

# **Modelo híbrido para mejorar la predicción de credit scoring: Un análisis comparativo de técnicas de minería de datos**

## **Resumen**

La evaluación del riesgo de pago o no pago de un solicitante de crédito es un tema relevante para las entidades financieras. Incluso un ligero incremento en la capacidad predictiva (grado de acierto en la predicción del pago/no pago) implica grandes ahorros. En esa medida, las organizaciones disponen de diversas técnicas de minería de datos para evaluar este riesgo. Un aspecto crucial es conocer comparativamente la capacidad predictiva de las técnicas y tratar de optimizarla. Este estudio utiliza datos de una entidad financiera para evaluar tres técnicas logit, redes neuronales y modelo híbrido (redes neuronales + algoritmos genéticos). Los resultados muestran que el modelo híbrido arroja una capacidad mayor que las redes neuronales y modelo logit. Se discuten implicancias prácticas de estos hallazgos.

Palabras clave: Credit Scoring, logit, redes neuronales, algoritmos genéticos, modelos híbridos

## **1. Introducción**

El contexto de crisis financiera de los últimos años le ha brindado prioridad a la evaluación del riesgo crediticio para diferentes entidades, tanto para prestamistas, como para bancos o los reguladores. En la medida que el riesgo crediticio se considera el problema más serio de la banca, los sistemas de credit scoring resultan herramientas relevantes para la evaluación y monitoreo del riesgo crediticio (Anagnostopoulos & Abedi, 2016; Basilea, 2004; Doumpos & Zopounidis, 2014). De ese modo, el estudio de la evaluación del riesgo crediticio cuenta con gran interés en la academia y ha sido materia de continua investigación (Abdou, Pointon, & El-Masry, 2008).

Específicamente, una entidad financiera confronta la decisión de otorgar o no el crédito a un solicitante. Para ello, debe realizar una ponderación de la probabilidad del solicitante de

cumplir o no el pago del crédito. Esta ponderación puede basarse en el juicio del evaluador o utilizar técnicas de minería de datos. La mayor cantidad de solicitantes y atributos a considerar ha hecho que cada vez las técnicas de minería de datos sean más utilizadas.

De ese modo, las técnicas empleadas en las entidades financieras han sido diversas: desde modelos paramétricos (p.e. logit, análisis discriminante) hasta modelos no paramétricos (p.e. redes neuronales, máquinas de soporte vectorial).

La literatura por su parte ha desarrollado estudios comparando diversas técnicas. En esa línea, varios artículos han comparado el poder predictivo de las redes neuronales versus técnicas paramétricas como el análisis discriminante, regresión logística, el modelo probit o el modelo logit. Los resultados muestran, en general, un mayor poder predictivo de las redes neuronales sobre las técnicas paramétricas (Abdou & Pointon, 2011; Cimpoeru, 2011; Tam & Kiang, 1992).

Sin embargo, con carteras de préstamos considerables, incluso una ligera mejora en la precisión del credit scoring puede reducir el riesgo de los acreedores, lo que se traduce considerablemente en futuros ahorros (Huang, Chen, & Wang, 2007). Por lo tanto, el reto actualmente está en lograr mejoras (aunque sean mínimas) en la capacidad predictiva.

En esa línea, un campo recientemente explorado es el de los modelos híbridos donde se utilizan más de una técnica de minería de datos que sean complementarios entre sí para lograr mejoras en la capacidad predictiva (Huang et al., 2007).

Diversos estudios utilizan redes neuronales conjuntamente con otras técnicas. Por ejemplo, Hsieh (2005) utiliza un modelo híbrido de análisis de clúster para optimizar la muestra que luego será tratada con redes neuronales. Lee and Chen (2005) la utiliza conjuntamente con regresiones multivariadas (Lee & Chen, 2005) y Lee, Chiu, Lu, and Chen (2002) la emplea con el análisis discriminante.

Por otro lado, la literatura señala que los algoritmos genéticos pueden ser usados como técnicas que optimizan las entradas (subconjunto de datos) y parámetros de otros modelos (Goldberg & Holland, 1988). Por ejemplo, Huang et al. (2007), emplea algoritmos genéticos para optimizar el número de atributos y los parámetros de un modelo de máquinas de soporte vectorial.

En base de la revisión efectuada, no se encuentra un estudio que compare técnicas paramétricas, con redes neuronales (puros) y modelos híbridos (redes neuronales con algoritmos genéticos como optimizador). En ese sentido, el objetivo de este estudio es aplicar tres técnicas: (1) paramétrica, con análisis logit; (2) no paramétrica-pura, con redes neuronales y (3) no paramétrica-híbrida, con redes neuronales y algoritmos genéticos. Y así comparar, en función a los resultados empíricos, su capacidad predictiva.

