UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO DE UN MODELO PARA LA PREDICCIÓN DE ATAQUES CARDÍACOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

MURILLO VIVANCO CARLOS ALFREDO
INGENIERO DE SISTEMAS

MACHALA
2022

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO DE UN MODELO PARA LA PREDICCIÓN DE ATAQUES CARDÍACOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

MURILLO VIVANCO CARLOS ALFREDO
INGENIERO DE SISTEMAS

MACHALA
2022

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DISEÑO DE UN MODELO PARA LA PREDICCIÓN DE ATAQUES CARDÍACOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

MURILLO VIVANCO CARLOS ALFREDO
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 21 DE SEPTIEMBRE DE 2022

MACHALA
2022

carlosmurillo

INFORME DE ORIGINALIDAD

9%	8%	2%	4%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

ENCONTRAR COINCIDENCIAS CON TODAS LAS FUENTES (SOLO SE IMPRIMIRÁ LA FUENTE SELECCIONADA)

1%

★ hdl.handle.net

Fuente de Internet

Excluir citas	Activo	Excluir coincidencias	< 15 words
Excluir bibliografía	Activo		

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, MURILLO VIVANCO CARLOS ALFREDO, en calidad de autor del siguiente trabajo escrito titulado DISEÑO DE UN MODELO PARA LA PREDICCIÓN DE ATAQUES CARDÍACOS MEDIANTE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las dispociones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 21 de septiembre de 2022

MURILLO VIVANCO CARLOS ALFREDO
2100723283

DEDICATORIA

Este trabajo de titulación lo dedico a mis padres quienes han sido un pilar fundamental durante todo mi proceso académico y de vida, brindándome consejos y su apoyo incondicional, los mismos que me han servido para seguir adelante y ser una persona de bien y poder cumplir con mi objetivo.

Carlos Alfredo Murillo Vivanco

AGRADECIMIENTO

En primer lugar le doy gracias a Dios por brindarme salud y bienestar para culminar mis estudios académicos. Agradezco a mis padres por estar en las buenas y en las malas conmigo, brindado su afecto lo cual ha permitido que no decaída en mis estudios académicos.

También agradezco a los docentes de mi carrera quienes a través de sus enseñanzas me han brindado conocimientos para mi desarrollo profesional, a mis compañeros de curso, con los cuales he compartido buenos momentos durante los años universitarios.

Y por último a mi tutor el Ingeniero Wilmer Rivas quien ha sido un guía fundamental brindándome su tiempo para atender cualquier inquietud y avanzar de manera correcta con el desarrollo del trabajo de titulación.

Carlos Alfredo Murillo Vivanco

RESUMEN

Un ataque al corazón es la necrosis isquémica del corazón, la cual es causada normalmente cuando las arterias que lo irrigan se encuentran obstruidas. El descubrimiento temprano de dicha enfermedad permite aumentar las posibilidades de salvar las vidas de muchas personas y permitiendo que se pueda tomar medidas en el cuidado de su salud.

Con el paso de los años con la ayuda de la tecnología se han dado grandes avances dentro del área de salud, como es el caso de la Inteligencia Artificial (IA), la cual ha permitido mejorar la atención de los pacientes al agilizar los procesos y obtener una mayor precisión en los diagnósticos médicos.

Las radiografías, las preparaciones con respecto a la anatomía patológica y los exámenes médicos están siendo utilizados para ayudar en el proceso de diagnóstico y tratamiento de los pacientes mediante el uso del aprendizaje automático.

Permitiendo que de esta manera existan proyectos encaminados a indagar las aplicaciones de la Inteligencia Artificial en algunas áreas sanitarias como son: la asistencial que se encarga de la prevención de enfermedades, el diagnóstico, tratamiento y seguimiento de la salud del paciente; el área de formación continua que permite a los especialistas interactuar con entornos virtuales de entrenamiento y aprendizaje; el área de investigación, que permite buscar pruebas de la utilidad de la IA en la salud mediante el uso de información médica, ayudando a que se desarrollen programas para numerosas enfermedades; y por último el área de gestión, en la cual se analizan grandes cantidades de datos históricos, lo que facilita tratar los recursos tanto como materiales y humanos de una mejor manera, frente a alguna situación específica.

Este trabajo se encuentra orientado dentro del campo de la ciencia de datos, y consiste en el desarrollo de un modelo de predicción de ataques cardíacos, a partir de un dataset de datos clínicos recopilados de pacientes, que fue conseguido en el sitio Kaggle, el cual cuenta con 65535 registros y está compuesto de 13 variables como son: edad, género, peso, estatura, índice de masa corporal, presión arterial sistólica (alta), presión arterial diastólica (baja), colesterol, glucosa, fumar, beber, ejercitar y cardio. Y haciendo uso de técnicas de aprendizaje automático, ya que dichas técnicas permiten la identificación de patrones en los datos, lo cual permitirá que el modelo aprenda de las observaciones y lo más importante permita hacer las predicciones.

Se utilizaron algunos algoritmos de Machine Learning como por ejemplo; Logistic Regression, Random Forest, MultiLayer Perceptron, Extreme Gradient Boosting, Gaussian Naive Bayes, Support Vector Machine, Decision Tree, Gradient Boosting y Light Gradient Boosted Machine, para determinar cuál realiza una mejor predicción, para escoger el algoritmo se hizo uso de las métricas de rendimiento las cuales fueron: matriz de confusión, exactitud, precisión, sensibilidad, puntuación F1 y el área bajo la curva, dándonos mejores resultados el algoritmo de Gradient boosting, teniendo los siguientes resultados: exactitud 72,33%, precisión 74,10%, sensibilidad 69,83%, puntuación F1 71,90% y área bajo la curva 72,36%. El cual se la guardo en un archivo de extensión .pkl para utilizarlo en la creación de un dashboard en una página web, que permite el ingreso de los datos acorde a las variables del dataset que se mencionaron con anterioridad, y predecir si una persona puede padecer una enfermedad cardiaca o no, y a su vez teniendo en cuentas los datos de las variables realizar recomendaciones al paciente.

Palabras claves: aprendizaje automático, enfermedad cardiovascular, inteligencia artificial, algoritmos, modelo predictivo

Abstract

A heart attack is ischemic necrosis of the heart, which is usually caused when the arteries supplying the heart are obstructed. Early detection of this cardiovascular disease increases the chances of saving the lives of many people and allows them to take measures to take care of their health.

Over the years, with the help of technology, great advances have been made in the area of healthcare, such as Artificial Intelligence (AI), which has made it possible to improve patient care by streamlining processes and obtaining greater precision in medical diagnoses.

X-rays, preparations with respect to pathological anatomy and medical examinations are being used to assist in the process of diagnosis and treatment of patients through the use of machine learning.

Allowing in this way that there are projects dedicated to explore the applications of Artificial Intelligence in some healthcare areas such as: the care area, which is responsible for disease prevention, diagnosis, treatment and monitoring of patient health; the area of continuing education, which allows specialists to interact with virtual training and learning environments; the research area, which allows the search for evidence of the usefulness of AI in health through the use of medical information, helping to develop programs for numerous diseases; and finally the management area, in which large amounts of historical data are analyzed, which facilitates the treatment of both material and human resources in a better way, against a specific situation.

This work is oriented within the field of data science, and consists of the development of a prediction model of heart attacks, from a dataset of clinical data collected from patients, which was obtained from the Kaggle site, which has 65535 records and is composed of 13 variables such as: age, gender, weight, height, body mass index, systolic blood pressure (high), diastolic blood pressure (low), cholesterol, glucose, smoking, drinking, exercise and cardio. And making use of machine learning techniques, since these techniques allow the identification of patterns in the data, which will allow the model to learn from the observations and most importantly to make predictions.

Some Machine Learning algorithms were used such as; Logistic Regression, Random Forest, MultiLayer Perceptron, Extreme Gradient Boosting, Gaussian Naive Bayes, Support Vector Machine, Decision Tree, Gradient Boosting and Light Gradient Boosted Machine, to determine which one performs a better prediction, to choose the algorithm

the performance metrics were used which were: Confusion matrix, accuracy, precision, sensitivity, F1 score and the area under the curve, giving us better results the Gradient boosting algorithm, having the following results: accuracy 72.33%, precision 74.10%, sensitivity 69.83%, F1 score 71.90% and area under the curve 72.36%. Which was saved in a .pkl extension file to be used in the creation of a dashboard on a web page, which allows the entry of data according to the variables of the dataset mentioned above, and predict whether a person may have a heart disease or not, and in turn taking into account the data of the variables to make recommendations to the patient.

Keywords: machine learning, cardiovascular disease, artificial intelligence, algorithms, predictive model.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

INTRODUCCIÓN

En los últimos años las enfermedades cardiacas han sido una de las primordiales causas de muertes a nivel global, dichas enfermedades son causadas cuando las arterias que lo irrigan el corazón se encuentran obstruidas, haciendo que su funcionamiento disminuya [1] y según estimaciones de la Organización Mundial de la Salud, demuestran que estas enfermedades cobran 17,9 millones de vidas cada año.[2]

Estas enfermedades pueden ser diagnosticadas con anterioridad, pero hay el problema que la población no cuenta con una educación acerca de la prevención de las mismas y la mayoría de las veces las áreas médicas no disponen de la tecnología necesaria para efectuar un diagnóstico prematuro. Gracias a los avances tecnológicos y la Inteligencia Artificial se pueden diseñar modelos y sistemas capaces de ayudar en el proceso de diagnóstico y tratamiento de los pacientes, por lo cual en este trabajo de investigación se utilizarán técnicas de aprendizaje automático, la cual es una rama de la Inteligencia Artificial que tiene como finalidad la de desarrollar técnicas que permitan a los ordenadores aprender por si solas, mediante el uso de un conjunto de datos [3], que permite la creación de algoritmos capaces de identificar patrones en los datos, haciendo que el modelo aprenda de las observaciones y lo más importante permita hacer las predicciones de una manera correcta.[4]

Existen diversos algoritmos desarrollados a partir del aprendizaje automático los mismos que son de una gran utilidad para la toma de decisiones de manera eficaz, dentro del ámbito de la salud, en el área de cardiología ha estado dando buenos resultados, por lo que se pueden crear modelos que garanticen una predicción de forma más exacta y de pronto análisis, permitiendo de esta manera el ahorro de tiempo y tomar decisiones en beneficio del paciente.[5]

El presente documento consta de la siguiente estructura:

Capítulo 1: se describe el ámbito, la justificación y se establecen los requerimientos.

Capítulo 2: se especifican los conceptos relevantes para la comprensión del trabajo como son: la fundamentación teórica, objetivos, diseño y ejecución del prototipo.

Capítulo 3: dentro de esta sección se incluyen los análisis de los datos que se obtuvieron y los resultados obtenidos.

1. CAPÍTULO I: DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. Ámbito de Aplicación: descripción del contexto y hechos de interés

Un ataque cardíaco se provoca cuando el flujo de sangre al corazón a través de las arterias se bloquea, lo cual trae consigo que en ocasiones se rompan y se formen coágulos que bloquean el flujo sanguíneo, lo que puede deteriorar o destruir parte del músculo cardíaco.

Las afecciones cardiovasculares son el principal motivo de fallecimiento tanto de hombres como de mujeres en todo el mundo. Por lo cual es importante detectar de forma inmediata un infarto ya que el tiempo de reacción puede ser crucial.

Además la importancia de realizar esta intervención es fundamental ya que la tasa de mortalidad es demasiado alta.[6]

Hoy en día la tecnología permite aplicar técnicas a grandes volúmenes de datos, un claro ejemplo es el Aprendizaje Profundo, el cual es una rama de la Inteligencia Artificial, cuyo fin es de desarrollar técnicas que permitan a las computadoras aprender, en el entorno de la salud resulta una herramienta clave para salvar vidas, ya que permite ayudar a los médicos a evaluar de un forma rápida el estado de salud de los pacientes.

En el presente caso, como trabajo de titulación, se busca diseñar un modelo para la predicción de ataques cardíacos haciendo uso de técnicas de aprendizaje automático, haciendo uso de una base de datos que contiene datos médicos de varios pacientes y comparará diferentes algoritmos de aprendizaje automático para averiguar con cuál se obtiene mejores resultados prediciendo si se producirá un infarto o no.

1.2. Establecimiento de requerimientos

Para crear el modelo de predicción de las enfermedades cardiacas, se necesitará los siguientes requerimientos:

- Recolección de información sobre el aprendizaje automático para el desarrollo del modelo.
- Un dataset constituido de datos médicos acerca de pacientes que han presentado una enfermedad cardíaca.
- Uso del entorno Jupyter Notebook para el diseño del modelo, el entrenamiento y pruebas del mismo.
- Uso de la biblioteca streamlit para la creación de una aplicación web, que permita el ingreso de los datos presentes en el dataset para realizar la predicción acorde al modelo generado.

