Aplicación de Red Neuronal Artificial para estimar el impago de préstamos bancarios.

Walter Jeremías López Flores.

Universidad Tecnológica Centroamericana UNITEC, Honduras.

wjlopez@unitec.edu

Resumen. El objetivo de este estudio es construir un modelo de Red Neuronal Artificial para la evaluación del riesgo crediticio en un banco de Honduras, mediante la aplicación de técnicas de aprendizaje de máquina supervisado y minería de datos a su cartera crediticia de personas naturales, para estimar si un cliente con determinadas características puede caer en impago de su préstamo. De una población de 38,156 préstamos y 30 características disponibles, 11 cuantitativas y 19 cualitativas se eligieron las variables independientes más representativas mediante un análisis de correlación, estableciendo un modelo base de tipo Perceptrón Multicapa con 6 variables en la capa de entrada, al que se fueron agregando las características de mayor relación con la variable dependiente *EnMora* en la capa de salida, para analizar su efecto sobre la capacidad predictiva del modelo constituido por 4 capas densas de 20 y 24 nodos y una capa central entre estas de tipo *dropout*. Se aplicó sobremuestreo SMOTE al subconjunto de entrenamiento para balancear las clases de la variable dependiente, cuyo modelo final presenta según las matrices de confusión y reportes de clasificación una exactitud de 99%, precisión de 99.43%, sensibilidad del 99.45%, especificidad de 95.15%, una curva ROC cuyo AUC es de 98.41% y una curva *Precision-Recall* con un AUC de 97.46%.

Palabras clave: Aprendizaje de Máquina, Red Neuronal Artificial, Perceptrón Multicapa, Riesgo de crédito, Impago de préstamos.

Application of Artificial Neural Network to estimate bank loan default

Abstract. The objective of this study is to build an Artificial Neural Network model for the evaluation of credit risk in a Honduran bank, by applying supervised machine learning techniques and data mining to its loan portfolio of natural persons, to estimate whether a client with certain characteristics may default on his loan. From a population of 38,156 loans and 30 available characteristics, 11 quantitative and 19 qualitative, to choose the most representative independent variables through a correlation analysis, establishing a Multilayer Perceptron base model with 6 variables in the input layer, to which they were adding the characteristics of greater relation with the dependent variable *default* on the output layer, to analyze its effect on the predictive capacity of the model consisting of 4 dense layers of 20 and 24 nodes and a central layer between these of dropout type. SMOTE oversampling was applied to the training

subset to balance the classes of the dependent variable, whose final model presents, according to the confusion matrices and classification reports, an accuracy of 99%, precision of 99.43%, sensitivity of 99.45%, specificity of 95.15%, a ROC curve whose AUC is 98.41% and a Precision-Recall curve with an AUC of 97.46%.

Keywords: Machine Learning, Artificial Neural Network, Multilayer Perceptron, Credit risk, Loans default.

1. Introducción.

El impacto de eventos como la crisis financiera mundial de 2008 y la pandemia mundial de Covid-19 a finales de 2019 genera, entre otros efectos, desaceleración y recesión prolongada en la economía global, fenómenos de los cuales es difícil recuperarse tanto para las personas, empresas y países. Lo anterior conlleva muchos riesgos implícitos, siendo uno de ellos el riesgo de crédito, pues tanto a nivel individual como empresarial, se requiere financiamiento oportuno para superar las adversidades que estas situaciones presentan en las necesidades de las personas y las empresas o particulares que las suplen, pero al presentarse dificultades para generar los ingresos necesarios dentro de ese ciclo económico, que les permitan ser sujetos a crédito ante la banca comercial, se presenta un doble riesgo para los bancos: que sus clientes con préstamos vigentes caigan en mora, y así mismo, ¿cómo evaluar nuevas solicitudes de crédito ante la incertidumbre, para pronosticar qué clientes potenciales con ciertas características a evaluar, puedan caer en mora si no se cuenta con historial crediticio a lo interno del banco?

Además, al considerar que la Inteligencia Artificial (IA) ha sido una de las áreas de las ciencias computacionales que mayor desarrollo han tenido desde mediados del siglo pasado, especialmente en su rama del aprendizaje automático o de máquina (ML por sus siglas en inglés), el cual tiene una amplia gama de aplicación en diferentes áreas del conocimiento e industrias, a nivel empresarial y financiero, incluyendo la gestión integral del riesgo; es oportuno conducir estudios que permitan a los bancos innovar y mejorar sus niveles de productividad y certeza en la toma de decisiones, especialmente en la colocación de préstamos, que es su negocio principal.

El objetivo de este estudio es proponer un modelo para la evaluación crediticia de los clientes en un banco en Honduras, que pueda ser reproducible para todo el sistema financiero nacional, entendiendo como clientes a las personas naturales que tienen préstamos vigentes y/o solicitan nuevas aplicaciones de crédito, para determinar si pueden o no caer en el impago de los mismos, construyendo una Red Neuronal Artificial a partir de su base de datos interna, diseñada con las bibliotecas de IA disponibles para el lenguaje de programación Python.

2. Marco Teórico.

El riesgo de crédito es la probabilidad de que una entidad no haga frente, parcial o totalmente, a su obligación de devolver una deuda o rendimiento convenido a su vencimiento sobre un instrumento financiero, ya sea por quiebra, iliquidez o cualquier

otra razón. La evaluación de este tipo de riesgo se basa en la probabilidad de que el prestatario incumpla con sus obligaciones, situación que en el argot financiero se dice que ocurre un *default*, el cual se relaciona con los ciclos económicos así: reduciéndose durante períodos de expansión económica, al mantenerse tasas de impago bajas, y sucediendo lo contrario en períodos de contracción económica [1].