De ese modo, el artículo se divide en una revisión de literatura donde se define el problema del credit scoring y se muestra las técnicas a comparar. Luego, se aplican dichas técnicas al caso de una entidad financiera latinoamericana; y finalmente, se discuten los resultados obtenidos.

## **2. Revisión de literatura**

### **Credit Scoring**

Se define como riesgo crediticio a la probabilidad de que un prestatario incumpla con los pagos de sus obligaciones. En este caso se clasifica al solicitante – previo al otorgamiento del crédito – como bueno o malo, en base de sus atributos (edad, estado civil, nivel de ingresos, etc) (Khashman, 2010; Laha, 2007). La clasificación de se puede realizar con técnicas de calificación de riesgos automatizadas para reconocer patrones que ayuden a la clasificación (Abdou & Pointon, 2011). Un tema central en el credit scoring es la capacidad predictiva del modelo. Las técnicas de minería de datos se basan en datos históricos y estiman un modelo (o

función matemática) que sirve para predecir el comportamiento de un nuevo solicitante (como buen o mal pagador). En ese sentido, la capacidad predictiva del modelo, en términos de su grado de acierto en la predicción, es un indicador clave para evaluar un modelo y también para comparar entre modelos alternativos (obtenidos con técnicas diferentes).

## **Técnicas de clasificación paramétricas y no paramétricas**

Las técnicas de clasificación que ofrece la minería de datos se pueden formular de diversas maneras; sin embargo, para este estudio resulta pertinente la clasificación técnicas paramétricas y no paramétricas (Abdou & Pointon, 2011; West, 2000).

Para ello, es importante considerar que:

$$Y = F(X_1, X_2, X_3, \dots, X_n) + \epsilon$$

Donde Y es la variable dependiente relacionada a la calidad del pagador (bueno o malo),  $X_i$  son las variables independientes (p.e. nivel de ingresos, edad, etc.) y  $\epsilon$  es el término de error (la parte aleatoria). F es la función – incluido los parámetros – que describe el comportamiento de Y a partir de las  $X_i$ .

Por una lado, las técnicas paramétricas (p.e. análisis discriminante, modelo logit) requieren la especificación a priori de una función (o modelo) que relacione las variables independientes con la variable dependiente que puede ser conocida (fundamentada en teoría) o asumida. Estas técnicas utilizan observaciones de Y y  $X_i$  para estimar los parámetros de la función, que luego se utilizarán para en la predicción de riesgo de nuevos solicitantes. De ese modo, estas técnicas han sido ejemplificadas extensamente en la literatura del credit scoring (e.g. Banasik, Crook, & Thomas, 2001; Greene, 1998). Sin embargo, una desventaja de las técnicas paramétricas es su estructura rígida, ya que en la medida que la función matemática sea inamovible, sólo deja libre la estimación de los parámetros. En ese sentido, estas técnicas pueden no resultar apropiadas para la representación de fenómenos que no siguen funciones

matemáticas conocidas. Por otro lado, las técnicas no paramétricas (p.e. redes neuronales artificiales), a diferencia de las paramétricas, no asumen a priori una función sino que la aproximan o calculan basados en las observaciones. De ese modo, se puede utilizar la misma función para fines predictivos en nuevos solicitantes. Asimismo, las redes neuronales han sido de uso extenso en la literatura del credit scoring (e.g. Cimpoeru, 2011; Khashman, 2011; Zhao et al., 2015).

## **Modelo Logit**

Este modelo permite predecir la probabilidad de que un individuo pertenezca a un grupo (o no) a partir de una serie de variables independientes que pueden ser tanto cuantitativas como cualitativas. Además, permite conocer el peso o efecto relativo de las variables explicativas en el modelo. Existen diversos tipos del modelo Logit. Uno de ellos es el dicotómico, que se utiliza cuando el número de alternativas es igual a dos y son excluyentes entre sí. Asimismo, está el de respuesta múltiple, que se utiliza cuando el número de alternativas a modelar es superior a dos.

