Modelo inteligente predictivo de IMC en pacientes de nutricionistas mediante algoritmos de Machine Learning: Regresión Logística y Redes Neuronales. Intelligent predictive model of BMI in nutritionists' patients using Machine Learning algorithms: Logistic Regression and Neural Networks.

Eleanor Varela-Tapia, MSig,[1], Iván Acosta-Guzmán, Ing.[1], Christopher Acosta-Varela, estudiante[2], Patricia María Marcillo-Sánchez, Lsi.[1], Ing. Darwin Guillermo Patiño-Pérez, Phd.[1,], and Joselyn Tumbaco-Bravo Ing.[1]
[1]Universidad de Guayaquil, Ecuador, eleanor.varelat@ug.edu.ec, ivan.acostag@ug.edu.ec, patricia.marcillos@ug.edu.ec, darwin.patinop@ug.edu.ec, joselyn.tumbacob@ug.edu.ec
[2]Escuela Superior Politécnica del Litoral, Ecuador, chriacos@espol.edu.ec
[1]https://orcid.org/my-orcid?orcid=0000-0002-5357-4046

Abstract – The residents of the Bastión Popular Cooperative in Ecuador have health problems due to inadequate food due to limited economic resources, poor nutrition, and lack of knowledge about healthy food in the people of this vulnerable area. On the other hand, this area has a limited number of nutritionists to serve the population, for this reason it is necessary that the professional can serve them more quickly to be able to cover the largest number of them during the working day, in this situation this project proposes the creation of a technological solution to help the specialist to minimize the time spent on manual calculations made during the consultation. The nutritionist as part of the diet prescription process must consider several variables and conditions of the patient to establish the correct BMI (body mass index) of each type of patient, so this study promotes the creation of an intelligent model that allows the doctor to obtain the BMI in a more agile way, for this case qualitative, quantitative and experimental research was applied using Machine Learning algorithms, Multinomial Logistic Regression and Dense Layer Neural Networks. As results it was obtained that the model based on Multinomial Logistic Regression obtained an efficiency level of 97.9% in test data, while the model based on Neural Networks with dense layer obtained an efficiency level of 98.95% for test data. Therefore, it was found that the Neural Networks classifier allows the nutritionist to avoid manual calculations and instead obtain the BMI with a high level of efficiency, thus saving time in the initial phase of the patient consultation, giving him/her the opportunity to use that time to attend more patients in the waiting room.

Translated with www.DeepL.com/Translator (free version).

Keywords-- Keywords: Multinomial Logistic Regression, Multiclass, Dense Neural Networks, Machine Learning, Supervised Learning.

Resumen – Los pobladores de la Cooperativa Bastión Popular en el Ecuador, presentan problemas de salud debido a la alimentación inadecuada producto de los limitados recursos económicos, mala alimentación, y falta de conocimiento acerca de alimentos saludables en los pobladores de esta zona vulnerable. Por otra parte, esta zona posee un limitado número de nutricionistas para atender a la población, por esta razón se necesita que el profesional pueda atenderlos de manera más rápida para que logre cubrir la mayor cantidad de ellos durante la jornada laboral, ante esa situación el presente proyecto propone la creación de una solución tecnológica que ayude al especialista a minimizar el tiempo empleado en cálculos manuales realizados durante la consulta. El nutricionista como parte del proceso de prescripción de la dieta debe considerar varias variables y condiciones del paciente para llegar a establecer el índice IMC correcto (índice de masa corporal) de cada tipo de paciente, por lo cual este estudio impulso la creación de un modelo inteligente que permita al médico obtener el IMC de manera más ágil, para este caso se aplicó la investigación cualitativa, cuantitativa y experimental empleándose algoritmos de Machine Learning, Regresión Logística Multinomial y Redes Neuronales de capa densa. Como resultados se obtuvo que con el modelo basado Regresión Logística Multinomial obtuvo un nivel de eficiencia del 97.9% en datos de prueba (test), mientras que el modelo basado en Redes Neuronales con capa densa obtuvo un nivel de eficiencia del 98,95% para datos de prueba (test). Encontrándose por ello que clasificador de Redes Neuronales permite al nutricionista evitar los cálculos manuales y en su defecto obtener el IMC con alto nivel de eficiencia, lográndose así ahorrar tiempo en la fase inicial de la consulta de los pacientes, dándole la oportunidad de destinar ese tiempo a lograr atender más pacientes que se encuentren en sala de espera.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2022.1.1.791
ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Palabras Clave: Regresión Logística Multinomial, Multiclase, Redes neuronales densas, Machine Learning, Aprendizaje Supervisado.

I. INTRODUCCIÓN

A. Planteamiento del problema

Limitada cantidad de médicos especialistas en nutrición presentes en la Cooperativa Bastión Popular destinados a atender las necesidades de consultas de salud nutricional debido a alimentación inadecuada por recursos limitados de los pobladores de esta zona vulnerable.

B. Objetivo general

Creación de un modelo inteligente que asista al médico en la obtención del índice de IMC de manera más directa y ágil a través de Algoritmos Supervisados de Machine Learning que permitan reducir el tiempo destinado a cálculos manuales pudiendo redestinar ese tiempo ahorrado a la atención de más pacientes.

Antecedentes

Desde la llegada de la pandemia COVID19 la cual se presentó a nivel mundial entre finales del 2019 e inicios del 2020 propagándose en las diversas regiones del mundo, muchas personas sufrieron grandes impactos en pérdidas humanas como familiares, amigos, afectándose en paralelo los ingresos por elevada cantidad de despidos en los trabajos ante la caída de consumo de los productos y servicios [1] debido de las cuarentenas continuas impuestas por los gobiernos en diversos países afectados. El recorte de personal, impacto a las familiar reduciendo sus ingresos mensuales teniendo que limitar sus gastas entre ellos el alimento diario, provocando el descuido de la alimentación adecuada [2] , si bien las actividades virtuales han tomado un auge luego de dos años consecutivos de pandemia el teletrabajo y la teleducación no ha podido llegar al 100% de la población, y ha promovió el aumento del sedentarismo sumado al consumo de comidas denominadas comidas que incluyen alta ingesta de azúcar, grasas y pocos nutrientes, ya que estos alimentos se preparan de una manera sencilla o se compra bajo el concepto de comida rápida en el Ecuador.

Tomando en consideración que la dieta balaceada debe incluir en granos, lácteos, legumbres, frutas, proteínas, hortalizas, en proteínas, en la época prepandemia el 50% de los ecuatorianos no consumía una dieta adecuada, en su defecto consumían productos que solo saciaban el hambre, pero que sin cubrir los requisitos necesarios para considerarse saludable. La comida rápida tiene diferentes tipos de procesamientos que le ha permitido posicionarse en volumen consumido por encima de los alimentos nutritivos, dado que se consiguen mediante una preparación de corto tiempo, a precio accesible y adicionalmente por falta de conocimiento nutricional en la población [3].

