

UNIVERSIDAD NACIONAL AGRARIA

LA MOLINA

ESCUELA DE POSGRADO

MAESTRÍA DE ESTADÍSTICA APLICADA



**“APLICACIÓN DE REDES NEURONALES Y REGRESIÓN
LOGÍSTICA PARA PREDECIR EL ÉXITO DE LA COMPRA
DE DEUDA DE UNA ENTIDAD FINANCIERA”**

Presentada por:

EMANUEL GUILLERMO MUÑOZ MUÑOZ

TESIS PARA OPTAR EL GRADO DE MAGISTER SCIENTIAE EN
ESTADÍSTICA APLICADA

Lima – Perú
2015

ÍNDICE

| | |
|---|----|
| II. INTRODUCCIÓN..... | 8 |
| 2.1. JUSTIFICACIÓN | 11 |
| 2.2. OBJETIVOS | 12 |
| 2.2.1. Objetivo General | 12 |
| 2.2.2. Objetivos Específicos..... | 12 |
| III. REVISIÓN DE LITERATURA | 5 |
| 3.1. Descubrimiento de conocimiento en base de datos. | 5 |
| 3.2. Procesos de Minería de Datos CRISP-DM..... | 7 |
| 3.3. Sistemas de bases de datos y Data Warehouse. | 7 |
| 3.4. Pre procesamiento..... | 8 |
| 3.5. Detección de Outliers..... | 9 |
| 3.6. Aprendizaje de máquina | 10 |
| 3.7. Clasificación | 10 |
| 3.8. Una Neurona | 11 |
| 3.9. Redes Neuronales..... | 12 |
| 3.10. Perceptrón Multicapas (MLP)..... | 14 |
| 3.11. Softmax | 15 |
| 3.12. Método de retención..... | 16 |
| 3.13. Validación Cruzada..... | 16 |
| 3.14. Aplicación de redes neuronales en entidades financieras | 17 |
| IV. MATERIALES Y MÉTODOS..... | 19 |
| 4.1. De oficina y cómputo..... | 19 |

| | | |
|-------|---|----|
| 4.2. | Datos del departamento de marketing de la entidad financiera | 19 |
| 4.3. | Métodos..... | 20 |
| ▪ | Arquitectura | 21 |
| ▪ | Número de neuronas de la capa de entrada..... | 21 |
| ▪ | Número de neuronas de la capa de salida | 21 |
| ▪ | Número de capas ocultas | 21 |
| ▪ | Número de neuronas en la capa escondida | 21 |
| ▪ | Valor neto del perceptrón..... | 22 |
| ▪ | Producir función de transferencia | 22 |
| ▪ | Producir factor de aprendizaje | 22 |
| ▪ | Producir regla de aprendizaje..... | 23 |
| ▪ | Algoritmo de aprendizaje propagación hacia atrás (backpropagation)..... | 23 |
| V. | RESULTADOS Y DISCUSIÓN | 25 |
| 5.1. | Resultados | 35 |
| 5.2. | Resultados consolidados | 41 |
| 5.3. | Validación Cruzada..... | 42 |
| VI. | CONCLUSIONES | 45 |
| VII. | RECOMENDACIONES | 47 |
| VIII. | REFERENCIAS BIBLIOGRÁFICAS | 48 |
| IX. | ANEXOS | 52 |
| 9.1. | ANEXO 1..... | 52 |
| 9.2. | ANEXO 2..... | 52 |
| 9.3. | ANEXO 3..... | 59 |
| 9.4. | ANEXO 4..... | 66 |
| 9.5. | ANEXOS 5 | 71 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Pasos para realizar el descubrimiento de información en una base de datos. Fuente: Han et al. (2006). | 5 |
| Figura 2: Integración de las técnicas de minería de datos con otras áreas. Fuente: Han et al. (2006). | 6 |
| Figura 3: Pasos iterativos de la aplicación del modelo CRISP-DM. Fuente: Makhabel (2015). | 7 |
| Figura 4: Marco de trabajo de un data warehouse. Fuente: Han et al. (2006). | 8 |
| Figura 5: Diagrama perceptron simplificado. Fuente: Dean (2014). | 11 |
| Figura 6: Esquema de una red neuronal con una sola capa oculta. Fuente: Dean (2014). | 13 |
| Figura 7: Patrones separados en un hiperplano. Fuente: Dean (2014). | 14 |
| Figura 8: Estructura del perceptron multicapa. Fuente: Dean (2014). | 15 |
| Figura 9: Método de retención o holdout method para validar un modelo. Fuente: Han et al. (2006). | 16 |
| Figura 10: Método de la validación cruzada. Fuente: Bramer (2007). | 17 |
| Figura 11: Análisis de los valores perdidos. | 27 |
| Figura 12: Análisis de los valores perdidos | 28 |
| Figura 13: Matriz de correlación. | 29 |
| Figura 14: Gráficos exploratorios de las variables consideradas. | 30 |
| Figura 15: Gráficos exploratorios de las variables consideradas. | 31 |
| Figura 16: Estructura de la red neuronal entrenada utilizando un perceptrón multicapa. | 36 |
| Figura 17: Estructura de la red neuronal entrenada utilizando un perceptron multicapa. | 37 |
| Figura 18: Eficiencia de predicción del método de Redes Neuronales Multicapa. | 42 |
| Figura 19: Eficiencia de predicción del método regresión logística. | 43 |
| Figure 20: Aplicación de red neuronal utilizando el algoritmo NEAT (Neuro Evolución de Topologías Aumentativas). | 72 |