1.3. Justificación del requerimiento a satisfacer

Según [2] dice que en los últimos años las enfermedades cardiovasculares han sido una de las principales causas de muerte a nivel mundial, por lo cual es importante poder predecir si algún paciente sufre de dichas enfermedades y de esta manera prevenir el riesgo de muerte.

En la actualidad el uso de la tecnología ha permitido dar grandes avances dentro del ámbito de la salud, una de ellas es la Inteligencia Artificial ya que las diversas ramas que la componen cuentan con características únicas que permiten el desarrollo de diversos trabajos en beneficio de la salud, como por ejemplo: detectar el riesgo de padecer ciertas enfermedades como: el cáncer, el alzheimer,Covid-19, retinopatía diabética, etc.

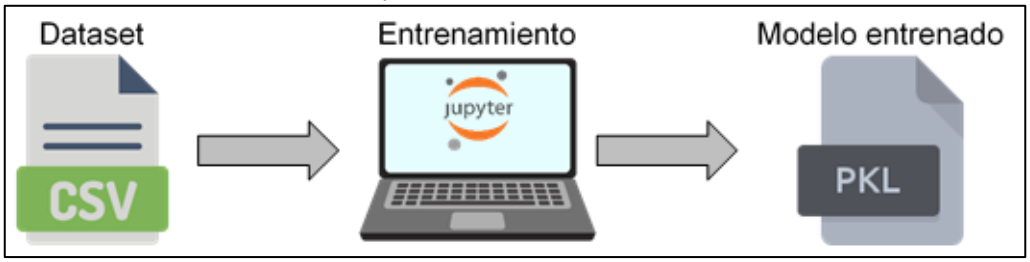
Por lo tanto, este proyecto integra el entrenamiento de un modelo utilizando un algoritmo de aprendizaje automático, que permitirá predecir si una persona presente una enfermedad cardiaca.

2. CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo tecnológico

El modelo de predicción desarrollado en el presente trabajo de titulación, hace uso de algunos modelos predictivos, se los cuales al hacer varias pruebas y viendo los resultados obtenidos se escogió el mejor y dio como resultado un archivo con extensión .pkl el cual contiene el modelo que permite realizar la predicción si una persona padece alguna enfermedad cardiaca o no.

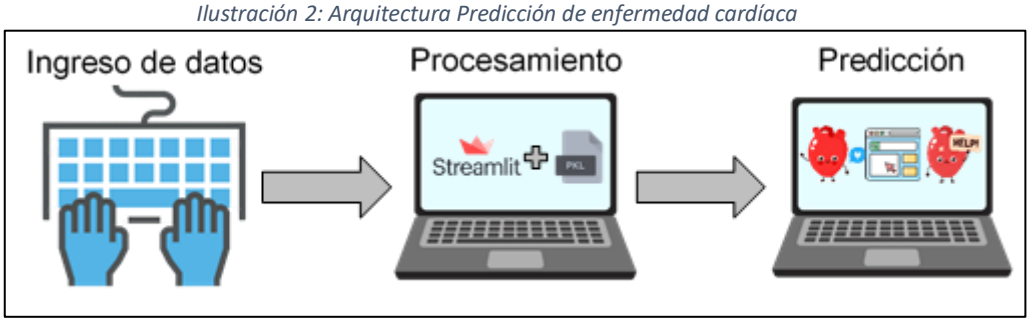
Ilustración 1: Arquitectura de entrenamiento del modelo



Fuente: *Elaboración propia*

Con el archivo .pkl, se procedió hacer uso de la librería streamlit la cual permite la creación de una página web, en la cual se importó el modelo entrenado y se diseñó para que se introdujeran los datos, acorde a las variables usadas en la creación del modelo para de esta forma obtener la predicción.

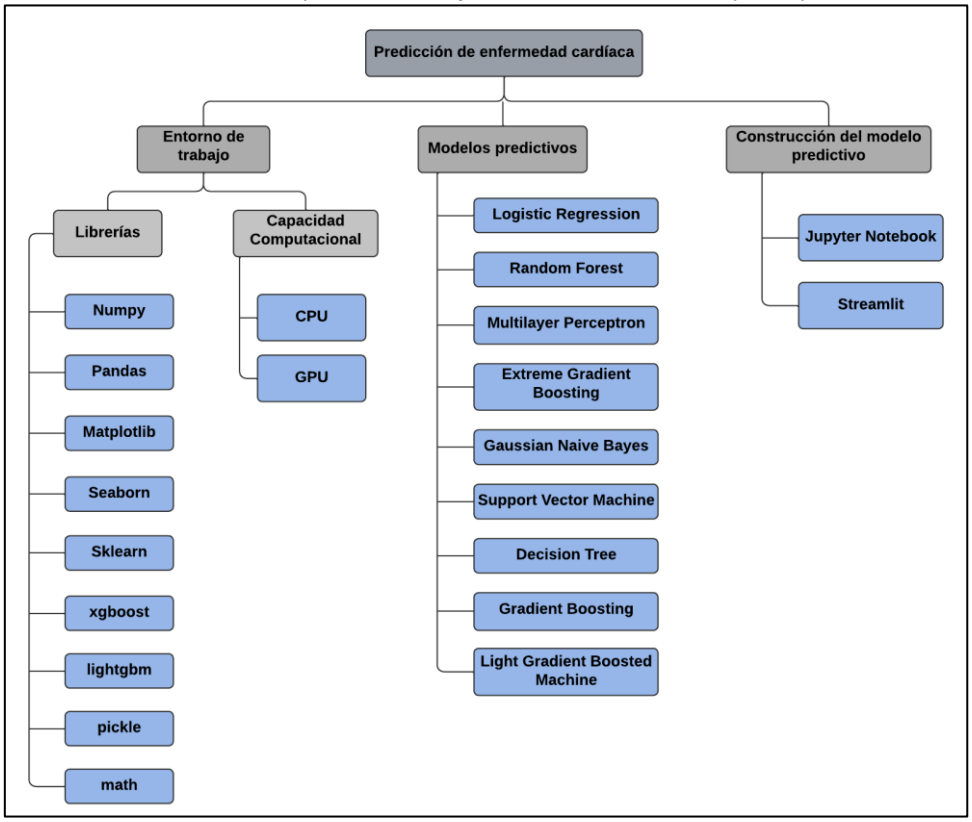
Ilustración 2: Arquitectura Predicción de enfermedad cardíaca



Fuente: *Elaboración propia*

2.2. Fundamentación teórica del prototipo

Ilustración 3:Mapa mental de la fundamentación teórica del prototipo



Fuente: *Elaboración propia*

2.2.1. Entorno de trabajo

2.2.1.1. Librerías

2.2.1.1.1. Numpy

Es un biblioteca usada para trabajar con matrices tanto unidimensionales como multidimensionales por su eficacia, además cuenta con varias características para el análisis estadístico y también para almacenar datos en matrices.[7]

2.2.1.1.2. Pandas

Es un librería utilizada para el cálculo numérico, además Pandas funciona bien en el manejo de objetos Data Frame, y permite la conversión de datos a una forma tabular, para que los datos sean leídos de una manera más fácil y estén más estructurados.[8]

2.2.1.1.3. Matplotlib

Permite la creación de gráficos estáticos, animados e interactivos, junto con las librerías de Scipy, Numpy y Pandas aportan un gran rendimiento.[9]

2.2.1.1.4. Seaborn

Es una librería que permite la creación de gráficos estadísticos, proporcionando una interfaz de alto nivel a matplotlib y se complementa de una excelente manera con las estructuras de datos de pandas.[10]

2.2.1.1.5. Sklearn

Es un paquete que incluye una colección de métodos para el aprendizaje automático, el cual está bien documentado y mantenido por la comunidad. Incluye varios algoritmos de clasificación entre los cuales se encuentran: máquinas de soporte de vectores, bosques aleatorios, gradient boosting, árbol de decisión, etc.[11]

2.2.1.1.6. Xgboost

Es una librería de aumento de gradiente la misma que se encuentra distribuida, optimizada y trazada para ser muy eficiente , flexible y portátil, permitiendo su implementación en algoritmos de aprendizaje automático relacionados al Gradient Boosting.[12]

2.2.1.1.7. Lightgbm

Es un marco mejorado para gradiente que hace uso de algoritmos de aprendizaje que se encuentran fundamentado en árboles. Está bosquejado para ser distribuido y eficaz permitiendo una mayor velocidad de entrenamiento y mayor eficiencia.[13]

2.2.1.1.8. Pickle

Es una librería que permite serializar objetos, permitiendo almacenar los mismos en archivos que pueden ser recuperados por la misma librería. [14]

2.2.1.1.9. Math

Este módulo proporciona acceso a las funciones matemáticas definidas en el estándar de C.[15]

2.2.1.2. Capacidad Computacional

Hoy en día el desarrollo de algoritmos eficaces, necesitan de algunos recursos que permitan su construcción, por lo cual es indispensable contar con características en el equipo que permitan su desarrollo, a esto se lo conoce como Capacidad Computacional. La CPU y la GPU son uno de los principales componentes que ayudan a mejorar el rendimiento de los equipos y de esta manera facilitar el proceso de creación de modelos, algoritmos, etc.

2.2.1.2.1. CPU

La Unidad Central de procesamiento o más conocido como CPU, es uno de los componentes fundamentales de un ordenador, ya que es el encargado de procesar los datos y realizar cálculos

2.2.1.2.2. GPU

La Unidad de Procesamiento de Gráficos a mejor conocido como GPU es un procesador que se encuentra formado por muchos núcleos pequeños que ofrecen un desempeño eficaz, ya que se divide las tareas de procesamiento.

Hoy en día se requiere un capacidad de cálculo superior por lo cual las CPU pueden resultar un opción no muy factible ya que utilizan un procesamiento de serie, lo cual hace que se demore al contar con grandes cantidades de datos, mientras que la GPU ofrecen una mayor velocidad de procesamiento debido a que su procesamiento es de forma paralela.[16]

2.2.1.3. Modelos predictivos

Son un conjunto de métodos que mediante los campos del aprendizaje automático, la recopilación de información y la identificación de patrones, permiten realizar predicciones, con la finalidad de ayudar en la toma de decisiones mediante estas técnicas de análisis de datos.

2.2.1.3.1. Logistic Regression

Es una herramienta de análisis estadística, que puede ser de uso interpretativo como predictivo. Siendo de gran utilidad cuando se cuenta con una variable dependiente y un conjunto de variables independientes o predictoras, las cuales pueden ser cuantitativas o categóricas.[17]

2.2.1.3.2. Random Forest

Es un algoritmo popular y muy eficiente para los problemas de clasificación y regresión, este modelo pertenece a la familia de métodos conjuntos, el principio del Random Forest es combinar varios árboles de decisión los mismos que son construidos utilizando una variedad de Bootstrap los mismos que provienen del entrenamiento y se va eligiendo aleatoriamente los nodos de cada subconjunto de variables explicitas.[18]

2.2.1.3.3. Multilayer Perceptron

Las redes neuronales artificiales son estructuras que se encuentran inspiradas en el funcionamiento del cerebro, dichas redes pueden realizar una estimación de la función de un modelo y operar funciones tanto lineales como no lineales aprendiendo de relaciones entre los datos. Una de las redes neuronales artificiales popular es la Perceptrón Multicapa la cual es una herramienta muy potente de modelado que aplica un procedimiento de entrenamiento supervisado haciendo uso de ejemplos de datos con resultados conocidos.[19]

2.2.1.3.4. Decision Tree

Es un modelo de aprendizaje supervisado, de tipo no paramétrico, permitiendo analizar conjuntamente variables nominales y cuantitativas, proporcionando predicción muy fiables con un gran conjunto de datos y es utilizado para la clasificación y regresión.[20]

2.2.1.3.5. Gaussian Naive Bayes

Es un algoritmo de clasificación el cual consiste en establecer etiquetas a las clases para así maximizar la probabilidad posterior de cada una de las muestras, bajo los supuestos de los volúmenes de datos independientes los cuales obedecen a la distribución gaussiana.[21]

2.2.1.3.6. Support Vector Machine

Es un modelo de aprendizaje automático, que aprende por medio de patrones de clasificación de los diversos datos con una precisión y reproducibilidad equilibradas. Aunque usualmente es utilizado para la regresión, dicho modelo se ha convertido en una de las herramientas más utilizadas para la clasificación.[22]

2.2.1.3.7. Gradient Boosting

Es una técnica de aprendizaje automático que es usada para el análisis de regresión y para dificultades de clasificación estadística, el cual combina modelos de predicción débiles, para de esta manera obtener un mejor modelo de forma iterativa.[23]

2.2.1.3.8. Extreme Gradient Boosting

Es una variante eficiente y escalable del modelo Gradient Boosting, este modelo es uno de los más efectivos dentro del aprendizaje automático, debido a las características de facilidad de uso, la facilidad para la paralelización y su excelente precisión para la predicción. Además ofrece una amplia perspectiva de los datos, que

permite manejarlos cuando estos sean diversos y complejos, es decir, cuando la distribución de las clases se encuentra bastante desequilibradas.[24]