La nutricionista Paola Jaramillo menciona que los hábitos alimenticios erróneos es el origen de múltiples enfermedades que tienden a causar graves daños en la salud de quien los ingiere [4].

En La Cooperativa Bastión Popular, la mayoría de los pobladores se encuentran con la situación de que su sueldo no supera al sueldo básico, algunos de ellos poseen emprendimientos, de los cuales varios debieron cerrar sus negocios porque no pudieron adaptarse a la modalidad online que surgía como una alternativa ante la situación de pandemia que debilitaba la estabilidad de los negocios, por otro lado existen los trabajadores ambulantes, quienes se vieron forzados a no poder salir a las calles ante la ordenanza de cuarentena, generándose que muchas personas no tuviesen lo suficiente para alimentar a sus familias ante la falta presupuesto, agudizando más la situación de salud de los pobladores empujándolos a optar por comida rápida y en casos extremos a mantenerse con una comida al día diaria, impactando en el mediano y largo plazo en el crecimiento de la población con desnutrición y obesidad. [5]

A finales del 2021 las personas han logrado en cierta medida adaptarse a la nueva normalidad, comenzado a recuperarse lentamente en la estabilidad psicológica y económica vía emprendimientos en casa y entregas a domicilio, ante la apertura de los gobiernos para dar paso a las entregas de alimento a domicilio.

Debido a la importancia de mantener un peso saludable [6] y considerando la baja cantidad de nutricionistas que atenden esta zona vulnerable, surge este proyecto el cual persigue aportar con un Modelo de Aprendizaje Automático entrenado mediante algoritmos de Machine Learning que apoye al médico especialista en Nutrición de Bastión Popular y del Ecuador u otro país ayudándolo a reducir los tiempos que se invierten en cálculos manuales que al momento requiere realizar el especialista. [7]

Por lo expuesto el presente estudio propone la creación de herramientas de apoyo vía uso de modelos inteligentes iniciando por la creación de uno que se encargue de identificar Índice de Masa Corporal cuya fórmula de cálculo varía dependiendo del tipo de paciente, en ese proceso de cálculo manual el medio debe considerar factores como la talla, peso, edad, sexo, actividad física, en el caso de las mujeres también se adiciona la consideración de estado de gestación. [8].

II. REVISIÓN DE LITERATURA

A. Análisis multivariante (AM)

La rama de estadística encargada del estudio, análisis, representación e interpretación de datos obtenidos de observaciones realizadas a más de una variable estadística sobre una muestra de sujetos de estudio se conoce como Análisis Multivariante. [9]

B. Clasificación estadística

La actividad de clasificar elementos de un determinado conjunto finito involucra la habilidad de establecer una división del conjunto de datos en subconjuntos más pequeños normalmente homogéneos, mediante el seguimiento de un criterio especifico empleado para lograr la clasificación.

Digital Object Identifier: (only for full papers, inserted by LACCEI).
ISSN, ISBN: (to be inserted by LACCEI).
DO NOT REMOVE

Cada sujeto de estudio o elemento del conjunto inicial pertenecerá a un solo subconjunto, el cual tiene un nombre que lo caracteriza. [9]

C. *Regresión logística*

Técnica de dependencia que busca explicar el comportamiento de un fenómeno definido por una variable (llamada dependiente Y) en función de una serie de factores componentes o elementos (Xs) que actúan como variables Independientes y están relacionadas con la variable dependiente (Y), a esta técnica se la conoce como modelo de probabilidad directa, la cual fue desarrollada por el estadístico DR. Cox en 185 [10]

D. *Regresión logística multinomial*

La regresión logística multinomial (MLR), también llamada regresión logística multiclase, corresponde una generalización de la regresión logística binaria discreta, pudiéndose emplear en el cálculo de un resultado para una variable Y multiclase discreta. Se utiliza para entender en que forma las características independientes (Xs, 6 en el presente estudio) están relacionadas con la Y (índice de masa corporal para el presente estudio).
Como técnica de regresión tradicional, se utiliza para predecir una variable dependiente categórica (Yc) a partir de múltiples predictores independientes (Xs). En este trabajo de investigación la variable Yc es de tipo multiclase que son los 4 tipos de estado del peso del paciente al llegar al consultorio: 1. Bajo peso, 2. Peso normal, 3. Peso mórbido y 4. Sobrepeso.[11]

E. *Regresión logística ordinal*

Los modelos de regresión logística ordinal utilizan la naturaleza ordinal de la variable dependiente mediante la descripción de varios modos de ordenación estocástica, este hecho elimina la necesidad de asignar puntajes o asumir de otra manera la cardinalidad en lugar de la ordinalidad. [12]

F. *Redes Neuronales*

Las redes neuronales artificiales están en capacidad de analizar y estudiar la estructura de la información y elaborar patrones o secuencias de datos que sean útiles para realizar clasificaciones predictivas o exploratorias [13].

Las redes neuronales tienen una importante capacidad para detectar datos similares, crear patrones y realizar una serie de acciones que permita generar como resultado datos importantes para ser estudiados y aplicados por expertos de diferentes áreas. [14].

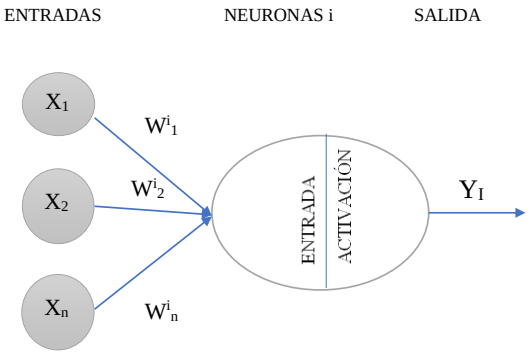


Fig. 1 Modelo sencillo de red neuronal.[17]

En la figura 1 se puede observar un esquema básico donde las neuronas se interconectan entre sí, en la red, las neuronas emplean una función determinada a sus entradas (valores) procedentes de las relaciones con las otras neuronas. [15]

En los Redes neuronales Artificiales (RNA) se enlazan los nodos mediante de sinapsis, dicha forma de enlace define la conducta de la red. A continuación, una de las estructuras más usadas llamada Perceptrón Multicapa [16]

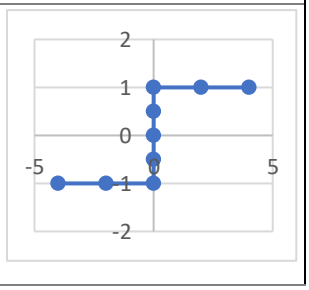
G. *Función de entrada*

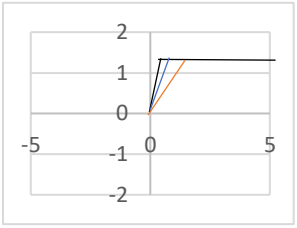
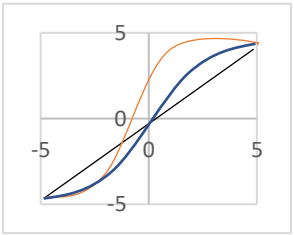
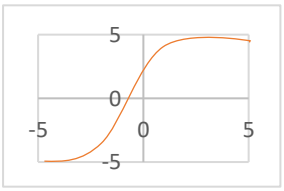
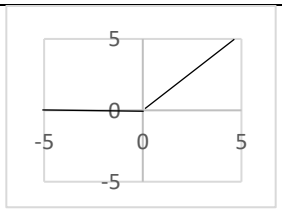
El propósito de la función de entrada es combinar varias entradas con sus valores, añadir los valores conseguidos de todas las conexiones de entrada y obtener un valor único. [16b]