2.1. Uso de Redes Neuronales Artificiales en la estimación de riesgo crediticio.

Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés) emulan la estructura y comportamiento del cerebro, utilizando procesos de ML para encontrar solución a diversos problemas, aplicando algoritmos matemáticos que encuentran relaciones no lineales entre conjuntos de datos, soliendo utilizarse como herramientas para predecir tendencias y/o clasificar conjuntos de datos [2].

Existen muchos tipos de ANN, las cuales varían según su aplicación, específicamente las de tipo Perceptrón Multicapa (MLP por sus siglas en inglés), son definidas por [3] como aquellas compuestas por varias capas, generalmente divididas en tres grupos: una capa de entrada (*input layer*), una o varias capas ocultas (*hidden layers*) y una capa de salida (*output layer*); también, constan de pesos entre sus conexiones, la función de activación, el valor para el sesgo (*bias*) y generalmente, se realiza un cálculo del error entre la salida deseada y la real para ajustar los pesos de forma inversa (*backpropagation*).

Respecto al número de capas, estas deben ser mayores si se trata de un problema no lineal y complejo, pero generalmente, un problema se puede representar bastante bien usando una o dos capas ocultas; y el número de neuronas por capa, usualmente se establece por ensayo y error, aunque existe también el criterio de considerar como valor referencial el promedio entre el número de entradas y salidas [4]. Las capas son densas cuando conectan cada neurona en una capa con todas las salidas de la capa anterior, también se conocen como capa completamente conectada [5].

Hay varios tipos de aprendizaje de máquina: supervisado, no supervisado y por refuerzo [6]. Cuando las redes son supervisadas, es requerido que el usuario especifique la salida deseada; a partir de ello, la red aprende a detectar la relación entre las entradas y salidas que fueron suministradas, siguiendo un proceso adaptativo e iterativo; y cuando la red ya ha sido entrenada, se le presentan nuevos datos, que no haya visto antes y se prueba para comprobar la bondad del conjunto de pesos, y cuando ofrece un rendimiento óptimo ya está lista para trabajar [2].

En el estudio [2] se aplica una metodología de medición de riesgo de crédito basado en un modelo MLP a partir de los datos de una cartera comercial, realizando un análisis exploratorio univariante y luego se fueron cruzando variables del cliente, del crédito y del comportamiento con la variable binaria *default* (fallidos y no fallidos), se investigó la correlación entre ellas, para establecer las características particulares del modelo de clasificación y determinar las ponderaciones necesarias para establecer la probabilidad de impago.

El uso de ANN para la calificación y evaluación de créditos ha sido efectivo durante la última década, pues su capacidad en tales aplicaciones se debe a la forma en que opera la red y la disponibilidad de datos de entrenamiento, lo cual es más evidente cuando se utilizan redes MLP basadas en el algoritmo de aprendizaje de retropropagación [7]. Cuando se alimenta la información de un solicitante de crédito a

la ANN, los atributos (respuestas del solicitante a un conjunto de preguntas o características) se toman como entrada a la red y se toma una combinación lineal de ellos con pesos arbitrarios, los atributos se combinan linealmente y están sujetos a una transformación no lineal representada por una determinada función de activación, luego se alimentan como entradas en la siguiente capa y así hasta la función final que produce valores que se pueden comparar con un punto de corte para la clasificación; cada caso de entrenamiento se envía a la red y se compara con el valor observado, el residual se propaga a la red y los pesos se modifican en cada capa según la contribución que cada peso hace al valor de error [7].

En cuanto a la selección de funciones de activación para los modelos ANN, según [4] se realiza por ensayo y error, de acuerdo al problema a resolver o al criterio del investigador, pues aún no existe un criterio estándar en la literatura; comúnmente la función logística o sigmoidea es utilizada con buenos resultados, aunque también se emplean las funciones *heaviside*, *signum*, lineal, *piece-wise linear*, tangente hiperbólica, ReLU (*Rectified Linear Unit*) y *rectifier softplus* descritas en [8], las cuales según [9] son llamadas también funciones de transferencia y se definen como la suma ponderada de la entrada se transforma en una salida en una capa de la red, y propone la siguiente metodología de selección de funciones para los nodos: si son de capa oculta, para MLP o CNN (*Convolutional Neural Network*) usar ReLU y en redes recurrentes usar sigmoidea o tangente; luego para la capa de salida, si es un problema de regresión, usar lineal y para clasificación, si es binaria, sigmoidea; y si es multiclase, *softmax* o sigmoidea.

Los métodos de regularización estocástica, como el abandono (*dropout*) son una forma eficaz de evitar el ajuste excesivo o sobre-entrenamiento (*over-fitting*) y se utilizan a menudo en la práctica debido a su simplicidad [10]. La capa de abandono en una red, establece aleatoriamente las unidades de entrada en 0 con una frecuencia de velocidad a cada paso durante el tiempo de entrenamiento, lo que ayuda a evitar el sobreajuste; las entradas que no están configuradas en 0 se escalan en $1/(1 - tasa)$ de modo que la suma de todas las entradas no cambia [11].

Los modelos de Keras, tienen dos modos: entrenamiento y prueba, los mecanismos de regularización se desactivan al momento de prueba, son reflejados en la pérdida de tiempo de entrenamiento, pero no en la pérdida de tiempo de prueba; además, la pérdida de entrenamiento es el promedio de las pérdidas para cada lote de datos de entrenamiento, durante la época (*epoch*) actual, debido a que su modelo cambia con el tiempo, la pérdida en los primeros lotes de una época es generalmente mayor que en los últimos lotes, resultando al final en una pérdida menor. En los modelos ANN, una época es un límite arbitrario, definido como una pasada sobre todo el conjunto de datos, que se utiliza para separar el entrenamiento en distintas fases, para el registro y evaluación periódica [12].