2.2.1.3.9. Light Gradient Boosted Machine

Es un modelo mejorado del Gradient Boosting el cual se encuentra basado en los árboles de decisión y en la idea de aprendices débiles, haciendo que se lo aplique en diversos campos debido a la alta precisión de predicción, la velocidad de procesamiento y la eficaz capacidad de reducir los problemas de sobreajustes.[25]

2.2.1.4. Construcción del modelo

2.2.1.4.1. Jupyter Notebook

Es una aplicación cliente-servidor de código abierto, que permite la creación de documentos computacionales los mismos que pueden compartirse.[26] Una de las plataformas que lo contienen es Anaconda Navigator la cual es una aplicación de escritorio que permite la administración de una manera fácil de aplicaciones, paquetes y entornos integrados sin el uso de la línea de comandos.[27]

Jupyter Notebook cuenta con un entorno que satisface algunas necesidades concretas que permite ajustarse a trabajos relacionados con la ciencia de datos y la simulación numérica. Al contar con una única interfaz facilita que los usuarios puedan escribir, documentar, ejecutar código, visualizar datos, realizar cálculos y observar los resultados. Presenta la ventaja de que el código se estructura en celdas independientes, lo que permite experimentar bloques de código de forma individual. El kernel por defecto es IPython lo que permite trabajar con el lenguaje de programación Python, otra ventaja que presenta es la de que posee varios kernels lo cual no limita solo al uso de Python, lo que permite que exista flexibilidad al momento de crear código y de realizar diversos análisis.[28]

2.2.1.4.2. Streamlit

Es un marco gratuito y de código abierto que permite crear y compartir de manera rápida aplicaciones web de aprendizaje automático y ciencia de datos. Es una biblioteca basada en Python la cual esta específicamente diseñada para científicos de datos o los ingenieros de aprendizaje automático, ya que es una herramienta fácil de aprender y usar, siempre y cuando se muestren los datos y recopilaciones de los parámetros necesarios para el modelado, dicha herramienta permite crear aplicaciones con una excelente apariencia con solo pocas líneas de código.

Uno de los aspectos relevantes de esta biblioteca es que no se necesita conocer conceptos básicos de desarrollo web, lo que permite la creación de las mismas de una manera rápida con una interfaz eficaz e intuitiva. Presenta varias ventajas como son: [29]

- No se requiere de experiencia o conocimiento de front-end.
- Es compatible con la mayoría de las bibliotecas de Python.
- Se necesita de menos código para crear las aplicaciones web.
- El almacenamiento en caché simplifica y acelera los procesos de cálculo.

2.3. Objetivos del prototipo

2.3.1. Objetivo General

Desarrollar un modelo para la predicción de enfermedades cardíacas aplicando algoritmos de Machine Learning.

2.3.2. Objetivos Específicos

- Investigar y revisar los algoritmos para la predicción con relación al aprendizaje automático.
- Investigar los factores que influyen en las enfermedades cardíacas para de esta manera buscar un dataset que ayude al entrenamiento del modelo.
- Obtener métricas de evaluación de las pruebas, para conocimiento de cuál es el mejor modelo predictivo.
- Diseñar una interfaz web que permita al usuario el ingreso de datos acorde al dataset y que este realice la predicción.

2.4. Diseño del prototipo

El diseño del prototipo consta de dos partes: la primera que consiste en hacer pruebas con la data obtenida del sitio Kaggle [30] y comparar los diferentes modelos de aprendizaje automático escogidos, mediante algunas métricas como son: exactitud, precisión, sensibilidad, puntuación F1 y área bajo la curva, las mismas que ayudan a conocer cuál es el mejor modelo y permitir guardarlo en un archivo extensión .pkl para el desarrollo de la segunda parte, la misma que consta en la elaboración de una interfaz web para el ingreso de datos acorde a las variables del dataset y de esta manera obtener la predicción si una persona sufre o no una enfermedad cardíaca.

2.4.1. Requisitos

2.4.1.1. Hardware

Tabla 1: Características del equipo

Características del equipo	
Marca	HP
Procesador	Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz
Memoria RAM	8 GB
Sistema Operativo	Windows 10 Pro

Fuente: *Elaboración propia*

2.4.1.2. Software

Tabla 2: Software usados para la elaboración del modelo

Software(librerías, aplicación, etc.)	Versión
Anaconda Navigator	conda 4.13.0
Jupyter Notebook	6.4.11
Python	3.9.7
Numpy	1.21.5
Pandas	1.4.2
Matplotlib	3.5.1
Seaborn	0.11.2
Warnings	3.6
Sklearn	1.0.2
Xgboost	1.6.1
Lightgbm	3.3.2
Mlxtend	0.20.0
Streamlit	1.10.0
Pickle	4.0
Math	3.10.5

Fuente: *Elaboración propia*

2.4.2. Dataset

El dataset usado para el modelo predictivo se llama cardio_train el mismo que es de extensión .csv, dicho dataset fue encontrado en la página Kaggle [30], el que cuenta con tres tipos de características de entrada que son:

- Objetiva: información fáctica
- Examen: resultados del examen médico
- Subjetiva: información dada por el paciente

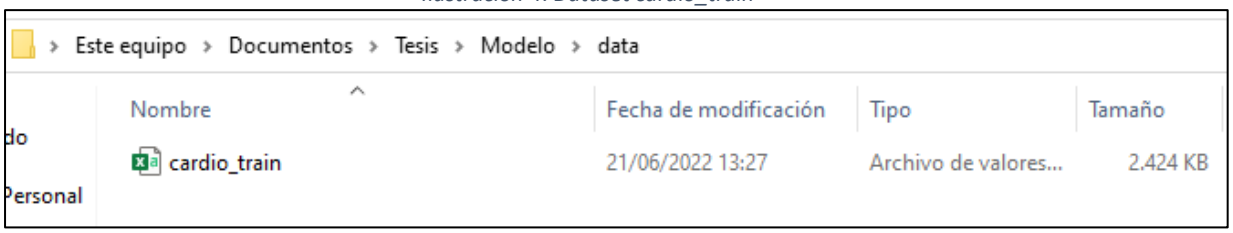
Las características presentes en el dataset son las siguientes:

Tabla 3: Características del dataset cardio_train

Características	Tipo de característica de entrada	Variable
Age	Objetiva	age
Height	Objetiva	height
Weight	Objetiva	weight
Gender	Objetiva	gender
Systolic blood pressure	Examen	ap_hi
Diastolic blood pressure	Examen	ap_lo
Cholesterol	Examen	colesterol
Glucose	Examen	gluc
Smoking	Subjetiva	smoke
Alcohol intake	Subjetiva	alco
Physical activity	Subjetiva	active
Presence or absence of cardiovascular disease	Variable objetivo	cardio

Fuente: *Elaboración propia*

Ilustración 4: Dataset cardio_train



Fuente: *Elaboración propia*

El dataset cardio_train cuenta con 65536 registros de pacientes, cabe recalcar que todos los valores del conjunto de datos se recopilaron en el momento del examen médico.

Además se hicieron cambios en algunas de las características del dataset las cuales fueron Age, Gender, Cholesterol y Glucose, los cambios realizados fueron cambiar el contenido de su registro de la siguientes manera:

Tabla 4: Cambios realizados en las características del dataset

Características	Cambios realizados
Age	Pasar de días a años
Gender	0: mujer y 1: hombre
Cholesterol	0: normal, 1: alto y 2: muy alto
Glucose	0: normal, 1: alto y 2: muy alto

Fuente: *Elaboración propia*

2.4.3. Diseño del modelo predictivo

Para el diseño del modelo se hizo uso de Jupyter Notebook, en el cual se siguieron una serie de fases propias de Machine Learning [31], tomando a consideración la metodología CRISP-DM de sus siglas en inglés (Cross Industry Standard Process for Data Mining) la misma que junta tareas necesarias para proyectos relacionados a la minería de datos, desde la fase de entendimiento de la problemática hasta la implementación de sistemas computarizados analíticos. [32] Las etapas a seguir son las siguientes:

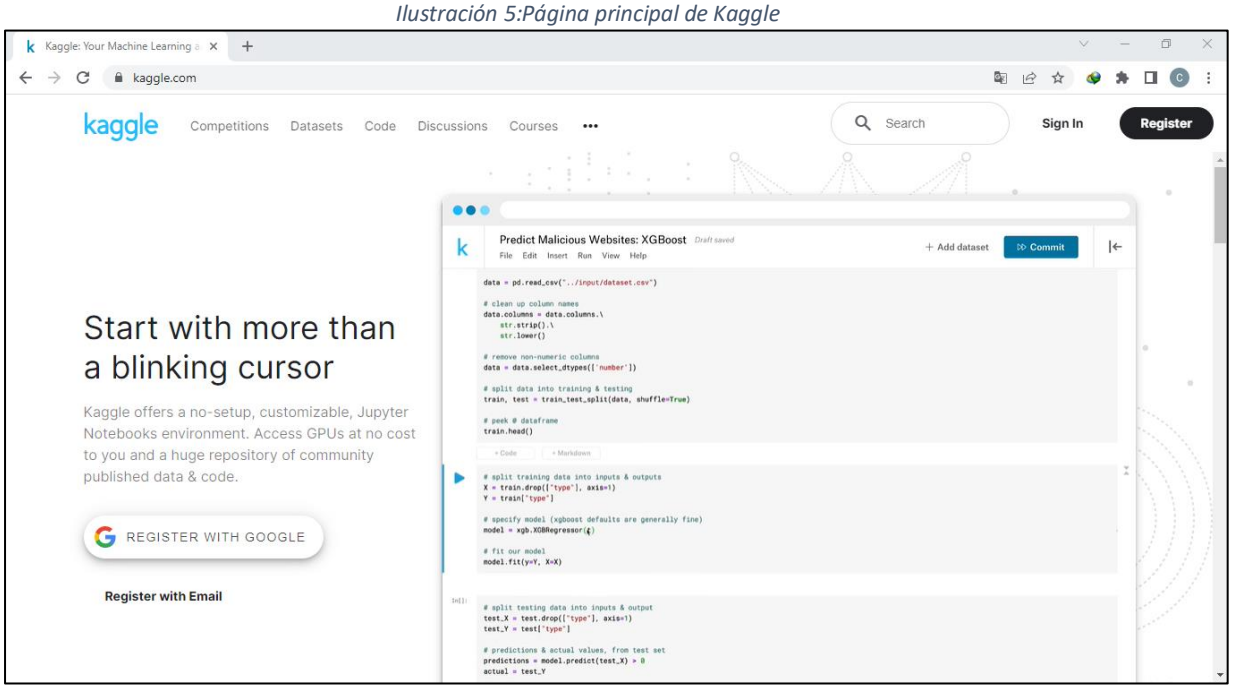
Etapa 1: Definir el objetivo

Es de suma importancia tener claro el problema que se quiere resolver, para de esta manera evitar conflictos a lo largo del proceso de creación del modelo predictivo.

Etapa 2: Recolección de la data

En esta etapa buscamos la data acorde al objetivo definido, teniendo en cuenta las fortalezas y limitaciones que esta pueda tener. Se puede encontrar la data en plataformas con repositorios de datos la más conocida es Kaggle la cual contiene la comunidad Data Science más voluminosa del mundo, con más de 536 mil miembros activos en 194 países y recibe más de 150 mil publicaciones por mes. Además de ofrecer una interfaz de Jupyter Notebooks personalizable, cuenta con GPUs a los cuales se puede acceder de manera gratuita, al igual que al gran conjunto de datos y códigos publicados en la comunidad.[33]

Ilustración 5:Página principal de Kaggle



Etapa 3: Preparar la data

Una vez que contamos con la data se realiza el preprocesamiento de la misma, lo que se conoce como la limpieza de los datos. El objetivo de esta etapa es manipular la data para que produzcan mejores resultados. Ejemplos de lo que se realiza en esta etapa es la de buscar datos nulos y datos duplicados para de esta manera eliminarlos e ir limpiando nuestro dataset, búsqueda de valores atípicos los cuales son observaciones cuyos valores son muy variantes a las observaciones del mismo grupo de datos, realizar cambios a las variables o campos de las mismas y escalarlos para así poder compararlos.