H. *Función de activación*

Esta función también llamada función transferencia recibe el valor deducido por la función de combinación, el cual modifica antes de trasladarlo a la salida [17]

TABLA 1 TABLA DE FUNCIONES DE ACTIVACIÓN

Nombre	Función	Representación
La Función escalón	$y(x) = \begin{cases} 1 \text{ si } x \geq \alpha, \\ -1 \text{ si } x \leq \alpha, \end{cases}$ *Donde* α *= valor umbral de la función de transferencia.*	

La Función Lineal, Produce combinaciones lineales de entrada.	$y(x) = \beta x.$	 $\beta = 0.5 \;\; \beta = -1 \;\; \beta = 2$
La Función sigmoide o función logística.	$y(x) = \frac{1}{1 + e^{\frac{-x}{\rho}}}$	 $\rho = 0.5 \;\; \rho = -1 \;\; \rho = 2$
Tangente hiperbólica	$y(x) = \tanh(x) = \frac{senh(x)}{cosh(x)} = \frac{1 - e^{-2x}}{1 + e^{-2x}}$	
Función rectificadora (ReLU)	$y(x) = \max(0, x).$	

I. *Función de Salida*

Esta función da el valor de salida de la neurona, en base al estado de activación de la neurona. [17]

$$y_i(t) = f(net_i(t))$$

J. *Matriz de confusión*

Una matriz de confusión permite identificación de la calidad de aprendizaje supervisado logrado por un algoritmo.

La matriz de confusión puede dar 4 posibles tipos de resultados:

- Verdadero Positivo (VP): Dando como positivo el valor real y prediciendo la prueba un positivo.
- Verdadero negativo (VN): Dando como negativo el valor real y prediciendo la prueba un negativo.
- Falso negativo (FN): Dando como negativo el valor real y prediciendo la prueba un negativo.
- Falso positivo (FP): Dando como negativo el valor real y prediciendo la prueba un positivo. [18]

K. *Métricas*

Para la evaluación algoritmos ML se cuenta con varias métricas [208] que combinan los diversos tipos de resultados arrojados en la matriz de confusión, pudiendo con ellos generarse las siguientes métricas:

Métricas básicas
- Accuracy (Índice general de calidad)
- Precision (Precisión)
- Recall (Sensibilidad)
- Specifity (Especificidad)

Métricas Adicionales
- F1-Score (F1, aka F-Score, F-Measure)
- AUC (Área bajo la curva ROC)

En este estudio se han obtenido las métricas 1, 3, 4 y 6 para para complementar el análisis del rendimiento del modelo, debido a que el Accuracy no es suficiente para obtener conclusiones por tratarse de un dataset con datos no balanceados en las clases de la variable dependente. La curva ROC proporciona una representación global de la exactitud diagnóstica, para la curva ROC creciente, refleja el compromiso existente entre sensibilidad y especificidad [23]

Matriz de Confusion		Valor Estimado por el Modelo	
		Negativo (N)	Positivo (P)
Valor Real	Negativo	a: TN	b: FP
	Positivo	c: FN	d: TP

Metrica	Formula
Specifity (Especificidad)	a/(a+b)
Recall (Sensibilidad, exaustividad)	d/(d+c)
Presicion (Precisión)	d/(d+b)
Accuracy (accurracy)	(a + d) / (a + b + c + d)
F1-score (aka F-Score, F-Measure)	2*(Recall * Precision) / (Recall + Precision)
AUC (Area under the ROC curve) El área bajo la curva ROC	Medida de lo bien que un parámetro puede distinguir entre dos grupos de diagnóstico (enfermo/normal).

FIG. 2 MATRIZ DE CONFUSIÓN, COMPOSICIÓN Y COMPRENSIÓN DE MÉTRICAS.

TABLA 2. MÉTRICAS BÁSICAS

Métricas Básicas	Formula
Specifity (Especificidad) Porcentaje de casos negativos reales detectados correctamente	a/(a+b)
Recall (Sensibilidad, exhaustividad). Porcentaje de casos positivos reales detectados correctamente	d/(d+c)
Presicion (Precisión) Porcentaje de predicciones positivas correctas	d/(d+b)
Accuracy (accurracy) Porcentaje de predicciones correctas. No recomendado cuando se emplea datasets no equilibrados	$\frac{(a + d)}{(a + b + c + d)}$

Para la resolución de problemas en los cuales la variable de salida está compuesta por dos valores diferentes (clases) las meticas anteriores son suficientes, sin embargo, cuando se trata de variables de salida en las cuales se cuenta con más de dos clases (en este estudio se tienen 4 clases) se recomienda el uso de métricas adicionales las cuales son F1 y AUC para complementar la evaluación del Modelo.[19]

TABLA 3. MÉTRICAS ADICIONALES

Métricas Adicionales para salida Multiclase	Formula
F1-score (aka F-Score / F-Measure) Es la media armónica (promedio) de la Precisión y Recall.	2*(Recall * Precision) / (Recall + Precision)
AUC (Area under the ROC curve) El área bajo la curva ROC es una medida de lo bien que un parámetro puede distinguir entre dos clases. la curva ROC coincidirá con la diagonal (el área = 0,5) para casos no distinguibles, el área bajo la curva ROC será igual a 1 para casos para clases con una separación adecuada de los valores de los grupos que confirman las clases. [19]	

III. METODOLOGÍA

A. *Investigación exploratoria*

Se utilizó distintas fuentes bibliográficas para la búsqueda de datos relacionados con el objeto de estudio, en este trabajo se procedió a identificar fuentes de datos disponibles (datasets) a nivel mundial relacionados con pacientes de especialistas de nutrición, escogiéndose la encuesta que contaba con las variables de entrada y salida de este objeto de estudio.[20]

B. *Investigación cualitativa*

Se realizó entrevistas a especialistas de la nutrición de Guayaquil, en las cuales se identificó que el especialista emplea un tiempo valioso al inicio de la consulta, pudiéndose reducir ese tiempo a través del uso de un módulo de Machine Learning que realice la predicción de estado del peso del paciente, optimizando el tiempo del médico el cual podrá atender más pacientes en su jornada laboral. [21]

C. *Investigación cuantitativa*

De los 178 encuestados se identificó que el 82% de los moradores encuestados del sector Bastión Popular Bloque 10 A, desean ser atendidos por un médico en menos de 30 minutos, por lo cual apoyan al médico para que adopte tecnologías que le ayuden a reducir el tiempo que emplea el medico al ejecutar cálculos manuales del factor IMC