ÍNDICE DE CUADROS

| | |
|--|----|
| Cuadro 1: Variables utilizadas en la investigación. | 20 |
| Cuadro 2: Dimensión de los datos. | 25 |
| Cuadro 3: Resumen de los datos de cada variables. | 26 |
| Cuadro 4: Resumen de los datos de cada variables. | 26 |
| Cuadro 5: Columna donde existen datos perdidos. | 28 |
| Cuadro 6: Porcentaje de valores perdidos en el total de los datos. | 28 |
| Cuadro 7: Números de datos perdidos por variables. | 28 |
| Cuadro 8: Cantidad de datos sin valores perdidos. | 28 |
| Cuadro 9: Transformaciones aplicadas a la variable Importe. | 29 |
| Cuadro 10: Tabla de variables independientes y variable dependiente con cuatro clases. | 31 |
| Cuadro 11: Tabla de variables independientes y variable dependiente con dos clases. | 32 |
| Cuadro 12: Tabla de variables independientes y variable dependiente con cuatro clases. | 32 |
| Cuadro 13: Tabla de variables independientes y variable dependiente con dos clases. | 33 |
| Cuadro 14: 74 pesos correspondientes a cada conexión de la red neuronal. | 37 |
| Cuadro 15: 58 pesos correspondientes a cada conexión de la red neuronal. | 38 |
| Cuadro 16: Tabla de predicciones con la variable de cuatro clases. | 38 |
| Cuadro 17: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados. | 39 |
| Cuadro 18: Tabla de predicciones con la variable de dos clases. | 39 |
| Cuadro 19: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados. | 39 |
| Cuadro 20: Tabla de importancia de las variables. | 40 |
| Cuadro 21: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados. | 40 |
| Cuadro 22: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados. | 41 |
| Cuadro 23: Tabla de resultados consolidados. | 41 |

| | |
|--|----|
| Cuadro 24: Datos preparados para aplicar la validación cruzada en la red neuronal multicapa. | 42 |
| Cuadro 25: Datos preparados para aplicar la validación cruzada en la regresión logística multinomial..... | 43 |
| Cuadro 26: Porcentaje de eficiencia de predicción consolidado..... | 44 |

I. RESUMEN

El propósito de este trabajo es presentar y aplicar la técnica de redes neuronales para predecir el éxito de la compra de deuda de una entidad financiera a otra. La técnica de redes neuronales se sustenta en el perceptron multicapa y en el algoritmo de retropropagación. Se explica el uso de la técnica mostrando sus diferentes pasos: establecimiento de la estructura, la función de activación (sigmoidea), el paradigma de aprendizaje, el factor de aprendizaje, la regla de aprendizaje, el algoritmo de aprendizaje (retropropagación), el entrenamiento y la evaluación de la red neuronal.

Se realizó un pre procesamiento para tener datos de calidad, con estos datos limpios se aplicó la técnica de redes neuronales en la clasificación y para evaluar esta clasificación se realizó el post procesamiento, se puede notar que todo este proceso es minería de datos. Lo más resaltante en las redes neuronales son los valores hallados para los parámetros, identificados a través del entrenamiento. Estos parámetros son denominados pesos dinámicos, en estos pesos está el conocimiento de la red neuronal para poder lograr la predicción.

El error de clasificación que se obtiene al aplicar a los datos de prueba a la red neuronal entrenada con las especificaciones ya señaladas fue de 22.89% y 4.31% para la red de cuatro y dos neuronas en la capa de salida respectivamente. Al probar con más de siete neuronas se obtuvieron errores de clasificaciones similares o mayores, esto se logró con un mayor costo computacional. Al comparar los resultados de los mismos datos con la regresión logística, se obtuvieron estos errores de clasificación 22.72% y 4.31% para los datos con la variable respuesta de cuatro y dos clases respectivamente.

También evaluaron los modelos de clasificación con la técnica de validación cruzada, en este caso se obtuvo 76.57% y 77.29% que son los porcentajes de la eficiencia de predicción de redes neuronales y regresión logística respectivamente, solo para los casos donde la variable respuesta de los datos era de cuatro niveles.

Palabras Claves: Perceptron, red neuronal, sigmoidea, retropropagación, regresión logística, clasificación, validación cruzada.

SUMMARY

The purpose of this paper is to present and apply the technique of neural networks in order to predict the success of the debt purchase of a financial entity to another. The neural network technique is based on the multilayer perceptron and back propagation algorithm. The use of the technique explains showing different steps: Establishing the structure, the activation function (sigmoid), the learning paradigm, the learning factor, the learning rule, the learning algorithm (back propagation), training and the neural network evaluation.

A pre-processing for having data quality, with these clean data was applied the neural networks technique the classification and for evaluating this classification the post processing was performed, one can see that this process is data mining. The highlights in neural networks are the values found for the parameters identified through training. These parameters are called dynamic weights, these weights is the knowledge of the neural network to accomplish the prediction.