Ilustración 6: Cambio de nombre a las variables del dataset

```
#Renombramos las columnas del dataset, para que se haga más fácil el trabajo
data.rename(columns={'age':'edad','gender':'genero','height':'altura','weight':'peso','BMI':'IMC','cholesterol':'colesterol','glu
data.rename(columns=lambda x:x.strip().replace(' ','_'),inplace=True)
```

Fuente: *Elaboración propia*

Ilustración 7: Verificación de si hay datos nulos

```
#Verificamos si hay datos Nulos, es decir, registros en blanco
data.isnull()
```

	edad	genero	altura	peso	IMC	ap_hi	ap_lo	colesterol	glucosa	fumar	beber	ejercitar	cardio
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
65530	False	False	False	False	False	False	False	False	False	False	False	False	False
65531	False	False	False	False	False	False	False	False	False	False	False	False	False
65532	False	False	False	False	False	False	False	False	False	False	False	False	False
65533	False	False	False	False	False	False	False	False	False	False	False	False	False
65534	False	False	False	False	False	False	False	False	False	False	False	False	False

65535 rows × 13 columns

```
#Se realiza la suma para conocer cuantos datos nulos hay en el dataset
data.isnull().sum()
```

```
edad          0
genero        0
altura        0
peso          0
IMC           0
ap_hi         0
ap_lo         0
colesterol    0
glucosa       0
fumar         0
beber         0
ejercitar     0
cardio        0
dtype: int64
```

Fuente: *Elaboración propia*

Ilustración 8: Verificación y eliminación de datos duplicados

```
#Verificamos si existen datos duplicados
data.duplicated()
```

```
0        False
1        False
2        False
3        False
4        False
         ...
65530    False
65531    False
65532    False
65533     True
65534    False
Length: 65535, dtype: bool
```

```
#Se realiza la suma para conocer el total de registros duplicados
data.duplicated().sum()
```

```
3415
```

```
#Se borra los datos duplicados
data.drop_duplicates(subset=None, keep='first', inplace=True)
```

```
#Verificamos que se borraron los datos duplicados
data.shape
```

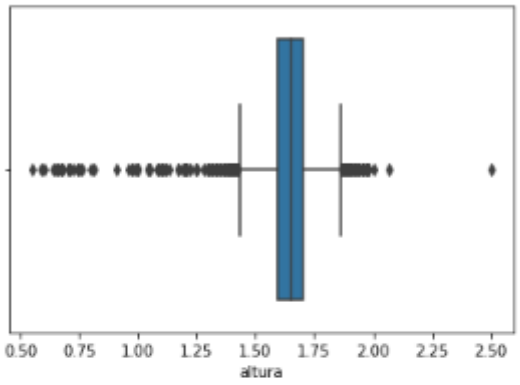
```
(62120, 13)
```

Fuente: *Elaboración propia*

Ilustración 9: Búsqueda de valores atípicos

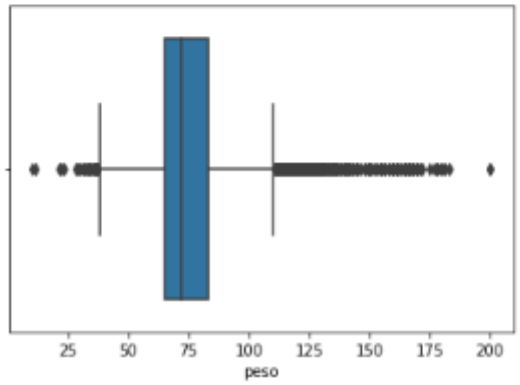
```
sns.boxplot(data['altura'])
```

```
<AxesSubplot:xlabel='altura'>
```




```
sns.boxplot(data['peso'])
```

```
<AxesSubplot:xlabel='peso'>
```



Fuente: *Elaboración propia*

Ilustración 10: Eliminación de los valores atípicos

```
#Borrar outliers
outliers = ['edad','altura','peso','IMC','ap_hi','ap_lo']
def outlier_removal(data,column):
    q1 = data[column].quantile(0.25)
    q3 = data[column].quantile(0.75)
    iqr = q3 - q1
    point_low = q1 - 1.5 * iqr
    point_high = q3 + 1.5 * iqr
    clean_HF = data.loc[(data[column] >  point_low) & (data[column] <  point_high)]
    return clean_HF

#Se hace la limpieza del conjunto de datos eliminando los valores atípicos
data_limp = outlier_removal(outlier_removal(outlier_removal(outlier_removal(outlier_removal(outlier_removal(data,'edad'),'altura
print(data.shape)
print(data_limp.shape)
```

```
(62120, 13)
(53804, 13)
```

Fuente: *Elaboración propia*

Etapa 4: Preparar datos de entrenamiento

Hecha la limpieza de los datos se procede a separar la data para el entrenamiento y prueba, comúnmente el 80% de la data es utilizada para el entrenamiento, y de esta manera poder entrenar los modelos para realizar la elección.

Ilustración 11. Separación de los datos de entrenamiento y de prueba

```
F1 = ['edad', 'genero', 'altura', 'peso', 'IMC', 'ap_hi', 'ap_lo','colesterol', 'glucosa', 'fumar', 'beber', 'ejercitar']
predictors = data_limp[F1]
target = data_limp["cardio"]

X_train, X_test, y_train, y_test = train_test_split(predictors, target, test_size = 0.2, random_state = 42)
```

```
sc=StandardScaler()
X_train = sc.fit_transform(X_train[F1])
X_test = sc.transform(X_test[F1])
```

```
print('X_train: ',X_train.shape)
print('X_test: ',X_test.shape)
print('y_train: ',y_train.shape)
print('y_test: ',y_test.shape)
```

```
X_train:  (43043, 12)
X_test:  (10761, 12)
y_train:  (43043,)
y_test:  (10761,)
```

Fuente: *Elaboración propia*

Etapa 5: Elección del algoritmo

Finalizada la etapa de limpieza de la data, debemos elegir el algoritmo más adecuado para resolver el problema, en este caso se hizo uso de algoritmos de aprendizaje supervisado los cuales se llaman así ya que para su entrenamiento necesitan un conjunto de datos de entrenamiento lo cual permitirá al algoritmo realizar las predicciones y compararlas con las etiquetes y de esta manera el aprendizaje será progresivo.[34]

Tabla 5:Algoritmos de aprendizaje supervisado a elegir

Nº	Algoritmos a elegir
1	**Logistic Regression**
2	**Random Forest**
3	**Multilayer Perceptron**
4	**Decision Tree**
5	**Gaussian Naive Bayes**
6	**Support Vector Machine**
7	**Gradient Boosting**
8	**Extreme Gradient Boosting**
9	**Light Gradient Boosted Machine**

Fuente: *Elaboración propia*

Ilustración 12. Modelo seleccionado Gradient Boosting

```
modelGB = GradientBoostingClassifier()
modelGB.fit(X_train,y_train)
y_predGB = modelGB.predict(X_test)

accGB= round(accuracy_score(y_test, y_predGB) * 100, 2)
print('Exactitud del Modelo GB: ',accGB)
lista.append(accGB)

GBprec=round(precision_score(y_test,y_predGB)*100,2)
print("Precisión del modelo GB: ",GBprec)
listaPre.append(GBprec)

GBreca=round(recall_score(y_test,y_predGB)*100,2)
print("Sensibilidad del modelo GB: ",GBreca)
listaRec.append(GBreca)

GBf1_s=round(f1_score(y_test,y_predGB)*100,2)
print('Puntuación F1 del modelo GB: ',GBf1_s)
listaF1.append(GBf1_s)

aucGB = round(roc_auc_score(y_test, y_predGB)*100,2)
print("AUC del Modelo GB: ",aucGB)
listaAUC.append(aucGB)
```

```
Exactitud del Modelo GB:  72.33
Precisión del modelo GB:  74.1
Sensibilidad del modelo GB:  69.83
Puntuación F1 del modelo GB:  71.9
AUC del Modelo GB:  72.36
```

Fuente: *Elaboración propia*

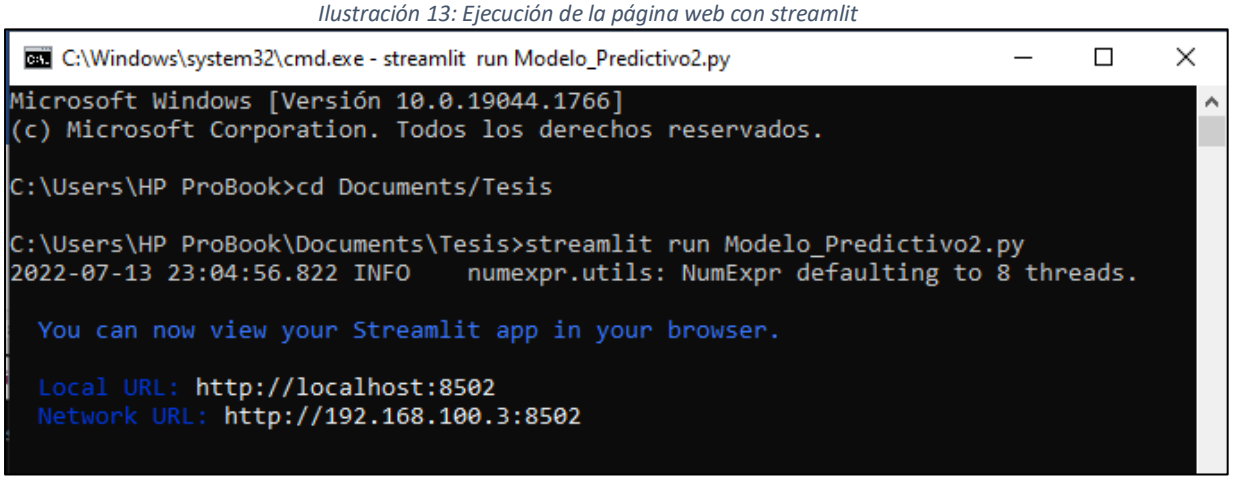
Etapa 6: Predicción

Se hace el ingreso de nueva información al modelo, para de esta manera obtener la predicción y verificar si el modelo predice de manera correcta o no.

2.5. Ejecución y/o ensamblaje del prototipo

Para poner en funcionamiento la página web creada con Streamlit, se debe ejecutar mediante consola la instrucción *streamlit run nombre_del_modelo.py,* pero antes de eso se debe ubicar en la carpeta donde se haya guardado.

Ilustración 13: Ejecución de la página web con streamlit



Fuente: *Elaboración propia*

Una vez colocada la instrucción se abrirá en nuestro navegador predeterminado la página, cabe recalcar que la misma se encuentra de forma local.

Ilustración 14: Página del modelo predictivo

Fuente: *Elaboración propia*

Se hace el ingreso de los datos acorde a las variables presentes en el dataset.

Ilustración 15:Ingrese de datos del paciente

Fuente: *Elaboración propia*

Una vez ingresados se da clic en el botón predecir y nos mostrará un mensaje si el paciente sufre o no una enfermedad cardíaca.

Ilustración 16:Predicción del modelo

Fuente: *Elaboración propia*

3. CAPÍTULO III. EVALUACIÓN DEL PROTOTIPO

3.1. Plan de evaluación

Para evaluar los resultados del modelo predictivo, se utilizaron métricas de rendimiento las cuales son muy importantes para elegir el mejor algoritmo, las métricas tomadas en cuentas fueron las siguientes: matriz de confusión, exactitud, precisión, sensibilidad, puntuación F1 y área bajo la curva.

3.1.1. Métricas de rendimiento

3.1.1.1. Matriz de confusión

Una matriz de confusión permite ilustrar el rendimiento del clasificador en función de los valores de verdaderos positivos (VP), falsos negativos (FN), falsos positivos (FP) y verdaderos negativos (VN). [35]

Se detalla a continuación a lo que se refieren los valores de la matriz de confusión:[36]

- Verdaderos positivos (VP): son la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- Verdadero negativos (VN): son la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- Falsos negativos (FN): son la cantidad de positivos que fueron clasificados incorrectamente como negativos.

- Falsos positivos(FP): son la cantidad de negativos que fueron clasificados incorrectamente como positivos.

