
Detección temprana de enfermedades del corazón mediante el aprendizaje automático

Heart Disease early detection using the Machine Learning

Dick Díaz Delgado

<https://orcid.org/0000-0002-9374-9349>

dick.diaz@unmsm.edu.pe

Universidad Nacional Mayor de San Marcos. Lima, Perú

RECIBIDO: 23/07/2022 - ACEPTADO: 18/08/2022 - PUBLICADO: 20/09/2022

RESUMEN

El presente trabajo de investigación tiene por objetivo comparar la precisión de siete algoritmos de clasificación para un conjunto de datos sobre enfermedades del corazón, permitiendo la detección temprana en paciente que tengan sospecha o no de la presencia de este mal que a simple no se puede diagnosticar. El alcance del proyecto se limita a analizar a doscientos noventa y nueve pacientes que realizaron sus pruebas para la creación de este conjunto de datos, con la posibilidad de continuar realizar constantes pruebas para mejorar la calidad de la información en la prueba. Para el proceso de clasificación se pondrá a prueba a 25% para el entrenamiento versus el 75% de los registros como muestra, los cuales pasarán por cada uno de los modelos utilizando la Matriz de Confusión y la curva Auc Roc. Se pudo apreciar que la precisión en el modelo de Regresión Lineal se obtuvo un 87.3%, mientras que en la curva Auc Roc se obtuvo una confiabilidad del 91% para el modelo MultiLayer Perceptron Prediction. La información de las pruebas realizadas permite obtener una precisión considerable en cuanto al modelo de Regresión Lineal indicando que trabaja de manera correcta con la misma, sin embargo, con una cantidad mayor de registros se podría mejorar aún más los resultados obtenidos actualmente.

Palabras clave: Aprendizaje automático; enfermedad del corazón; clasificación.

ABSTRACT

This research work have objective to compare the accuracy of seven classification algorithms for a set of data on heart diseases, allowing early detection in the patient who will have suspicion or not of the presence of this disease that simply cannot be detected. The scope of the project is limited to analyzing two hundred and ninety-nine patients who underwent their tests for the creation of this data set, with the possibility of continuing to carry out constant tests to improve the quality of the information in the test. For the classification process, 25% will be tested for training versus 75% of the records as a sample, which will go through each of the models using the Confusion Matrix and the Auc Roc curve. It was observed that the precision in the Linear Regression model was 87.3%, while in the Auc Roc curve a reliability of 91% was obtained for the MultiLayer Perceptron Prediction model. The information from the tests carried out allows obtaining considerable precision in terms of the Linear Regression model, indicating that it works correctly with it, however, with a greater number of records, the results currently obtained could be further improved.

Keywords: Machine learning; heart disease; classification.

I. INTRODUCCIÓN

El corazón es uno de los órganos más importantes del cuerpo humano, pues, es el encargado de bombear la sangre a cada espacio del organismo; tiene el tamaño promedio de un puño humano. Sin embargo, como todo órgano también suele presentar algunas enfermedades las cuales son conocidas como enfermedades cardiovasculares. “Las enfermedades cardiovasculares (ECV) son un grupo de trastornos del corazón y los vasos sanguíneos que incluyen la enfermedad coronaria (ataques al corazón), la cerebrovascular (ictus), la hipertensión (tensión alta), la insuficiencia cardíaca y la enfermedad cardíaca reumática, entre otras” (López y Macaya, 2009). Estos ataques suelen ser fenómenos agudos que son provocados en su mayoría por obstrucciones que impiden el paso de la sangre de manera frecuente del corazón al cerebro. Esto sucede de manera frecuente por la obstrucción en las paredes de los vasos sanguíneos debido a la adhesión de grasa, los posibles orígenes pueden suceder debido a la presencia de una combinación de factores de riesgo, como lo son el tabaquismo, dietas altas en grasas, obesidad, inactividad física, consumo nocivo de alcohol, hipertensión arterial, diabetes e hiperlipidemia. Los datos en grandes cantidades permiten ser utilizados para obtener información oculta y conocimientos útiles, a través del aprendizaje automático.

Bodero y López (2019) explican que el uso de la computación en la nube para poder aplicar Machine Learning, se ha convertido en una tendencia tecnológica debido a que en la actualidad son partícipes de un cambio importante para los seres humanos, ya sea en la vida diaria, como dentro de los procesos de instituciones u organizaciones a nivel global. Esto sin duda ya forma parte de un desarrollo innovador que permite utilizar un paradigma computacional que se encuentra orientado a facilitar y mejorar la realización de actividades complejas.

Este artículo tiene por objetivo analizar el rendimiento de siete algoritmos de clasificación utilizando el servicio de Google Colaboratory, para realizar el entrenamiento de un conjunto de datos, con la finalidad de presentar la mejor alternativa y ser ejecutado en la aplicación de machine learning de predicciones para pacientes con enfermedades al corazón de manera temprana.

II. MATERIAL Y MÉTODOS

Dataset

El conjunto de datos para este artículo proviene del sitio web de conjunto de datos de prueba estándar

de código abierto UCI. El conjunto de datos se divide en 76 atributos, pero todos los experimentos publicados se refieren al uso de un subconjunto de 14 de ellos. El campo objetivo se refiere a la presencia de la enfermedad cardíaca en el paciente. Se tendrá en cuenta la columna objetivo o columna número 14 para detectar la presencia de la enfermedad del corazón en el paciente. Para indicar la no presencia de la enfermedad se asignará el valor 0, y en caso de presencia el valor 1. El conjunto de datos para este experimento son de Cleveland.

