Supervised Learning Techniques for the Optimization of Diagnosis Processes of Diabetes in Public Health Centers

Yelena Chávez-Cujilan, Msc[1], Darwin Patiño-Pérez, Ph.D[1],Carlos García-Gutiérrez, Msc[2] , Ángel García-Gutiérrez, Msc[2], Miguel Botto-Tobar,MSc[1],Celia Munive-Mora, Lcd[3],Dalva Icaza-Rivera, MAE[4] [1]Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Física, Guayaquil, Ecuador, yelena.chavezc@ug.edu.ec, darwin.patinop@ug.edu.ec, miguel.bottot@ug.edu.ec,
[2]Universidad de Guayaquil, Guayaquil, Ecuador, carlos.garciagu@ug.edu.ec, angel.garciag@ug.edu.ec,
[3]DeSales University, Pensilvania, Estados Unidos de Norte América, cm3877@desales.edu,
[4]Universidad Estatal de Milagro, Facultad de Administración, Milagro, Ecuador, dicazar@unemi.edu.ec

Abstract. – One of the problems in public health care services in developing countries such as Ecuador, is the delay in the diagnosis of one of the main diseases that afflict the population such as diabetes, among public policies in the sector of health in this country is the optimization of diagnostic processes without using resources that affect the state budget in the short or long term. Through supervised learning techniques within the field of artificial intelligence, models can be created that allow the optimization of the diagnosis without the intervention of specialists for the interpretation of the results of patients who show signs of diabetes.

Keywords: Diabetes, Health Centers, Public Policies, Optimization, Supervised Learning.

Digital Object Identifier (DOI):
http://dx.doi.org/10.18687/LACCEI2022.1.1.779
ISBN: 978-628-95207-0-5 **ISSN:** 2414-6390

Técnicas de Aprendizaje Supervisado para la Optimización de Procesos de Diagnostico de Diabetes en los Centros de Salud Públicos

Yelena Chávez-Cujilan, Msc[1], Darwin Patiño-Pérez, Ph.D[1],Carlos García-Gutiérrez, Msc[2] ,
Ángel García-Gutiérrez, Msc[2], Miguel Botto-Tobar,MSc[1],Celia Munive-Mora, Lcd[3],Dalva Icaza-Rivera, MAE[4]
[1]Universidad de Guayaquil, Facultad de Ciencias Matemáticas y Física, Guayaquil, Ecuador, yelena.chavezc@ug.edu.ec, darwin.patinop@ug.edu.ec, miguel.bottot@ug.edu.ec,
[2]Universidad de Guayaquil, Guayaquil, Ecuador, carlos.garciagu@ug.edu.ec, angel.garciag@ug.edu.ec,
[3]DeSales University, Pensilvania, Estados Unidos de Norte América, cm3877@desales.edu,
[4]Universidad Estatal de Milagro, Facultad de Administración, Milagro, Ecuador, dicazar@unemi.edu.ec

Resumen. – Uno de los problemas en los servicios de atención de salud pública en los países en vías de desarrollo como Ecuador, es la demora en el diagnostico de una de las principales enfermedades que aquejan a la población como la diabetes, entre las políticas públicas en el sector de la salud en este país está la optimización de los procesos de diagnóstico sin que se utilicen recursos que afecten en un corto o largo plazo el presupuesto del estado. Mediante técnicas de aprendizaje supervisado dentro del campo de la inteligencia artificial, se pueden crear modelos que permitan la optimización del diagnóstico sin la intervención de especialistas para la interpretación de los resultados de los pacientes que muestran señales de diabetes.

Palabras Claves: Diabetes, Centros de Salud, Políticas Públicas, Optimización, Aprendizaje Supervisado.

Abstract. – One of the problems in public health care services in developing countries such as Ecuador, is the delay in the diagnosis of one of the main diseases that afflict the population such as diabetes, among public policies in the sector of health in this country is the optimization of diagnostic processes without using resources that affect the state budget in the short or long term. Through supervised learning techniques within the field of artificial intelligence, models can be created that allow the optimization of the diagnosis without the intervention of specialists for the interpretation of the results of patients who show signs of diabetes.

Keywords: Diabetes, Health Centers, Public Policies, Optimization, Supervised Learning.

I. INTRODUCCION

La diabetes es una condición de salud crónica de larga duración que afecta la forma en que su cuerpo convierte los alimentos en energía, la mayor parte de los alimentos que se consumen se descomponen en azúcar (también llamada glucosa) y se liberan en el torrente sanguíneo, cuando el nivel de azúcar en la sangre sube ver Fig. 1, le indica al páncreas que libere insulina; la insulina actúa como una llave para permitir que el azúcar en la sangre ingrese a las células de su cuerpo para usarla como energía[1].

Si se tiene diabetes, el cuerpo no produce suficiente insulina o no puede usar la insulina que produce tan bien como debería, cuando no hay suficiente insulina o las células dejan de responder a la insulina , demasiado azúcar en la sangre permanece en el torrente sanguíneo, con el tiempo, eso puede causar problemas de salud graves, como enfermedades cardíacas , pérdida de la visión y enfermedad renal[2].

Todavía no existe una cura para la diabetes, pero perder peso, comer alimentos saludables y mantenerse activo realmente puede ayudar, al igual de tomar los medicamentos según sea necesario, obtener educación y apoyo para el autocontrol de la diabetes y asistir a las citas de atención médica también pueden reducir el impacto de la diabetes en su vida[3].

En el Ecuador la diabetes afecta la vida de más de 1.3 millones de ecuatorianos, lo que representa más del 7.5 % de la población[4]. Cuando considera la magnitud de ese número, es fácil entender por qué todos deben estar al tanto de los signos de la enfermedad[5]. La diabetes no tratada puede causar complicaciones graves y, finalmente, convertirse en una amenaza para la vida, pero la detección temprana aumenta la probabilidad de controlar con éxito la enfermedad con un plan de tratamiento eficaz[6]. De hecho, si detecta los signos de un problema potencial en la etapa de prediabetes de la enfermedad, es posible que pueda detener la progresión antes de que desarrolle diabetes tipo-2[7]. Comience por familiarizarse con los factores de riesgo de la diabetes y los signos que debe observar que podrían indicar la aparición de la enfermedad.

