Análisis de reingresos de pacientes adultos utilizando redes neuronales recurrentes: caso de estudio en un hospital de tercer nivel de la ciudad de Medellín.

Deiner Alonso Rivera Soto

Trabajo de investigación para optar al título de:
Bioingeniero

Asesora
María Bernarda Salazar Sánchez, Doctora en Ingeniería Electrónica

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Bioingeniería
Medellín
2024

Cita	(Rivera Soto, 2024)
Referencia **Estilo APA 7 (2020)**	Rivera Soto, D. A (2024). "*Análisis de reingresos de pacientes adultos utilizando redes neuronales recurrentes: caso de estudio en un hospital de tercer nivel de la ciudad de Medellín"*. Proyecto de grado, Bioingeniería, Universidad de Antioquia, Medellín, 2024.

Grupo de Investigación Intelligent Information Systems Lab In2Lab

UNIVERSIDAD DE ANTIOQUIA
Vicerrectoría de Docencia

Sistema de Bibliotecas

Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: http://bibliotecadigital.udea.edu.co

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.
Decano: Julio César Saldarriaga.
Jefe departamento: John Fredy Ochoa Gómez.

Agradecimientos

Agradezco a todos los compañeros miembros del proyecto PRG 2022-52993 Predicción del riesgo de readmisión en pacientes hospitalarios del grupo de investigación IN2LAB de la Universidad de Antioquia, especialmente a la profesora María Bernarda Salazar Sánchez por su constante acompañamiento y sacrificio en el desarrollo de este proyecto, ya que sin ella sería imposible el desarrollo del mismo, además al compañero John Jader Caro Sánchez, quien aportó en gran medida al desarrollo de este.

Al laboratorio IN2LAB por darme la oportunidad y brindarme el espacio y todos los recursos de hardware y software para cumplir con los objetivos propuestos.

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos y abreviaturas

ML	Machine Learning.
ANN	Artificial Neural Networks.
RNN	Recurrent Neural Networks.
ERIC	Education Resources Information Center.
PE	Process Element.
MedPAC	Medicare Payment Advisory Commission.
SIS	Sistemas de información Sanitarios.
EDA	Exploratory Data Analysis

Resumen

La readmisión hospitalaria de pacientes es un fenómeno muy común a nivel nacional, es costoso y puede conllevar a procesos en los cuales la evaluación en la calidad de los servicios de salud por parte de los centros que lo prestan se puede ver altamente comprometida, debido a que esto es un índice de la calidad de servicio prestado al paciente durante su estancia. Por lo anterior es indispensable mitigar este tipo de impactos, ya que al predecir con anticipación las readmisiones puede aumentar la calidad de la atención hospitalaria y reducir los abultados costos a largo plazo. En el presente artículo se usarán dos técnicas de clasificación basadas en árboles: Árbol de Decisión y Bosques Aleatorios, y la técnica de aprendizaje profundo denominada Redes Neuronales Recurrentes con el objetivo de predecir la ventana de readmisión de un paciente a 3, 28 o más de 28 días desde una base de datos otorgada por un hospital de tercer nivel de la ciudad de Medellín. Los resultados muestran que no existe una correlación marcada entre las diferentes bases de datos por lo cual al aplicar el método de Bosques Aleatorios y Redes Neuronales Recurrentes, no se encuentra una mejora significativa de predicción con respecto al Árbol de Decisión.

Palabras clave:

Base de datos, Readmisión hospitalaria, Machine Learning, Árboles de Decisión, Bosques Aleatorios, Deep Learning, Redes Neuronales Recurrentes (RNN), Predicción.

Abstract

Hospital readmission of patients is a very common phenomenon nationwide, it is costly, and can lead to processes in which the assessment of the quality of healthcare services provided by the centers can be highly compromised, as this is an index of the quality of service provided to the patient during their stay. Therefore, it is essential to mitigate these types of impacts, as predicting readmissions in advance can increase the quality of hospital care and reduce long-term hefty costs. In this article, two classification techniques based on trees will be used: Decision Tree and Random Forests, and the deep learning technique called Recurrent Neural Networks with the aim of predicting the readmission window of a patient at 3, 28, or more than 28 days from a database provided by a tertiary hospital in the city of Medellín. The results show that there is no marked correlation between the different databases, therefore, when applying the Random Forests and Recurrent Neural Networks method, there is no significant improvement in prediction compared to the Decision Tree.

Keywords:

Database, Hospital Readmission, Hospital Readmission, Machine Learning, Decision Trees, Random Forests, Deep Learning, Recurrent Neural Networks (RNN), Prediction.

1. Introducción

Se define reingreso hospitalario como una nueva admisión al centro de salud por la misma causa en un periodo menor a 30 días desde el egreso hospitalario. En Colombia, este fenómeno es evaluado como un indicador de calidad según la resolución 256 de 2016, que se enfoca en los reingresos menores a 15 días por la misma causa. La relevancia de este tema se evidencia en estudios previos, como el realizado por [1], que encontró una prevalencia del 10.1 % de reingresos hospitalarios asociados a mayores tasas de mortalidad, costos y estancias prolongadas.

La identificación de factores de riesgo asociados a los reingresos hospitalarios es fundamental para generar intervenciones preventivas efectivas y mejorar variables como la estancia media, la mortalidad y el costo total de la atención. Dado el creciente costo de la atención sanitaria, existe un interés creciente por parte de los responsables políticos, investigadores y pagadores en reducir los reingresos hospitalarios. En este contexto, se busca desarrollar modelos predictivos que permitan comparar los reingresos hospitalarios entre diferentes instituciones e identificar pacientes de alto riesgo que requieran intervenciones específicas.

El objetivo de este estudio fue explorar el potencial de los modelos predictivos en reingresos hospitalarios y evaluar el poder predictivo de diversas variables independientes mediante el uso de técnicas de aprendizaje automático. En este informe encontrarán las fases de desarrollo del algoritmo de predicciones sobre reingresos hospitalarios, esto como herramienta de apoyo para el personal asistencial y administrativo en la toma de decisiones clínicas y de gestión en instituciones de salud.

2. Planteamiento del problema

El reingreso hospitalario es un fenómeno el cual plantea grandes desafíos para los diferentes sistemas de salud alrededor del mundo. En Colombia, como en muchos otros países, los reingresos hospitalarios son considerados como un indicador de calidad en la atención médica. Sin embargo, la recurrencia de ingresos hospitalarios por la misma causa dentro de un corto período de tiempo no solo implica un mayor costo para el sistema de salud, sino que también está asociada con resultados desfavorables para los pacientes, como una mayor morbimortalidad y una menor calidad de vida. Un estudio realizado por el Comité asesor de pagos de Medicare (MedPAC) en Estados Unidos [2] informó que el 17,6 % de las admisiones hospitalarias resultaron en reingresos dentro de los 30 días posteriores al alta, y el 76 % de estos fueron potencialmente evitables. En total, estos reingresos representaron $15 mil millones en gasto de Medicare. En un esfuerzo por frenar las tasas de reingreso hospitalario, parte de la Ley de Protección al Paciente y Atención Médica asequible penaliza a los hospitales con reingresos excesivos a los 30 días a través de un programa llamado Programa de Reducción de Reingreso Hospitalario. En el año fiscal 2013, más de 2.000 hospitales fueron penalizados con más de 280 millones de dólares. El 1 de octubre de 2014, la multa aumentó a un mínimo del 3% del reembolso de Medicare de un hospital y también incluyó varias condiciones más.

A pesar de los esfuerzos por parte de las instituciones de salud para abordar este problema, la identificación y el manejo efectivo de los factores de riesgo asociados con los reingresos hospitalarios continúan siendo un desafío. La complejidad de estos factores, que pueden incluir desde características clínicas y demográficas de los pacientes hasta aspectos relacionados con la atención médica recibida, dificulta la implementación de estrategias de prevención y gestión de reingresos efectivas. Es por lo anterior que surge la necesidad de desarrollar modelos predictivos precisos que puedan identificar de manera temprana a los pacientes con mayor riesgo de reingreso hospitalario. Estos modelos podrían ayudar a los profesionales de la salud a anticiparse a las necesidades de los pacientes, optimizar los recursos y mejorar los resultados clínicos. Sin embargo, para lograr este objetivo, es crucial comprender en profundidad los factores que contribuyen a los reingresos hospitalarios y evaluar la eficacia de diferentes enfoques de modelado predictivo en este contexto.

En este sentido, el presente estudio aborda esta problemática mediante el análisis de datos clínicos y la aplicación de técnicas básicas de aprendizaje automático, poniendo mayor énfasis en las redes neuronales recurrentes. Lo anterior como estrategia para buscar, identificar y evaluar los factores predictivos de reingreso hospitalario y desarrollar modelos predictivos robustos que puedan ser implementados en la práctica clínica para mejorar la atención y el manejo de los pacientes en riesgo de reingreso.

3. Objetivos

3.1. Objetivo general

Predecir el riesgo de reingreso hospitalario de pacientes en una institución prestadora de servicios de salud de alta complejidad de la ciudad de Medellín utilizando técnicas basadas en aprendizaje automático.

3.2. Objetivos específicos

1. Caracterizar la población de pacientes adultos de un hospital de alta complejidad de la ciudad de Medellín, Antioquia.
2. Identificar las principales variables relacionadas con los factores de riesgo de readmisión en la cohorte de pacientes definida.
3. Diseñar y evaluar una metodología enfatizada en la técnica de redes neuronales recurrentes y evaluar su rendimiento para la predicción de reingreso en la cohorte de pacientes definida.

4. Marco teórico

4.1. Readmisión Hospitalaria

Puede definirse como readmisión de un paciente a un hospital como el periodo de tiempo transcurrido tras egresar del mismo u otro centro hospitalario [3]. El tiempo para medir una readmisión varía, y la tendencia más común es entre los primeros 30 días posteriores al egreso. La tasa de reingresos en este lapso para diferentes centros médicos y países oscila entre el 5 % y el 19.66 % [4]. Sin embargo, actualmente los reingresos hospitalarios son eventos que se presentan con holgada frecuencia en el país, además, estos son frecuentes, costosos y de alta mortalidad [2]. Como consecuencia los reingresos hospitalarios reflejan las relaciones entre los niveles asistenciales y la calidad de la prestación del servicio y atención a personas de avanzada edad; por lo tanto, las entidades prestadoras de servicio de salud deben tomar medidas que reduzcan este fenómeno que incide y genera repercusiones negativas en los hospitales dado que, además de disminuir la calidad de vida del paciente, también genera un aumento considerable en los gastos médicos de la entidad prestadora de servicios de salud.

Ante la importancia de identificar los factores asociados a un mayor riesgo de reingresos, diversos autores han desarrollado varios modelos estadísticos, a partir de características del paciente disponibles en sistemas de información sanitarios (SIS). A pesar de las limitaciones que puedan presentar las bases de datos clínico-administrativas, tanto de atención primaria (OMI-AP) como hospitalaria (conjunto mínimo básico de datos [CMBD]) se consideran instrumentos útiles para valorar la efectividad de la atención sanitaria [5].

4.2. Técnicas de aprendizaje automático

El aprendizaje automático (ML del inglés, Machine Learning) es un procedimiento por el que un sistema aprende identificar o predecir características de un conjunto de datos a partir de patrones y relaciones funcionales. Mitchell [6] lo define afirmando que "*se dice que un programa de ordenador aprende de la experiencia E con respecto a una clase de tareas T y una medida de rendimiento P, si su rendimiento en las tareas T, medido por P, mejora con la experiencia E*". Por

lo que, las tareas del ML se describen según cómo el sistema puede procesar un ejemplo específico, y pretende comprender la estructura de los datos y ajustarlos a modelos para poder entender mejor.

A través de la historia el campo del ML ha experimentado una notable evolución desde sus inicios en la década de 1950. En sus primeros años, pioneros como Martin Minsky, John McCarthy y Frank Rosenblatt sentaron las bases de lo que hoy conocemos como inteligencia artificial y aprendizaje automático. A pesar de enfrentar desafíos financieros en la década de 1970, la ingeniosa creación del "Stanford Car" en 1979 demostró el potencial de la inteligencia artificial para superar obstáculos.