Los atributos que tomaron en cuenta dentro del conjunto de datos son: Edad del paciente, sexo definido entre hombre y mujer, tipo de dolor torácico nivel de angina, presión arterial en reposo entre el intervalo de 90 a 300, colesterol sérico entre el intervalo de 110 a 400, azúcar en sangre en ayunas sí se encontró o no, resultados electrocardiográficos en reposo sí muestra o no anomalías, frecuencia cardíaca máxima alcanzada entre el intervalo de 110 a 300, angina inducida por el ejercicio sí se produjo o no, pico anterior, pendiente del segmento ST del ejercicio máximo, número de vasos principales (0 – 3) coloreados por fluoroscopia, clasificación de infarto cardíaco, diagnóstico de enfermedad cardíaca (ver Tabla 1).

Tabla 1

Descripción de atributos del conjunto de datos seleccionado.

Nº	Atributos	Valores
1	Age	
2	Sex	
3	Chest pain	0.angina típica 1.angina atípica 2.dolor no anginoso 3.asintomático
4	Resting blood pressure	90 - 300
5	Serum cholesterol	110 – 400
6	Fasting blood sugar	1.true 0.false 0.normal
7	Results electrocardiographic in repose	1.tener anomalía 2.mostrar hipertrofia
8	Maximum heart rate reached	110 - 300
9	Induced angina exercise	1.yes 0.no
10	Previous peak	0.0 – 3.0
11	ST segment slope of maximum exercise	0.upsloping 1.flat 2.downsloping
12	Number of glasses main colored by fluoroscopy	0 – 3
13	Infarction classification cardiac	0.normal 1.defecto fijo 2.defecto
14	Presence or not of heart disease	0.no disease 1.disease

Fuente: elaboración propia.

En la figura 1, se puede observar que el conjunto de datos tiene a 160 de los pacientes con resultados de no tener la presencia de la enfermedad al corazón, y a 139 de los pacientes con resultados de si tener la presencia de la enfermedad al corazón.

Google Colaboratory

Baume (2012) afirmó lo siguiente:

Que es una herramienta que permite escribir y ejecutar código Python en la nube utilizando los servidores potentes de Google. También es posible incluir texto enriquecido, links como también imágenes. En caso de necesitar altas prestaciones de cómputo, el entorno permite configurar algunas propiedades del equipo sobre el que se ejecuta el código. En definitiva, el uso de Google Colaboratory permite tener a disposición un entorno para llevar a cabo tareas que serían difíciles de realizar en un equipo personal. Por otro lado, siguiendo la idea de Google Drive, Google Colab brinda la opción de compartir los códigos realizados lo que es ideal para trabajos en equipo.

K Nearest Neighbor

Moujahid, Inza y Larrañaga (s.f.) afirmaron lo siguiente:

El método de clasificación basado en los k-vecinos más cercanos (k-NN o knearest neighbors) se fundamenta en que las propiedades de un dato x de entrada son similares a las de los datos de su vecindad,

entonces éste pertenece a la misma clase que la clase más frecuente de sus k vecinos más cercanos.

Zapata-Tapasco (2014) afirmó lo siguiente:

Esta regla es adecuada para abordar un problema de aprendizaje, debido a sus propiedades estadísticas bien establecidas y la sencillez de implementación en problemas reales. El algoritmo general del método de los k vecinos más cercanos asume que todos los ejemplos corresponden a puntos en un espacio p-dimensional R^p , los cuales tienen establecida una clase C. Los datos de entrenamiento son de la forma presentada en (1).

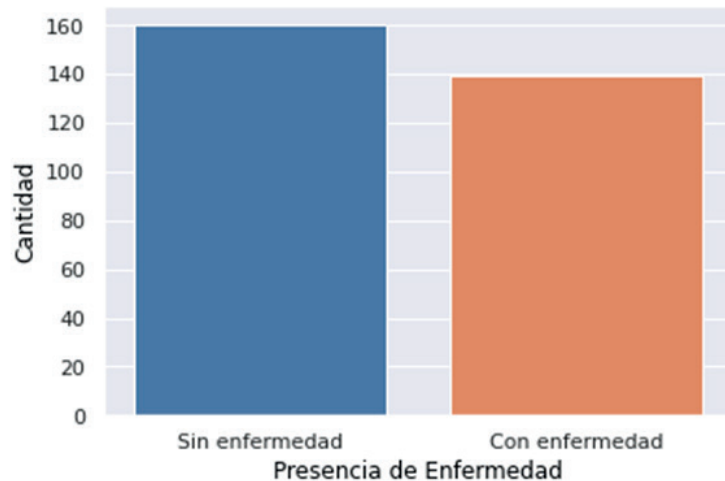
$$(x_i, c_i) = (x_{i1}, x_{i2}, \dots, x_{ip}, c_i) \quad (1)$$

“Para un nuevo punto X_q se calcula la distancia a los ejemplos de entrenamiento de acuerdo a la función de distancia escogida. Se seleccionan aquellos más cercanos y se asigna al nuevo dato una clase dependiendo del método de asignación” (Zapata-Tapasco, 2014).