Fig. 1 Diabetes

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

A. *Tipos de Diabetes*

Diabetes Tipo-1.- La diabetes tipo 1 es causada por una reacción autoinmune (el cuerpo se ataca a sí mismo por error) que impide que el cuerpo produzca insulina, aproximadamente del 5 al 10 % de las personas que tienen diabetes tienen el tipo 1, los síntomas de la diabetes tipo 1 a menudo se desarrollan rápidamente. Por lo general, se diagnostica en niños, adolescentes y adultos jóvenes, si tiene diabetes tipo 1, necesitará inyectarse insulina todos los días para sobrevivir, actualmente, nadie sabe cómo prevenir la diabetes tipo-1.

Diabetes Tipo-2.- Con la diabetes tipo-2, el cuerpo no usa bien la insulina y no puede mantener el azúcar en la sangre en niveles normales, alrededor del 90-95 % de las personas con diabetes tienen el tipo-2, se desarrolla a lo largo de muchos años y generalmente se diagnostica en adultos (pero cada vez más en niños, adolescentes y adultos jóvenes). Es posible que no se note ningún síntoma, por lo que es importante hacerse una prueba de azúcar en la sangre si se está en riesgo. La diabetes tipo-2 se puede prevenir o retrasar con cambios saludables en el estilo de vida, como perder peso, comer alimentos saludables y mantenerse activo.

Diabetes Gestacional. - La diabetes gestacional se desarrolla en mujeres embarazadas que nunca han tenido diabetes, si se tiene diabetes gestacional, el bebé podría correr un mayor riesgo de tener problemas de salud; la diabetes gestacional generalmente desaparece después de que nace el bebé, pero aumenta su riesgo de diabetes tipo-2 más adelante en la vida; es más probable que un bebé tenga obesidad cuando sea niño o adolescente, y también es más probable que desarrolle diabetes tipo-2 más adelante en su vida.

La Prediabetes. – En Ecuador, más de 1 de cada 3 adultos, tienen prediabetes, es más, más de 8 de cada 10 de ellos no saben que lo tienen; con la prediabetes, los niveles de azúcar en la sangre son más altos de lo normal, pero aún no lo suficientemente altos como para diagnosticar diabetes tipo-2. La prediabetes aumenta el riesgo de diabetes tipo-2, enfermedad cardíaca y accidente cerebrovascular.

B. *Optimización*

La optimización de procesos es la mejora que se aplica a los procesos a través de un mejor uso de los recursos para aumentar la eficiencia. Como resultados de la optimización de procesos se pueden alcanzar una mejora en los flujos de trabajo, mejoraras en la comunicación, predecir cambios y eliminar redundancias. En términos generales, la gestión y optimización de procesos es fundamental para la transformación digital en todas las empresas.

Con la optimización se puede mitigar el riesgo mediante el mapeo de actividades que facilitan la estandarización de procesos y su formalización, reduciendo los errores, repeticiones y dudas en los procedimientos, lo que reduce notablemente los riesgos.

Por otra parte, también se logra reducir los costos y se pueden reducir gastos, lográndose identificar fácilmente los desperdicios, lo que permite encontrar errores, recursos mal utilizados, cuellos de botella que comprometen la productividad. Es más fácil cumplir con los procesos estandarizados y monitoreados. Además, en caso de una auditoría, la transparencia en los procesos facilita los trámites y contribuye a los resultados deseados.

Uno de los objetivos de los sistemas de administración pública es la optimización de los procesos que contribuyan a agilizar los servicios de atención a los contribuyentes, así como evitar el desperdicio de recursos por parte de los organismos del estado. Los usuarios de los sistemas de salud públicos esperan que todo tipo de proceso de atención y diagnostico entre otros funcionen de forma rápida y eficiente, que además los diagnósticos y resultados de exámenes consuman el menor de los tiempos.

C. *Aprendizaje Supervisado*

En general, los sistemas de IA funcionan incorporando grandes cantidades de datos de entrenamiento etiquetados[8], analizando los datos en busca de correlaciones y patrones, y utilizando estos patrones para hacer predicciones sobre estados futuros[9]. De esta manera, un programa que recibe un dataset con características de síntomas de una enfermedad puede aprender a diagnosticar la enfermedad de forma real con aquellos síntomas de personas, incluso puede aprender a identificar y describir la enfermedad revisando millones de registros con características de una determinada enfermedad[10].

La programación de IA se centra en tres habilidades cognitivas como lo son el aprendizaje, el razonamiento y la autocorrección[11]. El proceso de aprendizaje es un aspecto de la programación de IA que se enfoca en adquirir los datos y crear las reglas sobre cómo convertir los datos en información procesable[12]. Las reglas, que se denominan algoritmos, proporcionan a los dispositivos informáticos instrucciones paso a paso sobre cómo completar una tarea específica[13].

Sin embargo, los procesos de razonamiento son un aspecto de la programación de IA que se centra en elegir el algoritmo correcto para alcanzar el resultado deseado. Existen técnicas basadas en refuerzo, impulso o simplemente *Boosting* [14] las cuales son consideradas como un conjunto de algoritmos cuya función principal es convertir a los clasificadores débiles en clasificadores fuertes (*weak learners to strong learners*), los cuales se han convertido en la corriente principal de la industria de la ciencia de datos porque han estado presentes en la comunidad de aprendizaje automático durante años. Boosting fue introducido por primera vez por Freund y Schapire en el año 1997 con su algoritmo AdaBoost y, desde entonces, Boosting ha sido una técnica predominante para resolver problemas de clasificación binaria y cuyos resultados se contrastarán con el uso de una red neuronal artificial como técnica de aprendizaje profundo[15].

II. Materiales y Metodos

A. *Materiales*

Se ha tomado un dataset que contiene 768 registros de pacientes cuyas características son: **pregnant_times** en el que se almacena el número de veces que la persona ha estado embarazada en el caso de que el paciente es mujer, **glucose** columna que almacena la concentración de glucosa en plasma a las 2 horas en una tolerancia oral a la glucosa, **blood_pressure** columna con el valor de la presión arterial diastólica (mm Hg), **tst** es el grosor del pliegue cutáneo del tríceps (mm), **insulin** característica asociada con la insulina sérica de 2 horas (mu U/ml), **BMI** es el índice de masa corporal (peso en kg/(altura en m)^2) , **dpf** es la función de pedigrí de diabetes, **age** es la edad en años, **is_diabetic** indicador binario de la presencia de diabetes.
Para la implementación de las técnicas de aprendizaje de maquina supervisado se utilizará Python[16] como herramienta de programación que se ejecutará sobre una máquina virtual en Google Colab[17] por estar configurada con todas las librerías requeridas para el uso de machine learning y deep learning, además se necesitará una conexión a internet con suficiente ancho de banda.