Ilustración 17: Matriz de confusión

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Fuente: [36]

3.1.1.2. Exactitud (Accuracy)

Es una de las métricas más utilizadas para la evaluación del modelo, la que describe la cantidad de predicciones que fueron correctas y la fórmula para calcularla es la siguiente: [37]

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN}$$

3.1.1.3. Precisión (Precision)

Esta medida permite conocer la cantidad de predicciones positivas que fueron correctas y la fórmula para obtenerla es la siguiente: [37]

$$Precision = \frac{VP}{VP + FP}$$

3.1.1.4. Sensibilidad (Recall)

Indica cuantos de los casos positivos predijo correctamente el modelo, sobre todo los casos positivos en los datos y la fórmula para calcularla es la siguiente:[37]

$$Recall = \frac{VP}{VP + FN}$$

3.1.1.5. Puntuación F1 (F1-Score)

Es una medida que combina la precisión y sensibilidad, generalmente se describe como la media de las dos, su fórmula es la siguiente:[37]

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

3.1.1.6. Área bajo la curva (AUC)

Esta métrica permite evaluar el rendimiento del modelo, en la que el criterio de evaluación es: cuanto mayor sea el AUC, mejor será el modelo.[38]

3.2. Resultados de la evaluación

3.2.1. Resultados de la prueba de entrenamiento

Prueba 1

Tabla 6: Parámetros de la Prueba 1

Parámetros	Valor
Porcentaje de datos de entrenamiento	90% - 48423 registros
Porcentaje de datos de prueba	10% - 5381 registros

Fuente: *Elaboración propia*

Resultados obtenidos de las métricas de exactitud, área bajo la curva, precisión, sensibilidad y puntuación F1 de los diferentes modelos

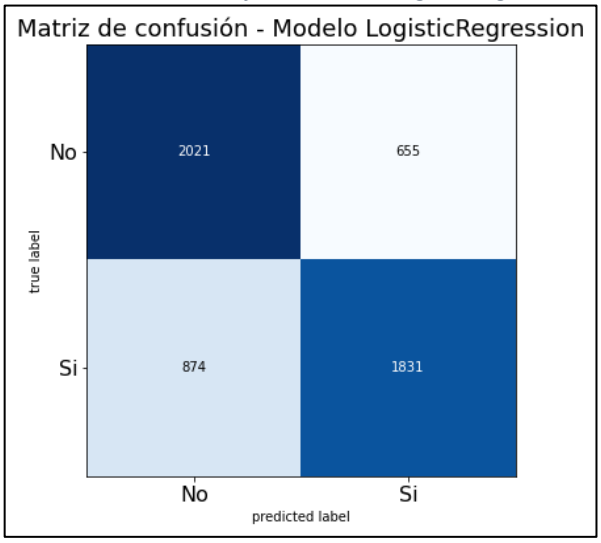
Tabla 7: Resultados de las métricas de la Prueba 1

Modelo	Exactitud - Accuracy	Área bajo la curva - AUC	Precisión - Precision	Sensibilidad - Recall	Puntuación F1 - F1-Score
Logistic Regresion	71.59	71.61	73.65	67.69	70.55
Random Forest	71.98	72	74.82	66.69	70.52
Multi Layer Perceptron	70.49	70.50	71.61	68.43	69.98
Decision Tree	59.91	59.92	60.38	58.93	59.64
Gaussian Naive Bayes	70.93	70.97	74.84	63.55	68.73
Support Vector Machine	71.51	71.55	75.13	64.77	69.57
Gradient Boosting	72.72	72.73	74.11	70.28	72.14
Extreme Gradient Boosting	72.12	72.14	73.73	69.21	71.40
Light Gradient Boosted Machine	72.59	72.61	74.4	69.32	71.77

Fuente: *Elaboración propia*

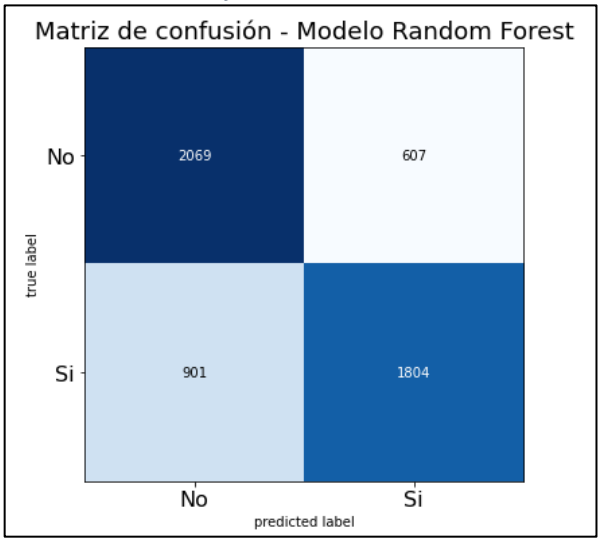
Matriz de confusión de los diferentes modelos

Ilustración 18: Matriz de confusión-Modelo Logistic Regression – Prueba 1



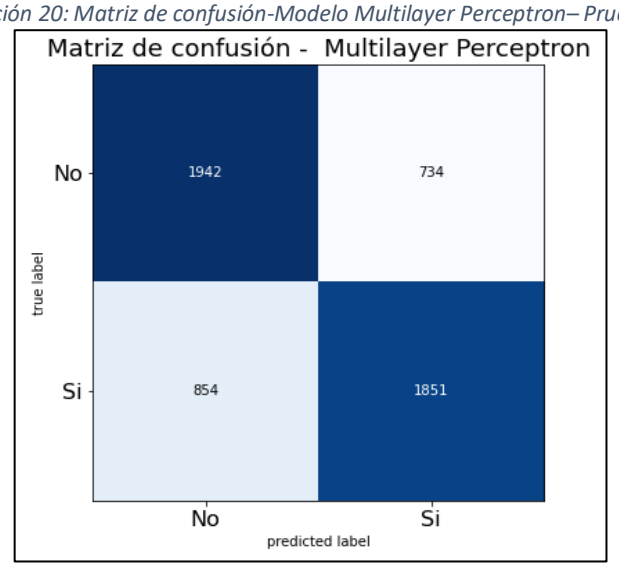
Fuente: *Elaboración propia*

Ilustración 19: Matriz de confusión-Modelo Random Forest – Prueba 1



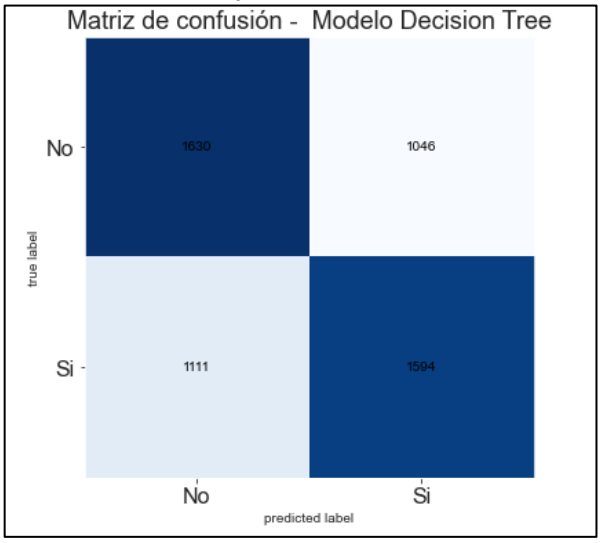
Fuente: *Elaboración propia*

Ilustración 20: Matriz de confusión-Modelo Multilayer Perceptron– Prueba 1



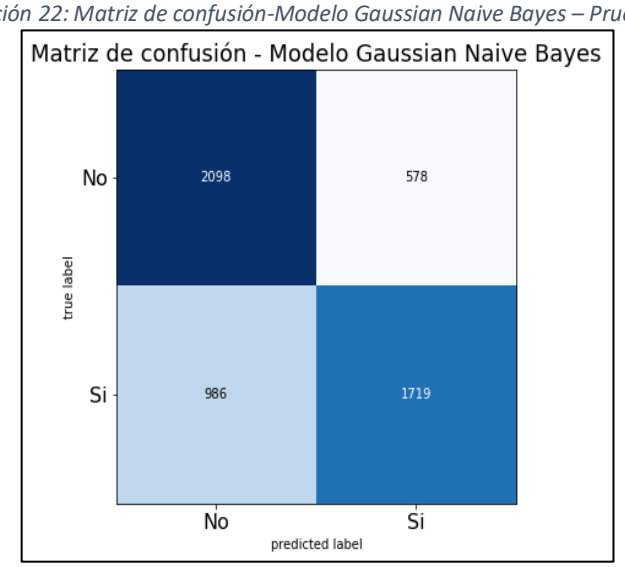
Fuente: *Elaboración propia*

Ilustración 21: Matriz de confusión-Modelo Decision Tree– Prueba 1



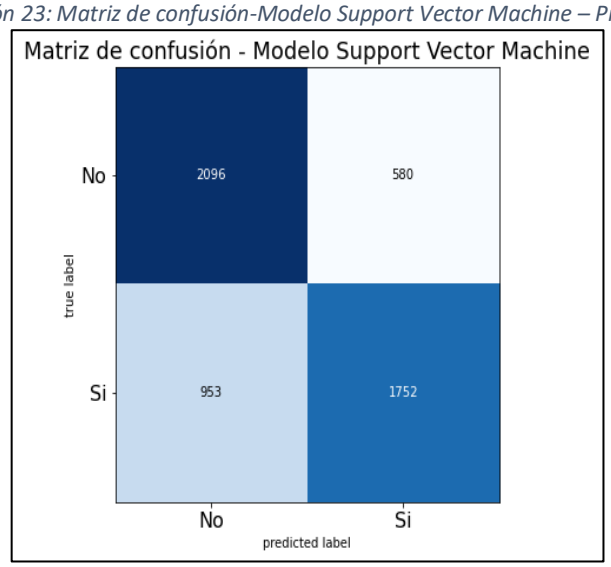
Fuente: *Elaboración propia*

Ilustración 22: Matriz de confusión-Modelo Gaussian Naive Bayes – Prueba 1



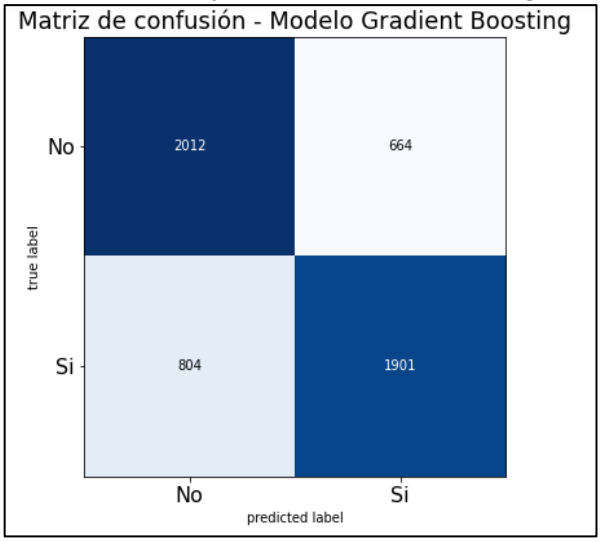
Fuente: *Elaboración propia*

Ilustración 23: Matriz de confusión-Modelo Support Vector Machine – Prueba 1



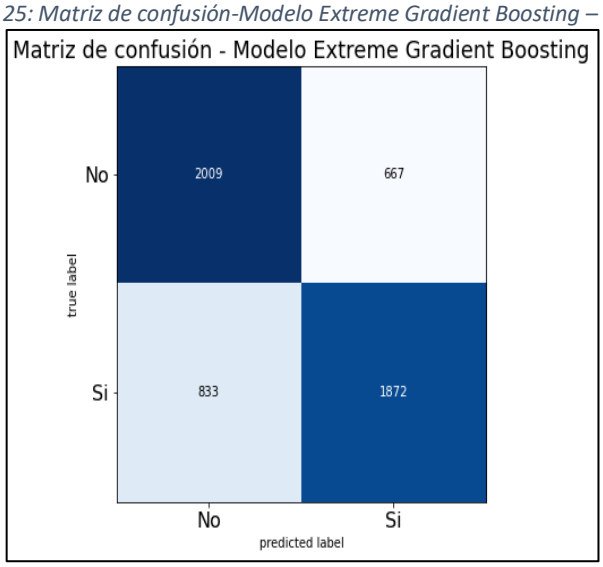
Fuente: *Elaboración propia*

Ilustración 24: Matriz de confusión-Modelo Gradient Boosting– Prueba 1



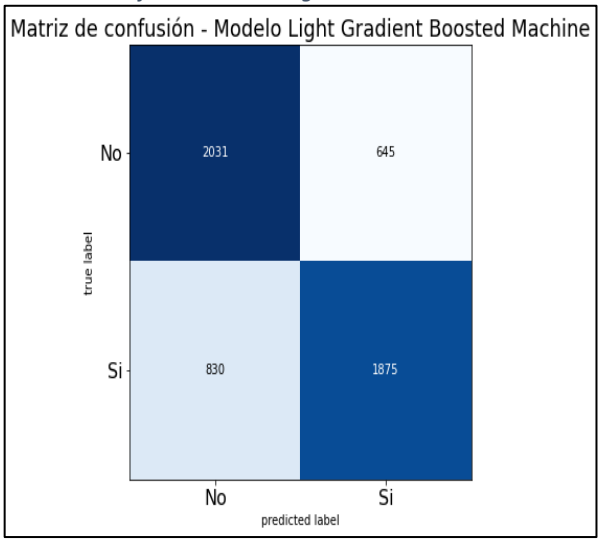
Fuente: *Elaboración propia*

Ilustración 25: Matriz de confusión-Modelo Extreme Gradient Boosting – Prueba 1



Fuente: *Elaboración propia*

Ilustración 26: Matriz de confusión-Modelo Light Gradient Boosted Machine – Prueba 1



Fuente: Elaboración propia

Prueba 2

Tabla 8: Parámetros de la Prueba 2

Parámetros	Valor
Porcentaje de datos de entrenamiento	80% - 43043
Porcentaje de datos de prueba	20% - 10761

Fuente: Elaboración propia

Resultados obtenidos de las métricas de exactitud, área bajo la curva, precisión, sensibilidad y puntuación F1 de los diferentes modelos

