

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ciencias Sociales y Humanísticas



**“MODELO ECONOMÉTRICO DE REDES NEURONALES APLICADO A
INVESTIGACIÓN DE CRÉDITO”**

TRABAJO DE TITULACIÓN

Previa la obtención del Título de:

MAGISTER EN ECONOMÍA Y DIRECCIÓN DE EMPRESAS

Presentado por:

FÉLIX MIGUEL CARRERA BURI

Guayaquil – Ecuador

2015

AGRADECIMIENTOS

A Dios por haberme privilegiado con el don del conocimiento y sabiduría.

A mis padres Félix y Victorita que siempre me apoyaron en mi vida profesional.

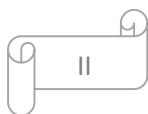
A mi Tío Ricardo Gastón por ser siempre mi gran amigo, por haberme guiado en mis momentos más confusos, por estar siempre en las buenas y malas.

A mi esposa Jovanna por estar cada día a mi lado enseñándome a ser una mejor persona para este mundo.

A mi gran amigo, hermano y compadre Juan Carlos Buenaño por haberme ayudado a entender la teoría neuronal.

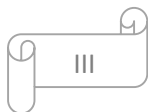
A mi maestro MSc. Gaudencio Zurita Herrera, por el conocimiento estadístico transmitido.

A la MSc. María Elena Romero por haberme ayudado a culminar otra vez otra etapa de mi vida.



DEDICATORIA

Para Milita, mi abuelita... Te amo con todas mis fuerzas, gracias por haberme dedicado tu vida.



TRIBUNAL DE GRADUACIÓN

MSc. Cristina Yoong Párraga
Presidente del Tribunal de Trabajo de titulación

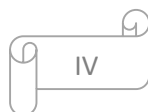
MSc. María Elena Romero
Director del Proyecto

MSc. Washington Macías

Revisor 1

MSc. María Fernanda Márquez

Revisor 2

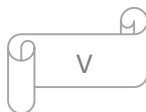


DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este Trabajo de Titulación, corresponde exclusivamente al autor, y al patrimonio intelectual de la misma ESCUELA

SUPERIOR POLITÉCNICA DEL LITORAL”

Félix Miguel Carrera Buri



ÍNDICE GENERAL

AGRADECIMIENTO.....	II
DEDICATORIA.....	III
TRIBUNAL DE GRADUACIÓN.....	IV
DECLARACIÓN EXPRESA.....	V
ÍNDICE GENERAL.....	VI
RESUMEN.....	VIII
ABSTRACT.....	X
ÍNDICE DE GRÁFICOS.....	XI
ÍNDICE DE TABLAS.....	XIII
CAPÍTULO 1.....	1
1.1 RESEÑA HISTÓRICA DE LOS MODELOS DE REDES NEURONALES.....	1
1.2 PROBLEMAS Y OPORTUNIDADES.....	9
1.3 JUSTIFICACIÓN DEL ESTUDIO.....	11
1.4 CARACTERÍSTICAS DEL MODELO A APLICAR.....	12
1.5 METODOLOGÍA.....	16
1.6 OBJETIVOS.....	19
CAPÍTULO 2.....	20
2.1 PLATAFORMA A USAR.....	20
2.2 IDENTIFICACIÓN DEL TIPO DE MODELO A USAR.....	21
2.3 VARIABLES.....	22
2.4 OPERACIONALIZACIÓN DE VARIABLES.....	24
2.5 ALGORITMO A USAR.....	26
CAPÍTULO 3.....	30
3.1 PREPARACIÓN DE LOS DATOS.....	30
3.2 ADMINISTRACIÓN DEL CONJUNTO DE DATOS.....	32

3.3	ENTRENANDO LA RED NEURONAL.....	35
3.4	PROBANDO LA RED NEURONAL.....	40
3.5	PREDICIENDO LA RED NEURONAL.....	43
	CAPÍTULO 4.....	47
4.1	IDENTIFICACIÓN DEL REPORTE.....	47
4.2	ANÁLISIS DEL REPORTE.....	49
4.3	RESULTADO DEL REPORTE.....	54
	CONCLUSIONES.....	56
	RECOMENDACIONES.....	57
	REFERENCIAS.....	58
	ANEXOS.....	59

RESUMEN

Hoy en día es de conocimiento cuán importante es el querer adquirir un bien o servicio que nos mejore la calidad de vida pero sin tener en cuenta que aquel bien y servicio se puede escapar de nuestro presupuesto sin ningún problema (Armstrong&Kotler, 2008). Para aquello lo que se busca es una forma de financiamiento y lo más común es un préstamo, este préstamo puede ser de tipo: Crédito para una hipotecario, automóvil, emprendimiento, estudiantil, etc. Para ese crédito se necesita calificar para obtenerlo cumpliendo los requisitos que solicita la entidad financiera a la cual se vaya a dirigir, por lo general los requisitos son garantías de pago en las que se toman en cuenta el historial de gastos, cuentas bancarias, historial laboral, garantes, etc. El problema es que no todos los que aplican califican para el crédito quedando fuera del préstamo deseado por la falta de requisitos por ejemplo: una persona de 20 años de edad que recién empieza su vida laboral no posee historial laboral necesario para la aplicación y es por eso que se lo descalifica. Los problemas ya mencionados crean una barrera que al ciudadano que quiere aplicar a crédito lo desestimule causando que la necesidad ya no sea prioritaria y pase a segundo plano dejando de ser un posible inversionista activo y por ende hasta causando la pérdida de la empatía hacia la institución financiera. Pero, ¿Es posible que esa misma persona fuera capaz de realizar los pagos necesarios sin poseer todos los requisitos de garantías? Pues eso se podría lograr investigando el comportamiento de esa persona para con sus finanzas, comportamiento en sus gastos, comportamiento en sus ahorros, comportamiento en sus inversiones y sacar de ese comportamiento las garantías necesarias para realizar el crédito deseado. Lo anterior se podría transformar a variables categóricas para crear un modelo que permita el análisis de esas variables, que permita encontrar ese patrón que se repite con una frecuencia estándar o variable, esa repetición de patrones es una red neuronal que permitirá predecir si esa persona es merecedora del crédito a obtener. Este tipo de modelo investigativo no asegura que todo aquel que se someta a su análisis vaya a salir calificado pero si disminuye el margen de error de la calificación el cual persiste en modelos estáticos. Con esto se podría ser testigos futuros de encontrar ciudadanos de edades entre 20 a 25 años empezando una vida laboral en que sean partícipes de un crédito hipotecario, automotriz, estudiantil o cualquiera que sea su fin. El problema está identificado y es por esto que es importante que se realice esta

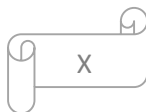
propuesta de investigación para demostrar con evidencia irrefutable cuan estático está el sistema financiero ecuatoriano y la cantidad de confianza hacia el cliente que se pierde.

Palabras claves: Modelo, neuronal, crédito, variables.

ABSTRACT

Today we know how important it is to want to purchase a good or service that we improve the quality of life but regardless of that good or service that can escape our budget without any problem (Armstrong & Kotler, 2008). For that what is sought is a form of financing and the most common is a credit, this credit can be of type: Mortgage Lending, Car credit, enterprise credit, student loans, etc. For that credit is needed to qualify to get it meets the requirements requesting the financial institution to which he will direct, usually the requirements are payment guarantees in the history of expenses, bank accounts, employment history are taken into account , guarantors, etc. The problem is that not all who apply qualify for the credit was being desired by the lack of such loan requirements: a 20-year-old who has just started his working life has no work history required for the application and is that disqualifies it. The aforementioned problems create a barrier that the citizen who wants to apply for credit it discourages causing the need is no longer a priority and pass background ceasing to be an active investor and may thus causing loss to empathy for the institution financial. But if that same person was able to make the necessary payments without having all collateral requirements?. Well, that could be done to investigate the behavior of that person with their finances, their spending behavior, and behavior in their savings, investment behavior and that behavior out of the necessary guarantees for the desired credit. This could transform categorical variables to create a model that allows the analysis of these variables, allowing find this pattern repeated with a standard or variable frequency, the repetition of patterns is a neural network that will predict if that person is credit worthy to obtain. This type of investigative model does not ensure that anyone who is subjected to analysis go out qualified but if the margin of error decreases qualification which persists in static models. With this could be future witnesses to find citizens aged 20 to 25 years starting a working life in which they are participants in a mortgage, auto, student or whatever end. The problem is identified and is why it is important that the proposed research is conducted with irrefutable evidence to demonstrate how static our financial system is and the amount of confidence to the client is lost.

Keywords: Model, neuronal, credit variables.



ÍNDICE DE FIGURAS

Figura 1.1 El perceptrón.....	2
Figura 1.2 La neurona.....	2
Figura 1.3 Alimentación directa de una red neuronal, totalmente conectada.....	4
Figura 1.4 Representación de un nodo de conexión.....	5
Figura 1.5 Entrenamiento en línea.....	7
Figura 2.1 Variables a evaluar.....	23
Figura 3.1 administración de conjunto de datos.....	31
Figura 3.2 selección de conjunto de datos.....	32
Figura 3.3 selección del rango del conjunto de datos.....	33
Figura 3.4 selección validación del rango del conjunto de datos.....	34
Figura 3.5 selección de entrenamiento del conjunto de datos.....	35
Figura 3.6 selección de las opciones de entrenamiento del conjunto de datos – Pestaña entrenar.....	35
Figura 3.7 selección de las opciones de entrenamiento del conjunto de datos – Pestaña configuración de red.....	37
Figura 3.8 selección de las opciones de entrenamiento del conjunto de datos – Pestaña Tiempo de ejecución.....	37
Figura 3.9 selección de las opciones de entrenamiento del conjunto de datos.....	38
Figura 3.10 aprendizaje de la red neuronal - Etapa entrenamiento.....	38
Figura 3.11 selección de prueba de la red neuronal.....	39
Figura 3.12 selección de prueba de la red neuronal - Opciones.....	40
Figura 3.13 selección de prueba de la red neuronal – Carga de datos de prueba.....	40
Figura 3.14 pre-visualización de prueba de la red neuronal.....	41
Figura 3.15 selección de predicción de la red neuronal.....	42
Figura 3.16 selección de predicción de la red neuronal - carga de datos.....	43

Figura 3.17 pre-visualización de la predicción de la red neuronal.....	43
Figura 3.18 resultado de la predicción del modelo econométrico.....	44
Figura 4.1 reporte del entrenamiento de la red neuronal.....	45
Figura 4.2 reporte de las pruebas de la red neuronal.....	46
4.3 Histogramas de probabilidad de categorías incorrectas tanto en entrenamientos como en pruebas.....	50

ÍNDICE DE TABLAS

Tabla 4.1: Reporte de NeuralTools al ejecutar al modelo.....	47
Tabla 4.2: Reporte de NeuralTools , búsqueda de mejor red.....	48
Tabla 4.3: Reporte de NeuralTools , matriz de clasificación para los casos de entrenamiento.....	49
Tabla 4.4: Reporte de NeuralTools , matriz de clasificación para los casos de prueba.....	50
Tabla 4.5: Reporte de NeuralTools , Informe de Entrenamiento-Prueba-Predicción de "Red entrenada en Conjunto de datos número 1"	51

CAPÍTULO 1

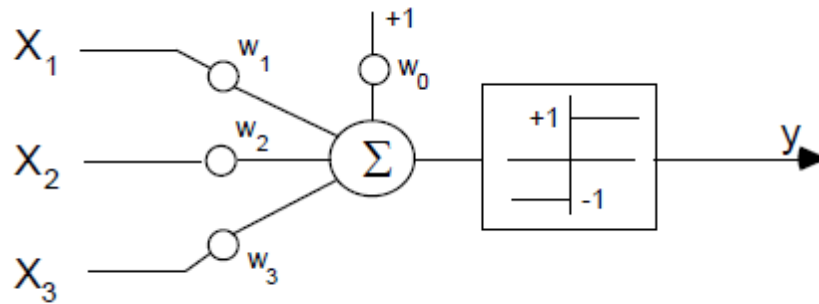
1.1 RESEÑA HISTÓRICA DE LOS MODELOS DE REDES NEURONALES.

La historia de las redes neuronales se remonta a la labor de tratar de modelar la neurona. El primer modelo (Armstrong&Kotler, 2008)de una neurona fue realizado por los fisiólogos, McCulloch y Pitts (1943). El modelo que crearon tenía dos entradas y una sola salida. McCulloch y Pitts observaron que una neurona no se activaría si sólo una de las entradas era activa. Los pesos (conectores de información, operadores lógicos) para cada entrada eran iguales, y la salida era binario (decisión de si o no). La neurona de McCulloch y Pitts se ha dado a conocer hoy como un circuito lógico (Mack&Lackes, 1998).

El perceptrón se desarrolló como el próximo modelo de la neurona por Rosenblatt (1958) como se ve en la figura 1.1. Rosenblatt, que era un fisiólogo, interconectó al azar los perceptrones para cambiar aleatoriamente los pesos a fin de lograr el "aprendizaje" (Mack&Lackes, 1998) .

El proceso electroquímico de una neurona funciona como una frecuencia de voltaje (Anderson y Rosenfeld, 1987). Las entradas a la neurona causa un reacción química tal que, cuando las sustancias químicas se acumulan a un cierto umbral, la neurona procesa la información. Como entradas superiores entran en la neurona, esta dispara a una frecuencia más alta, pero la magnitud de la salida de la neurona es lo mismo. La figura 1.2 es un modelo de una neurona. Una comparación visual de las figuras 1.1 y 1.2 se muestra los orígenes de la idea del perceptrón se remonta a la neurona. Externamente, un perceptrón parece asemejarse a la neurona con múltiples entradas y una sola salida. Sin embargo, esta similitud no hace realmente comenzar a modelar los complejos procesos electroquímicos que en realidad van en dentro de una neurona. El perceptrón es una representación matemática muy simple de la neurona (Mack&Lackes, 1998).

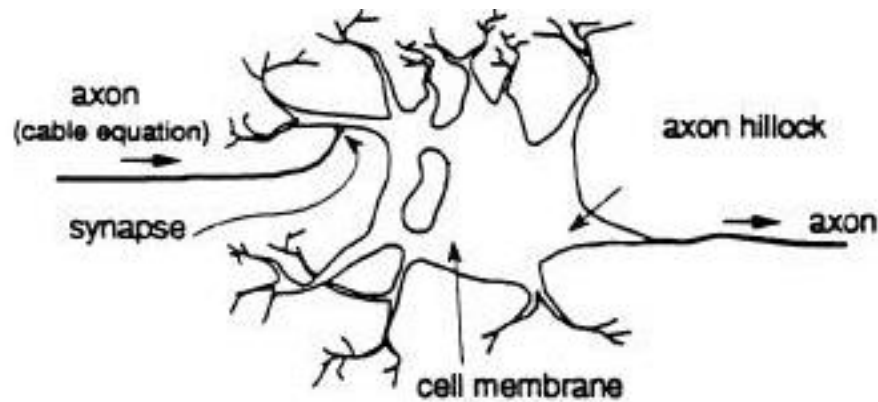
Figura 1.1.- El perceptrón.



Fuente: (Armstrong&Kotler, 2008).

Selfridge (1958) trajo la idea del espacio de peso al perceptrón. Rosenblatt ajustó los pesos en un método de ensayo y error. Selfridge ajustó los pesos para elegir al azar un vector de dirección. Si el rendimiento no lo hizo mejorar, los pesos fueron devueltos a sus valores anteriores, y un nuevo vector de dirección aleatorio fue elegido (Mack&Lackes, 1998).

Figura 1.2.- La neurona



Fuente: (Armstrong&Kotler, 2008).

Widrow y Hoff (1960) desarrollaron un método matemático para la adaptación de los pesos. Suponiendo que existía una respuesta deseada, un método de búsqueda de gradiente se puso en práctica, que se basa en minimizar el error al cuadrado. Este algoritmo más tarde se conocería como LMS, o media de mínimos cuadrados. LMS, y sus variaciones, se ha utilizado ampliamente en un sin número de aplicaciones, especialmente en los últimos años. Este método de búsqueda de gradiente proporcionó un método matemático para encontrar una respuesta que minimiza el error. El proceso de aprendizaje no es un proceso de ensayo y error. Aunque la era computacional disminuyó con el trabajo de Selfridge, el método LMS disminuyó la cantidad del tiempo de cómputo aún más, lo que hizo uso de perceptrones factibles (Mack&Lackes, 1998). A la altura de la red neuronal perceptrón o de investigación en la década de 1960, los periódicos estaban llenos de artículos que prometen los robots que podrían pensar. Pareció que perceptrones podrían solucionar cualquier problema. Un libro, Perceptrons (Minsky y Papert, 1969), trajo la investigación a un abrupto fin. El libro señala que perceptrones sólo podían resolver problemas linealmente separables. A es un perceptrón de un solo nodo. Perceptrons muestra que con el fin de resolver un problema n separable, se necesitan $n-1$ nodos. Un perceptrón podría entonces sólo resolver un problema $n-2$ separables, o un problema linealmente separable (Mack&Lackes, 1998).