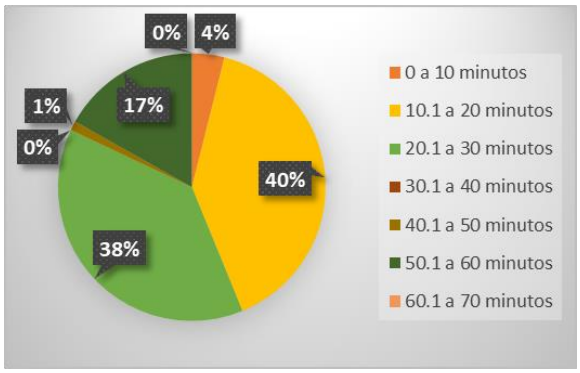


FIG. 3. TIEMPO ACEPTABLE PARA UNA CONSULTA DE NUTRICIÓN SEGÚN LA MUESTRA DE POBLADORES ENCUESTADOS DE BASTIÓN POPULAR.

D. *Lenguaje de Programación para Inteligencia Artificial*

Se analizaron los cuatro lenguajes de programación referente al índice TIOBE, se concluye que Python es el más adecuado para realizar el módulo de red neuronal, debido a que cuenta con un amplio rango de librerías para inteligencia artificial.[21]

Cuando se trata de análisis estadístico, R es el lenguaje de programación apropiado. Si se trata de tareas relacionadas con la visión por computadora, Matlab es la opción preferida. Si se trata de tareas relacionadas con la bioinformática o la biología, entonces Julia es el lenguaje de programación elegido. Pero si se trata de tareas generales como el procesamiento de datos y el procesamiento de resultados, entonces Python es el lenguaje de programación más apropiado [22].

E. *Entorno de desarrollo integrado (IDE) de PC Jupyter Notebook*

Se analizaron varios entornos de desarrollo entre ellos uno de PC y otro de acceso a recursos en la nube.

Interfaz web de código abierto es como se define Jupyter Notebook, que incluye ilustraciones, audios, textos, así mismo la ejecución de código mediante el navegador en varios lenguajes. Dicha ejecución se lleva a cabo a través de la comunicación con el núcleo, kernel, de cálculo. La interfaz web de Jupyter Notebook, en la actualidad además del núcleo de cálculo de Python, ha accedido el aumento de núcleos disponibles.

TABLA 4. CARACTERÍSTICAS DE JUPYTER

	• Código abierto • Admite hasta 40 idiomas e incluye lenguajes para Machine Learning. • Widgets interactivos • Incluye librerías científicas como Numpy, Matplotlib, SciPy, etc.

F. Entorno de desarrollo integrado (IDE) en la nube

Google Colaboratory llamado "Colab", es un entorno en línea gratuito basado en el principio de trabajo de la plataforma Jupyter notebook, pero Colab corre en la nube, el cual permite entrenar módulos de Aprendizaje Automático (Machine Learning), Aprendizaje Profundo (Deep Learning), Procesamiento de Lenguaje Natural NLP, entre otros. Brinda 12 horas de tiempo de ejecución continua, al cerrar la sesión los datos cargados se borran, pero el código desarrollado permanece almacenado y disponible en la cuenta Google del usuario, en el directorio Colaboratory de su cuenta Gmail. Colab permite ejecutar y programar en Python en un navegador y tiene las ventajas [21]: de

- No requiere configuración
- Da acceso gratuito a CPU
- Permite compartir contenido fácilmente

TABLA 5. UNIDADES DE PROCESAMIENTO DISPPONIBLES EN GOOGLE COLAB

CPU	GPU	TPU
Intel Xeon Processor with two cores @2.30Ghz and 13 GB RAM	Up to Tesla K80 with 12 GB of GDDR5 VRAM, Intel Xeon Processor with two cores @ 2.20 GHz and 13 GB RAM	Cloud TPU with 180 teraflops of computation, Intel Xeon Processor with cores @ 2.30 GHz and 13 GB RAM

En el presente estudio se escogió la Interfaz IDE Google Colab por su accesibilidad a recursos de hardware disponibles en la nube. Para el desarrollo de la solución de machine Learning se emplearon librerías de lenguaje Python como son:
Pandas, Numpy, Matplotlib, Keras, Tensorflow, Scikit-Learn, entre otras.

G. Preparación de conjunto de datos

Para esta investigación se realizó la búsqueda en diversos portales de encuestas realizadas por especialistas médicos en diversos países, encontrándose que la más adecuada y completa en variables de entrada se encontró en el portal web Global Dietary Data base, la encuesta elegida fue DIETA-PILOT Survey, 2012, realizada en el 2012 en el país de Rumanía, realizadas por Dunărea de Jos Universidad de Galaţi, impulsada por dos médicos de cabecera del condado Rumano, uno en la zona rural y otro en la zona urbana, y 4 de Bucarest. Los datos se encuentran almacenados en un archivo Excel, del cual se escogieron los siguientes campos:

Se analizaron las variables identificando que el campo Id_persona corresponde a un secuencial que no aporta información relevante, por otra parte, se escogió como variable de salida Categoria_bmi, retirando la segunda variable Energía para un futuro análisis.

TABLA 6. CAMPOS CONSIDERADOS EN EL ESTUDIO

No	Atributo	Descripción	Valores
0	sex	Sexo del encuestado	1 Hombre, 2 Mujer
1	age	Edad del encuestado	20 años a 88 años
2	edc	Nivel de estudios cursados o en curso.	1 Primaria 2 Sec. Básico 3 Sec. Bachillerato 4 Universidad
3	wgt	Peso en Kilogramos	40 kg a 137 kg
4	hgt	Estatura en centímetros	132 cm a 207 cm
5	eat_seq	Secuencia de la comida	Número 1 a 13
6	meal_type	Tipo de comida	1 Antes del desayuno 2 Desayuno 3 Entre el desayuno y la Almuerzo 4 Almuerzo 5 Entre la comida y la cena 6 Cena 7 Después de la cena
7	bmi_cat	Clasificación de los encuestados basada en el IMC para adultos ≥20 años	1 Bajo peso 2 Normal 3 Sobrepeso 4 Obeso

A partir de la data obtenida se procedió realizar la preparación y limpieza de registros, la muestra obtenida arrojo 1431 registros disponibles para las siguientes fases.

Mediante análisis exploratorio se procedió a escoger las variables de entrada y la variable salida para este estudio.

En esta investigación fue necesario emplear la librería pandas y numpy para el manejo de altos volúmenes de datos, de la librería sklearn.preprocessing se empleó la clase StandardScaler para escalamiento de datos, la librería sklearn.model_selection con su clase train_test_split para la división de la dataset en datos en grupo de entrenamiento y grupo de pruebas, la librería matplotlib.pyplot y la librería seaborn para ilustrar los datos de manera gráfica, la librería statsmodels.api para el análisis de correlación entre variables de entrada y de salida.