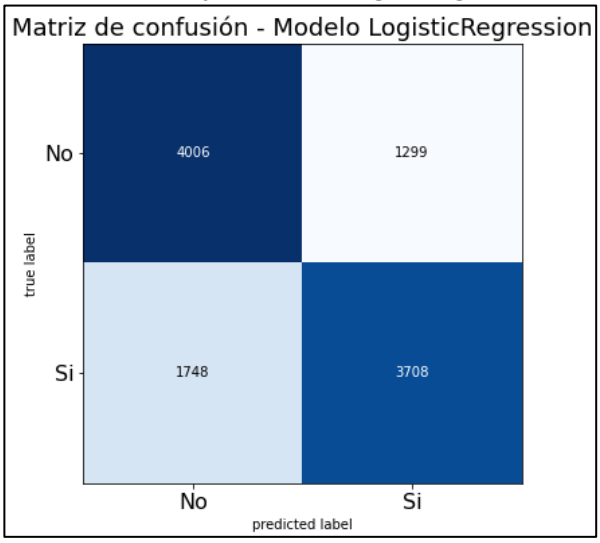
Tabla 9: Resultados de las métricas de la Prueba 2

Modelo	Exactitud - Accuracy	Área bajo la curva - AUC	Precisión - Precision	Sensibilidad - Recall	Puntuación F1 - F1-Score
Logistic Regresion	71.68	71.74	74.06	67.96	70.88
Random Forest	71.72	71.80	74.91	66.50	70.45
Multi Layer Perceptron	70.11	70.09	69.97	71.92	70.93
Decision Tree	61.49	61.48	62.03	61.99	62.01
Gaussian Naive Bayes	70.59	70.69	74.69	63.51	68.65
Support Vector Machine	71.47	71.57	75.52	64.70	69.69
Gradient Boosting	72.33	72.36	74.10	69.83	71.90
Extreme Gradient Boosting	71.75	71.79	73.75	68.75	71.16
Light Gradient Boosted Machine	72.08	72.13	74.34	68.64	71.37

Fuente: Elaboración propia

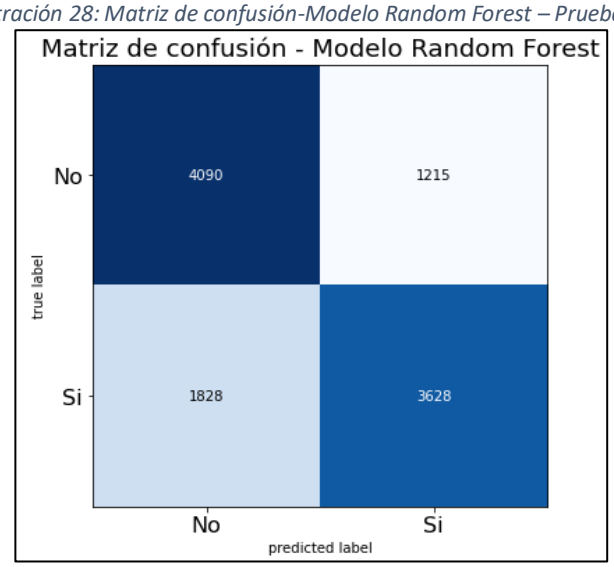
Matriz de confusión de los diferentes modelos

Ilustración 27:Matriz de confusión-Modelo Logistic Regression – Prueba 2



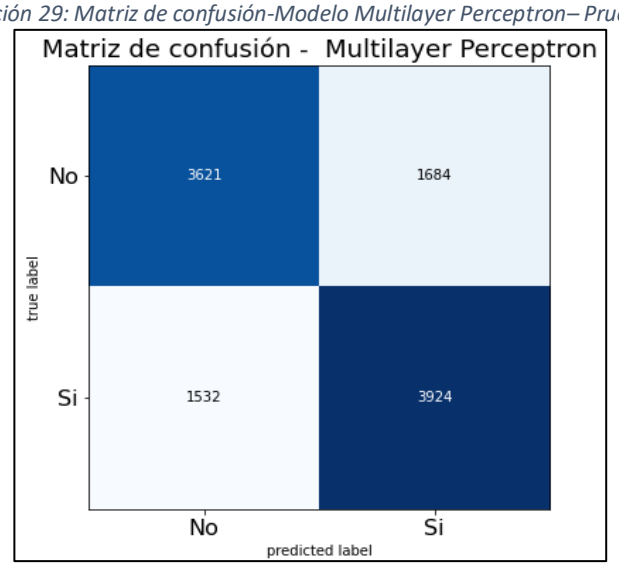
Fuente: *Elaboración propia*

Ilustración 28: Matriz de confusión-Modelo Random Forest – Prueba 2



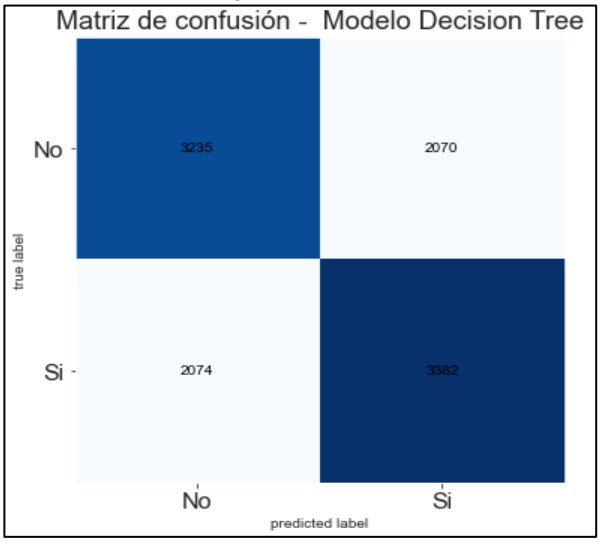
Fuente: *Elaboración propia*

Ilustración 29: Matriz de confusión-Modelo Multilayer Perceptron– Prueba 2



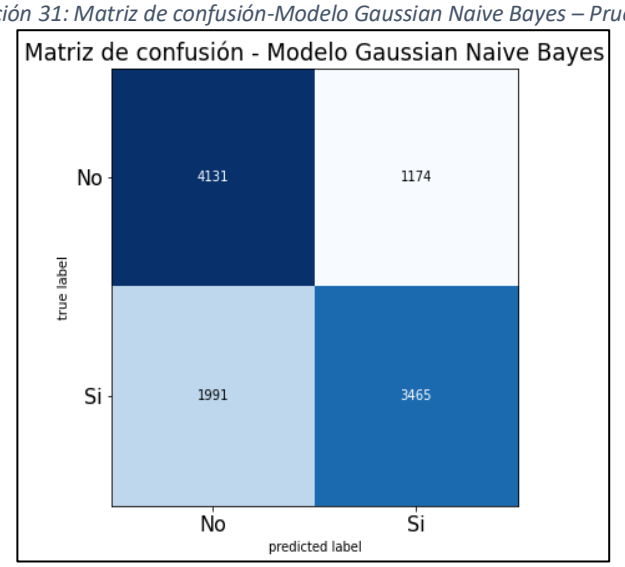
Fuente: *Elaboración propia*

Ilustración 30: Matriz de confusión-Modelo Decision Tree– Prueba 2



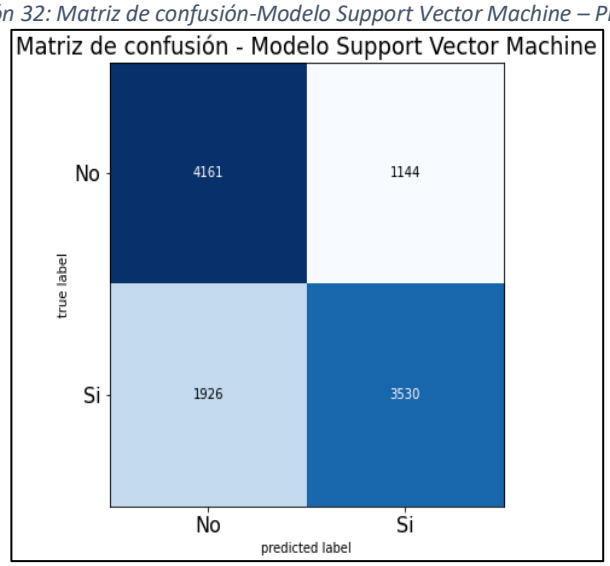
Fuente: *Elaboración propia*

Ilustración 31: Matriz de confusión-Modelo Gaussian Naive Bayes – Prueba 2



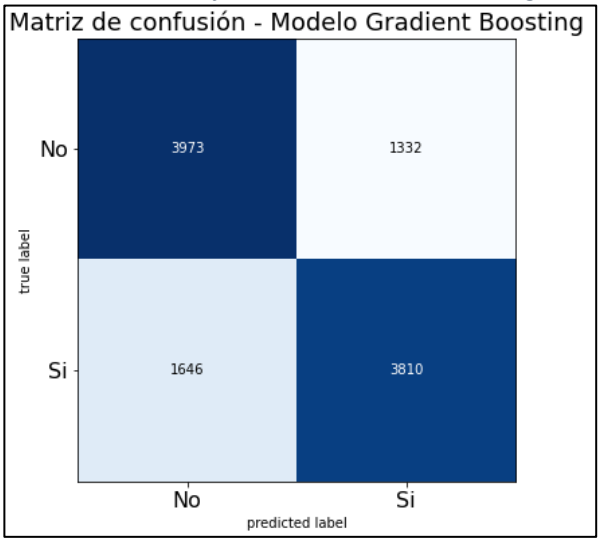
Fuente: *Elaboración propia*

Ilustración 32: Matriz de confusión-Modelo Support Vector Machine – Prueba 2



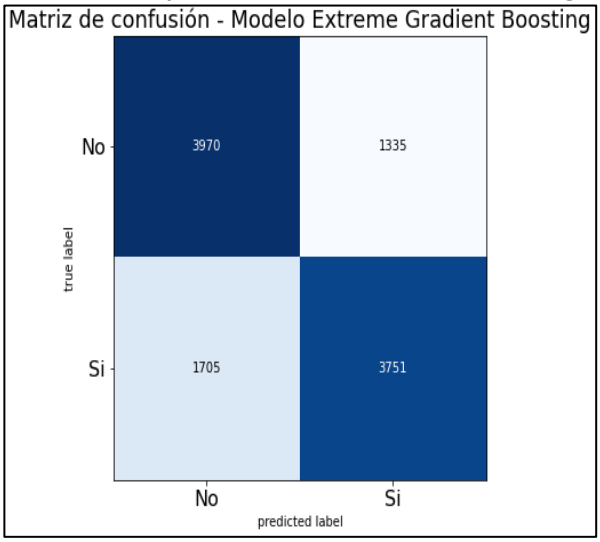
Fuente: *Elaboración propia*

Ilustración 33: Matriz de confusión-Modelo Gradient Boosting– Prueba 2



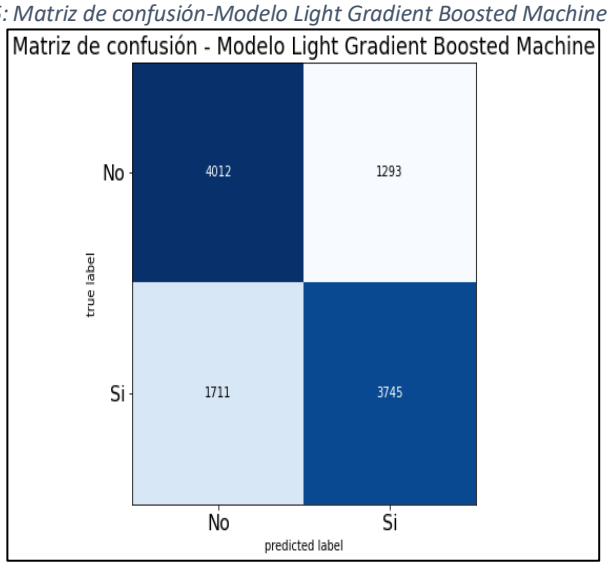
Fuente: *Elaboración propia*

Ilustración 34: Matriz de confusión-Modelo Extreme Gradient Boosting – Prueba 2



Fuente: *Elaboración propia*

Ilustración 35: Matriz de confusión-Modelo Light Gradient Boosted Machine – Prueba 2



Fuente: *Elaboración propia*

Prueba 3

Tabla 10: Parámetros de la Prueba 3

Parámetros	Valor
Porcentaje de datos de entrenamiento	70% - 37662
Porcentaje de datos de prueba	30% - 16142

Fuente: *Elaboración propia*

Resultados obtenidos de las métricas de exactitud, área bajo la curva, precisión, sensibilidad y puntuación F1 de los diferentes modelos