El propósito de las funciones de pérdida dentro de las ANN, es calcular la cantidad que un modelo debe buscar minimizar durante el entrenamiento. La función probabilística de pérdida llamada entropía cruzada binaria (*binary crossentropy*), calcula la pérdida entre etiquetas verdaderas y predichas cuando solo hay dos clases de etiquetas (0 y 1), para cada ejemplo debería haber un único valor de punto flotante por predicción [13]. Existe un algoritmo para optimización basada en gradientes de primer orden de funciones objetivas estocásticas, y en estimaciones adaptativas de momentos de orden inferior llamado *Adam*, que es un método sencillo de implementar,

computacionalmente eficiente, con pocos requisitos de memoria, invariante al cambio de escala diagonal de los gradientes y muy adecuado para problemas con grandes cantidades de datos o parámetros, así como para objetivos no estacionarios y problemas con gradientes muy ruidosos o dispersos [10].

2.2. Matriz de confusión, métricas y curvas ROC y Precision-Recall.

En los problemas de decisión binaria, el clasificador etiqueta ejemplos como positivos o negativos. La decisión tomada por el clasificador se puede representar en una estructura conocida como matriz de confusión o tabla de contingencia, la cual tiene cuatro categorías: los verdaderos positivos (TP) son observaciones correctamente etiquetadas como positivas, los falsos positivos (FP) se refieren a ejemplos negativos etiquetados incorrectamente como positivos (llamados error tipo I); los verdaderos negativos (TN) corresponden a los negativos etiquetados correctamente como negativos y finalmente, los falsos negativos (FN), que se refieren a ejemplos positivos etiquetados incorrectamente negativos (llamados error tipo II) [14]. A continuación, se muestra la estructura de una matriz de confusión y sus principales métricas.

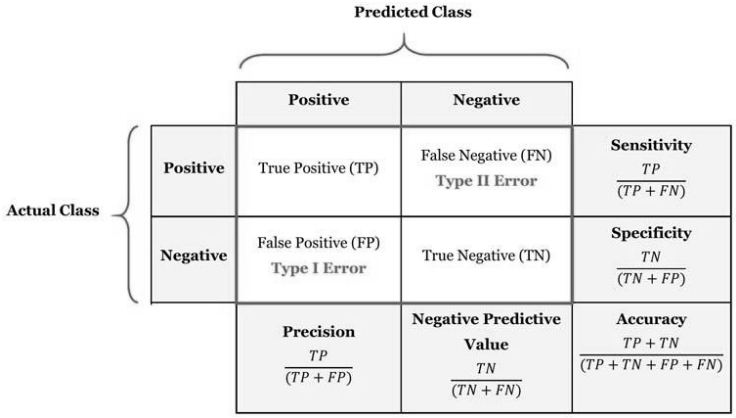

Fig. 1. Esquema de una matriz de confusión y fórmulas de sus métricas [15].

La precisión o *Precision*, indica la fracción de los casos predichos correctamente que resultaron realmente positivos, la cual es una métrica útil en los casos en que los FP son una preocupación mayor que los FN; mientras que la sensibilidad o *Sensitivity* (también conocida como *Recall*), indica la fracción de los casos positivos reales que pudo predecir correctamente el modelo, siendo una métrica útil en los casos en que los FN triunfan sobre los FP [16].

Otra métrica importante que surge de combinar *Precision* y *Recall* es el *F1-score* o puntuación F1, que por definición es la media armónica entre estas; la cual, según [17] se usa ampliamente para medir el éxito de un clasificador binario cuando una clase es rara, mientras que otras puntuaciones como *micro average* (micropromedio), *macro average* (macropromedio) y el promedio $F1$ por instancia, se utilizan en la clasificación de múltiples etiquetas y también demuestran cómo esta media armónica puede ser expresada también por sustitución como una función de cuenta de TP, FP y FN como se indica en la Ecuación (1):

$$F1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = \frac{2TP}{2TP + FP + FN} \quad (1)$$

En la práctica, cuando se intenta aumentar la precisión de los modelos, la sensibilidad también disminuye y viceversa; el puntaje $F1$ captura ambas tendencias en un solo valor y brinda una idea combinada de estas métricas, alcanzando su valor máximo cuando $Precision = Recall$ [16]. En la actualidad, el ML se ha alejado de simplemente presentar resultados de precisión al realizar una validación empírica de nuevos algoritmos, especialmente cuando estos generan probabilidades de valores de clase, por lo que se recomiendan utilizar curvas ROC (*Receiver Operator Characteristic*) que muestran cómo el número de ejemplos positivos clasificados correctamente varía con el número de ejemplos negativos clasificados incorrectamente, y curvas *Precision-Recall* (PR) como una alternativa a las curvas ROC para tareas con un gran sesgo en la distribución de clases [14].

2.3. Conjuntos de datos desequilibrados y sobremuestreo SMOTE.

Los conjuntos de datos desequilibrados son un problema común dentro del ML, ya que la mayoría de modelos de clasificación tradicionales no pueden manejarlos, pues ocurre un alto costo de clasificación errónea a menudo en la clase minoritaria, porque el modelo intentará clasificar toda la muestra de datos en la clase mayoritaria; para abordar este problema se usa una técnica llamada *oversampling* (sobremuestreo) que permite crear una muestra de clase específica para que la distribución de clases del conjunto de datos original pueda equilibrarse. La técnica SMOTE (*Synthetic Minority Oversampling Technique*) es una de las técnicas de sobremuestreo más populares, la cual funciona encontrando un grupo de k vecinos más cercanos en el espacio de características, luego encuentra aleatoriamente un punto dentro de este grupo, y finalmente usa el promedio ponderado para "falsificar" el nuevo punto de datos [18].