Otros autores han afirmado lo siguiente:

En caso que se presente un empate entre dos o más clases, se debe contar con alguna regla heurística para su ruptura. Algunas de estas reglas consisten en seleccionar la clase a la cual pertenece el vecino más cercano o seleccionar la clase con distancia media menor. (Moujahid, Inza y Larrañaga, s.f.)

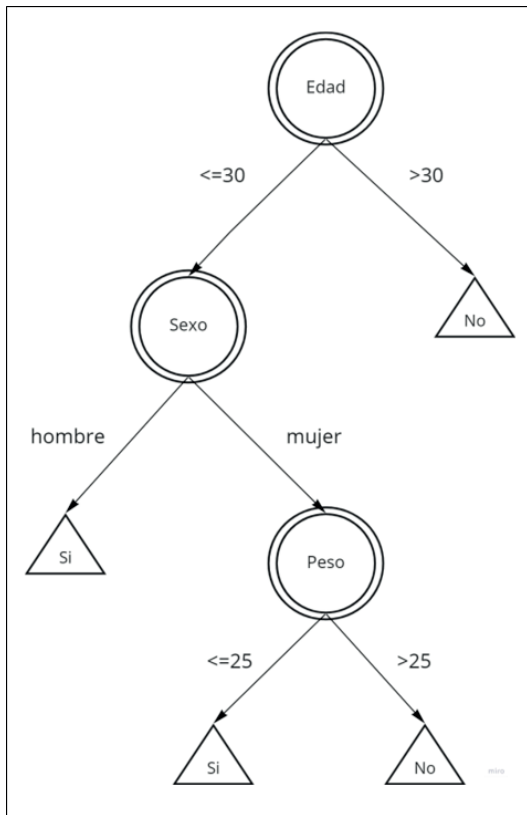
Figura 1
Cantidad de pacientes sin enfermedad y con enfermedad.



Decision Tree (ver Figura 2)

Figura 2

Ejemplo del diagrama de un árbol de decisión



Fuente: elaboración propia.

Rokach (2016) afirmó lo siguiente:

Es un clasificador expresado como una partición recursiva del espacio de instancia. El árbol de decisión consta de nodos que forman un árbol con raíz, lo que significa que es un árbol dirigido con un nodo llamado "raíz" que no tiene entrada bordes. Todos los demás nodos tienen exactamente un borde entrante. Un nodo con salida bordes se denomina nodo interno o de prueba. Todos los demás nodos se denominan hojas (también conocidos como nodos terminales o de decisión). En un árbol de decisión, cada nodo interno divide el espacio de instancia en dos o más subespacios de acuerdo con una cierta función discreta de los valores de los atributos de entrada.

Linear Regression

Bagnato (2018) afirmó lo siguiente:

La regresión lineal es un algoritmo de aprendizaje supervisado que se utiliza en

Machine Learning y también en estadística. En su versión más sencilla, lo que haremos es *dibujar una recta* que nos indicará la tendencia de un conjunto de datos continuos (si fueran discretos, utilizaríamos Regresión Logística). En estadísticas, regresión lineal es una aproximación para modelar la relación entre una variable escalar dependiente Y y una o más variables explicativas nombradas con X.

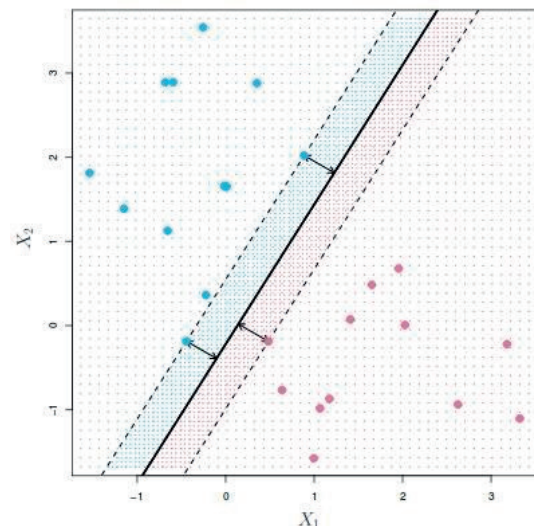
Support Vector Machine (ver Figura 3)

Smola et al. (1998) afirmaron lo siguiente:

La clasificación multi-clase está presente en muchos problemas del mundo real y la vida cotidiana, fundamentalmente inicia en las máquinas de soporte vectorial las que fueron diseñadas para tratar con problemas binarios (+/-1). Se podría decir que el algoritmo multi-clase más elegante, y sin duda el método más alineado con el principio de Vapnik de tratar siempre de resolver problemas directamente, el cual implica la modificación de la función objetivo SVM de tal manera que, de forma simultánea permite el cálculo de un clasificador multi-clase. (p. 233)

Figura 3

Maximal margin hyperplane (Amat, 2017).



