Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura

Maestría en Tecnologías de la Información

Trabajo Final

Implementación de procesos de explotación de información y visualización de datos: Estudio de caso en una Unidad de Cuidados Intensivos Coronarios.

Autor: Lic. BÁEZ, E. Darío

Director: Dra. María Inés Pisarello

Co-Director: Dr. Emanuel Irrazábal

Año 2.022

Resumen

En Argentina, la *enfermedad cardiovascular* lidera el ranking de muertes en adultos, se encuentra una cantidad preponderante de pacientes con patologías cardiovasculares clásicas como enfermedad coronaria e insuficiencia cardíaca. La atención de estos pacientes en su tratamiento tiene un elevado costo en términos de recursos humanos y equipos.

La era digital permite acortar las distancias médica al paciente, ofrece oportunidades de mejora en la calidad de atención y facilita el acceso a la misma buscando mejorar la exactitud, puntualidad, calidad y eficacia general de una decisión médica, siendo la Inteligencia Artificial (IA) una de las herramientas más utilizadas para dicho fin mediante la aplicación de los diferentes algoritmos de aprendizaje automático. El presente trabajo tiene como objetivo elaborar un modelo de aprendizaje automático basado en IA utilizando el algoritmo de aprendizaje supervisado, árbol de decisión que permita analizar y pronosticar el grado de afectación en los pacientes según su registro hospitalario, para su correspondiente hospitalización en una Unidad de Cuidados Intensivos Coronarios (UCIC). Es preciso tener en cuenta criterios de usabilidad y manejo oportuno de la información. La metodología utilizada sigue el método científico lo que nos permitió indagar y examinar todo tipo de estudios, resultados e investigaciones. El dataset utilizado fue obtenido del repositorio, provenientes del Instituto de Cardiología de Corrientes (ICC). Para que esos datos se conviertan en conocimiento, necesitan ser procesados y analizados a través de métodos estadísticos usando Inteligencia Artificial.

Palabras claves: Registro hospitalario, Informática médica, Modelos de procesos, Inteligencia artificial, Aprendizaje Automático, Árboles de Decisión, Análisis, Predicción, Pronóstico.

Abstract

Currently, heart diseases are one of the first reasons for death in adults in Argentina. Thus, there is a great number of patients with traditional heart diseases such as coronary disease and cardiac failure. Apart, its treatment is expensive, not only in relation to human resources but equipment too.

The arrival of the digital era has shortened the distance between the patient and its medical assistance, offering chances to improve the meeting with the patient facilitating its quality, punctuality and efficiency. Artificial Intelligence is in this sense, one of the most frequently used tools through which a range of Machine Learning algorithms and a variety of application domains are used.

The aim of this work is to generate an AI-based machine learning model through a supervised Machine Learning algorithm that allows drawing a decision tree to predict the level of severity of a patient's disease according to their hospital record, to decide if it is hospitalization necessary or not. Methodology follows the scientific approach and is based on factual data taken from Instituto de Cardiología de Corrientes (ICC). For these data to become knowledge, they need to be processed and analyzed through statistical methods using artificial intelligence.

Keywords: *hospital register, medical informatics, model of processes, Artificial Intelligence, Machine Learning, decision tree, analysis, prediction, forecast.*

Agradecimientos

Agradezco a Dios por la bendición de la vida, por guiarme a lo largo de mi existencia, ser el apoyo y fortaleza en aquellos momentos de dificultad y debilidad.

A las personas que me apoyaron e hicieron posible que este trabajo se realice con éxito, en especial, a mi tutora Dra. María Inés Pisarello, sin su paciencia y constancia, este trabajo no lo hubiese logrado tan fácil.

Al Dr. Emanuel Irrazábal quien guio por buen camino este proyecto. Al Ing. Fabián Bobadilla, de la Empresa Aliare, asesor en este proyecto, quien, con su aporte, orientó el desarrollo del presente trabajo. Al Dr. Jorge Parras, por su calidad humana, profesionalismo, paciencia invertidos.

A mi esposo y compañero Juan Carlos, por su cariño y apoyo incondicional durante todo el proceso, por estar en todo momento. A mi amiga y colega, Mgtr. Sofía Vallejos, a quien quiero como a una hermana, por compartir momentos significativos. Porque somos y formamos un gran equipo.

Índice de contenidos

Índice de figuras

Índice de tablas

Índice de Abreviaturas

A

AA
Aprendizaje Automático, xxiii
AD
Arboles de decisión, xxxiii

C

CDSS
Clinical Decisión-Support Systems, xi

E

ECV
enfermedad cardiovascular, xii
ER
Emergency Room, xxii

F

FUNCACORR
Fundación Cardiológica Correntina para la Asistencia, Docencia e Investigaciones Médicas, xiv

H

HCE
Historias Clínicas Electrónicas, xi

I

IA
Inteligencia Artificial, i
IC
Insuficiencia Cardíaca, xii
ICC
Instituto de Cardiología de Corrientes, i

M

ML
Machine Learning, xxvii

P

PNES
Programa Nacional de Estadísticas de Salud, xii

R

RL
Regresión Logistica, xxvii

S

SADC
Sistemas de apoyo a la decisión clínica, xii

T

TIC
Tecnologías de la Información y la Comunicación, xxii

U

UCIC
Unidad de Cuidados Intensivos Coronarios, i

Capítulo I

Introducción

1 Introducción

Esta sección está conformada por la introducción, el planteamiento del problema, su descripción, objetivo general con sus objetivos específicos, el alcance del proyecto, justificación e importancia.

1.1 Contexto y justificación:

La información disponible relacionada con la medicina está en constante cambio debido a fenómenos de crecimiento y expansión, lo que se conoce como el fenómeno de la sobrecarga de información. Para mantenerse actualizado un profesional de la salud debería intentar la tarea de localizar la información que le es relevante de entre un total de fuentes que se le presentan a la una tasa de 6000 artículos por día [1]. Paralelamente al enorme desarrollo de los conocimientos biomédicos, y la tecnología informática también fue creciendo hasta brindar soluciones para el adecuado manejo de estos nuevos conocimientos. Como sostienen los médicos del Departamento de Informática Médica y Servicio de Clínica Médica Hospital Italiano de Buenos Aires – Argentina, los sistemas clínicos de soporte para la toma de decisiones ("Clinical Decision-Support Systems" - CDSS) representan la parte de los sistema de información sanitaria destinada a proveer la información necesaria, relevante, contextual, actualizada y consensuada en el momento en que el médico toma una decisión con respecto a la situación clínica de un paciente concreto [2].

Los sistemas de información se enfocaron en recoger datos para la gestión y no en incrementar la calidad asistencial, eficiencia y seguridad del paciente, que es a lo que debían orientarse los sistemas de información de salud[3]. Es frecuente el no aprovechamiento de la predicción del comportamiento futuro de algunos problemas de salud presentes en las Historias Clínicas Electrónicas (HCE), basado en el entendimiento del pasado. Por esto se impulsó la implementación de sistemas electrónicos diseñados para ayudar a la toma de decisiones clínicas. Los mismos tienen la capacidad de interactuar con expertos humanos y son sistemas independientes que recolectan o replican datos de otros sistemas de información. Están destinados a mejorar la exactitud, puntualidad, calidad y eficacia general de una decisión concreta o de un conjunto de decisiones relacionadas 103 [4].

La aplicación de técnicas de IA en medicina puede ser un aporte fundamental en los avances médicos, tanto por la ayuda para conseguir diagnósticos y tratamientos más precisos como por la reducción de costes que puede implicar (por ejemplo, consiguiendo tiempos menores de hospitalización). Un elemento esencial para la mejora de la práctica clínica es la implantación

de sistemas. Las técnicas de aprendizaje automático es el principal eje de la inteligencia artificial [5]. Estas tecnologías, posibilitan analizar gran cantidad de datos de manera rápida para identificar patrones o modelos, los cuales se pueden emplear para tomar decisiones óptimas o predecir situaciones.

Aplicando estos conocimientos en el área médica sería posible la toma de decisiones clínicas. Los sistemas de apoyo a la decisión clínica (SADC) son sistemas de conocimiento activo que usan conjuntos de datos de pacientes para generar recomendaciones médicas que se integren con la HCE. Los mismos tienen como objetivo último dar soporte a los profesionales de salud en la toma de decisiones y contribuir a mejorar la interacción entre la evidencia científica y la información del paciente. La falta de información dada la carencia de sistemas informáticos eficientes, dificulta la toma de decisiones informadas y oportunas. Estas problemáticas, empeorarán en los próximos años [6], donde se proyecta un aumento de la población adulto mayor en un 45% para el año 2020 según datos de los Censos Nacionales disponibles en Instituto Nacional de Estadística y Censos de la República Argentina [7].

Por otra parte, la insuficiencia cardíaca (IC) representa actualmente el síndrome cardiovascular más frecuente en la sociedad occidental y todas las tendencias indican un considerable impacto económico de esta entidad en los sistemas de salud de la mayoría de los países. El Programa Nacional de Estadísticas de Salud (PNES) [8] del Ministerio de Salud y Acción Social de la Nación; a través de registros permanentes en las estadísticas vitales, permite analizar la mortalidad en todo el país; y evaluando las Estadísticas de Prestaciones, Rendimientos y Morbilidad Hospitalaria se extraen conclusiones sobre egresos hospitalarios. Las enfermedades cardiovasculares son aquellas que afectan al corazón y a todo el sistema arterial. La enfermedad cardiovascular (ECV) (infarto de miocardio, accidente cerebrovascular e insuficiencia cardíaca) lidera el ranking en muertes, ya sea a nivel global como en Argentina (según la Fundación Cardiológica Argentina 100.000 muertes anuales, 280 muertes por día) [9], como así también en años perdidos de vida ajustados por discapacidad según la Sociedad Argentina de Cardiología, en estos dos últimos años de pandemia se ha profundizado la problemática.

1.2 Objetivos del trabajo

La aplicación de técnicas de IA en medicina son un aporte fundamental en los avances médicos, tanto por la ayuda tanto para lograr diagnosis y tratamientos precisos como por la reducción de costos involucrados en los tiempos de hospitalización.

Se propone vincular el contexto de innovación del proceso de la actividad tecno-científica con la construcción de conocimientos, utilizando algoritmos de aprendizaje automático para modelar y simular las inferencias implícitas de los expertos en la toma de decisiones. Los algoritmos de IA, descubren conocimiento o patrones proponiendo alternativas para entender la problemática planteada, mientras que los de aprendizaje automático permiten que las máquinas aprendan de los datos para que puedan dar resultados precisos. Se propone un sistema que permita la visualización de información relevante para apoyar la toma de decisiones, con el objetivo de maximizar los beneficios clínicos y sociales, y minimizar los costos asociados a la gestión de camas hospitalarias básicas, intermedias y críticas.

1.3 Objetivos específicos

Para cumplimentar el objetivo general definido en el apartado anterior, se han propuesto la realización de determinados objetivos específicos, que lo componen y se indican a continuación.

- ✓ Establecer si el proceso de minería de datos permite generar conocimiento de la información analizada, para la toma de decisiones.
- ✓ Relevar información científica sobre los procesos de Explotación de la Información referente a las unidades de cuidados intensivos coronarios.
- ✓ Analizar las técnicas y herramientas existentes para desarrollar un sistema para la visualización de los datos.
- ✓ Determinar la viabilidad de desarrollo de un Sistemas de apoyo a la toma de decisiones clínicas.
- ✓ Implementar el modelo de proceso de explotación de la información para el caso de estudio propuesto.
- ✓ Aplicar las técnicas y herramientas de visualización de datos, para mejorar los procesos de comunicación de resultados y contribuir más eficazmente a la toma de decisiones.
- ✓ Validar los procesos con datos conocidos y desconocidos que permitan la generalización buscada.
- ✓ Crear el dataset necesario para la elaboración del modelo de aprendizaje automático.
- ✓ Desarrollar el modelo aprendizaje automático haciendo uso de los árboles de decisión implementado en Python para la clasificación, predicción y pronóstico del grado de afectación en pacientes con enfermedades cardiovasculares.

- ✓ Analizar los datos obtenidos y generar los informes necesarios para su adecuada interpretación.

1.4 Contenidos y organización del trabajo

Los contenidos que se han decidido tener en cuenta en este trabajo son:

- ✓ Estudiar los fundamentos teóricos del aprendizaje automático supervisado y no supervisado.
- ✓ Conocer los distintos tipos de tareas que existen en el aprendizaje automático.
- ✓ Aprender los conceptos básicos de cómo generar un algoritmo de predicción de la manera más precisa posible.
- ✓ Estudiar y aplicar técnicas comunes para:
    - Realizar un estudio previo de los datos utilizados para el modelo.
    - Generar el modelo de predicción.
    - Analizar los resultados del modelo mediante las distintas métricas en función del tipo de aprendizaje automático en el que nos encontramos.
    - Aplicar y estudiar algunas de las técnicas más conocidas y empleadas en esta rama de la inteligencia artificial.

1.5 Fundamentación

El Instituto de Cardiología de Corrientes (ICC) "Juana F. Cabral" fue creado por ley provincial el 23 de julio de 1986 como un organismo estatal de naturaleza autárquica. Ese mismo año fue concesionada la administración a la Fundación Cardiológica Correntina para la Asistencia, Docencia e Investigaciones Médicas (FUNCACORR). ONG sin fines de lucro. Allí se empieza a informatizar las admisiones del Instituto. Actualmente el Instituto cuenta con dos edificios de tres plantas, se realizan prácticas ambulatorias, consultas, electro-cardiografía, holter, ergometría y ecocardiografía, estudios hemodinámicos y cirugía cardiovascular.

La informática se convirtió en una herramienta ajena a la clínica y cuyo objetivo único era la administración económica y la facturación de servicios prestados a los pacientes. Los sistemas de información se enfocaron en recoger datos para la gestión y no en incrementar la calidad asistencial, eficiencia y seguridad del paciente.

En la era de los datos, la tecnología ayuda a las organizaciones sanitarias a transformarlos en evidencia y conocimiento clínico [9]. Con ello, se facilita la gestión y visualización de indicadores clave que permitan medir la calidad del servicio de salud. En este entorno sanitario

se trabaja con infinidad de datos y no siempre es fácil tener preparada la información más relevante, por ejemplo, podemos procesar el dato a partir del episodio para acabar explotando un indicador de rendimiento que a su vez alertará sobre la situación actual.

En este camino, gestionar y aumentar el valor de los datos y potenciar las capacidades analíticas ayudará a esta entidad de salud a cumplir con nuevas exigencias del sector. Más aún, en el diagnóstico de enfermedades puede presentarse dificultades, entre los inconvenientes más comunes se encuentra el insuficiente procesamiento de la información relacionada con el paciente, elevando considerablemente la posibilidad de que ocurran errores clínicos. En este escenario, se pretende proveer al médico de información específica y procesada de manera inteligente, en el momento adecuado para apoyar el proceso de toma de decisiones clínicas y así garantizar un mejor proceso de atención y cuidado de los pacientes. Gestionar de manera eficiente la atención, lo que es un desafío dado que las consultas hospitalarias han aumentado y esto es producto de los cambios demográficos, el aumento de la esperanza de vida, el aumento de enfermedades crónicas y el surgimiento de nuevos tratamientos [10].

Uno de los usos más populares de los sistemas de información en el campo de la medicina, es el de dar soporte al diagnóstico de diversas dolencias, tales como la diabetes, la hepatitis o múltiples tipos de cáncer, así como la segmentación de pacientes para una atención más inteligente según su grupo [3].

Estos, en general, son utilizados por el personal sanitario como herramienta de apoyo en la diagnosis o prognosis de un paciente, aportando una medida de la probabilidad de que dicho paciente desarrolle la afección de referencia.

1.6 Planteamiento del problema

1.6.1 Descripción de la situación problemática

La aplicación de este proyecto se centrará en los análisis de bases de datos con información actualizada y verificada sobre pacientes cardiópatas. Se consideran valores de indicadores clínicos que aportarán información clave para la toma de decisiones terapéuticas.

1.6.2 Situación conflicto: nudos críticos

En la actualidad, los científicos han presentado un gran interés por determinar procedimientos o técnicas que posibiliten identificar patrones sintomatológicos y el grado de afectación en pacientes diagnosticados con enfermedades cardiovasculares, y de esta manera anticipar situaciones que pongan en riesgo la vida de los pacientes.

Estudios médicos indican que existen personas asintomáticas, es decir, que sufren enfermedad cardiovascular sin manifestar señales [11].

El contexto de la pandemia del COVID-19, creó barreras de acceso a los servicios de salud. Desde que comenzó la pandemia, los servicios de salud fueron reorganizados o interrumpidos y muchos dejaron de brindar o redujeron su atención y en algunos casos se dio el surgimiento de modalidades nuevas o alternativas para garantizar la atención. Esto estaba en consonancia con las recomendaciones iniciales de la OMS de reducir al mínimo la atención no urgente en centros sanitarios mientras se luchaba contra la pandemia.

Las predicciones son herramientas que permiten aumentar la certeza en el diagnóstico. Muchos pacientes graves permanecen en salas inadecuadas por falta de espacio en unidades de cuidados cardiológicos especiales. Esta última situación es particularmente frecuente cuando el paciente, que ha superado el período más crítico de su enfermedad, permanece en la unidad coronaria (UC), bien sea por falta de camas en la sala de hospitalización convencional o porque su riesgo, aunque no justifica el ingreso en la UC, sobrepasa la capacidad de cuidados de una sala convencional.

1.6.3 Delimitación del problema

A continuación, se detalla en la Tabla 1 la delimitación del problema del presente tema de investigación.

Tabla 1: Delimitación del problema

DELIMITADOR	DESCRIPCIÓN
Campo	Investigación
Área	Ciencias de Datos
Aspecto	Predicción del grado de gravedad del paciente según los tipos de internación.
Tema	Predicción del grado de gravedad del paciente según los tipos de internación utilizando un modelo de árbol de decisión.

1.6.4 Evaluación del Problema

Para la evaluación de nuestro problema consideramos los siguientes aspectos relevantes [12]:

- ✓ Delimitado: El desarrollo del algoritmo, las pruebas, y base científica durante aproximadamente 20 semanas, será de gran ayuda y soporte para los futuros trabajos enfocados en este estudio.

- ✓ Claro: El empleo de una metodología clara y precisa, así como las diferentes herramientas a utilizar, datos que se actualizarán en tiempo real, fáciles de comprender y analizar, permitirán obtener la información necesaria para esta investigación.
- ✓ Evidente: Los datos que se obtendrán para este estudio de investigación se actualizarán constantemente, de modo tal que los resultados que se obtengan sean los más acertados.
- ✓ Original: Mediante el pronóstico, predicción y análisis de los datos obtenidos y mediante el uso de un modelo de árbol de decisión, considerando el número de pacientes propensos a sufrir enfermedades cardíacas. Se propone un modelo con habilidades y conocimientos acerca de un dominio particular, para resolver los problemas de forma similar a la de un experto humano, tecnología que es inédito en nuestra región.
- ✓ Contextual: Alentar al establecimiento de políticas públicas para el desarrollo y uso de nuevas tecnologías que permitan predecir, pronosticar, analizar y clasificar pacientes propensos a sufrir enfermedades cardíacas, con el fin de reducir el colapso de los hospitales, clínicas o centros médicos.
- ✓ Factible: No necesita de una gran cantidad de recursos financieros y puede ser realizado en un tiempo breve.

1.7 Formulación de la hipótesis

En base al estudio e investigación realizada, dudas e hipótesis planteada nos surge la siguiente pregunta con el fin de suplir una necesidad:

¿Cuál es el impacto que tendrá el empleo de un algoritmo de aprendizaje automático para obtener un modelo que identifique la gravedad para la asistencia de los pacientes cardiópatas?

1.7.1 Limitaciones del estudio

Durante el desarrollo del presente proyecto se evidenciaron los posibles inconvenientes podrían poner en riesgo la ejecución del mismo:

- ✓ La base de datos podría no contar con ciertas características necesarias de estándares para analizar el grado de afectación de los pacientes, se requiere de la adopción de un conjunto de estándares que permitan integrar e interoperar datos que proceden de diferentes sistemas.
- ✓ El lenguaje de programación a utilizar es Python, que utiliza librerías para la ejecución de los algoritmos de aprendizaje automático. Ciertas versiones no están actualizadas, teniendo que adaptarlas para su correcto uso.

- ✓ Falta de recursos tecnológicos para la utilización de las diferentes técnicas de aprendizaje automático.

1.8 Escenario

La propuesta se inserta en el proyecto de investigación que corresponde a dos grupos de investigación consolidados, pertenecientes a la Facultad de Ciencias Exactas y Naturales y Agrimensura, de la UNNE, conforman el equipo de trabajo.

Desde hace más de una década, el Grupo De Ingeniería Biomédica desarrolla sistemas computacionales especialmente diseñados para el procesamiento de biodatos. La producción de trabajos publicados en revistas, congresos, capítulos de libro es amplia, todos ellos presentados en ámbitos internacionales y nacionales. Cabe mencionar, también la co-dirección de proyectos de investigación acreditados por la Secretaría General de Ciencia y Técnica de la Universidad Nacional del Nordeste e incluidos en el Programa Nacional de Incentivos; la dirección y co-dirección de tesinas de grado y tesis de posgrado en el área y la formación de recursos, alumnos de grado y becarios de investigación.

El plan de tesis aquí presentado se enmarca en el proyecto vigente identificado como PI 18F004 "Procesamiento Digital de Biopotenciales", dirigido por el Dr. Jorge Emilio Monzón y co-dirigido por la Dra. María Inés Pisarello.

El equipo GICS-UNNE se centra en la investigación y aplicación de estándares, métodos y herramientas para contribuir a la calidad del software, tanto en el proceso como en el producto. En este marco, se ha realizado el estudio particular de procesos de explotación de información, y generado un sistema de apoyo a la decisión en el ámbito académico, como aporte a la analítica académica.

Se cuenta con publicaciones en revistas y congresos, existe una intensa actividad de formación de recursos humanos, mediante la dirección de tesis de Doctorado/Maestría/Especialidades y trabajos de fin de carrera. El co-director de la tesis, Dr. Emanuel Irrazabal pertenece a este grupo mencionado.

De los grupos mencionados anteriormente el plan de tesis propuesto es parte sustancial de este convenio.

1.9 Producción Científica

E. Báez, S. Vallejos, F. Bobadilla, M. Pisarello, "Sistemas de apoyo a la toma de decisiones clínicas (SADC) en una Unidad de Cuidados Intensivos Coronarios", V Congreso Argentino de Ingeniería, XI Congreso Argentino de la Enseñanza de la Ingeniería, 2021.

1.9.1 Actividades de Divulgación Científica

E. Báez, S. Vallejos, F. Bobadilla, G. Dapozo, E. Irrazabal, J. Monzón M. Pisarello, "Sistema de Soporte a las Decisiones Clínicas en UCI", 6° Edición, Jornadas de Articulación para la Innovación, 2020.

1.9.2 Expositor en panel de Ciencia de Datos

E. Báez, S. Vallejos, M. Pisarello, "Sistema de Soporte a las Decisiones Clínicas (SSDC)", Datos: Nuevas oportunidades de desarrollo. Ciencia-Empresa, UNRaf, 2021.

Capítulo II

Marco teórico

2 Marco teórico

En esta sección se presentan los fundamentos teóricos del trabajo, así como la información relevante a nivel médico y a nivel informático de los factores que intervienen en la medicina e informática.

2.1 La eSalud

Las integraciones de todos los sistemas digitales ofrecen innumerables ventajas a todos los actores implicados. El cuidado de la salud es un proceso continúo basado en la información y la comunicación, y necesita de la constante interacción e intercambio de información por parte de los diferentes actores intervinientes, como ser pacientes, prestadores, profesionales sanitarios, fisioterapeuta, aseguradores y el estado. Eysenbach define la eSalud como el campo que resulta de la intersección entre la informática médica, la salud pública y el conocimiento

sobre los servicios de salud, los cuales mejora a través de Internet y las tecnologías relacionadas [13].

Es clave en el entorno clínico actual el soporte a las actividades médicas por medio de sistemas de información hospitalaria bien diseñados [14], compuestos por aplicaciones que deben estar construidas con base a estándares.

2.1.1 El intercambio electrónico de información clínica

La gran mayoría de los sistemas de Historias Clínicas Electrónicas(HCE) disponibles, se encuentran orientados a problemas. Dentro de las funcionalidades deseables de una HCE se encuentra la posibilidad de gestionar la lista de problemas. Independientemente del tipo de interfaz utilizada, toda la información se almacena en un único lugar que se denomina Repositorio de Información Clínica (Clinical Data Repository –CDR–) [15]. En este CDR también se almacenan datos clínicos procedentes de otras fuentes de información distintas de la HCE como son los distintos efectores de estudios complementarios (laboratorio, radiología, ecografía, etc.)

2.2 Enfermedades cardiovasculares

El término EECCVV es un concepto genérico que se emplea para referirse a un conjunto de patologías y enfermedades diversas en sus causas o etiología y en sus manifestaciones clínicas como signos y síntomas [16]. Según la versión X de la Clasificación Internacional de Enfermedades de la OMS (CIE-X) los grandes grupos de las enfermedades del aparato circulatorio son:

- ✓ Fiebre reumática aguda.
- ✓ Cardiopatías reumáticas crónicas.
- ✓ Enfermedades hipertensivas incluyendo la eclampsia (hipertensión durante el embarazo).
- ✓ Cardiopatía isquémica (infarto de miocardio, angina de pecho).
- ✓ Enfermedad cardiopulmonar.
- ✓ Otras enfermedades del corazón (p.e. arritmias e insuficiencia cardíaca entre otras).
- ✓ Enfermedades cerebrovasculares (p.e. hemorragia, derrame, embolia, trombosis, apoplejía cerebral o ictus).
- ✓ Enfermedades de las arterias (p.e. aterosclerosis, aneurisma, embolia y trombosis arteriales entre otras).
- ✓ Enfermedades de las venas (p.e. tromboflebitis)

- ✓ Malformaciones congénitas del sistema circulatorio.
- ✓ Muerte súbita.