Después Perceptrons se publicó, la investigación de redes neuronales se fue sin fondos, y permanecería así, hasta que se desarrolló un método para resolver n separable problemas. Werbos (1974) fue el primero en desarrollar el algoritmo de propagación inversa. Entonces fue redescubierto independientemente por Parker (1985) y por Rumelhart y McClelland (1986), de forma simultánea. Propagación inversa es una generalización del algoritmo LMS de Widrow-Hoff y permitió a los perceptrones ser entrenados en una configuración de multicapa, así una red neural nodo $n-1$ podría construirse y entrenarse. Los pesos se ajustaron y se basaron en el error entre la salida y algunos de salida deseada (respuesta esperada). Como su nombre indica, los pesos se ajustan hacia atrás a través de la red neuronal, comenzando con la capa de salida y trabajando a través de cada capa oculta hasta que se alcanza la capa de entrada. La parte de atrás algoritmo de propagación cambia el esquema del perceptrón mediante el uso de una función sigmoideal como la función de aplastamiento. Las versiones anteriores del perceptrón era utilizando una función signo. La ventaja de la función sigmoideal sobre el función signo es que

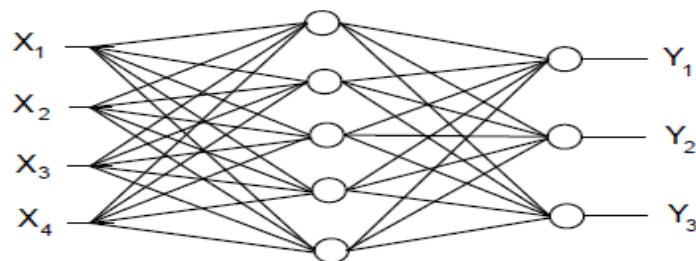
la función sigmoideal es diferenciable. Esto permite que el algoritmo de propagación inversa sirva para transferir la información a través del gradiente no lineal o función de aplastamiento, permitiendo que la red neural pueda converger a una respuesta lógica. Neurocomputing: Fundamentos de Investigación (Anderson y Rosenfeld, 1987) es una excelente fuente de la obra que se hizo antes de 1986. Se trata de una colección de documentos y ofrece una visión general interesante de los acontecimientos en el ámbito de las redes neuronales antes de 1986 (Mack&Lackes, 1998).

Aunque la edad de oro de la investigación de redes neuronales terminó hace 25 años, el descubrimiento de propagación inversa ha revitalizado la investigación que se realiza en esta área. La red neuronal de alimentación directa es la interconexión de perceptrones y es utilizado por la gran mayoría de los documentos revisados (Mack&Lackes, 1998).

La red neural de alimentación directa es una red de perceptrones con una función da aplastamiento diferenciable, por lo general es la función sigmoideal. La parte del algoritmo de propagación inversa ajusta los pesos basados en la idea de minimizar el error cuadrático. La función de aplastamiento diferenciable permite que el algoritmo de propagación inversa pueda ajustar los pesos a través de múltiples capas ocultas (Torra&Monte, 2013).

Al tener múltiples nodos en cada capa, los problemas de n-separables se pueden resolver, como el exclusivo-O, o el problema XOR, que no pudo ser resuelto sólo con el perceptrón. La figura 1.3 muestra una red neuronal totalmente conectada por alimentación directa; de entrada a salida, cada nodo está conectado a cada nodo de las capas adyacentes (Torra&Monte, 2013).

Figura 1.3.- Alimentación directa de una red neuronal, totalmente conectada.



Fuente: (Torra&Monte, 2013), alimentación directa de una red neuronal.

En la figura 1.4, los nodos individuales o perceptrones, son representativos de la neurona. La entrada para el nodo es la entrada a la red neural o, si el nodo es en una capa oculta o la capa de salida, la salida de una capa anterior. El nodo es la clave para el entrenamiento de la red neuronal. La propagación del nuevo algoritmo manda los cambios a los pesos a través de la red neuronal (cambiando los pesos de un nodo individual en un tiempo determinado).

Con cada iteración, se da la diferencia entre la salida de la red neural y la respuesta deseada es calculada. En el caso de una única salida, la salida de toda la red neural es la salida de un nodo individual cuyas entradas son las salidas de los nodos en la capa anterior. El algoritmo de propagación inversa es un LMS-como algoritmo para la actualización de las ponderaciones.

A continuación se muestra la derivación del algoritmo de propagación inversa, que trata de minimizar el error cuadrático (Rumelhart y McClelland, 1986) (Torra&Monte, 2013).

Las variables para la derivación de propagación inversa se definen como sigue:

\mathbf{X} es el vector de entrada del nodo;

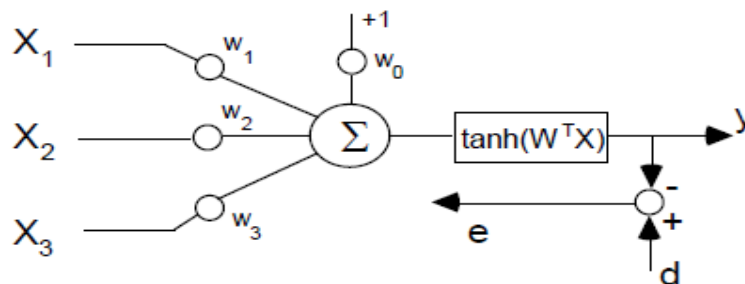
\mathbf{W} es el vector de pesos del nodo;

y es la salida del nodo;

\mathbf{d} es la respuesta deseada del nodo;

e es la diferencia entre la salida del nodo y la respuesta deseada.

Figura 1.4.- Representación de un nodo de conexión.



Fuente: (Torra&Monte, 2013), nodo de conexión.

En las aplicaciones de control es solo un área en la que se han aplicado las redes neuronales, pero el control tiene su propio conjunto de problemas a resolver en la aplicación de cualquier metodología. El principio principal detrás de los controles es cambiar el rendimiento de un sistema para ajustarse a un conjunto de especificaciones (Torra&Monte, 2013).

Este objetivo puede ser complicado por incertidumbres en el sistema, incluidas las no linealidades. La teoría de control tiene estado tratando de desarrollar metodologías para manejar cantidades cada vez mayores de incertidumbres.

Las redes neuronales se pueden aplicar cuando ningún conocimiento a priori existe en el sistema. Desafortunadamente, rara vez hay un modelo completo del sistema, pero a menudo, hay un modelo parcial del sistema disponibles (Torra&Monte, 2013).

Narendra y Parthasarathy (1990) iniciaron la actividad en el desarrollo adaptativo esquemas de control para plantas no lineales. Muchos de los papeles de Narendra se han ocupado con el control y la identificación de un sistema utilizando redes neuronales. Nguyen y Widrow (1990) trabajó en los sistemas de control de autoaprendizaje. Los papeles de Widrow tienen una larga historia de problemas de control e identificación que utilizan las redes neuronales (Torra&Monte, 2013).

Sus supuestos no han incluido un conocimiento y control de lazo abierto a priori solamente. El trabajo de Widrow nunca ha incluido la retroalimentación de circuito cerrado, con la excepción de su sugerencia para estabilizar un sistema inestable con retroalimentación, y a continuación, utilizar una red neuronal para alcanzar el rendimiento especificado. Narendra y Widrow haber hecho mucho trabajo de campo en el campo de la red neuronal basada control; la mayor parte de los documentos revisados en las siguientes secciones hacen referencia a la trabajo realizado por ellos. Las siguientes secciones revisan en detalle en los documentos específicos temas en el área de redes neuronales en aplicaciones de control (Torra&Monte, 2013).

Solo una cantidad limitada de información de cualquier sistema va a ser conocido. Por lo tanto, no es una buena metodología de diseño el escoger cualquier sistema al azar. La Integración de información de un sistema al control de una red neural debe de ser estudiada

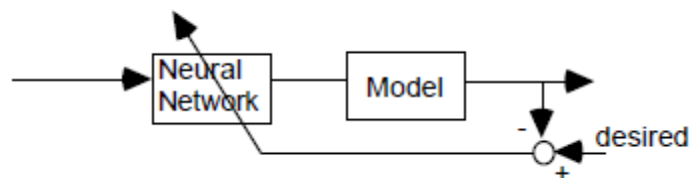
antes de realizarse ya que el uso de una información no estructurada para este tipo de metodología podría causar error en el modelo a aplicar, este tipo de advertencia ha sido sugerida en varios lugares por Selinsky y Guez (1989) y Linguini y Sakai (1989) entrenadores de la red neuronal fuera de línea con la dinámica de sistema conocidos antes de aplicar el controlador de red neuronal para el sistema final.

Por otra parte Joerding y Meador (1991) constreñidos por los pesos de la primera red neuronal fuera de línea, utilizó conocimiento de bases de datos a través de la modificación del algoritmo de entrenamiento para estabilizar esta metodología. A su vez Nordgren y Meckl (1993) incorporaron un conocimiento similar al de Joerding y Meador (1991) a través de una ruta de control paralelo a la red neuronal.

Selinsky y Guez (1989) y Iiguni y Sakai (1989) utilizaron el conocimiento de bases de datos de un sistema particular para entrenar la red neuronal fuera de línea, una práctica muy común en el control de la red. Los papeles de Selinsky y de Iiguni son ejemplos típicos de uso de un conocimiento de datos históricos. La idea básica es crear un modelo del sistema con tan tanto detalle como dispone y utilizar ese conocimiento de datos para entrenar la red neuronal, como se ve en la figura 1.5.

El problema con este método es que, si el modelo no es precisamente correcto, las no linealidades de la de redes neuronales (entrenamiento y prueba de la red neuronal), interactuando con las no linealidades del sistema real (predicción), no podrá funcionar como se espera (Torra&Monte, 2013).

Figura 1.5.- Entrenamiento en línea.



Fuente: (Torra&Monte, 2013), entrenamiento en línea de un conjunto de datos.

Una idea similar es utilizada por muchos otros investigadores, como Narendra y Parthasarathy (1990). Asumieron el acceso al sistema actual y crean un modelo de red neural del sistema. El modelo de red neuronal se utiliza para entrenar el controlador de red neural. Este método funciona casi tan bien como el uso del actual sistema de formación (Torra&Monte, 2013).

El primer problema es crear un buen modelo del sistema usando una red neural. Este método se basa en la red neural encontrar sus propias correlaciones entre las entradas y las salidas. El segundo problema es crear un controlador de red neural para controlar el modelo, que también se basa en la red neural para encontrar su correlación. Ninguno de los dos anteriores Joerding y Meador (1991) constreñidos los pesos de la red neuronal usando un conocimiento a priori en un algoritmo de entrenamiento modificado (Torra&Monte, 2013).

Ellos se dirigieron al problema de la incorporación de un conocimiento a priori acerca de una función de salida óptima en limitaciones específicas. Los dos enfoques generales son una Arquitectura Método de restricción y un método Peso restricción. Ambos asumen el conocimiento de la forma de la función de salida óptima, tales como monótona y concavidad. Una función monótona es uno cuya pendiente no cambia señal, y una función cóncava (convexa) tiene una pendiente que disminuye (aumenta) como el argumentos de la función aumentan. La salida deseada de la red neuronal es limitada a estos tipos de función (Torra&Monte, 2013).

Los dos métodos se utilizan para explotar la naturaleza matemática de la red neuronal de alimentación hacia adelante con una hiperbólica tangente función de aplastamiento. La tangente hiperbólica es monótona y cóncava; el signo de la tangente hiperbólica es el mismo que el signo de su argumento. El algoritmo de entrenamiento modificado consiste en el término de propagación hacia atrás, más la derivada de la función óptima. Es una idea interesante para codificar el a priori la información en la red neural. Estos métodos funcionan bien para el modelado del sistema. Sin embargo, no pueden usarse directamente en una aplicación de controlador (Torra&Monte, 2013).

1.2.PROBLEMAS Y OPORTUNIDADES DEL MODELO DE REDES NEURONALES.

Actualmente se sabe cuán importante es el querer adquirir un bien o servicio que nos mejore la calidad de vida pero sin tener en cuenta que aquel bien y servicio se puede escapar de nuestro presupuesto sin ningún problema (Armstrong&Kotler, 2008).

Para aquello lo que se busca es una forma de financiamiento y lo más común es un crédito, este crédito puede ser de tipo: Crédito hipotecario, crédito para automóvil, crédito para emprendimiento, crédito estudiantil, etc. Para ese crédito se necesita calificar para obtenerlo cumpliendo los requisitos que solicita la entidad financiera a la cual se vaya a dirigir, por lo general los requisitos son garantías de pago en las que se toman en cuenta el historial de gastos, cuentas bancarias, historial laboral, garantes, etc. El problema es que no todos los que aplican califican para el crédito quedando fuera del préstamo deseado por la falta de requisitos por ejemplo: una persona de 20 años de edad que recién empieza su vida laboral no posee historial laboral necesario para la aplicación y es por eso que se lo descalifica.

Los problemas ya mencionados crean una barrera que al ciudadano que quiere aplicar a crédito lo desestime causando que la necesidad ya no sea prioritaria y pase a segundo plano dejando de ser un posible inversionista activo y por ende hasta causando la pérdida de la empatía hacia la institución financiera. Pero si esa misma persona fuera capaz de realizar los pagos necesarios sin poseer todos los requisitos de garantías? Pues eso se puede lograr investigando el comportamiento de esa persona para con sus finanzas, comportamiento en sus gastos, comportamiento en sus ahorros, comportamiento en sus inversiones y sacar de ese comportamiento las garantías necesarias para realizar el crédito deseado.

Lo anterior se podría transformar a variables categóricas para crear un modelo que permita el análisis de esas variables, que permita encontrar ese patrón que se repite con una frecuencia estándar o variable, esa repetición de patrones es una red neuronal que permitirá predecir si esa persona es merecedora del crédito a obtener. Este tipo de modelo investigativo no asegura que todo aquel que se someta a su análisis vaya a salir calificado pero si disminuye

el margen de error de la calificación el cual persiste en modelos estáticos (Palisade, NeuralTools Testing version, 2013).

Con esto se podría ser testigos futuros de encontrar ciudadanos de edades entre 20 a 25 años empezando una vida laboral en que sean partícipes de un crédito hipotecario, automotriz, estudiantil o cualquiera que sea su fin. El problema está identificado y es por esto que es importante que se realice esta propuesta de investigación para demostrar con evidencia irrefutable cuan estáticos está nuestro sistema financiero y la cantidad de confianza hacia el cliente que se pierde. El modelo econométrico redes neuronales artificiales tiene muchas ventajas de oportunidad de espacio en el mercado financiero debido a que están basadas en la estructura del comportamiento del consumidor y por esa característica se distingue las siguientes ventajas a analizar.

1. **Aprendizaje:** El modelo econométrico redes neuronales artificiales tiene la destreza de aprender mediante una fase que se llama fase de aprendizaje. Esta consiste en proporcionar al modelo econométrico redes neuronales artificiales datos como entrada y a su vez se le indica cuál es el dato para la salida (respuesta) esperada.
2. **Auto organización:** El modelo econométrico redes neuronales artificiales crea su propia representación de la información en su interior, aliviando al investigador la programación del algoritmo ya que está en constante aprendizaje.
3. **Tolerancia a fallos:** Debido que el modelo econométrico redes neuronales artificiales almacena la información de forma redundante, ésta puede seguir procesando y respondiendo de manera aceptable aun si se afectara parcialmente.
4. **Flexibilidad:** el modelo econométrico redes neuronales artificiales puede manejar cambios no significativos en los datos de entrada, como el borrar un dato de una variable.
5. **Tiempo real:** La estructura del modelo econométrico redes neuronales artificiales es paralela, por lo cual si esto es efectuado con computadoras o en mecanismos electrónicos especiales, se pueden obtener respuestas en tiempo real.

Hay muchas buenas razones para el uso de este modelo econométrico redes neuronales artificiales, los avances en este ambiente aumentarán su notoriedad. Son excelentes como clasificadores/reconocedores de patrones – y pueden ser usadas donde las técnicas tradicionales no funcionan. Las redes neuronales pueden manejar excepciones y entradas de datos anormales, muy importante para sistemas que manejan un amplio rango de datos (Palisade, NeuralTools Webcasts, 2014).

1.3.JUSTIFICACIÓN DEL ESTUDIO.

El tema escogido es modelo econométrico de redes neuronales aplicado a investigación de crédito, este tema fue seleccionado por la naturaleza de beneficios que creará ante una sociedad que puede que carezca de requisitos tradicionales para la evaluación de créditos y prestaciones mediante un calificativo de aprobación y no aprobación.

Se deja en claro que para aplicar la teoría de redes neuronales, se debe tener un conocimiento de rango medio ha avanzado ya que lo que se quiere es que en la simulación se reduzca lo máximo que se pueda el margen de error y obtener la respuesta deseada ya que si no sirve el modelo predictivo se estaría dando paso a que los modelos estáticos sigan obteniendo su lugar en nuestro ambiente financiero haciendo de este un mercado limitado y limitante a su vez.

Este modelo lo que hará es que mediante la información histórica suministrada pueda aprender de aquellos datos e identificar el patrón de comportamiento entre las variables que permitan pronosticar con bajo margen de error y con un riesgo considerable, la calificación del cliente que aplica al crédito.

El tema es muy interesante ya que las redes neuronales trata de simular el comportamiento del cerebro humano, es decir conocer si el cliente al que se está evaluando será un cliente que beneficiará al negocio o simplemente un cliente que pueda presentar una amenaza para la institución financiera.

La tecnología está en constante cambio y este tipo de modelos poco a poco está dejando notar su presencia a lo largo del mundo laborar (financiero, médico, administrativo,

etc.), por eso es tema de investigación que puede aportar al conocimiento y desarrollo en el campo Ecuatoriano.

1.4.CARACTERÍSTICAS DEL MODELO A APLICAR.

Existen cuatro aspectos que caracterizan este modelo econométrico de red neuronal: su razonamiento, el componente de aprendizaje, tipo de agrupación realizada entre la información de entrada y salida, y la forma de representación de estas investigaciones.

1. **Redes Monocapa:** Se constituyen uniones laterales, cruzadas o autorrecurrentes entre las neuronas que corresponden a la única capa que establece la red. Se utilizan en tareas relacionadas con lo que se conoce como autoasociación; por ejemplo, para crear informaciones de entrada que se muestran a la red incompleta o deformada.

2. **Redes Multicapa:** Son aquellas que colocan grupos de neuronas concatenadas en varios pisos o capas. Una manera de diferenciar la capa a la que pertenece la neurona, consiste en observar en el origen de las señales que recepta a la entrada y el destino de la señal de salida. Según la clase de conexión, como se mostró con anticipación, se diferencian las redes feedforward, y las redes feedforward/feedback (Palisade, NeuralTools Webcasts, 2014).