El modelo Logit está definido por la siguiente función, siendo “ $X_i$ ” las variables independientes e “ $Y$ ” la variable dependiente:

$$Prob(Y = 1) = \frac{1}{1 + e^{-(\alpha + \beta_k X_{ki})}}$$

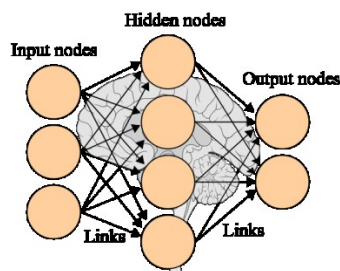
Esto debe leerse como: la probabilidad de pertenecer al grupo codificado como 1 (p.e. cliente que paga) está dada por la función matemática mostrada, dado un conjunto de variables  $X_i$ . El modelo logit tiene algunas ventajas sobre otros modelos paramétricos en el contexto financiero. Por ejemplo, respecto del análisis discriminante, el modelo logit no asume restricciones sobre la normalidad de la distribución de variables; pero en problemas financieros, utilizamos variables cualitativas que trasgreden dicho supuesto (Press & Wilson, 1978). Para la aplicación del modelo con fines predictivos se siguen los pasos detallados en el anexo 1.

## Redes neuronales artificiales

Una red neuronal se asemeja al cerebro humano en dos principales aspectos: (1) por el conocimiento adquirido del ambiente que se da través de un proceso de aprendizaje y (2) por los pesos asociados a las interconexiones entre neuronas, conocidos como pesos sinápticos, los cuales son utilizados para almacenar el conocimiento adquirido.

Las redes neuronales por definición son aproximadores de funciones universales (Haykin & Network, 2004), los cuales aproximan una función no-lineal  $f: A \rightarrow B$ , donde A y B, son conjuntos de datos de entrada y salida. Asimismo, poseen capas ocultas entre las capas de entrada y salida. Cada capa está constituida por neuronas o unidades de procesamiento (Figura N° 1).

Figura N°1. Red neuronal multi-level



Las neuronas realizan operaciones simples, transmitiendo sus resultados a los procesadores vecinos. La habilidad de la red neuronal en realizar relaciones no lineales entre sus entradas y salidas las convierte en técnicas útiles para el reconocimiento de patrones y en el modelaje de sistemas complejos (Bishop, 1995).

Estas redes pueden diferenciarse por su topología: redes feedforward y redes recurrentes. En las redes feedforward, todas las señales van desde la capa de entrada hacia la salida sin

existir ciclos ni conexiones entre neuronas de la misma capa de red neuronal. En las redes recurrentes, sí existen estas conexiones.

Otra tipología es aquella que distingue entre redes de aprendizaje supervisado y redes de aprendizaje no supervisado. En el tipo supervisado, la red utiliza un conjunto de observaciones donde para cada conjunto de variables independientes se conoce el valor de la variable dependiente. El algoritmo de aprendizaje consiste en la minimización del error en función de los datos de entrada y salida conocidos. En el tipo no supervisado, la red utiliza solamente un conjunto de observaciones de entrada. La red debe auto organizarse (o auto enseñarse) dependiendo de algún tipo de estructura existente en el conjunto de datos de entrada.

La red neuronal utilizada en este trabajo es Multilayer Perceptron que es del tipo Feedforward y con algoritmo de aprendizaje supervisado. Esta técnica puede resolver problemas diversos y con alto grado de complejidad (incluyendo la clasificación credit-scoring).

En este tipo de red, a partir de las entradas, la red realiza su procesamiento. La salida obtenida es comparada con la salida esperada. A partir del error obtenido, se aplica un proceso de ajuste de pesos, buscando minimizar el error.

La estructura de cada unidad de procesamiento es representada por entradas  $x_i$ , salidas  $\hat{y}_i$ , pesos  $w_i$ , un umbral  $\theta_i$ , y una función diferenciable  $\varphi$ , como se muestra a continuación:

$$\hat{y} = \varphi \left( \sum_{i=1}^k w_i x_i - \theta \right) \quad (1)$$

\*Para mayor detalle de las propiedades de esta ecuación, consultar Funahashi (1989).

En la ecuación (1) los pesos son calculados por medio de la regla de aprendizaje llamada de back-propagation (Rumelhart, Hinton, & Williams, 1985), la cual básicamente consiste en aplicar una familia de métodos de optimización basados en gradiente para encontrar el valor

óptimo de los pesos en función de minimizar la norma del error entre la salida deseada ( $y \in B$ ) y la salida calculada por la red neuronal  $\hat{y}$ , i.e.,

$$\min \|y - \hat{y}\| \quad (2)$$

Para la aplicación del modelo con fines predictivos se siguen los pasos detallados en el Anexo 2.

Una ventaja relativa de estas técnicas es que pueden representar funciones matemáticas complejas no lineales, resultando más flexibles que las técnicas paramétricas. Esta flexibilidad ha demostrado, bajo ciertas condiciones, superioridad en su poder predictivo respecto de las técnicas paramétricas (Abdou et al., 2008; Altman, Marco, & Varetto, 1994).