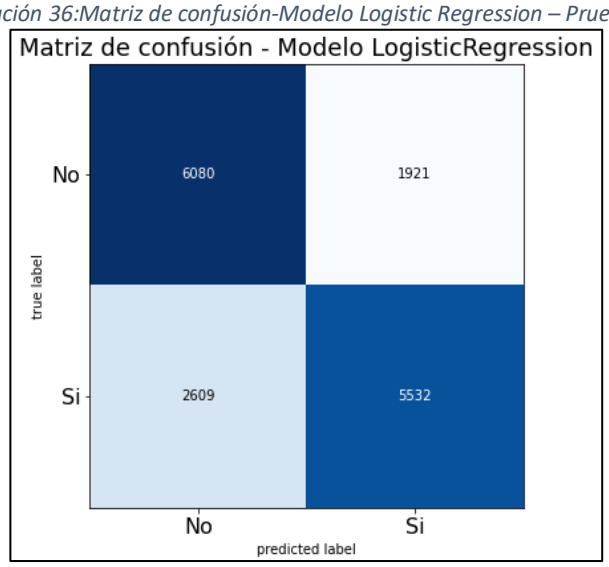
Tabla 11: Resultados de las métricas de la Prueba 3

Modelo	Exactitud - Accuracy	Área bajo la curva - AUC	Precisión - Precision	Sensibilidad - Recall	Puntuación F1 - F1-Score
Logistic Regresion	71.94	71.97	74.23	67.95	70.95
Random Forest	72	72.06	75.93	65.13	70.11
Multi Layer Perceptron	70.93	70.98	74.04	65.23	69.35
Decision Tree	61.14	61.14	61.42	61.74	61.58
Gaussian Naive Bayes	70.55	70.61	74.48	63.30	68.43
Support Vector Machine	71.63	71.69	75.65	64.50	69.63
Gradient Boosting	72.51	72.53	74.28	69.59	71.85
Extreme Gradient Boosting	71.83	71.86	73.87	68.30	70.97
Light Gradient Boosted Machine	72.38	72.41	74.59	68.59	71.47

Fuente: *Elaboración propia*

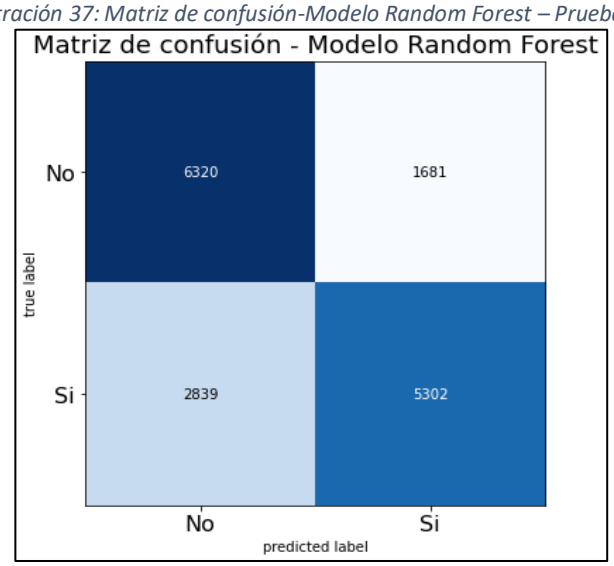
Matriz de confusión de los diferentes modelos

Ilustración 36:Matriz de confusión-Modelo Logistic Regression – Prueba 3



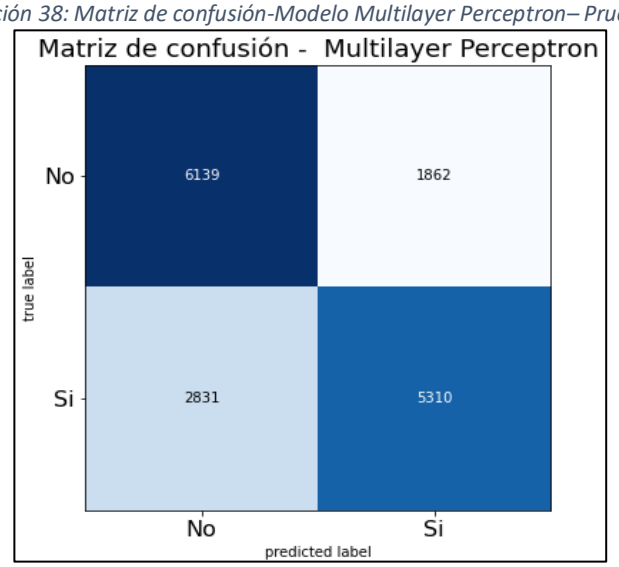
Fuente: *Elaboración propia*

Ilustración 37: Matriz de confusión-Modelo Random Forest – Prueba 3



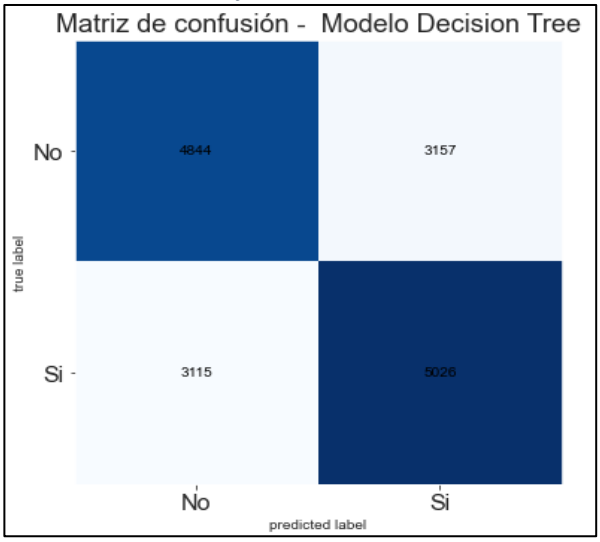
Fuente: *Elaboración propia*

Ilustración 38: Matriz de confusión-Modelo Multilayer Perceptron– Prueba 3



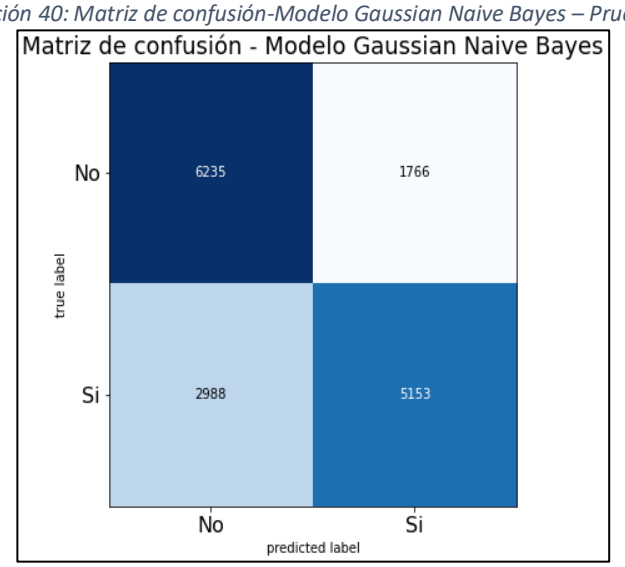
Fuente: *Elaboración propia*

Ilustración 39: Matriz de confusión-Modelo Decision Tree– Prueba 3



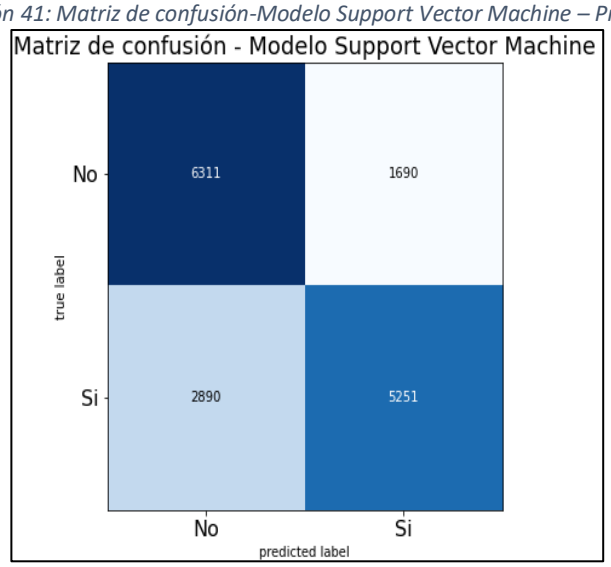
Fuente: *Elaboración propia*

Ilustración 40: Matriz de confusión-Modelo Gaussian Naive Bayes – Prueba 3



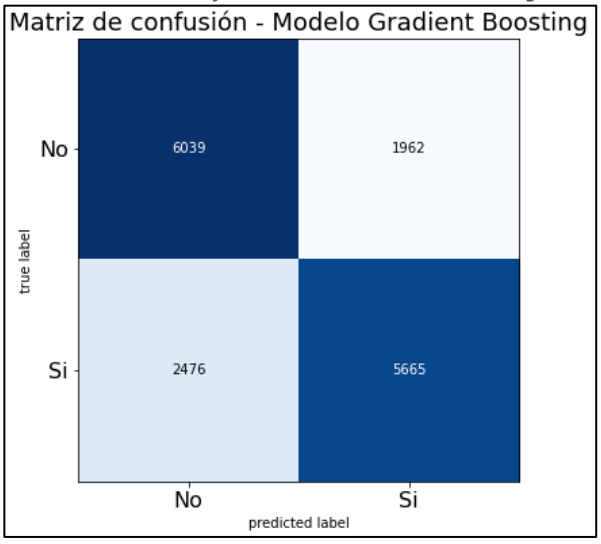
Fuente: *Elaboración propia*

Ilustración 41: Matriz de confusión-Modelo Support Vector Machine – Prueba 3



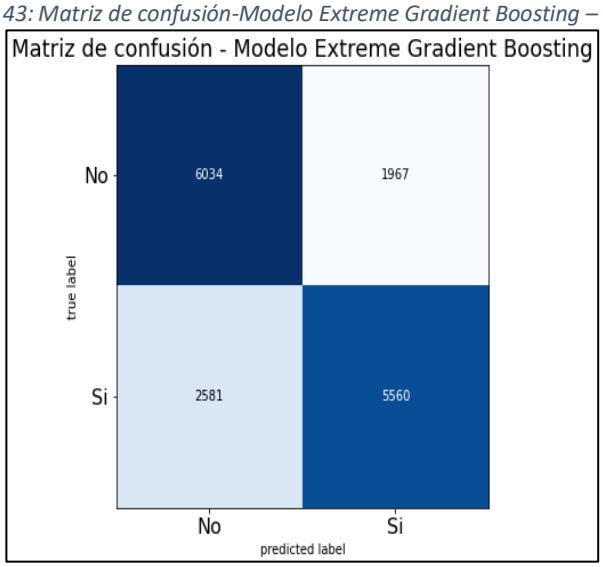
Fuente: *Elaboración propia*

Ilustración 42: Matriz de confusión-Modelo Gradient Boosting– Prueba 3



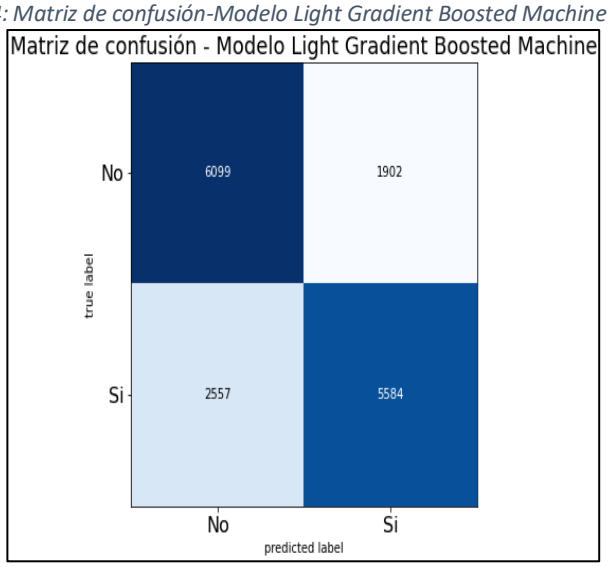
Fuente: *Elaboración propia*

Ilustración 43: Matriz de confusión-Modelo Extreme Gradient Boosting – Prueba 3



Fuente: *Elaboración propia*

Ilustración 44: Matriz de confusión-Modelo Light Gradient Boosted Machine – Prueba 3



Fuente: *Elaboración propia*

Teniendo en cuenta los resultados de las métricas de los diferentes modelos en las diferentes pruebas realizadas, el que estuvo mejor puntuado fue el Modelo Gradient Boosting, así mismo visualizando la matriz de confusión de dicho modelo se percata que este obtuvo menor cantidad de Falsos Negativos, a comparación del resto de modelos, dándonos las certeza que es el mejor modelo para realizar la predicción del conjunto de datos con el que se trabajó.

Escogido el modelo se lo guardo para realizar las pruebas de predicción en la página desarrollada con Streamlit, en la cual se escogió datos del dataset y se hizo uso de los tres archivos de pruebas que fueron guardados.