B. *Métodos*

Entre los métodos de aprendizaje supervisado[18] para realizar predicciones mediante clasificación, existen referencias que ubican a las técnicas basadas en *Boosting* como las más eficientes para realizar predicciones, por lo que se crearan modelos con Python basadas en: *Gradient Boosting, Xtreme Gradient Boosting, Adaptive Boosting, Cat Gradient Boosting*, *Light Gradient Boosting*. Además se creara un modelo basado en aprendizaje profundo o *Deep Learning* mediante una Red Neuronal Artificial[19] para predecir mediante clasificación.

Los algoritmos basados en *Boosting* utilizan un enfoque de aprendizaje basado en la idea de crear una regla de predicción altamente precisa mediante la combinación de muchas reglas relativamente débiles e imprecisas. Es una técnica secuencial que funciona según el principio de conjuntos[28]. Combina un conjunto de aprendizaje débil y ofrece una precisión de predicción mejorada. En cualquier instante t, los resultados del modelo se ponderan en función de los resultados del instante anterior t-1. Los resultados pronosticados correctamente reciben una ponderación más baja y los que no se clasifican correctamente reciben una ponderación más alta, es de señalar que un aprendizaje débil es ligeramente mejor que adivinar al azar [29].

Gradient Boosting o *Gradient Tree Boosting* es un algoritmo en el que se entrenan varios modelos secuencialmente y, para cada nuevo modelo, el modelo minimiza gradualmente la función de pérdida mediante el método de descenso de gradiente. El algoritmo toma los árboles de decisión como los más débiles porque los nodos en un árbol de decisión consideran una rama diferente de características para seleccionar la mejor división, lo que significa que todos los árboles no son iguales.

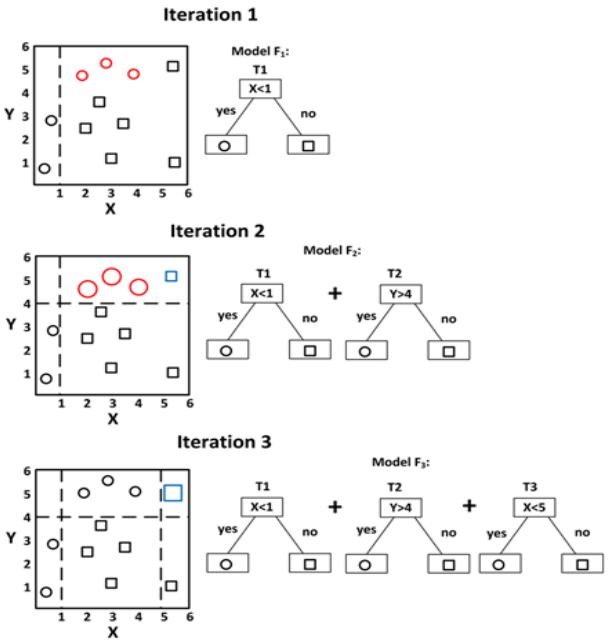

Fig. 2 *Gradient Boosting*

Por lo tanto, pueden capturar diferentes salidas de los datos todo el tiempo. Este algoritmo se construye secuencialmente porque, para cada nuevo árbol, el modelo considera los errores del último árbol y la decisión de cada árbol sucesivo se basa en los errores cometidos por el árbol anterior.

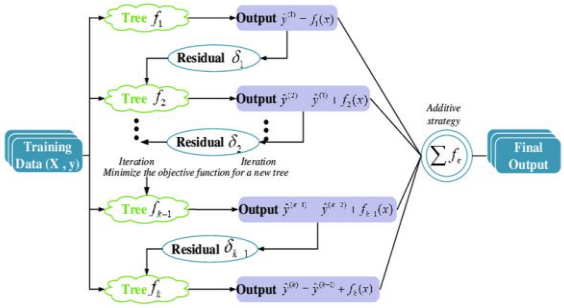

Fig. 3 *Xtreme Gradient Boosting*

Extreme Gradient Boosting o XGBoost, es un algoritmo considerado como una versión mejorada del algoritmo de aumento de gradiente[20] o *Gradient Boosting*, el procedimiento de trabajo de ambos es casi el mismo. Un punto crucial en XGBoost es que implementa el procesamiento paralelo a nivel de nodo, lo que lo hace más potente y rápido que el algoritmo del aumento de gradiente. XGBoost reduce el sobreajuste y mejora el rendimiento general al incluir varias técnicas de regularización y al configurar sus hiperparámetros[21]. Una fortaleza de este algoritmo es que no necesita preocuparse por los valores que faltan en el conjunto de datos, ya que, en el entrenamiento, el propio modelo aprende dónde encajar los valores que faltan, es decir, el nodo izquierdo o el nodo correcto.

Adaptive Boosting o AdaBoost, es un algoritmo que emplea una técnica de impulso en el aprendizaje automático que se utiliza como un método de conjuntos[22]. En el AdaBoost, todos los pesos se re-asignan a cada instancia en la que se otorgan pesos más altos a los modelos clasificados incorrectamente, y se ajusta a la secuencia de aprendiz débil o *weak learners* en diferentes pesos[23]. El Adaboost comienza haciendo predicciones sobre el conjunto de datos original en un lenguaje sencillo, y luego otorga el mismo peso a cada observación. Si la predicción realizada con el primer aprendiz es incorrecta, asigna mayor importancia a la declaración predicha incorrectamente y a un proceso iterativo[24]. Continúa agregando nuevos aprendices hasta que se alcanza el límite en el modelo.