Para encontrar la distancia entre los puntos vecinos, se discute en [19] que la función de distancia euclidiana es inapropiada para atributos nominales, y otra alternativa llamada VDM es inapropiada para atributos continuos, por lo ninguna es suficiente por sí misma para un conjunto de datos que utilice ambas; ante lo cual, describen una función adecuada de distancia heterogénea llamada *Heterogeneous Value Difference Metric* (HVDM) con la Ecuación (2):

$$HVDM(x, y) = \sqrt{\sum_{a=1}^{m} {d_a}^2(x_a, y_a)} \quad (2)$$

En Python existe un paquete llamado SMOTE-NC que permite crear datos sintéticos tanto para variables categóricas como continuas dentro de un conjunto de datos, decidiendo las categorías de una nueva muestra generada eligiendo la más frecuente de los vecinos más cercanos presentes durante la generación [20]. El procedimiento de validación cruzada después del sobremuestreo conduce a resultados sobre optimistas, mientras que realizarlo en los conjuntos de entrenamiento en cada iteración de un procedimiento de validación cruzada es la forma correcta de validar los resultados en escenarios desequilibrados [21].

3. Metodología.

El estudio es de enfoque cuantitativo, alcance explicativo y de tipo experimental; los datos provienen de fuente secundaria, extraídos de la base de datos interna de un banco de Honduras mediante consulta electrónica del reporte de Central de Información Crediticia (CIC) que se entrega al ente regulador: la Comisión Nacional de Banca y Seguros (CNBS) de dicho país, con saldos de los préstamos otorgados y que aún no han sido cancelados, los cuales van desde el año 2000 hasta diciembre de 2020. El conjunto de datos se compone de 51,696 registros y 30 campos de los que se identificarán las variables independientes más adecuadas para construir el modelo.

En cuanto a las herramientas utilizadas: la limpieza de los datos, su preparación y exploración preliminar fue realizada en Excel y se empleó el *framework* Anaconda para la manipulación de datos, diseño, entrenamiento, prueba y visualización de un modelo MLP en lenguaje Python 3.7 con las bibliotecas de IA Keras y Scikit-Learn codificando los algoritmos de ML supervisado en el editor Jupyter Notebook 6.0.

3.1. Población y muestra.

La población a estudiar estará compuesta solo por los préstamos concedidos a personas naturales; el tipo de muestra será censal, por lo que no se requiere definir tamaño muestral ni utilizar tipo de muestreo alguno, ya que se analizará todo el universo completo. Al delimitar la cartera crediticia filtrando únicamente los préstamos de personas naturales, esta queda establecida en 38,156 observaciones, de las cuales solo el 10.38% (3,959) se encuentran en mora y las restantes 34,197 no lo están (clase *EnMora* = 0), que sería la clase mayoritaria con un 89.62% de las observaciones.

3.2. Identificación y selección de variables.

La variable dependiente para el modelo es de tipo dicotómica y llamada *EnMora*, la cual indicará si el préstamo está en mora con el valor 1, o no lo está con el valor 0, en función de un arreglo bidimensional de variables predictoras, compuesto por 11 cuantitativas y 19 cualitativas, para un *dataframe* de 30 variables disponibles, analizando la intensidad y dirección de la relación de estas con la variable dependiente mediante la matriz de correlación de la Fig. 2.

El número de días de atraso del préstamo (NODIAA) es la única variable independiente cuantitativa que presenta una relación fuerte con la mora (0.69), mientras que solo dos características cualitativas presentan una relación importante con la variable respuesta, siendo la más fuerte el estado de operación (ESTAOP) con 0.89, seguida de la categoría de riesgo (CATRGO) con 0.54.

Todas las restantes no llegan a 0.1, por lo que no son significativas como para considerar que su efecto sea influyente dentro del modelo; a pesar de ello, se seleccionaron las siguientes 6 variables por considerarse de interés por los analistas de crédito del banco para predecir el comportamiento de la mora, dado que representan características inherentes tanto de los préstamos: duración del préstamo en años (DURACION), el monto del capital otorgado (MONOTO), y la tasa de interés (TASINT); así como de las personas: años de laborar (LABTIE), edad del cliente (EDAD) y su género (GENCOD).

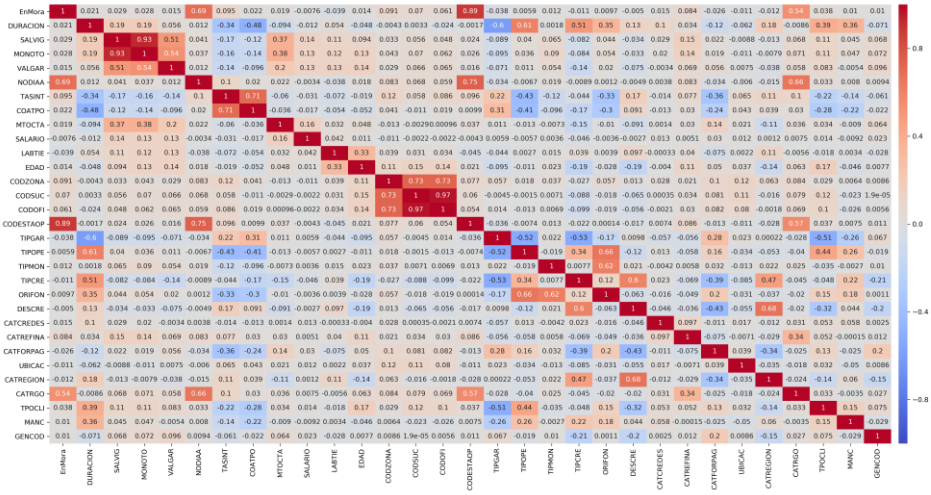

Fig. 2. Matriz de correlación de variables para el modelo MLP.