Se realizó la lectura y carga de datos en un objeto de tipo dataframe por medio de la función read_excel().

```
dataframe1.head()
```

	id	sex	age	edc	wgt	hgt	eat_seq	meal_type	bmi_cat	energy
0	32239	2	42	4	56	158	3	4	2	86.0
1	32240	2	40	4	76	170	3	6	3	299.2
2	32242	2	58	3	103	162	3	4	4	417.0
3	32242	2	58	3	103	162	3	6	4	152.5
4	32243	2	26	4	72	169	2	2	3	299.2

FIG. 4 INSPECCIÓN DE MUESTRA DE DATOS CARGADOS EN DATAFRAME.

Mediante comando info() se identificó la presencia 1431 filas disponibles para el estudio.

```
dataframe1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1431 entries, 0 to 1430
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   id          1431 non-null   int64
 1   sex         1431 non-null   int64
 2   age         1431 non-null   int64
 3   edc         1431 non-null   int64
 4   wgt         1431 non-null   int64
 5   hgt         1431 non-null   int64
 6   eat_seq     1431 non-null   int64
 7   meal_type   1431 non-null   int64
 8   bmi_cat     1431 non-null   int64
 9   energy      1431 non-null   float64
dtypes: float64(1), int64(9)
memory usage: 111.9 KB
```

FIG. 5. INSPECCIÓN DE TIPOS Y CANTIDADES DE DATOS DISPONIBLES EN EL DATAFRAME

H. Variables de entrada con Outlayers inferior y superior

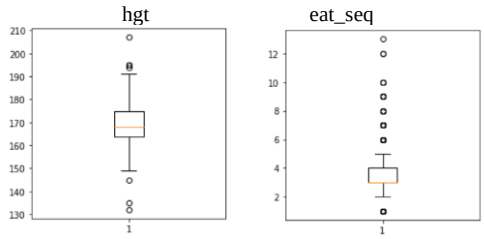


FIG. 6. VERIFICACIÓN DE PRESENCIA DE DATOS ATÍPICOS PRESENTES EN EL DATAFRAME.

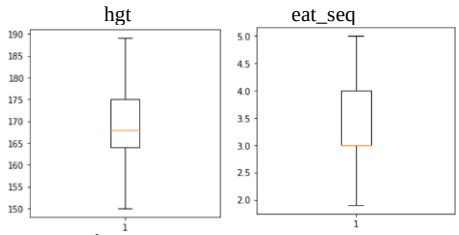


FIG. 7. VERIFICACIÓN DE DATOS CORREGIDOS PRESENTES EN EL DATAFRAME.

I. Escalamiento del conjunto de datos de entrada

```
from sklearn.preprocessing import StandardScaler
```

```
obj_sc = StandardScaler()
```

```
df_X1_esc = obj_sc.fit_transform(df_X1)
```

FIG. 8. CÓDIGO PARA ESCALAMIENTO DEL CONJUNTO DE DATOS DE ENTRADA.

```
df_X1_esc
```

	sex	age	edc	wgt	hgt	eat_seq	meal_type
0	0.938307	-1.917052	-0.428458	-1.656865	-0.530931	-1.404425	-1.341617
1	0.938307	-1.917052	-0.428458	-1.794452	0.290941	1.485532	-1.341617
2	0.938307	-1.917052	1.074294	-1.244103	-0.178700	0.553288	-1.341617
3	0.938307	-1.917052	1.074294	-1.794452	-0.296110	1.485532	-0.123436
4	0.938307	-1.917052	-0.428458	-1.656865	-0.765752	-0.378956	1.094746

FIG. 9. ESCALAMIENTO DEL CONJUNTO DE DATOS DE ENTRADA.

J. Codificación "OneHotEnconding" aplicado en datos de Salida

```
df_y1
```

	bmi_cat
18	1
439	1
313	2
1113	1
435	2
...	...
190	3
189	3
17	3
229	4
1281	2

1431 rows × 1 columns

```
df_y1_OHE
```

	0	1	2	3
0	1.0	0.0	0.0	0.0
1	1.0	0.0	0.0	0.0
2	0.0	1.0	0.0	0.0
3	1.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0
...	...	...	...	...
1426	0.0	0.0	1.0	0.0
1427	0.0	0.0	1.0	0.0
1428	0.0	0.0	1.0	0.0
1429	0.0	0.0	0.0	1.0
1430	0.0	1.0	0.0	0.0

1431 rows × 4 columns

FIG. 10. CODIFICACIÓN ONEHOTENCONDING APLICADO EN DATOS DE SALIDA.

K. División de dataset en Bloque de Entrenamiento y Pruebas

```
from sklearn.model_selection import train_test_split
```

```
# Separacion de Datos en Bloque Train y Test para Regreasion Logistica Multinomial
# --- Se realiza la separacion de datos en 80% train y 20% test
# --- Variable y es entero (indice IMC) para modelo Regresion Logistica
df_X1_train, df_X1_test, df_y1_train, df_y1_test =
      train_test_split(df_X1_esc, df_y1,test_size=0.2, random_state= 0)
```

```
# Separacion de Datos en Bloque Train y Test para Red Neuronal
# --- Se realiza la separacion de datos en 80% train y 20% test
# --- variable y se dividio en 4 columnas binarias (indice IMC) para modelo Red Neuronal
df_X2_train, df_X2_test, df_y2_train_OHE, df_y2_test_OHE =
        train_test_split(df_X1_esc, df_y1_OHE,test_size=0.2, random_state= 0)
```

FIG. 11 COMANDOS PARA DIVISIÓN DE DATASET EN BLOQUE DE ENTRENAMIENTO Y PRUEBAS.

L. Modelo de Regresión Logística Multinomial

Creacion del Modelo

```
from sklearn.linear_model import LogisticRegression
```

```
classifier=LogisticRegression(C = 1.0, class_weight = None, dual = False,
                              fit_intercept = True, intercept_scaling = 1,
                              l1_ratio =  None,  max_iter = 100, multi_class = 'multinomial',
                              n_jobs = None, penalty = 'l2', random_state = 0,  solver = 'lbfgs',
                              tol = 0.0001, verbose = 1, warm_start = False)
```

FIG. 12. COMANDOS PARA CREACION DEL MODELO.