CatBoost es un algoritmo de aprendizaje automático de código abierto recientemente de Yandex. Puede integrarse fácilmente con marcos de aprendizaje profundo como TensorFlow de Google y Core ML de Apple. Puede funcionar con diversos tipos de datos para ayudar a resolver una amplia gama de problemas a los que se enfrentan las empresas en la actualidad[25]. Para complementarlo, proporciona la mejor precisión de su clase. Es especialmente poderoso de dos maneras: Produce resultados de última generación sin la extensa capacitación de datos que normalmente requieren otros métodos de aprendizaje automático, y Proporciona un potente soporte listo para usar para los formatos de datos más descriptivos que acompañan a muchos problemas comerciales. El nombre "*CatBoost*" proviene de dos palabras "*Category*" y "*Boosting*". "*Boost"* proviene del algoritmo de aprendizaje automático de aumento de gradiente, ya que esta biblioteca se basa en la biblioteca de aumento de gradiente[26]. El aumento de gradiente es un poderoso algoritmo de aprendizaje automático que se aplica ampliamente a múltiples tipos de desafíos comerciales, como la detección de fraudes, elementos de recomendación, pronósticos y también funciona bien. También puede arrojar muy buenos resultados con relativamente menos datos, a diferencia de los modelos *Deep Learning* que necesitan aprender de una gran cantidad de datos.

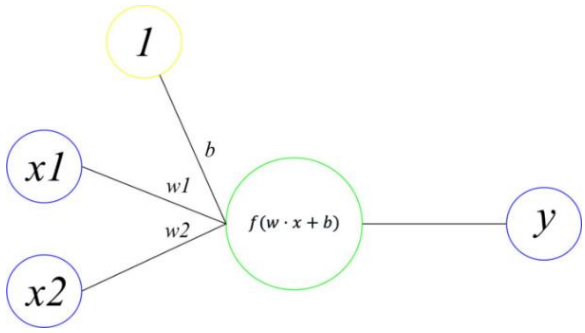

Fig. 4 Neurona Artificial

Un árbol de decisión de aumento de gradiente o GBDT es un algoritmo de aprendizaje automático muy popular que tiene implementaciones efectivas como XGBoost y muchas técnicas de optimización se adoptan a partir de este algoritmo. La eficiencia y la escalabilidad del modelo no están a la altura cuando hay más características en los datos[27].

Para este comportamiento específico, la razón principal es que cada función debe escanear todas las distintas instancias de datos para hacer una estimación de todos los posibles puntos de división, lo que requiere mucho tiempo y es tedioso. Para resolver este problema, se utiliza el modelo LGBM o Light Gradient Boosting Model[28] que utiliza dos tipos de técnicas que se basan en gradientes en el muestreo lateral o GOSS y en la agrupación de características exclusivas o EFB. Por lo tanto, GOSS en realidad excluirá la porción significativa de la parte de datos que tiene pequeños gradientes y solo usará los datos restantes para estimar la ganancia de información general.

El propósito de una red neuronal artificial es imitar cómo funciona el cerebro humano con la esperanza de que podamos construir una máquina que se comporte como un ser humano[29]. Una neurona artificial es el bloque de construcción central de una red neuronal artificial. La estructura de una neurona artificial es muy similar a una neurona biológica[30], consta de 3 partes principales, peso y sesgo como una dendrita denotada por w y b respectivamente, salida como un axón denotado por y, y función de activación como un cuerpo celular (núcleo) denotado por f(x). La x son las señales de entrada recibidas por la dendrita. En las neuronas artificiales, la entrada y el peso se representan como un vector, mientras que el sesgo se representa como un escalar. La neurona artificial [31] procesa las señales de entrada realizando un producto punto entre el vector de entrada y el vector de peso, agrega el sesgo, luego aplica una función de activación y finalmente propaga el resultado a otras neuronas.

La función de activación[31], es una función utilizada por una neurona artificial para obtener su salida, también se conoce como función de transferencia. El resultado del producto punto entre el peso y la entrada más el sesgo está en el rango de -inf y +inf, la función de activación tiene como objetivo asignar el resultado a un cierto rango dependiendo de la función[32].

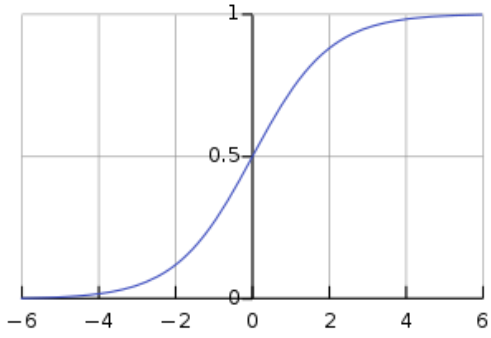

Fig. 5 Función Sigmoide

Existen muchas funciones de activación, pero la más importante es la función de activación sigmoide[33]. A menudo se usa como activación en la capa de salida para tareas de clasificación binaria. Sigmoid acota el resultado en el rango de 0 hasta 1, representa la probabilidad de x si x pertenece a la clase 1 o 0. Sigmoid toma una decisión mediante el umbral del resultado, si el resultado es ≥ 0.5 entonces x se clasifica como 1 de lo contrario, x se clasifica como 0.

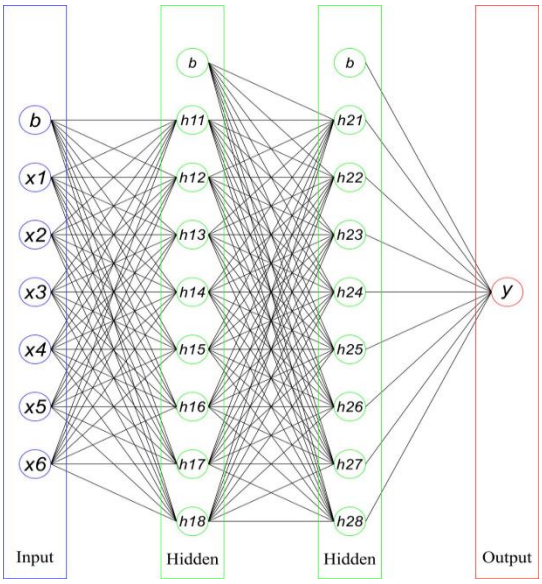

Fig. 6 Red Neuronal Artificial

La Red Neuronal Artificial o ANN, es un modelo computacional que imita la forma en que funcionan las células nerviosas en el cerebro humano. Las redes neuronales artificiales (ANN) utilizan algoritmos de aprendizaje que pueden hacer ajustes de forma independiente, o aprender, en cierto sentido, a medida que reciben nuevos datos. Esto las convierte en una herramienta muy eficaz para el modelado de datos estadísticos no lineales. Las ANN de aprendizaje profundo juegan un papel importante en el aprendizaje automático (ML) y son compatibles con el campo más amplio de la tecnología de inteligencia artificial (IA).

