores normales de laboratorio», *Merck Manuals - Professional Version*. [En línea]. Disponible en: http://www.merckmanuals.com/es-pr/professional/ap%C3%A9ndices/valores-normales-de-laboratorio/.

[60] J. W. Grzymala-Busse y M. Hu, «A comparison of several approaches to missing attribute values in data mining», en *International Conference on Rough Sets and Current Trends in Computing*, 2000, pp. 378-385.

[61] J. W. Grzymala-Busse, L. K. Goodwin, W. J. Grzymala-Busse, y X. Zheng, «Handling missing attribute values in preterm birth data sets», en *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*, 2005, pp. 342-351.

[62] G. E. Batista y M. C. Monard, «An analysis of four missing data treatment methods for supervised learning», *Appl. Artif. Intell.*, vol. 17, n.º 5-6, pp. 519-533, 2003.

[63] O. Troyanskaya *et al.*, «Missing value estimation methods for DNA microarrays», *Bioinformatics*, vol. 17, n.º 6, pp. 520-525, 2001.

[64] D. Banks, L. House, F. R. McMorris, P. Arabie, y W. Gaul, *Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, 15–18 July 2004*. Springer Science & Business Media, 2011.

[65] F. Honghai, C. Guoshun, Y. Cheng, Y. Bingru, y C. Yumei, «A SVM regression based approach to filling in missing values», en *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, 2005, pp. 581-587.

[66] A. K. Wong y D. K. Chiu, «Synthesizing statistical knowledge from incomplete mixed-mode data», *IEEE Trans. Pattern Anal. Mach. Intell.*, n.º 6, pp. 796-805, 1987.

[67] T. Schneider, «Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values», *J. Clim.*, vol. 14, n.º 5, pp. 853-871, 2001.

[68] S. Oba, M. Sato, I. Takemasa, M. Monden, K. Matsubara, y S. Ishii, «A Bayesian missing value estimation method for gene expression profile data», *Bioinformatics*, vol. 19, n.º 16, pp. 2088-2096, 2003.

[69] H. Kim, G. H. Golub, y H. Park, «Missing value estimation for DNA microarray gene expression data: local least squares imputation», *Bioinformatics*, vol. 21, n.º 2, pp. 187-198, 2005.

[70] M. A. Hall, «Correlation-based feature selection for machine learning», PhD, The University of Waikato, 1999.

[71] H. Liu y R. Setiono, «A probabilistic approach to feature selection-a filter solution», en *ICML*, 1996, vol. 96, pp. 319-327.

[72] J. Lorenzo, «Selección de Atributos en Aprendizaje Automático basado en la Teoría de la Información», PhD thesis, U. de Las Palmas de Gran Canaria, Dpto. de Informática y Sistemas, 2001.

[73] I. Tomek, «An experiment with the edited nearest-neighbor rule», *IEEE Trans. Syst. Man Cybern.*, n.º 6, pp. 448-452, 1976.

[74] C. E. Brodley, «Recursive automatic bias selection for classifier construction», *Mach. Learn.*, vol. 20, n.º 1-2, pp. 63-94, 1995.

[75] D. L. Wilson, «Asymptotic properties of nearest neighbor rules using edited data», *IEEE Trans. Syst. Man Cybern.*, n.º 3, pp. 408-421, 1972.

[76] S. Vluymans, N. Verbiest, C. Cornelis, y Y. Saeys, «Instance selection for imbalanced data», en *Workshop'Rough Sets: Theory and Applications'(RST&A); held at the 2014 Joint Rough Set symposium (JRS 2014)*, 2014.

[77] K. Hattori y M. Takahashi, «A new edited k-nearest neighbor rule in the pattern classification problem», *Pattern Recognit.*, vol. 33, n.º 3, pp. 521-528, 2000.

[78] J. C. Riquelme, J. S. Aguilar-Ruiz, y M. Toro, «Finding representative patterns with ordered projections», *Pattern Recognit.*, vol. 36, n.º 4, pp. 1009-1018, 2003.