The classification error obtained by applying to the test data to the neural network trained with the specifications already mentioned was 22.89% and 4.31% for the network of four two neurons in the output layer respectively. When testing with more than seven neurons errors similar or higher ratings were obtained, this was achieved with a greater computational cost. Comparing the results of the same data with logistic regression, these classification errors were obtained 22.89% and 4.31% for the response variable data with four and two kinds respectively.

In addition, classification models with the cross-validation technique, in this case 76.57% and 77.29% are percentages prediction efficiency logistic regression and neural networks respectively, only for cases evaluated was obtained where the variable response data was four levels.

Keywords: Perceptron, neural network, sigmoid, back propagation, logistic regression, classification, cross-validation.

II. INTRODUCCIÓN

Una institución financiera con más de 100 años en el mercado en todo el Perú utiliza la estrategia de fidelizar a sus clientes ofreciéndoles el servicio de la compra de deuda con una menor tasa de interés que tienen en otras instituciones financieras a nivel nacional. Para realizar la compra de esta deuda previamente se realiza una solicitud para saber si es o no posible concretarla. La entidad tiene pérdidas innecesarias en el pago de las solicitudes no aceptadas por las otras instituciones, por ello a través del departamento de marketing genera un reporte del historial de la compra de deuda de los clientes que han sido aceptadas y rechazadas. Hasta el momento solo se había realizado un análisis descriptivo con estos datos.

Con la ayuda de la minería de datos se dio un paso adelante para crear un modelo estadístico que ayude a predecir una compra de deuda, todo esto se realizó en tres etapas: pre procesamiento, clasificación o predicción y post procesamiento. En lo que respecta a la clasificación es discriminar a nuevos individuos en los grupos ya preestablecidos con la ayuda de una función.

La primera etapa de la minería de datos comprende la limpieza, y detección de outliers, el análisis de datos perdidos, la estandarización de variables y codificación de variables categóricas. Una vez preparados los datos se continúa con el proceso en la segunda etapa que consiste en la clasificación, usualmente se dividen los datos en dos partes. Según Witten y Frank (2005) la proporción suele ser de dos tercios (entrenamiento) y de un tercio (prueba), con el 70% de los datos se entrena la red neuronal multicapa y el modelo de regresión logística para clasificar, y el 30% de datos restantes se utiliza en la siguiente etapa para evaluar el modelo construido en la etapa dos. Reche (2013) menciona que la validación cruzada es una versión mejorada del método de retención, por eso también se evaluó la red neuronal multicapa con dicha técnica que mide la eficiencia de predicción del modelo. Con fines comparativos también se realizó la predicción con la Regresión Logística y se obtuvo resultados parecidos a la red neuronal en la etapa de evaluación del modelo.

El aprendizaje de una red neuronal se realiza por medio de reglas de aprendizaje, la del perceptrón de Rosenblatt (1962) y el algoritmo LMS de Widrow y Hoff (1994) fueron

diseñados con el objetivo de entrenar redes que tengan una sola capa. Estos tipos de redes son útiles, su limitación es que sólo pueden resolver problemas susceptibles de ser separados

linealmente. Frente a esta dificultad es necesario la generalización al uso de Redes Neuronales Multicapa. Rumelhart (1986).

De manera general, se explican algunos conceptos básicos para entender las Redes Neuronales; se describe la técnica, sus fórmulas y algoritmos de aprendizaje. Se aplica esta técnica para la clasificación utilizando datos supervisados en la cual la variable respuesta está organizada de manera que cada nivel de las clases definen el número de columnas para transformarla en variable dummy con “p” columnas codificadas con los valores “0” (no pertenencia a la clase) y “1” (pertenencia a la clase). Dichos datos se divide en dos partes una para el entrenamiento y otra para la validación de la red neuronal entrenada. El análisis se realiza con ayuda del programa estadístico R.

El objetivo general de la investigación es, desarrollar la técnica de redes neuronales con el perceptron multicapa y el algoritmo de retro propagación. Los objetivos específicos son: preprocesar datos que incluye la limpieza, el análisis de datos perdidos, detección de outliers, transformaciones, modelamiento y entrenamiento, el perceptrón multicapa con la regla de aprendizaje de retro propagación, y por último post-procesar y evaluar los datos con los parámetros de la red neuronal entrenada.

Los datos utilizados para la investigación fueron proporcionados por el departamento de marketing del banco que tiene presencia en el mercado de todo Perú. Los datos corresponden a las solicitudes de la compra de deuda que tienen sus clientes en otros bancos a nivel nacional.

2.1. JUSTIFICACIÓN

La ventaja de aplicar redes neuronales frente a los métodos paramétricos para caracterizar, diferenciar o clasificar a grupos de individuos surge del hecho de omitir en los datos el cumplimiento de normalidad multivariada, homocedasticidad e incluso se puede continuar el análisis existiendo pocos valores extremos. Kovalevski (2011).