La contribución fundamental del algoritmo vecinos más cercanos (Del inglés, Nearest neighbor) permitió capacitar a las máquinas para reconocer patrones y generar soluciones efectivas. Esto marcó un hito en el procesamiento de datos en los años ochenta, con la creación de modelos y sistemas expertos que encontraron amplia aceptación en el ámbito empresarial. En la década de 1980, figuras como Gerald Dejong introdujeron enfoques innovadores como el "Explanación Base Learning" (EBL), que permitieron no solo trabajar con variables existentes, sino también formular nuevas variables, ganando reconocimiento en la industria.

El desarrollo del programa "NetTalk" por Terry Sejnowski en 1985 demostró la capacidad de las máquinas para aprender la pronunciación de palabras, destacando aún más el potencial del aprendizaje automático. Sin embargo, a finales de la década de 1990, la inteligencia artificial experimentó un estancamiento, distanciándose de la IA como herramienta y consolidándose como un campo independiente. El nuevo milenio trajo consigo un renacimiento del aprendizaje automático, con empresas líderes como IBM y Microsoft impulsando su desarrollo. El lanzamiento de iniciativas como Azure Machine Learning de Microsoft en 2006 y Watson de IBM en 2011 marcó el comienzo de una nueva era para el ML, demostrando su potencial en aplicaciones del mundo real, como el reconocimiento de voz y el análisis de datos [7].

En el ámbito del Machine Learning, se distinguen tres tipos principales de modelos: supervisados, no supervisados y semi-supervisados, cada uno explicado brevemente a continuación:

- *Aprendizaje no supervisado:* se basan en datos no etiquetados. Aquí los datos de entrenamiento se suministran al sistema sin etiquetas ni valores preasignados, por lo cual el algoritmo de aprendizaje debe encontrar puntos en común entre los datos de entrada

dividiendo los datos en diferentes grupos en función de sus similitudes identificando patrones en estos.

- *Aprendizaje supervisado:* utilizan conjuntos de datos etiquetados para entrenar modelos predictivos. En este caso, a diferencia del método anterior, se cuenta con un dato fundamental que se conoce como "la variable objetivo" o "clase" [8]. Así se usan datos etiquetados y una colección de muestras para estimar una función. Aquí el objetivo es que el algoritmo pueda aprender comparando su salida real con las salidas enseñadas para encontrar errores y modificar el modelo. En pocas palabras, el algoritmo debe tener la capacidad de darle sentido a los datos sin la presencia del "supervisor".
- *Aprendizaje semi-supervisado:* aborda la escasez de muestras etiquetadas al combinar datos sin etiquetar con un número reducido de muestras etiquetadas, permitiendo la construcción de clasificadores más efectivos.

Los modelos predictivos basados en ML usan datos para detectar patrones y correlaciones difíciles de discernir mediante el análisis convencional, lo que permite a los profesionales de la salud tomar medidas preventivas para evitar reingresos innecesarios. Cabe destacar que el uso del aprendizaje profundo (Del inglés, Deep Learning), que es un subconjunto del aprendizaje automático basado en redes neuronales, ha logrado buenos resultados en la predicción de reingresos no planificados [5]. En 2023 [9], con el fin de evaluar la efectividad de los índices estadísticos generales para predecir readmisión de pacientes (LACE y HOSPITAL score), valida un modelo predictivo basado en regresión logística para estimar el riesgo de reingreso hospitalario por exacerbación de la enfermedad pulmonar obstructiva crónica en los 30 días siguientes tras el alta hospitalaria. Luego compara los resultados de ese modelo con los índices LACE y HOSPITAL score. El modelo propuesto incluyó inicialmente 136 variables que correspondían a datos sociodemográficos, clínicos, de función pulmonar, de calidad de vida, de tratamiento, recursos sanitarios utilizados y características del ingreso y del alta.

4.2.1. Árboles de Decisión (Decission Trees)

Los árboles de decisión son una de las representaciones del Machine Learning el cual se caracteriza principalmente por ser un método iterativo de aprendizaje supervisado usado esencialmente para problemas de clasificación o regresión, por lo cual la estrategia de aprendizaje

de este método se basa en el principio de “divide y vencerás”, ya que aquí se indagará codiciosamente cuáles son los puntos de división óptimos dentro del árbol. Como se observa en la **Figura *1***, el árbol de decisión comienza su proceso en un nodo raíz que no tiene ramas entrantes por ser el punto de partida. A este nodo se le conoce como “nodo principal” y es en donde se comienza el método de condiciones sucesivas, siguiendo un camino hacia abajo. Estas condiciones sucesivas constan de una serie de preguntas sobre las características asociadas a los elementos. Cada pregunta está en un nodo (nodo interno), y sus posibles respuestas apuntarán a los nodos hijos, lo que permitirá formar conjuntos homogéneos sobre los datos y clasificar los datos en un orden jerárquico, lo que le da su forma de árbol [9] [10].

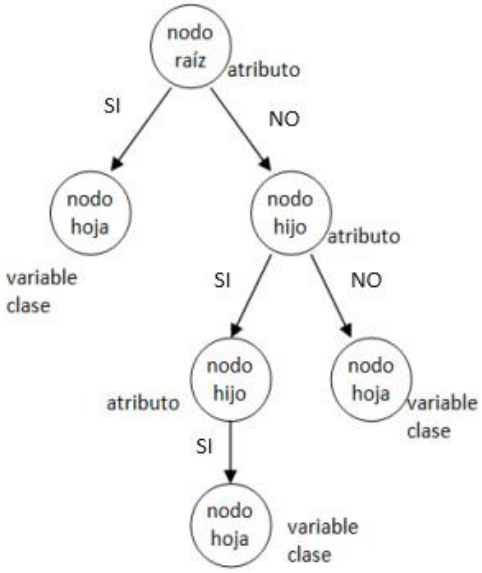

Figura 1. Estructura básica de un árbol de decisiones [9].

En su forma más simple, el nodo interno se plantea una pregunta cuya respuesta sólo se limita a “sí” o “no”, y aquí se dará la división en dos nodos hijos, en donde uno responde y forma el subconjunto con la respuesta “sí” y el otro con la respuesta “no”.

Para clasificar un elemento en una clase, basta con seguir el camino desde el nodo raíz hasta un nodo hoja, teniendo en cuenta las respuestas obtenidas. Así, cada elemento es asociado a la categoría del nodo hoja alcanzado. Algunos de los algoritmos de clasificación más importantes en el método de árboles de decisión son:

- **ID3**: Este método se basa en buscar el mejor atributo en el dataset que brinde la mayor ganancia de información (en relación con la entropía) para ir dividiendo los datos en clases homogéneas hasta llegar a los nodos hojas que determinen esta clase. Se debe tener en cuenta que para el uso de este método los atributos deben ser categóricos [11].
- **C4.5**: Este algoritmo viene siendo el sucesor del método ID3, pero a diferencia de este, el método C4.5 acepta atributos numéricos con valores continuos, y soporta un dataset con valores faltantes. Además, se puede usar la estrategia de poda del árbol para evitar el overfitting.
- **CART**: Este tipo de algoritmo es el más usado en la actualidad. Su principal característica es que se puede usar tanto para procesos de clasificación como para selección. Aquí se usa la metodología de división binaria recursiva, en donde en cada paso se divide todo el conjunto de datos en dos subconjuntos. Para medir el grado de homogeneidad, se usa el índice Gini, el cual mide el grado de impureza del nodo (eq. 1).

$$G = 1 - (prob.\ categ.\ 1)^2 - (prob.\ categ.\ 2)^2 \qquad \text{(eq. 1)}$$

El grado de purea o impureza del nodo depende del valor arrojado por el índice Gini, de la siguiente forma:

- $G = 0$: indica que el nodo es puro (datos que pertenecen a una sola categoría).
- $G > 0$: indica nodos impuros (datos con más de una categoría).

Finalmente, para definir la mejor partición, el algoritmo CART define una función de costo que asigna un puntaje al nodo padre de acuerdo con los valores del índice Gini individuales de sus nodos hijos.

4.2.2. Bosques Aleatorios (Random Forest)

Como se mencionó anteriormente, los árboles de decisión tienen una gran desventaja, y es que cuando se tiene una cantidad de datos lo suficientemente grande, estos funcionan muy bien en el entrenamiento (BIAS bajo), pero no lo hacen también cuando se desea hacer predicciones con nuevos datos, lo que se traduce en un overfitting, ya que los árboles de decisión son muy sensibles a los datos de entrenamiento. Es aquí donde los bosques aleatorios entran a complementar esta

desventaja y lo que a su vez lo convierte en uno de los algoritmos de aprendizaje automático supervisado más poderosos del Machine Learning.

Los bosques aleatorios corresponden a un conjunto de árboles de decisión (denominados clasificadores débiles) el cual trabaja con un conjunto de árboles de baja correlación (de aquí el término bosque) y los promedia [12]. Lo anterior permite combinar la *sencillez* que proporciona el método de árboles de decisión (BIAS bajo) con la *flexibilidad* de la que carecen (alta varianza). Es así como los bosques aleatorios permiten tener una mejor precisión de predicción cuando se necesita implementar nuevos datos al modelo.

La implementación de esta técnica se puede resumir en los siguientes pasos:

1. Número de árboles: normalmente se usan decenas o cientos de árboles.
2. Subset de datos: se define que datos serán usados para el entrenamiento del árbol (tomados del set de datos original). La elección de este subset se da mediante el método del "bootstrapping", que indica que se pueden repetir filas en cada subset.
3. Entrenamiento: Se toma cada subset de datos y de forma recurrente se realizan las particiones en el espacio de las variables para poder crear así cada árbol. Luego, en vez de tomar todas las características, se tomará únicamente una parte de estas.
4. Predicción: se introduce el nuevo dato a cada árbol, se obtiene el valor de la predicción otorgada por cada uno de estos árboles, y al final, dependiendo si se está realizando proceso de clasificación o regresión, se toma la categoría asignada por la mayor cantidad de árboles, o se toma el promedio de las predicciones de estos, respectivamente [13].

En conclusión, el Random Forest es una combinación de clasificadores débiles (árboles de decisión), lo cual se le conoce como ***Bagging***, el cual se centra en promediar los valores de los árboles incorrelacionados con el fin de hallar mejores aproximaciones evitando el overfitting.

4.2.3. Redes neuronales artificiales

Una red neuronal artificial es una de las aplicaciones más poderosas del Machine Learning, la cual consiste en enseñar a aprender a los sistemas informáticos de una manera similar a como lo hace el cerebro humano. A esto se le denomina aprendizaje profundo, ya que usa los nodos o capas de neuronas que se interconectan de una manera muy similar al mecanismo estructural del cerebro

humano. El concepto de red neuronal se adopta debido a que su diseño se enfoca en modelizar el mecanismo mediante el cual el sistema nervioso de un ser vivo realiza una tarea específica. Esto se logra mediante un conjunto de unidades fundamentales de procesamiento llamadas neuronas [14].

Cada neurona recibe una señal, ya sea continua o discreta, las pondera e integra, para luego transmitir esta información al resto de neuronas adyacentes. Cada conexión entre neuronas tiene asociado un peso el cual determina la importancia asociada a esta conexión y suele guardar el conocimiento que la red neuronal tiene sobre la tarea que se está desarrollando. Ahora, el procedimiento por el que se ajustan estos pesos para lograr el mejor resultado en una determinada tarea se llama *entrenamiento* o *aprendizaje,* y el ajuste de estos pesos es el principal mecanismo con el que aprende la red neuronal.