2.2.1 Unidad de cuidados intensivos

La medicina crítica se ha especializado en el cuidado y manejo de pacientes en estado crítico o en riesgo de desarrollarlo, atención que no puede proveerse en las salas regulares del hospital [17]. Los cuidados críticos corresponden a la etapa contemporánea e incluyen el monitoreo multiparámetro automatizado para el manejo de pacientes con deterioro multiorgánico, exámenes complementarios, dispositivos para el sostén básico y avanzado a la cabecera del enfermo y un equipo clínico multidisciplinario. La medicina crítica como disciplina se refiere a la ciencia del monitoreo y manejo del paciente crítico.

El análisis del desempeño de la red de atención médica muestra que muchos pacientes críticamente enfermos son evaluados y atendidos primero en unidades fuera de la unidad de cuidados intensivos, desde los servicios prehospitalarios, en sala de emergencia, emergency room (ER) y el mundo de la rehabilitación.

Según la revista de la Sociedad Española de Cardiología [18], las llamadas unidades de cuidados intermedios cardiológicos o unidades coronarias de cuidados intermedios, presentan los siguientes objetivos:

- ✓ proporcionar a cada paciente el grado de cuidados que requiere, ni excesivos ni insuficientes;
- ✓ optimizar los recursos estructurales, técnicos y humanos de forma que se eviten ingresos innecesarios en la UC y se faciliten los traslados desde ésta, lo que mejora la gestión de las camas, y
- ✓ facilitar la continuidad de cuidados y asistencial.

La HC es un instrumento básico para plasmar la atención médica del paciente y el desarrollo de las nuevas TIC, ha desencadenado la aparición de nuevas disciplinas y avances en las ciencias médicas como la telemedicina, la informática médica, entre otras. En la era de la información, estamos hablando de registros electrónicos de alcance total en la atención médica a la población. El término e-Salud o cibermedicina, se refiere a la aplicación de las tecnologías de la información y las comunicaciones para la atención de salud, la vigilancia y la documentación sanitaria así como la educación los conocimientos y las investigaciones en materia de salud [19].

2.3 Ciencia de Datos

La ciencia de datos es un ecosistema artificial emergente que configura una nueva era de la información. Se puede denominar como una disciplina híbrida que involucra matemáticas, estadística e informática.

Específicamente, desde el dominio de la informática, implica técnicas de aprendizaje automático (AA), minería de datos y visualización o representación [20].

2.3.1 Modelo de Procesos de explotación de información

A continuación, se describen los modelos de procesos relevantes en el área. El primer modelo desarrollado es el modelo de proceso KDD, Descubrimiento de Conocimiento en Bases de Datos, (del inglés Knowledge Discovery in DataBase), y el modelo de proceso CRISP-DM, (del inglés CrossIndustry Standard Process for Data Mining).

2.3.1.1 KDD

Es el proceso de extraer conocimiento útil a partir de datos. Este proceso puede hacerse manualmente, los expertos en algún dominio pueden consultar y analizar bases de datos para descubrir patrones que les ayuden a tomar decisiones.

Dado el gran volumen de datos acumulados en las HCE, y la incapacidad de los especialistas de identificar patrones de comportamiento y extraer conocimiento oculto en los datos almacenados para apoyar sus decisiones, surge la necesidad de aplicar la minería de datos. Esta tiene la capacidad suficiente para intervenir y optimizar algunas de las dimensiones clave en la gestión sanitaria como por ejemplo el diagnóstico y tratamiento, ayudando a los médicos a identificar los tratamientos más eficaces y definir mejores prácticas exportables o la gestión de recursos sanitarios. Es el descubrimiento del conocimiento KDD, el que se encarga de la preparación de los datos y la interpretación de los resultados obtenidos, los cuales dan un significado a estos patrones encontrados [21].

2.3.1.2 CRISP-DM

CRISP-DM (CRoss-Industry Standard Process for Data Mining) es la metodología para proyectos dedicados a extraer valor de los datos [22]. Se conceptualiza en 6 fases que van desde la fase de comprensión del problema hasta la puesta en producción de sistemas automatizados analíticos, predictivos y/o prospectivos. Estas fases dependen entre sí tanto en forma secuencial

como cíclica, pudiendo encontrarse interacciones que permitan mejorar la aproximación obtenida en otras fases anteriores. En la Fig. 1 se ilustra las fases del modelo.

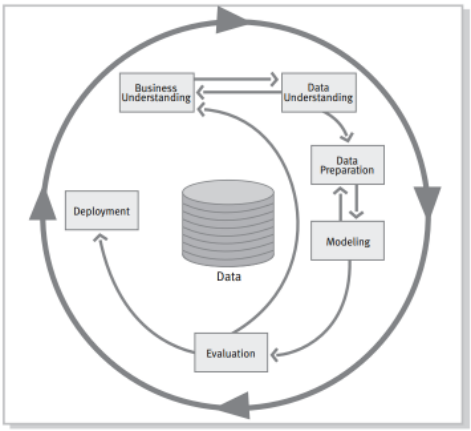


Fig. 1: Fases del modelo de referencia CRISP-DM. Fuente [22].

En la investigación clínica, los datos de pacientes pueden ser de la gestión económico-financiera y logística del sistema de salud. La integración o relación de estos sistemas con los sistemas de información clínica, plantea la posibilidad de explotar grandes volúmenes de datos. El primer paso para la preparación y análisis de datos será identificar los datos necesarios y sus repositorios. Como resultado de esta etapa se obtendrá una visión global del sistema documental que lleva a cabo la institución y es desde esta visión que se puede generar un plan de acción que permita cumplir con los requisitos necesarios para dar cuenta de un adecuado análisis inicial. No se refiere sólo a la noción de identificar todas las posibles fuentes de datos, sino a identificar todas las fuentes de datos aplicables.

2.3.2 Minería de datos

Es la aplicación de técnicas de procesamiento de información, para encontrar patrones útiles que se espera ayuden a las personas a acceder a la base de datos más eficientemente [23]. Utiliza métodos y estrategias de otras áreas o ciencias, entre las cuales se encuentra el aprendizaje automático. Trata de la aplicación de un conjunto de técnicas y tecnologías, el objetivo es obtener conocimiento de los datos. Este conocimiento puede tener un valor científico que ayude a determinar causas de patologías o a identificar poblaciones de riesgo, y así ayudar en la detección precoz de enfermedades.

Es también una herramienta útil para la toma de decisiones, la optimización de recursos y la detección de prácticas fraudulentas.

2.3.2.1 Minería de datos predictiva

Se encarga de establecer patrones de comportamiento, a partir del análisis de la información contenida en un conjunto de datos de una variable de interés, que le permite descubrir el comportamiento o predecir tendencias sobre cualquier evento desconocido.

Esta técnica permite predecir el valor de un atributo a partir de otros ya conocidos; identifica relaciones entre variables de un evento anterior las cuales sirven para predecir el comportamiento de dichas variables en los resultados de situaciones futuras [24].

El análisis predictivo en la atención médica ayuda a detectar los primeros signos de deterioro del paciente en la UCI y en el servicio de hospitalización general e identificar a los pacientes en riesgo [25].

En el ámbito médico la aplicación de la minería de datos tiene interés en varios campos [26]:

1. En el ámbito clínico resulta de ayuda para la identificación y diagnóstico de patologías. Así mismo tiene importancia para el descubrimiento de posibles interrelaciones entre diversas enfermedades.
2. Al nivel de medicina preventiva, resulta de interés para la detección de pacientes con factores de riesgo para sufrir una patología.

2.3.3 Inteligencia artificial

La IA tiene como objetivo que las computadoras realicen tareas que puedan ser hechas por la mente. Como lo son el raciocinio, percepción, visión, asociación, predicciones y planificación que hacen que seres humanos y animales tomen decisiones [27].

Mediante equipos informáticos en general, se trata de realizar tareas que sean relacionadas con inteligencia que comúnmente son realizadas por humanos. Es complicado encontrar una serie de características que abarque conceptualmente todas las áreas en la que se practica.

2.3.3.1 Aportes a la medicina

Nos encontramos con información médica en grandes cantidades, procedente de bases de datos científicas, sitios web e informes de las compañías y de los organismos especializados en salud. Están conformadas como bases de datos que pueden estar o no estructuradas, si los datos son válidos y bien interpretados pueden brindar grandes beneficios al optimizar y disminuir los costos y los tiempos de servicio en el área de la salud, pero también pueden ser utilizados para

realizar predicciones sobre enfermedades, para mejorar los tratamientos, capacitar a los médicos en lugares de difícil acceso y mejorar la calidad de vida.

En el futuro, el vínculo entre hombre-máquina en el campo de la medicina será más próximo; de esta manera las máquinas tendrán la labor de extracción, barrido y búsquedas de correlaciones, la tarea del médico sería solo de interpretar estas correlaciones y hallar nuevos tratamientos para mejorar su efectividad [28].

2.3.4 Algoritmos basados en aprendizaje supervisado

2.3.4.1 Regresión lineal

Para elaborar un modelo de regresión, debemos tener en cuenta el tipo de variables que vamos a introducir estas pueden ser categóricas o continuas. Los paquetes estadísticos son:

2.3.4.2 Regresión logística

Es una herramienta estadística de análisis heterogéneo, de utilización tanto aclaratoria como de predicción, permite estimar la probabilidad de una variable cualitativa binaria en función de una variable cuantitativa. Es un modelo que genera una probabilidad para cada valor de etiqueta discreto posible, en problemas de clasificación, al aplicar una función sigmoide a una predicción lineal [29].

La función sigmoide o función logística tiene forma de "S" o curva sigmoidea, que se define mediante la que se define mediante la ecuación (1):

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^{x}}{e^{x} + 1} \quad (1)$$

Dónde *S(x)* será la salida, teniendo un valor comprendido entre cero y uno; x serán los datos de entrada a la función.

Esta técnica puede considerarse una extensión de los modelos de regresión lineal, con la particularidad de que el dominio de salida está acotado al intervalo [0,1] y que el procedimiento de estimación, en lugar de mínimos cuadrados, utiliza el procedimiento de estimación de máxima verosimitud.

Se trata de un algoritmo de tipo supervisado utilizado para clasificación.

Así es como va a determinar el peso de cada variable dependiente, basadas en las similitudes que presentan las personas que les ocurran estos sucesos[30].

En RL lo que se pretende es estimar los parámetros de la ecuación (*β0, β1, β2,...βk*) de la función que pretendemos evaluar, está representada en la siguiente ecuación (2):

$$Z = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta kXk \quad (2)$$

β0 es la ordenada en el origen de la función de regresión, ***β1, β2,...βk*** representan los coeficientes de la pendiente dela recta y ***X1,X2,...Xk*** son las variables independientes o factores de riesgo.

Bayes ingenuo guassiano: El clasificador Bayesiano es conocido por tomar similitudes de cada elemento y sospechan que todas son independientes y no alteran la clasificación [31].

2.3.5 Aprendizaje automático

El aprendizaje automático o machine learning (ML) es un tipo de IA que proporciona a las computadoras la capacidad de aprender, sin ser programadas explícitamente. Es el campo dedicado al desarrollo de métodos computacionales para los procesos de aprendizaje [32] y [33]. Se centra en el desarrollo de programas informáticos que pueden cambiar cuando se exponen a nuevos datos. Sus componentes son:

El problema: es esencial en situaciones donde se necesite identificar patrones y realizar predicciones de comportamientos apoyados en datos reales. Identificar semejanzas, reconocer caracteres y hacer sugerencias.

La data: Los modelos de ML aprenden de los datos y pueden ajustarse a sí mismos para producir mejores resultados. Las principales causas por las que un conjunto de datos puede presentar problemas son [34]:

El ruido: Los datos introducidos presentan errores o valores muy atípicos. Ser incompletos**,** los datos contienen registros con características incompletas. Por lo tanto, la preparación de los datos resulta ser una tarea imprescindible previa a la realización del modelo. Existen varias formas de preparar los datos de una forma óptima:

Detección de errores y limpieza de los datos: La detección de outlier o valores atípicos, es una observación que es numéricamente distante del resto de los datos. Algunas de las técnicas más comunes para obtener estas métricas es mediante los histogramas o gráfico de caja conocidos como box-plot. A partir de la detección realizada, se procede a la limpieza del conjunto de datos eliminando los valores atípicos, limpiando los datos con ruido y rellenando los valores faltantes.

Datos faltantes: Por un lado, debemos analizar si es un registro que nos va a aportar la información suficiente o si simplemente se pudiera ignorar/eliminar. Por otro lado, existen formas de rellenar dichos valores obteniendo la media, la moda o el valor previsto que debería de existir en esa característica.

Transformación de los datos y selección de las características: En muchas ocasiones puede darse el caso donde no todas las características del conjunto de datos nos den una información relevante. Es por ello, por lo que puede existir la posibilidad de descartar alguna de estas características para tener un control de los datos más sencillo. Este paso se puede realizar manualmente para conjuntos de datos pequeños, pero requiere automatización para la mayoría de conjuntos de datos de tamaño realista. Hay herramientas de software disponibles para esto, como los conversores automáticos de ficheros PDF a datos estructurados, por ejemplo, Adobe Acrobat (Copyright © 2021 Adobe), éste fue el desarrollador original del formato PDF, el mismo incluye la función de conversión.

La calidad de la investigación clínica mejorará con la captura electrónica de datos, aportando una mejor calidad en la recolección de los datos. La dificultad reside en una mayor complejidad de preparación, lo que fuerza la adaptación de los profesionales involucrados en el ensayo clínico a un sistema de trabajo distinto.

Hay una relación inversa entre la calidad de los datos y el número de errores. La calidad es inversamente proporcional al número de errores y omisiones, ya que podríamos tener un BD libre de errores, pero con tal cantidad de omisiones que comprometa los resultados del estudio. La conexión de registros de 2 o más bases de datos que tienen información complementaria de un tema o contenido, permite identificar y conectar los registros que corresponden a un mismo individuo. Puede hacerse de forma manual, considerando la posibilidad de que la conexión sea correcta o incorrecta [34].

En el aprendizaje automático, los sistemas se entrenan para utilizar algoritmos especializados para estudiar, aprender y hacer predicciones y recomendaciones a partir de enormes cantidades de datos. Esta parte es clave porque es donde se realiza el entrenamiento de los algoritmos.

Los algoritmos que más se utilizan en los problemas de ML son los siguientes:

1. Regresión Lineal.
2. Regresión Logística.
3. Árboles de Decisión.

4. Random Forest.
5. SVM o Máquinas de vectores de soporte.
6. KNN o K vecinos más cercanos.
7. K-means.

El ML opera como un médico residente: aprende las reglas a partir de los datos. Esto empieza por las observaciones a nivel individual, a nivel del paciente y luego se mueve a través de una enorme cantidad de variables, a la búsqueda de combinaciones que predigan resultados o outcomes en forma confiable. La gran ventaja del ML es que puede manejar una gigantesca cantidad de predictores y combinarlos en forma no lineal e interactiva. Esto permite usar nuevos tipos de datos, cuyo volumen o complejidad antes eran imposibles de analizar.

El caso de uso del aprendizaje automático es ilimitado, ya que puede aplicarse a cualquier tipo de industria, desde el sector de producción primario, como la agricultura, pasando por el sector de servicios e incluso por actividades relacionadas con la sanidad.

2.3.5.1 Uso del aprendizaje automático en la medicina

El área médica podría beneficiarse de una relación cercana a la informática, de esta forma se mejoran los procesos que son complejos y que tienen errores como la evaluación diferencial. Esto es realizado por el aprendizaje automático, arrancan de un bloque de datos de entrenamiento y mejorar las técnicas de clasificación y predicciones [27].

En algunos países, en los últimos años ha evolucionado la implementación de registros electrónicos, los institutos nacionales de la salud tienen datos clínicos muy importantes almacenados. Para que toda esta información se transforme en conocimiento, necesitan ser ejecutados y analizados mediante técnicas estadísticas complejas, como se realizan en diferentes países utilizando: razonamientos basados en sucesos, redes neuronales, clasificadores bayesianos, regresión o máquinas de soporte vectorial; esto permite que sea más fácil el diagnóstico de enfermedades como: apendicitis, cáncer de mama, hepatopatía crónica.

El aprendizaje profundo está enfocado a un grupo de modelos que le brinda a una computadora que seleccione automáticamente características de niveles superiores indispensables para poder clasificar partiendo de los datos en su estado original usando varios niveles de representación. El modelo va a intentar copiar los niveles de actividad de las neuronas, las que se acumulan el 80% en el cerebro y donde se producen los pensamientos. El programa aprende, podrá identificar patrones en las representaciones digitales gráficos y más datos [27].

2.3.6 Usabilidad de los datos

El uso de datos de buena calidad ayuda a los esfuerzos para lograr un desarrollo que sea sostenible. Se deben establecer estándar para la forma en que se reporta la información para que sea más fácil planificar, rastrear y comparar el progreso y los resultados de diferentes proyectos y actividades. Acceder a los mejores datos disponibles y utilizarlos es crucial para fundamentar decisiones efectivas que promuevan el desarrollo y brinden un apoyo esencial. Cabe destacar la necesidad de obtener datos de entrenamiento etiquetados. Para ello, los datos de entrada deberán estar formados por las distintas características que lo forman y su etiqueta, es decir, a qué clase pertenece. Importa mucho la calidad y cantidad de información que se consiga ya que impactará directamente en lo bien o mal que luego funcione el modelo. Un problema importante en la minería de datos es la clasificación de las reglas de aprendizaje, las cuales tratan de encontrar más reglas que van dividiendo los datos en las clases predefinidas. Cuando tenemos un solo conjunto de datos, dividimos el conjunto grande en dos más pequeños: uno para entrenamiento y otro para prueba.

El porcentaje de los datos de cada una de las divisiones supone un conflicto ya que, si el conjunto de datos de entrenamiento es muy grande, la capacidad de aprendizaje será mucho mayor, pero puede existir un sobre entrenamiento del modelo, lo que causa que el algoritmo de aprendizaje puede quedar ajustado a unas características muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo [24]. La información que fue seleccionada para el entrenamiento es utilizada como datos de entrada y de salida del algoritmo. El modelo aprenderá de la información de entrenamiento.

2.3.7 Aprendizaje supervisado y No supervisado

Da inicio desde un grupo de datos que se encuentra etiquetado, esto nos indica que se encuentra conformado por información que contiene la variable que nos interesa predecir. Con el uso de algoritmos de ML se produce un modelo de predicción el cual es entrenado con este grupo de datos. Y cuando este ya haya aprendido se procede a realizar una medición de rendimiento del modelo con el que se está trabajando.

En los problemas del mundo real, la mayoría de las veces, los datos no vienen con etiquetas predefinidas, así que se deben desarrollar modelos de aprendizaje automático que puedan clasificar correctamente estos datos, encontrando por sí mismos algunos puntos en común en las características, que se utilizarán para predecir las clases sobre nuevos datos.

2.3.8 Regresión lineal y regresión múltiple

La regresión simple es el cálculo de la ecuación correspondiente a la línea que mejor describe la relación entre la respuesta y la variable que la explica [27]. La ecuación representa la línea que mejor se ajusta a los puntos en un gráfico de dispersión. Suma las distancias al cuadrado entre los puntos reales y los puntos definidos por la recta estimada.

La clasificación automática de objetos o datos es uno de los objetivos del ML. Podemos considerar tres tipos de algoritmos:

- ✓ Clasificación supervisada: posee datos de entrenamiento y cada dato está asociado a una etiqueta. Construimos un modelo en la fase de entrenamiento o training; utilizando dichas etiquetas, que nos dicen si un objeto está clasificado correcto o incorrectamente por el modelo. Construido el modelo podemos utilizarlo para clasificar nuevos datos, ya no necesitan etiqueta para su clasificación, aunque sí la necesitan para evaluar el porcentaje de objetos bien clasificados.
- ✓ Clasificación no supervisada: los datos no tienen etiquetas o no queremos utilizarlas; y estos se clasifican a partir de su estructura interna como ser propiedades, características.
- ✓ Clasificación semisupervisada: algunos datos de entrenamiento tienen etiquetas. Este último caso es muy típico en clasificación de imágenes. Estos se pueden considerar algoritmos supervisados que no necesitan todas las etiquetas de los datos de entrenamiento.

En la Tabla 2 se sintetizan sus diferencias, las características de cada uno y para qué se utilizan.

Tabla 2: Diferencias, características y uso del aprendizaje supervisado y no supervisado.

Parámetros	**Aprendizaje automático Supervisado**	**Aprendizaje automático No supervisado**
Los datos de entrada	Los algoritmos se entrenan utilizando datos etiquetados	Los algoritmos se utilizan contra datos que no están etiquetados
Complejidad Computacional	Método más simple	Computacionalmente complejo.
Exactitud	Alta precisión	Menos precisa

2.3.9 Árboles de Clasificación

En coincidencia con Britos, Hosin [33] en lo que se refiere al análisis con árboles, existen dos enfoques principales: los árboles de decisión y los árboles de regresión. Ambos constituyen métodos predictivos de segmentación, conocidos como árboles de clasificación.

A través de diferentes procedimientos estadísticos se determina la división más discriminante los criterios seleccionados, aquella que permite diferenciar mejor a los distintos grupos del criterio base, con lo que se obtiene así, una primera segmentación [29]. A partir de ella, se realizan nuevas segmentaciones de cada uno de los segmentos resultantes y así sucesivamente hasta que el proceso finaliza con alguna norma estadística. El resultado es un conjunto de reglas que pueden visualizarse fácilmente mediante la estructura o gráfico de un árbol. Los métodos estadísticos y de ML basados en árboles, engloban a un conjunto de técnicas supervisadas no paramétricas que consiguen segmentar el espacio de los predictores en regiones simples, dentro de las cuales es más sencillo manejar las interacciones. Son útiles en la exploración de datos, permiten identificar de forma rápida y eficiente las variables predictoras más importantes. Son capaces de seleccionar predictores de forma automática.

El procedimiento de clasificación basado en árboles, clasifica casos en grupos o pronostica valores de una variable dependiente basada en los valores de las variables independientes [33].

Segmentación. Identifica individuos que pueden ser miembros de un grupo específico.

Estratificación. Asigna los casos a una categoría de entre varias, por ejemplo, pacientes de alto riesgo, bajo riesgo o riesgo intermedio.

Predicción. Crea reglas y las utiliza para predecir eventos futuros.

Los tres tipos de árboles más utilizados son [35]: árboles CHAID, árboles CART y árboles QUEST

- ✓ Árboles CHAID (Chi-square Automatic Interaction Detector). Detector automático de interacción chi-cuadrado.
- ✓ Árboles CART (Classification and Regression Tree). Árbol de clasificación y regresión. Es una alternativa al CHAID exhaustivo para árboles de clasificación con variables dependientes categóricas.
- ✓ Árboles QUEST (Quick, Unbiased, Efficient, Statistica Tree). Árbol estadístico rápido, imparcial, eficiente. Consiste en un algoritmo de clasificación arborescente creado especialmente para solventar dos de los principales problemas que presentan los métodos CART y CHAID exhaustivo a la hora de dividir un grupo de sujetos en función de una variable independiente.

2.3.9.1 Árbol de decisión

Los árboles de decisión (AD) son modelos predictivos formados por reglas binarias del tipo si/no, con las que se consigue repartir las observaciones en función de sus atributos y predecir así el valor de la variable respuesta.

El AD se compone de ramas y nodos. Las ramas, indican los posibles caminos generados automáticamente de acuerdo a la decisión tomada. El nodo interno contiene una evaluación en referencia a algún valor de una de las propiedades. El nodo de probabilidad indica que debe ocurrir un evento aleatorio de acuerdo a la naturaleza del problema. El nodo hoja representa el valor que devolverá el árbol de decisión.

2.3.9.1.1 Métricas en el árbol de decisiones

Impureza y entropía de Gini: Sólo funciona con objetivos categóricos, ya que solo hace divisiones binarias. La impureza de Gini se calcula utilizando la siguiente ecuación (3):

$$\text{Gini} = 1 - \sum_{i=1}^{c} (pi)^2 \quad (3)$$

Cuanto menor sea la impureza de Gini, mayor es la homogeneidad del nodo. La impureza de Gini de un nodo puro, de la misma clase, es cero.

Ganancia de información: Propiedad estadística que mide qué tan bien un atributo dado separa los ejemplos de entrenamiento de acuerdo con su clasificación objetivo.

Reducción de varianza: Es un algoritmo usado para variables objetivo continuas. Este algoritmo usa la ecuación 4, es la fórmula estándar de la varianza para escoger el criterio de división. La división con la varianza más baja se escoge para dividir la población.

$$Varianza = \frac{\sum(x - \bar{x})^2}{n} \quad (4)$$

ID3 (dicotomizador iterativo 3): Genera árboles más pequeños y no es útil en datos continuos.
C4.5: Es una versión avanzada de ID3 que también funciona con datos continuos basados en un valor de umbral. Después de la creación de árboles, se puede podar.
CART: Es un árbol de clasificación y regresión que genera árboles en función de si la variable de salida es categórica o numérica. El CART detecta que no es posible obtener más ganancias en el atributo y deja de dividirse.

2.4 La visualización de datos

Es una representación gráfica de información y datos. Al usar elementos visuales como tablas, gráficos y mapas, las herramientas de visualización de datos brindan una forma accesible de ver y comprender tendencias, valores atípicos y patrones en los datos. La visualización de datos utiliza diversos elementos gráficos, dispone de valores, de gráficos, de mapas, con la misión de facilitar la comprensión y explorar. El cometido es que descubramos la historia, no que nos las den presentada, sino que se pueda ir a buscar una información determinada, interaccionar, filtrar, interactuar con esos datos y construir un relato propio. En la visualización hay un componente dinámico**,** por lo cual las cosas cambian, se puede explorar esos datos, hay exploración, búsqueda, selección, filtro, cuando cambia un valor, cambia la gráfica, y evidentemente es más contextual, sin conocer el contexto no se podrá comprender qué filtrar y qué se quiere buscar [36].