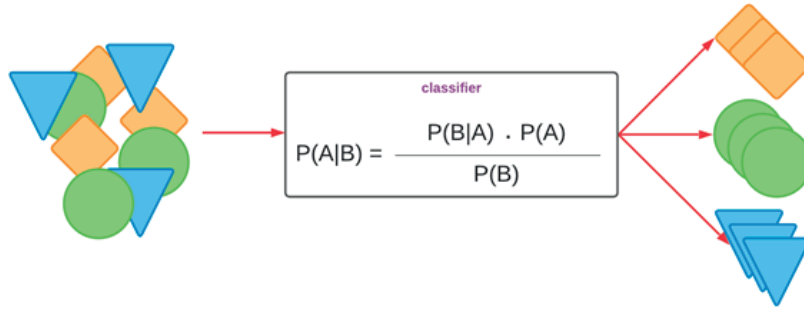
Naive Bayes (ver Figura 4)

Chandra et al. (2007) afirmaron lo siguiente:

El algoritmo clasificador Naïve-Bayes (NBC), es un clasificador probabilístico simple con fuerte suposición de independencia. Aunque la suposición de la independencia de

Figura 4

La fórmula simple del teorema de Bayes (Alam, 2022).



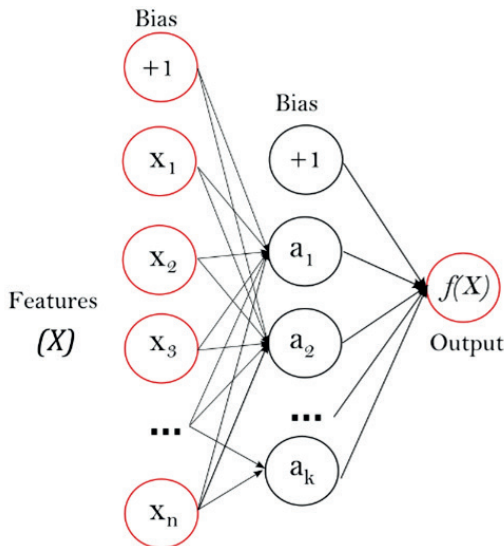
los atributos es generalmente una suposición pobre y se viola a menudo para los conjuntos de datos verdaderos. A menudo proporciona una mejor precisión de clasificación en conjuntos de datos en tiempo real que cualquier otro clasificador. También requiere una pequeña cantidad de datos de entrenamiento. El clasificador Naïve-Bayes aprende de los datos de entrenamiento y luego predice la clase de la instancia de prueba con la mayor probabilidad posterior. También es útil para datos dimensionales altos ya que la probabilidad de cada atributo se estima independientemente.

algoritmo de retropropagación. En retropropagación, los datos se procesan en la dirección de avance para estimar el objetivo y calcular la pérdida, luego se procesa en la dirección de retroceso ajustando los pesos para minimizar la pérdida.

MultiLayer Perceptron Prediction (ver Figura 5)

Figura 5

MLP de una capa oculta. (Pedregosa et al., 2011)



Pedregosa et al. (2011) afirmó lo siguiente:

Es un aprendizaje profundo simple y eficiente marco que funciona según el principio del

Random Forest

Amat (2020) menciona lo siguiente:

Un modelo Random Forest está formado por un conjunto (ensemble) de árboles de decisión individuales, cada uno entrenado con una muestra aleatoria extraída de los datos de entrenamiento originales mediante bootstrapping). Esto implica que cada árbol se entrena con unos datos ligeramente distintos. En cada árbol individual, las observaciones se van distribuyendo por bifurcaciones (nodos) generando la estructura del árbol hasta alcanzar un nodo terminal. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

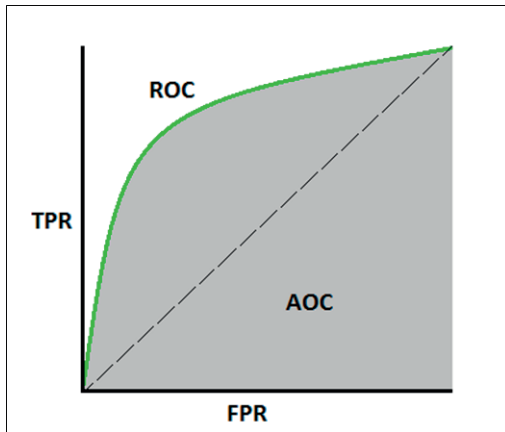
Curva Auc Roc

Narkhede (2022) afirmó lo siguiente:

Permite medir el rendimiento para encontrar problemas de clasificación en diversas configuraciones según el umbral. ROC es una curva de probabilidad y AUC representa el grado o medida de separabilidad. Indica cuánto es capaz el modelo de distinguir entre clases. Cuanto mayor sea el AUC, mejor será el modelo para predecir 0 clases como 0 y 1 clases como 1, recordando que el modelo que más se acerque a 1 será

considerado el mejor. Por analogía, cuanto mayor sea el resultado AUC, mejor será el modelo para poder distinguir los pacientes con enfermedad y sin enfermedad del corazón.

Figura 6
AUC - ROC Curve [Image 2]



(Image courtesy: My Photoshopped Collection)

El proceso para el desarrollo de este documento se muestra en la Figura 7. Iniciaremos con la información de nuestro conjunto de datos generado en base a los síntomas que pueden percibirse para un posible diagnóstico de enfermedad al corazón, los cuales servirán como entrada al algoritmo de predicción, luego, a través del modelo de evaluación que es el método de introducir una matriz de confusión para verificar el nivel de precisión en la clasificación de cada algoritmo, junto con la Curva

Roc para medir su rendimiento. Por último, obtendremos el algoritmo con la mayor precisión en la predicción de enfermedades del corazón.