Entrenamiento del Modelo

```
classifier.fit(df_X1_train, df_y1_train)
```

FIG. 13. COMANDOS PARA ENTRENAMIENTO DEL MODELO

Predicción del Modelo entrenado

```
# Predicción de los resultados con el Conjunto de Testing
arr_y_pred_train = classifier.predict(df_X1_train)
arr_y_pred_test  = classifier.predict(df_X1_test)
```

FIG. 14. COMANDOS PARA PREDICCIÓN DEL MODELO

Evaluación del Modelo

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm_train = confusion_matrix(df_y1_train, arr_y_pred_train)
cm_test =  confusion_matrix(df_y1_test , arr_y_pred_test)
Accuracy_test2 = accuracy_score(df_y1_test, arr_y_pred_test)
Accuracy_train2 =  accuracy_score(df_y1_train, arr_y_pred_train)
```

```
Acc_test: 97.9 , Acc_train: 96.06
```

FIG. 15. COMANDOS PARA ENTRENAMIENTO DEL MODELO

Se identifica que para el modelo de regresión Logística Multinomial con datos conocidos el índice de calidad general (accuracy de train) se ubica en el 96.06%, mientras que para datos desconocidos (accuracy de test) alcanzo el 97.9%.

Siendo un índice alto el modelo se presenta como una opción viable para emplearse en situaciones que incluyan clasificaciones de variable multiclase.

M. Modelo de Red neuronal con capas densas

La construcción del modelo de Machine Learning en general conlleva varias fases del proceso las cuales se muestran en la figura 16, donde el proceso inicia con la comprensión del problema a resolver, establecimiento de los criterios de evaluación del modelo, preparación de los datos, construir el modelo, entrenamiento, afinamiento del modelo y puesta en producción, y mantenimiento (reentrenamiento).

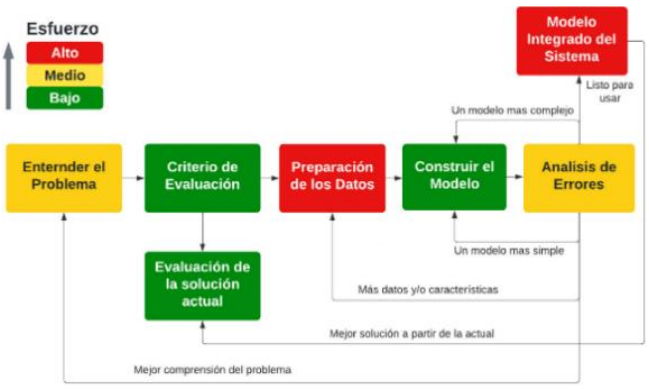


FIG. 16. PROCESO PARA CONSTRUCCIÓN DE UN MÓDULO DE MACHINE LEARNING

Creacion del Modelo de Red Neuronal Artificial (RNA)

```
import tensorflow as tf
import re

# Librerías de preprocesamiento  y técnicas NLP
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing.text import Tokenizer

# Librerias de seoparación de datos en bloques de Train y test
from sklearn.model_selection import train_test_split

# Librerias para crear el modelo
from keras.models import Model
from keras.models import Sequential

from keras.layers import Input
from keras.layers.embeddings import Embedding
from keras.layers.core import Activation, Dropout, Dense
```

FIG. 17. CÓDIGO PARA CREACION DEL MODELO DE RNA

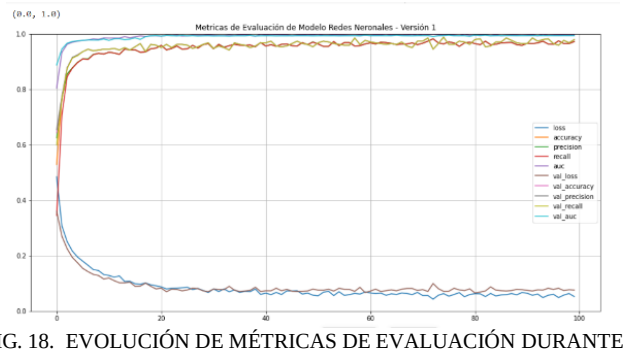


FIG. 18. EVOLUCIÓN DE MÉTRICAS DE EVALUACIÓN DURANTE E ENTRENAMIENTO.

Parámetros empleados en el modelo RN

```
from keras import models, layers

ver       = '1'
clasi     = 'NCC'
acti      ='softmax'
alg_perd  ='binary_crossentropy'
opt       ='adam'
itera     = 100
max_imput = 7
max_output = 4
red       = "In7-D.14.rl-Dp0.1-Out.Den.Sf-Loss.BinCro.OpAdam0.01"
obj_opt = tf.keras.optimizers.Adam(learning_rate=0.01)
BS = 8
Val_split = 0.3
```

FIG. 19. CÓDIGO PARA ESTABLECIMIENTO DE CREACIÓN DEL MODELO DE RNA

```
#Presentacion de la arquitectura del modelo de la red neuronal
from tensorflow import keras as keras
keras.utils.plot_model(model, show_shapes=True)
```

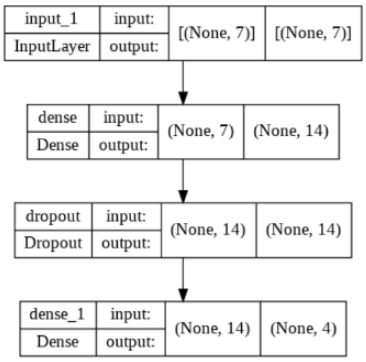


FIG. 20. ARQUITECTURA DE CAPAS Y NEURONAS EN LA RED NEURONAL CREADA.

Evaluación de Modelos

```
# Matriz de confusio con datos de test
print(f"{cm_test}")
suma1 = sum(cm_test)
suma2 = sum(suma1)
print(f"{suma1}" )
print(f"{suma2}" )
```

```
[[  1   0   0   0]
 [  0 123   1   0]
 [  0   3 120   0]
 [  0   0   2  37]]
[  1 126 123  37]
```

```
# Matriz de confusio con datos de train
print(f"{cm_train}")
suma1 = sum(cm_train)
suma2 = sum(suma1)
print(f"{suma1}" )
print(f"{suma2}" )
```

```
[[  0  20   0   0]
 [  0 481   7   0]
 [  0   3 452   3]
 [  0   0  12 166]]
[  0 504 471 169]
```

FIG. 21. MATRIZ DE CONFUSION GENERADA PARA DATOS DE TEST Y DATOS DE TRAINIG

Cálculo de métrica de General de Calidad (Acurracy) de Train y Test

```
#Calculo Accurancy score de test usando matriz de confusion
#Calculo Accurancy score de train usando matriz de confusion

Accuracy_test = (cm_test[0,0] + cm_test[1,1]+ cm_test[2,2]+cm_test[3,3])/sum(sum(cm_test))
Accuracy_train = (cm_train[0,0] + cm_train[1,1]+ cm_train[2,2]+cm_train[3,3])/sum(sum(cm_train))

# Indice de calidad general del modelo
print(f"Accuracy_test: {truncate(Accuracy_test*100,2)} , Accuracy_train: {truncate(Accuracy_train*100,2)}")
```

```
Accuracy_test: 97.9 , Accuracy_train: 96.06
```

FIG. 22. CÓDIGO PARA CÁLCULO DE MÉTRICA DE GENERAL DE CALIDAD (ACURRACY) EN DATOS DE ENTRENAMIENTO (TRAIN) Y PRUEBAS (TEST)

Ejecución de la predicción de salida empleando datos de Train y Test.