4. Resultados.

Se construyó un modelo base de tipo MLP con 4 capas densas ocultas y una capa *dropout* intermedia, utilizando funciones de activación de tipo ReLU, mientras que la capa de salida tendrá un solo nodo, con una función de activación sigmoidea, ideal para predecir una variable binaria. Para el modelo base se tomaron las 6 variables de entrada seleccionadas anteriormente, por ser atributos básicos inherentes a los préstamos y clientes, y también porque son conocidas de antemano, aún para nuevos clientes que tomarán un préstamo por primera vez.

Tabla 1. Diseño de los modelos MLP utilizados en el estudio.

Estructura		**Parámetros**					
Capa	Nodos	Modelo base	1: base + CATRGO	2: base + ESTAOP	3: base + NODIAA	4: base + CATRGO y NODIAA	5: Todas
Entrada	$X =$	6	7	7	7	8	9
Oculta 1 (densa)	20	140	160	160	160	180	200
Oculta 2 (densa)	24	504	504	504	504	504	504
Oculta 3 (dropout)	24	0	0	0	0	0	0
Oculta 4 (densa)	20	500	500	500	500	500	500
Oculta 5 (densa)	24	504	504	504	504	504	504
Salida (densa)	1	25	25	25	25	25	25
Parámetros totales:		1,673	1,693	1,693	1,693	1,713	1,733
Parámetros entrenables:		1,673	1,693	1,693	1,693	1,713	1,733
Parámetros no entrenables:		0	0	0	0	0	0

La escala de la variable MONOTO, se cambió, normalizándola en lugar de usar sus valores originales. Las 3 variables que resultaron con mayor correlación se fueron

agregando al modelo base para analizar de qué manera contribuyen a explicar la mora, tanto individualmente como en conjunto, combinado en distintos experimentos las variables en la capa de entrada, así como el número de nodos que conforman cada capa oculta hasta alcanzar los mejores resultados (ver Tabla 1).

Tabla 2. Resultados de entrenamiento de los modelos MLP diseñados.

Epoch	Modelo base		1: base + CATRGO		2: base + ESTAOP		3: base + NODIAA		4: base + CATRGO y NODIAA		5: Todas	
	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc
1	0.3647	0.8926	0.3607	0.8901	0.3625	0.8940	0.0855	0.9765	0.1185	0.9730	0.4976	0.9554
2	0.3421	0.8966	0.2810	0.9107	0.1695	0.9451	0.0493	0.9858	0.0471	0.9879	0.0482	0.9875
3	0.3366	0.8966	0.2696	0.9133	0.0341	0.9912	0.0436	0.9880	0.0399	0.9893	0.0403	0.9894
4	0.3366	0.8966	0.2606	0.9172	0.0217	0.9950	0.0405	0.9889	0.0373	0.9903	0.0361	0.9911
5	0.3341	0.8966	0.2595	0.9154	0.0173	0.9959	0.0402	0.8996	0.0366	0.9907	0.0329	0.9915
Score	0.3331	89.66%	0.2489	92.03%	0.0051	100.0%	0.0411	99.23%	0.0327	99.12%	0.0317	99.00%

El conjunto de datos se dividió en dos subconjuntos: el de entrenamiento conformado por el 70% de las unidades muestrales que equivale a 26,709 préstamos, y el de prueba con el 30% restante equivalente a 11,447. El entrenamiento de los modelos, se realizó durante cinco *epochs*, utilizando el optimizador *Adam* y la función de pérdida *binary crossentropy* (ver Tabla 2). Tanto la función de pérdida se va reduciendo en todos los modelos con respecto al base, así como la exactitud de todos va aumentando en cada época.

Tabla 3. Matrices de confusión con los resultados de entrenamiento y prueba.

Modelo		**Entrenamiento**				**Prueba**			
		0	**1**	**Total**	**Loss/Acc**	**0**	**1**	**Total**	**Loss/Acc**
Base	**0**	23,946	0	23,946	0.3316	10,251	0	10,251	0.3336
	1	2,763	0	2,763	89.66%	1,196	0	1,196	89.55%
	Total	26,709	0	26,709		11,447	0	11,447	
1: base + CATRGO	**0**	23,567	379	23,946	0.2489	10,070	181	10,251	0.2497
	1	1,750	1,013	2,763	92.03%	730	466	1,196	92.04%
	Total	25,317	1,392	26,709		10,800	647	11,447	
2: base + ESTAOP	**0**	23,946	0	23,946	0.0051	10,251	0	10,251	0.0051
	1	1	2,762	2,763	100.0%	0	1,196	1,196	100.0%
	Total	23,947	2,762	26,709		10,251	1,196	11,447	
3: base + NODIAA	**0**	23,869	77	23,946	0.0411	10,221	30	10,251	0.0463
	1	128	2,635	2,763	99.23%	70	1,126	1,196	99.13%
	Total	23,997	2,712	26,709		10,291	1,156	11,447	
4: base + CATRGO y NODIAA	**0**	23,936	10	23,946	0.0327	10,227	4	10,251	0.0402
	1	224	2,539	2,763	99.12%	120	1,076	1,196	98.91%
	Total	24,160	2,549	26,709		10,347	1,080	11,447	
5: Todas	**0**	23,942	4	23,946	0.0317	10,248	3	10,251	0.0384
	1	262	2,501	2,763	99.00%	135	1,061	1,196	98.79%
	Total	24,204	2,505	26,709		10,383	1,064	11,447	

Al comparar los resultados de clasificación y métricas entre los modelos con los datos de entrenamiento y prueba para evaluar la capacidad de clasificación de cada uno (ver Tabla 3), se demuestra que no hubo *over-fitting*, por lo que se presentan los reportes de clasificación de los modelos de prueba (ver Tabla 4). Los resultados, reflejan que el modelo base, con las características proporcionadas fue incapaz de aprender a clasificar los préstamos cuando la clase $EnMora = 1$, lo cual se explica por la baja correlación que mantienen con la variable dependiente, además de ser la clase minoritaria.