III. RESULTADOS

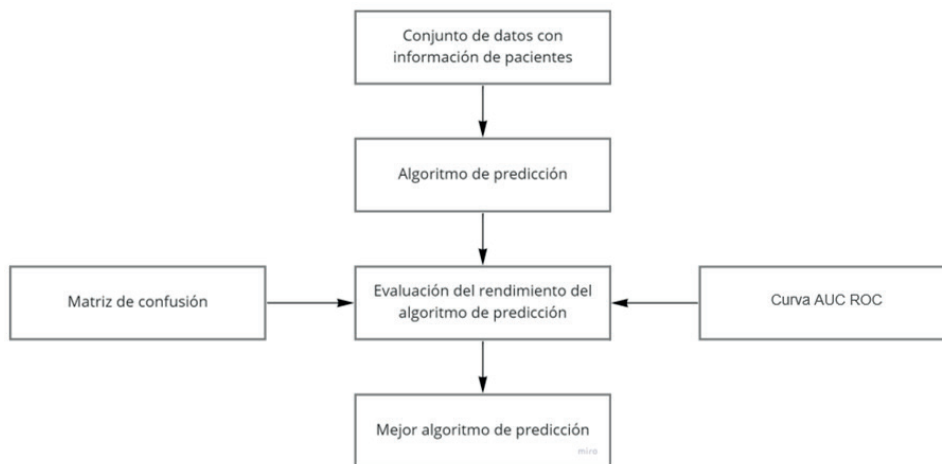
Para comparar los pros y los contras de los modelos de clasificación, es necesario proporcionar métricas para evaluar el nivel de rendimiento de los modelos. Aquí dividimos la muestra en cuatro clases como verdaderos ejemplos. (Verdadero positivo, TP), falso positivo (FP), ejemplos de verdadero negativo (Verdadero negativo, TN) y falso negativo ejemplos (Falso Negativo, FN) (Zapata-Tapasco et al , 2014). Sean TP, FP, TN y FN, respectivamente, los correspondientes números de muestras, $TP+FP+TN+FN=n$, n es el tamaño de la muestra, y la matriz de confusión del resultado de la clasificación se muestra en la siguiente tabla 2.

Tabla 2
Matriz de confusión. Fuente: elaboración propia.

Clases reales	Previsiones	
	Ejemplos verdaderos	Ejemplos falsos
Ejemplos verdaderos	TP	FN
Ejemplos falsos	FP	TN

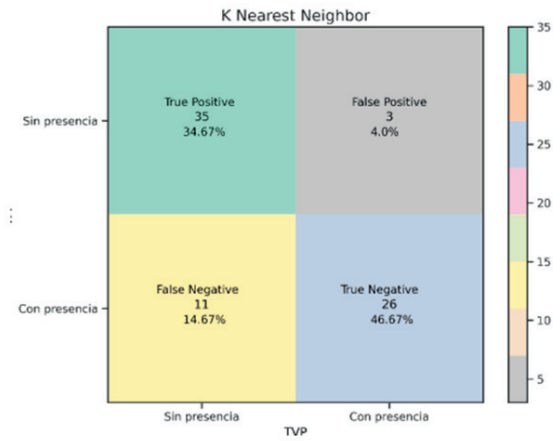
En este experimento, la relación entre el conjunto de entrenamiento y el conjunto de predicciones es 75:25. Usando los datos del conjunto de entrenamiento para que el modelo entrene, y se utilizó el modelo entrenado y predicción establecida como entrada en el componente de predicción.

Figura 7
Arquitectura de Procesos.



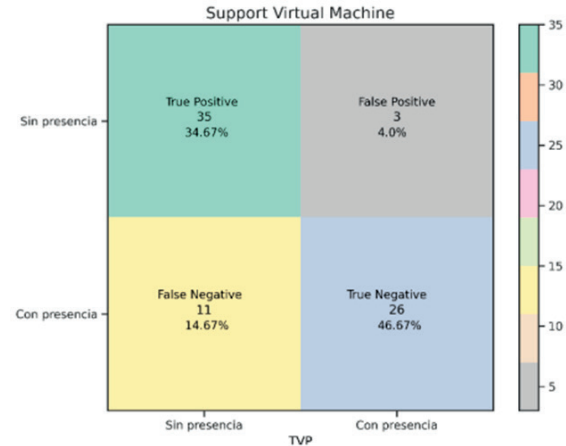
Fuente: elaboración propia.

Figura 8
Matriz de Confusión del Modelo K Nearest Neighbor.



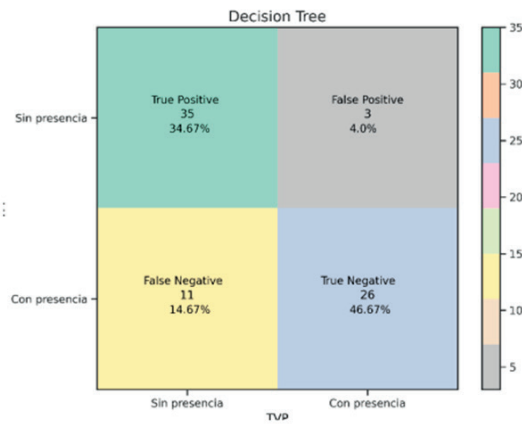
Fuente: Elaboración propia.

Figura 11
Matriz de Confusión del Modelo Linear Regression.