3. **Razonamiento de las Redes Neuronales.** La estructura de las redes neuronales consta en la organización y disposición de las neuronas creando capas alejadas considerablemente de la entrada y salida de la red. En esta forma, los puntos esenciales de la red son: el número de pisos, número de neuronas por pisos, el impacto de conectividades y el tipo de conexiones de neurona a neurona (Palisade, NeuralTools Webcasts, 2014).

4. **Mecanismo de Aprendizaje.** El aprendizaje es el paso por el cual la red neuronal cambia sus parámetros en exclamación a una información de entrada. Las modificaciones que resultan durante el proceso de aprendizaje se minimizan a la desintegración, transformación y generación de conexiones entre las neuronas, la generación de una reciente conexión quiere decir que el peso de información de la misma neurona pasa a tener un valor diferente de cero, una conexión se desintegra en el momento que su peso se vuelve cero.

Hay evidencia para afirmar que el proceso de aprendizaje ha concluido (la red ha aprendido) cuando los valores de los pesos se mantienen estabilizados ($dw_{ij} / dt = 0$) (Palisade,

NeuralTools Webcasts, 2014). Un criterio para distinguir las normas de aprendizaje se resuelve en observar si la red puede aprender durante su actividad habitual, o si el aprendizaje presume la desconexión de la red.

Otro criterio suele razonar dos tipos de normas de aprendizaje: las de aprendizaje supervisado y las correspondientes a un aprendizaje no supervisado, estas normas dan pie a una de las simbolizaciones que se ejecutan de las RNA: Redes neuronales con aprendizaje supervisado y redes neuronales con aprendizaje no supervisado. La diferencia fundamental entre ambos tipos consiste en la coexistencia o no de un agente externo (supervisor) que inspeccione el aprendizaje de la red.

Redes con Aprendizaje Supervisado. El proceso de aprendizaje se efectúa mediante un entrenamiento vigilado por un agente externo (supervisor, experto) que establece la réplica que correspondería forjar la red a partir de una entrada concluyente.

El supervisor demuestra la salida de la red y en el caso de que ésta no concuerde con la esperada, se derivará a cambiar los pesos de las conexiones, con la finalidad de obtener que la salida se avecine a la ambicionada (Palisade, NeuralTools Webcasts, 2014).

Se suponen tres maneras de ejecutar este tipo de aprendizaje:

1. Aprendizaje por corrección de error: Comprende en reformar los pesos en relación de la diferenciación entre los valores esperados y los resultantes en la salida de la red; esto quiero decir, en función del error esperado.

2. Aprendizaje por refuerzo: Se compone del razonamiento de no dar indicación alguna durante el ejercicio exactamente la salida que se espera que suministre la red ante una concluyente entrada. El trabajo del supervisor se minimiza a indicar por medio de una señal de complemento o refuerzo si la salida resultante en la red se ajusta a la esperada (éxito = + 1; o fracaso = -1), y en función de ello se ajustan los pesos fundamentándose en un mecanismo de probabilidades.

3. Aprendizaje estocástico: Esta manera de aprendizaje está direccionado básicamente en hacer en los valores de los pesos de las conexiones de la red cambios aleatorios

y evaluar su efecto a partir del objetivo esperado y de las distribuciones de probabilidad (Palisade, NeuralTools Webcasts, 2014).

Redes con Aprendizaje No Supervisado. Estas redes no requieren influencia externa para ajustar los pesos de las conectividades entre las neuronas. La red no recepta información alguna por medio del entorno que le direcciona si la salida creada es o no la indicada, por ello se encuentran muchas posibilidades en cuanto a la interpretación de la salida de estas redes.

En casos ya evidenciados, la salida muestra el grado de semejanza, similitud o familiaridad entre la información que se le está mostrando en la entrada y las informaciones que se le han presentado en el pasado. En otro caso se puede realizar una codificación de la información de entrada, creando a la salida una imagen codificada de la entrada, con bits aminorados, pero manteniendo la información principal de los datos, o algunas redes con aprendizaje no supervisado lo que ejecutan es un mapeo de cualidades, generándose en las neuronas de salida una práctica geométrica que constituye un mapa topográfico de las particularidades de los datos de entrada, de tal forma que si se muestran a la red informaciones semejantes, siempre serán afectadas neuronas de salidas cercanas entre sí, en la misma zona del mapa (Palisade, NeuralTools Webcasts, 2014).

En lo habitual en este tipo de aprendizaje se suelen suponer dos tipos:

1. Aprendizaje Hebbiano: Consiste fundamentalmente en el ajuste de los pesos de las conexiones de acuerdo con la correlación, esto quiere decir si las dos unidades son activas (positivas), se provoca un reforzamiento de la conexión. Por el contrario cuando una es activa y la otra pasiva (negativa), se causa en la conexión un debilitamiento.

2. Aprendizaje competitivo y cooperativo: Las neuronas rivalizan (y se ayudan) unas con otras con el fin de llevar a cabo una labor dada. Con esta pauta de aprendizaje se procura que cuando se muestre a la red ciertos datos de entrada, solo una de las neuronas de salida se active, por lo tanto las neuronas rivalizan por activarse, permaneciendo finalmente una, o una por grupo, como neurona triunfadora (Palisade, NeuralTools Webcasts, 2014)

Estos dos mecanismos de asociación dan lugar a dos tipos de redes neuronales: las redes heteroasociativas y las autoasociativas. Una red heteroasociativa podría considerarse

aquella que computa cierta función, que en la mayoría de los casos no podrá expresarse analíticamente, entre un conjunto de entradas y un conjunto de salidas, correspondiendo a cada posible entrada una determinada salida.

Existen redes heteroasociativas con conexiones feedforward, feedforward/feedback y redes con conexiones laterales. También existen redes heteroasociativas multidimensionales y su aprendizaje puede ser supervisado o no supervisado (Palisade, NeuralTools Webcasts, 2014).

Por otra parte, una red autoasociativa es una red cuya principal misión es reconstruir una determinada información de entrada que se presenta incompleta o distorsionada (le asocia el dato almacenado más parecido). Pueden implementarse con una sola capa, existen conexiones laterales o también autorrecurrentes, habitualmente son de aprendizaje no supervisado (Palisade, NeuralTools Webcasts, 2014).

Tipo de Asociación entre las Informaciones de Entrada y Salida. Las RNA son sistemas que almacenan cierta información aprendida; esta información se registra de forma distribuida en los pesos asociados a las conexiones entre neuronas de entrada y salida.

Existen dos formas primarias de realizar esa asociación de entrada/salida. Una primera sería la denominada heteroasociación, que se refiere al caso en el que la red aprende parejas de datos $[(A_1, B_1), (A_2, B_2), \dots, (A_n, B_n)]$, de tal forma que cuando se presente cierta información de entrada A_i , deberá responder generándola correspondiente salida B_i .

La segunda se conoce como autoasociación, donde la red aprende ciertas informaciones A_1, A_2, \dots, A_n , de tal forma que cuando se le presenta una información de entrada realizará una autocorrelación, respondiendo con uno de los datos almacenados, el más parecido al de la entrada (Palisade, NeuralTools Webcasts, 2014).

Representación de la Información de Entrada y Salida. Las redes neuronales pueden también clasificarse en función de la forma en que se representan las informaciones de entrada y las respuestas o datos de salida. Así un gran número de redes, tanto los datos de entrada como de salida son de naturaleza analógica, cuando esto ocurre, las funciones de activación de las neuronas serán también continuas, del tipo lineal o sigmoidal.

Otras redes sólo admiten valores discretos o binarios a su entrada, generando también unas respuestas en la salida de tipo binario. En este caso, las funciones de activación de las neuronas son de tipo escalón. Existe también un tipo de redes híbridas en las que las informaciones de entrada pueden ser valores continuos, aunque las salidas de la red son discretas (Palisade, NeuralTools Webcasts, 2014).

1.5.METODOLOGÍA

Una aplicación de redes neuronales artificiales, comprende varias fases o etapas, para el desarrollo y validación de la estructura se tienen las siguientes fases:

Definición de la red neuronal.- Se determina el número de neuronas de las capas de entrada, oculta y salida, así como también las diferentes funciones de activación que se utilizarán en las neuronas de la capa oculta y salida (Palisade, NeuralTools Academic Newsletters, 2015).

Esta fase comprende la elección del modelo de red, y obtención del conjunto de variables significativas para la resolución del problema. La definición de la red neuronal, también es entendida como una selección del paradigma neuronal, en la cual se debe tener en consideración los siguientes aspectos:

1. **Tamaño de la red.** Selección del tamaño de la red.
2. **Tipo de problema a resolver.** La salida de la red puede ser interpretada de diversas formas: optimización o clasificación.
3. **Tipo de asociación.** Está dividida en dos tipos: la memoria auto-asociativa (aprendizaje no supervisado) y la memoria hetero-asociativa (aprendizaje supervisado) (Palisade, *NeuralTools Manuals*, 2013).

Entrenamiento de la red neuronal.- En esta etapa se define el tipo de entrenamiento que se realizará, si es aprendizaje supervisado o no supervisado, también se determinan los algoritmos de entrenamiento.

La fase de entrenamiento comprende inicialmente, la definición de uno de los tipos conocidos como métodos de entrenamiento, este proceso se efectúa mediante un algoritmo de

aprendizaje (Palisade, NeuralTools Manuals, 2013) . El aprendizaje es un proceso de ajuste de los pesos entre las conexiones de las capas. La red posee la propiedad de modificarse en función de la necesidad de aprender la información que se le presentó en la definición de la red neuronal (Palisade, NeuralTools Manuals, 2013).

Utilización de la red neuronal.- La fase de utilización es propiamente la ejecución de la estructura de la red, se inicia cuando se presenta una entrada a la red y termina generando una salida en función a las entradas proporcionadas (Palisade, NeuralTools Manuals, 2013). La principal diferencia entre la utilización y el aprendizaje es que la utilización no realiza el ajuste de los pesos y de la misma forma, no hay un cálculo para la cantidad de errores ocurridos, estos procedimientos se realizan solamente durante el aprendizaje (Palisade, NeuralTools Manuals, 2013).

Mantenimiento de la red neuronal.- La mayoría de las redes se entrenan para solucionar problemas dinámicos, por lo tanto es necesaria e indispensable una validación continua para garantizar una buena utilización. A medida que transcurra el tiempo podrán surgir nuevos conjuntos de datos reales aún desconocidos por la red, y de esta forma, será necesario un nuevo aprendizaje o, dependiendo del caso realizar una nueva definición (Palisade, NeuralTools Manuals, 2013).

Entendiendo estos parámetros la metodología a seguir es usar la base de datos suministrada de una entidad bancaria en la que será filtrada entre los postulantes que fueron aprobados para cierto crédito y los que no aprobaron el crédito, se toma el grupo que no aprobaron el crédito porque esa es la población a investigar.

En esta investigación se usará el software de herramientas neuronales de Palisade (Palisade, NeuralTools Webcasts, 2014) en la que se tomará es base de datos filtrada y analizamos el tamaño que la red va a tener, es decir cuanta carga de datos va a soportar para que después definir el problema a resolver, en este caso es la de optimizar los criterios de aprobación de los créditos.

Luego de esto identificamos las entradas y salidas de programación del modelo de scoring para posteriormente pasar a entrenar a la red neuronal, este entrenamiento será supervisado ya que el algoritmo a usar requiere de manipulación de las variables a adjuntar.

Después de eso se entra al proceso de utilización de la red neuronal que a diferencia del entrenamiento, este ya ejecutará el reporte final en el que se encontrará la optimización de los datos suministrados y el poder tomar la decisión correcta.

Se exige como parte final de la metodología es que a este modelo se le debe dar el mantenimiento adecuado ya que puede haber solicitudes de nuevas variables o algún enfoque semejante (Palisade, NeuralTools Manuals, 2013).

1.6.OBJETIVOS.

Objetivo general.

Crear y ejecutar un modelo de scoring mediante la técnica de redes neuronales para conocer el comportamiento financiero de un cliente y mejorar la predicción de la condición de buen o mal pagador.

Objetivo específicos.

- Identificar la población objetivo que se someterá a la prueba.
- Identificar el tipo de crédito más negado.
- Determinar las posibles causas que causan la negación del crédito.
- Identificar el patrón neuronal repetitivo.
- Ejecutar y analizar el reporte del modelo econométrico.

CAPÍTULO 2

2.1.PLATAFORMA A USAR.

NeuralTools ofrece a Microsoft Excel –el programa de modelación y análisis de datos estándar de la industria - un nuevo y eficaz grupo de herramientas de modelación. NeuralTools es un programa auxiliar de redes neuronales para Microsoft Excel que permite analizar datos en las hojas de cálculo de Excel y trabajar en el entorno familiar de Microsoft Office.

Combinando un eficaz administrador de datos y la administración de un buen algoritmo de redes neuronales, NeuralTools permite la facilidad de uso y los informes de Microsoft Office, y las predicciones sólidas y precisas de las redes neuronales.

Si sabe cómo usar Excel, sabrá usar NeuralTools. NeuralTools funciona como lo hace Excel, con barras de herramientas, menús y funciones personalizadas de hojas de trabajo, y todo ello dentro de Excel.

A diferencia de los programas de redes neuronales de funcionamiento independiente, con NeuralTools no hay una pronunciada curva de aprendizaje ni costos iniciales de formación, porque se trabaja como usted trabaja en Excel.

Los datos y variables están en hojas de cálculo de Excel. Puede usar fórmulas estándar de Excel para hacer cálculos así como tablas de pivote y ordenación de Excel.

Los informes y gráficas de los análisis se ofrecen en el formato estándar de Excel y pueden utilizar todas las funciones de formato incorporadas de Excel.

Las redes neuronales son capaces de aprender complejas relaciones de datos.

Al imitar las funciones de un cerebro, pueden diferenciar patrones de datos y luego extrapolar predicciones cuando se introducen nuevos datos. Los problemas en los que se usan redes neuronales se pueden dividir en dos grupos generales:

1. **Problemas de clasificación:** Problemas en los que se trata de determinar el tipo de categoría al que pertenece un elemento desconocido. Como ejemplo están los diagnósticos médicos o las predicciones de capacidad de pago de un crédito.
2. **Problemas numéricos:** Situaciones en las que debe predecir un resultado numérico específico. Como ejemplo están la previsión del precio de acciones o la predicción del nivel de ventas durante un periodo de tiempo futuro.

Las redes neuronales se usan en una amplia variedad de aplicaciones, que incluyen: predicciones en el mercado de valores, asignación de riesgo en créditos y préstamos, detección de fraude crediticio, previsiones de ventas, previsiones comerciales en general, riesgo en inversiones, diagnósticos médicos, investigación científica y sistemas de control.

NeuralTools incorpora los más modernos algoritmos de redes neuronales para hacer las mejores predicciones tanto en problemas de clasificación (denominados predicción de categoría en NeuralTools) como en problemas numéricos (Palisade, NeuralTools Manuals, 2013)

2.2. IDENTIFICACIÓN DEL TIPO DE MODELO A USAR.

El tipo de modelo a usar es un modelo basado en la regresión lineal del tipo multivariado, tomando como población objetivo una población homogénea, generalmente de 10 a 20 variables y con observaciones de clientes que varían y otros que no. Este tipo de modelo neuronal basado en la regresión lineal multivariada calcula la probabilidad por defecto que no es más que lo que hace un modelo de regresión logística y también calcula la pérdida esperada.

Los datos en NeuralTools se estructuran en casos y variables. Se trabaja con un conjunto de datos, o un conjunto de variables estadísticas, que se encuentran en columnas contiguas con los nombres de las variables en la primera fila del conjunto de datos (Palisade, NeuralTools Manuals, 2013).

Cada fila del conjunto de datos es un caso. Cada caso tiene un conjunto de valores de variable independiente y un valor conocido o que falta para la variable de salida dependiente. El trabajo de NeuralTools consiste en predecir los valores de variable de salida de los casos en los que no son conocidas mediante el algoritmo establecido (Palisade, NeuralTools Manuals, 2013).

El Administrador de conjunto de datos de NeuralTools permite definir los conjuntos de datos y las variables. Luego podrá usar estas variables predefinidas para entrenar y probar redes neuronales, sin tener que seleccionar una y otra vez los datos que desea analizar. Puede colocar todos los casos históricos conocidos en un conjunto de datos y los casos para los que desea predecir los resultados en otro conjunto de datos diferente. También puede combinar todos los datos –los datos históricos conocidos y los datos que desea predecir– en un solo conjunto de datos (Palisade, NeuralTools Manuals, 2013).

2.3.VARIABLES.

Las variables a usar están divididas en 3 clasificaciones:

Variable numérica independiente: Es la variable que presenta separaciones o interrupciones en la escala de valores que puede tomar. Estas separaciones o interrupciones indican la ausencia de valores entre los distintos valores específicos que la variable pueda asumir independientemente (Palisade, NeuralTools Manuals, 2013).

Variable categórica independiente: Es la variable que puede tomar distintos valores ordenados siguiendo una escala establecida e independiente, aunque no es necesario que el intervalo entre mediciones sea uniforme (Palisade, NeuralTools Manuals, 2013).

Variable categórica dependiente: Es la última y única variable dependiente en el experimento, es la variable que puede tomar distintos valores ordenados siguiendo una escala establecida pero con característica de dependencia (Palisade, NeuralTools Manuals, 2013).

Las variables a usar son:

Cliente: Número de cliente evaluado por orden de solicitud receptada.

Edad: Edad en años del cliente.

Ingreso: Recopilación mensual de sueldos y salarios del cliente.

Egresos: Gastos mensuales del cliente.

Antigüedad: Número de años de trabajo del cliente en un puesto fijo.

Deuda: Monto en deuda por préstamos anteriores a la evaluación actual.

Garante: Activos fijos o representante de liquidez como seguro para la evaluación del crédito.

Tasa: Tasa efectiva y/o nominal que da la institución para el préstamo.

Meses de préstamo: Tiempo por el cual debe ser cancelado el préstamo a solicitar.

Préstamo: Monto en efectivo solicitado

Calificación: Aprobación o no aprobación del crédito solicitado.