Una red neuronal artificial tiene tres o más capas que están interconectadas, la primera capa consta de neuronas de entrada las mismas que envían datos a las capas más profundas, que a su vez enviarán los datos de salida finales a la última capa de salida; todas las capas internas están ocultas y están formadas por unidades que cambian adaptativamente la información recibida de una capa a otra a través de una serie de transformaciones. Cada capa actúa como una capa de entrada y salida que permite a la ANN comprender objetos más complejos, en conjunto, estas capas internas se denominan capa neural y las unidades en la capa neuronal intentan aprender sobre la información recopilada al pesarla de acuerdo con el sistema interno de ANN. Estas pautas permiten que las unidades generen un resultado transformado, que luego se proporciona como salida a la siguiente capa.

C. Métricas de Evaluación para Clasificación.

TABLA I
MATRIZ DE CONFUSION

		Predicción	
		0	1
Real	0	TN	FP
	1	FN	TP

Matriz de Confusión.- También conocida como matriz de error, aunque no es una métrica se la toma como referencia para determinar si el modelo ha clasificado apropiadamente. Cuando el modelo clasifica adecuadamente se tienen dos valores. Verdaderos Positivos, cuando el modelo ha predicho que SI y en realidad SI. Verdaderos Negativos, aquí el modelo ha predicho que NO y en realidad es un NO. Cuando no ha clasificado adecuadamente se tienen los siguientes valores. Falso Positivo, cuando el modelo ha predicho que SI y en realidad es un NO. Falso Negativo, es cuando el modelo ha predicho que NO pero en realidad es SI.

1)Accuracy.- La exactitud es la cantidad de predicciones positivas que fueron correctas.

$$\text{Accuracy} = \frac{(TN+TP)}{(FP+TP)+(TN+FN)} \quad (1)$$

2)Precision.- La precisión es el porcentaje de casos positivos detectados.

$$\text{Presicion} = \frac{TP}{(TP+FP)} \quad (2)$$

3)Recall .- La sensibilidad es una métrica muy utilizada en el campo de la medicina y es la capacidad de poder detectar correctamente en el caso de la salud, la enfermedad entre los enfermos.

$$\text{Recall} = \frac{TP}{(TP+FN)} \quad (3)$$

4)Specificity.- La especificidad es una métrica que se usa en el campo de la medicina, y es la capacidad de poder identificar los casos de pacientes sanos entre todos los sanos.

$$\text{Specificity} = \frac{TN}{(TN+FP)} \quad (4)$$

5) La curva ROC o curva Característica Operativa del Receptor[34] es el gráfico, que expone el rendimiento de un clasificador binario en función del umbral de corte[35], exponiendo la tasa de verdaderos positivos (TPR) contra la tasa de falsos positivos (FPR) [36] umbral.

III. METODOLOGIA

La metodología utilizada está basada en técnicas de aprendizaje automático o *machine learning* al igual que una técnica de aprendizaje profundo o *deep learning* por medio de una red neuronal artificial, los pasos a seguirse se basaron en:

1)Tratamiento de los Datos
2)Creación del Modelo
3)Fase de entrenamiento
4)Fase de Prueba
5)Evaluación del modelo con sus métricas
6)Aplicación del modelo

Fase-I
Para el tratamiento de los datos, se ha tomado un dataset que proviene de Kaggle y que además se encuentra en los principales portales públicos para el estudio de la diabetes.

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   pregnant_times  768 non-null    int64
 1   glucose         768 non-null    int64
 2   blood_pressure  768 non-null    int64
 3   tst             768 non-null    int64
 4   insulin         768 non-null    int64
 5   bmi             768 non-null    float64
 6   dpf             768 non-null    float64
 7   age             768 non-null    int64
 8   is_diabetic     768 non-null    int64
```

Fig.7 Dataset Diabetes

El dataset está conformado por 768 registros con características de pacientes que tienen y no tienen diabetes, hay 500 registros de pacientes sin diabetes y 268 registros con diabetes.

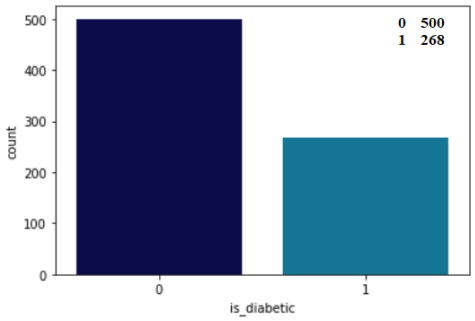

Fig.8 Histogramas de Pacientes

Del total de registros se designó el 10% para pruebas por lo que X_test tiene 77 registros, el 90% restante se designaron para entrenamiento por lo que X_train tiene 691 registros.

```
[ ]  print(X_train.shape,X_test.shape)

(691, 8) (77, 8)
```

Fig.9 Registros para Train y Test

Fase II

En esta fase se exponen las etapas que van desde la creación del modelo, entrenamiento, prueba, evaluación y aplicación de estos.

```
[36] #AdaBoosting
     from sklearn.ensemble import AdaBoostClassifier

     abc = AdaBoostClassifier(n_estimators=50,learning_rate=1)
     model = abc.fit(X_train, y_train)
     y_pred = model.predict(X_test)

     # Model Accuracy, con que frecuencia es correcto el clasificador?
     accuracy_ada = metrics.accuracy_score(y_test, y_pred)
     print("Accuracy AdaBoost: ",accuracy_ada)
     acc.append(accuracy_ada)

     Accuracy AdaBoost:  0.7532467532467533
```

Fig.10 Modelos Basados en *Boosting*

Al algoritmo expuesto en la (Fig. 10) se les ha realizado una serie de mejoras que se reflejan en (Fig. 11)

```
[31] #GradientBoosting
     from sklearn.ensemble import GradientBoostingClassifier

     modelGB = GradientBoostingClassifier(random_state=100,
                                          n_estimators=150,min_samples_split=100, max_depth=6)
     modelGB.fit(X_train, y_train)
     gbk_predict = modelGB.predict(X_test)
     gbk_score = modelGB.score(X_test,y_test)
     gbos_score=metrics.accuracy_score(y_test, gbk_predict)
     print("Gradient Boosting Score :",gbos_score)
     acc.append(gbos_score)