Índice de masa corporal: IMC

Presión arterial sistólica: ap_hi

Presión arterial diastólica: ap_lo

Tabla 12: Datos para realizar pruebas en la página

	edad	género	altura	peso	IMC	ap_hi	ap_lo	colest	glucosa	fumar	beber	ejercitar	cardio
1	50	1	1.68	62.0	21.97	110	80	0	0	0	0	1	0
2	55	0	1.56	85.0	34.93	140	90	2	0	0	0	1	1
3	52	0	1.65	64.0	23.51	130	70	2	0	0	0	0	1
4	48	0	1.56	56	23.01	100	60	0	0	0	0	0	0

Fuente: *Elaboración propia*

Resultados obtenidas del archivo guardado de la prueba 1

Tabla 13: Resultados de la prueba 1

Conjunto de datos	Predicción del modelo	Predicción establecida
1	No presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca
2	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
3	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
4	Si presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca

Fuente: *Elaboración propia*

Resultados obtenidas del archivo guardado de la prueba 2

Tabla 14: Resultados de la prueba 2

Conjunto de datos	Predicción del modelo	Predicción establecida
1	No presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca
2	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
3	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
4	No presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca

Fuente: *Elaboración propia*

Resultados obtenidas del archivo guardado de la prueba 3

Tabla 15: Resultados de la prueba 3

Conjunto de datos	Predicción del modelo	Predicción establecida
1	Si presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca
2	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
3	Si presenta una enfermedad cardíaca	Si presenta una enfermedad cardíaca
4	Si presenta una enfermedad cardíaca	No presenta una enfermedad cardíaca

Fuente: *Elaboración propia*

Finalizada las pruebas en la página con los datos del dataset, se puede decir que el modelo que utiliza un 80% de datos para el entrenamiento y 20% para prueba obtiene mejores resultados al momento de realizar la predicción, ya que fue el único que obtuvo los cuatros registros de manera correcta.

3.3. Conclusiones

Como resultado del modelo de predicción de enfermedades cardíacas se concluye que:

- Mediante la revisión de artículos y revistas de carácter científico se obtuvo mayor conocimiento sobre el aprendizaje automático como los algoritmos que existen para realizar predicciones y las métricas a considerar para evaluar y escoger el mejor modelo.
- Aplicando las técnicas investigadas se logró realizar un modelo de predicción que permite mediante el ingreso de datos conocer si una persona sufre o no una enfermedad cardíaca.
- No todos los algoritmos de machine learning son factibles para la predicción, ya que algunos cuentan con mejores parámetros que ayudan a obtener mejores resultados a comparación de otros.

3.4. Recomendaciones

En base a lo realizado en este proyecto se recomienda:

- Realizar una investigación exhaustiva para de esta manera tener mayor conocimientos en los temas que se llegasen a topar a lo largo del proyecto.
- Hacer la limpieza de los datos que se vayan a usar para de esta manera obtener mejores resultados y evitar inconvenientes que puedan surgir.
- Que al momento de realizar las pruebas correspondientes se usen los parámetros que puedan tener los modelos de machine learning, ya que esto ayudara a variar un poco los resultados y obtener mejores efectos en los modelos.

Bibliografía

[1] J. Azmi, M. Arif, M. T. Nafis, M. A. Alam, S. Tanweer, y G. Wang, «A systematic review on machine learning approaches for cardiovascular disease prediction using medical big data», *Med. Eng. Phys.*, vol. 105, p. 103825, jul. 2022, doi: 10.1016/j.medengphy.2022.103825.

[2] «Enfermedades cardiovasculares». https://www.who.int/es/health-topics/cardiovascular-diseases#tab=tab_1 (accedido 13 de septiembre de 2022).

[3] M. Á. Morales Hernández *et al.*, «Algoritmos de aprendizaje automático para la predicción del logro académico», *RIDE Rev. Iberoam. Para Investig. El Desarro. Educ.*, vol. 12, n.º 24, jun. 2022, doi: 10.23913/ride.v12i24.1180.

[4] J. F. Ávila-Tomás, M. A. Mayer-Pujadas, y V. J. Quesada-Varela, «La inteligencia artificial y sus aplicaciones en medicina II: importancia actual y aplicaciones prácticas», *Aten. Primaria*, vol. 53, n.º 1, pp. 81-88, ene. 2021, doi: 10.1016/j.aprim.2020.04.014.

[5] P. I. Dorado-Díaz, J. Sampedro-Gómez, V. Vicente-Palacios, y P. L. Sánchez, «Aplicaciones de la inteligencia artificial en cardiología: el futuro ya está aquí», *Rev. Esp. Cardiol.*, vol. 72, n.º 12, pp. 1065-1075, dic. 2019, doi: 10.1016/j.recesp.2019.05.016.

[6] «El tiempo, factor clave de supervivencia al sufrir infarto agudo de miocardio - Sociedad Española de Cardiología». https://secardiologia.es/comunicacion/notas-de-prensa/notas-de-prensa-sec/2031-tiempo-factor-clave-de-supervivencia-sufrir-infarto-agudo-de-miocardio (accedido 13 de septiembre de 2022).

[7] S. I. L. Naranjo, N. A. E. Brito, V. A. V. Núñez, y E. M. R. Ordóñez, «Analysis of the use of the Python programming language for statistical calculations.», *Espirales Rev. Multidiscip. Invesitgación Científica*, vol. 6, n.º 2, pp. 1-13, 2022.

[8] F. Nelli, «The pandas Library—An Introduction», en *Python Data Analytics: With Pandas, NumPy, and Matplotlib*, Berkeley, CA: Apress, 2018, pp. 87-139. doi: 10.1007/978-1-4842-3913-1_4.

[9] E. Bisong, «Matplotlib and Seaborn», en *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Berkeley, CA: Apress, 2019, pp. 151-165. doi: 10.1007/978-1-4842-4470-8_12.

[10] M. L. Waskom, «seaborn: statistical data visualization», *J. Open Source Softw.*, vol. 6, n.º 60, p. 3021, abr. 2021, doi: 10.21105/joss.03021.

[11] J. Hao y T. K. Ho, «Machine Learning Made Easy: A Review of Scikit-learn Package in Python Programming Language», *J. Educ. Behav. Stat.*, vol. 44, n.º 3, pp. 348-361, jun. 2019, doi: 10.3102/1076998619832248.

[12] W. Liu, Z. Chen, y Y. Hu, «XGBoost algorithm-based prediction of safety assessment for pipelines», *Int. J. Press. Vessels Pip.*, vol. 197, p. 104655, jun. 2022, doi: 10.1016/j.ijpvp.2022.104655.

[13] D. Wang, L. Li, y D. Zhao, «Corporate finance risk prediction based on LightGBM», *Inf. Sci.*, vol. 602, pp. 259-268, jul. 2022, doi: 10.1016/j.ins.2022.04.058.

[14] L. M. Aksman *et al.*, «pySuStaIn: A Python implementation of the Subtype and Stage Inference algorithm», *SoftwareX*, vol. 16, p. 100811, dic. 2021, doi: 10.1016/j.softx.2021.100811.

[15] «math — Funciones matemáticas — documentación de Python - 3.10.7». https://docs.python.org/es/3/library/math.html (accedido 13 de septiembre de 2022).

[16] W. Barreiros *et al.*, «Efficient microscopy image analysis on CPU-GPU systems with cost-aware irregular data partitioning», *J. Parallel Distrib. Comput.*, vol. 164, pp. 40-54, jun. 2022, doi: 10.1016/j.jpdc.2022.02.004.

[17] K. R. Vergaray, «Modelo predictivo para la detección temprana de estudiantes con alto riesgo de deserción académica», *Innov. Softw.*, vol. 2, n.º 2, pp. 6-13, 2021.

[18] J. J. Espinosa-Zúñiga, «Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito», *Ing. Investig. Tecnol.*, vol. XXI, n.º 3, 2020, Accedido: 13 de septiembre de 2022. [En línea]. Disponible en: https://www.redalyc.org/articulo.oa?id=40471792003

[19] H. Taud y J. F. Mas, «Multilayer Perceptron (MLP)», en *Geomatic Approaches for Modeling Land Change Scenarios*, M. T. Camacho Olmedo, M. Paegelow, J.-F. Mas, y F. Escobar, Eds. Cham: Springer International Publishing, 2018, pp. 451-455. doi: 10.1007/978-3-319-60801-3_27.

[20] M. T. Huyut y H. Üstündağ, «Prediction of diagnosis and prognosis of COVID-19 disease by blood gas parameters using decision trees machine learning model: a retrospective observational study», *Med. Gas Res.*, vol. 12, n.º 2, pp. 60-66, jun. 2022, doi: 10.4103/2045-9912.326002.

[21] A. N. N. Azmi, S. Khairunniza-Bejo, M. Jahari, F. M. Muharram, y I. Yule, «Identification of a suitable machine learning model for detection of asymptomatic Ganoderma boninense infection in oil palm seedlings using hyperspectral data», *Appl. Sci. Switz.*, vol. 11, n.º 24, 2021, doi: 10.3390/app112411798.

[22] D. A. Pisner y D. M. Schnyer, «Chapter 6 - Support vector machine», en *Machine Learning*, A. Mechelli y S. Vieira, Eds. Academic Press, 2020, pp. 101-121. doi: 10.1016/B978-0-12-815739-8.00006-7.

[23] C. Bentéjac, A. Csörgő, y G. Martínez-Muñoz, «A comparative analysis of gradient boosting algorithms», *Artif. Intell. Rev.*, vol. 54, n.º 3, pp. 1937-1967, mar. 2021, doi: 10.1007/s10462-020-09896-5.

[24] B. Ibrahim, F. Majeed, A. Ewusi, y I. Ahenkorah, «Residual geochemical gold grade prediction using extreme gradient boosting», *Environ. Chall.*, vol. 6, p. 100421, ene. 2022, doi: 10.1016/j.envc.2021.100421.

[25] J. Fan, X. Ma, L. Wu, F. Zhang, X. Yu, y W. Zeng, «Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external

meteorological data», *Agric. Water Manag.*, vol. 225, p. 105758, nov. 2019, doi: 10.1016/j.agwat.2019.105758.

[26] «Project Jupyter». https://jupyter.org (accedido 13 de septiembre de 2022).

[27] «Anaconda | Anaconda Distribution», *Anaconda*. https://www.anaconda.com/products/distribution (accedido 13 de septiembre de 2022).

[28] «Jupyter Notebook: documentos web para análisis de datos, código en vivo y mucho más», *IONOS Digital Guide*. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/ (accedido 13 de septiembre de 2022).

[29] «Python Tutorial: Streamlit». https://www.datacamp.com/tutorial/streamlit (accedido 13 de septiembre de 2022).

[30] Svetlana Ulianova, «Cardiovascular Disease dataset». https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset (accedido 13 de septiembre de 2022).

[31] C. Janiesch, P. Zschech, y K. Heinrich, «Machine learning and deep learning», *Electron. Mark.*, vol. 31, n.º 3, pp. 685-695, sep. 2021, doi: 10.1007/s12525-021-00475-2.

[32] V. Plotnikova, M. Dumas, y F. P. Milani, «Applying the CRISP-DM data mining process in the financial services industry: Elicitation of adaptation requirements», *Data Knowl. Eng.*, vol. 139, p. 102013, may 2022, doi: 10.1016/j.datak.2022.102013.

[33] T. rédac, «Kaggle: todo lo que hay que saber sobre esta plataforma», *Formation Data Science | DataScientest.com*, 14 de diciembre de 2021. https://datascientest.com/es/kaggle-todo-lo-que-hay-que-saber-sobre-esta-plataforma (accedido 13 de septiembre de 2022).

[34] W. H. Lee, M. Antoniades, H. G. Schnack, R. S. Kahn, y S. Frangou, «Brain age prediction in schizophrenia: Does the choice of machine learning algorithm matter?», *Psychiatry Res. Neuroimaging*, vol. 310, p. 111270, abr. 2021, doi: 10.1016/j.pscychresns.2021.111270.

[35] R. R. Sanni y H. S. Guruprasad, «Analysis of performance metrics of heart failured patients using Python and machine learning algorithms», *Glob. Transit. Proc.*, vol. 2, n.º 2, pp. 233-237, nov. 2021, doi: 10.1016/j.gltp.2021.08.028.

[36] Y. Boo y Y. Choi, «Comparison of mortality prediction models for road traffic accidents: an ensemble technique for imbalanced data», *BMC Public Health*, vol. 22, n.º 1, 2022, doi: 10.1186/s12889-022-13719-3.

[37] S. H. Kok, A. Azween, y N. Jhanjhi, «Evaluation metric for crypto-ransomware detection using machine learning», *J. Inf. Secur. Appl.*, vol. 55, p. 102646, dic. 2020, doi: 10.1016/j.jisa.2020.102646.

[38] B. S. dos Santos, M. T. A. Steiner, y R. H. P. Lima, «Proposal of a method to classify female smokers based on data mining techniques», *Comput. Ind. Eng.*, vol. 170, p. 108363, ago. 2022, doi: 10.1016/j.cie.2022.108363.