Actualmente el banco que ofrece el servicio de compra de deuda realiza sólo un análisis descriptivo de los datos de cuantas solicitudes son aceptadas y rechazadas. Con esta propuesta se pretende conocer con anticipación si una solicitud de compra de deuda de una institución financiera a otra será aceptada o rechazada, la manera de lograr esta clasificación es utilizando redes neuronales. Con ello se trata de captar a aquellos clientes que se encuentren al borde de su límite de nivel de endeudamiento en el sistema financiero o que consideren como opción consolidar el total de sus deudas en un solo crédito que le permita tener una sola tasa de interés (de preferencia más baja), una sola comisión, un solo seguro de desgravamen, un solo porte y demás gastos financieros, así el cliente podrá tener un mejor manejo de su deuda y compromiso de pago, de esta forma estamos logrando fidelizarlo con un solo producto en la entidad.

2.2. OBJETIVOS

2.2.1. Objetivo General

Desarrollar la técnica de redes neuronales con el perceptron multicapa y el algoritmo de retro propagación.

2.2.2. Objetivos Específicos

- a) Preprocesar datos que incluye la limpieza, el análisis de datos perdidos, detección de outliers, transformaciones.
- b) Modelar y entrenar el perceptrón multicapa con la regla de aprendizaje de retro propagación.
- c) Post-procesar y evaluar los datos con los parámetros de la red neuronal entrenada.

III. REVISIÓN DE LITERATURA

3.1. Descubrimiento de conocimiento en base de datos.

La minería de datos también conocida como el descubrimiento de conocimiento en base de datos o también llamado (KDD), surgió en los años 80, se mejoró en los 90 y actualmente continúa fortaleciéndose. Han *et al.* (2006), mencionan un dicho popular: “estamos viviendo en la era de la información”, porque empresas de todo el mundo generan grandes cantidades de datos medidos en Terabytes o Petabytes. Estas fuentes de datos provienen de bases de datos, textos, imágenes, la web, sensores, etc. Liu (2007). En este mar de datos interviene la minería de datos como una herramienta poderosa de recopilación y descubrimiento de información valiosa, determinando la calidad en términos de “exactitud, integridad, coherencia, oportunidad de credibilidad e interpretabilidad”. Han *et al.* (2006) recalca que algunas personas tratan a la minería de datos como todo el proceso de descubrir conocimiento, mientras que otros utilizan el término refiriéndose solo a un paso importante de todo el proceso, el cual se presenta en la Figura 1.

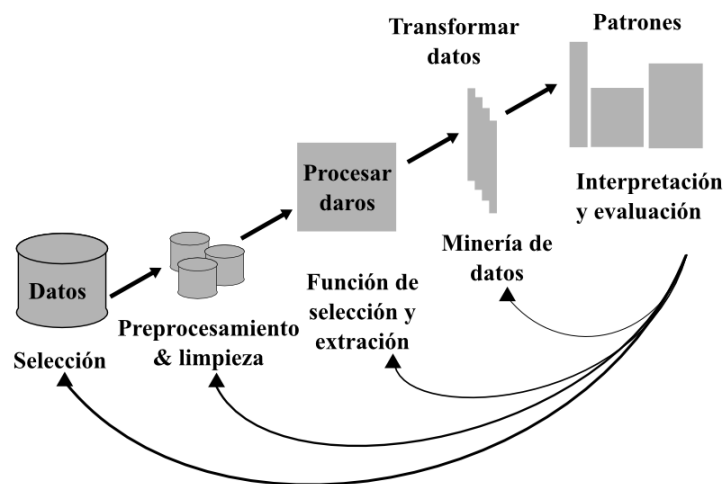


Figura 1: Pasos para realizar el descubrimiento de información en una base de datos.
Fuente: Han et al. (2006).

La Figura 1 presenta las diferentes etapas del KDD. Según Liu (2007) todos estos se resumen en tres pasos; pre procesamiento, minería de datos y post procesamiento. En el primer paso se realiza la limpieza, integración, selección y transformación de datos. En el segundo paso se realiza la minería de datos. Y en el tercer paso se realiza la validación y presentación del conocimiento. Todo el proceso de KDD es iterativo, se hacen varias repeticiones para conseguir un resultado final satisfactorio.

La minería de datos utiliza la estadística y parte de las ciencias de la computación. En la Figura 2 se observan algunos métodos, herramientas y técnicas que influyen para descubrir el conocimiento. Han *et al.* (2006).