Figura 2.1.- Variables a evaluar.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Etiqueta	Cliente	Edad	Ingreso	Egresos	Antigüedad	Deuda	Garante	Tasa	Meses de Préstamo	Préstamo	Calificación
3	Predecir	1	19	\$ 1,100.00	\$ 400.00	2	no	no	10.5	36	\$15,000.00	
4	Predecir	2	19	\$ 1,300.00	\$ 500.00	1	no	no	10.5	30	\$25,000.00	
5	Predecir	3	20	\$ 2,000.00	\$ 800.00	2	no	no	10.5	24	\$34,000.00	
6	Probar	4	38	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	12	\$25,000.00	si
7	Probar	5	39	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	30	\$30,000.00	si
8	Probar	6	40	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	18	\$12,000.00	si
9	Probar	7	36	\$ 5,000.00	\$ 3,000.00	13	no	si	10.5	36	\$30,000.00	si
10	Probar	8	37	\$ 5,000.00	\$ 3,000.00	18	no	si	10.5	60	\$45,000.00	si
11	Probar	9	38	\$ 5,000.00	\$ 3,000.00	19	si	si	10.5	48	\$47,000.00	si
12	Probar	10	39	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$32,000.00	si
13	Probar	11	40	\$ 5,000.00	\$ 3,000.00	21	si	si	10.5	30	\$27,000.00	si
14	Probar	12	41	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	24	\$25,000.00	si
15	Probar	13	42	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	12	\$30,000.00	si
16	Probar	14	43	\$ 5,000.00	\$ 3,000.00	19	no	si	10.5	30	\$12,000.00	si
17	Probar	15	44	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	18	\$30,000.00	si
18	Probar	16	45	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$45,000.00	si
19	Probar	17	38	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	60	\$47,000.00	si
20	Probar	18	39	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	48	\$32,000.00	si
21	Probar	19	40	\$ 5,000.00	\$ 3,000.00	12	si	si	10.5	12	\$27,000.00	si
22	Probar	20	36	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	30	\$25,000.00	si

2.4. OPERACIONALIZACIÓN DE VARIABLES.

Se sugiere los siguientes pasos:

- 1.- Definición del estimador que se desea predecir (BAHAMÓN, 2013).
- 2.- Selección de las variables para realizar el análisis de su comportamiento a través del tiempo lo cual permitiría predecir resultados futuros del estimador seleccionado. Es esta actividad se debe contemplar la definición de la información base para el análisis. Es necesario definir una base de datos con las variables más significativas que permitan realizar en forma efectiva el análisis. Variables seleccionadas para los créditos desembolsados y cancelados: tipo de la tarjeta o BIN, fecha de corte (mes y año), valor mora, pago mínimo, días en mora, código mora, ciudad, cupo tarjeta, rango SMLV (rango en que se encuentra el cupo según el número de salarios mínimos legales vigentes. Rango 1=si el cupo es menor a 10 SMLV, Rango 2= si el cupo esta entre 10 y 20 SMLV, Rango 3= si el cupo esta entre 20 y 50 SMLV, y, Rango 4= si el cupo es mayor a 50 SMLV) (BAHAMÓN, 2013).
- 3.- Buscar y recolectar la información determinando el tamaño de la muestra (cantidad de datos mínimos y suficientes para realizar el análisis). Realizar una normalización de los datos (BAHAMÓN, 2013).
- 4.- Cargar los datos como si fuera una matriz. Para nuestro ejercicio, y al igual que para el caso de evaluación y clasificación, se utiliza MATLAB [20] para cargar la información mediante la función de importación (BAHAMÓN, 2013).
- 5.- Creación y definición de cada una de las variables: definición de vectores que contienen los valores de las variables, cada una de las columnas representa una variable (i.e., el tiempo es una variable continua) (BAHAMÓN, 2013).
- 6.- Determinar la función que se debe aprender. En línea con el numeral 2.- es preciso definir la variable a estimar y con respecto a qué variables se va a realizar dicha estimación (i.e., con respeto al tiempo determinando si es una variable cíclica y continua) (BAHAMÓN, 2013).

7.- Normalización de las variables: se debe revisar cómo es el comportamiento de las variables mediante el análisis del histograma que generan sus valores y se grafica (BAR) para revisar como es la distribución de la función (i.e., comportamiento Gaussiano). Si la distribución de una variable no es normal se puede utilizar herramientas estadísticas para la normalización de la misma. No obstante, también se puede usar métodos de normalización no estadística como el siguiente ejemplo: $Variable_normalizada = ((valor_variable - \min(valor_variable)) / (\max(valor_variable) - \min(valor_variable)))$. La función HISTFIT permite ver el comportamiento de una variable a través del tiempo. Esta función muestra la gráfica normal con diferentes distribuciones de probabilidad. Usar una normalización Gaussiana: $Variable_normalizada = ((valor_variable - \text{mean}(valor_variable)) / \text{std}(valor_variable))$. El procedimiento de normalización debe seguirse para todas las variables. La normalización se puede hacer desde el momento en que son seleccionadas las variables y puede usarse Excel o Matlab indistintamente (BAHAMÓN, 2013).

8.- En resumen los pasos para la construcción de esta solución en MATLAB son:

- Cargue de los datos como si fuera una matriz
- Crear y definir las variables usando vectores (cada una de las columnas representa una variable)
- Determinar la función objetivo o función de aprendizaje
- Normalizar las variables: se revisa cómo es el comportamiento de las variables mediante el análisis de sus valores y distribución en el tiempo (i.e., a través de un histograma)
- Seleccionar el tipo de RNA a utilizar. Se usan los vectores transpuestos para convertir las columnas en filas y poder utilizar la RNA. Cuando se utiliza la función “newff” para la creación de la red neuronal se tiene por defecto “tansig” para las capas ocultas y salida lineal por defecto
- Definir el vector de salida o Target, que corresponde a la variable que se quiere predecir y/o estimar
- Realizar los diferentes ajustes para entrenar la RNA
- Realizar el proceso de aprendizaje

- Ejecutar este proceso hasta que los resultados sean similares a la función de entrada.

NOTA: existen otras herramientas como SPSS o MODELER que también pueden ser usadas para el desarrollo e implementación de este tipo de modelos (BAHAMÓN, 2013).

Es preciso tener en consideración variables cuantitativas y cualitativas, correspondientes a cada una de las modalidades de cartera (i.e., comercial, consumo, hipotecaria, tarjeta de crédito, microcrédito, etc.) y a cada problema que se está analizando y solucionando (i.e., clasificación o predicción). Si existe una situación en la que se presenta una fuerte correlación entre las variables explicativas del modelo, significa que es posible tener problemas de multicolinealidad por lo que se sugiere analizar y evaluar dicho nivel de correlación que existe entre las variables. Para evitar estos problemas se realiza un análisis de componentes principales (ver sección 5.2.) que permite pasar de un espacio linealmente dependiente (LD) a un espacio linealmente independiente (LI), permitiendo estimar el modelo de redes neuronales. Es importante revisar la relación que existe entre la varianza de la variable observada con respecto a la varianza de las demás variables. En este punto se establece cuáles son las variables cuyo comportamiento afecta el de la variable observada (BAHAMÓN, 2013).

2.5.ALGORITMO A USAR.

El proceso de aprendizaje de los mapas auto-organizados (SOFM) consiste inicialmente en una ordenación global y luego un ajuste fino o especialización de las neuronas. Este es un ejemplo de un algoritmo auto-organizado:

- 1.- Inicialización de los pesos sinápticos W_{ijk} (BAHAMÓN, 2013).
- 2.- Para cada iteración presenta el patrón $x(t)$ con base en la función de distribución $p(x)$ del espacio sensorial (capa de entrada) (BAHAMÓN, 2013).
- 3.- Calcular similitud, utilizando la distancia euclídea, para cada neurona $i = (i, j)$ entre su vector de pesos sinápticos W_{ij} y el actual vector de entradas x (BAHAMÓN, 2013).

$$d^2(w_{ij}, x) = \sum_{k=1}^n (w_{ijk} - x_k)^2$$

4.- Determinar la neurona ganadora $g = (g1, g2)$ cuya distancia sea la menor (BAHAMÓN, 2013).

5.- Actualizar los pesos sinápticos de la neurona ganadora $g = (g1, g2)$ y de las vecinas (BAHAMÓN, 2013)

$$W_{ijk}(t + 1) = W_{ijk}(t) + \alpha(t)h(|i-g|,t)(X_k(t) - W_{ijk}(t))$$

Siendo $\alpha(t)$ el ritmo de aprendizaje, $h(\cdot)$ la función de vecindad (BAHAMÓN, 2013).

6.- Volver a 2 hasta el número máximo de iteraciones (BAHAMÓN, 2013).

Las simulaciones de un mapa auto-organizado (SOFM) usualmente son más rápidas que las de BP ya que su algoritmo requiere de menos recursos computacionales (BAHAMÓN, 2013).

Existe un caso de estudio reconocido a nivel mundial, basado en la técnica de mapas de rasgos auto-organizados que es el “Análisis de la corrupción en los países” en el cual se revisa la posible relación que existe entre la corrupción de un gobierno y el terrorismo (BAHAMÓN, 2013).

Su acercamiento se hace desde el punto de vista de minería de datos lo cual lo hace más llamativo relacionando variables macroeconómicas con la percepción de la corrupción en los países en desarrollo (BAHAMÓN, 2013).

LVQ – Learning Vector Quantization es un modelo supervisado compuesto por una capa simple de neuronas sin relaciones de vecindad, que está orientado a clasificación de patrones en donde se premia a aquellas neuronas que clasifican correctamente un determinado patrón actualizando sus pesos y castigando a las que clasifican erróneamente. Se sugiere la

utilización del modelo de mapas auto-organizados (SOFM) para la visualización de datos y no emplearse en clasificación (BAHAMÓN, 2013).

Redes No Supervisadas Realimentadas - Modelo Hopfield.

A los modelos neuronales se les puede incluir un esquema de realimentación lo cual complica el análisis de su operación debido a que la información viaja hacia delante y hacia atrás convirtiéndose en un sistema dinámico. El funcionamiento de una red unidireccional es estable en fase de ejecución o recuerdo y arroja resultados inmediatos. En cambio, en las redes realimentadas el problema es la estabilidad de la respuesta cuya demostración usa el método de Lyapunov para el cual un sistema dinámico de variables de entrada (ecuaciones diferenciales) se encuentra en reposo solo en el origen (BAHAMÓN, 2013).

$$\dot{x}_i = \frac{dy}{dx} = F(t, x_1, x_2, \dots, x_n)$$

La función de Lyapunov se denomina también función de energía de Lyapunov ya que es la generalización del concepto físico de energía (BAHAMÓN, 2013).

$$V = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \leq 0, \forall x_i$$

El modelo de Hopfield fue el punto de partida para que las RNA “renacieran” con fuerza. Este modelo está basado en la definición de una función de energía de la red y utiliza modelos de la física estadística. Su arquitectura consiste de una sola capa donde las neuronas (similares a las del perceptrón) están conectadas entre sí, un elemento de umbral con entradas $x_j(t)$ y salidas $y_i(t)$ (binarias) (BAHAMÓN, 2013).

Los pesos son continuos (reales). La suma de las entradas ponderada con los pesos sinápticos menos el umbral, corresponde al potencial postsináptico calculado por la neurona (BAHAMÓN, 2013).

$$h_i(t) = \sum_j w_{ij} x_j(t) - \theta_i$$

Para obtener la salida digital de la neurona se aplica al potencial local una función de tipo escalón $f(t)$ por lo que la siguiente regla rige la dinámica (determinística) de la red:

$$y_i(t) = f(h_i(t)) = f \left[\sum_j w_{ij} x_j(t) - \theta_i \right] = x_i(t + 1)$$

Los esquemas de actualización o dinámicas pueden ser asíncronos o modo serie (en un instante t solo una neurona actualiza su estado) y síncronos o modo paralelo (en un instante t varias neuronas actualizan su estado). La red de Hopfield opera como una memoria auto-asociativa por lo que puede eliminar o filtrar el ruido en patrones. El estado estable de la red puede ser un mínimo local de la función de energía. La red almacena o memoriza estos resultados (mínimos locales) y en esto consiste su esquema de aprendizaje. La regla de aprendizaje deberá encontrar el conjunto de pesos sinápticos W y que la función de energía tenga esos patrones como mínimos locales (estados estables) (BAHAMÓN, 2013).

CAPITULO 3

3.1.PREPARACIÓN DE LOS DATOS.

Como se mencionó es importante proveer de técnicas a los analistas para que conozcan y comprendan el conjunto de reglas que las redes neuronales artificiales utilizan para resolver los problemas de evaluación del riesgo de crédito y la predicción de la morosidad en la cartera (BAHAMÓN, 2013).

Existen tres técnicas de extracción de reglas: Neurorule, Trepan y Nefclass. Las dos primeras producen una buena precisión de clasificación y son capaces de extraer los conjuntos de reglas y los árboles de decisión para todos los conjuntos de datos (BAHAMÓN, 2013).

Las tablas de decisión pueden ser usadas para representar estas reglas a través de un formato gráfico intuitivo de fácil interpretación y verificación por parte de los analistas y/o evaluadores. Estas técnicas ayudan a validar que la red neuronal está funcionando bien y sus resultados son buenos. Entender y comprender cuáles son las reglas con las que opera internamente la red neuronal permite comprobar lo buena que es la RNA (BAHAMÓN, 2013).

La normalización de los datos es clave, permite establecer por experiencia y a priori un primer acercamiento a la relación que tienen los datos entre sí. Por ejemplo, si se tiene un conjunto continuo de datos en el tiempo es posible aplicar una normalización gaussiana de forma tal que permita entender y manejar los datos de una forma adecuada (BAHAMÓN, 2013).

Los análisis temporales (en el tiempo) son bastante útiles sobre todo si se trata de investigaciones o estudios que como en nuestro caso requieren de proyecciones y/o predicciones del comportamiento futuro de las variables. Existen formas sencillas de normalizar la información utilizando una simple codificación de los datos, por ejemplo, en el caso del BIN o identificador de las tarjetas de crédito se puede asignar un número de 1 a N que correspondan a cada BIN (BAHAMÓN, 2013).

Es importante entender los datos y como se relacionan entre sí. Para realizar una simulación rápida, sencilla y eficaz, se decidió solo tomar las tarjetas de crédito empresariales.

Al aplicar un Mapa Autoorganizado – SOM a los datos de entrada, la red neuronal agrupa y clasifica la información según sus características y empieza a encontrar cuáles son los grupos que guardan similitud y cuáles no (BAHAMÓN, 2013).

Es preciso reconocer e identificar cuáles son las variables representativas que causan, con mayor fuerza, que unos registros estén más cerca que otros. Para esto, fueron realizadas varias simulaciones con diferentes dimensiones para las matrices (ver Anexo 1.-). Para una de las simulaciones, la red neuronal se escogió de dimensiones 4X4, lo que implica tener una red de 16 neuronas. Es posible realizar simulaciones con más dimensiones pero, dado el número total de registros a analizar (874), este dimensionamiento de 4X4 se considera adecuado (BAHAMÓN, 2013).

De igual forma, se puede ensayar varias topologías y diferentes conjuntos de datos con miras a entender los grupos que pueden ser de alto riesgo. Se observa cómo las simulaciones con los mapas autoorganizados – SOM hacen que el agrupamiento y distribución de los datos se realice en forma natural a través del análisis de componentes. Además, esta técnica permite reducir las dimensionalidades.

Los planos de pesos de las variables de entrada del modelo auto-organizado muestran la relación que existe entre los pesos de las variables de entrada de una neurona con sus vecinas. Estos planos de pesos también son conocidos como los planos de componente. Los colores más claros representan los pesos más grandes y los colores más oscuros representan los pesos más pequeños (BAHAMÓN, 2013).

Las neuronas rojas representan conexiones positivas y fuertes y las negras significa que no existe conexión o relación alguna. Si el patrón de conexiones de dos entradas es similar, se puede asumir que las entradas están “altamente” correlacionadas (BAHAMÓN, 2013).

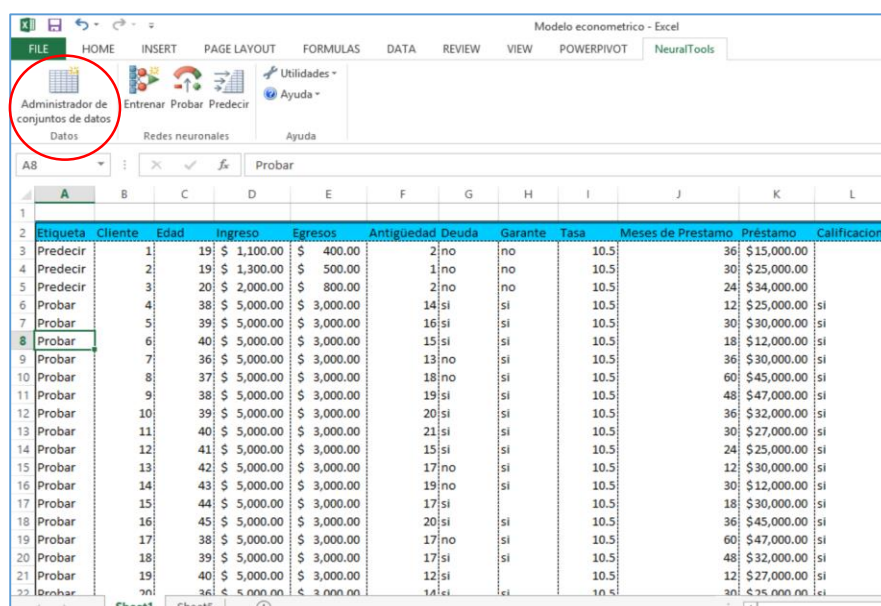
El objetivo de este modelo es que se tiene una base de datos que nos provee información de solicitudes de créditos de clientes que desean acceder a un tipo de préstamo como se presenta en el gráfico 3.1. Se observa que los 3 primeros clientes son los que el modelo va a evaluar si se les aprueba el crédito o no mediante la predicción neuronal.

3.2.ADMINISTRACIÓN DEL CONJUNTO DE DATOS.