Los tableros dinámicos son realizados con datos, con escalas, usando líneas, barras y con forma de color y tamaños. Las escalas, pueden ser nominales, ordinales y de intervalo. Las líneas se utilizan para representar conexiones o serie de puntos. Las líneas ayudan a la audiencia a comprender la tendencia, por ejemplo. Las barras tienen un impacto visual en los pesos de algunos fenómenos, dividiendo esos fenómenos en grupos y dando diferentes percepciones sobre medidas cuantitativas o datos cuantitativos [37]. El uso de formas y colores también ayuda a la audiencia a interpretar valores cualitativos, mejor que datos cuantitativos. El mal uso o uso excesivo de ciertos colores puede tener un efecto contrario, podrían inducir a error en la interpretación. Otro problema común es resaltar la información importante en lugar de centrar la atención en el objetivo de la realidad medida [36].

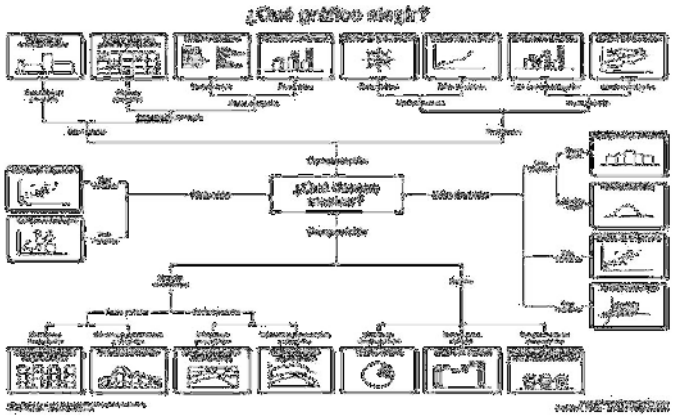


Fig. 2: ¿Qué gráfico elegir? Fuente:[38].

¿Qué método de visualización es más efectivo? Parte de la respuesta la tenemos en la Fig. 2 tomada de: The Extreme Presentation(tm) Method [38].
Los métodos posibles de visualización se pueden agrupar en cuatro categorías:

Distribuciones: Se utiliza en el comienzo de la etapa de exploración de datos, cuando queremos comprender las variables. Aquí podemos encontrar variables de dos tipos cuantitativas y categóricas. Dependiendo del tipo y cantidad de variables, el método de visualización que vamos a utilizar.

Comparaciones: En esta categoría el objetivo es comparar valores a través de diferentes categorías y con el tiempo (tendencia). Los gráficos más comunes son los diagramas de barras y los diagramas de puntos y líneas.

Relaciones: El objetivo es comprender la relación entre dos o más variables. La visualización más utilizada en esta categoría es el gráfico de dispersión.

Composiciones: El objetivo es comprender cómo está compuesta o distribuida una variable; ya sea a través del tiempo o en forma estática. Los más comunes aquí son los diagramas de barras y los gráficos de tortas.

De una forma muy genérica, en los gráficos se representa la evolución (variación en el tiempo) y la distribución (segmentación del dato). La creatividad puede jugar un papel clave en una buena visualización de datos [37].

2.4.1 Diseño de la interfaz

Una interfaz define cómo recibimos la información. El medio con que el usuario interactúa con la plataforma debe incluir elementos que lo guíen a realizar las acciones que se plantearon como parte de los objetivos. Lo que se pretende de las interfaces de IA es traer información compleja de una forma simple.

La jerarquía de los contenidos y la relación entre ellos también deben ser expresadas a través del diseño gráfico. La IA busca optimizar la interfaz para personalizarla y mostrar cada vez información más relevante, simple y de forma prolija [37].

La presentación de una interfaz se considera desde la producción de la interfaz y el consumo o interpretación de lo que presenta. Para esto debemos diseñarlas de manera inteligente, efectiva y creíble. Un factor a tener en cuenta es cómo aprenden los consumidores a entender y juzgar la calidad de una presentación.

2.4.2 Modelo de desarrollo del prototipo

El Modelo de prototipos pertenece a los modelos de desarrollo evolutivo [39]. Debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar muchos recursos. Las características fundamentales de un prototipo son:

- ✓ El Tiempo: El prototipo se desarrolla en menos tiempo para poder ser probado o testeado.
- ✓ El Coste: La inversión en un modelo de prototipo es ajustada, lo que requiere un uso óptimo de los recursos.
- ✓ Conciso: Debe incluir los requisitos y características básicas de la aplicación para poder evaluar su funcionamiento y utilidad.
- ✓ Evolutivo: El prototipo evoluciona gracias a la interacción con los usuarios
- ✓ Funcional: Es una aplicación que funciona.

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. El prototipo se prueba y modifica cuando es necesario, y los resultados se anotan en la revisión de los bosquejos y los dibujos en funcionamiento [39].

Los tipos de prototipo son:

Desechables: Sirve para eliminar dudas sobre lo que quiere el cliente.

Evolucionario: Modelo parcialmente construido que pasa de ser prototipo a ser el software.

En este proyecto se propone utilizar el modelo evolucionario, ya que se trabaja con algo tangible a partir del cual se puede definir un punto de partida o desechar completamente una idea.

2.4.3 Python

Se trata de un software libre, de altas prestaciones y multifuncional demostrando un impacto tecnológico alto. El lenguaje de programación Python cumple con buenas expectativas y es tomado en cuenta para el desarrollo de softwares.

Esto quiere decir que puede ser utilizado para cualquier evento que se presente, podrá ser ejecutado en cualquier entorno, es adaptable y podrá ser modificado las veces que sean necesarias si así lo necesita.

“Python ha ido ganando en adeptos en comunidades como la de software libre, científica y educacional, por su sencillez y posibilidad de concentrarse en los problemas actuales” Holguín [40].

Es compatible con plataformas y sistemas operativos distintos, como Linux, Windows y Mac. Pero también, son utilizados en dispositivos inteligentes. Es una herramienta multipropósito y puede ser utilizada para aplicaciones científicas, telecomunicaciones, interfaces gráficas, juegos para equipos móviles y otros.

Python es una herramienta que está en constante cambio y es cada vez mejor porque los desarrolladores hacen el lanzamiento de una nueva versión cada 6 meses. Estas a su vez mejoran la compatibilidad de los programas de versiones anteriores [41].

Para realizar el presente trabajo investigativo usamos Python como lenguaje de programación, por la facilidad de comprensión y por la variedad de información que podemos encontrar para el desarrollo eficiente del código.

2.4.3.1 Librerías utilizadas para lenguaje Python

1.Sklearn: Está conformada por un conjunto de algoritmos de ML de alto nivel que solucionan problemas sean estos supervisados o no supervisados de escala intermedia. Es fácil de ser utilizado, por la productividad de la documentación, proporciona una interfaz sencilla y la relación con otras Apis. Tiene mínimas dependencias y es repartida bajo la licencia BSD[1] resumida [42].

El paquete de scikit-learn cuenta con la biblioteca de ML que es comúnmente utilizado. Utiliza la interacción y modalidades que tiene Python para brindar los prototipos de manera veloz y sencilla. Hay diferentes paquetes de aprendizaje automático y cada uno de ellos tiene su fortaleza en distintos modelos. Los elementos de esta librería comparten un conjunto uniforme de métodos que depende de su propósito:

Los estimadores pueden ajustar modelos a partir de datos, los predictores pueden hacer predicciones sobre nuevos datos y los transformadores convierten datos de una representación a otra [43].

La clase DecisionTreeRegressor del módulo sklearn.tree permite entrenar árboles de decisión para problemas de regresión. El árbol de decisión viene en el algoritmo CART (árbol de

[1] Berkeley Software Distribution

clasificación y regresión) y además posee varios hiperparámetros. Los más importantes son aquellos que detienen el crecimiento del árbol, conocidas como condiciones de stop:

max_depth: Profundidad máxima que puede alcanzar el árbol.

min_samples_split: Número mínimo de observaciones que debe de tener un nodo para que pueda dividirse.

min_samples_leaf: Número mínimo de observaciones que debe de tener cada uno de los nodos hijos para que se produzca la división.

max_leaf_nodes: Número máximo de nodos terminales.

random_state: Semilla para que los resultados sean reproducibles. Tiene que ser un valor entero.

2. Numpy: Numpy es una biblioteca del lenguaje Python para computación científica, da soporte para crear vectores y matrices grandes, multidimensionales, proporcionan el uso competente de cálculos numéricos de un lenguaje de programación de nivel elevado.

Esta librería brinda dos elementos que son esenciales: un elemento de matriz N dimensional y un elemento de función universal además de útiles capacidades de álgebra lineal, transformación de Fourier y números aleatorios [44].

3. **Pandas:** Es un paquete de Python que proporciona estructuras de datos similares a los dataframes de R, el cual es un entorno de software libre y lenguaje de programación con un enfoque al análisis estadístico. Pandas depende de Numpy, la librería que añade un potente tipo matricial a Python. Los dataframes son las estructuras de datos que utiliza la librería panda para trabajar con la información [44].

Capítulo III

Metodología de la Investigación

Esta sección expone la metodología de la investigación y se definen: las etapas del diseño metodológico, los criterios de validación aplicados a un experimento supervisado en el campo de la medicina y la solución.

3 Planteamiento de la pregunta de investigación

El propósito de la investigación consiste en identificar cuál es el algoritmo de aprendizaje automático que permita clasificar y predecir el grado de complejidad del estado de un paciente cardiópata [45]. Para el mapeo sistemático se define la siguiente pregunta de investigación:

RQ: ¿Cuál es el impacto que tendrá el empleo de un algoritmo de aprendizaje automático para obtener un modelo que identifique la gravedad para la asistencia de los pacientes?

Esto posibilitará obtener un registro de los estudios vigentes para el análisis y predicción del grado de complejidad de la insuficiencia cardíaca. Se ha segmentado en sub-preguntas de investigación más específicas con el propósito de ser planteada.

SRQ1: ¿Cómo se clasifican los algoritmos de ML utilizados en los estudios existentes?

SRQ2: ¿Qué algoritmos se utilizan en cada estudio existente para predecir y analizar el grado de complejidad de pacientes cardiópatas?

Las interrogantes que se establecieron previamente, permitirán clasificar los algoritmos que utilizan ML y cuáles son los más utilizados para la clasificación y predicción del grado de complejidad del paciente, con el objetivo de contribuir en el desarrollo de posteriores investigaciones.

3.1 Hipótesis

En esta investigación, se enuncia la siguiente hipótesis:

El algoritmo de árbol de decisión es el más apropiado para la clasificación y análisis de registros, para estructurar un modelo que identifique correctamente el grado de afectación de un paciente cardiópata a través de pruebas a los pacientes como Análisis de sangre, electrocardiograma, ecocardiograma, las que ayudan a los médicos a determinar la causa de cualquier signo y síntoma y a decidir un plan de internación y tratamiento.

3.1.1 Variables de la investigación

Las variables definidas a continuación contribuyen en la realización del proyecto:

Variable dependiente: Predicción de la gravedad de la insuficiencia cardíaca que requiere tratamientos especializados en UCIC.

Con este contexto, la variable, se define como la predicción de la gravedad del paciente cardiópata. La precisión del algoritmo de ML, árbol de decisión, equivale a la variable independiente.

En síntesis, el planteamiento de esta investigación tiene como propósito predecir la gravedad del paciente cardiópata e identificar cuál es el porcentaje de precisión del modelo realizado, utilizando árboles de decisión.

3.2 Metodología de la Investigación

Para el desarrollo de este proyecto se seleccionó el método hipotético deductivo (MHD), conocido también como método científico, oportuno para obtener y analizar todo tipo de resultados y búsquedas, y de esta manera elaborar un prototipo innovador que se validará en pruebas supervisadas y monitoreadas, además de fundamentarse en investigaciones experimentales. A través de este procedimiento, se concluyó el presente estudio con una síntesis lógica.

3.3 Modalidad de la investigación

El proceso de la investigación se establece en la recopilación de datos y reseñas con respecto a un enunciado específico, con el objetivo de hacer comparaciones y suscitar nuevos estudios. Por este motivo, el desarrollo de la presente investigación fue 30% análisis bibliográfico, debido a que se consultaron diferentes fuentes académicas/científicas y 70% práctico, por la aplicación de algoritmos de ML para justificar resultados a través de pruebas, realizando una serie de predicciones

3.4 Tipo de investigación

En el presente estudio se utilizó el siguiente tipo de investigación:

Investigación cuasi experimental: "es aquella que tiene como objetivo poner a prueba una hipótesis causal manipulando (al menos) una variable independiente donde por razones logísticas o éticas no se puede asignar las unidades de investigación aleatoriamente a los grupos" (Fernández-García 2014)[46].

En un cuasiexperimento se obtienen diferentes tipos de hipótesis que se ajustan a los datos. Son tácticas dirigidas por algunos objetivos, que procuran analizar las relaciones entre variables independientes y la variable dependiente o de respuesta [29].

Capítulo IV

Resultados

4 Resultados

En esta sección se presenta los resultados obtenidos en la elaboración de esta tesis, basados en la aplicación de la metodología CRIPS-DM que se desarrolla en las siguientes etapas: Entendimiento de los datos, preparación de los datos, limpieza de los datos, generar una serie de modelos, verificación del algoritmo, construcción del modelo y el diseño metodológico.

4.1 Modelo CRIPS-DM

En este proyecto se seleccionó el modelo CRISP-DM dado que es flexible, ya que se necesita realizar una exploración de grandes cantidades de datos sin un objetivo de modelado específico. Es un modelo flexible que permite empezar con el proyecto una vez recibida una primera muestra. Se tendrán que incluir en la metodología actividades de carácter exploratorio que ayuden a determinar los objetivos de negocio a través de análisis sobre los propios datos. Con base a las etapas que presenta esta metodología, se desarrolló gran parte de la planeación del proyecto, adicionando tareas referentes al control del estado del proyecto las que se describen a continuación.

4.2 Entendimiento de los Datos

Para recolectar los datos necesarios en la investigación se realizaron entrevistas a los especialistas en este tema. Cada una de las variables que se tuvieron en cuenta fueron descritas para optimizar la comprensión de las mismas. Los datos contenidos en el almacén fueron sometidos a un análisis basado fundamentalmente en cuanto a representación de la realidad, consistencia, campos innecesarios, campos vacíos y datos de naturaleza híbrida o poco genuina. El médico es el que valora la analítica sanguínea, las pruebas de laboratorio y constantes vitales.

4.2.1 Preparación y análisis de datos

El conjunto de acciones y tecnologías para la preparación y análisis de datos consta de los siguientes pasos:

1. Acceso a los datos.
2. Recuperación de los datos.
3. Limpieza de los datos.
4. Formateo de los datos.
5. Combinación de los datos.

4.2.2 Acceso a los datos

Hay muchas fuentes de datos dentro del Instituto de Cardiología de Corrientes.

Según Velázquez, Navarro, Cobos [48]

"*La dificultad principal en el empleo de la captura electrónica de datos reside en una mayor complejidad de preparación, lo que fuerza la adaptación de los profesionales involucrados en el ensayo clínico a un sistema de trabajo distinto*".

La selección, recolección y definición adecuadas de las variables aportan la información requerida para cumplir el objetivo del trabajo, permiten visualizar previamente la validez del enfoque metodológico propuesto.

4.2.3 Recuperación de datos

En esta fase se define el conjunto de datos que se va a utilizar en el proyecto. En la exploración surge una combinación de datos estructurados y semiestructurados en diferentes tipos de repositorios. Datos del tipo historia clínica electrónica que registran sucesos de la enfermedad actual o historia médica. Otros datos son las pruebas de laboratorio, constantes vitales, farmacoterapia que está recibiendo o que ha recibido el paciente.

Fue necesario importarlos a un repositorio común antes de continuar con los pasos siguientes. El acceso y la recuperación son procesos manuales. Estos pasos requirieron una combinación de la experiencia de profesionales de salud y de TI. Dentro de este marco, la recuperación de los datos se realizó con un equipo multidisciplinar pequeño de profesionales, conformado por un conjunto de personas, con diferentes formaciones académicas y experiencias profesionales como médico cirujano cardiólogo, ingeniero en sistemas de software, Dra. En matemáticas e investigadores de la UNNE.

El análisis de los datos inicia con la descripción de los resultados de cada variable por separado. Para algunas variables existen intervalos de referencia que se establecen en personas sanas mediante métodos estadísticos. Estar fuera de este rango puede indicar presencia o ausencia de enfermedad o riesgo de padecerla, por este motivo se agregó esta información extra al conjunto de datos brindados. El médico especialista indicó también que además de estar fuera de rango es importante saber que tan fuera está. Las variables fuera de rango fueron propuestas por el equipo de profesionales informáticos.

Los datos se presentan en la Tabla 2.

Tabla 2: Resumen de los datos

N	Variable	Descripción
0	InCabNPrin	Número de registro. Valor de identificación o índice.
1	PacieNro	Número de Paciente. Valor de identificación único del Paciente.
2	Edad	Edad del paciente
3	PacieSexo	Sexo del paciente
4	InternacionesEnEMER	Cantidad de veces que el paciente ha sido internado en emergencia
5	InternacionesEnPISO	Cantidad de veces que el paciente ha sido internado en piso.
6	InternacionesEnRCVA	Cantidad de veces que el paciente ha sido internado en Recuperación Cardiovascular Adultos.
7	InternacionesEnRCVP	Cantidad de veces que el paciente ha sido internado en Recuperación Cardiovascular Pediátricos.
8	InternacionesUTI	Cantidad de veces que el paciente ha sido internado en Unidad de Terapia Intensiva
9	UltPotasio	Último valor de laboratorio de Potasio mmol/L.
10	FueraRangoPotasio3.5-5	Valor 0-1 si está en el rango 3.5 a 5 mmol/L
11	UltGlobulosRojos	Último valor de laboratorio de Glóbulos Rojosmillones/mm3
12	FueraRango GlobRojos4-5.2	Valor 0-1 si está en el rango 4 a 5.2 millones/mm3.
13	UltHemoglobina	Último valor de laboratorio de Hemoglobina g/dL
14	FueraRango Hemoglobina12-16	Valor 0-1 si está en el rango 12 a 16 g/dL
15	UltGlucemia	Último valor de laboratorio de Glucemia en mg/dL
16	FueraRango Glucemia70-100	Valor 0-1 si está en el rango 70 a 100 mg/dL
17	UltHematocrito	Último valor de laboratorio de Hematocrito en %
18	FueraRango Hematocrito	Valor 0-1 si está en el rango 36-46 %
19	UltCreatininaSerica	Último valor de laboratorio de Creatinina Sérica en mg/dL
20	FueraRango CreatininaSerica	Valor 0-1 si está en el rango 0.5-0.9 mg/dL
21	ECO_FEY	Valor de ecografía Fracción de eyección en %
22	ECO_IndiceDeMasa	Valor de ecografía Índice De Masa
23	ECO_Tisular	Valor de ecografía Tisular cm^2
24	ECO_VolumenAuriIzq	Valor de ecografía Volumen Aurícula Izquierda ml/m
25	FueraRango FEY50-75	Valor 0-1 si está en el rango de 50-75 %

N	Variable	Descripción
26	FueraRango IxMasa<95F<115M	Valor 0-1 si está en el rango de Masa
27	FueraRango ECO_Tisular<15	Valor 0-1 si es menor a 15 cm^2
28	FueraRango ECO_VolumenAuriIzq <34	Valor 0-1 si es menor a 34 ml/m
29	ECGAnormal	Actividad eléctrica del corazón
30	Obito	Defunción o fallecimiento de la persona

Esta fue la primera oportunidad para validar los datos durante el proceso de preparación y análisis. Con la participación de los expertos del dominio, a partir de los agrupamientos encontrados, se identificaron los patrones.

4.2.4 Limpieza de los datos

Analizando la base de datos, se encontró que el sistema almacena información detallada de identidad, salud y enfermedad de los pacientes. La misma está compuesta por hallazgos, consideraciones, resultados de exámenes complementarios e información sobre tratamientos instaurados, además de detalles del proceso de atención de cada uno de ellos. Como el conjunto de datos es demasiado grande, se extraerán a archivos de valores separados por comas (CSV) para un análisis posterior. Toda la información en el archivo CSV es anónima y no puede ser rastreada hasta ningún paciente específico. Esto garantiza que los datos de los pacientes individuales no se utilizarán indebidamente si los datos son robados o pirateados. Se aplicaron procedimientos de disociación de la información, de modo que los titulares de los datos no sean identificables, la técnica de disociación que fue empleada no permite identificar a persona alguna, conforme lo previsto en el artículo 28 de la Ley 25326 (Ley de protección de los datos personales).

Surgieron diferentes problemas, los conjuntos de datos representados en tablas no fueron perfectos, estaban expresados en diferentes formas, utilizaban abreviaturas, contenían errores de codificación, etc., y corregirlos de forma manual no fue del todo viable. Entre las situaciones más frecuentes se pueden nombrar: valores perdidos, valores fuera de rango, valores nulos y espacios en blanco que ocultan valores. También hubo algunos valores atípicos que podrían sesgar los resultados del análisis. Se tuvieron que corregir errores en los datos que afectan a la calidad, como por ejemplo eliminar caracteres extraños que pueden dificultar su proceso. La limpieza y la localización de valores extraños como por ejemplo posibles valores negativos se

realizó con consultas en SQL. En la sección 4.2.5 se detalla el procedimiento de análisis y limpieza de los datos.

4.2.5 Formateo

Una vez limpio el conjunto de datos, necesitó ser formateado para continuar con la preparación y análisis de datos. Este paso incluyó resolver problemas como ajustar múltiples formatos de fecha en los datos. También algunas variables de datos no son necesarias para el análisis y, por lo tanto, debieron ser eliminadas del conjunto de datos. Nuevamente, este es un paso que se beneficia de la automatización. Se debió revisar los datos que se han extraído y asegurar de que se entienda lo que significan todas las columnas. A su vez verificar si hay valores faltantes, que están en intervalos adecuados e, inclusive, en el formato de datos adecuados.

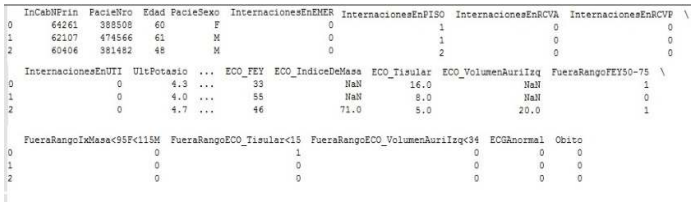
4.2.6 Combinación de los datos

Cuando los datos se han limpiado y formateado, el siguiente paso para la preparación y análisis de datos es transformarlo, fusionando, dividiendo o uniendo los conjuntos de entrada. Una vez que los datos se cargaron en el área de preparación del almacén de datos, existe una segunda oportunidad para la validación.

En la construcción del Dataset1 algunos datos a tener en cuenta, fueron los siguientes:

- ✓ El dataset inicial con el que comenzó el trabajo contenía 696 registros y 31 columnas. La primera fila o cabecera del mismo contiene los nombres de cada campo.
- ✓ Los espacios en blanco en los campos de tipo texto y los puntos en los campos de tipo numérico corresponden a valores faltantes o perdidos.
- ✓ Cada registro tiene un identificador único de registro o índice (InCabNPrin)

En la siguiente figura 3 se ilustra el dataset inicial devuelto por la función head() de la librería Pandas.



	InCabNPrin	PacieNro	Edad	PacieSexo	InternacionesEnEMER	InternacionesEnPISO	InternacionesEnRCVA	InternacionesEnRCVP	\
0	64261	388508	60	F	0	1	0	0	
1	62107	474566	61	M	0	1	0	0	
2	60406	381482	48	M	0	2	0	0	

	InternacionesEnUTI	UltPotasio	...	ECO_FEY	ECO_IndiceDeMasa	ECO_Tisular	ECO_VolumenAuriIzq	FueraRangoFEY50-75	\
0	0	4.3	...	33	NaN	16.0	NaN	1	
1	0	4.0	...	55	NaN	8.0	NaN	0	
2	0	4.7	...	46	71.0	5.0	20.0	1	

	FueraRangoIxMasa<95F<115M	FueraRangoECO_Tisular<15	FueraRangoECO_VolumenAuriIzq<34	ECGAnormal	Obito
0	0	1	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0

Fig. 3: Dataset inicial volcado por función head()

4.3 Análisis exploratorio y preprocesamiento

El primer paso, es realizar un pequeño análisis exploratorio del dataset; es decir, se hace uso de algunas herramientas de estadística, junto con algunas visualizaciones para entender un poco más los datos que se dispone. Entre las herramientas disponibles a utilizar están: Python, matplotlib, pandas y sci-kit learn. Utilizando simples expresiones de Python, se carga la base de datos en un dataframe de Pandas; lo que va a permitir manipular los datos con suma facilidad. Como primer paso, mediante la librería de pandas y la función read_csv(), se genera un dataframe a través de la lectura del archivo CSV. En la Fig. 4 se muestra código en Python para incorporar librerías de numpy y pandas lo que permite leer el dataset.

```
In [1]: #     Tratamiento de datos      #
        # ------------------------------#
        import numpy as np
        import pandas as pd
        datos  = pd.read_csv("7mo DATASET - Hoja 1.csv", sep=',')
        #Devuelve los primeros elementos de la estructura
        dato.head(3)
        datos.info()
```

Fig. 4: Captura de pantalla de incorporación de librerías

La función **info()** de Pandas se utiliza para obtener un resumen conciso del marco de datos. Es útil cuando se hace un análisis exploratorio de los datos.