Las RNA se destacan por su estructura *paralelizable* y su elevada capacidad de *generalización,* lo que se traduce en producir salidas correctas ante entradas no usadas durante el entrenamiento. Otras propiedades de las RNA son:

- **No linealidad:** Característica muy importante si se intenta modelar o predecir sistemas determinados por puntos no lineales.
- **Adaptabilidad:** Capacidad para ajustar los pesos ante las variaciones en el entorno (datos de entrada no estacionarios).
- **Tolerancia ante fallos:** Los fallos operacionales en ciertas regiones de la red solo afectan débilmente su rendimiento, por la distribución de la información almacenada [15].

Tal como ha descrito, una red neuronal es una interconexión de unidades de procesamiento simples conocidos como neuronas (PE, del inglés process element). El PE tiene varias entradas, y las combina con una suma básica en la entrada, que a su vez se modifica por una función de transferencia, y el valor de la salida de esta función de transferencia se reflejará en la salida del PE.

La salida del PE puede conectarse con otra red de PE mediante conexiones correspondientes a la efectividad sináptica de las conexiones neuronales de la red. Una configuración básica de red neuronal podría incluir n entradas, m capas ocultas y una salida, como se ilustra en la **Figura *2***.

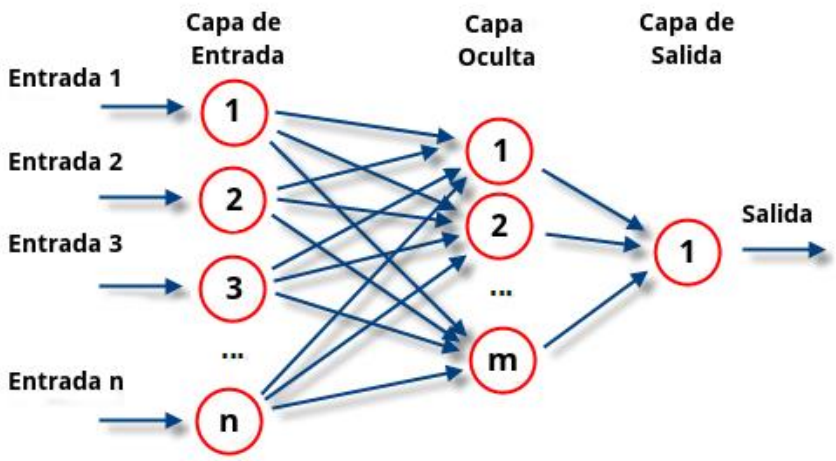

Figura 2. Diagrama de una Neurona Artificial. Tomado de [16].

Las capas ocultas y de salida procesan múltiples entradas para generar salidas. Cada unidad en la capa oculta multiplica las entradas por pesos sinápticos asociados, derivados del proceso de entrenamiento, y luego las suma antes de pasarlas a través de una función de activación que escala la salida a un rango específico, típicamente entre 0 y 1. Estas salidas se convierten en entradas para la siguiente capa, y así sucesivamente hasta alcanzar las salidas finales. La elección de la función de activación permite adaptar la red a la tarea específica en cuestión, con preferencia frecuente por funciones logarítmicas o tangente hiperbólica debido a su continuidad y simplificación de los algoritmos de entrenamiento. Estas características no lineales garantizan que un mayor nivel de activación por encima de un umbral máximo no afecte adicionalmente a la salida, proporcionando funciones de saturación [16]. A continuación, se abordará este concepto desde el modelo de neurona y sus funciones internas.

En el modelo convencional de neurona se pueden identificar cinco conceptos básicos:

- Señales de entrada *(zi[t]):* Son los datos que se procesan en la neurona, y son aquellos que llevan consigo la información del entorno. Estos pueden provenir del exterior de la red, puede ser el output de otra neurona, o puede ser el feedback de la misma neurona que lo recibe.
- Sinapsis: Esta se caracteriza por un propio peso Wji, y este peso se asocia a la sinapsis que se da entre la unidad i-ésima con la neurona j-ésima.

- Sesgo: Este se caracteriza por aumentar o disminuir la entrada de la neurona conforme el valor de entrada sea positivo o negativo, lo que incide en la capacidad de procesamiento de esta.
- Integrador: Aquí se suman todas las entradas que convergen en la neurona y el sesgo. Estas entradas ya tienen ponderados sus propios pesos.
- Función de activación: Esta función permite acotar la amplitud del valor de salida de la neurona. El rango depende de la función de activación usada, las cuales se mencionarán a continuación:
  - *Función identidad:* tiene la forma *gi(x)=x. Esta* se usa cuando no se desea acotar la salida de la neurona. Y se usa generalmente para las neuronas de entrada a la red o en modelamiento de sensores.
  - *Función escalón:* Es también conocida matemáticamente como función de Heaviside, sólo permite dos valores de salida (1 ó 0) y adopta la forma:

$$f(x) = \begin{cases} 0, si\ x < 0 \\ 1, si\ x \geq 0 \end{cases} \qquad \text{(eq. 2)}$$

  - *Función logística:* Esta es una de las representaciones de las funciones sigmoideas, las cuales se caracterizan por ser crecientes, acotadas y transformar el argumento de entrada de una manera no lineal. La función logística se define matemáticamente como:

$$\boldsymbol{g_L(x) = \frac{1}{1+e^{-x}}} \qquad \text{(eq. 3)}$$

  - *Función tangente hiperbólica (tanh(x)):* Es otro tipo de función sigmoidea, la cual a diferencia de la función logística mencionada anteriormente, está acotada entre –1 y 1. Su ecuación se define como:

$$\boldsymbol{\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}} \qquad \text{(eq. 4)}$$

Ahora, cuando se trata de diseñar una red neuronal artificial, es importante tener en cuenta, además de lo anteriormente mencionado, dos parámetros muy importantes, los cuáles se mencionarán a continuación:

- **Número de neuronas en la capa oculta:** este hiperparámetro indica la cantidad de neuronas que procesarán las diferentes características de los datos e influye en su capacidad de modelado, representación de características, tiempo de entrenamiento y capacidad de generalización. Este se debe seleccionar con precaución para obtener el máximo rendimiento en la tarea específica en la que se está interesado, pero a su vez evitar sobreajustes en el proceso de entrenamiento, además de consumo excesivo de memoria y tiempos de ejecución. Esto generalmente se logra a través de técnicas de búsqueda en cuadrículas (proceso de grid search), validación cruzada o ajuste manual, según la experiencia y el conocimiento del problema.

- **Tasa de aprendizaje inicial:** Este hiperparámetro clave en el contexto de redes neuronales en donde principalmente se busca controlar el valor de los ajustes que se realizan a los pesos de las neuronas de la red en el proceso de entrenamiento de esta. Lo anterior con el objetivo de determinar desde el principio qué tan grandes serán los pasos que se darán en dirección opuesta al gradiente durante el ajuste de los pesos de la red neuronal. Su importancia radica en que una tasa de aprendizaje bien ajustada puede permitir que el modelo converja lo más eficientemente hacia el mínimo global de la función de pérdida; mientras que una función de pérdida elegida incorrectamente puede conllevar a un sobreajuste del modelo o a la convergencia prematura sobre un mínimo local no global.

4.2.4. Redes neuronales recurrentes

En los últimos años se ha producido una amplia variedad de topologías de redes neuronales, sin embargo, la mayoría de ellas se encuentran ubicadas en dos grandes grupos: las redes multicapa de alimentación hacia adelante (feed-forward) y las redes recurrentes (RNR). Las redes feed-forward no tienen ciclos, las neuronas están organizadas en capas que se conectan de manera unidireccional. Generalmente estas redes son denominadas estáticas, pues producen una única salida para un conjunto de entrada, o sea, el estado de una red es independiente del estado anterior [17].

Por otro lado, las RNN son un tipo de aprendizaje profundo caracterizado por procesar y obtener la información otorgada por datos con información secuencial. Es así como las RNN son sistemas dinámicos no lineales que identifican regularidades temporales en los datos procesados, permitiendo aplicaciones a fenómenos secuenciales como videos, subtitulación, imágenes,

procesamiento natural del lenguaje, etc. La diferencia con respecto al resto de las técnicas de aprendizaje profundo es que estas no asumen la independencia de los datos de entrada, sino que capturan sus dependencias secuenciales y temporales para predecir comportamientos que se podrían dar en un tiempo posterior.

Las RNN reciben una serie de inputs y generan, una vez tratados, un output, con la especial y esencial diferencia de que, en las neuronas de estas, el output de un timestep se pasa al siguiente, creando un bucle que permite retener información pasada. En otras palabras, una RNN se retroalimenta mediante la detección de dependencias en variables de gran longitud gracias a la conservación de la información. El output del modelo no sólo es ahora una función de unos determinados inputs, sino también de la memoria de la red en sí misma [18] [19] [21].

4.3. Métricas de evaluación

4.3.1. Matriz de confusión

Al momento de implementar un modelo de Machine Learning es indispensable analizar los resultados los resultados en la clasificación del modelo. Es por esto que la matriz de confusión es la herramienta más popular y simple para visualizar gráficamente el rendimiento del modelo comparando las predicciones con las clases reales de los datos. La matriz de confusión consta básicamente de una tabla de dos dimensiones en la cual se puede evidenciar la cantidad de observaciones que el modelo aplicado pudo clasificar correcta e incorrectamente; para esto, la matriz tiene sus cuatro entradas:

- *TP:* Número de muestras positivas clasificadas correctamente.
- *FP:* Número de muestras positivas clasificadas incorrectamente.
- *TN:* Número de muestras negativas clasificadas correctamente.
- *FN:* Número de muestras negativas clasificadas incorrectamente.

Así, con los valores de estas cuatro entradas es que es posible calcular las métricas más importantes para determinar la calidad del modelo, estas son:

- *Accuracy (precisión):* cantidad de predicciones correctas en comparación con el número total de predicciones. Se calcula numéricamente como:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad \text{(eq. 6)}$$

- *Recall (Sensibilidad):* Proporción de casos positivos que fueron correctamente predichos por el modelo. Se expresa como:

$$Recall = \frac{TP}{TP+FN} \qquad \text{(eq. 7)}$$

- *Precision (Valor predictivo positivo):* Mide la proporción de casos positivos que son realmente clasificados como positivos. Se calcula como:

$$precision = \frac{TP}{TP+FP} \qquad \text{(eq. 8)}$$

- *F1 Score:* Esta métrica permite combinar la Precision y Recall en una misma medida para determinar el rendimiento general del modelo implementado. Su fórmula es:

$$F1\ Score = 2 \cdot \frac{\text{Pr}\,ecision \cdot recall}{\text{Pr}\,ecision + recall} \qquad \text{(eq. 9)}$$

Las métricas anteriormente mencionadas permitirán determinar, según sus valores, la capacidad del modelo para realizar las mejores predicciones posibles en el proceso de validación, así se podrá realizar la implementación y comparación de diferentes modelos con el fin de decidir cuál tiene la mayor capacidad de predecir el comportamiento que se está estudiando [19].

5. Base de datos

La información clínica electrónica (EHR, por sus siglas en inglés) se ha convertido en información indispensable para los investigadores en salud, ya que estos aportan información y registros enfocados en el paciente, tales como información demográfica, diagnósticos médicos, resultados de pruebas de laboratorio, planes de tratamiento y medicamentos [20] de un hospital de tercer nivel de la ciudad de Medellín, Antioquia.

Para el presente trabajo, los datos electrónicos fueron otorgados en formato csv. Estos datos contienen la información de los episodios del hospital de tercer nivel, registrados entre octubre de 2017 y marzo de 2023. La base de datos se compone de 167.013 registros totales correspondientes a 109.761 pacientes únicos. Cada registro tiene asociadas 40 variables las cuales hacen referencia a información general, información de ingreso, información de estancia e información de egreso. De esta se excluyen todos aquellos episodios cuyo diagnóstico principal pertenezca al COVID-19, ya que estos se pueden considerar como datos atípicos. Además, sólo se trabajará con aquellos episodios que presenten reingreso, por lo tanto, se tomaron únicamente los registros de pacientes que presentaron reingreso hospitalario. En esta base de datos los ingresos con esta característica equivalen a un total de 28324 los cuales representan un 17 % de los registros totales como base de estudio. En la **Figura *3*** se muestra la distribución de este comportamiento.