### **Algoritmos genéticos**

Los algoritmos genéticos son básicamente métodos de optimización y búsqueda que toman su inspiración en los conceptos propuestos por Darwin en su teoría de la selección natural de las especies. De ese modo, los sistemas desarrollados que se basan en este principio se utilizan para la búsqueda de soluciones a problemas complejos que tienen un gran espacio de posibles soluciones (Vargas da Cruz et al., 2009).

La tarea de los algoritmos genéticos es buscar un espacio de hipótesis candidatas para identificar la “mejor hipótesis”. En este sentido, dicha hipótesis es aquella que logra optimizar una medida numérica predefinida para el problema en cuestión, llamada también *hipótesis fitness* (Mitchell, 1997). Por ejemplo, si el problema en cuestión fuera el de conseguir la mejor estrategia al jugar ajedrez, el fitness podría ser definido como el número de juegos ganados de un individuo contra otros individuos de una misma población.

Esto se puede lograr ya que se adopta el siguiente procedimiento: cada posible solución a un problema se codifica en una estructura llamada “cromosoma”, que está compuesta por cadenas binarias de caracteres. Estos cromosomas representan a los individuos que luego serán



evaluados y trabajados por los operadores para así seleccionar la mejor solución al problema presentado. Los mejores operadores conocidos de algoritmos genéticos son el de Reproducción, Crossover y Mutación (Vargas da Cruz et al., 2009).

Si bien existen diferentes implementaciones de los algoritmos genéticos, típicamente siguen una misma estructura. Primero, el algoritmo opera por iteraciones, actualizando el espacio de hipótesis candidatas llamadas población. Luego, en cada iteración todos los miembros de la población son evaluados según la función fitness antes definida. Seguidamente, una nueva población es generada probabilísticamente mediante la selección de los individuos más aptos de la población actual. La probabilidad de selección de un individuo en base de su fitness está dada por la siguiente ecuación (Mitchell, 1997):

$$\Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^p Fitness(h_j)}$$

Donde la función “Fitness( $h_i$ )” es el puntaje fitness obtenido de la evaluación de individuo  $h_i$ , dividido entre la sumatoria de los puntajes fitness de los demás individuos.

Por último, se les aplica el proceso de reproducción a algunos de los individuos antes seleccionados, es decir, se copia y selecciona un cromosoma con alta probabilidad de selección, en función a su puntaje fitness, de manera intacta para ser usados en la siguiente población. Mientras que a los cromosomas no copiados, se les aplica el operador de crossover y mutación. El operador de crossover produce dos nuevos descendientes de dos cromosomas de la población actual mediante el intercambio de las partes de los cromosomas padres. El operador de mutación intercambia, de manera estocástica, un bit de un cromosoma con el otro posible valor válido del alfabeto binario.

Por ejemplo, dado los siguientes individuos:

Individuo 1	1	0	0	1	1
Individuo 2	1	1	1	0	0

Al aplicar el crossover de un punto de corte en el tercer bit, se obtendría lo siguiente:

Descendiente 1	1	1	1	1	1
Descendiente 2	1	0	0	0	0

Por otro lado, si se aplicara el proceso de mutación al individuo 1, se obtendría lo siguiente:

Individuo 1	1	0	0	1	1
-------------	---	---	---	---	---

↓

Mutación 1	1	0	1	1	1
------------	---	---	---	---	---

En la aplicación, este método de mutación se asegura que los individuos de la población tengan características diversas y hace posible introducir información que no estaba presente en ninguno de los individuos. Sin embargo, se tiene que tener mucho cuidado con la probabilidad de mutación, pues puede alterar más de lo deseado a la población de estudio. Esta probabilidad es definida por el investigador y las necesidades del problema.

## Modelos híbridos

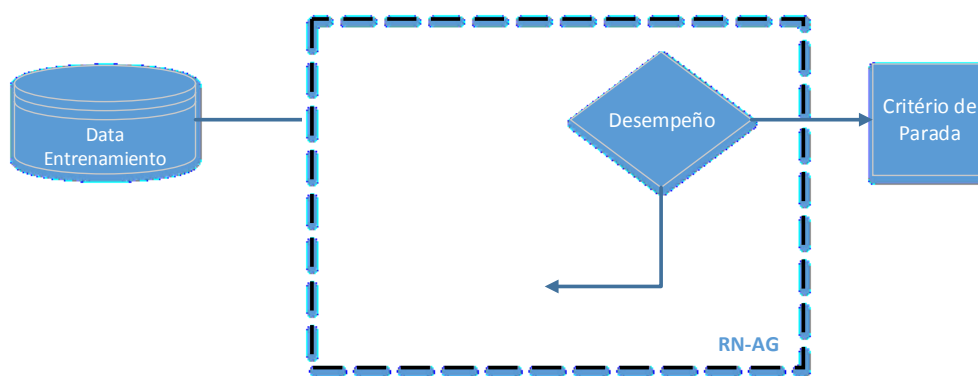
Cuando se utiliza las redes neuronales, se enfrenta el problema de establecer los óptimos parámetros de la red. Mayormente, la fijación de estos parámetros está basada en la experiencia del investigador y/o utilizando técnicas restrictivas como la estimación del número de neuronas que no necesariamente son los parámetros óptimos para obtener la mejor capacidad predictiva.