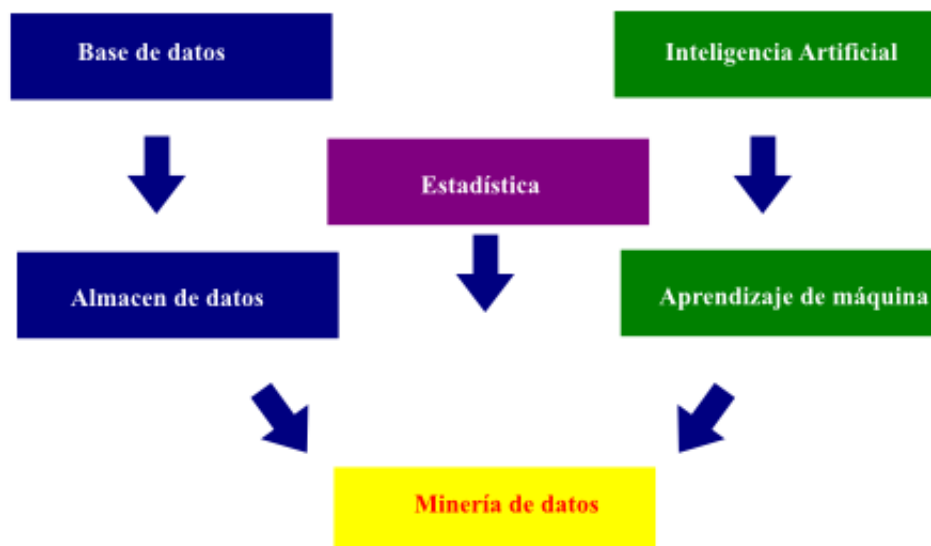


Figura 2: Integración de las técnicas de minería de datos con otras áreas. Fuente: Han et al. (2006).

Aplicar la minería de datos y en particular la clasificación es posible en los actuales momentos porque se tienen los recursos y equipos con mayores capacidad. Se puede almacenar grandes cantidades de datos y realizar cálculos con mayor rapidez, también están disponibles programas tanto libres como comerciales que realizan el proceso de KDD. Además existe mucho interés en el estudio de la relación entre cliente y empresa con la minería de datos, Berry y Linoff (2004).

3.2. Procesos de Minería de Datos CRISP-DM

La implementación de minería de datos en los negocios se denomina “proceso estándar de industria cruzada de minería de datos” o CRISP-DM. En este proceso existen seis fases que son presentadas en la Figura 3: comprensión del negocio (objetivos y requerimientos desde una perspectiva no técnica), comprensión de datos (familiarizarse con los datos teniendo presente los objetivos del negocio), preparación de datos (obtener dataset), modelado (aplicar las técnicas de minería de datos a los dataset), evaluación (de los modelos de la fase anteriores para determinar si son útiles a las necesidades del negocio), implementación (explotar utilidad de los modelos, integrándolos en las tareas de toma de decisiones de la organización). Makhabel (2015).

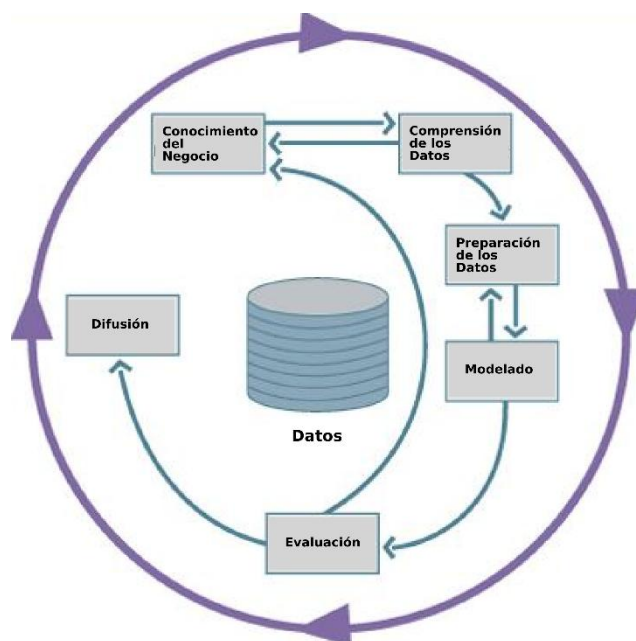


Figura 3: Pasos iterativos de la aplicación del modelo CRISP-DM. Fuente: Makhabel (2015).

3.3. Sistemas de bases de datos y Data Warehouse.

Según Gareth (2013) la minería de datos realiza un buen uso de las tecnologías de base de datos. El almacén de datos se considera como un repositorio para guardar datos a medida que pasa el tiempo. Estos datos pueden provenir de diversas fuentes, pueden ser datos

estructurados, no estructurados, dinámicos, espaciales, temporales, secuencias biológicas, provenientes de sensores, hipertexto, multimedia, códigos, de la web, o de las redes sociales, organizándolos con la finalidad de facilitar la toma de decisiones de los altos directivos como lo menciona Gareth (2013).



Figura 4: Marco de trabajo de un data warehouse.

Fuente: Han et al. (2006).

La Figura 4 presenta el marco de trabajo de un data warehouse el cual facilita la toma de decisiones en una empresa, los datos almacenados en bases de datos se organizan en torno a temas importantes (por ejemplo, el cliente, artículo, proveedor, zona de venta y actividad). Los datos almacenados ofrecen información de forma resumida desde una perspectiva histórica, como por ejemplo en los últimos 6 a 12 meses.

Un almacén de datos por lo general se modela mediante una estructura de datos multidimensional llamado cubo de datos, en el que cada dimensión corresponde a un atributo o un conjunto de atributos en el esquema.

3.4. Pre procesamiento

“El propósito fundamental de la preparación de los datos es la manipulación y transformación de los datos sin refinar, con esto el descubrimiento de conocimiento en el conjunto de datos es más fácil de obtenerlo.” Pyle (1999). Cuando se realiza una predicción con datos sin refinar (incompletos, redundantes y sin claridad), los resultados obtenidos serán imprecisos induciendo a interpretaciones sin coherencias. Para obtener resultados adecuados se debe contar con datos de calidad y esto se logra realizando un pre procesamiento, esto implica: limpieza, integración, transformación y reducción de los datos.