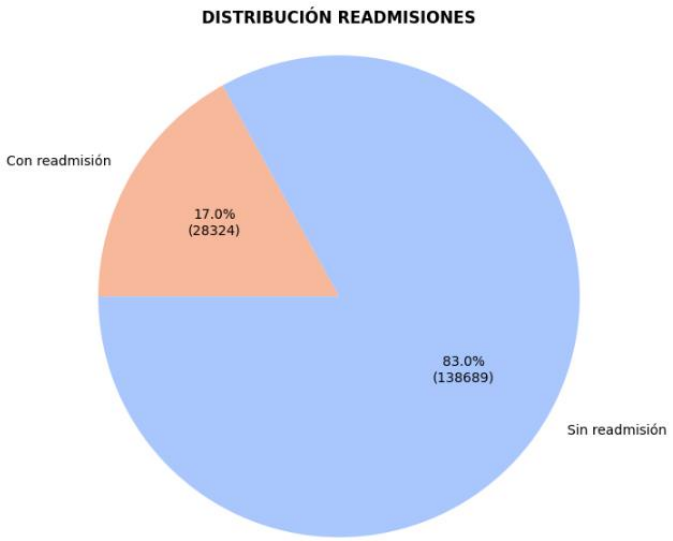

Figura 3. Distribución de pacientes sin readmisión vs pacientes con readmisión.

La **Tabla 1** muestra cada una de las variables usadas para el estudio, y seguidamente, la **Figura *4*** se observa la distribución de los tipos de variables existentes en la base de datos y su porcentaje en el total de variables presentes en el dataset.

Tabla 1. Listado variables del conjunto de datos originales.

Variable	**No readmitidos**	**Readmisión temprana**	**Readmisión media**	**Readmisión tardía**
		Media* ± *SD		
Demográficas				
Edad	57.4 ± 24.5	59.7 ± 21.78	60.68 ± 19.82	62.28 ± 21.56
Hombre	0.51 ± 0.50	0.52 ± 0.50	0.52 ± 0.50	0.49 ± 0.50
Mujer	0.49 ± 0.50	0.48 ± 0.50	0.48 ± 0.50	0.51 ± 0.50
Médicas				
Días de estancia	8.24 ± 10.36	9.05 ± 10.54	9.43 ± 10.89	8.68 ± 10.41
Diagnóstico principal				
Clasificación crónica dx principal*	0.46 ± 0.50	0.58 ± 0.49	0.58 ± 0.49	0.65 ± 0.48
Número de visitas	0.33 ± 1.02	2.19 ± 2.27	2.49 ± 2.48	2.46 ± 2.37
Número de diagnósticos	4.42 ± 1.27	4.63 ± 1.19	4.49 ± 1.18	4.67 ± 1.13
Número de procedimientos	6.23 ± 30.09	6.16 ± 29.39	8.44 ± 39.44	8.31 ± 34.11
Causas de ingreso**				
Causa ingreso medicina interna	0,21 ± 0.40	0,21 ± 0.41	0,20 ± 0.40	0,3 ± 0.46
Causa ingreso medicina general	0,02 ± 0.13	0,03 ± 0.18	0,02 ± 0.15	0,03 ± 0.16
Causa ingreso ortopedia	0,10 ± 0.29	0,06 ± 0.23	0,07 ± 0.26	0,06 ± 0.24
Causa ingreso especialidad pediátrica	0,06 ± 0.23	0,03 ± 0.18	0,02 ± 0.13	0,04 ± 0.19
Causa ingreso especialidad oncológica	0,02 ± 01.4	0,03 ± 0.17	0,06 ± 0.24	0,04 ± 0.19
Causa ingreso otra especialidad	0,18 ± 0.39	0,18 ± 0.39	0,26 ± 0.44	0,24 ± 0.43
Causa ingreso especialidad quirúrgica	0,40 ± 0.49	0,43 ± 0.50	0,35 ± 0.48	0,29 ± 0.45
Causa ingreso apoyo terapéutico	0.00 ± 0.02	0.00 ± 0.00	0.00 ± 0.03	0.00 ± 0.02
Causa ingreso trasplante	0,01 ± 0.08	0,02 ± 0.13	0,02 ± 0.13	0,01 ± 0.11
Procedencias***				

Procedencia general adultos	0,80 ± 0.40	0,79 ± 0.40	0,86 ± 0.35	0,79 ± 0.41
Procedencia urgencia mayor 24 años	0,12 ± 0.33	0,14 ± 0.35	0,1 ± 0.30	0,14 ± 0.35
Procedencia urgencia menor 24 años	0,01 ± 0.09	0,02 ± 0.13	0,01 ± 0.09	0,01 ± 0.1
Procedencia pediatría	0,04 ± 0.19	0,03 ± 0.16	0,01 ± 0.10	0,03 ± 0.16
Procedencia UCI	0,02 ± 0.15	0,02 ± 0.13	0,01 ± 0.12	0,02 ± 0.14
Procedencia UCE	0,01 ± 0.11	0,01 ± 0.09	0,01 ± 0.1	0,01 ± 0.11
Procedencia cirugía	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.01	0.00 ± 0.01
Alta el mismo día	0.004 ± 0.07	0.01 ± 0.08	0.004 ± 0.07	0.006 ± 0.08
Transfusión	0.10 ± 0.30	0.14 ±0.35	0.15 ± 0.36	0.12 ± 0.32
Antibióticos	0.46 ± 0.50	0.50 ±0.50	0.54 ± 0.50	0.46 ± 0.50
Existe hemograma****	0.16 ± 0.37	0.16 ± 0.37	0.17 ± 0.38	0.21 ± 0.41

** Indica binariamente si el diagnóstico principal se puede clasificar como crónico o no crónico.*
***Indica las diferentes causas por las cuáles se da ingreso al episodio (9 en total).*
****Indica la procedencia desde la cual llega el episodio (7 en total).*
*****Cantidad de pacientes que presentan pruebas de hematocrito.*

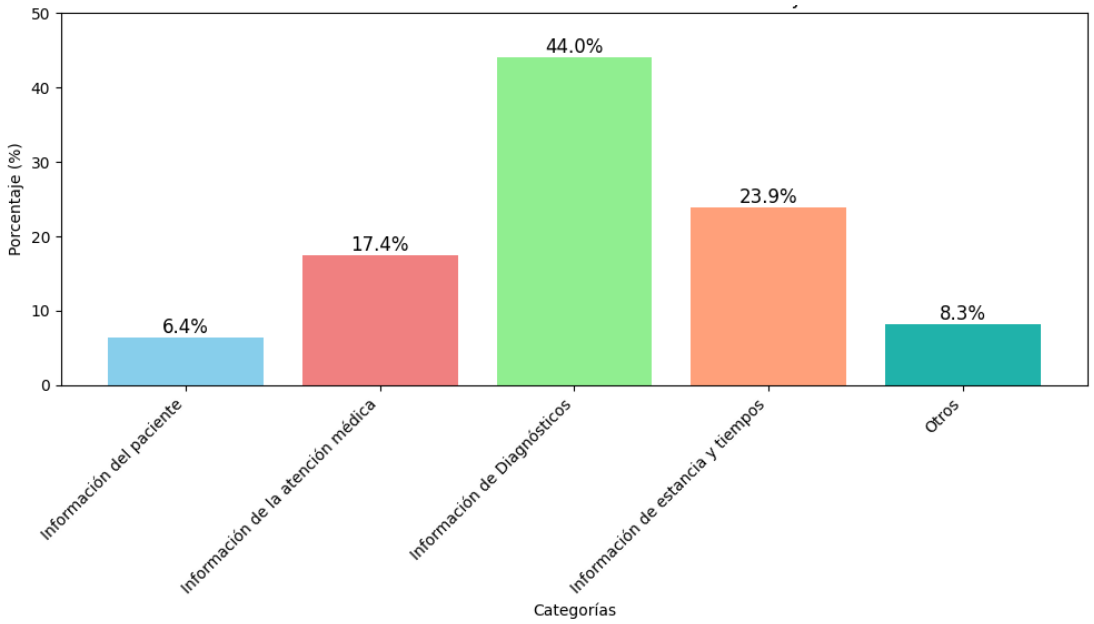

Figura 4. Distribución de los tipos de variables.

Como se puede observar en la **Tabla 1**, para los pacientes readmitidos, en promedio un 60,3 % de los episodios que presentan readmisión tienen un diagnóstico clasificado como crónico entre los cuales aquellos que presentan readmisión tardía son los que muestran un mayor porcentaje. También es importante resaltar, según la distribución de edades, el centro de salud se puede

catalogar como hospital geriátrico ya que la mayoría de los episodios oscilan entre los 60 y 80 años (ver **Figura 5** y **Figura 6**).

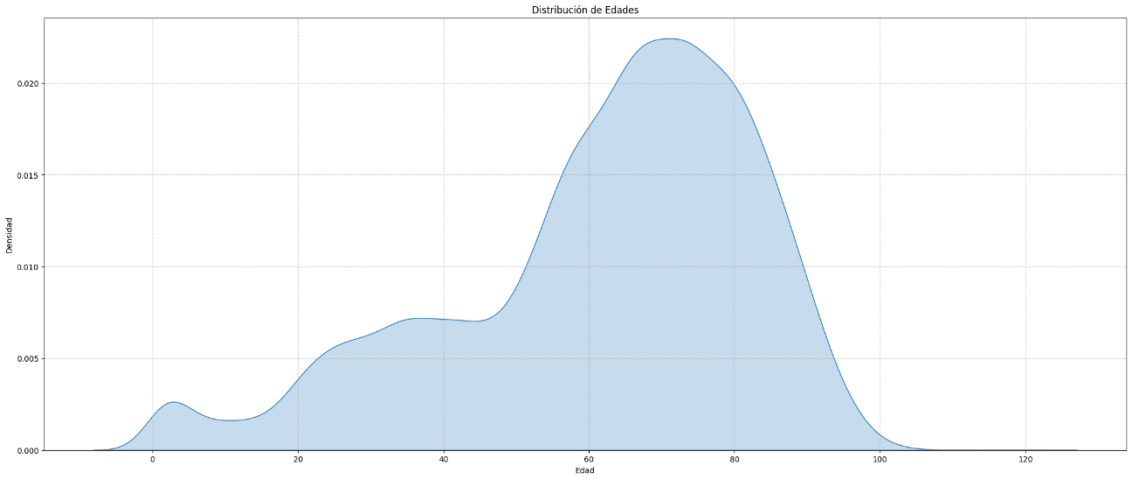

Figura 5. *Distribución de edades de los episodios.*

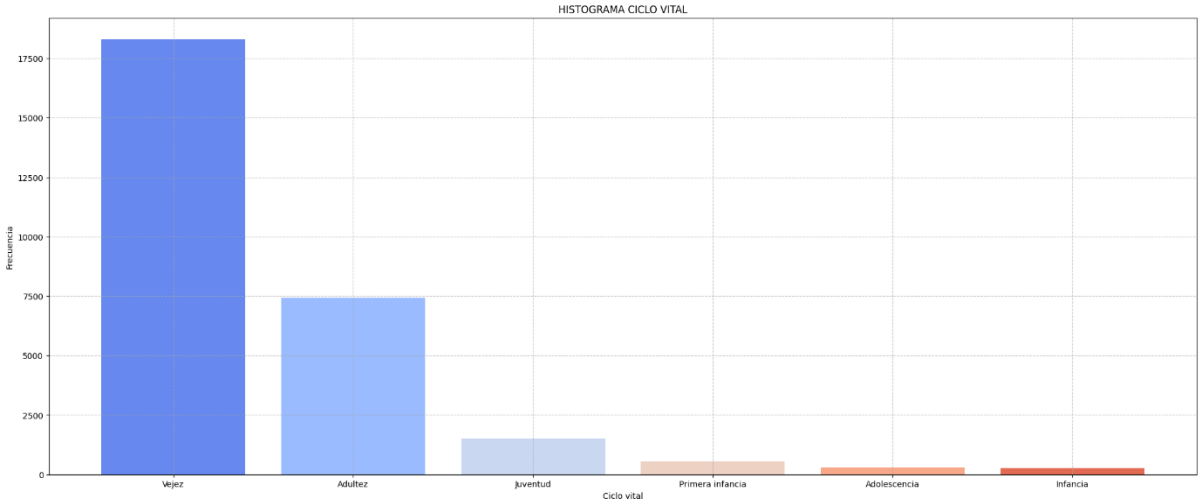

Figura *6.* Distribución ciclo vital de los episodios.