[79] N. Ye, *The Handbook of Data Mining*. Lawrence Erlbaum Associates, Incorporated, 2004.

ANEXO 1. DATOS DE LOS ESPECIALISTAS

- MSc. Dr. Rosendo Seferino Ybargollín Hernández. Doctor en Medicina. Especialista de Primer y Segundo Grados en Cardiología. Especialista en Hemodinámica y Cardiología Intervencionista. Máster en Urgencias Médicas. Subdirector de Cardiología Intervencionista. Cardiólogo en el Hospital Provincial Universitario Cardiocentro "Ernesto Che Guevara" de Villa Clara. Años de experiencia: 36.
- MSc. Dr. Francisco Luis Moreno Martínez. Doctor en Medicina. Especialista de Primer y Segundo Grados en Cardiología. Especialista en Hemodinámica y Cardiología Intervencionista. Máster en Urgencias Médicas. Vicepresidente del Capítulo de Villa Clara de la Sociedad Cubana de Cardiología. Cardiólogo en el Hospital Provincial Universitario Cardiocentro "Ernesto Che Guevara" de Villa Clara. Miembro del *American College of Cardiology*. Años de experiencia: 21.
- Dr. Luis Felipe Vega Fleites. Doctor en Medicina. Especialista de Primer Grado en Cardiología. Especialista en Hemodinámica y Cardiología Intervencionista. Jefe del Servicio de Hemodinámica y Cardiología Intervencionista. Cardiólogo en el Hospital Provincial Universitario Cardiocentro "Ernesto Che Guevara" de Villa Clara. Años de experiencia: 23.
- MSc. Dr. Jesús Arturo Satorre Ygualada. Doctor en Medicina. Especialista de Primer y Segundo Grados en Cardiología. Máster en Urgencias Médicas. Presidente del Capítulo de Villa Clara de la Sociedad Cubana de Cardiología. Cardiólogo en el Hospital Provincial Universitario Cardiocentro "Ernesto Che Guevara" de Villa Clara. Años de experiencia: 22.
- Dr. C. Elibet Chávez González. Doctor en Medicina. Especialista de Primer y Segundo Grados en Cardiología. Especialista en Arritmias y Estimulación Cardíaca. Máster en Urgencias Médicas. Doctor en Ciencias Médicas. Jefe de la Sección de Marcapasos y Arritmias del Grupo Nacional de Cardiología. Cardiólogo en el Hospital Provincial Universitario Cardiocentro "Ernesto Che Guevara" de Villa Clara. Miembro del *American College of Cardiology.* Años de experiencia: 15.

- MSc. Dr. Iguer Fernando Aladro Miranda. Doctor en Medicina. Especialista de Grado en Cardiología. Especialista en Hemodinámica y Cardiología Intervencionista. Máster en Urgencias Médicas. Jefe de la Sala de Hospitalización de Hemodinámica y Cardiología Intervencionista. Cardiólogo en el Hospital Provincial Universitario Cardiocentro “Ernesto Che Guevara” de Villa Clara. Años de experiencia: 14.
- Dr. Álvaro Luis Lagomasino Hidalgo. Doctor en Medicina. Especialista de Primer Grado en Cirugía General. Especialista de Segundo Grado en Cirugía Cardiovascular. Profesor de Mérito. Profesor Consultante. Vicesecretario de la Sociedad Cubana de Cardiología. Subdirector Quirúrgico. Cirujano Principal en el Hospital Provincial Universitario Cardiocentro “Ernesto Che Guevara” de Villa Clara. Años de experiencia: 45.
- Dr. Leonardo López Ferrero. Doctor en Medicina. Especialista de Primer y Segundo Grados en Cardiología. Especialista en Hemodinámica y Cardiología Intervencionista. Jefe del Servicio de Hemodinámica y Cardiología Intervencionista. Cardiólogo en el Instituto de Cardiología y Cirugía Cardiovascular de La Habana. Años de experiencia: 28.