Una alternativa a esta estimación heurística es la utilización de modelos híbridos, el cual es consistente en utilizar una técnica para optimizar los parámetros o condiciones que luego la otra técnica la utilizará para resolver el problema de negocio en sí (en nuestro caso, credit scoring). Un referente cercano es el trabajo de Huang et al. (2007), quienes utilizan un modelo híbrido (combinando algoritmos genéticos y máquinas de soporte vectorial) para resolver el problema del credit scoring.

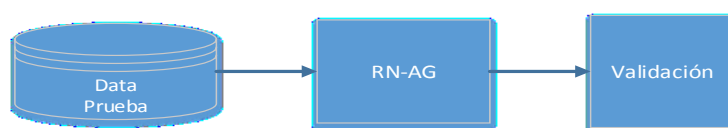
Uno de los problemas en el entrenamiento de redes neurales tipo Multi-Layer Perceptron (MLP) con algoritmo de entrenamiento *backpropagation* es la definición de sus parámetros. Pequeñas diferencias en estos parámetros pueden conducir a diferencias referentes al desempeño de la red neuronal. En esa medida, se expone un modelo híbrido que consiste en utilizar el algoritmo genético para optimizar los parámetros de la red.

Los parámetros de una red neuronal que pueden ser optimizados son el número de entrenamientos de la red (i.e. número de iteraciones con el cual los pesos de la red son ajustados); la tasa de aprendizaje que es el parámetro que posee gran influencia durante el procesamiento de entrenamiento de la red neuronal, de manera que una tasa de aprendizaje baja torna el aprendizaje lento, y un tasa de aprendizaje alta provoca oscilaciones en el entrenamiento de la red, lo que dificulta la convergencia en el proceso de aprendizaje. Otro parámetro de la red neuronal es el “momentum” que tiene por objetivo aumentar la velocidad de entrenamiento de la red neuronal. Un criterio importante para el entrenamiento de la red neuronal es el número de ciclos o iteraciones en el cual el conjunto de entrenamiento es presentado a la red neuronal. Un número excesivo de ciclos puede llevar a la red neuronal a la pérdida de generalización u “overfitting”, mientras que un pequeño número de ciclos la red neuronal no podrá aproximar la función objetiva, teniendo como consecuencia un bajo desempeño o “underfitting”. Para mayor información de diversos parámetros de la red neuronal ver Haykin and Network (2004).

**Figura N° 2. Modelo Híbrido – Etapa 1**



La Figura N° 2, presenta un diagrama de bloques del modelo híbrido propuesto. La data de entrenamiento es usada para ajustar los pesos sinápticos de la red neuronal por medio del algoritmo backpropagation. En cada iteración el algoritmo genético se evalúa el desempeño de la red neuronal y se optimiza los parámetros de momentum, la tasa de aprendizaje y el número de iteraciones de la red neuronal. Esto se realiza cíclicamente hasta que el desempeño se mantenga estable.



**Figura 3. Modelo Híbrido – Etapa 2**

Como muestra la Figura N° 3, una vez optimizados los parámetros de la red, es utilizada una data de prueba para validar el modelo de la red neuronal optimizada.

### 3. Método

Para ilustrar comparativamente la resolución del problema de credit scoring, se utilizará el historial de créditos de una entidad financiera latinoamericana. Las técnicas de análisis serán las desarrolladas en las secciones anteriores: logit, redes neuronales, e híbrida (red neuronal y algoritmo genético). La base de datos tiene 10 atributos (Tabla N° 1).