Tabla 4. Reportes de clasificación de los modelos con datos de prueba.

Modelo base

	precision	recall	f1-score	support
0	0.90	1.00	0.94	10251
1	0.00	0.00	0.00	1196
accuracy			0.90	11447
macro avg	0.45	0.50	0.47	11447
weighted avg	0.80	0.90	0.85	11447

Modelo 1: base + CATRGO

	precision	recall	f1-score	support
0	0.92	0.99	0.95	10251
1	0.82	0.21	0.34	1196
accuracy			0.91	11447
macro avg	0.87	0.60	0.65	11447
weighted avg	0.91	0.91	0.89	11447

Modelo 2: base + ESTAOP

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10251
1	1.00	1.00	1.00	1196
accuracy			1.00	11447
macro avg	1.00	1.00	1.00	11447
weighted avg	1.00	1.00	1.00	11447

Modelo 3: base + NODIAA

	precision	recall	f1-score	support
0	0.99	1.00	1.00	10251
1	0.98	0.94	0.96	1196
accuracy			0.99	11447
macro avg	0.99	0.97	0.98	11447
weighted avg	0.99	0.99	0.99	11447

Modelo 4: base + CATRGO y NODIAA

	precision	recall	f1-score	support
0	0.99	1.00	0.99	10251
1	1.00	0.90	0.95	1196
accuracy			0.99	11447
macro avg	0.99	0.95	0.97	11447
weighted avg	0.99	0.99	0.99	11447

Modelo 5: Todas

	precision	recall	f1-score	support
0	0.99	1.00	0.99	10251
1	1.00	0.89	0.94	1196
accuracy			0.99	11447
macro avg	0.99	0.94	0.97	11447
weighted avg	0.99	0.99	0.99	11447

Cabe reportar que se realizó una prueba en el modelo base con todas las demás variables que fueron excluidas, obteniendo resultados similares, por lo que ninguna de ellas contribuye a mejorar el aprendizaje de este para clasificar la mora. Al agregar la categoría de riesgo del préstamo al modelo base, este comienza a aprender a clasificar los que caen en mora, pero comete muchos errores de Tipo I y Tipo II, situación que se reduce sustancialmente al sustituir esta variable por el número de días de atraso en el experimento 3, el cual aprende a predecir la clase minoritaria con buenos resultados.

El modelo aprende a identificar los préstamos que caen en mora de manera perfecta en el experimento 2 dado que conoce el estado de la operación, pero esto debido a que todos los que tienen estado vigente se encuentran al día, y cualquier estado distinto implicaría que el préstamo ha caído en mora, por lo que no es una característica idónea para predecir; también, al agregar todas las variables restantes al modelo base se obtienen resultados casi perfectos, pero esto porque está incluida ESTAOP, y es como si las otras le restaran exactitud (ver Fig. 3).

En base a los resultados obtenidos, se selecciona como preferido el modelo 3, que presenta un AUC bajo la curva ROC (AUROC) de 98.14%, un AUC de 97.09% bajo la curva PR, y una exactitud de 99.13%; aunque los modelos 4 y 5 obtuvieron mayores valores en dichos indicadores, dicho experimento 3 presenta el *Recall* y *F1-score* más

altos para la clase minoritaria, con 0.94 y 0.96 respectivamente, observando también que en los siguientes modelos estos fueron disminuyendo, por lo que se elige este como el modelo MLP preferido, el cual se representa gráficamente en la Fig. 4.