Hay columnas con tipo «object» cuando debería ser un float o Int. Esto es porque hay filas que no tienen valor para esas columnas. En la Fig. 5 se ilustran los tipos de datos de las columnas del dataset.

```
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   InCabNPrin                696 non-null    int64
 1   PacieNro                  696 non-null    int64
 2   Edad                      696 non-null    int64
 3   PacieSexo                 696 non-null    object
 4   InternacionesEnEMER       696 non-null    int64
 5   InternacionesEnPISO       696 non-null    int64
 6   InternacionesEnRCVA       696 non-null    int64
 7   InternacionesEnRCVP       696 non-null    int64
 8   InternacionesEnUTI        696 non-null    int64
 9   UltPotasio                695 non-null    float64
 10  FueraRangoPotasio3.5-5    696 non-null    int64
 11  UltGlobulosRojos          642 non-null    float64
```

Fig. 5:Variables y sus tipos de datos.

Lo primero a controlar es si existen valores faltantes o nulos; en el caso de un paciente pudo no haberse recolectado la edad del mismo o la información de algún análisis clínico. Estos valores suelen tomar la forma de NaN o None también se indica la ausencia de valor utilizando alguna palabra reservada o etiqueta como NULL, esto se puede realizar utilizando el método isnull() como se ilustra en Fig. 6 donde se aplica la función al conjunto de datos.

```
In [2]: # Controlando la cantidad de registros
        datos['InCabNPrin'].count()

Out[2]: 696

In [3]: # Controlando valores nulos
        datos.isnull().any().any()

Out[3]: True
```

Fig. 6: Funciones de conteo y determinación de nulos.

El método devuelve el valor "True", lo que indica que existen valores nulos en el dataset. Estos valores pueden tener una influencia significativa en el modelo predictivo, por lo que siempre es una decisión importante determinar la forma en que se los va a manejar. Las alternativas son:

1. Dejarlos como están, lo que a la larga va a traer bastantes problemas ya que en general los algoritmos no los suelen procesar correctamente y provocan errores.
2. Eliminarlos, lo que es una alternativa viable, aunque, dependiendo la cantidad de valores nulos, puede afectar significativamente el resultado final del modelo predictivo. El objeto se puede eliminar por completo o lo podemos omitir en ciertos casos.
3. Agregar objetos con todos los valores posibles, incluso se pueden ponderar los valores de acuerdo a su probabilidad de aparecer.
4. Inferir su valor. En este caso, lo que se puede hacer es tratar de inferir el valor faltante y reemplazarlo por el valor inferido. Esta suele ser generalmente la mejor alternativa a seguir.

En esta primera propuesta se intenta utilizar la última alternativa. Se procede a inferir los valores faltantes utilizando la media aritmética para los datos cuantitativos. En el primer paso se identifican qué columnas del dataset corresponde a cada tipo de datos; para realizar esto se utiliza el atributo dtypes del dataframe de Pandas. A medida que se exploran las características, se debe prestar atención a cualquier columna que:

- ✓ Esté mal formateada.
- ✓ Requiere más datos o mucho preprocesamiento para convertirse en una característica útil.
- ✓ Contiene información redundante.

En la Fig. 7 se muestra como en Python se agrupan a las columnas según su tipo haciendo uso de la función groupby()

```
In [4]: # Agrupando columnas por tipo de datos
        tipos = datos.columns.to_series().groupby(datos.dtypes).groups
        # Armando lista de columnas categóricas
        ctext = tipos[np.dtype('object')]
        len(ctext)  # cantidad de columnas con datos categóricos.
Out[4]: 1

In [5]: # Armando lista de columnas numéricas
        columnas = datos.columns  # lista de todas las columnas
        cnum = list(set(columnas) - set(ctext))
        len(cnum)
Out[5]: 30
```

Fig. 7: Agrupar columnas por tipos de datos.

4.3.1 Limpieza del conjunto de datos

Para limpiar el conjunto de datos, es necesario manejar los valores que faltan y las características categóricas, los valores obtenidos para algunas columnas que contienen datos nulos.

Ahora, que se pudo separar las 31 columnas que tiene el dataset, se observa que una (1) columna contiene datos categóricos y treinta (30) contienen datos cuantitativos. Se procede a inferir los valores faltantes. En la Fig. 8: se ilustra la asignación de los valores faltantes y en la Tabla 3: Valores de Media aritmética a asignar.

```
In [6]: # Completando valores faltantas datos cuantititavos
        for c in cnum:
            mean = datos.mean()
            datos[c] = datos[c].fillna(mean)

In [7]: # Completando valores faltantas datos categóricos
        for c in ctext:
            mode = datos[c].mode()[0]
            datos[c] = datos[c].fillna(mode)
```

Fig. 8: Asignación de los valores faltantes.

Tabla 3: Valores de Media aritmética

Estudio	Media
Glóbulos Rojos	4.1
Potasio	4.1
Hemoglobina	13.6
Glucemia	140.00
Hematocrito	37
Creatinina Sérica	0.97
ECO. FEY	55
ECO Índice De Masa	127
ECO Tisular	17
ECO Volumen Auri Izq	29

En la librería Pandas, existe una función útil, denominada **describe()**, para conocer una serie de estadísticas del conjunto de datos. Esta función permite conocer valores típicos de manera inmediata, como los distintos percentiles de cada una de las características, la media, los valores

mínimos y máximos, etc. En Fig. 9 y Fig. 10 ilustran los valores devueltos para la función aplicado a este primer dataset.

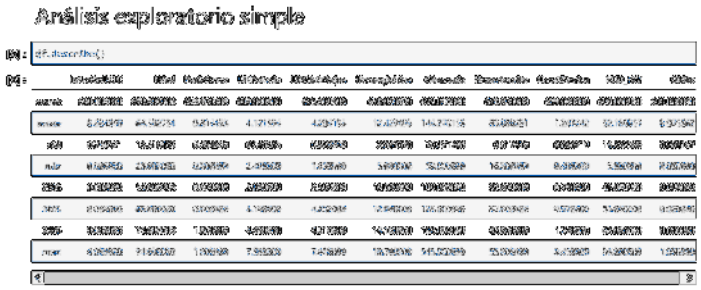


Fig. 9: Resultado estadístico de todo el Dataset.

estadística del dataframe: \A

	PacieNro	Edad
count	690.000000	690.000000
mean	274783.337681	64.592754
std	181619.546435	13.316433
min	2562.000000	23.000000
25%	103125.500000	56.000000
50%	254437.000000	66.000000
75%	483266.250000	74.000000
max	514327.000000	94.000000

Fig. 10: Detalle de estadística para algunas variables.

Este conjunto resultado es presentado al especialista médico Dr. Jorge Parra, quien indicó que los valores estadísticamente son los que se presentan con más frecuencia.

Con un conjunto de datos procesados y explorados, es necesario crear una matriz de variables independientes y un vector de variables dependientes. El método de selección de variables que se utilizó fue elección hacia adelante (Fordward Stepwise Regression). En este método, las variables se introducen secuencialmente en el modelo. La primera variable que se introduce es la de mayor correlación (+ o -) con la variable dependiente en este caso óbitos. La variable independiente se introducirá en la ecuación solo si cumple el criterio de entrada.

Se selecciona la variable independiente cuya correlación parcial sea la mayor y que no esté en la ecuación. El procedimiento termina cuando ya no quedan variables que cumplan el criterio de entrada.

Identificamos la importancia de los predictores en el modelo, siendo óbito la variable target. En esta primera etapa del trabajo, se descartan las variables de identificación como 'InCabNPrin' o 'PacieNro', los principales resultados obtenidos fueron que tomando óbito como variable se presenta un problema de clasificación binaria. Como se observa en la Fig. 11, la relación entre

el número de muestras en la clase 1 que representa vivo y el número de muestras en la clase 2 que representa óbito, en el conjunto la cantidad de elementos de clase 1 es de 99%. La regresión logística se utiliza para la clasificación y el resultado final es que ignora la clase 2 y clasifica todas las muestras de entrenamiento como de la clase 1.

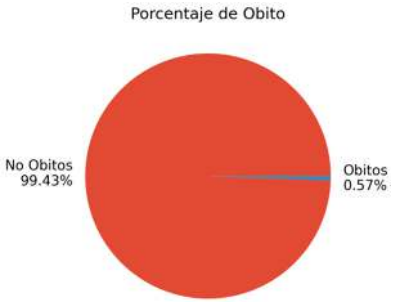


Fig. 11: Relación entre el número de muestras en cada clase.

Con la versión de la biblioteca **scikit-learn** instalada, podemos recuperar la propiedad **feature_** que contiene los coeficientes encontrados para cada variable de entrada [29] y [43].

En la Fig. 12 se ilustra el cálculo de los predictores tomando como target la variable óbito del dataset.

```
importancia_predictores = pd.DataFrame(
                            {'predictor': datos.drop(columns = "Obito").columns,
                             'importancia': modelo.feature_importances_}
                            )

print("Importancia de los predictores en el modelo")
print("-------------------------------------------")
importancia_predictores.sort_values('importancia', ascending=False)
```

Fig. 12: Aplicación de la propiedad feature.

Estos coeficientes pueden proporcionar la base para una puntuación de importancia. Esto supone que las variables de entrada tienen la misma escala o se han escalado antes de ajustar al modelo. Los valores se muestran en la Tabla 4.

Tabla 4: Puntuación de importancia de los predictores

Orden	Predictor	Importancia
6	UltHematocrito	0.491646
8	ECO_FEY	0.216633
1	UltPotasio	0.209942
7	UltCreatininaSerica	0.081779
0	Edad	0.000000

En el análisis práctico del conjunto de datos realizaremos un estudio en profundidad de cada una de las características, ya que, al ser un conjunto de dato orientado al diagnóstico médico, el conocimiento en esta área resulta ser insuficiente para poder entender los valores que representan estos datos. Según refiere el especialista Dr. Jorge Parras *"...Existen indicadores muy útiles sobre la evolución y el pronóstico de un paciente, y ayudan a identificar quién precisa un mayor seguimiento y control médicos. Los niveles de colesterol, glucosa o presión arterial no son suficientes para poder identificar a tiempo, y de forma precisa, este tipo de enfermedades. Son importantes los marcadores sanguíneos para el diagnóstico y el pronóstico de las enfermedades cardiovasculares como por ejemplo hemoglobina que se encuentra en los glóbulos rojos"*.

En las Fig. 13 se muestra representación del histogramas de la variable glóbulos rojos.

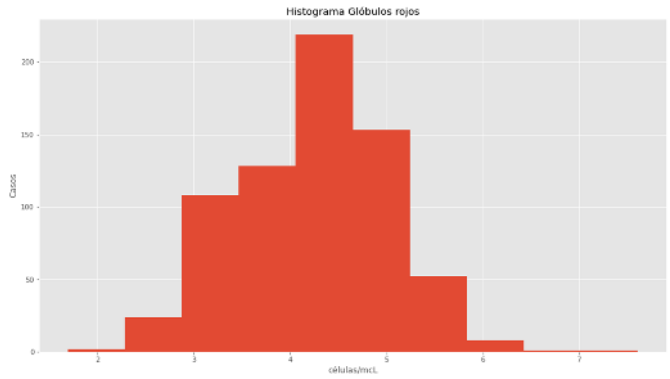


Fig. 13: Histograma para valores de glóbulos rojos.

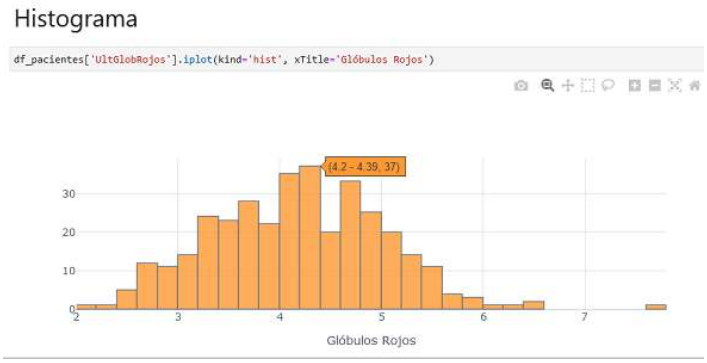


Fig. 14: Visualización de histograma glóbulos rojos.

La herramienta de visualización utilizada para representar la Fig. 14, identifica los picos y sus valores de manera más clara, son los conglomerados más altos de las barras, representan los

valores más comunes. Hay 37 registros con los valores comprendidos entre 4.2 y 4.9 millones/mm^3.

En la Fig. 15 se muestra el histograma de la variable potasio, identificándose el valor pico 4.1 mmol/L en el gráfico.

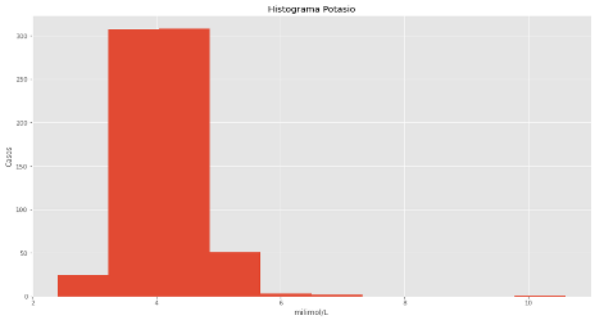


Fig. 15: Histograma de potasio.

En la Fig. 16 se ilustra el histograma de la variable creatinina sérica y ecografía FEY, identificándose para la creatinina sérica el pico muy próximo al valor 1 mg/dL. Para la ecografía FEY el valor pico se muestra próximo a 50%.

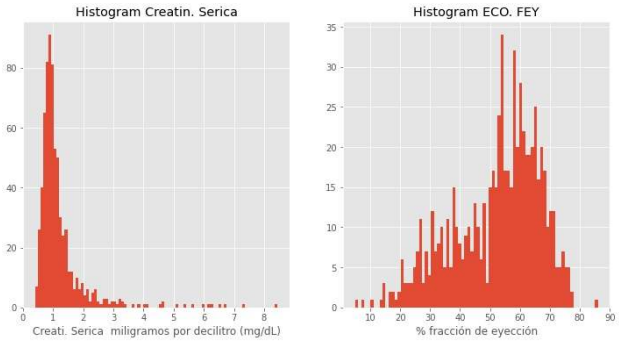


Fig. 16: Histograma de creatinina sérica y ecografía FEY.

Se mostró al especialista Dr. Jorge Parras el histograma de cada una de las características, y así poder analizar a grandes rasgos que se encuentra en cada una de ellas para lograr detectar alguna anomalía/outlayer. El especialista señaló que un resultado normal es de 0.7 a 1.3 mg/dL y que la interacción entre corazón y riñón a menudo se encuentra alterada en pacientes cardiópatas. Ambas comparten factores de riesgo y a menudo coexisten, pudiendo empeorarse mutuamente.

Una vez observadas las características del dataset, se analizan algunas de ellas en más detalle, mediante las cajas de bigotes. Los diagramas de caja-bigotes, también conocidos como boxplots o box and whiskers; los cuales son una presentación visual que permiten identificar la dispersión y simetría.

Para los valores de ecografía de fracción de eyección, el valor medio es 52%. Se observa en la Fig. 17 que el desplazamiento de las gráficas de caja hacia arriba indica que hay una mayor concentración de valores entre 50% y el 75% de los mismos, se presentan datos atípicos, alejados de los valores normales.

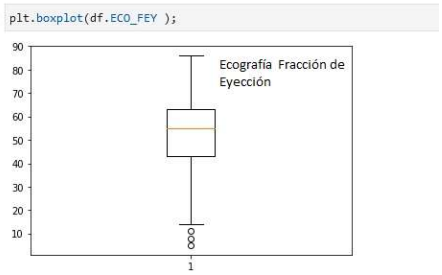


Fig. 17: Caja de bigotes para atributo ecografía fracción de eyección.

Para la variable edad, a simple vista en la Fig. 18 se destaca la media de edad en la que se encuentran los pacientes, alrededor de los 65 años, mientras que la mínima edad es de unos 23 años, y la máxima de 94 años, como se observa en la gráfica siguiente.

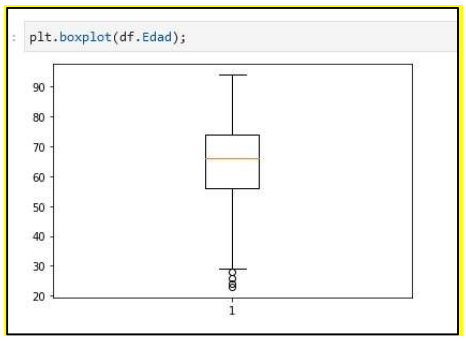


Fig. 18: Caja de bigotes para atributo edad.

Aquí también se observan datos atípicos, alejándose estos de los valores normales. En este caso, los valores podrían ser determinantes ya que, en el ámbito de la medicina, valores diferentes a lo normal suelen ser importantes. A pesar de encontrar valores anómalos en algunas características, no se los va a tomar como errores ya que pueden ser un punto definitivo a la hora de realizar un diagnóstico médico.

Como se indicó en el Capítulo II sobre la visualización de datos o data viz, su objetivo principal es comunicar información de manera clara y eficiente a través de gráficos estadísticos. Aquí ayuda al especialista a analizar y razonar datos y pruebas, hace que los datos complejos sean más comprensibles y utilizables.

En la Fig. 19 se utilizan dos herramienta de visualización, al compararlas, una presenta mayor referencia de información que la otra.

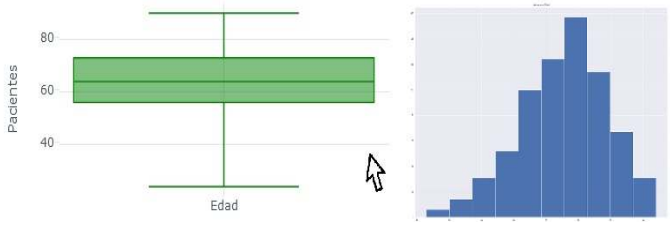


Fig. 19: Visualización caja de bigotes e histograma de atributo edad.

Una visualización rápida hace que sea mucho más fácil conocer los datos. En la Fig. 20 se ilustra la exploración de los datos de forma intuitiva, la herramienta despliega información interactiva.

Fig. 20: Exploración caja de bigotes atributo edad.

Según refiere el especialista Dr. Jorge Parras *"...se nota una distribución cuasi Gausiana para la edad que llega hasta los 90 años..."*

Con esto se puede comenzar a explorar los datos, por ejemplo, determinar el porcentaje de óbitos que están incluidos en la base de datos con la que estamos trabajando.

```
In [14]: # Calculando el porcentaje de Obitos sobre toda la base de datos
         porcent_Obito = (datos[datos.Obito  > 0]['Obito'].count() * 1.0
                / datos['Obito'].count()) * 100.0
         print("El procentaje de Obito de la base de datos es {0:.2f}%" .format(porcent_Obito))

Out[14]: El procentaje de Obito de la base de datos es 0.57%
```

Fig. 21: Cálculo porcentual por categoría.

Es posible ver en la Fig. 21, cómo utilizando funciones de Python se calculan los porcentajes. El porcentaje de pacientes óbitos es bajo, solo un 0.57 % del total de la base de datos, que cuenta con 696 registros. Este es un dato importante a tener en cuenta ya que, al existir tanta diferencia entre las clases a clasificar, esto puede afectar considerablemente al algoritmo de aprendizaje. En la Fig. 22 se muestra el uso de la función de agrupamiento para determinar las cantidades en cada categoría y la representación gráfica de las mismas.

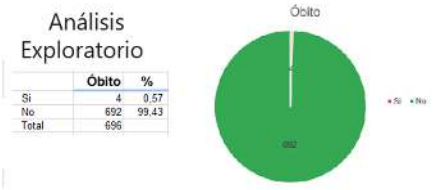


Fig. 22: Porcentaje de óbitos en el dataset.

En la Fig. 23 se ilustra el conteo de registros en cada categoría y el gráfico que permite comparar ambos totales.

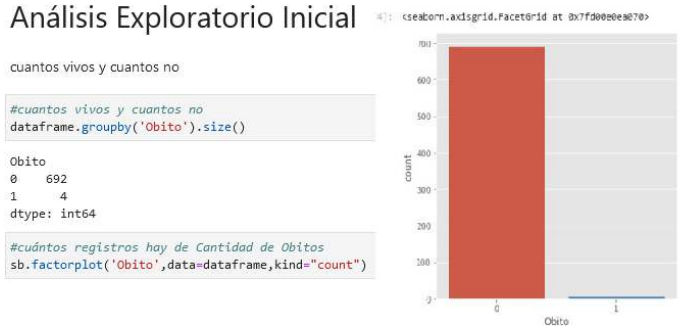


Fig. 23: Totales por clases del atributo óbitos en el dataset.

En este caso al ser sólo dos posibilidades: vivo o muerto, o pertenecientes a clase 1 y clase 2, se habla de una matriz binaria. Más adelante se realiza un análisis del tema.

4.4 Entrenamiento

Una vez analizado el conjunto de datos, es posible generar una serie de modelos utilizando distintas técnicas estudiadas teóricamente. Cabe recordar que, para realizar un correcto análisis, se debe dividir el conjunto de datos en dos subconjuntos, denominados training y test, respectivamente.

Para determinar la gravedad de los pacientes se desarrolló un prototipo aplicando el algoritmo de aprendizaje supervisado, se utilizó en esta instancia árboles de decisión. El dataset que fue utilizado inicialmente contiene 696 registros de pacientes atendidos en FUNCACORR. Este prototipo tomó el 80% de los registros, determinados en la base de datos para su entrenamiento y el valor de porcentaje faltante fue orientado a pruebas. La técnica de clasificación empleó la extracción de características, para el entrenamiento y para las pruebas. Utilizando la versión 0.23.1 de scikit-learn o sklearn que como se mencionó en capítulo II en 2.4.3.1, de los paquetes para ciencia de datos y aprendizaje automático, se tomará específicamente la función train_test_split().

Primero se debe importar train_test_split() y NumPy antes de poder usarlos, por lo que puede comenzar con las declaraciones **import**. En la Fig. 24 se ilustra la importación de las librerías necesarias.

```
Python
>>> import numpy as np
>>> from sklearn.model_selection import train_test_split
```

Fig. 24: Importación de librerías numpy y sklearn.

Con train_test_split(), debe proporcionar las secuencias que se va a dividir, así como los argumentos opcionales, lo que se muestra en la Fig. 25.

```
# División de los datos en train y test
# ------------------------------------------------------------------------------
X_train, X_test, y_train, y_test = train_test_split(
                                       datos.drop(columns = "Obito"),
                                       datos['Obito'],
                                       random_state = 123
                                   )
X_train = X_train.replace((np.inf, -np.inf, np.nan), 0).reset_index(drop=True)
# Creación del modelo
# ------------------------------------------------------------------------------
modelo = DecisionTreeRegressor(
            max_depth         = 7,
            random_state      = 123
         )
```

Fig. 25: Uso de la función train_test_split().

Options son los argumentos de palabras clave opcionales que puede usar para obtener el comportamiento deseado:

train_size es el número que define el tamaño del conjunto de entrenamiento. Si proporciona un real, entonces debe estar entre 0.0 y 1.0 y definirá la parte del conjunto de datos utilizada para la prueba. Si proporciona un entero, representará el número total de muestras de capacitación. El valor predeterminado es None.

test_size es el número que define el tamaño del conjunto de prueba. Es muy similar a train_size. Debe proporcionar train_size o test_size. Si no se proporciona ninguno, entonces el porcentaje predeterminado del conjunto de datos que se usará para la prueba es 0.25, o 25 por ciento.

random_state es el objeto que controla la aleatorización durante la división. Puede ser un entero, se usa para establecer una semilla en el generador aleatorio, de modo que sus divisiones de prueba de tren siempre sean deterministas.

Como ya se mencionó, el dataset que fue utilizado inicialmente contiene 696 registros. Este prototipo tomó el 80% de los registros, determinados en la base de datos para su entrenamiento y el valor de porcentaje faltante fue orientado a pruebas.

Se importaron las librerías correspondientes, utilizando las declaraciones **import** de Python. Se proporciona las secuencias que desea dividir, así como los argumentos opcionales.

Shuffle es el objeto booleano (True por defecto) que determina si se baraja el conjunto de datos antes de aplicar la división.

De entre las ventajas del árbol de decisiones especificadas en el Capítulo II, resaltamos algunas de las principales virtudes:

- ✓ **Integral:** los árboles de decisiones obligan a evaluar todos los posibles resultados de una elección.
- ✓ **Gráfico:** No se basan en fórmulas. Son fáciles de entender y el beneficio de usar modelos en la toma de decisiones es que fácilmente permite ser compartido con otros para que aporten sus opiniones.

Para crear el árbol se utiliza la librería de sklearn tree.DecisionTreeClasifier pues se busca un árbol de clasificación, no de regresión. El código de la figura 26 crea el árbol, lo entrena utilizando el conjunto de entrenamiento y lo dibuja.

```
# Entrenamiento del modelo
# ------------------------------------------------------------------------------
modelo.fit(X_train, y_train)
# Estructura del árbol creado
# ------------------------------------------------------------------------------
fig, ax = plt.subplots(figsize=(30, 30))

print(f"Profundidad del árbol: {modelo.get_depth()}")
print(f"Número de nodos terminales: {modelo.get_n_leaves()}")

plot = plot_tree(
            decision_tree = modelo,
            feature_names = datos.drop(columns = "Obito").columns,
            class_names   = 'Obito',
            filled        = True,
            impurity      = False,
            fontsize      = 10,
            precision     = 2,
            ax            = ax
       )
texto_modelo = export_text(
                   decision_tree = modelo,
                   feature_names = list(datos.drop(columns = "Obito").columns)
               )
print(texto_modelo)
```

Fig. 26: Creación de árbol con la función Decision_Tree.

La función export_text() representa esta misma información en formato texto. Esto genera una estructura jerárquica para la clasificación de objetos. El árbol representado en forma gráfica se lo ve en la figura 27.