6. Metodología

Para desarrollar la efectividad un modelo de aprendizaje automático es esencial seguir un proceso meticuloso que abarque desde la recopilación inicial de datos hasta la evaluación de las métricas resultantes del modelo implementado. Cada etapa de este proceso implica mejorar el rendimiento del modelo y garantizar que este refleje con precisión el fenómeno que se está analizando. Es por esto que el preprocesamiento y análisis exploratorio de los datos tienen un valor indispensable para garantizar data de alta calidad permitiendo que el modelo de predicción se desarrolle de la manera más correcta y efectiva posible. En la **Figura 7** se observa el flujo de procesos a seguir para el desarrollo de las etapas del proceso que permiten obtener y diseñar los modelos.

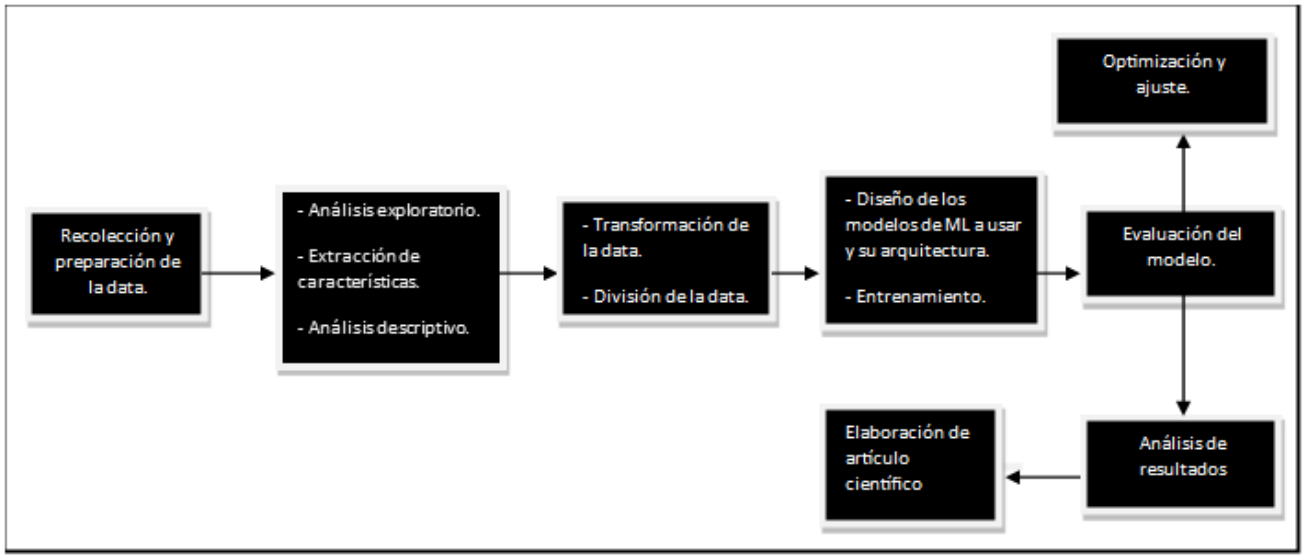

Figura 7. *Diagrama de flujo metodológico para el desarrollo del proyecto*

Se busca diseñar un modelo que tenga la capacidad de predecir la readmisión en tres ventanas de tiempo diferente, es por esto por lo que la variable respuesta del problema tendrá tres opciones a predecir. En la **Tabla 2** se muestra las tres opciones de variable respuesta en el dataset y una descripción temporal de la misma.

Tabla 2. Variables respuesta de la base de datos

Variables respuesta	Descripción
Readmisión temprana	Indica si el paciente presenta un reingreso entre las primeras 72 horas luego de ser dado de alta del mismo centro hospitalario y por el mismo diagnóstico principal.
Readmisión media	Indica si el paciente presenta un reingreso entre 72 horas y 28 días luego de ser dado de alta del mismo centro hospitalario y por el mismo diagnóstico principal
Readmisión tardía	Indica si el paciente presenta un reingreso mayor a 28 días luego de ser dado de alta del mismo centro hospitalario y por el mismo diagnóstico principal

6.1. Recolección de la información

La base de datos usada para este estudio fue facilitada por el Hospital de alta complejidad de la ciudad de Medellín. Este dataset presenta un total de 167.013 ingresos, la cual, además tiene información demográfica, estancia, ingreso y egreso, también lectura de exámenes de laboratorio.

6.2. Análisis exploratorio

Los datos del estudio presentan información solo de pacientes que readmitidos (28.324), es decir personas que tuvieron uno o más reingresos. Para este caso se realizan diferentes estrategias que permiten identificar los tipos de variables y el comportamiento de estas en términos temporales y relacionales, se analizan correlaciones y diferentes gráficos que permitan evidenciar las diferentes características y los diferentes casos a analizar para tener una base de datos adecuada para el desarrollo de los modelos a implementar. Es así como se hace un análisis de valores nulos en cada una de las variables, outliers (valores atípicos) y valores temporales. Se utilizaron diferentes métodos de codificación para las variables, según fuese necesario: Label enconding y One hot enconder.

6.3. Análisis del desbalance de datos

En muchos sistemas de aprendizaje automático, se asume que el conjunto de datos de entrenamiento tiene una distribución equilibrada de clases, es decir, hay una cantidad similar de datos en cada clase. Sin embargo, en la práctica, a veces hay con conjuntos de datos donde una clase tiene muchas más instancias que otra, lo que lleva a un desequilibrio de clases. En este escenario, una clase se convierte en la clase mayoritaria y la otra en la clase minoritaria [22]. Este desequilibrio puede causar que los modelos de aprendizaje automático desarrollen un sesgo hacia la clase mayoritaria, lo que afecta negativamente su capacidad para predecir correctamente la clase minoritaria. Para abordar este problema se pueden implementar las siguientes estrategias:

- *Sobremuestreo de la clase minoritaria:* Se generan instancias adicionales de la clase minoritaria para igualar el número de instancias de la clase mayoritaria.

- *Submuestreo de la clase mayoritaria:* Se reducen el número de instancias de la clase mayoritaria para que coincida con el número de instancias en la clase minoritaria.

- *Ignorar instancias de la clase minoritaria:* Algunos enfoques optan por ignorar completamente las instancias de la clase minoritaria y se centran en reconocer patrones comunes en lugar de tratar de discriminar entre clases [22].

Ahora, una técnica ampliamente utilizada para abordar el desequilibrio de clases por medio del sobremuestreo es SMOTE (Synthetic Minority Over-sampling Technique). SMOTE genera instancias sintéticas de la clase minoritaria tomando cada muestra de la clase minoritaria y creando ejemplos sintéticos a lo largo de los segmentos de línea que conectan sus vecinos más cercanos de la misma clase, logrando así disminuir el desbalanceo de clases entre las muestras [23].

6.4. Diseño de los modelos

Ahora, con el fin de aplicar el proceso de predicción de readmisión hospitalaria, se aplican modelos tanto de Machine Learning como Deep Learning, más específicamente se usarán modelos de clasificación por medio de árboles de Decisión (Incluyendo Random Forest), además de máquinas de soporte vectorial. Mientras que para el caso de aprendizaje profundo se usará el modelo de

Redes Neuronales Recurrentes debido a que se desea realizar predicciones sobre el set de datos secuencial. Las técnicas de modelado que se implementan son:

6.4.1. Árboles de decisión

Como se mencionó anteriormente, este modelo permite realizar un modelo de clasificación o regresión. Para nuestro caso se realizará el proceso de clasificación en el cual se tendrán en cuenta ciertos hiperparámetros que determinarán, según sus valores la calidad del modelo. La **Tabla 3** muestra cada uno de los hiperparámetros a usar.

Tabla 3. Listado de hiperparámetros a usar para el modelo de árbol de decisiones.

Hiperparámetro	Descripción
criterion	Permite determinar la función de medida que indica la calidad con que un nodo del árbol se divide en su proceso de construcción.
max_depth	Permite determinar la profundidad que tomará el árbol de decisión. El objetivo de su uso es generar una poda para evitar el sobreajuste en el proceso de entrenamiento.
min_samples_split	Se usa para definir el mínimo número de muestras que se debe tener en un nodo para que este pueda ser dividido en sus dos nodos hijos durante la construcción del árbol
min_samples_leaf	Se usa para determinar el número mínimo de muestras que debe tener un nodo para considerarse una hoja en lugar de nodo interno.

6.4.2. Bosques Aleatorios

Se usará este modelo con el objetivo de mejorar la capacidad de predicción con el árbol de decisiones. La **Tabla 4** muestra los hiperparámetros que se tendrán en cuenta para la construcción del modelo. Es importante tener en cuenta que a mayor número de *n_estimators* se puede mejorar la capacidad de generalización del modelo y evitar el sobreajuste; pero este a su vez puede tener

una repercusión en los tiempos y recursos en el proceso de entrenamiento (y en algunas ocasiones no genera mejoras muy significativas).

Teniendo en cuenta la aplicación de los modelos y los hiperparámetros a tener en cuenta, el paso más importante es definir qué valor de cada uno de estos según las mejores métricas de evaluación (accuracy, recall, F1-score). Es por lo anterior que para todos los modelos se aplica el proceso de GridSearch en Python, el cual se encarga de determinar cuál es la mejor combinación posible de un conjunto de opciones las cuales son entregadas manualmente, lo anterior para generar las mejores métricas de predicción.

Tabla 4. Listado de hiperparámetros a usar para el modelo de Random Forest

Hiperparámetro	**Descripción**
criterion	Permite determinar la función de medida que indica la calidad con que un nodo de cada uno de los árboles del bosque se divide en su proceso de construcción.
n_estimators	Indica el número de árboles que se construirán en el bosque.
max_depth	Permite determinar la profundidad que tomará cada árbol. El objetivo de su uso es generar una poda para evitar el sobreajuste en el proceso de entrenamiento.
min_samples_split	Se usa para definir el mínimo número de muestras que se debe tener en un nodo para que este pueda ser dividido en sus dos nodos hijos durante la construcción del árbol
min_samples_leaf	Se usa para determinar el número mínimo de muestras que debe tener un nodo para considerarse una hoja en lugar de nodo interno.

6.5. Evaluación del modelo

Al momento de implementar un modelo de Machine Learning es indispensable analizar los resultados los resultados en la clasificación del modelo. Es por esto que la matriz de confusión es la herramienta más popular y simple para visualizar el rendimiento del modelo comparando las predicciones con las clases reales de los datos. Por lo tanto, se usará este modelo para presentar visualmente el comportamiento y determinar el nivel de calidad de predicción de los modelos

implementados y por medio de esta y de sus valores determinar cuál o cuáles modelos tienen una mejor capacidad de predicción, tales como el accuracy, el recall y el F1 Score.

7. Análisis y resultados

7.1.Base de datos

En este estudio no se analizan 5.981 episodios correspondientes a COVID 19, es decir, que el dataset final tiene un total de 28.324 episodios en donde se tiene información de pacientes entre los 0 y los 119 años (tal como se observó en la **Figura 5**), de los cuales el 50.2 % y 49.8 % pertenecen a pacientes de género masculino y femenino, respectivamente (ver **Error! Reference source not found.**).