**Tabla N° 1. Atributos de la tabla clientes**

Variable	Concepto	Tipo
ESCT	Estado civil	Categorica
NDEP	Número de dependientes	Numérica
RENDA	Renta familiar	Numérica
VBEM	Valor del bien adquirido	Numérica
NPARC	Número de cuotas	Numérica
VPARC	Valor de la cuota	Numérica
TEL	Si tiene teléfono	Categorica
IDADE	Edad del cliente	Numérica
RESMS	Tiempo de residencia (meses)	Numérica
ENTRADA	Valor de entrada	Numérica
CLASE	Clasificación de pago (1 Pago / 0 No pago)	Categorica

El atributo clase (variable dependiente) indica si el cliente pagó o no pago su crédito. Los otros atributos se consideran variables independientes. La base de datos se ha dividido en una muestra de entrenamiento y otra de prueba. La Tabla N° 2 consigna la cantidad de observaciones por cada muestra.

**Tabla N° 2. Muestras**

	<b>Entrenamiento 1</b>	<b>Prueba 1</b>
Observaciones	1500	577

## **4. Resultados**

### **Resultados modelo logit**

Con esa estructura de datos y configuración de la muestra, los datos se procesaron con el software Rapid Miner 7.1. En la fase de entrenamiento, se ejecutó inicialmente el modelo con las diez variables independientes que se disponían en la base de datos. Los resultados (Anexo N° 3) muestran que algunas de estas variables incrementan la probabilidad de pago (p.e. IDADE); mientras otras reducen esa probabilidad (p.e. RESMS).

Los resultados muestran (Tabla N° 3) que, en el grupo de entrenamiento, la capacidad predictiva (clasificación correcta) se sitúa en 72.8%. La siguiente ecuación expresa la exactitud de la clasificación, la cual será usada en todo el artículo:

$$Clasificación\ correcta = \frac{N^o vp + N^o vn}{N^o vp + N^o vn + N^o fp + N^o fn}$$

Donde *vp* son los verdaderos positivos, *vn* son verdaderos negativos, *fp* son los falsos positivos y *fn* son los falsos negativos.

**Tabla N° 3. Modelo 10 variables - Fase de entrenamiento - Capacidad de predicción**

<b>Concepto</b>	<b>Train</b>
Clasificación correcta (%)	72.80%
Clasificación incorrecta (%)	27.20%

Luego, procedemos a ejecutar el modelo con las muestras de prueba manteniendo los coeficientes resultantes de la fase de entrenamiento. Los resultados presentados en la Tabla N° 4 indican que la capacidad predictiva de esta fase es ligeramente menor a la fase de entrenamiento (promedio 72.8% vs 70.88%).

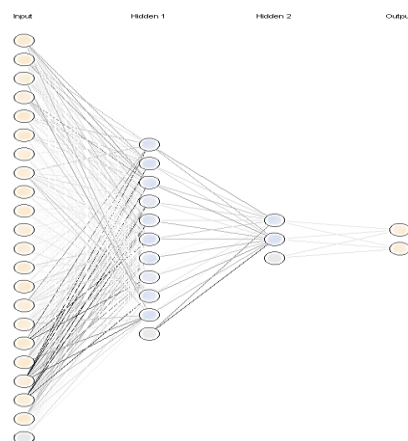
**Tabla N° 4. Modelo 10 variables - Fase de prueba - Capacidad de predicción**

Concepto	Prueba
Clasificación correcta (%)	70.88%
Clasificación incorrecta (%)	29.12%

## Resultados de la red neuronal

Con la estructura de datos y muestras señaladas en el numeral anterior, los datos se procesaron con el software Rapid Miner 7.1.

En la fase de entrenamiento, se ejecutó inicialmente la red neuronal con las diez variables independientes que se disponían en la base de datos (Figura N° 4). La red neuronal es de tipo Multi-Layer Perceptron (MLP), con algoritmo de aprendizaje Backpropagation. La red posee dos capas escondidas: la primera capa tiene 10 procesadores (neuronas); y la segunda capa tiene 2 procesadores (neuronas). Los parámetros de la red neuronal son: learning rate = 0.3; momentum = 0.2; y el número de iteraciones en el entrenamiento = 100.



**Figura N° 4. Red neuronal**

Los resultados muestran (Tabla N° 5) que en el grupo de entrenamiento la capacidad predictiva (clasificación correcta) se sitúa en 78.33%.

**Tabla N° 5. Modelo 10 variables - Fase de entrenamiento - Capacidad de predicción**

Concepto	Train 1
Clasificación correcta en %	78.33%
Clasificación Incorrecta en %	21.67%

Luego procedemos a ejecutar la red neuronal con las muestras de prueba manteniendo los coeficientes resultantes de la fase de entrenamiento. Los resultados (Tabla N° 6) indican que la capacidad predictiva de esta fase es similar a la fase de entrenamiento (75.22% vs 78.33%).