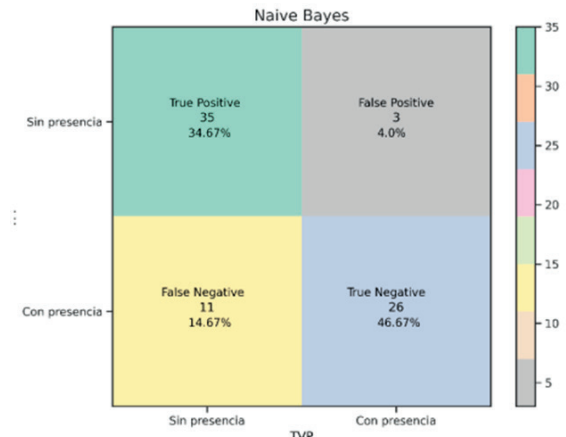
Fuente: Elaboración propia.

Figura 9
Matriz de Confusión del Modelo Decision Tree.



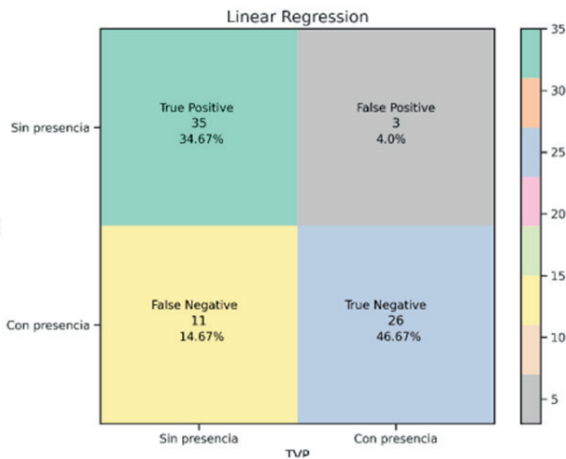
Fuente: Elaboración propia.

Figura 12
Matriz de Confusión del Modelo Naive Bayes.



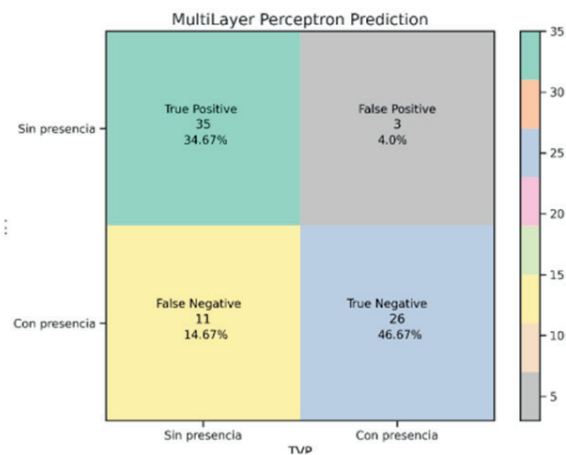
Fuente: Elaboración propia.

Figura 10
Matriz de Confusión del Modelo Linear Regression.



Fuente: Elaboración propia.

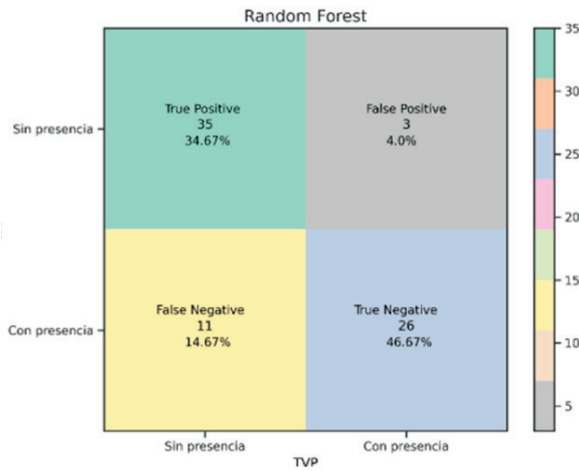
Figura 13
Matriz de Confusión del Modelo MultiLayer Perceptron Prediction.



Fuente: Elaboración propia.

Figura 14

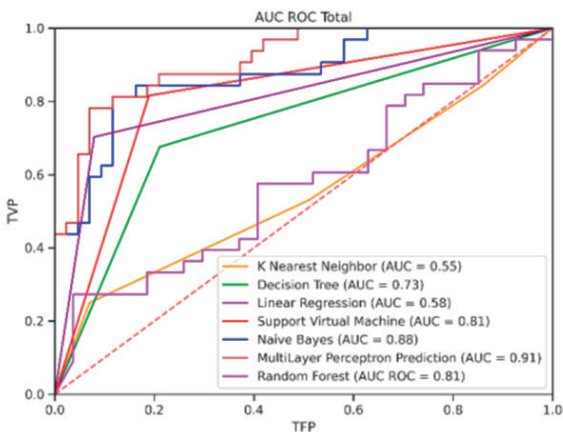
Matriz de Confusión del Modelo Random Forest.



Fuente: Elaboración propia.

Figura 15

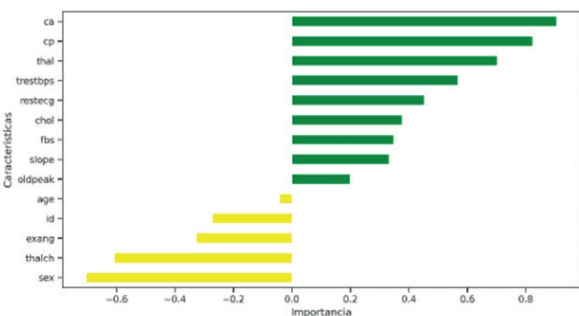
Comparación de las curvas ROC (Receiver Operating Characteristic) y AUC (Area Under the ROC Curve) de los siete modelos aplicados: Knearest neighbors, Decision Tree, Linear Regression, Support Vector Machine, Naive Bayes, Multilayer Perceptron Prediction, Random Forest.