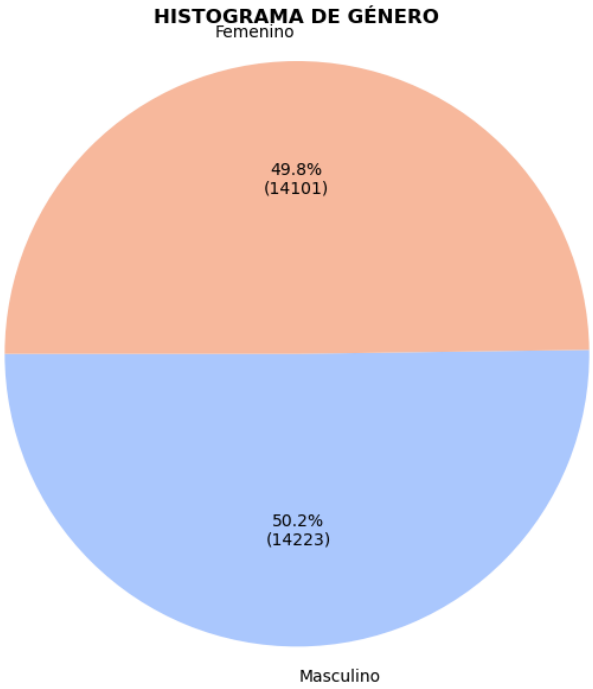

Figura 8. *Distribución de género para pacientes en readmisión.*

La variable respuesta tiene tres categorías a evaluar en el modelo, y como es de esperarse, la probabilidad de presentar un reingreso es directamente proporcional al número de días transcurridos luego del egreso del centro hospitalario. Esto indica que la mayor cantidad de los datos se concentra en reingresos tardíos, luego reingresos medios, y en la menor cantidad reingresos tempranos. En la **Figura *9*** se presenta un histograma que representa la cantidad de episodios por cada tipo de reingreso. En la misma figura se puede observar el alto desbalance que se tiene sobre la clase mayoritaria y la clase minoritaria (readmisión tardía y readmisión temprana). Es por lo

anterior, que se presenta la necesidad realizar balance de clases para evitar un sesgo en el modelo al momento de realizar el entrenamiento y validación.

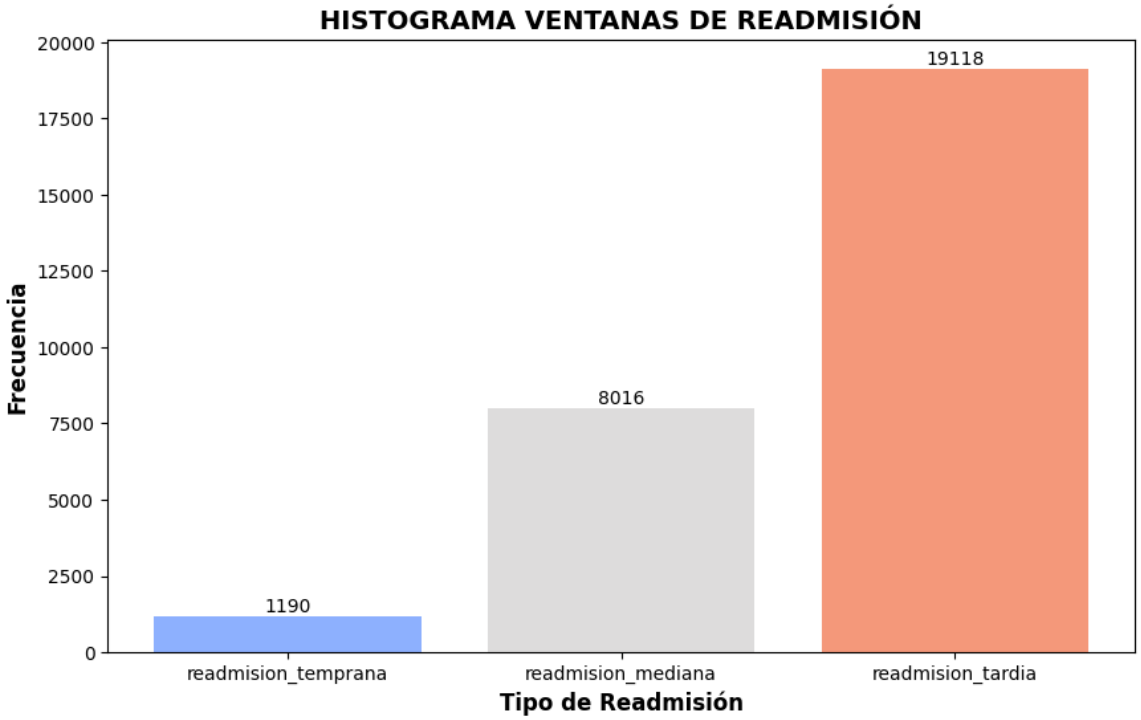

Figura 9. *Distribución de las variables respuesta de la base de datos.*

La **Figura *9*** muestra claramente la gran diferencia en la frecuencia entre cada variable, por lo que se insiste en la aplicación de algún método de sobremuestreo o submuestreo con el objetivo de equilibrar esta desproporción y poder equiparar la base de datos con el fin de tener predicciones con buenas métricas. Por otra parte, al analizar cómo se correlacionan las variables continuas con la variable respuesta (readmisión temprana, readmisión media, readmisión tardía), en la **Figura 10** se puede observar que no hay ningún tipo de comportamiento lineal, es decir no hay linealidad entre las variables analizadas. Además, se puede observar una superposición entre las clases (variables respuesta), lo que implica que el modelo puede presentar dificultad para identificar la clase minoritaria inmersa entre las dos clases restantes.

Con base en lo anterior, se puede concluir que no existe un comportamiento proporcional lineal marcado dado que ninguno de los valores de correlación obtenidos supera 0.3, tal como se evidencia en los diagramas de dispersión (ver **Figura 11**).

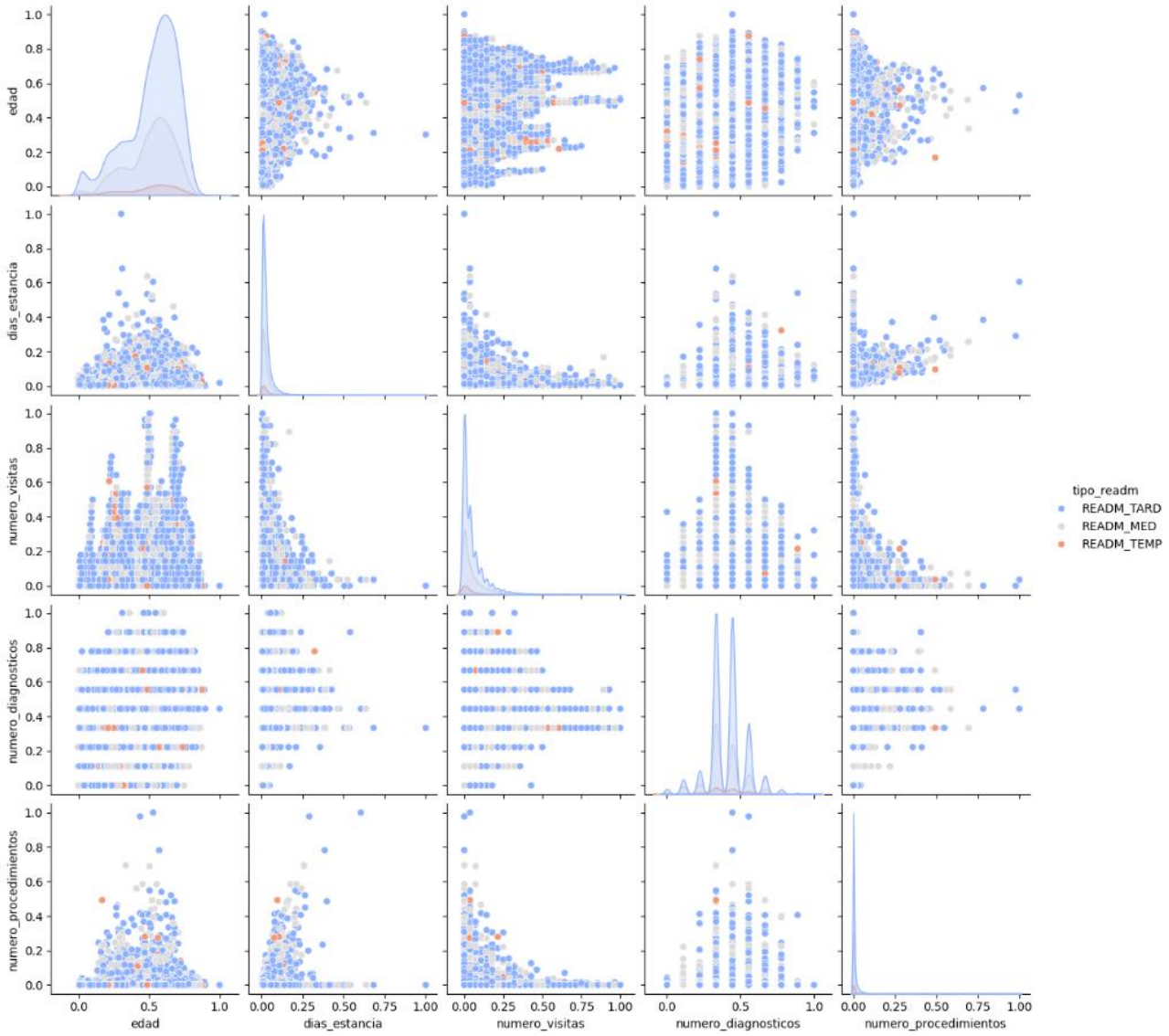

Figura 10. *Matriz de dispersión para las variables continuas del dataset.*

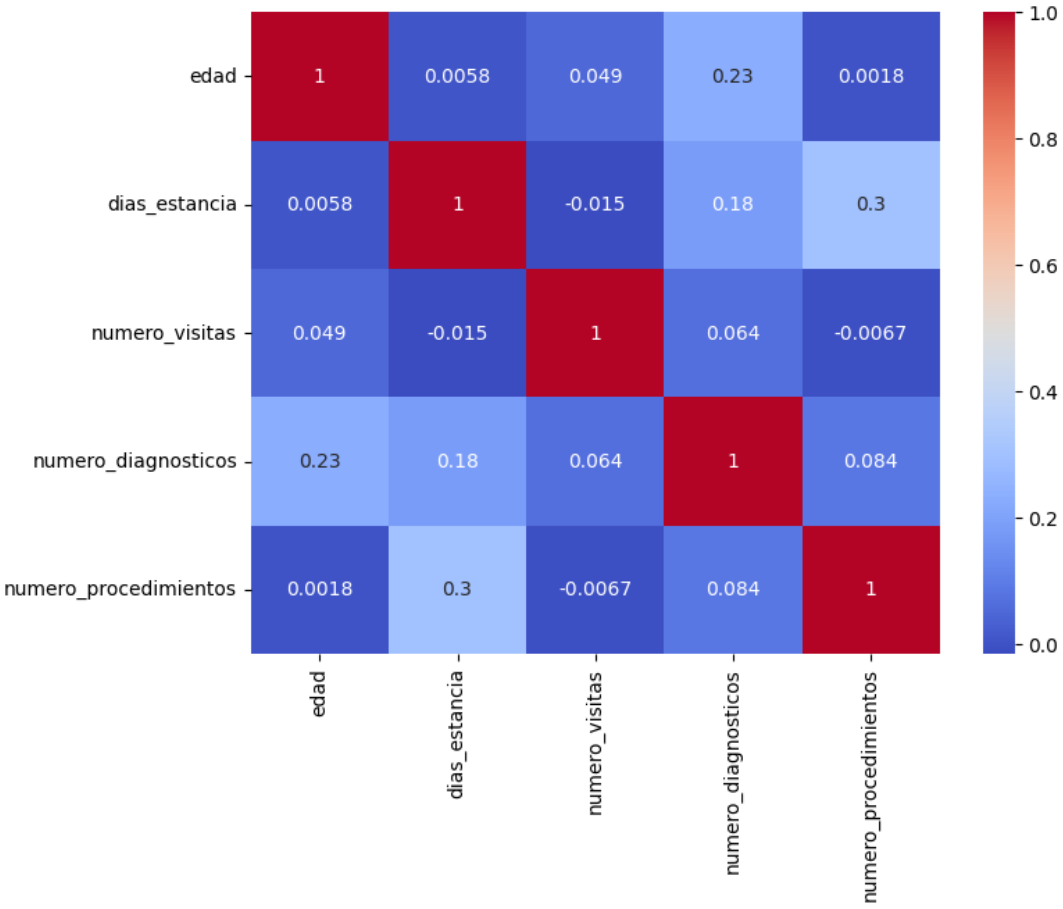

Figura 11. *Diagrama de correlación para las variables continuas del dataset.*

7.2. Tratamiento del desbalance de clases

Como se observó en la **Figura *9***, se presenta un desbalance marcado entre la clase mayoritaria y la minoritaria en las variables respuesta. Para este caso se aplica la técnica del submuestreo y el sobremuestreo con el fin de definir cuál de estas genera un conjunto de datos que presente mejores resultados en la aplicación de los modelos (si lo presenta). Para el caso del sobremuestreo se aplica el método SMOTE de la librería *imblearn* de Python. Para este caso se aplica un sobremuestreo que generará datos aleatorios hasta igualar en las dos clases minoritarias la cantidad de datos presente en la variable readmisión tardía. Para el caso del submuestreo se usa el método RandomUnderSampled() de la librería Imbalanced Learn de Python, lo cual permite igualar las dos clases mayoritarias a la cantidad de datos de la clase minoritaria.