Los datos en NeuralTools se estructuran en casos y variables. Se trabaja con un conjunto de datos, o un conjunto de variables estadísticas, que se encuentran en columnas contiguas con los nombres de las variables en la primera fila del conjunto de datos. Cada fila del conjunto de datos es un caso (Palisade, NeuralTools Manuals, 2013).

Cada caso tiene un conjunto de valores de variable independiente y un valor conocido o que falta para la variable de salida dependiente. El trabajo de NeuralTools consiste en predecir los valores de variable de salida de los casos en los que no son conocidas (Palisade, NeuralTools Manuals, 2013).

Figura 3.1 de administración de conjunto de datos



	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Etiqueta	Cliente	Edad	Ingreso	Egresos	Antigüedad	Deuda	Garante	Tasa	Meses de Prestamo	Préstamo	Calificación
3	Predecir	1	19	\$ 1,100.00	\$ 400.00	2	no	no	10.5	36	\$15,000.00	
4	Predecir	2	19	\$ 1,300.00	\$ 500.00	1	no	no	10.5	30	\$25,000.00	
5	Predecir	3	20	\$ 2,000.00	\$ 800.00	2	no	no	10.5	24	\$34,000.00	
6	Probar	4	38	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	12	\$25,000.00	si
7	Probar	5	39	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	30	\$30,000.00	si
8	Probar	6	40	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	18	\$12,000.00	si
9	Probar	7	36	\$ 5,000.00	\$ 3,000.00	13	no	si	10.5	36	\$30,000.00	si
10	Probar	8	37	\$ 5,000.00	\$ 3,000.00	18	no	si	10.5	60	\$45,000.00	si
11	Probar	9	38	\$ 5,000.00	\$ 3,000.00	19	si	si	10.5	48	\$47,000.00	si
12	Probar	10	39	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$32,000.00	si
13	Probar	11	40	\$ 5,000.00	\$ 3,000.00	21	si	si	10.5	30	\$27,000.00	si
14	Probar	12	41	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	24	\$25,000.00	si
15	Probar	13	42	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	12	\$30,000.00	si
16	Probar	14	43	\$ 5,000.00	\$ 3,000.00	19	no	si	10.5	30	\$12,000.00	si
17	Probar	15	44	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	18	\$30,000.00	si
18	Probar	16	45	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$45,000.00	si
19	Probar	17	38	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	60	\$47,000.00	si
20	Probar	18	39	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	48	\$32,000.00	si
21	Probar	19	40	\$ 5,000.00	\$ 3,000.00	12	si	si	10.5	12	\$27,000.00	si
22	Probar	20	36	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	30	\$25,000.00	si

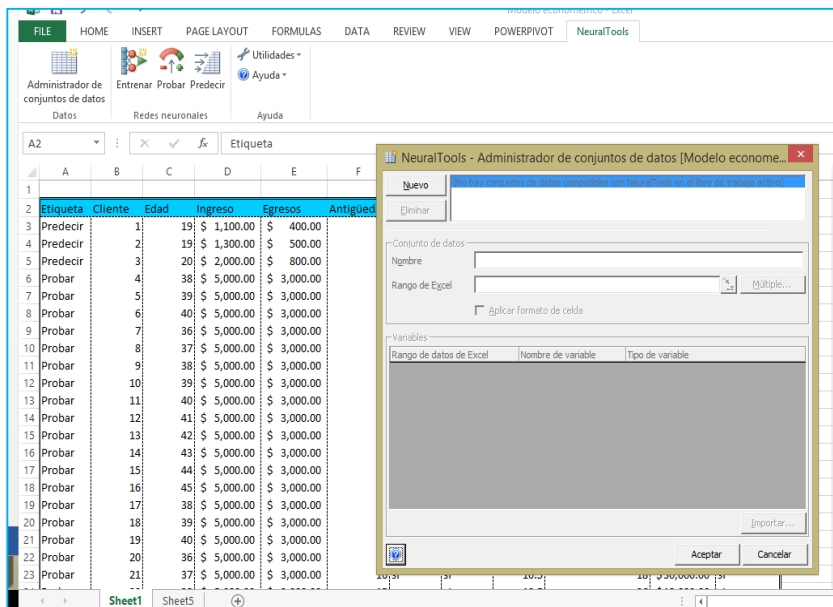
Fuente: Elaborado por el autor.

El Administrador de conjunto de datos de NeuralTools permite definir los conjuntos de datos y las variables. Luego podrá usar estas variables predefinidas para entrenar y probar redes neuronales, sin tener que seleccionar una y otra vez los datos que desea analizar (Palisade, NeuralTools Manuals, 2013).

Puede colocar todos los casos históricos conocidos en un conjunto de datos y los casos para los que desea predecir los resultados en otro conjunto de datos diferente. También puede combinar todos los datos, los datos históricos conocidos y los datos que desea predecir— en un solo conjunto de datos (Palisade, NeuralTools Manuals, 2013).

El primer paso es seleccionar el botón administrador de conjunto de datos y se obtiene la siguiente ventana:

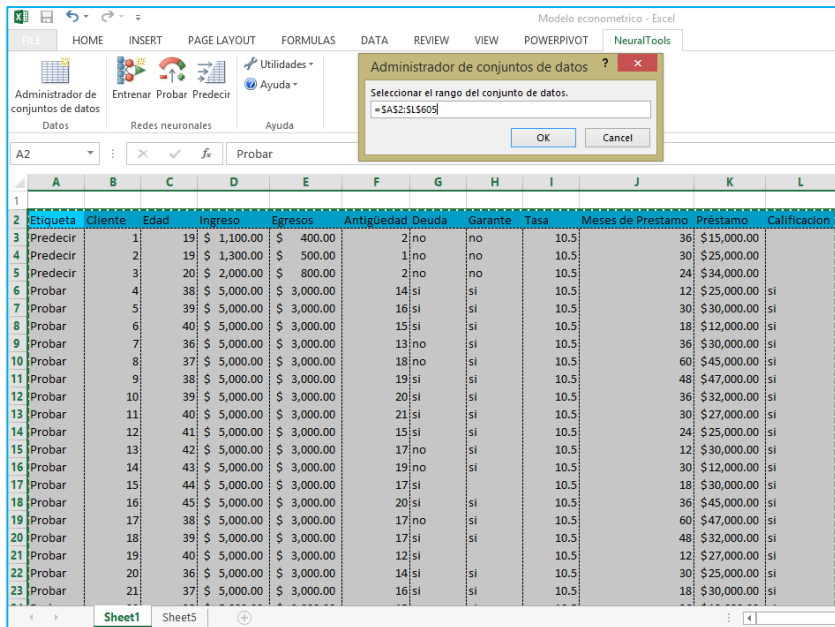
Figura 3.2 selección de conjunto de datos



Fuente: Elaborado por el autor.

Una vez ya abierta la ventana de selección de conjunto de datos, se presiona el botón *nuevo* para que se pueda habilitar el conjunto de datos. Luego de eso se selecciona el botón de selección de rango de Excel el cual se seleccionará todo el conjunto de datos.

Figura 3.3 selección del rango del conjunto de datos.

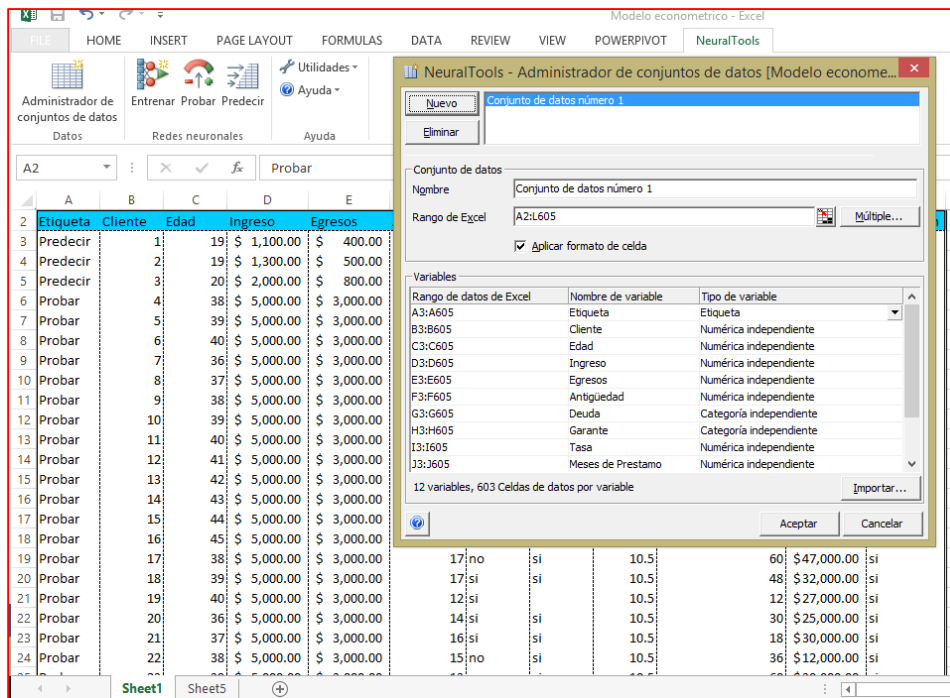


Fuente: Elaborado por el autor.

Cada variable del conjunto de datos tiene asociado un nombre y un rango de celdas de Excel. Cada columna del rango contiene datos de una variable diferente. Un conjunto de datos puede incluir múltiples bloques de celdas, lo cual permite poner los datos en diferentes hojas de un mismo libro de trabajo.

Cuando se definen conjuntos de datos, NeuralTools trata de identificar las variables en el bloque de celdas que rodean a la selección actual de Excel. Esto facilita la configuración del conjunto de datos con nombres de variable en la fila superior y las variables distribuidas en columnas (Palisade, NeuralTools Manuals, 2013).

Figura 3.4 selección validación del rango del conjunto de datos.



Fuente: Elaborado por el autor.

Una vez seleccionado el rango del conjunto de datos observamos como carga todas las variables que se ha definido. Posteriormente damos en aceptar y finalizamos la administración del conjunto de datos del modelo.

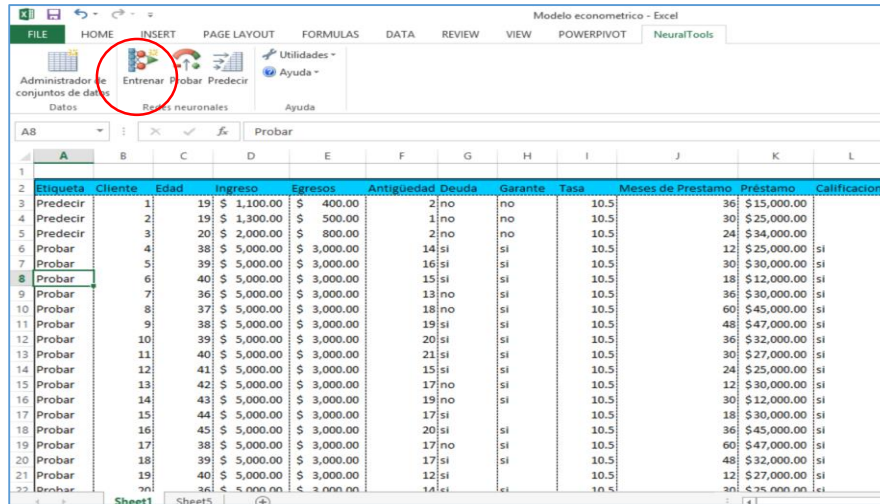
3.3. ENTRENAMIENTO DE LA RED NEURONAL.

Después de definir el conjunto de datos que contiene los casos con valores históricos conocidos, puede entrenar una red neuronal usando esos datos.

Hay diferentes opciones que determinan el tipo de red que generará NeuralTools. Dependiendo de la naturaleza de los datos, las diferentes opciones de red pueden generar redes entrenadas de mejor funcionamiento (es decir, redes que predicen respuestas mejor) (Palisade, NeuralTools Manuals, 2013).

El proceso de prueba –que se realiza después del entrenamiento– ofrece una medida precisa del funcionamiento de la red entrenada a la hora de predecir valores de salida (Palisade, NeuralTools Manuals, 2013).

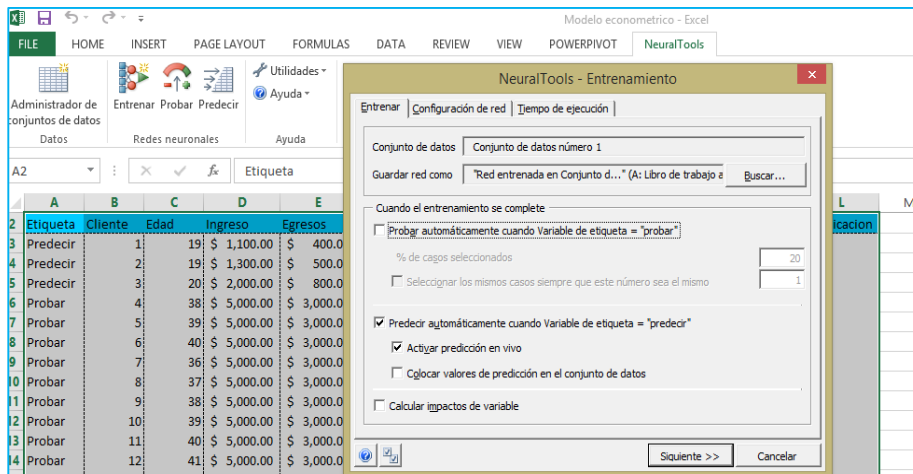
Figura 3.5 selección de entrenamiento del conjunto de datos



Fuente: Elaborado por el autor.

Al seleccionar el botón de entrenamiento se nos abre una ventana que nos permitirá escoger las diferentes opciones de cómo queremos que sea el entrenamiento del conjunto de datos seleccionado.

Figura 3.6 selección de las opciones de entrenamiento del conjunto de datos – Pestaña entrenar.



Fuente: Elaborado por el autor.

En la pestaña entrenar encontramos el nombre del conjunto de datos a entrenar, quedó definido como conjunto de datos número 1 en la administración del conjunto de datos. La red será guardada como “red entrada en el conjunto de datos seleccionados”.

Siguiendo dando lectura a la pestaña de entrenamiento encontramos la opción de: cuando el entrenamiento se complete, esta es una opción automatizada en que permite realizar las pruebas y las predicciones del modelo de forma directa pero por motivos de investigación se realizará el procedimiento paso a paso.

Si se fija bien en la parta de probar automáticamente cuando variable de etiqueta = “probar” hay unos números, esos números nos indica qué porcentaje del conjunto de datos queremos usar para las pruebas.

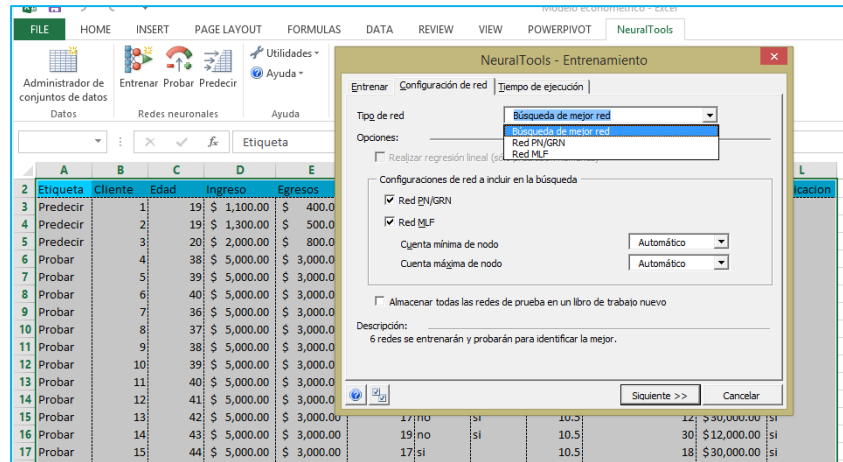
Por lo general se usan entre el 20% y el 30% del conjunto de datos para pruebas, como la base que se está usando sólo posee un poco más de 600 datos es recomendable solo mandar a un 20% para pruebas ya que se necesita la mayor cantidad de datos para que la red neuronal entrene y aprenda.

En la siguiente pestaña llamada configuración de red NeuralTools facilita la selección de una configuración de red al ofrecer una búsqueda de mejor red (Palisade, NeuralTools Manuals, 2013).

Cuando se selecciona, NeuralTools entrena y prueba una serie de configuraciones de red neuronal para generar la que realiza mejores predicciones de los datos (Palisade, NeuralTools Manuals, 2013).

La mejor configuración se determina basándose en los datos de prueba, por lo tanto debe seleccionarse la opción “Probar automáticamente” en la pestaña Entrenamiento de la búsqueda de Mejor red (Palisade, NeuralTools Manuals, 2013).

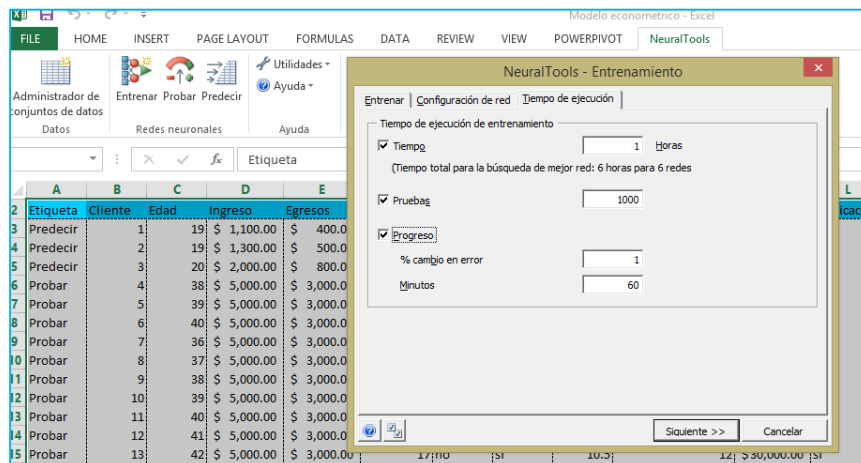
Figura 3.7 selección de las opciones de entrenamiento del conjunto de datos – Pestaña configuración de red.