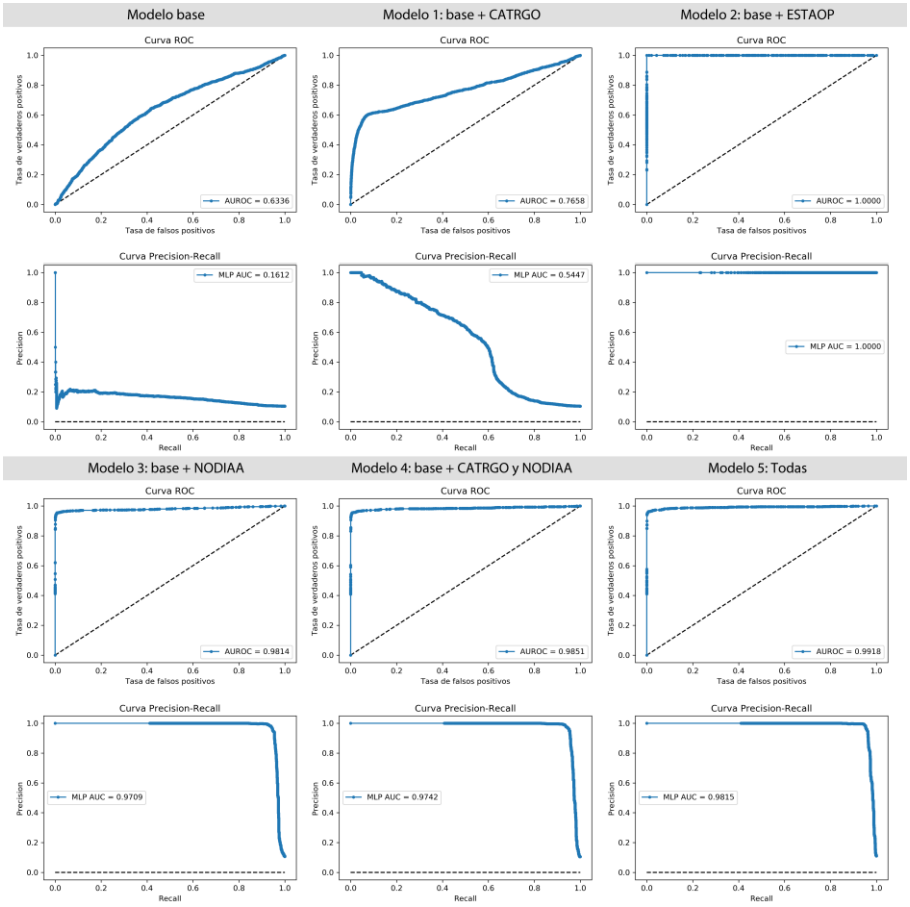

Fig. 3. Curvas ROC y PR de los modelos MLP con los datos de prueba.

4.1. Sobremuestreo SMOTE.

A pesar que el modelo seleccionado demuestra en sus resultados, tasas altas de reconocimiento sobre la clase minoritaria, se le aplica un sobremuestreo SMOTE al subconjunto de datos de entrenamiento, para analizar si presenta alguna mejora al entrenar con clases balanceadas, para lo cual se utilizó el paquete SMOTE-NC de la biblioteca *imblearn* de Python, indicándole que la variable GENCOD era categórica, quedando cada clase binaria con 23,946 observaciones para un *dataframe* de 47,892 registros. Al volver a entrenar el modelo 3 con los nuevos datos sintéticos equilibrados en sus clases de salida, durante 5 e*pochs*, y luego proporcionarle los datos de prueba se obtienen los resultados presentados en la Tabla 5, se alcanzan prácticamente los mismos

valores de exactitud, precisión y sensibilidad; la capacidad de predicción de valores negativos baja a 95.5%, un 2.15% menos con respecto al mismo modelo entrenando sin sobremuestreo, pero la especificidad sube en 1.06%, lo cual indica una ligera mejora en la capacidad de categorizar la clase minoritaria al utilizar el sobremuestreo SMOTE para entrenar el modelo seleccionado.

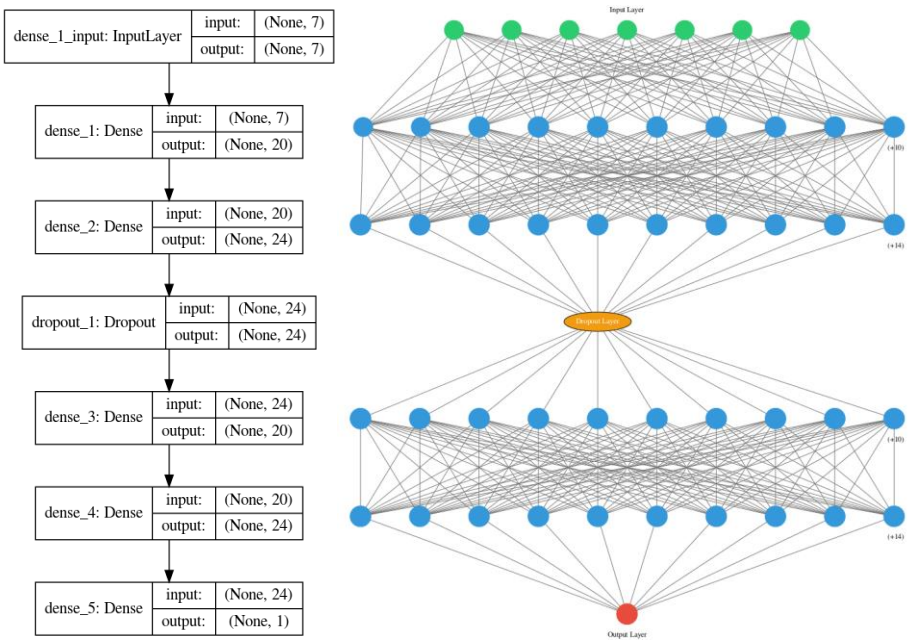

Fig. 4. Diagrama del modelo MLP seleccionado.

Los errores de Tipo I se reducen y mejoran las curvas de clasificación después de aplicar SMOTE para optimizar el modelo elegido, su AUROC subió en 0.27% de 0.9814 a 0.9841, y el AUC de la curva PR sube 0.37%, de 0.9709 a 0.9746.

Tabla 5. Comparación de resultados del modelo MLP optimizado con SMOTE.

Modelo 3 con datos de prueba entrenado con clases desbalanceadas					**Modelo 3 con datos de prueba entrenado con clases balanceadas mediante SMOTE-NC**			
	0	**1**	**Total**	**Métricas**	**0**	**1**	**Total**	**Métricas**
0	10,221	30	10,251	Sen = 0.9971	10,195	56	10,251	Sen = 0.9945
1	70	1,126	1,196	Esp = 0.9415	58	1,138	1,196	Esp = 0.9515
Total	10,291	1,156	11,447	Acc = 0.9913	10,253	1,194	11,447	Acc = 0.9900
Pre = 0.9932		Neg Pre = 0.9740			Pre = 0.9943		Neg Pre = 0.9531	

	precision	recall	f1-score	support
0	0.99	1.00	1.00	10251
1	0.97	0.94	0.96	1196
accuracy			0.99	11447
macro avg	0.98	0.97	0.98	11447
weighted avg	0.99	0.99	0.99	11447

	precision	recall	f1-score	support
0	0.99	0.99	0.99	10251
1	0.95	0.95	0.95	1196
accuracy			0.99	11447
macro avg	0.97	0.97	0.97	11447
weighted avg	0.99	0.99	0.99	11447