7.3. Análisis modelos

A continuación, se aplicarán diferentes modelos de clasificación con el objetivo de encontrar aquel que genera la mejor predicción para el problema readmisión hospitalaria de pacientes.

7.3.1. Árbol de decisión

Se comienza implementando el modelo de árbol de decisiones, ya que es el modelo de clasificación más simple a implementar. Para este caso se probarán tres estrategias diferentes con el fin de analizar el mayor valor del accuracy arrojado por el método del árbol de decisiones:

- Análisis con el dataset original sin balanceo.
- Análisis con dataset submuestreado.
- Análisis con dataset sobremuestreado.

En la **Tabla 5** se observan los resultados para cada uno de los tres procedimientos estudiados. Allí se puede observar un mayor valor de precisión obtenido con la base de datos sin realizar ningún procedimiento de balanceo de datos; esto se traduce en la capacidad del modelo de realizar predicciones sin necesidad de alcanzar equilibrio en la cantidad de valores de las diferentes clases. Se resalta que a la hora de subdividir los datos con train_test_split se considera el desbalance de clases con *stratify*.

Tabla 5. Valor accuracy para árbol de decisión con y sin balanceo de los datos.

Procedimiento	**Valor de precisión obtenido**
Árbol de decisión con base de datos original	0.6097
Árbol de decisión con datos submuestreados	0.4115
Árbol de decisión con datos sobremuestreados	0.5615

Posteriormente se inicia la búsqueda de los mejores hiperparámetros apoyados en GridSearch con el objetivo de obtener la mejor capacidad de predicción para este modelo (*'criterion'= 'gini', 'max_depth'=6, 'min_samples_leaf'=5, 'min_samples_split'=10*). Así, obteniendo estos hiperparámetros se aplica el proceso de modelado obteniendo las métricas: *accuracy=0.6983, recall=0.7, F1 Score=0.64.* La **Figura 12** presenta la matriz de confusión obtenida por el modelo

de árbol de decisión implementado. Esta presenta la capacidad de predicción para las 3 clases (variables respuesta) en donde 0: Readmisión Mediana, 1: Readmisión Tardía, 2: Readmisión Temprana.

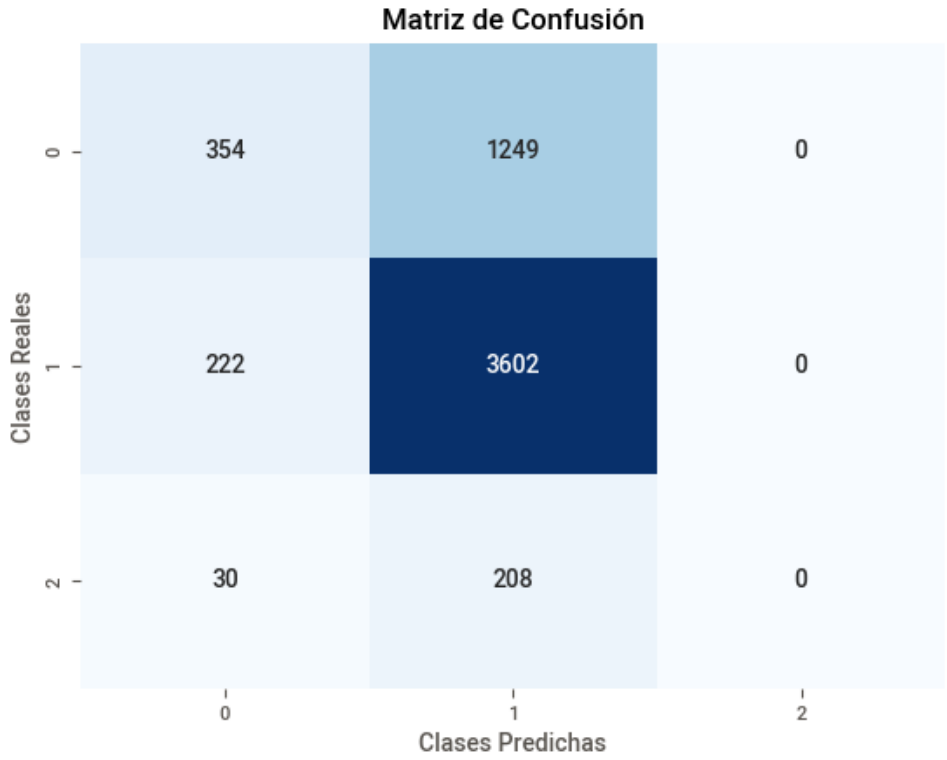

Figura 12. *Matriz de confusión modelo árbol de decisión.*

En la matriz de confusión presentada en la **Figura 12** se puede observar una nula capacidad del modelo para predecir readmisiones tempranas (entre 0 y 3 días) y una alta capacidad para la predicción de readmisiones tardías (entre 3 y 28 días). Esto se debe principalmente al fenómeno expuesto y mencionado con base en la **Figura 10** en donde se observa un alta dispersión y superposición de los datos correspondientes a la clase minoritaria, que en este caso sería la del bajo poder de predicción, además cabe resaltar que como es de esperarse, la clase mayoritaria contiene más del 67 % de todos los datos totales, por lo cual la probabilidad de predecir con mayor precisión esta clase aumenta.

7.3.2. Bosques aleatorios

Para la aplicación de este modelo se aplica exactamente el mismo proceso del modelo de árboles de decisión, para esto se plica igualmente un proceso de GridSearch con el objetivo de encontrar los mejores hiperparámetros (*'bootstrap'=True, 'criterion'='gini', 'max_depth=None', 'max_features'='auto', 'min_samples_leaf'=8, 'min_samples_split'=8, 'n_estimators'=67*). Obteniendo así las métricas: *accuracy=0.699, recall=0.7, F1 Score=0.64.*

Como se puede observar, el modelo de Bosques Aleatorios no mejoró significativamente el modelo de árboles de decisión debido a que las métricas de predicción fueron prácticamente idénticas, cuyo motivo se puede dar debido al proceso de bootstrapping en donde los datos de los árboles tomados pueden contener, al igual que el modelo anterior, la mayoría de variables respuesta de la clase mayoritaria.

A continuación, se muestra la matriz de confusión en la **Figura 13.** Con esta se puede observar con más claridad lo antes expuesto, en donde las predicciones de cada una de las variables respuesta son muy similares a las predicciones realizadas por el modelo de árbol de decisión. Por lo tanto, se puede concluir que el modelo de bosques aleatorios no mejora el modelo de árbol de decisión. Luego y se procede a implementar el modelo de Deep Learning para analizar si este puede aportar una mejor capacidad de predicción.

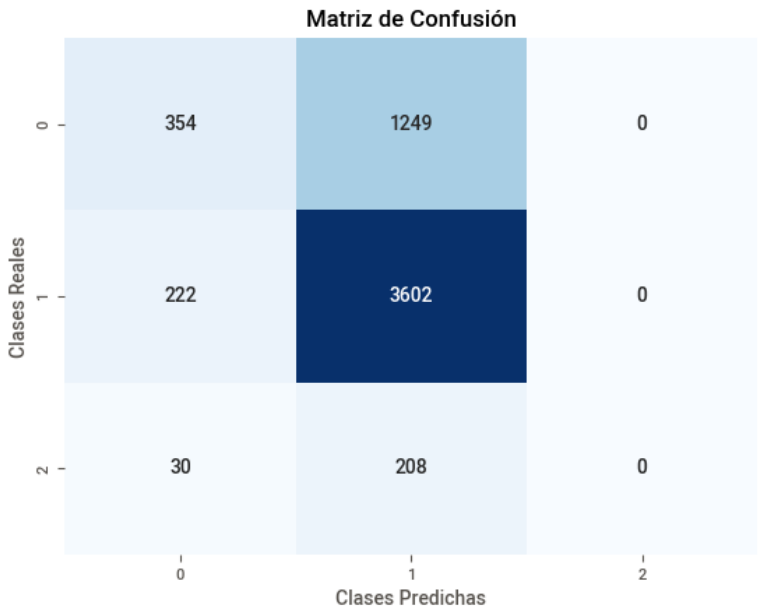

Figura 13. *Matriz de confusión modelo de Bosques Aleatorios.*

7.3.3. RNN

Para este proceso, al igual que los dos modelos implementados previamente, se aplicará un proceso de GridSearch con método de validación cruzada, con el fin de encontrar el mejor número de neuronas en la capa oculta del Perceptrón Multicapa, para así maximizar la precisión del modelo. Para este caso se usarán 1, 5, 10, 15 y 25 neuronas usando el Accuracy como evaluador del rendimiento del modelo. En la **Figura 14** se puede observar gráficamente los resultados de este proceso.

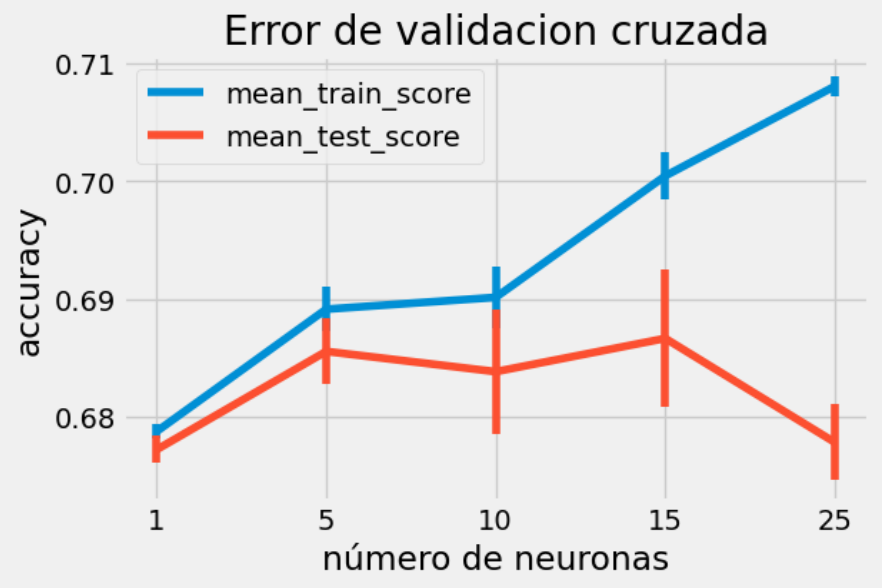

Figura 14. *Error de validación cruzada en el método GridSearchCV para número de neuronas.*

A continuación, se usará el mismo proceso de grid search con el objetivo de ajustar la tasa inicial de aprendizaje. Lo anterior con el objetivo de controlar la magnitud de los ajustes que se dan a los pesos de la red neuronal durante el proceso de entrenamiento de la misma para así determinar desde el principio qué tan grandes serán los pasos que se darán en dirección opuesta al gradiente durante el ajuste de los pesos de la red neuronal. Para este caso se usarán probarán los valores: 0.0001, 0.001, 0.01, 0.1, 1, 10 y 100. Además, se usará el Accuracy para evaluar el rendimiento del modelo. Cabe resaltar que los valores de la tasa de aprendizaje se distribuyen de manera uniforme en una escala logarítmica, ya que aquí los valores del hiperparámetro varían en orden de magnitud (de 10 en 10), así se permite una mejor visualización de cómo afecta el rendimiento del modelo en todo el rango de valores (ver **Figura 15**).