Fuente: Elaborado por el autor.

La siguiente pestaña es tiempo de ejecución en el cuál se definirá el tiempo del entrenamiento de la red neuronal, el número de simulaciones de aprendizaje y el porcentaje de cambio de error.

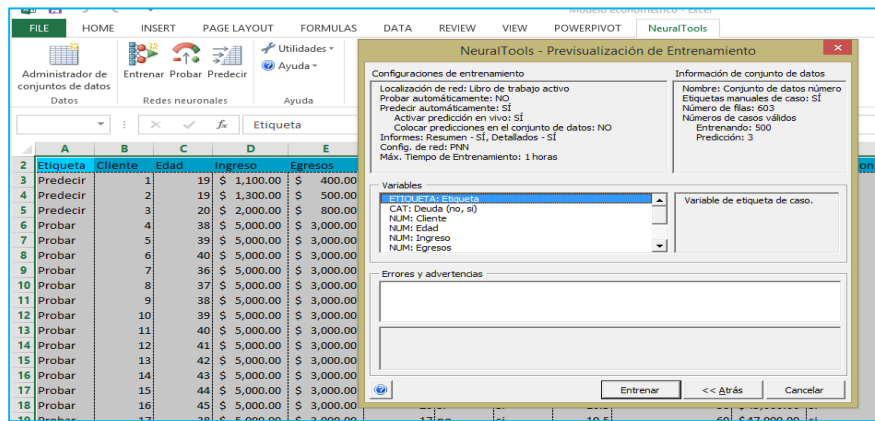
Figura 3.8 selección de las opciones de entrenamiento del conjunto de datos – Pestaña Tiempo de ejecución.



Fuente: Elaborado por el autor.

Una vez seleccionadas las opciones de entrenamiento y configuración de red, NeuralTools previsualiza lo que hará durante el entrenamiento de la red. Como el entrenamiento es el proceso que requiere más tiempo en la modelación de redes neuronales, conviene revisar la configuración de entrenamiento antes de proceder. NeuralTools tratará de identificar cualquier problema que encuentre en los datos para que pueda corregirlo antes de que se realice el entrenamiento (Palisade, NeuralTools Manuals, 2013).

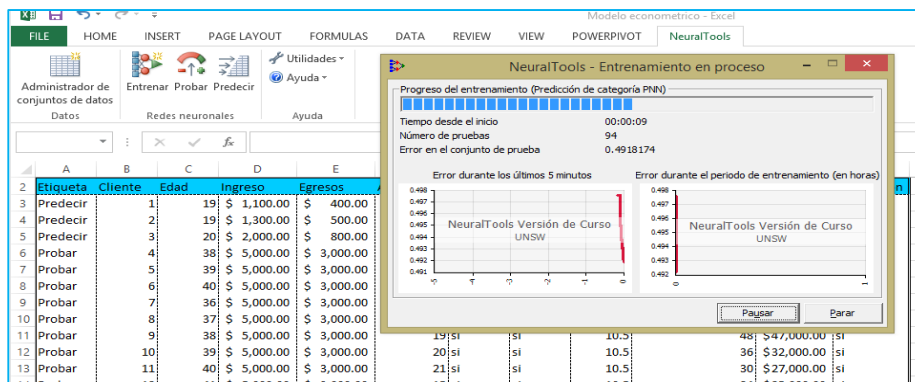
Figura 3.9 selección de las opciones de entrenamiento del conjunto de datos.



Fuente: Elaborado por el autor.

Se manda a entrenar a la red neuronal mediante la ejecución de los parámetros seleccionados y se muestra el aprendizaje de la red.

Figura 3.10 aprendizaje de la red neuronal - Etapa entrenamiento.



Fuente: Elaborado por el autor.

Cuando NeuralTools procede con el entrenamiento de la red neuronal con los datos, informa de funcionamiento del proceso. Normalmente, la red mejora cada vez más según evoluciona el entrenamiento, ya que NeuralTools genera redes que hacen mejores predicciones de los datos con menos errores (Palisade, NeuralTools Manuals, 2013).

Los gráficos se actualizan para mostrar el progreso de NeuralTools durante el entrenamiento. El entrenamiento se detiene cuando se alcanza cualquiera de las condiciones de parada establecidas, como puede ser el tiempo máximo de entrenamiento (Palisade, NeuralTools Manuals, 2013). Si ha seleccionado probar automáticamente la red o predecir los valores de salida que faltan en el conjunto de datos, esta operación se realizará después del entrenamiento (Palisade, NeuralTools Manuals, 2013). Al finalizar este proceso damos por terminada la parte del entrenamiento.

3.4.PRUEBA DE LA RED NEURONAL

Una vez entrenada la red neuronal pasamos a la etapa de pruebas que no es más que tomar el aprendizaje del entrenamiento para probarlo en tantas simulaciones sea posibles para luego predecir los datos seleccionados.

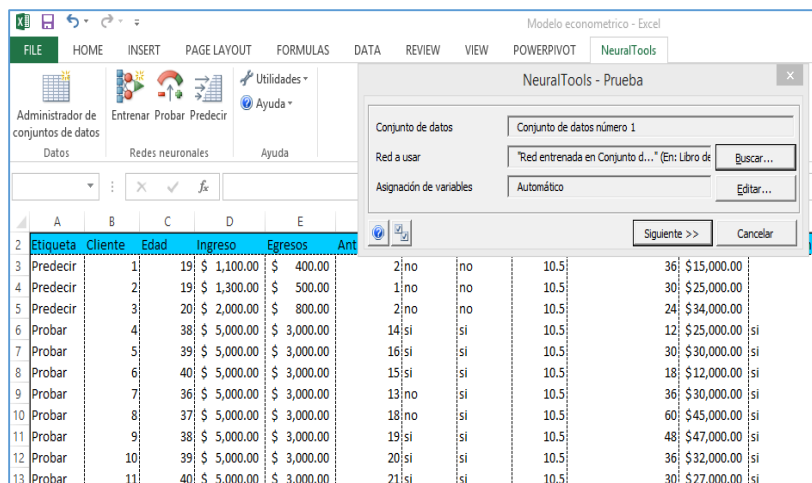
Figura 3.11 selección de prueba de la red neuronal.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2	Etiqueta	Cliente	Edad	Ingreso	Egresos	Antigüedad	Deuda	Garante	Tasa	Meses de Prestamo	Préstamo	Calificación
3	Predecir	1	19	\$ 1,100.00	\$ 400.00	2	no	no	10.5	36	\$15,000.00	
4	Predecir	2	19	\$ 1,300.00	\$ 500.00	1	no	no	10.5	30	\$25,000.00	
5	Predecir	3	20	\$ 2,000.00	\$ 800.00	2	no	no	10.5	24	\$34,000.00	
6	Probar	4	38	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	12	\$25,000.00	si
7	Probar	5	39	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	30	\$30,000.00	si
8	Probar	6	40	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	18	\$12,000.00	si
9	Probar	7	36	\$ 5,000.00	\$ 3,000.00	13	no	si	10.5	36	\$30,000.00	si
10	Probar	8	37	\$ 5,000.00	\$ 3,000.00	18	no	si	10.5	60	\$45,000.00	si
11	Probar	9	38	\$ 5,000.00	\$ 3,000.00	19	si	si	10.5	48	\$47,000.00	si
12	Probar	10	39	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$32,000.00	si
13	Probar	11	40	\$ 5,000.00	\$ 3,000.00	21	si	si	10.5	30	\$27,000.00	si
14	Probar	12	41	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	24	\$25,000.00	si
15	Probar	13	42	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	12	\$30,000.00	si
16	Probar	14	43	\$ 5,000.00	\$ 3,000.00	19	no	si	10.5	30	\$12,000.00	si
17	Probar	15	44	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	18	\$30,000.00	si
18	Probar	16	45	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$45,000.00	si
19	Probar	17	38	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	60	\$47,000.00	si
20	Probar	18	39	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	48	\$32,000.00	si
21	Probar	19	40	\$ 5,000.00	\$ 3,000.00	12	si	si	10.5	12	\$27,000.00	si
22	Probar	20	36	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	30	\$35,000.00	si

Fuente: Elaborado por el autor.

Durante la prueba se comprueba la red neuronal entrenada para ver cómo realiza la predicción de los valores de salida conocidos. Los datos de prueba son normalmente un subconjunto de los datos históricos con valores de salida conocidos. Este subconjunto no se utiliza para el entrenamiento de la red. Cuando los datos de prueba están en un conjunto de datos separado, NeuralTools empareja las variables del conjunto de datos de prueba con los datos de entrenamiento. Como sucede con el entrenamiento, NeuralTools previsualiza la configuración de la prueba antes de realizarla (Palisade, NeuralTools Manuals, 2013). Al presionar el botón de probar se nos abre una pequeña ventana donde nos muestra el nombre del conjunto de datos que va a probar, la red neuronal ya entrenada y la asignación de variables de forma automática.

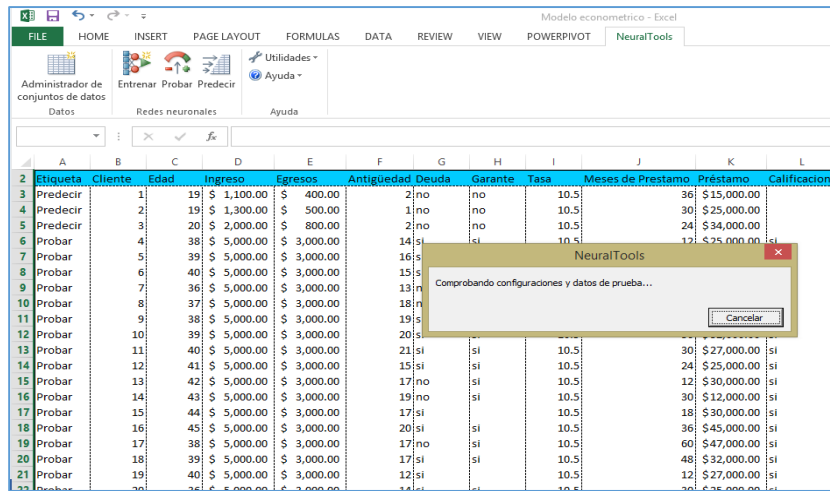
Figura 3.12 selección de prueba de la red neuronal - opciones.



Fuente: Elaborado por el autor.

Se presiona siguiente y se observa como el conjunto de datos acepta la red neuronal de entrenamiento.

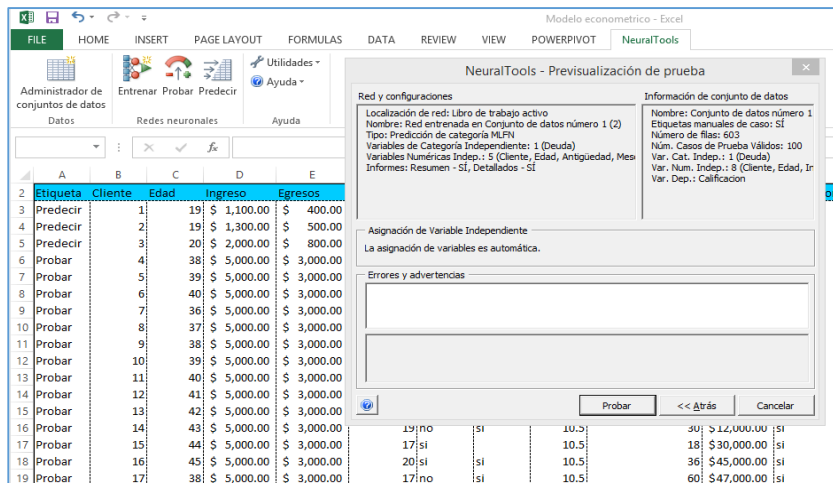
Figura 3.13 selección de prueba de la red neuronal – Carga de datos de prueba.



Fuente: Elaborado por el autor.

El modelo al aceptar el entrenamiento de la red neuronal nos muestra una ventana que se llama pre-visualización de prueba en la que nos muestra la red neuronal y sus configuraciones y la información del conjunto de datos a probar.

Figura 3.14 pre-visualización de prueba de la red neuronal.



Fuente: Elaborado por el autor.

La prueba (así como las predicciones) se ejecutan mucho más rápido que el entrenamiento. NeuralTools informa del funcionamiento de la predicción de las respuestas conocidas de los datos de prueba. Esto ayuda a ver si la red tendrá un buen funcionamiento haciendo predicciones cuando se aplique a casos con valores de salida desconocidos (Palisade, NeuralTools Manuals, 2013). Con esto termina la etapa de pruebas.

3.5.PREDICCIÓN DE LA RED NEURONAL.

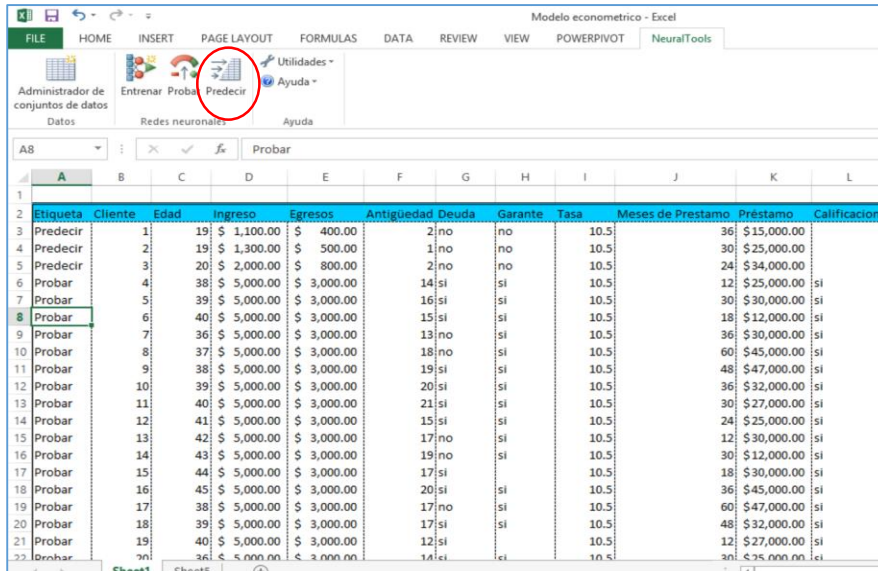
La utilidad final de una red neuronal es la predicción. La red entrenada se aplica a los nuevos casos de los que no se conocen los valores de salida, pero se quieren predecir. NeuralTools ofrece dos métodos de predicción:

1. Un método por comando para la predicción de valores de los casos de un conjunto de datos (Palisade, NeuralTools Manuals, 2013).

2. Predicción en vivo, en el que los valores de las variables independientes de un caso de la hoja de cálculo se pueden entrenar y NeuralTools calcula automáticamente el valor de salida de la predicción (Palisade, NeuralTools Manuals, 2013).

Cuando se predicen valores para un grupo de casos de un conjunto de datos, el cuadro de diálogo Predicción sirve para configurar el proceso de predicción. Puede hacer la predicción sólo para los casos a los que le faltan valores de salida y activar la predicción en vivo para hacer modificaciones en los datos para ver cómo afectan a las predicciones. Se pueden usar diferentes redes entrenadas para ver la diferencia entre los valores de la predicción. Como sucede con el entrenamiento y la prueba, NeuralTools primero previsualiza los datos y la configuración que se usarán en la predicción. Luego, las predicciones se envían a la hoja de cálculo de Excel (Palisade, NeuralTools Manuals, 2013).

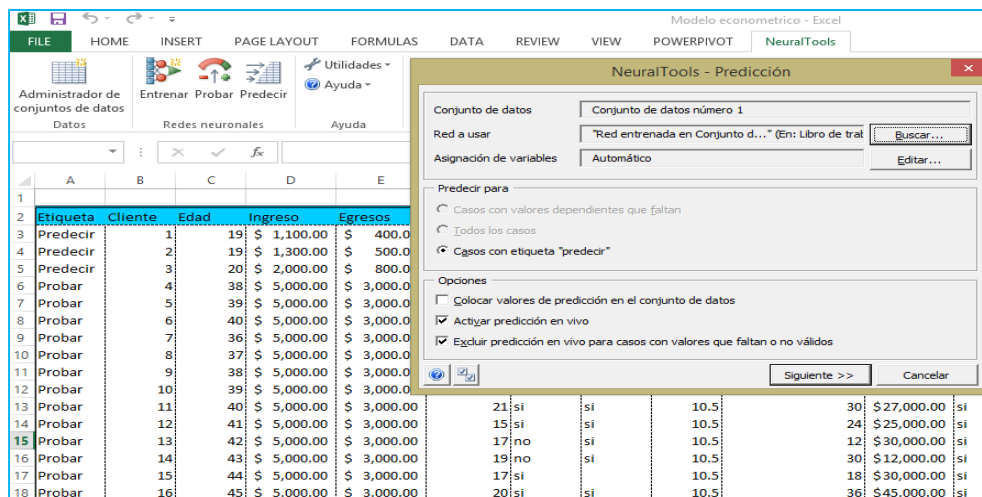
Figura 3.15 selección de predicción de la red neuronal.



Fuente: Elaborado por el autor.

Al seleccionar la opción predecir se nos abre una pequeña ventana donde nos muestra el nombre del conjunto de datos que va a probar, la red neuronal ya entrenada y probada, y la asignación de variables de forma automática. Nos indica que va a predecir solo las etiquetas que programamos para este fin, separando el conjunto de datos de entrenamiento y de pruebas.