Uno de los principales problemas en el data warehouse es la limpieza de los datos. Con el fin de mejorar la calidad de estos se les aplica un pre procesamiento, en el cual se realizan las siguientes tareas: estimar valores perdidos, identificar outliers, suavizar datos, corregir datos inconsistentes y resolver redundancias causadas por la integración, Rahm y Do (2000). En la investigación de Hernández y Rodríguez (2013) se menciona que la integración tiene la función de crear información de forma homogénea, esto se logra obteniendo datos de diferentes fuentes de datos. Además resuelve problemas de representación integrando los datos desde diferentes fuentes. Según Hernández y Rodríguez (2013) uno de los aspectos importantes en la transformación es la normalización, que consiste en cambiar la escala de valores de los datos a un rango pre-especificado. Esto ayuda a acelerar la fase de aprendizaje en la construcción y ajustes de los modelos. Los valores son transformados o normalizados con la siguiente función $V' = \frac{V-x}{S}$ donde v es la variable a normalizar, x es el promedio de la variable y S su desviación estándar, este tipo de normalización funciona adecuadamente cuando los valores outliers pueden afectar el rango de datos (pero no lo elimina). La reducción de datos busca obtener una representación menor del conjunto de datos que es mucho más pequeña en volumen pero produce los mismos o casi iguales resultados analíticos. Como los datos almacenados en el data warehouse puede resultar en terabytes, las tareas complejas de la minería de datos pueden demorar mucho tiempo ejecutarse sobre el conjunto completo de datos. Dentro de las estrategias se tiene: agregación del cubo de datos, discretización y la reducción de la dimensionalidad. Hernández y Rodríguez (2013).

3.5. Detección de Outliers

Seo (2002) menciona el método de la desviación estandar para detectar valores atípicos, el cual se define como:

- Método 2SD: $x \pm 2\sigma$

Como mínimo el 75% de los valores deben estar dentro de 2σ de la media.

- Método 3SD: $x \pm 3\sigma$

El 89% de los valores deben estar dentro de 3σ de la media. El 94% dentro de 4σ de la media.

Donde x corresponde a la media aritmética de la muestra y σ corresponde a la desviación estándar de la muestra. Los valores por fuera de estos intervalos se consideran valores atípicos.

La inecuación de Chebyshev indica que en cualquier distribución de probabilidad casi todos los valores están cercanos a la media. No más de $1/k^2$ valores de la distribución pueden estar a más de k desviaciones estándar de la media o también, al menos $1 - 1/k^2$ de los valores de la distribución están dentro de k desviaciones estándar de la media. Aunque la inecuación de Chebyshev es útil para cualquier dato de cualquier distribución, ésta limita las observaciones a una pequeña proporción dentro de k desviaciones estándar de la media.

3.6. Aprendizaje de máquina

Gareth (2013) menciona, "Aprendizaje de máquina se refiere a un amplio conjunto de herramientas para la comprensión de los datos".

El aprendizaje de máquina está determinado por el desarrollo en paralelo que han tenido la estadística y la ciencia de la computación, además forma parte de la inteligencia artificial y una de sus aplicaciones es la regresión. Como el procedimiento para resolver estos problemas es complejo este aprendizaje es considerado una caja negra, pero no debe ser el único enfoque se lo puede comparar como un engranaje, donde cada pieza cumple su función y la interacción con las demás aportan al cumplimiento del objetivo, entre estas técnicas se tiene: árboles de clasificación, soporte de máquinas de vectores y redes neuronales.

3.7. Clasificación

Barajas y Morales (2009) mencionan, que el proceso de asignar una observación p variada predictora en uno de varios grupos preestablecidos se denomina clasificación. El objetivo básico es construir un modelo estadístico que tome la información de las p variables para resumirla en un indicador con el cual se puede clasificar la observación de manera correcta en uno de los grupos. En la literatura estadística se pueden encontrar varios métodos de

clasificación como: análisis discriminante, regresión logística, árboles de clasificación, soporte de máquinas de vectores, redes neuronales, entre otros.

La clasificación es la discriminación de nuevos individuos a un grupo o población ya predefinida. Para lograr esta clasificación se hace uso de un conjunto de datos que serán divididos en dos partes: una parte servirá para construir y entrenar un modelo estadístico y el resto de los datos será para evaluar la eficiencia del modelo con los datos sin clasificar. Berry y Linoff (2004).

3.8. Una Neurona

La neurona o también llamada perceptron simple de McCulloch-Pitts(1943), se compone de una suma ponderada de sus neuronas de entradas, seguido por una función no lineal llamada función de activación, originalmente una función de umbral que establece en un rango de valores comprendidos entre. Michie *et al.* (1994):

$$y_k = \begin{cases} 1 & \text{si } \sum_j w_{kj}x_j - u_k \geq 0 \\ 0 & \text{otra manera} \end{cases} \quad (3.1)$$

x_j Es el número de neuronas que se conecta con la j variables predictoras.

w_{kj} Corresponde a los pesos aleatorios en cada conexión en cada k capas.

u_k Es la función de activación en cada una de las neuronas.