En la **Figura 15** se muestra cómo cambia la precisión del modelo conforme se varía el número de neuronas en la capa oculta de la red neuronal, aquí la línea de color azul nos muestra cómo va aumentando la precisión con los datos de entrenamiento conforme se aumenta el número de neuronas, mientras la línea naranja nos da la misma información para los datos de prueba. Se puede observar claramente que para un número de 15 neuronas en la capa oculta se obtiene la mayor precisión en los datos de prueba y una gran precisión en los datos de entrenamiento, por lo cual este sería el valor óptimo de número de neuronas a usar para la implementación de este modelo.

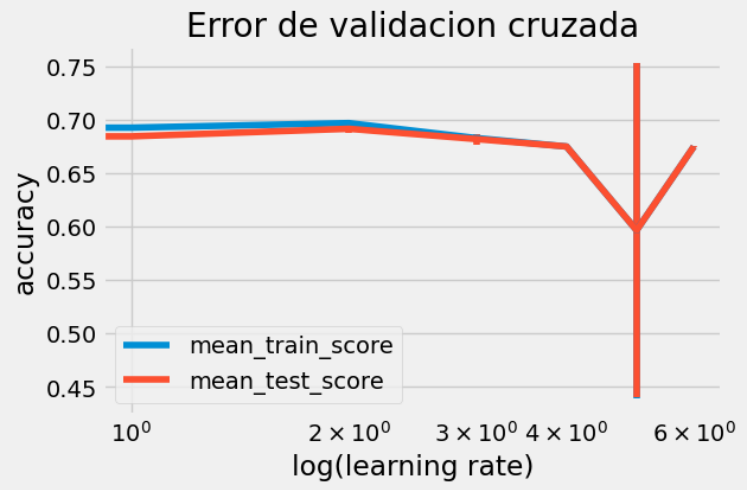

Figura 15. Error *de validación cruzada en el método GridSearchCV para tasa de aprendizaje inicial.*

La **Figura 15** indica el comportamiento de la precisión de la red neuronal conforme se aumenta la tasa inicial de aprendizaje en base logarítmica. Aquí se puede ver el comportamiento de la precisión del modelo conforme se aumenta el valor de la tasa inicial de aprendizaje, en donde la mejor precisión se alcanza en 100, por lo cual este es el valor de tasa inicial de aprendizaje que puede proporcionar la mejor efectividad de este modelo. En la **Tabla 6** se puede observar el resultado de la búsqueda aleatoria de hiperparámetros para encontrar los mejores parámetros para el modelo utilizando validación cruzada, mostrando a su vez los mejores resultados obtenidos.

Tabla 6. Análisis de rendimiento de diferentes combinaciones de hiperparámetros en el modelo de Red Neuronal.

Tasa de aprendizaje inicial	Tamaño de capas ocultas	Parámetro de Regularización	Precisión media en datos de prueba	Desviación estándar en datos de prueba	Precisión media en datos de entrenamiento	Desviación estándar en datos de entrenamiento
0.01	10	0.001	0.683166	0.001880	0.700607	0.000347
0.01	10	0.01	0.682566	0.006168	0.698065	0.005142
0.001	10	0.001	0.681260	0.004706	0.699213	0.002016
0.01	(10, 10)	1.0	0.681260	0.003929	0.701949	0.002375
0.01	(10, 10)	10.0	0.681118	0.000476	0.701102	0.001471
0.1	10	0.1	0.681083	0.004592	0.699636	0.001404
0.001	(10, 10)	1.0	0.681083	0.004693	0.703855	0.003958
0.1	(10, 10)	0.001	0.680977	0.003768	0.701243	0.000750
0.001	(10, 10)	0.001	0.680836	0.005457	0.703591	0.002957
0.001	10	1.0	0.680483	0.007718	0.700148	0.003331

Observando la **Tabla 6** se puede inferir que los modelos con tasas de aprendizaje inicial de 0.01 y 0.001 tienden a tener una precisión media en los datos de prueba ligeramente más alta que los modelos con una tasa de aprendizaje de 0.1. Esto podría indicar que las tasas de aprendizaje más bajas son más efectivas para este problema de predicción de readmisiones. Además, la mayoría de los modelos tienen una precisión media en los datos de prueba que varía entre aproximadamente 0.6808 y 0.6832. Esto sugiere que las diferentes configuraciones de hiperparámetros no tienen un impacto significativo en el rendimiento del modelo por lo cual se pueden implementar hiperparámetros que aporten mayor eficiencia en tiempo y gasto computacional.

8. Conclusiones

- Los árboles de decisión y bosques aleatorios muestran un rendimiento similar en términos de precisión (accuracy), recall y F1 Score. Ya que ambos alcanzan una precisión de alrededor del 69.8-69.9%, un recall de 70%, y un F1 Score de 0.64. Así se puede inferir que en términos de eficiencia el árbol de decisión, en comparación con el bosque aleatorio que es más robusto, es un modelo más atractivo para este tipo de análisis
- Las Redes Neuronales Recurrentes (RNN) muestran una precisión media ligeramente superior en los datos de prueba, oscilando entre 68.08% y 68.32%.
- La consistencia en las métricas de entrenamiento y prueba para las RNN, ya que las desviaciones estándar son relativamente bajas. Esto sugiere que el modelo tiene una buena capacidad de generalización evitando que se realice un sobreajuste sobre los datos de entrenamiento.
- Las métricas de precisión no son superiores al 80%, por lo cual se puede mejorar esta respuesta considerando el análisis de importancia de las variables de entrada en la relación a la variable de interés de la base de datos implementada.

Referencias

[1] Caballero, A., Pinilla, M. I., Mendoza, I. C. S., & Peña, JRA. (2016). "Frecuencia de reingresos hospitalarios y factores asociados en afiliados a una administradora de servicios de salud en Colombia". Cadernos De Saúde Pública, 32(7), e00146014. https://doi.org/10.1590/0102-311X00146014

[2] Epstein AM. "Revisiting readmissions - changing the incentives for shared accountability". N Engl J Med 2009; 360:1457-9.

[3] Jencks SF, Williams MV, Coleman EA. "Rehospitalizations among patients in the Medicare fee-for-service program". N Engl J Med 2009; 360:1418-28.

[4] Hansen LO, Young RS, Hinami K, Leung A, Williams MV. "Interventions to reduce 30-day rehospitalization: a systematic review". Ann Intern Med 2011; 155:520-8.

[5] Serna, Natalia, Riascos, Álvaro. Granados, Marcela. Predicción de Readmisiones, Mortalidad e Infecciones en la UCI usando Técnicas de Aprendizaje de Máquina. Tomado el día 22 de enero del año 2024 de la página web: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.alvaroriascos.com/researchDocuments/healthEconomics/WorkingPaperSpanish.pdf

[6] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill, New York.

[7] Ramírez H. Denniye, Cárdenas Juan M., "El Machine Learning a Través de los tiempos y los Aportes a la Humanidad". Universidad Libre Seccional Pereira. Facualtad de Ingenierías. Pereira-2018.

[8] Arana, C. (2021). *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales* (No. 797). Serie Documentos de Trabajo.

[9] Campo Félix, Álvarez Daniel. "Validación de un Modelo Predictivo de Reingreso Hospitalario en Pacientes Ingresados por Exacerbación de Enfermedad Pulmonar Obstructiva Crónica". Universidad de Valladolid 2023. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://uvadoc.uva.es/bitstream/handle/10324/60239/TFG-M2877.pdf?sequence=1&isAllowed=y (Acceso: 15/09/2023).

[10] Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. *Nature biotechnology*, *26*(9), 1011-1013.

[11] Martínez, R. E. B., Ramírez, N. C., Mesa, H. G. A., Suárez, I. R., Trejo, M. D. C. G., León, P. P., & Morales, S. L. B. (2009). Decision trees as a tool in the medical diagnosis. *Revista médica de la Universidad Veracruzana*, *9*(2), 19-24.

[12] Hastie, T., Friedman, J., y Tibshirani, R. (2001). The Elements of Statistical Learning. Nueva York, Estados Unidos: Springer New York. DOI: 10.1007/978-0-387-21606-5

[13] Ho, T.K. (1995). Random Decision Forests. Proceedings of 3rd International Conference on Document Analysis and Recognition, 278-282. Canada. Piscataway: IEEE.

[14] Roya Najafi-Vosough, Javad Faradmal, Seyed Kianoosh Hosseini, Abbas Moghimbeigi, Hossein Mahjub. "Predicting Hospital Readmission in Heart Failure Patients in Iran: A Comparison of Various Machine Learning Methods". Healthc Inform Res. 2021 October;27(4):307-314. https://doi.org/10.4258/hir.2021.27.4.307 pISSN 2093-3681 • eISSN 2093-369X (Acceso: 09/10/2023).

[15] Itchhaporia Dipti, Snow B. Peter, ALmassy J. Robert, Oetgen J. William. "Artificial Neural Networks: Current Status in Cardiovascular Medicine". Journal of the American College of Cardiology.

[16] Calvo, Jorge. "Create Neural Networks from Mathematics". European Valley Institute. 2020. Tomado el día 01 febrero de la página web: https://www.europeanvalley.es/noticias/crear-red-neuronal-desde-las-matematicas/

[17] Cruz, I. B., Martínez, S. S., Abed, A. R., Ábalo, R. G., & Lorenzo, M. M. G. (2007). Redes neuronales recurrentes para el análisis de secuencias. *Revista Cubana de Ciencias Informáticas*, *1*(4), 48-57.

[18] García Laura, Ibarreta Carlos. "El Deep Learning: una Perspectiva General y su Aplicación en el Campo del Healthcare". Facultad de Ciencias Económicas y Empresariales. Universidad Pontificia Comillas. Madrid- 2022. Clave: 201702688.

[19] Bhargava K Reddy, Dursun Delen. "Predicting hospital readmission for lupus patients: An RNN-LSTM-based deep-learning methodology". Department of Management Science and Information Systems, Spears School of Business, Oklahoma State University, Tulsa, OK, 74106, USA.

[20] Teo, K., Yong, C. W., Chuah, J. H., Hum, Y. C., Tee, Y. K., Xia, K., & Lai, K. W. (2023). Current trends in readmission prediction: an overview of approaches. *Arabian journal for science and engineering*, *48*(8), 11117-11134.

[21] Ortiz, J. A. P. (2002). Modelos predictivos basados en Redes Neuronales recurrentes de tiempo discreto. *Universidad de Alicante. Departamento de lenguaje y sistemas informáticos. [Documento en lınea http://www. conicyt. cIl573 Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto. pdf] [22/07/08]*.

[22] Japkowicz, Nathalie. "The class imbalance problem: Significance and strategies." Proc. of the Int'l Conf. on Artificial Intelligence. Vol. 56. 2000.

[23] Ortiz, J. A. P. (2002). Modelos predictivos basados en Redes Neuronales recurrentes de tiempo discreto. *Universidad de Alicante. Departamento de lenguaje y sistemas informáticos. [Documento en lınea http://www. conicyt. cIl573 Modelos predictivos basados en redes neuronales recurrentes de tiempo discreto. pdf] [22/07/08]*.