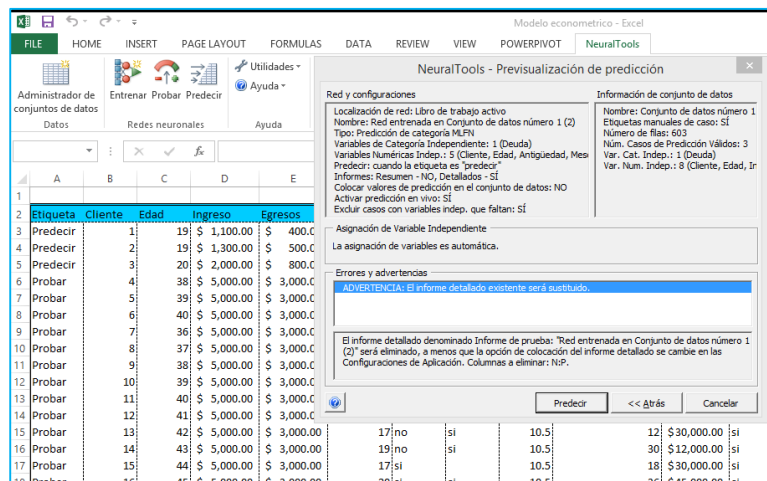
Figura 3.16 selección de predicción de la red neuronal – carga de datos.



Fuente: Elaborado por el autor.

Se cargan los datos a predecir y las pruebas ya ejecutadas para realizar la simulación correspondiente y se ordena a ejecutar la pronosticación.

Figura 3.17 pre-visualización de la predicción de la red neuronal



Fuente: Elaborado por el autor.

Arroja el resultado del pronóstico dando como evidencia de que sí se les aprueba el crédito a los 3 clientes que habían sido negados mediante la evaluación con un modelo estático. Los valores de salida de la predicción se muestran junto a los casos para los que se realiza la predicción. En esta pantalla, los valores de la predicción están en color morado. Cuando se activa la predicción en vivo, NeuralTools añade automáticamente una fórmula de Excel a la celda en la que se muestra el valor de la predicción. Esta fórmula genera el valor de predicción; por lo tanto, si cambia los valores de la variable independiente de un caso, el valor de la predicción se recalcula automáticamente (Palisade, NeuralTools Manuals, 2013). Usando la predicción en vivo puede simplemente escribir los datos de los nuevos casos directamente en Excel y generar automáticamente una nueva predicción, sin tener que pasar por el cuadro de diálogo de Predicción (Palisade, NeuralTools Manuals, 2013).

Figura 3.18 resultado de la predicción del modelo econométrico.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2		Etiqueta	Cliente	Edad	Ingreso	Egresos	Antigüedad	Deuda	Garante	Tasa	Meses de Prestamo	Préstamo	Calificación	Informe de predicción: "Red entrenada"	
3		Predecir	1	19	\$ 1,100.00	\$ 400.00	2	no	no	10.5	36	\$15,000.00		Etiqueta usada	Predicción
4		Predecir	2	19	\$ 1,300.00	\$ 500.00	1	no	no	10.5	30	\$25,000.00		predecir	si
5		Predecir	3	20	\$ 2,000.00	\$ 800.00	2	no	no	10.5	24	\$34,000.00		predecir	si
6		Probar	4	38	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	12	\$25,000.00	si		
7		Probar	5	39	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	30	\$30,000.00	si		
8		Probar	6	40	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	18	\$12,000.00	si		
9		Probar	7	36	\$ 5,000.00	\$ 3,000.00	13	no	si	10.5	36	\$30,000.00	si		
10		Probar	8	37	\$ 5,000.00	\$ 3,000.00	18	no	si	10.5	60	\$45,000.00	si		
11		Probar	9	38	\$ 5,000.00	\$ 3,000.00	19	si	si	10.5	48	\$47,000.00	si		
12		Probar	10	39	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$32,000.00	si		
13		Probar	11	40	\$ 5,000.00	\$ 3,000.00	21	si	si	10.5	30	\$27,000.00	si		
14		Probar	12	41	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	24	\$25,000.00	si		
15		Probar	13	42	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	12	\$30,000.00	si		
16		Probar	14	43	\$ 5,000.00	\$ 3,000.00	19	no	si	10.5	30	\$12,000.00	si		
17		Probar	15	44	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	18	\$30,000.00	si		
18		Probar	16	45	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$45,000.00	si		
19		Probar	17	38	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	60	\$47,000.00	si		
20		Probar	18	39	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	48	\$32,000.00	si		
21		Probar	19	40	\$ 5,000.00	\$ 3,000.00	12	si	si	10.5	12	\$27,000.00	si		
22		Probar	20	36	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	30	\$25,000.00	si		
23		Probar	21	37	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	18	\$30,000.00	si		

Fuente: Elaborado por el autor.

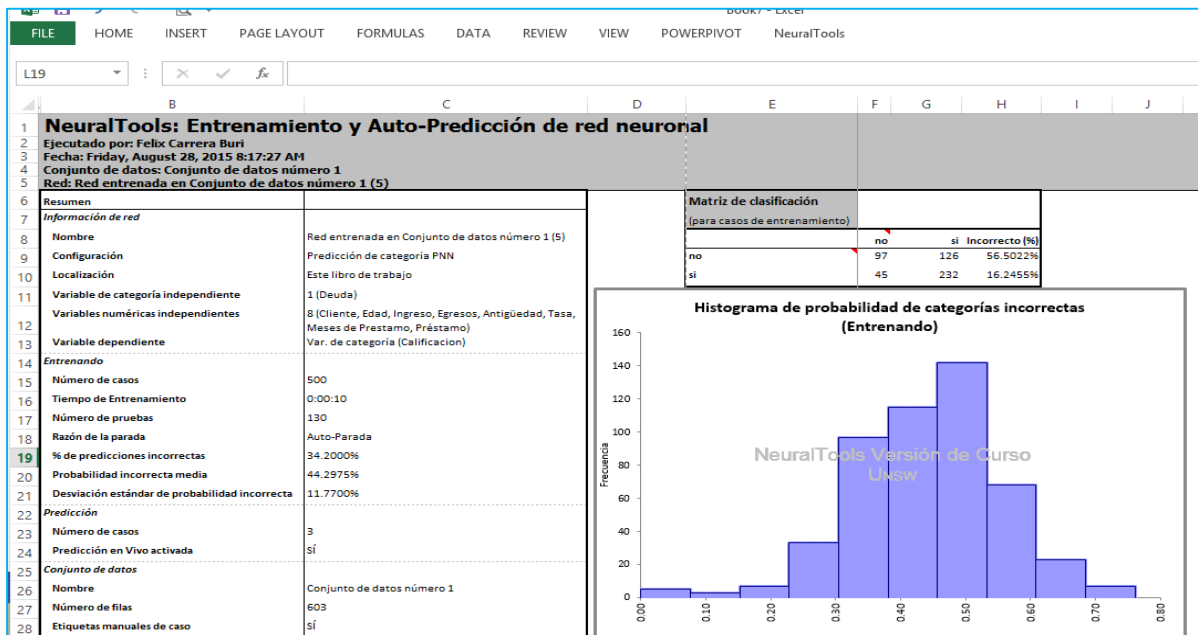
CAPÍTULO 4

4.1. IDENTIFICACIÓN DEL REPORTE.

El modelo crea informes de resumen y detallados del entrenamiento, la prueba y la predicción. Los informes de resumen se muestran en sus propias hojas de cálculo y contienen información general sobre la prueba o el entrenamiento. El informe detallado ofrece información caso por caso y se muestra junto a los datos para los que se genera el informe. Además, la mayoría de la información del informe de resumen se puede encontrar en el Informe detallado como un comentario añadido a la celda de título; esa versión del informe de resumen se conoce como resumen rápido (Palisade, NeuralTools Manuals, 2013).

Una vez ejecutado el entrenamiento de la red neuronal se visualiza a lado del conjunto de datos los resultados del entrenamiento de la red neuronal y a su vez se crea un reporte del entrenamiento de la red neuronal.

Figura 4.1 reporte del entrenamiento de la red neuronal.



Fuente: Elaborado por el autor.

Una vez ejecutado la prueba de la red neuronal se visualiza a lado del conjunto de datos los resultados de la prueba de la red neuronal y a su vez se crea un reporte de las pruebas de la red neuronal.

Figura 4.2 reporte de las pruebas de la red neuronal.

Resumen		Matriz de clasificación (para casos de prueba)		
Información de red				
Nombre	Red entrenada en Conjunto de datos número 1 (2)		no	si
Configuración	Predicción de categoría MLFN (5 nodos)		0	0
Localización	Este libro de trabajo		16	84
Variable de categoría independiente	1 (Deuda)			Incorrecto (%)
Variables numéricas independientes	5 (Cliente, Edad, Antigüedad, Meses de Prestamo, Préstamo)			0.0000%
Variable dependiente	Var. de categoría (Calificación)			
Probando				
Número de casos	100			
% de predicciones incorrectas	16.0000%			
Conjunto de datos				
Nombre	Conjunto de datos número 1			
Número de filas	603			
Etiquetas manuales de caso	sí			
Asignación de variables	Automático			
Variables de Categoría Independiente utilizadas	Nombres para entrenamiento			
Variables Numéricas Independientes utilizadas	Nombres para entrenamiento			
Variable dependiente	Var. de categoría (Calificación)			

Fuente: Elaborado por el autor.

Los informes detallados generados en NeuralTools están disponibles inmediatamente para su análisis en StatTools; de hecho, aparecen automáticamente en la lista de conjuntos de datos del Administrador de conjunto de datos de StatTools. Esto facilita el uso de StatTools para obtener resultados estadísticos adicionales a los que contienen los informes de resumen de NeuralTools (Palisade, NeuralTools Manuals, 2013). Por ejemplo, un informe de resumen de una prueba incluye un histograma de residuales (definido como las diferencias entre los valores reales y los de predicción). Basándose en el histograma, los residuales pueden aparecer como aproximadamente distribuidos normalmente. Para probar la hipótesis de una distribución normal, se puede aplicar una de las pruebas de normalidad de StatTools a la variable Residuales del Informe detallado. La función de predicción en vivo de NeuralTools facilita la visualización del efecto de los cambios de los valores independientes sobre la predicción. Con la predicción en vivo, se pueden usar otras funciones disponibles en Excel para explorar la relación entre las variables independientes y la dependiente (Palisade, NeuralTools Manuals, 2013).

4.2. ANÁLISIS DEL REPORTE.

Tabla 4.2. – Reporte de NeuralTools al ejecutar al modelo

NeuralTools: Entrenamiento, Auto-Prueba y Auto-Predicción de red neuronal

Ejecutado por: Felix Carrera Buri

Fecha: Friday, August 28, 2015 9:32:27 AM

Conjunto de datos: Conjunto de datos número 1

Resumen

Información de red

Nombre	Red entrenada en Conjunto de datos número 1 (5)
Configuraciones incluidas en la búsqueda	PNN, MLFN 2 nodos
Mejor configuración	Predicción de categoría PNN
Localización	Este libro de trabajo
Variable de categoría independiente	1 (Deuda)
Variables numéricas independientes	8 (Cliente, Edad, Ingreso, Egresos, Antigüedad, Tasa, Meses de Prestamo, Préstamo)
Variable dependiente	Var. de categoría (Calificación)

Entrenando

Número de casos	500
Tiempo de Entrenamiento	0:00:06
Número de pruebas	130
Razón de la parada	Auto-Parada
% de predicciones incorrectas	34.2000%
Probabilidad incorrecta media	44.2975%
Desviación estándar de probabilidad incorrecta	11.7700%

Probando

Número de casos	100
% de predicciones incorrectas	29.0000%
Probabilidad incorrecta media	44.3014%
Desviación estándar de probabilidad incorrecta	9.8643%

Predicción

Número de casos	3
-----------------	---

Predicción en Vivo activada	Sí
Conjunto de datos	
Nombre	Conjunto de datos número 1
Número de filas	603
Etiquetas manuales de caso	Sí

Al ejecutar el reporte general del modelo se encuentra la siguiente lectura:

Información de red.

Se encuentra el nombre dado por defecto, en este caso red de entrada de conjunto de datos número 1, en las configuraciones incluidas en la búsqueda se encontraron que las mejores redes halladas son las redes de tipo PNN y MLFN en 2 a 6 nodos lo cual significa que entre el segundo piso o capa de la red neuronal hasta el sexto nodo se encontró el patrón de comportamiento que definió al modelo para ejecutar su decisión pero que en la mejor predicción se la lleva la red de categoría PNN.

El reporte también da la ubicación de los datos, en este reporte por obvias razones da la ubicación del mismo libro pero si se usa fuentes externas o base de datos de servidores dará la ubicación exacta del origen de extracción de los datos.

En la lectura del reporte se revela las variables usadas en el modelo divididas por el tipo de variables y discretizadas a su vez por las categorías implicadas en la ejecución del modelo, se observa la variable de categorías independientes “deuda” la cual es el historial crediticio de los clientes en la base suministrada, las variables numéricas independientes y al final la variable dependiente la cual rige al modelo que es la calificación de aprobación o no aprobación del crédito.

Tabla 4.2. – Reporte de NeuralTools , búsqueda de mejor red.

Búsqueda de mejor red			
	% incorrecto	Tiempo de Entrenamiento	Razón de parada de entrenamiento
PNN	29.00%	0:00:06	Auto-Parada
MLFN 2 nodos	73.00%	0:04:13	Parado por el usuario

En la tabla 2 se observa que la mejor red seleccionada para este tipo de modelo es la red de tipo PNN con un porcentaje de 29% de probabilidad incorrecta de predicción versus a una red MLFN que posee un porcentaje de 73% de probabilidad incorrecta de predicción.

Sección de entrenamiento.

Dota al informe del número de casos usados para el entrenamiento el cual fueron 500 casos de los 600, es decir el 83.40% del total de observaciones de la base de datos, con un tiempo de entrenamiento de seis segundos en aprendizaje y 130 pruebas en que las que se encontró el patrón de repetición o de comportamiento no variable o con menos variabilidad. La razón de la parada se debe a que se limita a un tiempo establecido al entrenamiento de la red neuronal, dentro del entrenamiento se hace una pequeña prueba de predicciones pero solo muestra el porcentaje de las incorrectas la cual revela un resultado de 34,20% dando una probabilidad incorrecta media de 44.30%, esto indica que el entrenamiento va a acomodar el modelo para que tenga una mejor distribución de probabilidades para conseguir una desviación estándar de probabilidad incorrecta con un 11.77%.

Tabla 4.3. – Reporte de NeuralTools , matriz de clasificación para los casos de entrenamiento.

Matriz de clasificación			
<i>(para casos de entrenamiento)</i>			
	no	si	
no	97	126	Incorrecto (%) 56.5022%
si	45	232	16.2455%

En el entrenamiento se observa que de los 500 casos para entrenar se subdividieron entre los que se les otorgó el crédito mediante un modelo estático y a quienes no distribuidos de la siguiente forma: de los 223 casos que no se les otorgó el crédito, el modelo de entrenamiento predijo que a 126 casos sí se les podía hacer el préstamo según el patrón de comportamiento y que de los 277 que se les otorgó el crédito, 45 no se debía otorgar ya que según el historial crediticio tienen la capacidad de pago pero poseen morosidad en la cancelación de sus cuotas.

Sección probando.

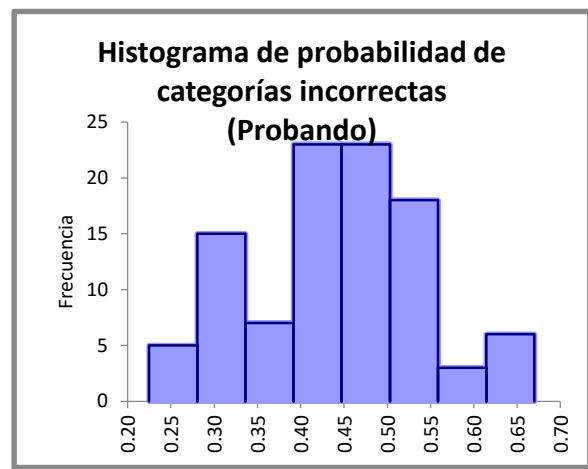
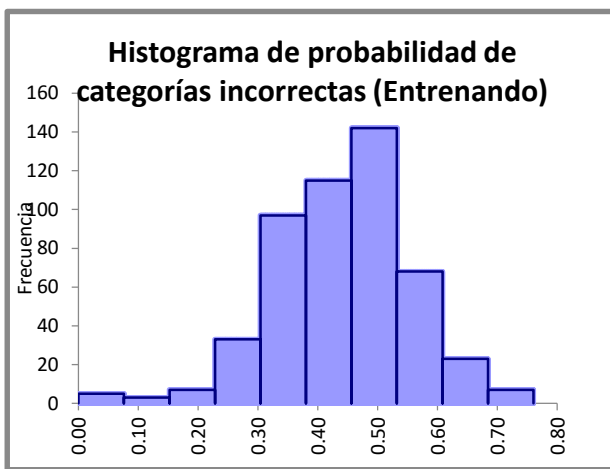
Dota al informe del número de casos usados para las pruebas las cuales fueron 100 casos de los 600, es decir el 16.60% del total de observaciones de la base de datos, en la predicción del modelo arroja un 29.00% de predicciones incorrectas con una probabilidad incorrecta media de 44.30%, esto indica que las pruebas va a acomodar el modelo para que tenga una mejor distribución de probabilidades para conseguir una desviación estándar de probabilidad incorrecta con un 9.86%.

Tabla 4.4. – Reporte de NeuralTools , matriz de clasificación para los casos de prueba.

Matriz de clasificación		Red	
<i>(para casos de prueba, modelo estático)</i>			
	no	si	
no	0	0	Incorrecto (%) 0.0000%
si	29	71	29.0000%

El modelo ya corregido por el entrenamiento lanza las pruebas para la predicción y revela que no existen clientes evaluados con categorías de préstamos no otorgados pero que a los que les otorgaron el crédito hay 29 casos de las 100 de pruebas que no se debía otorgar por el tema de historial crediticio.

4.1 Histogramas de probabilidad de categorías incorrectas tanto en entrenamientos como en pruebas.



4.3. RESULTADO DEL REPORTE

Tabla 4.5. – Reporte de NeuralTools , Informe de Entrenamiento-Prueba-Predicción de "Red entrenada en Conjunto de datos número 1".