Fuente: Elaboración propia.

Figura 16

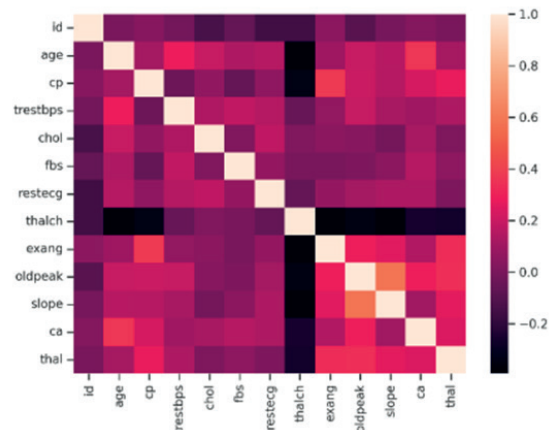
Gráfica de Barras para mostrar el atributo más importante que influye de forma positiva y/o negativa en la variable 'class' del DataSet.



Fuente: Elaboración propia.

Figura 17

Matriz de correlación.

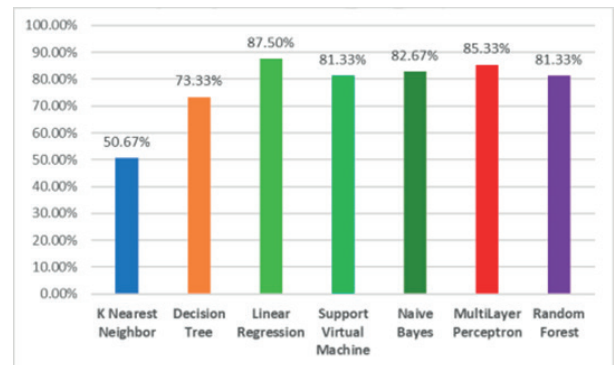


Fuente: Elaboración propia.

Resumimos los resultados de los siete algoritmos de clasificaciones anteriores y realizamos la comparación.

Figura 18

Comparación del nivel de precisión de predicción de los siete modelos: Knearest neighbors, Decision Tree, Linear Regression, Support Vector Machine, Naive Bayes, Multilayer Perceptron Prediction, Random Forest.



Fuente: Elaboración propia.

Tabla 3

Comparación de los accuracy de los siete modelos aplicados: Knearest neighbors, Decision Tree, Linear Regression, Support Vector Machine, Naive Bayes, Multilayer Perceptron Prediction, Random Forest.

Modelos	Accuracy
K Nearest Neighbor	50.67%
Decision Tree	73.33%
Linear Regression	87.5%
Support Virtual Machine	81.33%
Naive Bayes	82.67%
MultiLayer Perceptron	85.33%
Random Forest	81.33%

IV. DISCUSIONES

Con respecto a las curvas ROC (Receiver Operating Characteristic) aplicado al Dataset para medir la eficacia de la predicción del modelo, el que mejor se comporta es el Modelo MultiLayer Perceptron Prediction con un AUC (área bajo la curva) de 0.91 el cual nos da una idea de que el modelo funciona bien. La siguiente curva ROC mejor comportada es la del modelo Naive Bayes con un área bajo la curva (AUC) de 0.88.

En relación con la importancia de los atributos con la variable class en el Dataset y mediante el modelo de Regresión Lineal, se tiene que: el atributo "Ca - Clasificación de infarto cardíaco" es el más importante, pues influye positivamente en caer tener presencia de alguna enfermedad al corazón; mientras el atributo "Sex - Sexo" influye negativamente.

V. RECOMENDACIONES

Se recomienda continuar investigando, realizando más pruebas a diversas personas que presenten o no la enfermedad del corazón con el fin de aumentar los registros para un mejor desempeño de los algoritmos de clasificación.

VI. CONCLUSIONES

La información de las pruebas realizadas permite obtener una precisión considerable en cuanto al modelo de Regresión Lineal indicando que trabaja de manera correcta con la misma, sin embargo, con una cantidad mayor de registros se podría mejorar aún más los resultados obtenidos actualmente.

La herramienta Colab permitió la codificación en el lenguaje Python, el cuál a través de sus servicios de procesamiento masivo de datos (tecnologías de CPU, GPU, TPU), la obtención de resultados óptimos y en poco tiempo, todo gracias a los potentes servidores que pueden ejecutar los códigos por bloque o sectores, para mejorar el rendimiento en los procesos de algoritmos de clasificación.

Con el avance continuo de los métodos de minería de datos, hemos estudiado varios métodos de diagnóstico de enfermedades para el corazón.

Finalmente, podemos ver que los algoritmos de minería de datos a través de la investigación, las técnicas de aprendizaje automático y otras tecnologías, han hecho contribuciones destacadas en el campo de la medicina y el diagnóstico de enfermedades. Esperando que puede ayudar a los médicos a hacer mejores juicios en cuanto a los

diagnósticos tempranos sobre el estado de una enfermedad, minimizando el riesgo de muerte.