**Tabla N° 6. Modelo 10 variables - Fase de prueba - Capacidad de predicción**

Concepto	Test 1
Clasificación correcta en %	75.22%
Clasificación Incorrecta en %	24.78%

## **Resultados de la red neuronal optimizado con algoritmo genético**

La red neuronal fue entrenada con las mismas variables utilizadas anteriormente, las cuales se disponían en la base de datos. Los parámetros optimizados por el algoritmo genético, son learning rate = 0.386, momentum = 0.198 y el número de iteración en el entrenamiento = 624. Los resultados muestran que (Tabla N° 7), en el grupo de entrenamiento, la capacidad predictiva (clasificación correcta) se sitúa en 82.67%.

**Tabla N° 7. Modelo 10 variables - Fase de entrenamiento - Capacidad de predicción**

Concepto	Train 1
Clasificación correcta en %	82.67%
Clasificación Incorrecta en %	17.33%

Luego, procedemos a ejecutar el modelo con las muestras de prueba manteniendo los coeficientes resultantes de la fase de entrenamiento. Los resultados (Tabla N° 8) indican que la

capacidad predictiva de esta fase es menor a la fase de entrenamiento (76.95% vs 82.67%). Esto es de esperar debido a que el conjunto de prueba (test) no fue presentado antes a la red neuronal.

**Tabla N° 8. Modelo 10 variables - Fase de prueba - Capacidad de predicción**

Concepto	Test 1
Clasificación correcta en %	76.95%
Clasificación Incorrecta en %	23.05%

## 5. Discusión

Los resultados reflejan que la capacidad predictiva de la red neuronal es mayor que la del modelo logit tanto en la fase de entrenamiento ( $\Delta 5.53\%$ ; 78.33% vs 72.8%) como en la fase de prueba ( $\Delta 4.34\%$ ; 75.22% vs 70.88%). Este resultado es coherente con trabajos previos (e.g. Bonilla, Olmeda, & Puertas, 2003). Posiblemente, la predicción del pago de un crédito en esta entidad financiera tenga un comportamiento más complejo que el asumido por el modelo Logit. Por otro lado, los resultados reflejan que la capacidad predictiva del modelo híbrido es mayor que la red neuronal tanto en la fase de entrenamiento ( $\Delta 4.34\%$ ; 82.67% vs 78.33%) como en la fase de prueba ( $\Delta 1.73\%$ ; 76.95% vs 75.22%). Este resultado es coherente con el trabajo de Huang et al. (2007) quienes encuentran también una superioridad de un modelo híbrido (máquina de soporte vectorial optimizado con algoritmo genético) respecto de un modelo puro.

## 6. Conclusiones

Las entidades financieras pueden recurrir a distintas técnicas para predecir el pago (o no) de un solicitante. Por lo tanto, un aspecto relevante es la capacidad predictiva de estas técnicas. Los resultados de este estudio nos muestran, por un lado, que las redes neuronales consistentemente son superiores al modelo logit; y por otro lado, que la propuesta de un modelo híbrido resulta ligeramente superior al modelo puro (sólo red neuronal). Si bien la mejora del algoritmo híbrido no es alta, sí resulta útil en la medida en que, debido a las grandes cantidades de créditos que se someten a evaluación en las entidades financieras, una mejora mínima en la



capacidad predictiva puede resultar en ingentes cantidades de dinero ahorrados para las organizaciones financieras.