     Gradient Boosting Score : 0.7012987012987013
```

```
[32] #XGBoosting
     import xgboost as xgb
     from sklearn.linear_model import LinearRegression as LR

     modelXGB = xgb.XGBClassifier(objective='binary:logistic', n_estimators=10,
                                  seed=123)
     modelXGB.fit(X_train, y_train)
     preds_xgb = modelXGB.predict(X_test)

     accuracy_xgb = float(np.sum(preds_xgb == y_test))/y_test.shape[0]
     print('Accuracy de XGBoost: ', accuracy_xgb)
     acc.append(accuracy_xgb)

     Accuracy de XGBoost:  0.7662337662337663
```

```
[40] import lightgbm as lgb

     model = lgb.LGBMClassifier(learning_rate=0.09,max_depth=-5,random_state=42)
     model.fit(X_train,y_train,eval_set=[(X_test,y_test),(X_train,y_train)],
               verbose=0,eval_metric='logloss')
     # Get predicted classes
     y_pred = model.predict(X_test)

     # Model Accuracy, how often is the classifier correct?
     accuracy_lig =  metrics.accuracy_score(y_test, y_pred)
     print("Accuracy LGBM: ",accuracy_lig)
     acc.append(accuracy_lig)
     #print('Training accuracy {:.4f}'.format(model.score(X_train,y_train)))
     #print('Testing accuracy {:.4f}'.format(model.score(X_test,y_test)))

     Accuracy LGBM:  0.7142857142857143
```

```
[56] #CatBoosting
     from catboost import CatBoostClassifier
     from sklearn import metrics

     cat_features = [0, 1]
     # Initialize CatBoostClassifier
     model = CatBoostClassifier(iterations=2,learning_rate=1,depth=2)
     # Fit model
     model.fit(X_train, y_train, cat_features)
     # Get predicted classes
     y_pred = model.predict(X_test)
     # Get predicted probabilities for each class
     preds_proba = model.predict_proba(X_test)
     # Get predicted RawFormulaVal
     preds_raw = model.predict(X_test, prediction_type='RawFormulaVal')

     # Model Accuracy, how often is the classifier correct?
     accuracy_cat =  metrics.accuracy_score(y_test, y_pred)
     print("Accuracy CatBoost: ",accuracy_cat)
     acc.append(accuracy_cat)

     0:      learn: 0.5911708        total: 676us    remaining: 676us
     1:      learn: 0.5428066        total: 1.9ms    remaining: 0us
     Accuracy CatBoost:  0.7272727272727273
```

Fig.11 Mejoras de los Modelos Basados en *Boosting*

El modelo de red neuronal artificial es creado con Keras, tiene una capa de entrada con 8 neuronas y dos capas ocultas más una capa de salida, según (Fig. 13).

```
#Creación del Modelo - ANN
model = Sequential()
model.add(Dense(12,input_dim=8,kernel_initializer='uniform',activation='relu'))
model.add(Dense(8,kernel_initializer='uniform',activation='relu'))
model.add(Dense(1,kernel_initializer='uniform',activation='sigmoid'))

#Compilacion del Modelo
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

#Entrenamiento del Modelo
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=500, verbose=0)

#Evalucion del Modelo
scores = model.evaluate(X_train, y_train)
print("%s: %.2f%%" % (model.metrics_names[1], scores[1] * 100))
temp = scores[1]