VII. REFERENCIAS

- [1] Alam, B. (2022). Implementing Naive Bayes Classification Using Python. Recuperado el 20 de julio de 2022 de <https://hands-on.cloud/implementing-naive-bayes-classification-using-python/>
- [2] Amat, J. (2017). Máquinas de Vector Soporte (Support Vector Machines, SVMs). Recuperado el 20 de julio de 2022 de https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines
- [3] Amat, J. (2020). Random Forest con Python. Recuperado el 20 de julio de 2022 de https://www.cienciadedatos.net/documentos/py08_random_forest_python.html
- [4] Bagnato, J.I. (2018). Aprende Machine Learning, Regresión Lineal en español con Python. Recuperado el 20 de julio de 2022 de <https://www.aprendemachinelearning.com/tag/regresion-lineal/#:~:text=La%20regresi%C3%B3n%20lineal%20es%20un,discretos%2C%20utilizar%20ADamos%20Regresi%C3%B3n%20Log%C3%ADstica>
- [5] Baume, G. (2021) Breve introducción a Google Colab. Recuperado el 20 de julio de 2022 de <http://fcaglp.unlp.edu.ar/~gbaume/grupo/Publicaciones/Apuntes/GoogleColab.pdf>
- [6] Boderó, E., y López, M. (2019). Google Colaboratory como alternativa para el procesamiento de una red neuronal convolucional. *Revista Espacios*. Vol. 41, N° 07, Pág. 22. ISSN 07981015.
- [7] Chandra, B., M. Gupta y M.P. Gupta. (2007). Robust Approach for Estimating Probabilities in Naive-Bayes Classifier. In International Conference on Pattern Recognition and Machine Intelligence (pp. 11-16), Springer, Berlin, Heidelberg, December.
- [8] López, A. y Macaya, C. (2009). Libro de la Salud Cardiovascular del Hospital Clínico San Carlos y La Fundación BBVA. *Fundación BBVA*. ed. 612.1
- [9] Mercado, D., Pedraza, L., Martínez, E. (2015). Comparación de Redes Neuronales aplicadas a la predicción de Series de Tiempo. *Prospect*. Vol 13, N° 2, 88-95. <https://doi.org/10.15665/rp.v13i2.491>

- [10] Mosquera, R., Castrillón, O. D., y Parra, L. (2018). Máquinas de Soporte Vectorial, Clasificador Naïve Bayes y Algoritmos Genéticos para la Predicción de Riesgos Psicosociales en Docentes de Colegios Públicos Colombianos. *Información Tecnológica*, 29(6), 153-162. <https://dx.doi.org/10.4067/S0718-07642018000600153>
- [11] Moujahid, A., Inza, I., Larrañaga, P. Clasificadores KNN. Departamento de Ciencias de la Computación e Inteligencia Artificial. Universidad del País Vasco Euscal Herriko Unibertsitatea. Recuperado el 20 de julio de 2022 de <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf>
- [12] Narkhede, S. Understanding AUC-ROC Curve. Recuperado el 20 de julio de 2022 de <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>
- [13] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, *JMLR* 12, pp. 2825-2830.
- [14] Smola, A.J. y B. Schölkopf. (1998). Learning With Kernels (Vol. 4). *GMD-Forschungszentrum Informationstechnik*.
- [15] Sullá-Torres, J., Gómez-Campos, R., y Cossio-Bolaños, M.A. (2018). Aplicación de un árbol de decisión difusa con clasificación de ambigüedad para determinar el exceso de peso en escolares. *Revista Mexicana de Ingeniería Biomédica*, 39(2), 128-143. <https://doi.org/10.17488/rmib.39.2.1>
- [16] Rokach, L., Maimon, O. (2005). Decision Tree. The Data Mining and Knowledge Discovery Handbook. *Springer*. (pp.165-192). https://doi.org/10.1007/0387-25465-X_9
- [17] Rokach, L. (2016) Decision Trees. Recuperado el 20 de julio de 2022 de <http://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf>
- [18] Zapata-Tapasco, A., Mora-Flórez, J., y Pérez-Londoño, S. (2014). Método basado en clasificadores k-NN parametrizados con algoritmos genéticos y la estimación de la reactancia para localización de fallas en sistemas de distribución. *Revista Facultad de Ingeniería Universidad de Antioquia* (70), 220-232. [Fecha de consulta 25 de mayo de 2022]. ISSN: 0120-6230. Recuperado el 20 de julio de 2022 de <https://www.redalyc.org/articulo.oa?id=43030033020>
- [19] Zhihua, Z. (2016), Machine learning. Beijing: Tsinghua University Pres. *Research Square*. pp.121-139, 298-300.

Fuentes de financiamiento:

Propia.

Conflictos de interés:

El autor declara no tener conflictos de interés.

Contribución del Autor

El autor analizó el rendimiento y precisión de siete algoritmos de clasificación utilizando el servicio de Google Collaboratory; del cual eligió el algoritmo que alcanzó el mejor resultado, siendo el algoritmo de Regresión Lineal la mejor alternativa, demostrando el mejor comportamiento al ejecutar la aplicación de machine learning de predicciones para pacientes con enfermedades al corazón de manera temprana. Lo que permitirá a los médicos un diagnóstico previo como apoyo anticipado al momento de analizar los casos a nuevos pacientes