Profundidad del árbol: 5

```
Número de nodos terminales: 9
|--- UltCreatininaSerica <= 3.13
|   |--- ECO_FEY <= 35.50
|   |   |--- UltCreatininaSerica <= 0.69
|   |   |   |--- UltPotasio <= 4.00
|   |   |   |   |--- value: [1.00]
|   |   |   |--- UltPotasio >  4.00
|   |   |   |   |--- value: [0.00]
|   |   |--- UltCreatininaSerica >  0.69
|   |   |   |--- ECO_FEY <= 34.50
|   |   |   |   |--- value: [0.00]
|   |   |   |--- ECO_FEY >  34.50
|   |   |   |   |--- UltPotasio <= 3.80
|   |   |   |   |   |--- value: [1.00]
|   |   |   |   |--- UltPotasio >  3.80
|   |   |   |   |   |--- value: [0.00]
|   |--- ECO_FEY >  35.50
|   |   |--- value: [0.00]
|--- UltCreatininaSerica >  3.13
|   |--- UltHematocrito <= 41.50
|   |   |--- value: [0.00]
|   |--- UltHematocrito >  41.50
|   |   |--- UltGlucemia <= 86.00
|   |   |   |--- value: [0.00]
|   |   |--- UltGlucemia >  86.00
|   |   |   |--- value: [1.00]
```

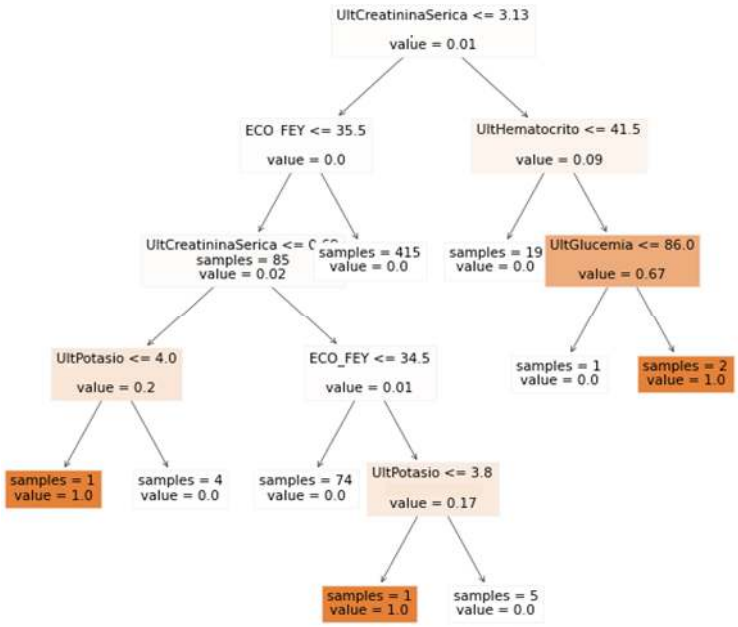


Fig. 27: Árbol de clasificación.

4.5 Pruebas del algoritmo

Al momento de realizar la división de los datos adquiridos, se determinó que no se asigna valor a la opción **train_size,** lo que define que el 25% de los mismos se destinaría para las pruebas del algoritmo, mientras que el valor restante se utilizaría para realizar el entrenamiento.

4.5.1 Verificación del algoritmo árboles de decisión

En la Fig. 28 se visualiza la matriz de confusión resultante de las pruebas desarrolladas por el modelo. Esta herramienta también es conocida como matriz de error o tabla de contingencia.

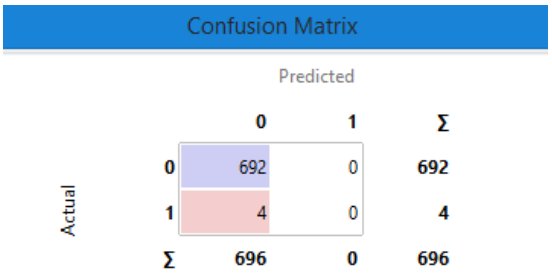


Fig. 28: Resultado de la matriz de confusión.

En los casos de clasificación binaria, una de las clases es la que más nos interesa hacer una identificación mejor, es decir, es la clase que perseguimos. Para nuestro caso los óbitos. Como desarrollador, una de las comprobaciones que se debió realizar es verificar que este algoritmo haya realizado la predicción correcta, evaluando así la especificidad y la sensibilidad del modelo. Para determinar que el modelo es correcto, lo ideal es que la sensibilidad y la especificidad sean iguales a uno. Esto puede indicar que el modelo identificó correctamente los verdaderos positivos y verdaderos negativos, o que no hubo falsos negativos ni falsos positivos. En la Fig. 29 se ilustra la matriz de confusión y sus partes.

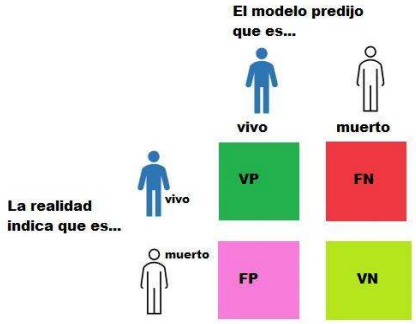


Fig. 29: Esquema de la matriz de confusión.

La exactitud o Accuracy, se refiere a lo cerca que está el resultado de una medición del valor verdadero. Según la ecuación 5, se calcula como:

$$\boldsymbol{Accuracy} = \frac{(\boldsymbol{VP+VN})}{(\boldsymbol{VP+FP+FN+VN})} (5)$$

La matriz de confusión devuelve los valores que se ilustra en la Fig. 30.

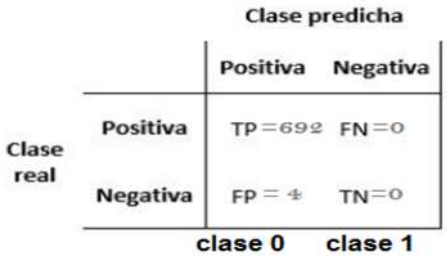


Fig. 30: Valores de la matriz de confusión

La aplicación de la formula sobre los valores obtenidos queda de la siguiente forma:

Accuracy = (692+0)/(692+4+0+0)= 692/696=0.99

Es decir, que un 99% de los pacientes a los que evaluamos estarán realmente vivos.

Para el cálculo de la Precisión, se aplica la ecuación 6:

$$Precisión = \frac{VP}{(VP + FP)} \quad (6)$$

Se debe calcular para cada una de las clases.

En este caso precisión clase 0= 692/(692+4)= 692/696= 0.99 y la Precisión clase 1= 0/(0+4)=0/4= 0.0

La sensibilidad y la especificidad son dos valores que indican la capacidad de nuestro estimador para discriminar los casos positivos, de los negativos. La sensibilidad se representa como la fracción de verdaderos positivos, mientras que la especificidad, es la fracción de verdaderos negativos.

4.5.2 La Sensibilidad/ Sensitivity o Recall

Sensibilidad= 692/(692+0)=692/692= 1

En el área de la salud decimos que la sensibilidad es la capacidad de poder detectar correctamente el fallecimiento entre los enfermos.

4.5.3 La Especificidad o Especificity

También conocida como la Tasa de Verdaderos Negativos, true negative rate o TN. Según la ecuación 7, se calcula:

$$E = \frac{VN}{(VN + FP)} \qquad (7)$$

En este caso la Especificidad= 0/(0+4) = 0/4= 0

La tasa de falsos positivos se calcula como FP / FP + TN, donde FP es el número de falsos positivos y TN es el número de verdaderos negativos (FP + TN es el número total de negativos). Es la probabilidad de que se produzca una falsa alarma, lo que supone que se dé un resultado positivo cuando el valor verdadero sea negativo.

La tasa de falsos positivos= 4 /(4 + 0) =4/4= 1

Recall: Se conoce como Tasa de Verdaderos Positivos, True Positive Rate o TP. Es la proporción de casos positivos que fueron correctamente identificadas por el algoritmo. Se calcula según la ecuación 8, así:

$$\boldsymbol{Recall = \frac{VP}{(VP + FN)}} \qquad (8)$$

Recall = 692/(692+0)= 692/692 =1

F1 SCORE resume la precisión y sensibilidad en una sola métrica. Es de gran utilidad cuando la distribución de las clases es desigual, como en este caso, cuando el número de pacientes con una condición es de 0.57 % y el otro es 99.43%, lo que en el campo de la salud es común.

Se calcula siguiendo la ecuación 9:

$$F1 = \frac{2(Recall * Precisión)}{(Recall + Precisión)} \qquad (9)$$

Conforme a estas nuevas métricas podemos obtener **cuatro casos posibles para cada clase:**

- ✓ **Alta precisión y alto recall:** El modelo de ML escogido maneja perfectamente esa clase.
- ✓ **Alta precisión y bajo recall:** El modelo de ML escogido no detecta la clase muy bien, pero cuando lo hace es altamente confiable.
- ✓ **Baja precisión y alto recall:** El modelo de ML escogido detecta bien la clase, pero también incluye muestras de la otra clase.

- ✓ **Baja precisión y bajo recall:** El modelo de ML escogido no logra clasificar la clase correctamente.

Para la clase 1,

F1= 2*(1 * 0.0)/(1 + 0.0)= 2* (0/1)= 2 * 0= 0

Cuando tenemos un dataset con desequilibrio, suele ocurrir que obtenemos un alto valor de precisión en la clase mayoritaria y un bajo recall en la clase minoritaria, cero en este caso. En el campo de la salud ésta circunstancia es particularmente frecuente y por ello se debe recurrir al balanceo de clases.

4.6 Problemas causados por datos desbalanceados

Tanto en el grupo de entrenamiento y el de prueba el conjunto de datos es no balanceados, la tasa de precisión obtenida es falsamente alta. Por ejemplo, en datos no balanceados, cuando la proporción de muestras positivas y negativas es 9:1, cuando su precisión es del 90%, la predicción del modelo tiende a clasificar las muestras en la clase mayoritaria. Conduce a una predicción ineficaz. La información contenida en la clase minoritaria es muy limitada, por lo que es difícil determinar la distribución de los datos de la misma, es decir, es difícil encontrar las reglas dentro de ella.

4.7 Interpretaciones

De los resultados experimentales se pueden extraer las siguientes proposiciones:

El modelado de la explotación de datos es un proceso de aproximación cíclica, el cual se debe ir mejorando a medida que se conoce más de la información con la cual se está trabajando. Es por esto que es necesario reiniciar el ciclo hasta que la información obtenida satisfaga el requerimiento que la produjo. Las características que utilizaremos como entradas para aplicar el algoritmo serán valores numéricos, continuos en lo posible, tomaremos las columnas "ECO_FEY", "UltCreatininaSerica", "UltPotasio". Valores categóricos como por ejemplo sexo= v-Hombre y m-Mujer, se puede intentar pasarlo a valor numérico, pero no es recomendable pues no hay una distancia real. No conviene utilizar características o variables que estén correlacionados o que sean escalares de otros. "UltCreatininaSerica" tienen una correlación "UltPotasio" ambas poseen una correlación de características de Pearson = 0.12. En la Fig. 31 se ilustra la aplicación de la función *corr()* la que encuentra la correlación entre las columnas del dataframe. Los valores del coeficiente de correlación de Pearson aparecen coloreados.

```
colormap = plt.cm.viridis
plt.figure(figsize=(15,15))
plt.title('Correlación de características de Pearson', y=1.05, size=20)
sb.heatmap(Pacientes_encoded.astype(float).corr(),linewidths=0.1,vmax=1.0,
           square=True, cmap=colormap, linecolor='white'
```

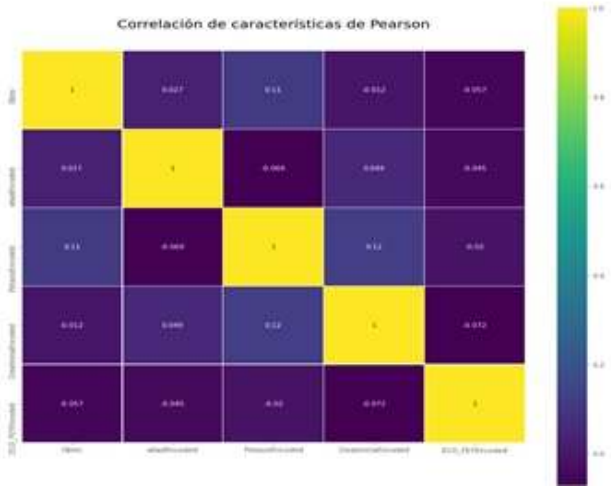


Fig. 31: Matriz de correlación de Pearson.

En palabras del especialista, se pueden identificar una relación entre el potasio y la creatinina lo que está explicado por la función renal.

El coeficiente de correlación puede tomar un rango de valores de +1 a -1. Un valor de 0 indica que no hay asociación entre las dos variables. Un valor mayor que 0 indica una asociación positiva.

4.8 Regresión Logística

A partir de un conjunto de datos de entrada o características ('Ult. Hematocrito', 'Ult. Creatinina Sérica', 'ECO_FEY', 'Edad'), la salida será discreta**,** no continua por eso se utiliza Regresión Logística y no Regresión Lineal. Como se vio en el capítulo II en 2, la Regresión Logística es un algoritmo Supervisado y se utiliza para clasificación, la variable dependiente es categórica.

Se procedió a clasificar un problema con posibles estados o un número finito de etiquetas o clases. Se procede a transformar varios de los datos de entrada en valores categóricos. Las edades, las separamos en: Menor de 21 años pertenecen al grupo 0, entre 21 años y 40 años como del grupo 1, entre más de 40 años y 60 años como del grupo 2, al grupo 3 pertenecen quienes estén entre 61 años y 80 años, grupo 4 más de 80 años, etc. El potasio menos de 3.5 mmol/L y más de 5 mmol/L. La ecografía de fracción de eyección FEY menor 50% y FEY mayor 75%.

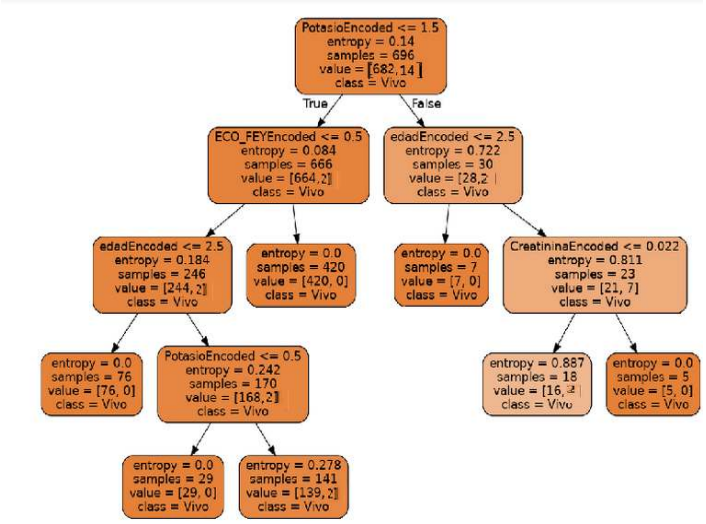


Fig. 32: Árbol de clasificación.

En la Fig. 32 se interpreta que en el nodo superior o raíz del árbol, se puede ver que parte de un espacio con 696 muestras (samples = 696) de las que 682 pertenecen a la clase 0 y 14 a la clase 1 (value = [682, 14]) . Según el concepto de entropía, cuanto mayor es la entropía, mayor es la incertidumbre y se necesita más información para resolver las cosas. La entropía es de 0.14. La característica escogida sobre la que realizar la pregunta es 'PotasioEncoded' y la pregunta en cuestión es "¿es 'PotasioEncoded' menor o igual a 1.5mmol/L?" ('PotasioEncoded' ≤ 1.5). Esto divide el conjunto de datos en dos bloques: El primero es el formado por aquellos puntos que responden positivamente a la pregunta anterior; este subconjunto de los datos está formado por unas 666 muestras de la clase 0 o vivos. El segundo nodo, de color salmón, está formado por los puntos que no responden positivamente a la pregunta realizada. Se trata de 30 puntos, 28 perteneciente a la clase 0 y 2 a la clase 1. La característica escogida sobre la que realizar la pregunta es 'EdadEncoded' y la pregunta en cuestión es "¿es EdadEncoded' menor o igual a 2.5?" ('EdadEncoded' ≤ 2.5). Esto vuelve a dividir el conjunto de datos en dos bloques. El bloque de la izquierda formado por aquellos puntos que responden positivamente a la pregunta anterior es un nodo terminal. La entropía se utiliza en las versiones ID3 y C4.5 del algoritmo, su entropía en éste nodo es 0, es una muestra completamente homogénea, de los 7 valores de la muestra, todos pertenecen a la clase 0 o vivos. Es un nodo hoja que representa una determinada clase o clase distribuida.

Ahora se crea el Modelo de Regresión Logística. Se procede a definir las variables de “X” y “y” que vamos emplear en el modelo. Cargando las variables de las 4 columnas de entrada en

X excluyendo la columna óbito. Se agregas la columna óbito en la variable. Se crea el modelo y se realiza el ajuste fit(X,y) al conjunto de entradas X y salidas 'y' como se ve en la Fig. 33 .

```
from sklearn import linear_model
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score

model = linear_model.LogisticRegression()
model.fit(X,y)
```

Fig. 33: Creación del modelo de regresión logísitca

Aquí se utiliza el modelo implementado en Sklearn (sklearn.linear_model.LogisticRegression()) para elaborar un modelo clasificador binario que utilice las variables de entrada X para predecir si el paciente está vivo o muerto. El método fit se encarga de ajustar los parámetros de regresión a los datos. Se confirma que tan bueno fue el modelo utilizando model.score() que nos devuelve la precisión media de las predicciones, en nuestro caso del 99.42% como se observa en la figura 34 .

```
predictions = model.predict(X)
print(predictions[25:35])
```

```
[0 0 0 0 0 0 0 0 0 0]
```

```
model.score(X,y)
```

```
0.9942528735632183
```

Fig. 34: Precisión media de las predicciones

Se procede a subdividir el conjunto de datos de entrada en un set de entrenamiento y otro para validar el modelo. En la Fig. 35 se muestra como se subdividen los datos de entrada en forma aleatoria, mezclados utilizando 80% de registros para entrenamiento y 20% para validar. Se vuelve a compilar el modelo de Regresión Logística pero esta vez sólo con 80% de los datos de entrada y se calcula el nuevo scoring que ahora da 99.46%.

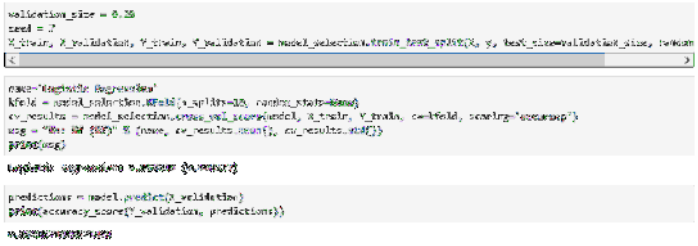


Fig. 35: Evalúa una puntuación mediante validación cruzada.

Al hacer las predicciones o clasificación, utilizando conjunto de validación cruzada como se observa en la Fig. 35, es decir del subconjunto que se había apartado. En este caso vemos que los aciertos fueron del 99.28%

4.8.1 Prueba de capacidad predictiva.

A continuación, para la prueba, se seleccionan 2 registros a priori del conjunto de datos, uno con el valor Óbito 1 y otro con valor 0. En la Tabla 5 se muestra un registro del dataset cuando el campo óbito es 1.

Tabla 5: Dataset óbito= 1.

Atributo	Valor
TotalInter	0
Edad	62
PacieSexo	0
UltPotasio	7
UltHemoglobina	9
UltGlucemia	438
UltHematocrito	28
UltCreatininaSerica	3
ECO_FEY	60

Al hacer la consulta, etiquetada como óbito = 1 con los siguientes datos, si el potasio es 7mmol/L, al ser >5mmol/L es del grupo 2, la edad es 62 años pertenece al grupo 3 entre 60 años y 80 años, la creatinina tiene valor 3mg/l pertenece al grupo 0 por ser <= 3.5mg/l, y el valor de ecografía FEY 60% es del grupo 1 por estar comprendida 50% entre 75%. Los datos que ingresan son 'edadEncoded', 'PotasioEncoded', 'CreatininaEncoded', 'ECO_FEYEncoded'.

En la Fig. 36 se observa que con estos valores de edad, potasio, creatinina, Ecografía FEY el modelo predice [0] Vivo, lo que es un error ya que óbito es 1.

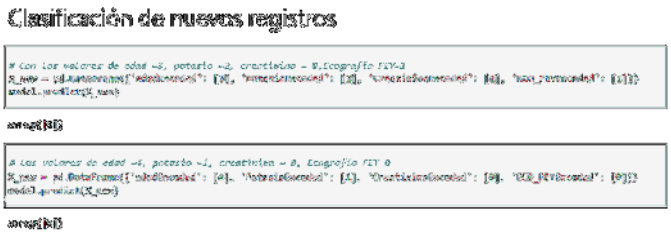


Fig. 36: Predicción de nuevos valores utilizando predict().

Para otro registro cuando el campo obito=0 y los valores de edad =4, potasio =1, creatinina = 0, ecografía FEY 0, se obtiene [0] Vivo, lo cual es un acierto.

Se puede ver la precisión con que se acertaron cada una de las clases, en el conjunto de datos de entrenamiento contamos con que una de las clases de muestra es una clase minoritaria, por lo que obtenemos un alto valor de precisión en la clase Mayoritaria y un bajo recall en la clase Minoritaria. En la Fig. 37 se muestran los valores de las métricas de precisión. Es interesante notar que en la columna de f1-score obtenemos buenos resultados, pero son engañosos, pues están reflejando una realidad parcial. Lo cierto es que el modelo no es capaz de detectar correctamente los casos de óbito 1.

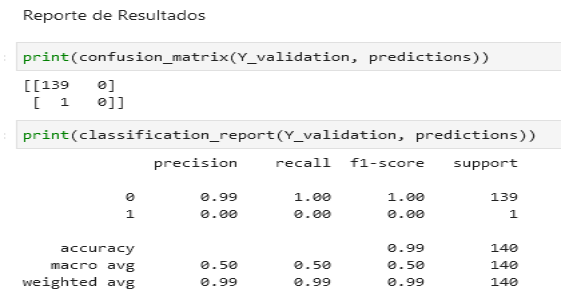


Fig. 37. Métricas de precisión del modelo

A continuación, se procede a tomar valor de las variables y recorrer el árbol. Con los valores de edad =3, potasio =2, creatinina = 0, Ecografía FEY=1 el sistema predice [0] Vivo con una probabilidad de acierto: [69.57]%. Ver el seguimiento en el gráfico de la Fig. 38.

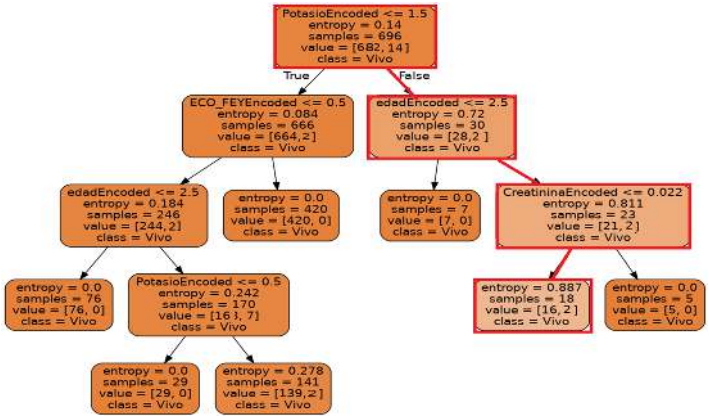


Fig. 38: Seguimiento en el árbol.

Dentro de este marco, tomando otro registro cuando el campo óbito=0 se obtiene los resultados que se ilustra en la Tabla 6.

Tabla 6: Dataset óbito= 0.

Atributo	**Valor**
TotalInter	2
Edad	87
PacieSexo	1
UltPotasio	4
UltHemoglobina	13
UltGlucemia	141
UltHematocrito	40
UltCreatininaSerica	1
ECO_FEY	38

El potasio es 4mEq/l al estar entre 3.5mEq/l y 5 mEq/l es del grupo 1, la edad 87 años pertenece al grupo 4**,** mayor a 80 años, la creatinina tiene valor 1mg/l pertenece al grupo 0 por ser <= 3.5mg/l, y el valor de ecografía FEY es 30% es del grupo 0 por estar <= 50%.

Los datos que ingresan son 'edadEncoded', 'PotasioEncoded', 'CreatininaEncoded', 'ECO_FEYEncoded'. Los valores de edad =4, potasio =1, creatinina = 0, ecografía FEY 0, el sistema predice: [0] Vivo con una probabilidad de Acierto: [100]%.

En esta fase se realizaron las entrevistas con el médico y los expertos del dominio, las evaluaciones correspondientes a las características los elementos de entrada y los elementos de salida generados a partir de los hallazgos encontrados y en función del de lo planteado las tareas de diagnóstico/pronóstico médico.

El especialista especificó las variables de confusión, estas son variables predictorias de la respuesta o efecto, externas a la relación principal que se analizan, no son un mero paso intermedio entre la exposición y la respuesta, y simultáneamente relacionadas con la variable

independiente. Por ejemplo, los glóbulos rojos relacionados con hematocrito y hemoglobina. Su presencia genera un sesgo o error al evaluar la relación entre las variables independientes(X) y dependiente (Y). Se intenta buscar un modelo en el que, con el menor número de variables posibles, independientes y de control, se genere una predicción más precisa y válida de la respuesta evaluada.