22/22 [==============================] - 0s 2ms/step - loss: 0.3860 - accuracy: 0.8220
accuracy: 82.20%
```

Fig.12 Modelo de ANN

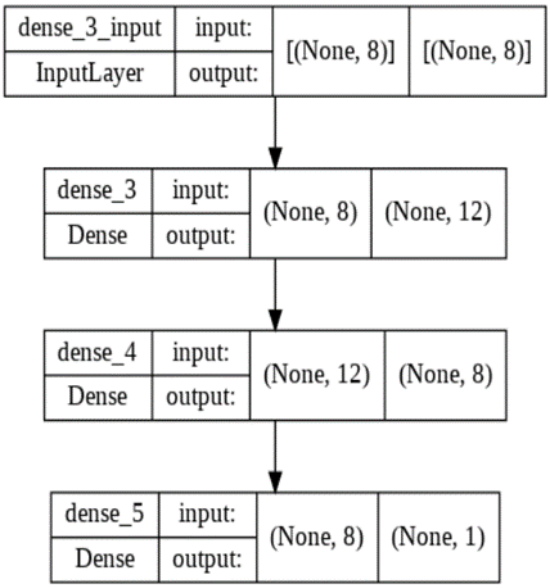

Fig.13 Capas de la ANN

La creación de la matriz de confusión fue muy importante para poder probar las métricas de clasificación, la matriz resultante de las pruebas desarrolladas por el modelo según la (Fig.14).

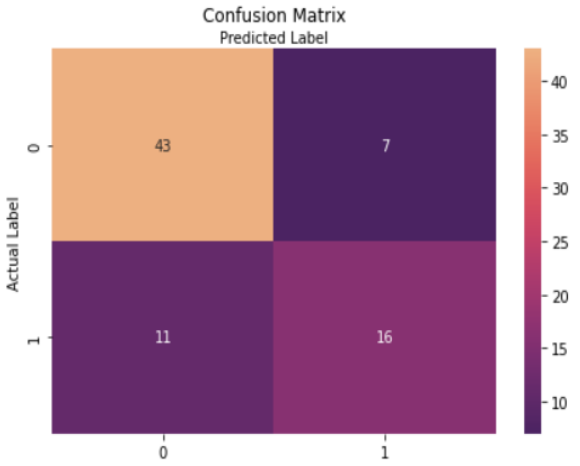

Fig.14 Matriz de Confusión - XGBoost

La matriz de confusión también conocida como matriz de error o tabla de contingencia y cuya descripción se refleja en la Tabla I también sirve para determinar las métricas expuestas desde (5) hasta (11). De los 77 registros de tomados para test, la clasificación obtenida por el XGBoost es la mejor, el modelo clasificó correctamente 43 y 16 registros.

La reducción del tiempo de diagnóstico de diabetes está asociada con la rápida observación del resultado del examen por parte de un especialista, situación que ahora puede reemplazarse con el uso de un modelo de que ha aprendido a detectar si el resultado determina si el paciente es diabético o no diabetico.

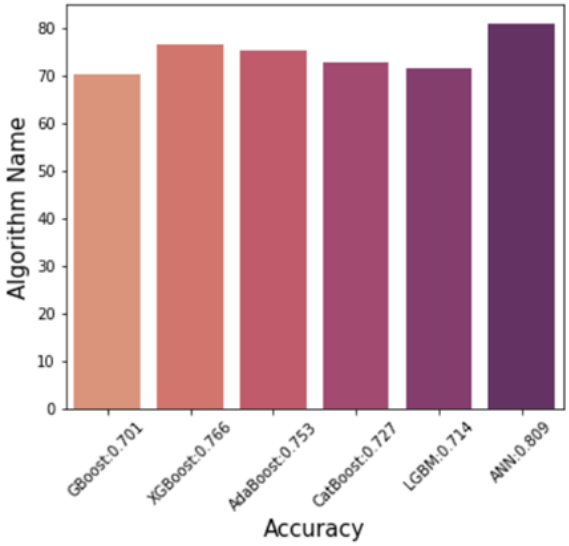

Fig.15 Curva de Sensibilidad

En Fig. 15 se refleja el nivel de exactitud que se ha alcanzado con los algoritmos basados en *Boosting*, entre los que sobresale *XGBoost* que refleja una exactitud del 76.6% frente a los otros modelos a excepción del modelo de ANN que refleja una exactitud aproximada del 81%. Aunque las técnicas de Boosting ofrecen resultados óptimos de predicción, esto no dejan de ser técnicas que usan a los árboles de decisión.

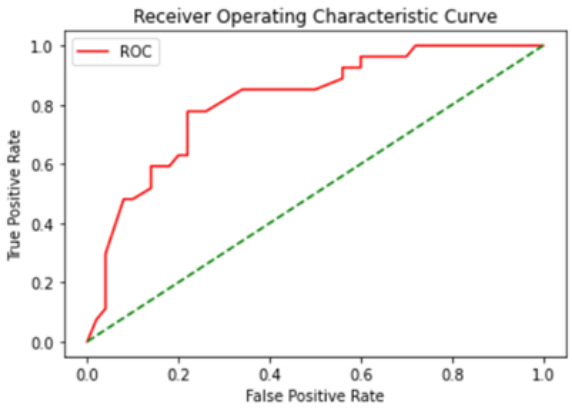

Fig.16 Curva ROC

El estudio de curva ROC según la (Fig. 16), representa una técnica estadística para definir la exactitud diagnosticada.

La función auc() selecciona como entradas la sensibilidad y la precisión regresando el valor del área bajo la curva, el cual puede ser tomado como síntesis del desempeño del modelo. Para conseguir el valor de AUC, se emplea la función roc_auc_score() (dato de entrada conocido previamente). En esta ocasión retorna la estimación de AUC, contenida entre 0.5 (clasificador al azar) y 1.0 (clasificador óptimo).

Por otra parte, al haberse ejecutado el modelo de red neuronal artificial se ha alcanzado un acurracy muy provechoso para el diagnóstico de diabetes. La pérdida del modelo a la hora de predecir es aceptable ya que alcanza valores mínimos, corroborando el nivel de exactitud del 81% aprox.

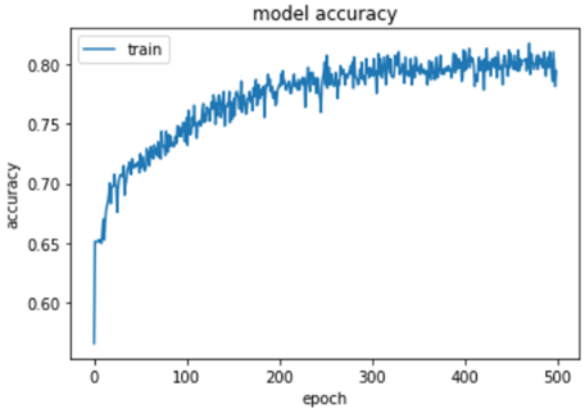

Fig.17 Curva de exactitud del modelo de ANN

Para el desarrollo de los modelos se seleccionó la técnica de aprendizaje supervisado, basadas en *Boosting* y todas aquellas variantes que alcanzaron un nivel de predicción aceptable, sin que superen al XGBoosting que alcanza el 76.6% de *accuracy* aproximadamente.

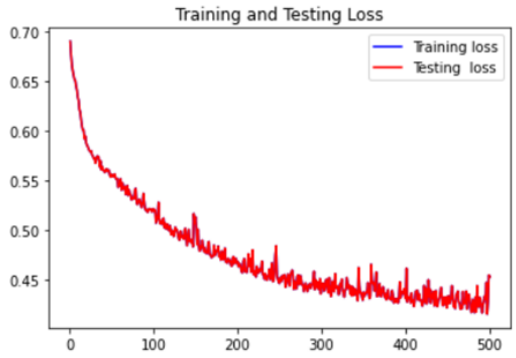

Fig.18 Curva de Perdida

Se puede concluir que los modelos de la Fig. 15 han aprendido a realizar adecuadamente la clasificación, en el caso del XGBoosting se obtuvo como resultado una exactitud de predicción del 76.6% frente al modelo basado en ANN con el cual se alcanzó el 81% de exactitud con lo que se determina que los modelos basados en redes neuronales artificiales alcanzan un nivel de precisión mucho mayor que las técnicas basadas en arboles e impulso.

Con el desarrollo de este predictor, se puede ayudar al sector de la salud, para diagnosticar la presencia o no de la diabetes. Por otra parte, los organismos del sistema de salud podrían brindar un mejor servicio ajustado a las políticas públicas agilizando el proceso de diagnóstico, y con un presupuesto relativamente bajo que cubra la compra de un computador en caso de que no exista uno, con acceso a internet y con un compilador de Python de licencia open source.

Referencias

[1] World Health Organization, "The Global Diabetes Compact: what you need to know," *Oms*, 2021.

[2] J. R. Wright, "What Was Known About Childhood Diabetes Mellitus Before the Discovery of Insulin?," *Pediatr. Dev. Pathol.*, 2021.

[3] A. C. Mühlbacher, A. Sadler, and C. Juhnke, "Personalized diabetes management: what do patients with diabetes mellitus prefer? A discrete choice experiment," *Eur. J. Heal. Econ.*, vol. 22, no. 3, 2021.

[4] INEC, "Diabetes, segunda causa de muerte después de las enfermedades isquémicas del corazón |," 2022. [Online]. Available: https://www.ecuadorencifras.gob.ec/diabetes-segunda-causa-de-muerte-despues-de-las-enfermedades-isquemicas-del-corazon/. [Accessed: 15-Mar-2022].