```
arr_y_pred_train = classifier.predict(df_X1_train)
arr_y_pred_test  = classifier.predict(df_X1_test)

Accuracy_test2 = accuracy_score(df_y1_test, arr_y_pred_test)
Accuracy_train2 =  accuracy_score(df_y1_train, arr_y_pred_train)
```

```
Accuracy_test: 97.9 , Accuracy_train: 96.06
```

FIG. 23. CÓDIGO PARA PREDICCIÓN DE SALIDA EN DATOS DE ENTRENAMIENTO (TRAIN) Y PRUEBAS (TEST)

Comparación visual de resultados obtenidos con el modelo de Clasificación de variable dependiente tipo multiclase empleando un clasificador basado en Red Neuronal con capa densa para los datos desconocidos (Test).

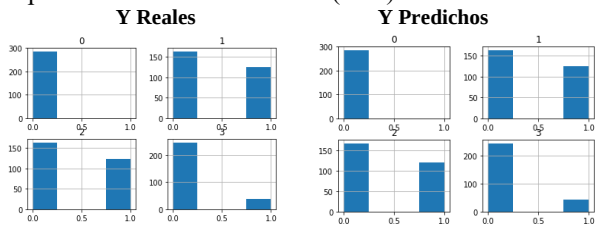


FIG. 24. COMPARACIÓN VISUAL DE DATOS REALES DE CADA CLASE VERSUS DATOS PREDICHOS EN LOS DATOS DE TEST

IV. RESULTADOS

Se identifica que el modelo de Machine Learning basado en algoritmo de clasificación de Regresión Logística Multinomial para resolver predicción de categoría de peso de personas adultas presento el nivel de eficiencia general (acurracy de train) del 96.06% para casos conocidos (data de Train) y un nivel de eficiencia general (acurracy de test) del 97.9% para casos desconocidos (data de test).

Version	Accuracy _train	Precision_train	Recall_train	AUC_train	Accuracy test	Precision test	Recall test	AUC test
1	0.980769	0.980769	0.980769	0.997319	0.989547	0.989547	0.989547	0.997495
2	0.965035	0.965035	0.965035	0.992349	0.975610	0.975610	0.975610	0.996807
3	0.971154	0.971154	0.971154	0.996989	0.982578	0.982578	0.982578	0.997293
4	0.959790	0.959790	0.959790	0.997321	0.975610	0.975610	0.975610	0.999219
5	0.979895	0.979895	0.979895	0.995960	0.975610	0.975610	0.975610	0.994703
6	0.965035	0.965004	0.964161	0.986886	0.958188	0.958188	0.958188	0.995097
7	0.917832	0.926763	0.895979	0.972254	0.885017	0.896797	0.878049	0.981611
8	0.996503	0.996503	0.996503	0.999328	0.975610	0.975610	0.975610	0.999688
9	0.968531	0.969325	0.966783	0.995177	0.961672	0.965035	0.961672	0.998480
10	0.979021	0.979021	0.979021	0.997026	0.965157	0.965157	0.965157	0.998841
11	0.979895	0.979895	0.979895	0.997840	0.968641	0.968641	0.968641	0.998934

FIG. 25. TABLA DE EVOLUCION DE MÉTRICAS DE EVALUACIÓN EN 11 CONFIGURACIONES DIFERENTES DE REDES NEURONALES PARA VARIABLE DE SALIDA TIPO MULTICLASE

En el caso del Modelo construido basado en clasificador de Redes neuronales con capas densas, se halló que luego de 11 variantes de red se pudo alcanzar la optimización del modelo, encontrándose que la conminatoria más eficiente para una red de capa densa, fue con el uso de función de activación Softmax, algoritmo de perdida Binary Crossentropy, y optimizador Adam con ratio de aprendizaje 0.01, una capa Dropout al 10% para minimizar el sobre entrenamiento, y Bach size de 8, dando como resultados de métricas los siguientes:

TABLA 7. METRICAS DE RED NEURONAL PARA VARIABLE DEPENDIENTE DE TIPO MULTICLASE

Métricas	Formula	Medición con datos conocidos (Train)	Medición con datos desconocidos	Evaluación
Accuracy - Porcentaje de predicciones correctas	a/(a+b)	98.0769%	98.9500%	✓
Recall - Porcentaje de casos positivos reales detectados correctamente	d/(d+b)	98.0769%	98.9547%	✓
Presicion - Porcentaje de predicciones positivas correctas	$\frac{(a+d)}{(a+b+c+d)}$	98.0769%	98.9547%	✓
F1-score - Es la media armónica (promedio) de la Precisión y Recall.	2*(Recall * Precision) / (Recall + Precision)	98.0769%	98.9547%	✓
AUC - Área bajo la curva ROC	Medida de lo bien que un parámetro puede distinguir entre grupos.	99.7319%	99.7495%	✓

Se verifica que las métricas básicas y adicionales se observa que los indicadores de la arquitectura 1 de redes Neuronales presento los índices más altos de entre las pruebas y afinamientos realizados.

V. CONCLUSIONES

- Con el presente trabajo se constató que al establecer los parámetros adecuados al modelo de Redes neuronales se logra alcanzar niveles de rendimiento en las métricas de calidad superiores a las alcanzadas por el Clasificador basado en Algoritmo de Regresión Logística.
- El Modelo clasificador basado en Redes Neuronal, en un inicio arrojo valores cercanos al 50% de accurracy, pero luego de ejecutarse varias pruebas experimentales variando los parámetros de la red como cantidad de capas, numero de epoch, función de error, función de activación, porcentaje de datos de entrenamiento, funciones de activación, se logró progresivamente elevar el nivel de accurracy hasta llegar al 98.95% en dados de prueba (test).
- Se sugiere continuar con estudio empleando clasificadores como Ramdon Forest, K-vecinos Más Cercanos (KNN), u otras que sean de utilidad para crear soluciones inteligentes que apoyen a la mejora en la atención a residentes de las zonas vulnerables del Ecuador u otros países, para que sean incluidos en futuros trabajos de investigación.

AGRADECIMIENTO

Se agradece a la Universidad de Guayaquil por apoyar la investigación del proyecto "Análisis y Diseño de un sistema de asesoramiento para médicos nutricionistas de la Cooperativa Bastión Popular bloque 10 a usando una red neuronal previo a emitir las prescripciones nutricionales".

REFERENCIAS

[1] El Comercio (2020)."USD 15 863 millones suman pérdidas causadas por pandemia en Ecuador - El Comercio,". Recuperado el 30 de mayo, 2020. [Online]. Available: https://www.elcomercio.com/actualidad/negocios/perdidas-economia-pandemia-ecuador-coronavirus.html. [Accessed: 18-Feb-2022].

[2] El Universo (2021). "Pandemia aumenta el sobrepeso y la obesidad en Ecuador | Informes | Noticias | El Universo,". Recuperado el 14 de febrero, 2021. [Online]. Available: https://www.eluniverso.com/noticias/informes/pandemia-aumenta-el-sobrepeso-y-la-obesidad-en-ecuador-nota/. [Accessed: 18-Feb-2022].

[3] Primicias (2019). "Solo el 50% de las familias ecuatorianas accede a una dieta nutritiva,". Recuperado el 1 de julio, 2019. [Online]. Available: https://www.primicias.ec/noticias/sociedad/ninos-desnutricion-dieta-alimentos-hambre/. [Accessed: 18-Feb-2022].