4.9 Revisión de la temática propuesta

Al utilizar la variable óbito como target, una parte importante de la información ha quedado fuera, ya que no se incluyen los fallecidos fuera de FUNCACOR. A través de la comunicación directa con quienes representan técnicamente a la entidad, quienes realizaron ponderaciones de evaluación de los contenidos de la propuesta, se identificó que los datos estadísticos de defunciones en la fundación son ambiguos. En el conjunto de datos no se consideraron las enfermedades o condiciones que forman parte de la secuencia de eventos que llevaron a la muerte del paciente. Ante lo expuesto, se decidió incluir otros objetivos en el estudio, por lo que se busca establecer reglas de negocio de como conceder una cama en UCIC a aquellos pacientes clasificados como de alto riesgo con un alto score, y derivar aquellos pacientes de menos cuidado a otro tipo de internación. Esto permite reducir significativamente la carga de trabajo de los médicos especialistas, y/o aumentar el volumen de datos procesados con la misma cantidad de personal.

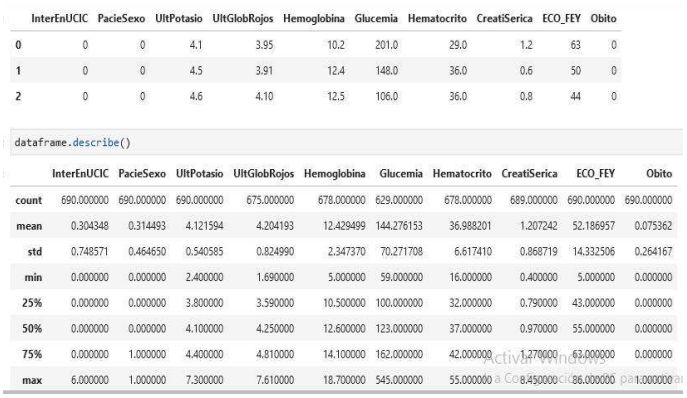
Se procede a realizar un análisis exploratorio de un nuevo dataset. Los datos de los pacientes se recogieron a partir de la historia clínica electrónica y de la intranet de FUNCACORR. A diferencia del dataset anterior, sólo se toma la última internación del paciente si tuviera más de una. El período de la muestra abarca un lapso de tiempo de un año. El análisis de los datos se realizó con algunas herramientas de estadística y con algunas visualizaciones para entender los datos disponibles como se hizo con el dataset anterior. Utilizando expresiones de Python, se procede a cargar la base de datos en un dataframe de Pandas; lo que va a permitir manipular los datos. En la Fig. 39 se ilustra la llamada a las librerías necesarias y la lectura del conjunto de datos.

```
import pandas as pd
import numpy as np

datos2 = pd.read_csv( r"Dat_Ultimo.csv",delimiter=';')
dataframe = datos2[['InterEnUCIC', 'PacieSexo' ,'UltPotasio','UltGlobRojos' ,'Hemoglobina' ,'Glucemia' ,'Hematocrito'
dataframe.head()
```

Fig. 39:Llamada a las librerías y lectura del dataset.

La función **head()** de Pandas se utiliza para obtener un resumen conciso del marco de datos. Útil para un análisis exploratorio de los mismos. En la Fig. 40 se muestra el resultado de utilizar head() en el dataset lo que provoca el volcado de los primeros 3 registros a modo de muestra.



	InterEnUCIC	PacieSexo	UltPotasio	UltGlobRojos	Hemoglobina	Glucemia	Hematocrito	CreatiSerica	ECO_FEY	Obito
0	0	0	4.1	3.95	10.2	201.0	29.0	1.2	63	0
1	0	0	4.5	3.91	12.4	148.0	36.0	0.6	50	0
2	0	0	4.6	4.10	12.5	106.0	36.0	0.8	44	0

```
dataframe.describe()
```

	InterEnUCIC	PacieSexo	UltPotasio	UltGlobRojos	Hemoglobina	Glucemia	Hematocrito	CreatiSerica	ECO_FEY	Obito
count	690.000000	690.000000	690.000000	675.000000	678.000000	629.000000	678.000000	689.000000	690.000000	690.000000
mean	0.304348	0.314493	4.121594	4.204193	12.429499	144.276153	36.988201	1.207242	52.186957	0.075362
std	0.748571	0.464650	0.540585	0.824990	2.347370	70.271708	6.617410	0.868719	14.332506	0.264167
min	0.000000	0.000000	2.400000	1.690000	5.000000	59.000000	16.000000	0.400000	5.000000	0.000000
25%	0.000000	0.000000	3.800000	3.590000	10.500000	100.000000	32.000000	0.790000	43.000000	0.000000
50%	0.000000	0.000000	4.100000	4.250000	12.600000	123.000000	37.000000	0.970000	55.000000	0.000000
75%	0.000000	1.000000	4.400000	4.810000	14.100000	162.000000	42.000000	1.270000	63.000000	0.000000
max	6.000000	1.000000	7.300000	7.610000	18.700000	545.000000	55.000000	8.450000	86.000000	1.000000

Fig. 40: Volcado de una muestra del dataset y sus valores estadísticos.

Esta información incluye el número de muestras 690, el valor medio, la desviación estándar, el valor mínimo, máximo, la mediana y los valores correspondientes a los percentiles 25% y 75% para las columnas. Lo más relevante es el número de instancias 690 y el número de atributos 10, se presentan en la Tabla 7: Atributos de la muestra.

Tabla 7: Atributos de la muestra.

Variable	**Descripción**
InterEnUCIC	Cantidad de veces que el paciente ha sido internado en La Unidad de Cuidados Intensivos Coronarios.
PacieSexo	Sexo del paciente 0 Mujer 1 Varón
UltPotasio	Último valor de laboratorio de Potasio.
UltGlobulosRojos	Último valor de laboratorio de Glóbulos Rojos.
Hemoglobina	Último valor de laboratorio de Hemoglobina
Glucemia	Último valor de laboratorio de Glucemia
Hematocrito	Último valor de laboratorio de Hematocrito
CreatininaSerica	Último valor de laboratorio de Creatinina Sérica
ECO_FEY	Valor de ecografía Fracción de eyección
Óbito	Defunción o fallecimiento de la persona

Con la función info() se obtiene examen general de los datos de cada columna. Indica el número de las mismas, tipo de dato de cada columna y tamaño total del dataset. Hay columnas con tipo float e Int. En la Fig. 41 se ilustra la aplicación de la función info() y las variables con sus tipos de datos. Los dos tipos que se presentan son el tipo real y tipo entero.

```
print(dataframe.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 690 entries, 0 to 689
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   InterEnUCIC   690 non-null    int64
 1   PacieSexo     690 non-null    int64
 2   UltPotasio    690 non-null    float64
 3   UltGlobRojos  675 non-null    float64
 4   Hemoglobina   678 non-null    float64
 5   Glucemia      629 non-null    float64
 6   Hematocrito   678 non-null    float64
 7   CreatiSerica  689 non-null    float64
 8   ECO_FEY       690 non-null    int64
 9   Obito         690 non-null    int64
dtypes: float64(6), int64(4)
memory usage: 54.0 KB
None
```

Fig. 41: La función info() sobre el dataset y sus tipos de datos.

Para las variables, se calcula la media muestral, el intervalo de confianza y desviación estándar para la muestra. En la Fig. 42 se ilustra la aplicación de las funciones mean(), std() y sum() entre otras usadas para cálculos estadísticos.

```
# Clasificación con Datos Desbalanceados
age_avg = dataframe['UltPotasio'].mean()
age_std = dataframe['UltPotasio'].std()
age_null_count = dataframe['UltPotasio'].isnull().sum()
age_null_random_list = np.random.randint(age_avg - age_std, age_avg + age_std, size=age_null_count)

conValoresNulos = np.isnan(dataframe['UltPotasio'])

dataframe.loc[np.isnan(dataframe['UltPotasio']), 'UltPotasio'] = age_null_random_list
dataframe['UltPotasio'] = dataframe['UltPotasio'].astype(int)
print("Potasio Promedio: " + str(age_avg))
print("Desvio Std Potasio: " + str(age_std))
print("Intervalo para asignar Potasio aleatoria: " + str(int(age_avg - age_std)) + " a " + str(int(age_avg + age_std)))
```

```
Potasio Promedio: 4.12159420289855
Desvio Std Potasio: 0.5405854032395461
Intervalo para asignar Potasio aleatoria: 3 a 4
```


Fig. 42: Aplicación de las funciones mean(), std() y sum().

Con un conjunto de datos procesados y explorados, es necesario crear una matriz de variables independientes y un vector de variables dependientes.

La interpretación de la correspondencia entre la variable dependiente (resultado) y el total de variables independientes (predictoras), aquí la tarea es predecir si un paciente debe ir a UCIC. Los resultados de cuantos pacientes en referencia a la cantidad de internaciones en UCIC se ilustran en la Fig. 43. Se observa que la cantidad máxima de internaciones en UCIC es 6, la mayoría de los valores son 0 (cero) internaciones contabilizando 548 casos. En esta situación es conveniente convertir el valor numérico a nominal, en un proceso de discretización.

```
InterEnUCIC
0    548
1    105
2     20
3      8
4      6
5      1
6      2
dtype: int64
```

Fig. 43: Cantidad de internaciones en UCIC.

Para visualizar los valores de cantidad de internaciones en UCIC, se creó el gráfico de barras de la Fig. 44.

```
# Visualizamos
sb.factorplot('InterEnUCIC',data=datos,kind="count")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f811e8c9b50>
```

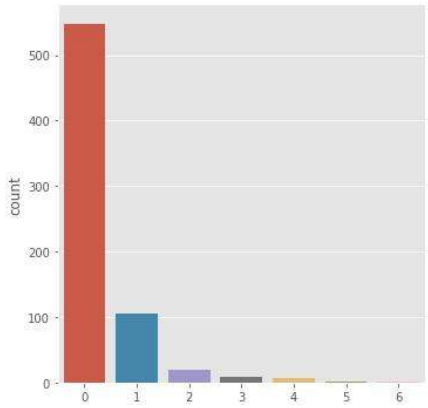


Fig. 44: Comparación del conjunto de datos utilizando Matplotlib.

Como se indicó en Capítulo II, Matplotlib es una librería de visualización de datos con Python. El color de cada punto, representa la cantidad de internaciones en UCIC. En la Fig. 45 se aprecia la ya antes comentada relación entre el potasio y la creatinina y su detalle en la Fig. 44.

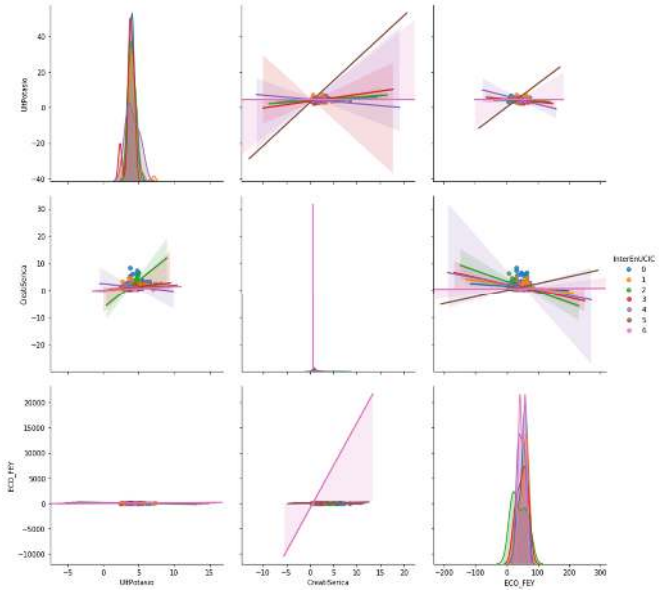


Fig. 45: Correlación entre las variables.

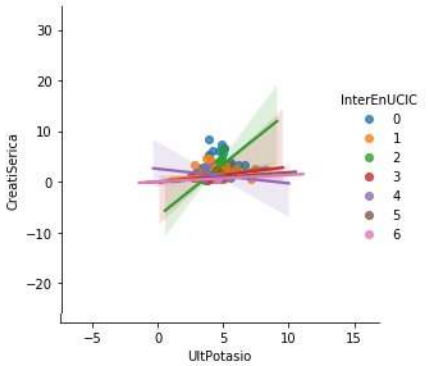


Fig. 46: Detalle del gráfico relación entre el potasio y la creatinina.

Cuando se crea un modelo, es importante estudiar la distribución de la variable respuesta, ya que es lo que interesa predecir, y las variables independientes, aquellas que determina el valor de la variable dependiente.

Se analizan gráficamente las variables independientes mediante histogramas, como se muestran en las figuras.

Veremos gráficamente los datos para tener una idea de la dispersión de los mismos. En las figuras Fig. 47 se ilustran los histogramas de las variables creatinina sérica y de la ecografía de la fracción de eyección.

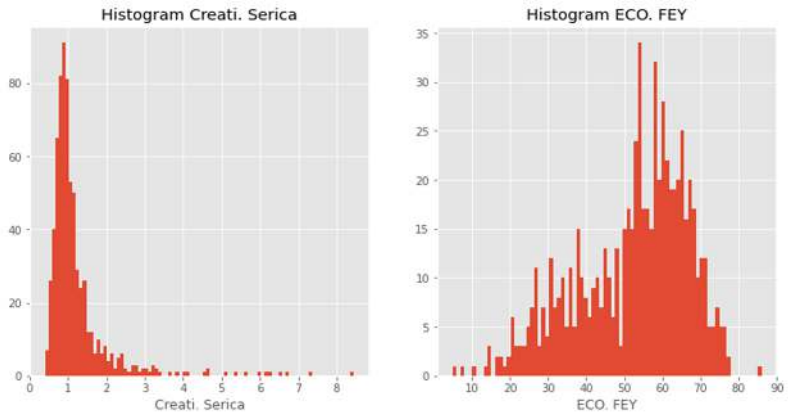


Fig. 47: Histograma de creatinina sérica y ecografía FEY.

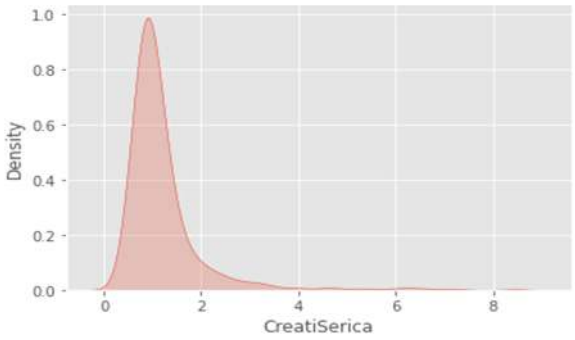


Fig. 48: Curva de densidad creatinina sérica.

La Fig. 48 se aprecia el pico del gráfico de densidad, muestra dónde los valores se concentran en el valor del intervalo 1,22 mg/l. El valor mínimo 0,4 mg/l y el máximo 7,3 mg/l. En la Fig. 49 se ilustra el histograma de la variable Potasio y en la Fig. 50 en la se observa datos estadísticos correspondientes la misma variable. El valor promedio 4.1215mmol/L coincide con el pico más alto del histograma. Estos datos, así presentados, se ponen a criterio del especialista Dr. Jorge Parras a quien el resumen le permite visualizar las características de la distribución de los datos reconociendo el nivel de precisión esperado.

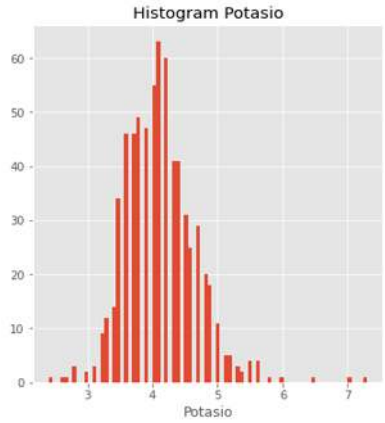


Fig. 49: Histograma de potasio.

```
Potasio Promedio: 4.12159420289855
Desvio Std Potasio: 0.5405854032395461
Intervalo para asignar Potasio aleatoria: 3 a 4
```

Fig. 50: Estadísticos para variable potasio.

Se procede a categorizar las internaciones en UCIC, en 2 grupos, aquellos que nunca estuvieron en UCIC 0 y otro grupo aquellos que estuvieron 1 o más veces en UCIC. En Fig. 51 ilustra el código en Python que separa el conjunto de datos en las dos categorías posibles.

```
# Categorizar InterEnUCIC 0  mas veces
dataframe.loc[ dataframe['InterEnUCIC'] == 0, 'InterEnUCICEncoded']    = 0
dataframe.loc[ dataframe['InterEnUCIC'] >= 1, 'InterEnUCICEncoded'] = 1
```

Fig. 51: Código Python para categorizar.

El total en cada categoría arroja 548 pacientes que nunca se internaron (0) y 142 pacientes que se internaron una vez o más. En Fig. 52 se ilustra la obtención de estos valores por medio de la función groupby().

```
dataframe=Pacientes_encoded
print(dataframe.groupby('InterEnUCICEncoded').size())
```

```
InterEnUCICEncoded
0.0    548
1.0    142
dtype: int64
```

Fig. 52: Uso de la función groupby().

Utilizando Scatter plots en la representación de correlaciones se puede ver cómo una variable afecta a la otra, los gráficos de scatter son muy útiles para analizar la correlación entre potasio, creatinina sérica y ecografía FEY.

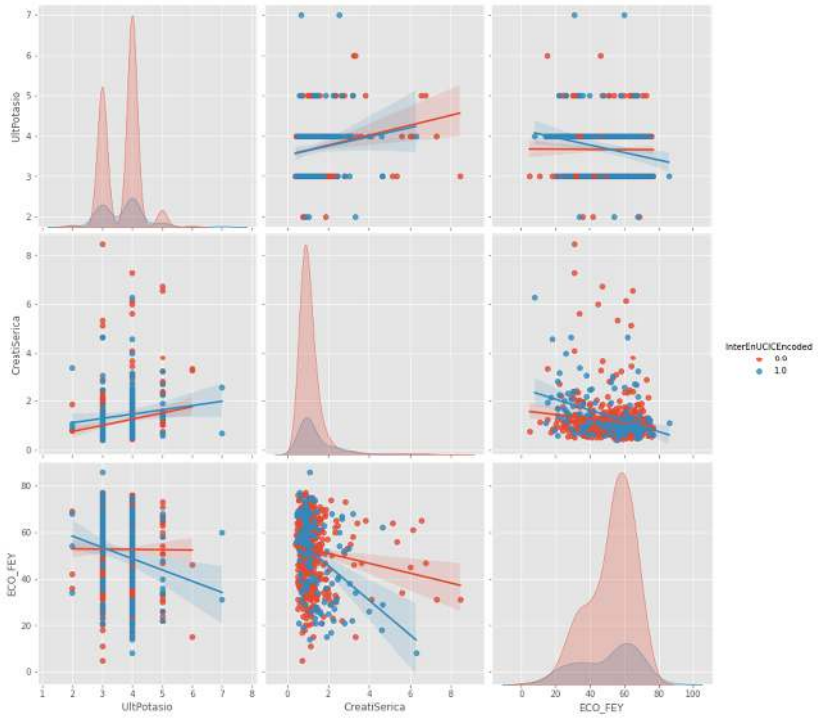


Fig. 53: Correlaciones entre las variables.

En Fig. 53 se ilustra las correlaciones entre las variables cuando las internaciones en UCIC han sido categorizadas.

Con esta información acerca de la distribución, se puede ir más allá de los datos de muestra sin procesar y hacer inferencias estadísticas e identificar a qué distribución se ajustan mejor los datos. Es apreciable la distancia que separa el cluster representado por los puntos rojos, con 0 internaciones, de los otros que se muestran más mezclados.

4.10 Regresión con Árboles de Decisión

Scikit Learn cuenta con un módulo en donde se incluye todo lo referente al algoritmo de árboles de decisión, se debe importar la clase que en este caso será DecisionTreeRegressor [40]. En Fig. 54 se ilustra el llamado a todas las librerías necesarias.

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import plot_tree
from sklearn.tree import export_graphviz
from sklearn.tree import export_text
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

Fig. 54: Inclusión de librerías de Scikit Learn.

Se construye un modelo utilizando este algoritmo, se procede a crear los parámetros "x" e "y" y llamar al método fit() para realizar el entrenamiento y posteriormente la instrucción predict() para realizar una predicción.

En al Fig. 55 se ilustra la creación del modelo y su ajuste.

```
# Creación del modelo
# ------------------------------------------------------------------------------
modelo = DecisionTreeRegressor(
            max_depth         = 5,
            random_state      = 123
           )

# Entrenamiento del modelo
# ------------------------------------------------------------------------------
modelo.fit(X_train, y_train)
```

Fig. 55: Creación del modelo de regresión.

Las configuraciones que ofrece el algoritmo DecisionTreeRegressor para mejorar el modelo a construir. En al Fig. 56 se ilustra los parámetros disponibles para la creación del modelo.

```
class sklearn.tree. DecisionTreeRegressor (criterion='mse', splitter='best', max_depth=None,
min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None,
max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort=False)  [source]
```

Fig. 56 Parámetros de la clase DecisionTreeRegressor.

Para configurar el modelo se puede especificar "criterion", Scikit Learn utiliza la media del error cuadrado o "mse" por sus siglas en inglés para implementar la separación de los datos. Por defecto se utiliza esta y es la única disponible, para la versión de Scikit Learn 0.18 se cuenta adicionalmente con el criterio de error absoluto promedio (MAE), pero se puede trabajar con la media del error cuadrado sin ningún problema.

La siguiente configuración es "splitter", que es la estrategia utilizada para la división en cada nodo. En este caso se cuenta con dos opciones "best" que sería la mejor división, y "random"

que elige de manera aleatoria la separación. Por defecto se encuentra seleccionada la opción “best” que por supuesto es la mejor opción.

Al no colocar ningún valor en “max_depth”, entonces el algoritmo selecciona de manera automática los nodos de manera que los expande hasta que todas las hojas estén puras o hasta que todas las hojas contengan menos datos. Si se hace esto es muy probable que el algoritmo caiga en sobreajuste, por lo que se probar variar el valor y ver el que mejor se adapte.

Como se indicó en Capítulo II [41], tres funciones de impureza son frecuentemente usadas: índice de Gini, la entropía y el error de clasificación ("misclassification"). Se procede a la creación y representación en las Fig. 57 y Fig. 58 de un árbol con profundidad 5.

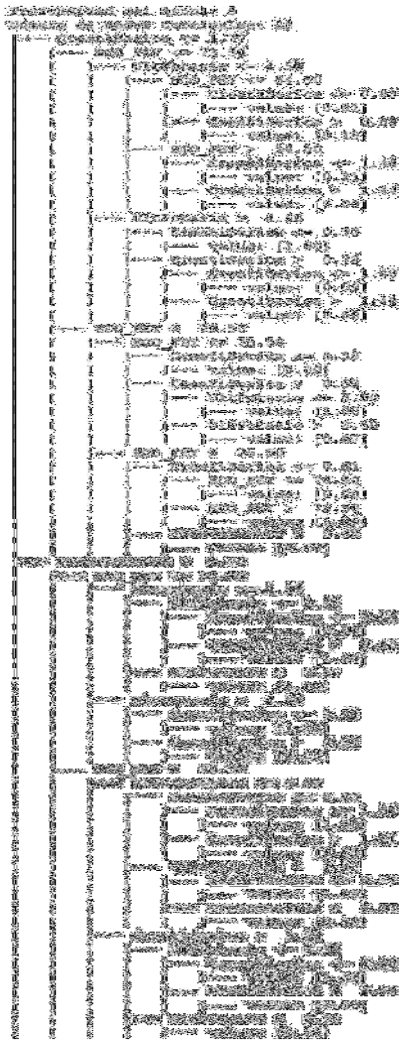


Fig. 57: Representación del árbol profundidad 5.

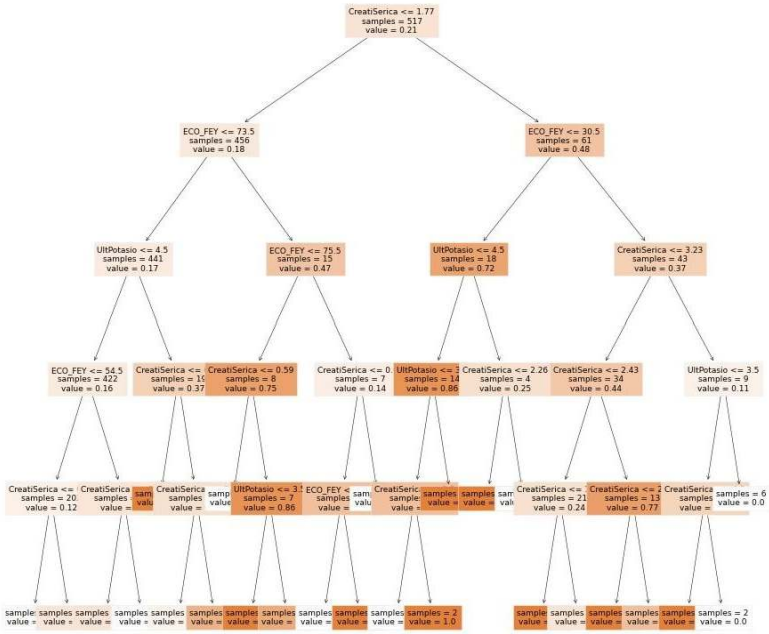


Fig. 58: Representación gráfica del árbol de profundidad 5.

Se procede a contabilizar la coincidencia entre las observaciones de test y los valores predichos en la matriz de confusión.