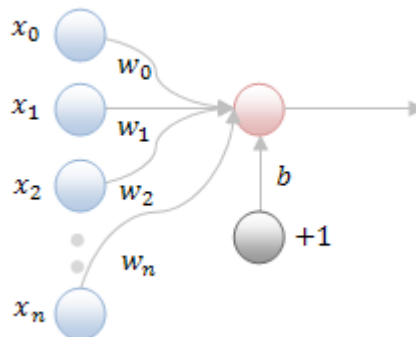


Figura 5: Diagrama perceptron simplificado. Fuente: Dean (2014).

En la Figura 5, se observa la arquitectura de una neurona, la cual consta de un determinado número de entradas y una salida X_i , se asocia cada entrada a un peso W_i ($i=1, \dots, I$). Consta

de un parámetro adicional b de la neurona denominado sesgo o balance que es el peso asociado a la entrada que se establece en $+1$.

$$a = \sum_i w_i x_i \quad (3.2)$$

La ecuación 3.2 es la suma total del producto de las neuronas de entradas por sus respectivos pesos.

A continuación se presentan las funciones de activación más comunes:

a) Lineal.

$$y = a \quad (3.3)$$

La función 3.3 no realiza ninguna transformación del valor neto de la neurona.

b) Sigmoide (función logística).

$$y = \frac{1}{1 + e^{-a}} \quad (y \in [0, 1]) \quad (3.4)$$

La función 3.4 se aplica en la salida de la neurona es decir al valor neto, si este valor es muy elevado lo reduce y si es muy pequeño lo aumenta, estos valores varían entre cero y uno.

c) Sigmoide (tanh).

$$y = \tanh(a) \quad (y \in [-1, +1]) \quad (3.5)$$

La función 3.5 se aplica en la salida de la neurona es decir al valor neto, si este es muy elevado lo reduce y si es muy pequeño lo aumenta, estos valores varían entre menos uno y uno.

La ecuación 3.1 se desglosa en dos partes: el valor neto determinado por la ecuación 3.2 y la función umbral que puede estar definida por la ecuaciones 3.3, 3.4 o 3.5. Mackay (2003).

3.9. Redes Neuronales

El campo de las Redes Neuronales ha surgido a partir de diversas fuentes, que van desde la fascinación de la humanidad con la comprensión y emulación del cerebro humano a cuestiones más amplias como simular las capacidades humanas como el habla y el uso del

lenguaje. También se aplica en la práctica comercial y disciplinas de la ingeniería científica como el reconocimiento de patrones, modelización y predicción. Michie *et al.* (1994).

Según Dean (2014) esta técnica se ha desarrollado en la última mitad del siglo XX y fue inspirada en un modelo matemático de una neurona del núcleo biológico en los animales que permite el aprendizaje.

El objetivo de este método es obtener combinaciones lineales de las entradas con características derivadas para luego modelar el destino como una función no lineal, de esta manera se obtiene un potente método de aprendizaje que se puede aplicar en una variedad de campos. Las redes neuronales abarcan una gran clase de modelos y métodos de aprendizaje, se considera como un modelo de regresión o clasificación de dos etapas que se representa en la Figura 6. Hastie *et al.* (2009).

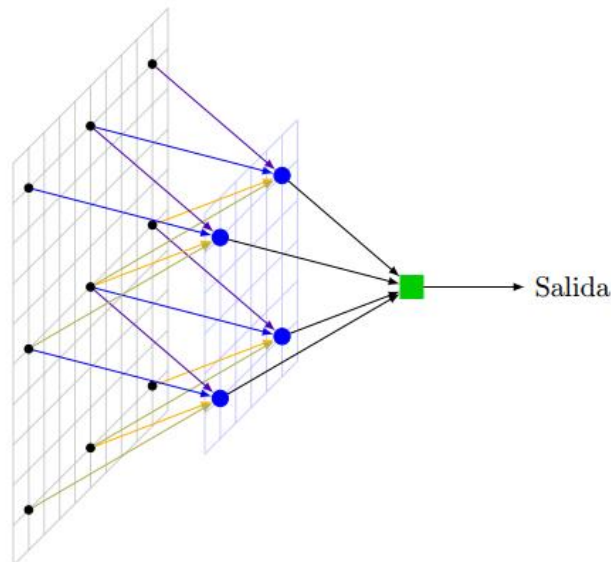


Figura 6: Esquema de una red neuronal con una sola capa oculta. Fuente: Dean (2014).

Según Mackay (2003), cuando se describe un algoritmo de red neuronal normalmente se analizan tres cosas: arquitectura, regla de actividad y regla de aprendizaje.

Este algoritmo de aprendizaje empieza con un grupo aleatorio de pesos y los modifica en base a una función del error que se calcula con la diferencia entre las observaciones y los valores esperados de las variables presentes en las neuronas de entrada. Este tipo de método se establece como aprendizaje supervisado, porque ajusta los pesos de la red inspeccionando las respuestas deseadas que de antemano se conocen, de esta forma se entrena a una red neuronal. La limitante del perceptrón es que puede ser entrenado para

clasificar patrones pertenecientes a clases linealmente separables en un hiperplano mostrado en la Figura 7, en otras palabras es útil cuando se presenta situaciones de clasificación donde la variable respuesta o clase tiene dos niveles. Por esta razón Dean (2014) menciona que las redes neuronales más utilizadas son de tres capas (entradas, oculta, salida).