5. Conclusiones y Trabajo a Futuro.

El estudio entrega un modelo MLP construido en Python a partir de los datos internos que conforman la cartera de préstamos de personas naturales de un banco hondureño. Dicha herramienta es de gran utilidad práctica para los analistas de crédito, ya que les permitirá evaluar si un cliente con ciertas características ingresadas en las variables de entrada seleccionadas, puede caer en mora del préstamo que tiene vigente, o que está solicitando en caso de ser nuevo, siendo el número de días de atraso la variable que más contribuye a explicar la mora para que el modelo aprenda a clasificar los préstamos o solicitudes que pueden caer en impago.

El modelo seleccionado aprende a clasificar las categorías binarias de la variable de salida de manera adecuada, inclusive de la clase minoritaria sin necesidad de aplicar sobremuestreo, aunque su capacidad de clasificación mejora ligeramente cuando las etiquetas de esta se encuentran balanceadas.

La investigación al estar delimitada a un solo banco, no provee un modelo generalizado para todo el sistema financiero del país en cuestión; sin embargo, la metodología propuesta puede replicarse en cualquier otro banco, cooperativa o empresa financiera del sistema nacional, o de cualquier otro país, con solo cambiar los datos de entrada y ajustando el algoritmo a estos. De igual manera, el ente regulador, puede diseñar y entrenar un modelo general, a partir de la base de datos consolidada que mantiene de todas las entidades supervisadas del sistema financiero dentro de su CIC.

En investigaciones futuras, se pueden considerar variables macroeconómicas dentro del modelo, u otras variables exógenas, para no limitarlo a los datos internos de un banco u otra institución financiera.

Referencias.

1. Saavedra García, M. L., & Saavedra García, M. J. (2010). Modelos para medir el riesgo de crédito de la banca. Cuadernos de Administración, 23(40), 295-319.
2. Pérez Ramírez, F. O., & Fernández Castaño, H. (2007). Las Redes Neuronales y la Evaluación del Riesgo de Crédito. Revista Ingenierías, 6(10), 77-91. Obtenido de http://udem.scimago.es/index.php/ingenierias/article/view/225.
3. Peña, M., & Orellana, J. (2018). Red neuronal para clasificación de riesgo en cooperativas de ahorro y crédito. Congreso de Ciencia y Tecnología, 13(1), 121-124. doi:10.24133/cctespe.v13i1.710.
4. Llano, L., Hoyos, A., Arias, F., & Velásquez, J. (2007). Comparación del desempeño de funciones de activación en redes Feedforward para aproximar funciones de datos con y sin ruido. Avances en Sistemas e Informática, 4(2), 79-87.
5. Google. (s.f.). Glosario sobre aprendizaje automático. Obtenido de Developers Google: https://developers.google.com/machine-learning/crash-course/glossary.
6. Lee, I., & Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. Business Horizons, 63(2), 157-170. doi:https://doi.org/10.1016/j.bushor.2019.10.005.

7. Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. Expert Systems with Applications, 37, 6233-6239. doi:10.1016/j.eswa.2010.02.101.
8. Raschka, S. (2016). Activation Functions for Artificial Neural Networks. Obtenido de mlxtend: https://rasbt.github.io/mlxtend/user_guide/general_concepts/activation-functions.
9. Brownlee, J. (2021). How to Choose an Activation Function for Deep Learning. Obtenido de Machine Learning Mastery: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/.
10. Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv.org. doi:arckiv.org/pdf/1412.6980v9.pdf.
11. Keras. (s.f.). Dropout layer. Obtenido de Keras API reference: Regularization layers: https://keras.io/api/layers/regularization_layers/dropout/.
12. Keras. (s.f.). Keras FAQ. Obtenido de Keras.io: https://keras.io/getting_started/faq/.
13. Keras. (s.f.). Probabilistic losses. Obtenido de Keras API Reference: Losses: https://keras.io/api/losses/probabilistic_losses/.
14. Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. Proceedings of the 23rd International Conference on Machine Learning - ICML '06. doi:10.1145/1143844.1143874.
15. Data Science. (2019). What is Confusion Matrix and Advanced Classification Metrics? Obtenido de Data Science and Machine Learning: https://manisha-sirsat.blogspot.com/2019/04/confusion-matrix.html.
16. Bhandari, A. (2020). Everything you Should Know about Confusion Matrix for Machine Learning. Obtenido de Medium Analytics Vidhya: https://analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/.
17. Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding Classifiers to Maximize F1 Score. ArXiv.org. doi:arxiv.org/abs/1402.1892.
18. Jiang, P., Zhang, J., & Zou, J. (2019). Credit Card Fraud Detection Using Autoencoder Neural Network. arXiv.org. doi:arxiv.org/abs/1908.11553.
19. Wilson, D. R. & Martínez, T. R. (1997). Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research, 6(1997), 1-34.
20. Aguilar, F. (2019). SMOTE-NC in ML Categorization Models for Imbalanced Datasets. Obtenido de Medium Analytics Vidhya: https://medium.com/analytics-vidhya/smote-nc-in-ml-categorization-models-fo-imbalanced-datasets-8adbdcf08c25.
21. Senoane Santos, M., Pompeu Soares, J., Henriques Abreu, P., Araújo, H. & Santos, J. (2018). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches. IEEE Computational Inteligence Magazine, 13(4), 59-76. doi:10.1109/MCI.2018.2866730.