[5] J. Pérez-Galarza *et al.*, "Prevalence of overweight and metabolic syndrome, and associated sociodemographic factors among adult Ecuadorian populations: the ENSANUT-ECU study," *J. Endocrinol. Invest.*, vol. 44, no. 1, 2021.

[6] I. Tapager, K. R. Olsen, and K. Vrangbæk, "Exploring equity in accessing diabetes management treatment: A healthcare gap analysis," *Soc. Sci. Med.*, vol. 292, 2022.

[7] Y. K. Kong, K. S. Song, M. E. Jung, M. Kang, H. J. Kim, and M. J. Kim, "Discovery of GCC5694A: A potent and selective sodium glucose co-transporter 2 inhibitor for the treatment of type 2 diabetes," *Bioorganic Med. Chem. Lett.*, vol. 56, 2022.

[8] A. Chapman *et al.*, "Dataset search: a survey," in *VLDB Journal*, 2020, vol. 29, no. 1.

[9] D. Sarkar, R. Bali, and T. Sharma, *Practical Machine Learning with Python*. 2018.

[10] D. P. Pérez, R. S. Bustillos, C. M. Mora, and M. Botto-Tobar, "Prediction of Covid19 with the use of Random Forests Algorithm and Artificial Neural Networks," *Ecuadorian Sci. J.*, vol. 4, no. 2, pp. 101–110, Sep. 2020.

[11] J. Cabanelas Omil, "Inteligencia artificial ¿Dr. Jekyll o Mr. Hyde?," *Mercados y Negocios*, no. 40, pp. 5–22, 2019.

[12] J. Luna, "Tipos de aprendizaje automático," *Medium*, 2018.

[13] J. I. Bagnato, "Algoritmo k-Nearest Neighbor | Aprende Machine Learning," *July, 10th*, 2018. .

[14] InteractiveChaos, "Gradient Boosting," *2021*, 2021. .

[15] Y. Ren, "Python Machine Learning : Machine Learning and Deep Learning With Python ," *Int. J. Knowledge-Based Organ.*, vol. 11, no. 1, 2021.

[16] P. Mathur, *Machine Learning Applications Using Python*. 2019.

[17] Fuat, "Google Colab Free GPU Tutorial," *DEEP LEARNING TURKEY*, 2018. .

[18] J. Brownlee, "Supervised and Unsupervised Machine Learning Algorithms," *Understand Mach. Learn. Algorithms*, 2016.

[19] A. M. Giancarlo Zaccone, Md. Rezaul Karim *et al.*, "Deep Learning With Python Develop Deep Learning Models On Theano And TensorFlow Using Keras," *Mach. Learn.*

Using R, vol. 26, no. 3, 2019.
[20] J. J. Espinosa Zúñiga, "Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito," *Ing. Investig. y Tecnol.*, vol. 21, no. 3, 2020.
[21] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-August-2016, pp. 785–794, Mar. 2016.
[22] H. Li, M. Zeng, M. Lu, X. Hu, and Z. Li, "Adaboosting-based dynamic weighted combination of software reliability growth models," *Qual. Reliab. Eng. Int.*, vol. 28, no. 1, 2012.
[23] H. Asil and J. Bagherzadeh, "A new approachto image classification based on adeep multiclass AdaBoosting ensemble," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 5, 2020.
[24] H. Shao, H. Wei, X. Deng, and S. Xing, "Short-term wind speed forecasting using wavelet transformation and AdaBoosting neural networks in Yunnan wind farm," *IET Renew. Power Gener.*, vol. 11, no. 4, 2017.
[25] J. Ding, Z. Chen, L. Xiaolong, and B. Lai, "Sales Forecasting Based on CatBoost," in *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, 2020.
[26] S. Sakila, S. Garg, T. Yeole, and H. Yadav, "Earthquake time prediction using catboost and SVR," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 1, 2019.
[27] K. Koshelev and Y. Kivshar, "Light trapping gets a boost," *Nature*, vol. 574, no. 7779. 2019.
[28] D. Somashekhara Reddy and Chandrasekhar, "Generalized light gradient boost classifier for traffic aware seamless mobility management in heterogeneous network," *Indian J. Comput. Sci. Eng.*, vol. 11, no. 1, 2020.
[29] A. K. Gorshenin and V. Y. Kuzmin, "Neural Network Forecasting of Precipitation Volumes Using Patterns," *Pattern Recognit. Image Anal.*, vol. 28, no. 3, 2018.
[30] R. Y. Choi, A. S. Coyner, J. Kalpathy-Cramer, M. F. Chiang, and J. Peter Campbell, "Introduction to machine learning, neural networks, and deep learning," *Transl. Vis. Sci. Technol.*, vol. 9, no. 2, 2020.
[31] U. Michelucci, *Applied deep learning: A case-based approach to understanding deep neural networks*. 2018.
[32] J. Moolayil, *Learn Keras for Deep Neural Networks*. 2019.
[33] Y. Zhou, H. Deng, T. Xu, T. Miao, and Q. Wu, "Unsupervised deep estimation modeling for tomato plant image based on dense convolutional auto-encoder," *Nongye Gongcheng Xuebao/Transactions Chinese Soc. Agric. Eng.*, vol. 36, no. 11, 2020.
[34] R. Delgado, "Introducción a la Validación Cruzada (k-fold Cross Validation) en R," *R Pubs*, 2018.
[35] K. Munir, H. Elahi, A. Ayub, F. Frezza, and A. Rizzi, "Cancer diagnosis using deep learning: A bibliographic review," *Cancers (Basel).*, vol. 11, no. 9, 2019.
[36] S. Fiorini, F. Hajati, A. Barla, and F. Girosi, "Predicting diabetes second-line therapy initiation in the Australian population via timespan-guided neural attention network," *bioRxiv*, 2019.