## 7. Referencias

- Abdou, H., & Pointon, J. (2011). Credit scoring, statistical techniques and evaluation criteria: A review of the literature. . *Intelligent Systems in Accounting, Finance and Management*, 18(2-3), 59-88. doi: 10.1002/isaf.325
- Abdou, H., Pointon, J., & El-Masry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems with Applications*, 35(3), 1275-1292.
- Altman, E. I., Marco, G., & Varetto, F. (1994). Corporate distress diagnosis: Comparisons using linear discriminant analysis and neural networks (the Italian experience). *Journal of banking & finance*, 18(3), 505-529.
- Anagnostopoulos, Y., & Abedi, M. (2016). Risk Pricing in Emerging Economies: Credit Scoring and Private Banking in Iran. *International Journal of Finance & Banking Studies (2147-4486)*, 5(1), 51-72.
- Banasik, J., Crook, J., & Thomas, L. (2001). Scoring by usage. *Journal of the Operational Research Society*, 997-1006.
- Basilea, C. D. S. B. D. (2004). Convergencia internacional de medidas y normas de capital. *Banco de Pagos Internacionales, junio*.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*: Oxford university press.
- Bonilla, M., Olmeda, I., & Puertas, R. (2003). Modelos paramétricos y no paramétricos en problemas de credit scoring. *Spanish Journal of Finance and Accounting/Revista Española de Financiación y Contabilidad*, 32(118), 833-869.
- Cimpoeru, S. S. (2011). Neural networks and their applications in credit risk assesment. Evidence from the romanian market. [Article]. *Technological and Economic Development of Economy*, 17(3), 519-534. doi: 10.3846/20294913.2011.606339
- Doumpos, M., & Zopounidis, C. (2014). Credit Scoring *Multicriteria Analysis in Finance* (pp. 43-59): Springer.
- Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3), 183-192.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning*, 3(2), 95-99.
- Greene, W. (1998). Sample selection in credit-scoring models. *Japan and the world economy*, 10(3), 299-316.
- Haykin, S., & Network, N. (2004). A comprehensive foundation. *Neural Networks*, 2(2004).
- Hsieh, N.-C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications*, 28(4), 655-665.
- Huang, C.-L., Chen, M.-C., & Wang, C.-J. (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33(4), 847-856.
- Khashman, A. (2010). Neural networks for credit risk evaluation: Investigation of different neural models and learning schemes. *Expert Systems with Applications*, 37(9), 6233-6239.
- Khashman, A. (2011). Credit risk evaluation using neural networks: Emotional versus conventional models. *Applied Soft Computing*, 11(8), 5477-5484.
- Laha, A. (2007). Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. *Advanced Engineering Informatics*, 21(3), 281-291.
- Lee, T.-S., & Chen, I.-F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications*, 28(4), 743-752.
- Lee, T.-S., Chiu, C.-C., Lu, C.-J., & Chen, I.-F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications*, 23(3), 245-254.
- Mitchell, T. M. (1997). *Machine learning*. MA: McGraw-Hill Boston.

- Press, S. J., & Wilson, S. (1978). Choosing between logistic regression and discriminant analysis. *Journal of the American Statistical Association*, 73(364), 699-705.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation: DTIC Document.
- Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management Science*, 38(7), 926-947.
- Vargas da Cruz, A., Barbosa, C. H., Lazo, J. G., Figueiredo, K., Almeida, L. F., Pacheco, M. A. C., . . . Valdivia, Y. J. T. (2009). Decision Support Methods *Intelligent Systems in Oil Field Development under Uncertainty* (pp. 23-96): Springer.
- West, D. (2000). Neural network credit scoring models. *Computers & Operations Research*, 27(11), 1131-1152.
- Zhao, Z., Xu, S., Kang, B. H., Kabir, M. M. J., Liu, Y., & Wasinger, R. (2015). Investigation and improvement of multi-layer perceptron neural networks for credit scoring. *Expert Systems with Applications*, 42(7), 3508-3516.

## **Anexo 1 – Pasos para ejecutar modelo Logit**

1. Se divide aleatoriamente las observaciones disponibles en dos muestras (de entrenamiento y de prueba).
2. Se ejecuta el modelo logit con la muestra de entrenamiento. Esta fase permitirá la estimación de los parámetros de la función logit.
3. Se revisa la capacidad predictiva del modelo (número de aciertos respecto del total de observaciones).
4. Se ejecuta la predicción, con los parámetros estimados en la fase de entrenamiento, utilizando la muestra de prueba.
5. Se revisa la capacidad predictiva del modelo (con los datos de prueba).

## **Anexo 2 – Pasos para ejecutar la redes neuronales**

6. Se divide aleatoriamente las observaciones disponibles en dos muestras (de entrenamiento y de prueba).
7. Se ejecuta el modelo con la muestra de entrenamiento. Esta fase permitirá la estimación de los parámetros de la red.
8. Se revisa la capacidad predictiva del modelo (número de aciertos respecto del total de observaciones).
9. Se ejecuta la predicción, con los parámetros estimados en la fase de entrenamiento, utilizando la muestra de prueba.
10. Se revisa la capacidad predictiva del modelo.

**Anexo N° 3. Modelo 10 variables - Fase de entrenamiento - coeficientes**

<b>Attribute</b>	<b>Weight</b>
ESTC = 1	-0.303526
ESTC = 0	0.2329103
ESTC = 2	-0.055487
ESTC = 3	-0.030253
NDEP = 0	-0.010535
NDEP = 2	0.0307913
NDEP = 1	0.0759148
NDEP = 3	0.176871
NDEP = 4	-0.010982
NDEP = 6	0.119329
NDEP = 5	0.1074953
NDEP = 7	0.1029291
TEL = 0	0.0096206
TEL = 1	-0.062175
RENDA	0.0787017
VBEM	-0.147739
NPARC	-0.43795
VPARC	-0.233612
IDADE	1.2895122
RESMS	-0.829242
ENTRADA	0.1467641