Edad	Meses de Prestamo	Préstamo	Calificacion	Etiqueta usada	Predicción	Predicción%	Incorrecto%	Correcto /Incorrecto
19	36	\$ 15,000.00		predecir	si	74.19%		
19	30	\$ 25,000.00		predecir	si	93.70%		
20	24	\$ 34,000.00		predecir	si	98.85%		
38	12	\$ 25,000.00	si	Probar	si	68.93%	31.07%	Correcto
39	30	\$ 30,000.00	si	Probar	si	58.18%	41.82%	Correcto
40	18	\$ 12,000.00	si	Probar	si	76.04%	23.96%	Correcto
36	36	\$ 30,000.00	si	Probar	si	59.18%	40.82%	Correcto
37	60	\$ 45,000.00	si	Probar	si	50.80%	49.20%	Correcto
38	48	\$ 47,000.00	si	Probar	no	51.03%	51.03%	Incorrecto
39	36	\$ 32,000.00	si	Probar	si	52.03%	47.97%	Correcto
40	30	\$ 27,000.00	si	Probar	si	57.27%	42.73%	Correcto
41	24	\$ 25,000.00	si	Probar	si	70.94%	29.06%	Correcto
42	12	\$ 30,000.00	si	Probar	si	55.95%	44.05%	Correcto
43	30	\$ 12,000.00	si	Probar	si	57.23%	42.77%	Correcto
44	18	\$ 30,000.00	si	Probar	no	53.10%	53.10%	Incorrecto
45	36	\$ 45,000.00	si	Probar	si	53.25%	46.75%	Correcto
38	60	\$ 47,000.00	si	Probar	si	57.78%	42.22%	Correcto
39	48	\$ 32,000.00	si	Probar	no	51.78%	51.78%	Incorrecto
40	12	\$ 27,000.00	si	Probar	si	76.37%	23.63%	Correcto
36	30	\$ 25,000.00	si	Probar	si	68.29%	31.71%	Correcto
37	18	\$ 30,000.00	si	Probar	si	55.64%	44.36%	Correcto
38	36	\$ 12,000.00	si	Probar	si	68.55%	31.45%	Correcto
39	60	\$ 30,000.00	si	Probar	no	60.23%	60.23%	Incorrecto
40	48	\$ 45,000.00	si	Probar	no	51.26%	51.26%	Incorrecto
41	36	\$ 47,000.00	si	Probar	si	55.33%	44.67%	Correcto
37	30	\$ 32,000.00	si	Probar	si	50.75%	49.25%	Correcto
38	24	\$ 27,000.00	si	Probar	si	53.44%	46.56%	Correcto

39	12	\$ 25,000.00	si	Probar	si	51.18%	48.82%	Correcto
40	36	\$ 30,000.00	si	Probar	si	66.97%	33.03%	Correcto

En esta sección final se muestra que de los 3 casos a predecir, los 3 se les puede otorgar el crédito según las probabilidades de éxito expuestas en la tabla 4. Este modelo econométrico abre oportunidades a personas sin historial creditico para la obtención de préstamos para inversiones de activos. No hay la necesidad de garantes cuando la institución financiera se hace cargo del activo como garantía. Se da como nuevo mercado objetivo a clientes jóvenes que empiezan su vida laboral. Se abre un nuevo portafolio de posibles clientes con fidelidad a las instituciones financieras que evalúen mediante esta metodología. El margen de error del modelo neuronal es mínimo frente al margen de error de los modelos estáticos, el modelo econométrico neuronal se basa en un algoritmo de razonamiento lógico. La negación de crédito más negado ha sido por la incapacidad de pagos puntuales, el patrón neuronal repetitivo está basado en el algoritmo de la red neuronal en este caso es la red PNN.

CONCLUSIONES

Este tipo de modelo investigativo no asegura que todo aquel que se someta a su análisis vaya a salir calificado pero si disminuye el margen de error de la calificación el cual persiste en modelos estáticos.

Con este tipo de metodología se podrá ser testigos futuros de encontrar ciudadanos de edades entre 19 a 24 años empezando una vida laboral en que quieran ser partícipes de un crédito hipotecario, automotriz, estudiantil o cualquiera que sea su fin.

El modelo econométrico redes neuronales artificiales tiene muchas ventajas de oportunidad de espacio en el mercado financiero debido a que están basadas en la estructura del comportamiento del consumidor.

Posee una la naturaleza de beneficios que creará ante una sociedad que puede que carezca de requisitos tradicionales para la evaluación de créditos y prestaciones mediante un calificativo de aprobación y no aprobación.

El modelo econométrico redes neuronales artificiales expande su mercado objetivo logrando una diversificación de nuevos productos para la comunidad.

La comunidad experimentará la sensibilidad de las aprobaciones de créditos y microcréditos aprobados según su capacidad de pago.

La metodología neuronal no es muy efectiva con pequeñas instituciones financieras como son las cooperativas de ahorro y crédito por las pocas variables que se manejan.

La metodología neuronal es costosa para ciertas instituciones que no poseen bases de datos robustas.

Con la metodología neuronal podrá dejar de lado los modelos estáticos de calificación crediticia y pasar a modelos dinámicos que permitan una optimización en las inversiones de las instituciones financieras.

Se debe dar un mantenimiento semestral al modelo econométrico de redes neuronales ya que las bases de datos usadas se van actualizando mediante sus variables a través del tiempo cambiando parámetros de consideración para la evaluación crediticia.

RECOMENDACIONES

La recomendación que se puede en este tipo de investigación es la de demostrar la necesidad de un cambio en la matriz de investigación crediticia a nivel financiero para que haya más oportunidades en la población, para que así, se logre las prestaciones de microcréditos a mayor escala y abrir un nuevo mercado de clientes.

No se recomienda este tipo de modelos a aquellas instituciones financieras que posean un bajo número de clientes como por ejemplo las cooperativas de crédito y ahorro por el costo que implica estos modelos, sí se desea buscar un nicho de mercado para este tipo de metodologías, se puede aplicar para instituciones financieras de tipo Bancos grandes y medianos.

En el caso de las aseguradoras, este tipo de metodología es muy recomendable ya que les puede dar una mejor percepción del cobro del servicio de seguro y un cálculo óptimo de las primas de seguro.

Para los corredores de bolsa de valores este tipo de metodología es muy recomendable ya que les permite tomar decisiones más acertadas en cuestiones de comportamiento de las inversión o ventas de acciones y así disminuir el riesgo.

Se recomienda que el agente a ocupar la administración del modelo econométrico neuronal deba ser una persona con amplio conocimiento estadístico y financiero que le permita decidir los resultados del modelo ya que a final de cuentas todo cae en el rango del investigador.

REFERENCIAS

Armstrong&Kotler. (2008). *Fundamentos del Marketing*. Bogotá: Editorial Universidad del Rosario.

BAHAMÓN, R. V. (2013). *bdigital*. Retrieved from Unal:

<http://www.bdigital.unal.edu.co/46564/1/08901050.2013.pdf>

Mack&Lackes. (1998). *Neuronal Networks*. Berlin: Heidelberg.

Palisade. (2013). *NeuralTools Manuals*. Retrieved from Palisade Corp.:

http://www.palisade.com/downloads/manuals/ES/NeuralTools5_ES.pdf

Palisade. (2013). *NeuralTools Testing version*. Retrieved from Palisade Corp.:

<Http://www.palisade-lta.com/neuraltools/>

Palisade. (2013). *NeuralTools Tutorials*. Retrieved from Palisade Corp: [http://www.palisade-](http://www.palisade-lta.com/risk/5/tips/es/gs/)

[lta.com/risk/5/tips/es/gs/](http://www.palisade-lta.com/risk/5/tips/es/gs/)

Palisade. (2014). *NeuralTools Webcasts*. Retrieved from Palisade Corp: [http://www.palisade-](http://www.palisade-lta.com/seminarios/webcasts.asp)

[lta.com/seminarios/webcasts.asp](http://www.palisade-lta.com/seminarios/webcasts.asp)

Palisade. (2015). *NeuralTools Academic Newsletters*. Retrieved from Palisade Corp:

<http://www.palisade.com/academic/Updates.asp>

Palisade. (2015). *NeuralTools Academic software*. Retrieved from Palisade Corp:

<http://www.palisade.com/academic/Updates.asp>

Torra&Monte. (2013). *Modelos neuronales aplicados a la economía*. Barcelona.

ANEXOS

FORMA AUTOMÁTICA PARA LA EJECUCIÓN DEL MODELO.

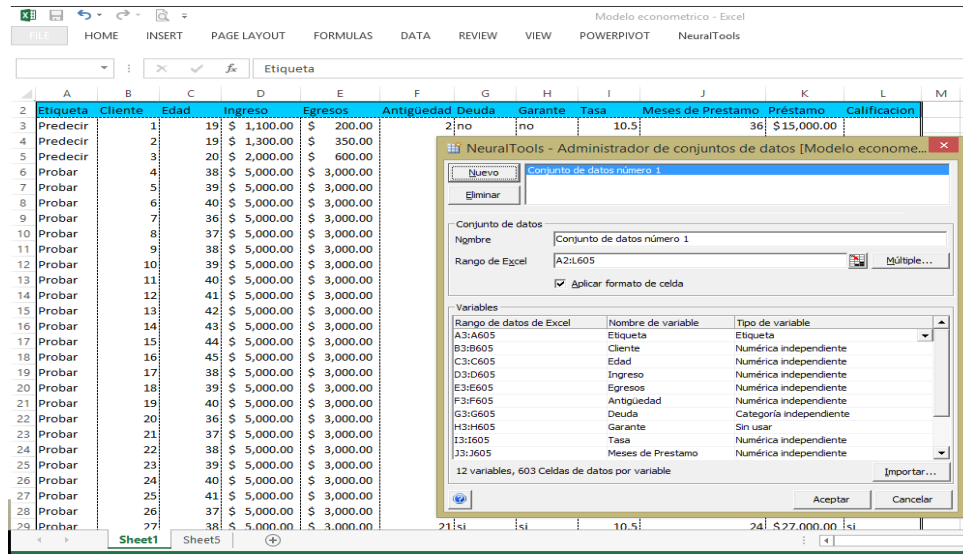
1. Seleccionar y cargar el conjunto de datos a usar.

Anexo 1 – Selección de datos.

The screenshot shows the Microsoft Excel interface with the 'Administrador de conjuntos de datos' (Data Source Administrator) dialog box open. The dialog box is titled 'Administrador de conjuntos de datos' and contains the text 'Seleccionar el rango del conjunto de datos.' Below this text is an input field containing the formula '= \$A\$2:\$L\$605'. There are 'OK' and 'Cancel' buttons at the bottom of the dialog box. The background spreadsheet shows a data table with columns labeled 'Etiqueta', 'Cliente', 'Edad', 'Ingreso', 'Egresos', 'tamo', 'Préstamo', and 'Calificacion'. The data rows are numbered 2 through 28, and the dialog box is positioned over the data range from row 2 to row 605.

Fuente: Elaborado por el autor

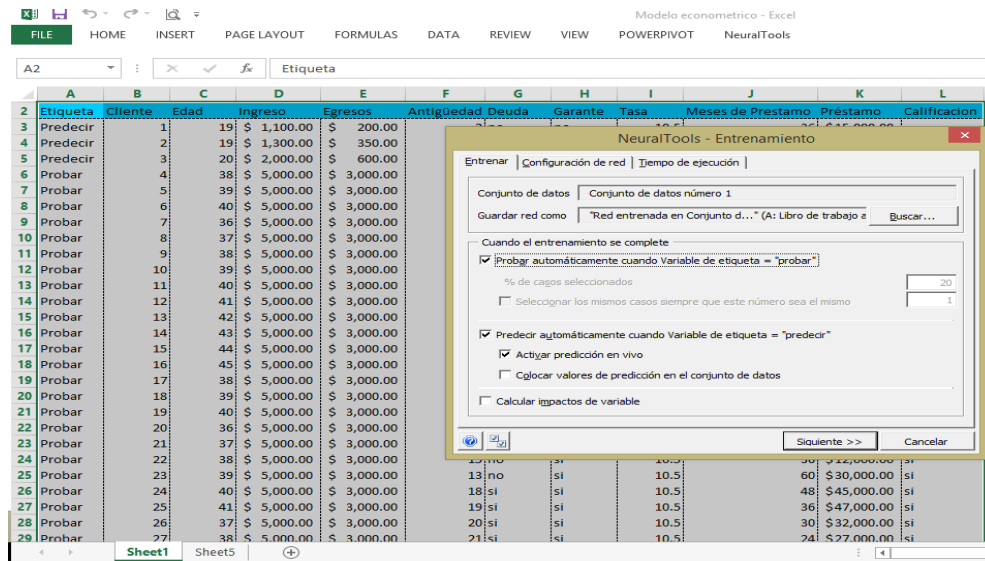
Anexo 2 – Carga del conjunto de datos



Fuente: Elaborado por el autor.

2. Seleccionar el entrenamiento del conjunto de datos y activar la opción “probar automáticamente cuando este la variable de etiqueta = Probar”.

Anexo 3 – Configuración automática de parámetros de entrenamiento



Fuente: Elaborado por el autor.

- Se pre-visualiza las configuraciones automáticas del entrenamiento y pruebas y se ejecuta.

Anexo 4 – Pre-visualización del entrenamiento del conjunto de datos.

Fuente: Elaborado por el autor.

Anexo 5 – Entrenamiento y prueba automática del conjunto de datos.

Fuente: Elaborado por el autor.

Anexo 6 – Informe del entrenamiento, prueba y predicción automática del conjunto de datos.

NeuralTools: Entrenamiento, Auto-Prueba y Auto-Predicción de red neuronal

Ejecutado por: Felix Carrera Buri
 Fecha: Wednesday, September 2, 2015 2:00:55 AM
 Conjunto de datos: Conjunto de datos número 1
 Red: Red entrenada en Conjunto de datos número 1 (4)

Resumen	
Información de red	
Nombre	Red entrenada en Conjunto de datos número 1 (4)
Configuraciones incluidas en la búsqueda	PNN, MLFN 2 a 6 nodos
Mejor configuración	Predicción de categoría PNN
Localización	Este libro de trabajo
Variable de categoría independiente	1 (Deuda)
Variables numéricas independientes	8 (Cliente, Edad, Ingreso, Egresos, Antigüedad, Tasa, Meses de Prestamo, Préstamo)
Variable dependiente	Var. de categoría (Calificación)
Entrenando	
Número de casos	500
Tiempo de Entrenamiento	0:00:14
Número de pruebas	130
Razón de la parada	Auto-Parada
% de predicciones incorrectas	34.2000%
Probabilidad incorrecta media	44.2975%
Desviación estándar de probabilidad incorrecta	11.7700%
Probando	
Número de casos	100
% de predicciones incorrectas	29.0000%
Probabilidad incorrecta media	44.3014%
Desviación estándar de probabilidad incorrecta	9.8643%

Fuente: Elaborado por el autor.

Anexo 7 – Informe de la predicción automática del conjunto de datos.

Informe de Entrenamiento-Prueba-Predicción de "Red entrenada en Conju

	Ingreso	Egresos	Antigüedad	Deuda	Garante	Tasa	Meses de Prestamo	Préstamo	Calificación	Etiqueta usada	Predicción	Predicción%	Incorrecto%	Correcto/Incorrecto
3	\$ 1,100.00	\$ 200.00	2	no	no	10.5	36	\$15,000.00		predecir	si	74.19%		
4	\$ 1,300.00	\$ 350.00	1	no	no	10.5	30	\$25,000.00		predecir	si	93.70%		
5	\$ 2,000.00	\$ 600.00	2	no	no	10.5	24	\$34,000.00		predecir	si	98.85%		
6	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	12	\$25,000.00	si	probar	si	68.93%	31.07%	Correcto
7	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	30	\$30,000.00	si	probar	si	58.18%	41.82%	Correcto
8	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	18	\$12,000.00	si	probar	si	76.04%	23.96%	Correcto
9	\$ 5,000.00	\$ 3,000.00	13	no	si	10.5	36	\$30,000.00	si	probar	si	59.18%	40.82%	Correcto
10	\$ 5,000.00	\$ 3,000.00	18	no	si	10.5	60	\$45,000.00	si	probar	si	50.80%	49.20%	Correcto
11	\$ 5,000.00	\$ 3,000.00	19	si	si	10.5	48	\$47,000.00	si	probar	no	51.03%	51.03%	Incorrecto
12	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$32,000.00	si	probar	si	52.03%	47.97%	Correcto
13	\$ 5,000.00	\$ 3,000.00	21	si	si	10.5	30	\$27,000.00	si	probar	si	57.27%	42.73%	Correcto
14	\$ 5,000.00	\$ 3,000.00	15	si	si	10.5	24	\$25,000.00	si	probar	si	70.94%	29.06%	Correcto
15	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	12	\$30,000.00	si	probar	si	55.95%	44.05%	Correcto
16	\$ 5,000.00	\$ 3,000.00	19	no	si	10.5	30	\$12,000.00	si	probar	si	57.23%	42.77%	Correcto
17	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	18	\$30,000.00	si	probar	no	53.10%	53.10%	Incorrecto
18	\$ 5,000.00	\$ 3,000.00	20	si	si	10.5	36	\$45,000.00	si	probar	si	53.25%	46.75%	Correcto
19	\$ 5,000.00	\$ 3,000.00	17	no	si	10.5	60	\$47,000.00	si	probar	si	57.78%	42.22%	Correcto
20	\$ 5,000.00	\$ 3,000.00	17	si	si	10.5	48	\$32,000.00	si	probar	no	51.78%	51.78%	Incorrecto
21	\$ 5,000.00	\$ 3,000.00	12	si	si	10.5	12	\$27,000.00	si	probar	si	76.37%	23.63%	Correcto
22	\$ 5,000.00	\$ 3,000.00	14	si	si	10.5	30	\$25,000.00	si	probar	si	68.29%	31.71%	Correcto
23	\$ 5,000.00	\$ 3,000.00	16	si	si	10.5	18	\$30,000.00	si	probar	si	55.64%	44.36%	Correcto

Fuente: Elaborado por el autor.