Como se indicó, probar variar el valor y ver el que mejor se adapte. En las Fig. 59 se crea un nuevo árbol con profundidad 6 y en la Fig. 60 se ilustra la generación de su correspondiente matriz de confusión.

```
# Crear Arbol de decision con profundidad = 6
decision_tree = tree.DecisionTreeClassifier(criterion='entropy',
                                            min_samples_split=20,
                                            min_samples_leaf=5,
                                            max_depth = 6,
                                            class_weight={1:3.5})
decision_tree.fit(x_train, y_train)
```

Fig. 59: Creación de árbol con profundidad 6 .

```
# Haciendo la Matriz de Confusión
from sklearn.metrics import confusion_matrix
print ('Confusion Matrix 1')

print(confusion_matrix(Y_validation, predictions))
```

```
Confusion Matrix 1
[[270   1]
 [ 73   1]]
```

Fig. 60: Creación de la matriz de confusión

Se evalúa la capacidad predictiva del árbol calculando el accuracy en el conjunto de datos de test. A continuación, se evalúa la proporción de las instancias predichas correctamente, estas

son TP y TN, sobre la suma total de elementos evaluados. Realizando el cálculo según la ecuación 5:

$$\text{Accuracy} = 270 + 1/ (270 + 1+ 73+ 1)=271/345 =$$

$$\text{Accuracy} = 0.78550724$$

Exactitud para la prueba =0.78550724

Para encontrar el balance entre la profundidad y complejidad del árbol con respecto a la capacidad predictiva del modelo en datos de test, normalmente se hace crecer el árbol de decisión hasta su mayor extensión y luego se ejecuta el proceso de poda para identificar el subárbol óptimo, en este caso se hace crecer la profundidad a 6. El árbol se representa en la Fig. 61.

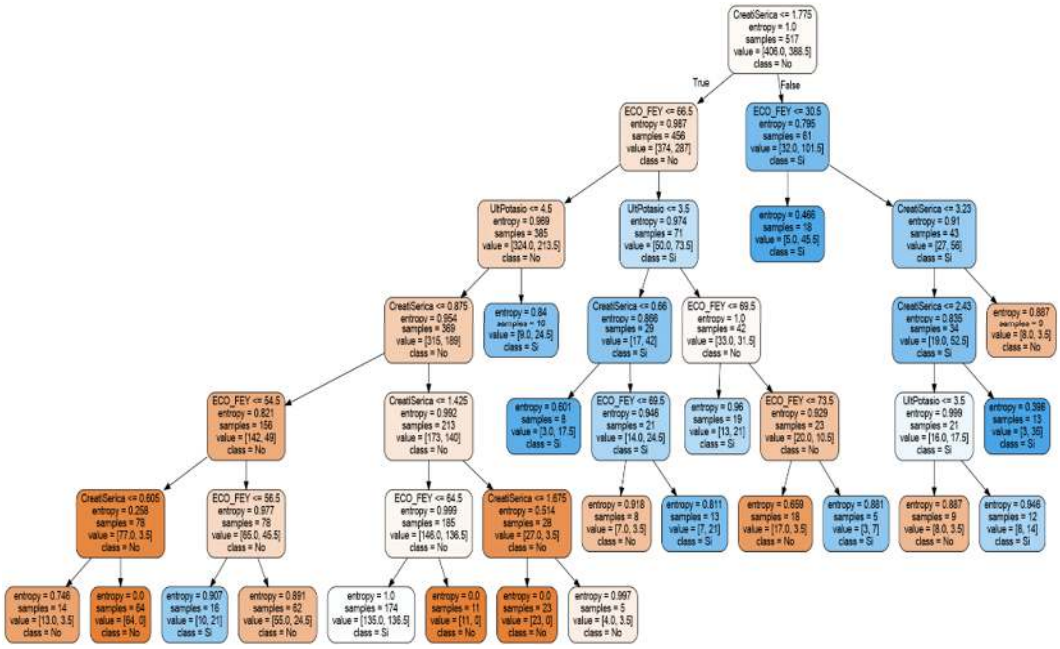


Fig. 61: Representación del árbol con profundidad 6.

La regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (InterEnUCICEncoded) en función de las variables independientes o predictoras. Aquí la regresión es utilizada para predecir el resultado de una variable categórica de internaciones en UCIC. Este método de regresión permite estimar la probabilidad de una variable cualitativa binaria, en este caso internado, que puede tomar los posibles valores No, SI en función de variables cuantitativas.

4.11 Ajuste del modelo por medio de fórmulas

La herramienta regresión exploratoria evalúa todos posibles combinaciones de posibles variables explicativas de entrada, buscando modelos de mínimos cuadrados ordinarios (Ordinary Least Squares OLS) que expliquen mejor la variable dependiente en el contexto del criterio especificado por el usuario. En la Fig. 62 muestra su implementación en Python.

```
# Reg = ols ('variable dependiente ~ variable independiente, marco de datos)
m = ols('InterEnUCICEncoded ~ CreatiSerica',dataframe).fit()
print (m.summary())
```

Fig. 62: Implementación de modelos de mínimos cuadrados ordinarios.

En la Fig. 63 se ilustra los valores devueltos por el modelo de mínimos cuadrados ordinarios tomando como variable dependiente a InterEnUCICEncoded.

OLS Regression Results

Dep. Variable:	InterEnUCICEncoded	R-squared:	0.012
Model:	OLS	Adj. R-squared:	0.010
Method:	Least Squares	F-statistic:	8.225
Date:	Fri, 19 Nov 2021	Prob (F-statistic):	0.00426
Time:	11:37:52	Log-Likelihood:	-350.08
No. Observations:	690	AIC:	704.2
Df Residuals:	688	BIC:	713.2
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.1446	0.026	5.509	0.000	0.093	0.196
CreatiSerica	0.0507	0.018	2.868	0.004	0.016	0.085

Omnibus:	139.689	Durbin-Watson:	2.164
Prob(Omnibus):	0.000	Jarque-Bera (JB):	234.915
Skew:	1.428	Prob(JB):	9.75e-52
Kurtosis:	3.131	Cond. No.	3.41

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Fig. 63: Valores de OLS para la variable InterEnUCICEncoded.

Se calcula el coeficiente de correlación, denominado R, es una forma de ver la influencia de cada una de las variables independientes por separado con la variable dependiente y el coeficiente de determinación, también conocido como R^2 (R cuadrado) para determinar qué tan bien la línea de regresión se ajusta a los datos.

Se toma la variable dependiente InterEnUCICEncoded utilizando el método Mínimos cuadrados, los valores R cuadrado múltiple y R cuadrado ajustado son medidas del rendimiento del modelo. Los valores posibles varían de 0,0 a 1,0. Para este modelo el valor de R cuadrado es 0.012 y R cuadrado ajustado 0.010. El valor R cuadrado ajustado siempre es un poco más bajo que el valor R cuadrado múltiple, porque refleja la complejidad del modelo (la cantidad de variables) ya que se relaciona con los datos y es, por lo tanto, una medida más exacta del rendimiento del modelo. Un valor R cuadrado ajustado de 0.012 indicará que el modelo explica aproximadamente el 1 por ciento de la variación en la variable dependiente. Dicho de otra manera, el modelo cuenta aproximadamente el 1%de casos de Internaciones en UCIC.

A continuación, se toman las variables UltPotasio, CreatiSerica y ECO_FEY en la estimación **ols**, lo que se muestra en la Fig. 64.

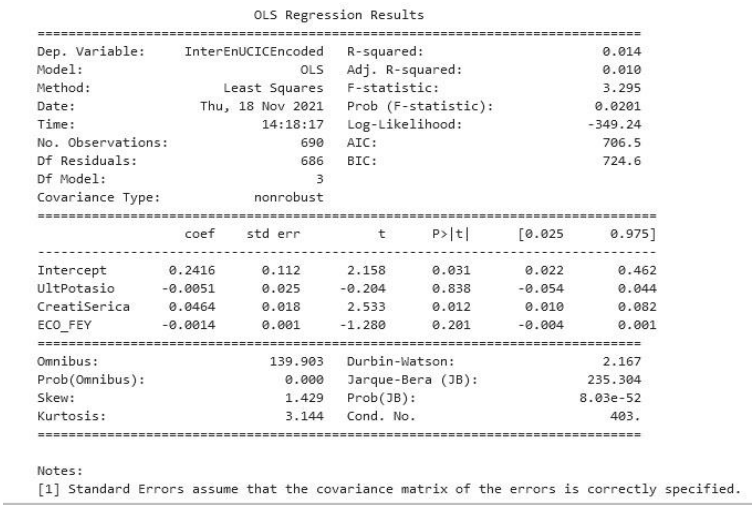
```
m = ols('InterEnUCICEncoded ~ UltPotasio + CreatiSerica + ECO_FEY',dataframe).fit()
print (m.summary())
```

```
                            OLS Regression Results
```

Fig. 64: Estimación Mínimos Cuadrados Ordinarios.

La función para la estimación ols tiene el esquema OLS (respuesta, entrada) La distribución t de Student describe cómo se espera que se comporte la media de una muestra con un cierto número de observaciones (su n). Se utiliza t para probar si el coeficiente correspondiente es diferente de 0. (Hipótesis H0: coef == 0, H1: coef ! = 0)

Pr> | t | es el valor p para esta prueba de hipótesis. Un valor bajo de p significa que puede rechazar la hipótesis nula y aceptar la hipótesis alternativa (coef! = 0). Los valores que se obtuvieron se muestran en la Fig. 65.



OLS Regression Results

Dep. Variable:	InterEnUCICEncoded	R-squared:	0.014
Model:	OLS	Adj. R-squared:	0.010
Method:	Least Squares	F-statistic:	3.295
Date:	Thu, 18 Nov 2021	Prob (F-statistic):	0.0201
Time:	14:18:17	Log-Likelihood:	-349.24
No. Observations:	690	AIC:	706.5
Df Residuals:	686	BIC:	724.6
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.2416	0.112	2.158	0.031	0.022	0.462
UltPotasio	-0.0051	0.025	-0.204	0.838	-0.054	0.044
CreatiSerica	0.0464	0.018	2.533	0.012	0.010	0.082
ECO_FEY	-0.0014	0.001	-1.280	0.201	-0.004	0.001

Omnibus:	139.903	Durbin-Watson:	2.167
Prob(Omnibus):	0.000	Jarque-Bera (JB):	235.304
Skew:	1.429	Prob(JB):	8.03e-52
Kurtosis:	3.144	Cond. No.	403.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Fig. 65: Resultados de estimación ols para UltPotasio, CreatiSerica y ECO_FEY.

El estadístico t es el coeficiente dividido por su error estándar. Por ejemplo, para potasio su coeficientes -0.0051 y su error estándar es 0.025 lo que nos da t= -0.0051/0.025 = -0.204

El error estándar es una estimación de la desviación estándar del coeficiente. Se puede considerar como una medida de la precisión con la que se mide el coeficiente de regresión.

El valor *P* dice qué tan seguro se puede estar de que cada variable individual tiene alguna correlación con la variable dependiente.

En Regresión Logística (RL) lo que se pretende es estimar los parámetros de la ecuación (β0, β1, β2,...βk) de la función visto en la ecuación Nr. 2, para este caso los valores son los que se muestran en la Fig. 66.

```
β estimados:
Intercept       0.241634
UltPotasio     -0.005071
CreatiSerica    0.046359
ECO_FEY        -0.001403
dtype: float64
```

Fig. 66: Parámetros de la ecuación.

Z = 0.241634 -0.005071 x UltPotasio + 0.046359 x CreatiSerica -0.001403 x ECO_FEY

Al realizar este estudio de investigación analítico, se pasa por una reflexión de qué variables incluir en el trabajo, esto no pasa exclusivamente por un proceso de elección estadística, ya que esta técnica no distingue entre asociaciones de índole causal y las debidas a otros factores, incluso a las debidas a sesgos en el estudio.

4.12 Construcción del modelo

Para crear el modelo predictivo se hace uso de la librería Sklearn para Machine Learning. Para la regresión lineal, tenemos los siguientes hiperparámetros [42]:

fit_intercept: True/False, para quedar o quitar la constante β₀ de nuestro modelo. Si la quitamos la recta pasará obligatoriamente por el punto 0 del eje de abscisas.

normalize: True/False, para normalizar los datos o no, normalmente la regresión lineal suele funcionar mejor con datos normalizados, para que todas las variables estén a la misma escala.

En Python Scikit-Learn implementa la regresión logística en la clase sklearn.linear_model.LogisticRegression. Esta implementación permite añadir el argumento solver que determina el algoritmo de optimización a usar ("newton-cg", "lbfgs", "liblinear", "sag" o "saga") y, en el caso lbfgs. La Fig. 67 ilustra la aplicación de **lbfgs** en la clase.

Los hiperparámetros deben estar bien configurados para obtener un rendimiento óptimo. Entonces, optimizar los hiperparámetros de un modelo es una tarea crucial para aumentar el rendimiento del algoritmo seleccionado.

```
model = linear_model.LogisticRegression(solver='lbfgs')
model.fit(X,y)
```

```
LogisticRegression()
```

Fig. 67: Uso de lbfgs en LogisticRegression.

Una vez que el modelo de clasificador se entrenó con los datos, puede realizar predicciones sobre los datos del conjunto de prueba. Esto se hace llamando al comando *predict* en el clasificador y proporcionándole los parámetros que necesita para hacer predicciones, que son las características de su conjunto de datos de prueba.

El modelo ha sido entrenado y ahora puede ser usado para predecir nuevas instancias en el conjunto de datos de test, lo que se muestra en la Fig. 68. Para esto se usa la función predict().

```
# Relaciona Input con Output

# conectar predicciones con salidas
for i in range(50,80):
    print("i:",i, X[i], predictions[i])
```

```
i: 50 [ 4.    5.61 34.  ] 0.0
i: 51 [ 3.    0.93 50.  ] 0.0
i: 52 [ 3.    1.16 53.  ] 0.0
i: 53 [ 4.    3.43 64.  ] 1.0
i: 54 [ 3.    1.15 32.  ] 0.0
i: 55 [ 4.    2.37 30.  ] 0.0
i: 56 [ 3.    0.88 72.  ] 0.0
i: 57 [ 5.    1.41 31.  ] 0.0
i: 58 [ 4.    1.47 61.  ] 0.0
i: 59 [ 3.    0.88 66.  ] 0.0
i: 60 [ 4.    0.99 65.  ] 0.0
i: 61 [ 4.    0.67 71.  ] 0.0
```

Fig. 68: Valores devueltos para el conjunto de testeo

Se espera que el modelo predictivo elaborado a partir de estos datos sirva de guía al especialista del centro para tomar una decisión sobre la internación en UCIC, un paciente con: Potasio=4.9mmol/L, Creati.Serica= 0.4 mg/dL y Ecografía FEY= 37.0% NO Debería ir a UCIC. La Fig. 69 ilustra la aplicación de estos valores al modelo y el resultado devuelto [0] lo que indica que no debe ir a UCIC lo que es correcto.

```
# Relaciona Input con Output
# define input
new_input = [[4.9 ,  0.94,  37.0]]
# get prediction for new input
new_output = model.predict(new_input)
print("Pruebo:",new_input, new_output)
```

```
Pruebo: [[4.9, 0.94, 37.0]] [0.]
```

```
model.score(X,y)
```

```
0.7927536231884058
```

Fig. 69: Prueba del modelo para resultado devuelto [0].

A continuación, se prueba tomar una decisión sobre la internación en UCIC, un paciente con: potasio=3.0mmol/L, Creati.Serica =8.5mg/dL y ecografía FEY=31.0% el que SI debería ir a UCIC. La Fig. 70 ilustra la aplicación del caso.

```
# predigo
# define input
new_input = [[3.0 ,  8.5, 31.0 ]]
# get prediction for new input
new_output = model.predict(new_input)
print("Pruebo:",new_input, new_output)
```

```
Pruebo: [[3.0, 8.5, 31.0]] [1.]
```

Fig. 70: Prueba del modelo donde SI debe ir a UCIC.

El resultado devuelto [1] lo que indica que si debe ir a UCIC. Esto representa un acierto del modelo.

Para éste modelo, su matriz de confusión es la que se ilustra en la Fig. 71.

```
print(confusion_matrix(Y_validation, predictions))
```

```
[[270   1]
 [ 73   1]]
```

Fig. 71: Matriz de confusión.

Se evalúa la capacidad predictiva del modelo calculando F1-score, que tiene en cuenta la precisión y recall en el conjunto de datos de test. Los valores son ilustrados en la Fig. 72.

```
# Reporte de clasificación
print(classification_report(Y_validation, predictions))
```

	precision	recall	f1-score	support
0.0	0.79	1.00	0.88	271
1.0	0.50	0.01	0.03	74
accuracy			0.79	345
macro avg	0.64	0.50	0.45	345
weighted avg	0.73	0.79	0.70	345

Fig. 72: Capacidad predictiva del modelo

Las pruebas y los resultados son presentados y discutidos con los especialistas del centro. Ante la inexistencia de una herramienta automatizada que contribuya a establecer un diagnóstico rápido y eficaz sobre establecer y tratar algunas enfermedades cardiacas para su internación en UCIC, investigadores del Instituto CUCAICOR proponen la creación un sistema informático capaz de brindar un diagnóstico en un promedio de 3 minutos y con un índice alto de certeza.

4.13 Diseño Metodológico

Para este proyecto se propone un DSS activo, el que mostrará escenarios para mejorar la toma de decisiones; dirigido por modelos, se ajustará a modelos estadísticos; y dirigidos por datos ya

que los mismos son necesarios para definir los modelos iniciales[50]. Para poder realizar y cumplir los objetivos planteados para la realización del prototipo de software vamos a utilizar es el CICLO DE VIDA DE CREACIÓN DE PROTOTIPOS el cual tendrá avances dependiendo de las diferentes diligencias que lo rodean como se presentó en el Capítulo II.

4.13.1 Recolección y refinamiento de requisitos

Aumentar el conocimiento en cuanto a la interacción hombre-máquina: Algoritmo de inteligencia artificial. árboles de decisión con Python: regresión y clasificación, sistemas de computación basados en conocimiento, motor de base de datos en sqlite3, lenguaje de programación Python.

Después de terminada la primera instancia se intentará solucionar todas las dudas referentes a los temas investigados y paralelamente el estudio de las técnicas y métodos utilizados para la realización de aplicaciones basadas en sistema de conocimientos y habilidades para mejorar las capacidades científicas, investigativas y operativas.

4.13.2 Análisis de los requisitos del prototipo

En esta etapa del ciclo de vida el proceso de reunir y analizar los requisitos se hace primordial para de esta forma proceder a definir la funcionalidad del prototipo. Algunos de los requisitos se desprenden de la charla con los especialistas del centro.

4.13.3 Diseño rápido del prototipo

El diseño de la etapa anterior debe ser traducido a una forma entendible para la máquina, por esta razón se utilizará un lenguaje de programación como Python por las características y ventajas ya enumeradas en Capítulo II en la sección 2.4.3. Se trata de un modelo, evolucionario; el cual es un modelo parcialmente construido que puede pasar de ser prototipo a ser software, pero no tiene una buena documentación y calidad.

4.13.4 Elaboración del código

El lenguaje de programación será Python, ya que permite el desarrollo de interfaz gráfica. Se puede interactuar con la base de datos Sqlite3 que es muy adecuada para el desarrollo básico y pruebas, es portátil y utiliza sintaxis SQL estándar con pequeñas modificaciones.

4.13.5 Refinamiento del prototipo

Terminada la implementación del código se comenzará con las pruebas para el prototipo. Para el prototipo se realizarán pruebas sobre un motor de base datos específico como es sqlite3.

4.14 Marco conceptual

Se plantea, la necesidad no sólo de dotar de infraestructura y equipamiento complejos a las unidades para atender correctamente a los pacientes con enfermedad coronaria, sino también la obligación de gestionar con eficiencia unos recursos escasos y costosos. Los sistemas complejos deben ser estudiados como múltiples componentes no lineales, emergentes y de comportamiento dinámico que interactúan entre ellos. En esta propuesta se busca integrar aspectos tecnológicos y sociales cómo las características de los usuarios y de la organización, formas de trabajo y comunicación, etc.

Estas dimensiones o aspectos son:

- ✓ Hardware, software e Infraestructura: Esencialmente técnica. Son los equipos y aplicaciones utilizados para interactuar con el sistema.
- ✓ Contenido clínico: Todos los tipos de datos e información del tipo texto, datos numéricos, imágenes, señales biológicas, etc., las que constituyen el lenguaje de las aplicaciones clínicas.
- ✓ Interfaz humano-computadora: Aspectos de la computadora que los usuarios puedan ver, tocar o escuchar, ya que interactúan con él.
- ✓ Gente: Representa a los humanos, todos los que interactúan de alguna manera con el sistema, (desde desarrolladores hasta el usuario final, incluyendo a los especialistas médicos.
- ✓ Comunicación y procesos: Entiende que el cuidado continuado se logra a través del trabajo en equipo, y para eso es necesaria la comunicación. También analiza los procesos asistenciales, de forma que los sistemas los representen correctamente.
- ✓ Características organizacionales y políticas internas: Políticas, procedimientos y cultura de la organización.
- ✓ Regulaciones: Normativas y reglamentos externos, los cuales pueden facilitar o limitar muchos aspectos de las dimensiones anteriores.
- ✓ Medición y monitoreo: Evaluación de consecuencias, tanto intencionales como no intencionales de aplicación de TIC y uso.

4.14.1 Interfaz humano-computadora

Al iniciar la etapa de desarrollo del software de la capa clínica, el aplicativo fue diseñado teniendo en cuenta los procesos asistenciales de cada nivel de atención, entendiendo que el

repositorio de datos clínicos debía ser único. Inicialmente se desarrolló la interfaz para el especialista en el tratamiento de los pacientes.

4.14.2 Propiedades del desarrollo del prototipo de software

En este proyecto se desarrolló una herramienta para la predicción. Se propone crear un prototipo operativo con el que los usuarios pudieran interaccionar, debería simular el aspecto y las impresiones que producirían la interfaz en planificación, el prototipo puede servir como base para especificaciones operacionales. El análisis predictivo es una forma de análisis avanzado que utiliza datos nuevos e históricos para pronosticar la necesidad de internación en UCIC. Esta herramienta se puede utilizar como instrumento de trabajo en valoraciones clínicas.

El software de atención médica sirve de apoyo a la decisión médica que ofrece asistencia, orientación en el entorno de la atención al Instituto de Cardiología de Corrientes.

Para ejecutar el código se tiene instalado Python -versión 3.6- y varios paquetes usados comúnmente en DataScience. La llamada a los mismos se ilustra en la Fig. 73

```
#Librerias de IA
from sklearn import linear_model
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import plot_tree
from sklearn.tree import export_graphviz
from sklearn.tree import export_text
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
import os
```

Fig. 73: Bloque en Python de importación de librerías.

El diseño de la interfaz gráfica de usuario de interacción persona-computadora, fue construido con kit de herramientas GUI Tk estándar de Python. La Fig. 74 se ilustra la llamada a las librerías.

```
#Se importan librerias de trabajo
# apagaremos los warnings
import warnings; warnings.simplefilter('ignore')

import numpy as np
import tkinter as tk
from tkinter import ttk
import tkinter.font as font
from PIL import ImageTk, Image
from tkinter import Menu
from tkinter import messagebox
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
from matplotlib.figure import Figure
import numpy as np
import pandas as pd
```

Fig. 74: Inclusión del paquete tkinter («interfaz Tk»)

Este proyecto está orientado a agilizar el proceso de apoyo a la decisión médica sobre internación en UCIC. Esta herramienta está pensada para usuarios tanto del campo de la informática como de la medicina.

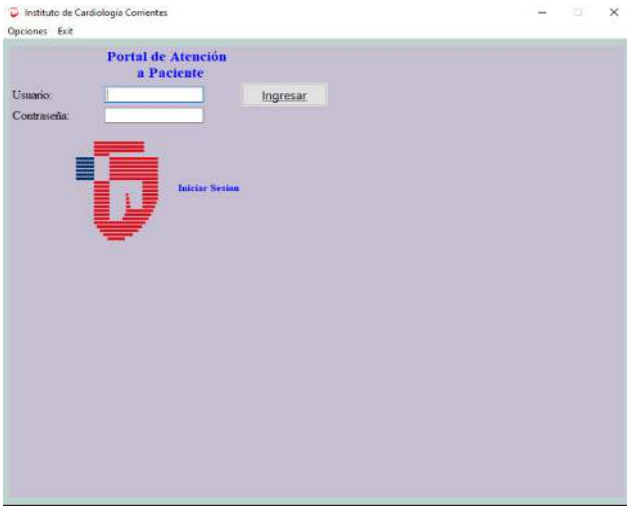


Fig. 75: Portal de Atención al paciente.

4.14.3 Seguridad de datos

Lo esencial es que el sistema permite restringir los accesos para que únicamente personal de salud autorizado pueda consultar la información. La nómina del personal autorizado está en la base de datos. Este acompaña al programa y que contiene todos los elementos de la misma, como son las tablas, índices o los datos. El acceso es muy rápido. En la Fig. 76 muestra la llamada a la librería de SQLite en Python.

```
#Librerias de SQL
import sqlite3
```

Fig. 76: Integración a Python usando el módulo sqlite3

El sisma debe tener la capacidad de rastrear la hora, fecha y lugar de consulta y solicitar la autenticación de quien accede. Se propone y desarrolla un prototipo del módulo que permite autenticar el nombre de usuario y la contraseña proporcionados, la que se muestra en la

Fig. 77.

Fig. 77: Interface del Sistema.

Parte del código del módulo de autenticación se muestra en Fig. 78.

```
def login(self, *args):
    # Conección y acceso a BBDD
    db = sqlite3.connect('bd_ICC.db3')
    c = db.cursor()

    usuario = self.E_1.get()
    contr = self.E_2.get()
    c.execute('SELECT * FROM usuarios WHERE usuario = ? AND pass = ?', (usuario, contr))
    elUsuario=c.fetchall()
    miApellidoNombre= ""
    for usuario in elUsuario:
        miApellidoNombre=usuario[3]
    if elUsuario:
        # Acceso a BBDD  con éxito
        messagebox.showinfo(title = "Login correcto", message = "Usuario y contraseña correctos")
        self.L_4.configure( text = "Bienvenido : " + miApellidoNombre )
        self.entrada_usuario = ""
        self.entrada_clave = ""
        self.B_2.grid(row = 5, column = 2, sticky = "e")
        self.B_1.grid_forget()
    else:
        # Acceso fallido a BBDD
        messagebox.showerror(title = "Login incorrecto", message = "Usuario o contraseña incorrecta")
    # Desconección y cierre de acceso a BBDD
    c.close()
```

Fig. 78: Módulo de autenticación.