[4] Telégrafo (2020), "Expertos advierten posibles efectos nocivos en Ecuador a causa de la mala alimentación,". Recuperado el 16 de octubre, 2020. [Online]. Available: https://www.eltelegrafo.com.ec/noticias/sociedad/6/expertos-advierten-posibles-efectos-nocivos-en-ecuador-a-causa-de-la-mala-alimentacion. [Accessed: 18-Feb-2022].

[5] A. Tumbaco Bravo Joselyn Denisse, "Análisis y Diseño de un Sistema de Asesoramiento para médicos nutricionistas de la Cooperativa Bastión Popular Bloque 10 A usando una red neuronal previo a emitir las prescripciones nutricionales," Universidad de Guayaquil, 2021. Recuperado el 18 de marzo de 2022. [Online]. Available: http://repositorio.ug.edu.ec/handle/redug/56206

[6] L.Cardoso, Y. Cuervom . Andrés and J. Murcia (2015). "Porcentaje de grasa corporal y prevalencia de sobrepeso-obesidad en estudiantes universitarios de rendimiento deportivo de Bogotá, Colombia Body fat percentage and prevalence of overweight-obesity in college students of sports performance in Bogotá, Colom," Nutr. clínica y dietética, vol. 36(3), pp. 68–75, 2016, doi: 10.12873/363cardozo. Recuperado el 18 de marzo de 2022. [Online]. Available: https://revista.nutricion.org/PDF/cardozo.pdf

[7] I. El Naqa and M. J. Murphy, "What Is Machine Learning?," Mach. Learn. Radiat. Oncol., pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3_1. Recuperado el 18 de marzo de 2022. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-18305-3_1

[8] W. S. Carmona and A. Jesús Sanchez-Oliver, "Índice de masa corporal: ventajas y desventajas de su uso en la obesidad. Relación con la fuerza y la actividad física," Nutr. clínica en Med., vol. XII, no. 3, pp. 128–139, 2018, doi: 10.7400/NCM.2018.12.3.5067. Recuperado el 18 de marzo de 2022. [Online]. Available: https://www.researchgate.net/publication/329245325

[9] C. Cuadras, C. (2007). Nuevos métodos de análisis multivariante. CMC editions (Barcelona). Recuperado el 18 de marzo de 2022 [Online]. Available: http://www.majefre.udc.es/wp-content/uploads/2015/03/metodos_multivariantes.pdf

[10] J. M. Rueda, V. G. Ferrer, T. D. González, & N. C. Carvajal (2018). Regresión logística binaria para crear un modelo predictivo de daño hepático en el paciente séptico. Acta Médica del Centro, 12(1), 10-18. Recuperado el 18 de marzo de 2022. [Online]. Available: https://www.medigraphic.com/cgi-bin/new/resumen.cgi?IDARTICULO=77321

[11] C. B. Guzman, R. Wang, O. Muellerklein, M. Smith, & C. G. Eger (2022). Comparing Stormwater Quality and Watershed Typologies Across the United States: A Machine Learning Approach. Water Research, 118283. Recuperado el 18 de marzo de 2022 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0043135422002469

[12] V.A. Andrade Saltos (10-ene-2020). Comparativa entre regresión logística ordinal, redes neuronales artificiales y Gradient boosting; en la predicción de la satisfacción laboral en Ecuador. Recuperado el 18 de marzo de 2022 [Online] Available: http://dspace.espoch.edu.ec/handle/123456789/14279

[13]. C. Piñeiro, P. De Llano & M. Rodríguez (2013). ¿Proporciona la auditoría evidencias para detectar y evaluar tensiones financieras latentes? Un diagnóstico comparativo mediante técnicas econométricas e inteligencia artificial. Revista Europea de Dirección y Economía de La Empresa, 22, 115–130. Recuperado el 18 de marzo de 2022. [Online]. Available: https://doi.org/10.1016/j.redee.2012.10.001

[14] M. Sosa. (2007). Inteligencia artificial en la gestión financiera empresarial. Pensamiento y Gestión, 23, 153–186. Recuperado el 18 de marzo de 2022. [Online]. Available: https://www.redalyc.org/pdf/646/64602307.pdf

[15] A. Bosch, J. Casas-Roma y T. Lozano (2019). Deep learning : principios y fundamentos. Primera edición digital. Barcelona: Editorial UOC, 2019. Print.

[16] F. Villada, , N. Muñoz, , & E. García-Quintero. (2016). Redes Neuronales Artificiales aplicadas a la Predicción del Precio del Oro. Información tecnológica, 27(5), 143-150. Recuperado el 18 de marzo de 2022 [Online]. Available: http://dx.doi.org/10.4067/S0718-07642016000500016

[17] A. Morera, & J. Alcalá. (2018). Introducción a los modelos de redes neuronales artificiales El Perceptrón simple y multicapa (Doctoral dissertation, Tesis de pregrado, Universidad Saragoza]. Saguan Repositorio Institucional de Documentos . Recuperado el 18 de marzo de 2022. [OnLine]. Available: https://cutt. ly/lEVVf7v).

[18] S. Ghoneim.(2019). Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?. Recuperado el 18 de marzo de 2022. [Online]. Available: https://towardsdatascience.com/accuracy-recall-precision-f-score-specificity-which-to-optimize-on-867d3f11124

[19] MedCalc (2022). ROC curve analysis. Recuperado el 18 de marzo de 2022. [Online]. Available: https://www.medcalc.org/manual/roc-curves.php

[20] A. Y. Hernández & F. J. Duque (2020). Inteligencia artificial al servicio de la auditoría: una revisión sistemática de literatura. Revista Ibérica de Sistemas e Tecnologías de Información, (E27), 213-226. Recuperado el 18 de marzo de 2022. [Online]. Available:https://www.proquest.com/openview/8a2868ccf43245be9a642a31d5454ca4/1?pq-origsite=gscholar&cbl=1006393

[21] Arcata, R. A., Machaca, R. C., Montoya, L. G., & Puma, H. R. (2021). Aplicación de regresión logística para la predicción de demanda por especialidad médica en consulta externa hospitalaria. Innovación y Software, 2(2), 44-59. Recuperado el 18 de marzo de 2022. [Online]. Available: https://revistas.ulasalle.edu.pe/innosoft/article/view/45/43

[22] E. M. Rojas (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. Revista Ibérica de Sistemas e Tecnologias de Informação, (E28), 586-599. Recuperado el 18 de marzo de 2022. [Online]. Available: https://www.proquest.com/openview/c7e24c997199215aa26a39107dd2fe98/1?pq-origsite=gscholar&cbl=1006393