Cuando un usuario accede al sistema, es la primera interfaz que se le muestra. El software permite que una vez completado el primer módulo, en donde se realiza su identificación o inicio de sesión, el usuario pueda navegar por todos los demás módulos según sus preferencias de selección. La Fig. 79 es una captura de pantalla donde se realiza la identificación o inicio de sesión, la misma solicita el nombre de usuario y clave para el ingreso.

Fig. 79: Respuesta de identificación fallida.

El programa se reparte en módulos: Gráfico de dispersión, predicción, configuración, visualización de datos, etc. Cada módulo tiene una funcionalidad específica. Una de las características relevantes de la propuesta es el desarrollo del concepto de módulos que pueden incorporarse como herramientas del programa en función de su disponibilidad y de las preferencias del usuario.

La estructura del programa se realizó sobre la base de una revisión exhaustiva y un estudio experimental también exhaustivo, desarrollado en este trabajo en las secciones anteriores.

El efecto funcional que pretenden esta solución técnica es que el análisis predictivo provea una puntuación o probabilidad para cada paciente, para dar apoyo en las decisiones tomadas en el centro de salud, indicando valores de potasio, creatinina sérica y ecografía de fracción de eyección como se ilustra en la Fig. 80.

Fig. 80: Interfaz de carga de valores para consulta.

Con este módulo se recoge información del paciente que puede ser utilizada de ayuda por el especialista para realizar el examen o la valoración, para conocer la probabilidad de que sea destinado a UCIC en un futuro. En Fig. 81 se ilustra el resultado de la prueba del modelo donde un paciente con: potasio=3.0 mmol/L, creatinina sérica=8.5mg/dL y ecografía FEY=31.0% SI DEBERÍA IR A UCIC.

Fig. 81: Resultado de una predicción de caso.

Se aplica la regresión logística**,** la creación de modelos de clasificación binaria. Se lee el archivo csv, el cual, por sencillez, se considera que estará en el mismo directorio que el archivo de Python. Se asigna mediante Pandas a la variable dataframe. Creamos el modelo y hacemos que se ajuste al conjunto de entradas X y salidas 'y'. Una vez compilado el modelo, le hacemos clasificar nuestro conjunto de entradas X (potasio, creatina sérica y ecografía FEY) utilizando el método "predict(X)" y revisamos la salida. Lo que podemos ver en la Fig. 82.

```
def Predecir(self, *args):
    dataframe = pd.read_csv( r"Ult_dataEncode.csv",delimiter='\t')

    # Creamos el Modelo de Regresión Logística
    X = np.array(dataframe.drop(['InterEnUCICEncoded'],1))
    y = np.array(dataframe['InterEnUCICEncoded'])
    # Completando valores faltantas datos cuantititavos
    for c in cnum:
        mean = dataframe[c].mean()
        dataframe[c] = dataframe[c].fillna(mean)
    # Controlando que no hayan valores faltantes
    dataframe.isnull().any().any()
    # Tomamos los valores
    v_Potasio=float(self.entrada_Potasio.get())
    v_Creatinina=float(self.entrada_Creatinina.get() )
    v_EcoFEY=float( self.entrada_EcoFEY.get()  )
    # Creamos el Modelo de Regresión Logística
    X = np.array(dataframe.drop(['InterEnUCICEncoded'],1))
    y = np.array(dataframe['InterEnUCICEncoded'])
    model = linear_model.LogisticRegression(solver='lbfgs')
    model.fit(X,y) #Entrenamos
    # Relaciona Input con Output
    # define input 4.9 ,  0.94,  37.0
    new_input = [[v_Potasio ,  v_Creatinina,  v_EcoFEY]]
    # Hace la predicción
    new_output = model.predict(new_input)
    V_Ley=""
    self.L_3.configure( text = "" )
    if (new_output == 0):
        V_Ley=" NO "
    else:
        V_Ley=" SI "

    #Muestra resultados
    self.L_3.configure( text = "Resultado Si/No: {}.".format( V_Ley) )
    v_score =model.score(X,y)
```

Fig. 82: Módulo de predicción.

La visualización de los datos se hace a través de tablas, gráficos y clasificaciones que muestran visualmente las clasificaciones más probables obtenidas de las predicciones.

Es un prototipo de la aplicación informática que sirve de ayuda específica en la valoración del estado del paciente. Podemos observar en la Fig. 83, la gráfica de dispersión de los datos, se observa cómo se confrontan las variables de una consulta con las de distribución del conjunto.

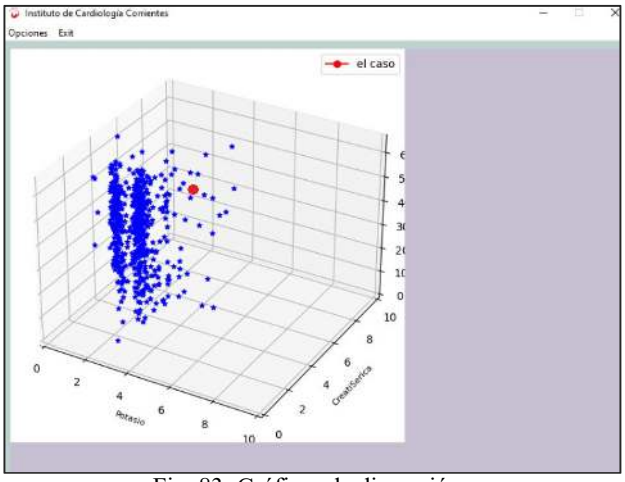


Fig. 83: Gráficas de dispersión.

El gráfico de dispersión muestra información adicional, tiene la capacidad para mostrar las relaciones no lineales entre las variables potasio, creatina sérica y ecografía FEY. Facilita explorar las relaciones potenciales entre las tres variables y el caso de estudio. La creación y representación gráfica se logra con las librerías las cuales se muestran son llamadas en el inicio del programa. En la Fig. 84 muestra la llamada a las librerías necesarias.

```
# Importa Grafica
# Permite la generación de gráficos
from matplotlib import pyplot
# Permite agregar eje tridimensionales
from mpl_toolkits.mplot3d import Axes3D
# Permiten obtener de distintos modos números aleatorios
import random
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
# Visualizacion
import plotly
```

Fig. 84: Importar librerías gráficas.

La portada, independientemente de la maquetación concreta, incluye los siguientes elementos: Nombre del proyecto, versión, desarrollado por y desarrollado para o nombre del cliente. En Fig. 85 se ilustra la interfaz de portada del prototipo.

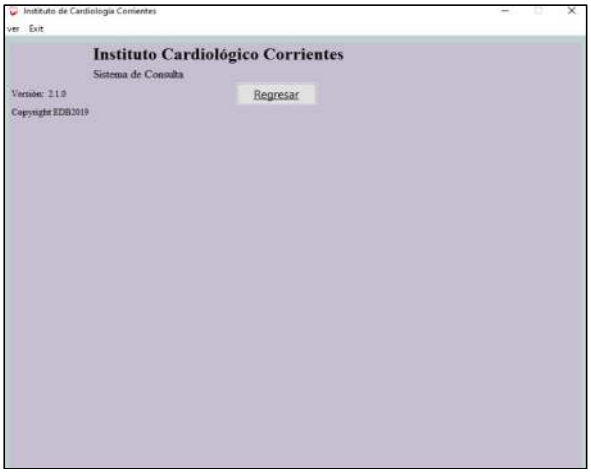


Fig. 85: Portada de versionado del prototipo.

- ✓ El modelo de prototipo utilizado, pertenece a los modelos de desarrollo evolutivo. Las etapas que involucró fueron:
- ✓ Comunicación.
- ✓ Analogía.
- ✓ Plan rápido.
- ✓ Modelado, diseño rápido.
- ✓ Construcción del Prototipo.
- ✓ Desarrollo, entrega y retroalimentación.

En el proceso de desarrollo de este prototipo, se tuvo que iterar, lo que se entiende como volver a especificar, rediseñar, volver a evaluar, hasta que el equipo, estuvo de acuerdo en que la fidelidad y el nivel de acabado del prototipo en evolución es lo suficientemente alto. La valoración que aquí se tuvo en cuenta es la de F1-score, que tiene en cuenta la precisión y recall, cálculo ya desarrollado en párrafos anteriores. Presentado el prototipo al Dr. Parras y realizando pruebas del mismo, el especialista afirmó que "*el modelo tiene un comportamiento muy lógico, y está muy bueno que quede en el gráfico expuesto… las variables de predicción son con las que trabajamos, que hay algunos sistemas de internet que piden de ocho a diez valores, lo malo que tiene eso es buscar esos datos lleva mucho tiempo y se trabaja o estima con datos de otra población"*

Capítulo V

Conclusiones y recomendaciones

5 Conclusión y Futuras Líneas de Investigación

En este capítulo se presentan las principales conclusiones obtenidas tras la realización de la propuesta. Además, se aportan algunas posibles líneas de investigación futuras que pueden enriquecer el modelo desarrollado.

5.1 Conclusión

Una de las conclusiones más importantes que surgió de este trabajo es la importancia del uso de la IA en el campo médico, mediante la aplicación de algoritmos de aprendizaje automático que permiten técnicas mejoradas de clasificación e interpretación de datos, predicción de enfermedades, ahorro de tiempo y recursos disponibles.

En la Tabla 8 muestra y sintetiza las distintas fases del proceso de Machine Learning y las tareas realizadas en este trabajo.

Tabla 8: Síntesis las principales fases.

Fase	Tarea	Detalle
Entender los datos	Análisis exploratorio de datos	Se hicieron gráficos y estadísticas descriptivas para comprender mejor a los Datos. Ayudó a Estimar si los datos son suficientes, y relevantes, para construir un modelo.
Definir un Criterio de Evaluación	Se utilizó el error cuadrático medio así como métricas de precisión, recall y F1	Se proponen casos de estudio que servirán para entender y evaluar el modelo propuesto. Se compara un valor predicho y un valor observado o conocido.
Evaluación de la solución actual	Medición del rendimiento de la solución actual utilizando métricas como: Matriz de confusión y sus clasificadores	Se genera puntuaciones numéricas de predicciones para cada registro de una fuente de datos de predicciones.
Implementación del modelo	Diseño un *modelo* básico de SADC	Se crea un modelo inicial de una aplicación que se utiliza para dar apoyo a la toma de decisiones clínicas.

Por otra parte, las herramientas de minería de datos y estadística multivariada fueron útiles ya que se dispone de un volumen de datos históricos importante. En este proceso, la minería de datos genera conocimiento por medio de la depuración, enriquecimiento y transformación de datos que sirve para la creación de un modelo en el que se evalúa un conjunto de casos. Tras el análisis, podemos deducir que las reglas de comportamiento identificadas colaboran con la toma de decisiones de los médicos de la UCIC. A lo largo del documento se ha logrado cumplir con el objetivo general que es determinar la viabilidad de desarrollo de un sistema de apoyo a la toma de decisiones clínicas en una Unidad de Cuidados Intensivos Coronarios. Los resultados experimentales de la investigación son alentadores mostrando que el modelo empleado es capaz de alcanzar una buena precisión predicción de pacientes cardiópatas.

En efecto, permite determinar el estado de gravedad de un paciente y qué conducta tomar según su estado, reduciendo sustancialmente el tiempo que un médico debe invertir para identificarlo. El objetivo principal del presente estudio fue elaborar un modelo de aprendizaje automático que permita predecir, analizar y pronosticar el grado de gravedad en los pacientes con enfermedad coronaria mediante el uso de árboles de decisión para prevenir el congestionamiento de la UCIC. Para la resolución del objetivo planteado, se aplicó el modelo de regresión logística de scikit-learn que, ha permitido clasificar y determinar el grupo de pacientes que deben ser atendidos en UCIC, tomando como variables independientes y relevantes: potasio, creatina sérica y ecografía FEY.

En el presente trabajo, la información conformada por datos de tipo cuantitativo se tradujo en visualizaciones, lo que evidenció patrones, tendencias y anomalías, lo que amplió el horizonte de lo visible. Es un proceso exitoso, ya que logra visibilizar la información y desde allí permite nuevas interpretaciones, descubriendo relaciones no percibidas en una primera instancia entre las partes.

Se concluye que, referente al estudio y a los análisis experimentales que han sido efectuados en otros modelos de predicción, el modelo de regresión logística nos devolvió resultados muy acertados. El impacto que tendrá el empleo del algoritmo de aprendizaje automático por su capacidad para ayudar a realizar diagnóstico de forma temprana, relativamente rápida y con precisión, generando un valioso conocimiento para los expertos en salud. De lo expresado por el especialista se rescata:

"…tener un resumen de una gran cantidad de datos enormes y la importancia de trabajar con datos locales. Los especialistas trabajan con datos y guías que vienen de afuera y no siempre obtenemos los mismos resultados…lo bueno de esto es poder sacar conclusiones con pacientes locales…"

5.2 Futuras Líneas de Investigación

En este proyecto no se abarcó la totalidad de los datos contenidos en las HCE, sino solo aquellos que contienen incidencia en la UCIC. Podrían surgir nuevas investigaciones, teniendo en cuenta la caracterización de los otros tipos de internaciones o la caracterización de los óbitos. Se sugiere tomar los datos históricos de más de un año. El objetivo es mantener la consistencia con los datos de entrada y prototipo del modelo de análisis predictivo.

Es importante tener presente que ningún modelo o algoritmo es el mejor para un determinado problema ya que la propia naturaleza de los datos afectará la elección de los modelos. Por ello, es conveniente evaluar la factibilidad de utilizar otras herramientas en el contexto de las HCE.

Se utilizó Python como lenguaje de programación porque se adapta mejor a los requerimientos para realizar las predicciones y análisis. Se pueden usar también otros lenguajes de programación para las respectivas pruebas y análisis.

El aplicativo fue diseñado teniendo en cuenta los procesos asistenciales de cada nivel de atención. Es una herramienta que logra colaborar con el cumplimento de los objetivos de la organización de manera eficiente y efectiva, contribuyendo con la mejora de procesos asistenciales.

Referencias

[1] D., Luna, Garfi, L., de Quiros, F. B., Gomez, A., and M., Martinez, (2001). Sistemas de Prescripción Electrónica. *InfoSUIS*, *10*, 3-6. Disponible en: https://www.hospitalitaliano.org.ar/multimedia/archivos/servicios_attachs/_prescripelectr.pdf[Accedido: 29, Sep, 2020].

[2] D., Luna, L., Garfi., F. B., de Quiros, A., Gomez and M., Martinez, "Sistemas de Prescripción Electrónica", *InfoSUIS*, *10*, 3-6, 2016.

[3] Carnicero, J., & Fernández, A., "Manual de salud electrónica para directivos de servicios y sistemas de salud", 2012.

[4] F. V. Navarro, and D. P., Rosa, "Sistema de soporte a las decisiones clínicas relacionadas con el diagnóstico precoz de enfermedades", Revista Cubana de Informática Médica, vol. *8*(3), pp. 533-544, 2016. [En línea], Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592016000300006

[5] S. J., Russell, and P., Norvig, "Inteligencia Artificial: un enfoque moderno" (No. 04; Q335, R8y, 2004). En línea], Disponible en: https://luismejias21.files.wordpress.com/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf

[6] L., Tundidor Montes de Oca, D., Nogueira Rivera, & A., Medina León, "Exigencias y limitaciones de los sistemas de información para el control de gestión organizacional", Revista Universidad y Sociedad, *10*(1), 8-14, 2018.

[7] "INDEC: Instituto Nacional de Estadística y Censos de la República Argentina". INDEC: Instituto Nacional de Estadística y Censos de la República Argentina. [En línea], Disponible en: https://www.indec.gob.ar/ (Accedido: 30, Sep, 2020)

[8] Dirección de Estadísticas e Información de la Salud (portal oficial del Estado argentino), Disponible en: https://www.argentina.gob.ar/salud/deis, [Acecesido: Nov. 12, 2007].

[9] de Cardiología, S. A. (2020). Documento de posición Sociedad Argentina de Cardiología– Fundación Cardiológica Argentina: Enfermedad Cardiovascular en tiempos de COVID-19, Disponible en: https://www.sac.org.ar/institucional/documento-de-posicion-sac-fca-enfermedad-cardiovascular-en-tiempos-de-covid-19/, [Acecesido: Nov. 14, 2007].

[10] R. E., Jiménez Paneque, "Indicadores de calidad y eficiencia de los servicios hospitalarios: Una mirada actual". Revista Cubana de Salud Pública, 30(1), 2004. Recuperado en 30 de Sep, 2020, de http://scielo.sld.cu/scielo.php?script=sci_arttextandpid=S0864-34662004000100004&lng=es&tlng=es.

[11] E., Morales-Blamhir, Carrillo-Pérez, Diego and Jesús, Rosas-Romero, Acute Pulmonary Embolism. 2014.

[12] M., Morales, "Diseño de la investigación. Planteamiento del problema de investigación", Seminario de investigación, 2016. [En línea], Disponible en: https://www. flipsnack. com/cencalli7/planteamiento-del-problema-de-investigacion. html.

[13] Eysenbach G. What is eHealth? J Med Internet Res.; 3(2): E20, 2001 Apr-Jun.

[14] Organización Panamericana de la Salud. Revisión de estándares de interoperabilidad para la eSalud en Latinoamerica y el Caribe. Washington, DC : OPS, 2016.

[15] Julio Bonisa, Juan J Sanchoa, Ferran Sanza, Sistemas informáticos de soporte a la decisión clínica, páginas 39-44, Febrero 2004

[16] Mihai Gheorghiade, MD, FACC,* Peter S. Pang, STATE-OF-THE-ART PAPER, Acute Heart Failure Syndromes, Vol. 53, No. 7, 2009© 2009 by the American College of Cardiology Foundation ISSN 0735-1097/09

[17] Ochoa Parra, Dr. Marcelo,'Historia y evolución de la medicina crítica: de los cuidados intensivos a la terapia intensiva y cuidados críticos', Acta Colombiana de Cuidado Intensivo, 2017

[18] Alonso, J. J., Sanz, G., Guindo, J., García-Moll, X., Bardají, A., & Bueno, H. (2007). Unidades coronarias de cuidados intermedios: base racional, infraestructura, equipamiento e indicaciones de ingreso. Revista española de cardiología, 60(4), 404-414.

[19] E. S., Gutiérrez and S., Vañó, "eSalud: aplicaciones y tendencias", Fundación Gaspar Casal, 2016.

[20] Maryam Ahmed, "50 principios de la ciencia de datos: innovaciones fundamentales Guía Breve", Liberty Vittert Blume, 2021

[21] Giraldo Mejía, J. C., & Vargas Agudelo, F. A. (2011). Aplicación de la técnica regresión logística de la minería de datos en el proceso de descubrimiento de conocimiento (KDD) en bases de datos operativas o transaccionales.

[22] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer y R. Wirth, "CRISP-DM 1.0", Step-by-step data mining guide, 2000.

[23] Marulanda Echeverry, Carlos Eduardo; López Trujillo, Marcelo; Mejía Salazar, María Helena, Minería de datos en gestión del conocimiento de pymes de Colombia, Revista Virtual Universidad Católica del Norte, núm. 50, febrero-mayo, 2017, pp. 224-237

[24] C., García Cambronero, and I., Moreno Gomez, (2006). ALGORITMOS DE APRENDIZAJE: KNN & KMEANS. http://www.it.uc3m.es/~jvillena/irc/practicas/08-09/06.pdf

[25] Víctor Martínez Velasco, Analítica predictiva en la toma de decisiones de la ingeniería: ejemplos de aplicaciones, Universitat Politècnica de Catalunya. Escola d'Enginyeria de Barcelona Est, 2019

[26] Dávila Hernández, Frank, & Sánchez Corales, Yovannys. (2012). Técnicas de minería de datos aplicadas al diagnóstico de entidades clínicas. *Revista Cubana de Informática Médica*, *4*(2), 174-183. Recuperado en 14 de febrero de 2022, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1684-18592012000200007&lng=es&tlng=es

[27] Raúl Benítez, Gerard Escudero, Samir Kanaan, David Masip Rodó, Inteligencia artificial avanzada, Editorial: Editorial UOC, 2013. ISBN 10: 8490298874 / ISBN 13: 9788490298879.

[28] Expósito Gallardo, María del Carmen, & Ávila Ávila, Rafael. (2008). Aplicaciones de la inteligencia artificial en la Medicina: perspectivas y problemas. ACIMED, 17(5) Recuperado en 14 de febrero de 2022, de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352008000500005&lng=es&tlng=es.

[29] Julian J. Faraway,Linear Models with Python. Taylor & Francis Ltd,2020

[30] Centro de Servicios Informáticos de la Universidad Nacional de Educación a Distancia. (s.f.). "Regresión Logística: Fundamentos y aplicación a la investigación sociológica" [PDF] https://www2.uned.es/socioestadistica/Multivariante/Odd_Ratio_LogitV2.pdf

[31] Hernandez, Alexander Baez, La detección del fraude contable utilizando técnicas de Minería de datos,Centro de Investigación en Alimentación y Desarrollo, AC

[32] Portilla, Jose. 2018. *Data Science and Machine Learning Bootcamp with R*. https://www.udemy.com/data-science-and-machine-learning-bootcamp-with-r/.

[33] Britos, P., Hossian, A., García-Martinez, R. y Sierra, E. 2005. Minería de Datos Basada en Sistemas Inteligentes. Nueva Librería.

[34] Marcos Orellana, Priscila Cedillo Detección de valores atípicos con técnicas de minería de datos y métodos estadísticos, Enfoque UTE, V.11-N.1, Ene.2020, pp. 56-67
[35] Berlanga, V., Rubio Hurtado, M. J., & Vilà Baños, R. (2013). Cómo aplicar árboles de decisión en SPSS. *REIRE. Revista d'Innovació i Recerca en Educació, 2013, vol. 6, num. 1, p. 65-79*.
[36] Jose Berengueres (Goodreads Author),Marybeth Sandell,Ali Fenwick,Barbara Covarrubias (Editor),Angels Berengueres, Visualización de Datos & Storytelling febrero 2020.
[37] Manuela Aparicio and Carlos J. Costa. 2015. Data visualization. Commun. Des. Q. Rev 3, 1 (November 2014), 7–11. DOI: https://doi.org/10.1145/2721882.2721883
[38] Dr. Abela, The Extreme Presentation(tm) Method, Fuente:https://extremepresentation.typepad.com/blog/2009/01/qu%C3%A9-gr%C3%A1fico-elegir-chart-chooser-in-spanish.html
[39] J. Juzgado, N. (1996). Procesos de construcción del software y ciclos de vida. España: UniversidadPolitécnica de Madrid.
[40] Challenger-Pérez, Ivet; Díaz-Ricardo, Yanet; Becerra-García, Roberto Antonio, El lenguaje de programación Python,Ciencias Holguín, vol. XX, núm. 2, abril-junio, 2014
[41] Andrés Marzal Isabel Gracia, Introducción a la programación con Python, Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I https://ns2.elhacker.net/timofonica/manuales/Introduccion_%20Programacion_Python.pdf
[42] Fabián Pedregosa; Gaël Varoquaux; Alexandre Gramfort; Vincent Michel; Bertrand Thirion; Olivier Grisel; Mathieu Blondel; Peter Prettenhofer; Ron Weiss; Vincent Dubourg; Jake Vanderplas; Alexandre Passos; David Cournapeau; Matthieu Perrot; Édouard Duchesnay (2011). "Scikit-learn: Machine Learning en Python". Revista de investigación sobre aprendizaje automático. 12 : 2825-2830.
[43] Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B., & Varoquaux, G. (2014). Aprendizaje automático para neuroimagen con scikit-learn. Frontiers in Neuroinformatics, 0(FEB), 14. https://doi.org/10.3389/FNINF.2014.00014
[44] Arturo Fernández Montoro,PYTHON 3 al descubierto 2º Edición (Spanish Edition), RC Libros (1 Junio 2013)
[45] Assaf, D., Gutman, Y., Neuman, Y., Segal, G., Amit, S., Gefen-Halevi, S., Shilo, N., Epstein, A., Mor-Cohen, R., Biber, A., Rahav, G., Levy, I., and Tirosh, A. (2020). Utilization of machine-learning models to accurately predict the risk for critical COVID-19. Internal and Emergency Medicine, 15(8), 1435–1443. https://doi.org/10.1007/s11739-020-02475-0
[46] Fernández-García, P., Vallejo-Seco, G., Livacic-Rojas, P. E., & Tuero-Herrero, E. (2014), Validez Estructurada para una investigación cuasi-experimental de calidad. Anales dePsicología, https://doi.org/10.6018/analesps.30.2.166911
[47] Bono-Cabré R. Diseños cuasi-experimentales y longitudinales. España: Universidad de Barcelona/Facultad de Psicología/Departamento de Metología de las Ciencias del Comportamiento; 2012.
[48] Ignasi Velázqueza, Xavier Navarrob, Albert Cobosa,Captura electrónica de datos. Impacto en la calidad de la investigación clínica.
[49] Jean-Pierre Bassand, Christian W. Hamm, Diego Ardissino, Guidelines for the diagnosis and treatment of non-ST-segment elevation acute coronary syndromes: The Task Force for the Diagnosis and Treatment of Non-ST-Segment Elevation Acute Coronary Syndromes

of the European Society of Cardiology, European Heart Journal, Volume 28, Issue 13, July 2007, Pages 1598–1660, https://doi.org/10.1093/eurheartj/ehm161

[50] Julio Bonisa, Juan J Sanchoa, Ferran Sanza,Sistemas informáticos de soporte a la decisión clínica, páginas 39-44, Febrero 2004.