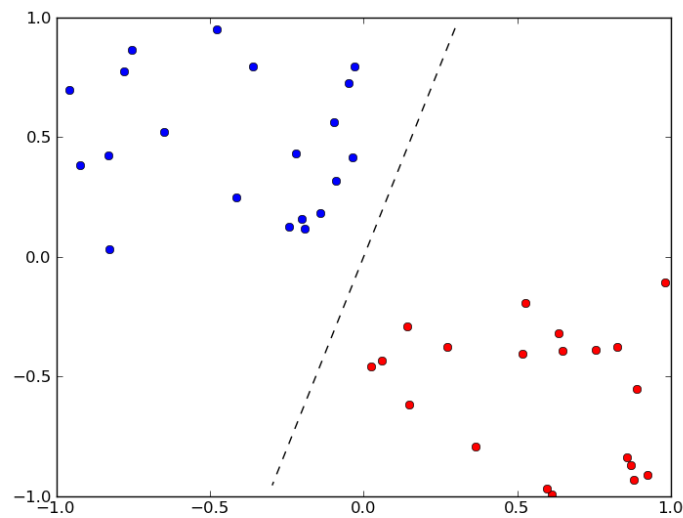


Figura 7: Patrones separados en un hiperplano. Fuente: Dean (2014)

La Figura 7 representa la función de salida para dos clases.

$$f(Z, W, X) = ZhWX \quad (3.6)$$

Ecuación 3.6 se determina una clasificación binomial representada en la Figura 7. X es una matriz de las variables de entrada, que se multiplica con W que el vector de los pesos inicializados de forma aleatoria con valores pequeños, Z es un valor establecido en uno que sirve para crear un equilibrio en el perceptron y h es una constante que ayuda al ajuste de los pesos.

3.10. Perceptrón Multicapas (MLP)

Un método de redes de aprendizaje supervisado es el perceptrón multicapas, donde la activación de la neurona de McCulloch-Pitts se generaliza en la siguiente fórmula.

$$a = f_i(\sum w_i x_i) \quad (3.7)$$

La función 3.7 sirve para obtener el valor neto de cada neurona en la red, el valor neto es la sumatoria del producto entre los pesos W y las variables de entrada X .

La Figura 8 presenta un (MLP) donde la capa de entrada está formada por los nodos de entradas X_i ($i=1, 2, 3, 4$), los nodos Y_i ($i=1, 2$) que conforman la capa de salida, la capa oculta se encuentra en el medio, la cual no es visible ni para las entradas ni las salidas. Además a diferencia de las otras capas esta no mantiene su tamaño fijo, se la puede considerar como un cuello de botella con la finalidad de hacer un modelo sencillo, generalizando los patrones que no se han visto antes, Michie *et al.* (1994).

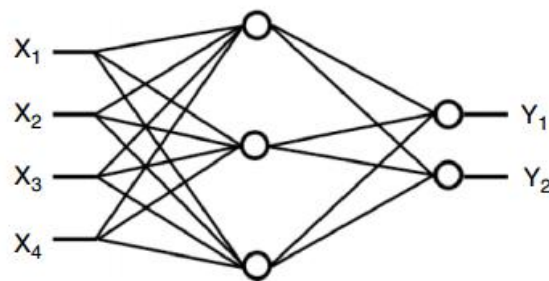


Figura 8: Estructura del perceptrón multicapa. Fuente: Dean (2014).

$$f_1 Z_1, W, X = Z_1 hWX, y f_2 Z_2, W, X = Z_2 hWX \quad (3.8)$$

La función 3.8 se utiliza cuando la clase tiene dos o más niveles, creando una función por cada categoría presentando una arquitectura como en la Figura 8. Los resultados son estimaciones que indican la clase a la que pertenece el individuo.

3.11. Softmax

Softmax es una función de activación o de transferencia que se ubica en la salida de cada neurona en la red para suavizar el valor neto, la suma de todas ellas es igual uno, interpretándose como la probabilidad de pertenencia a la clase correspondiente de un individuo cuando la variable respuesta presenta dos o más categorías, entonces la red neuronal tendrá una neurona en la última capa por cada nivel de la variable respuesta. Dean (2014).

3.12. Método de retención

Reche (2013), menciona un problema recurrente en el ajuste de modelos que es el sobreajuste, esto es cuando el modelo se ajusta muy bien a un conjunto de datos, pero puede no ser útil en el ajuste de otros. Si se disponen de las suficientes observaciones, una alternativa es dividir la muestra en dos submuestras, de forma que una se utilice para ajustar el modelo y la otra para evaluación del mismo, este proceso se llama método de retención o holdout method. La ventaja de este método es que es muy rápido a la hora de calcular. Sin embargo, este método no es muy preciso debido a la variación de los resultados en la validación del modelo.

Conceptualmente sería como lo presenta la Figura 9:

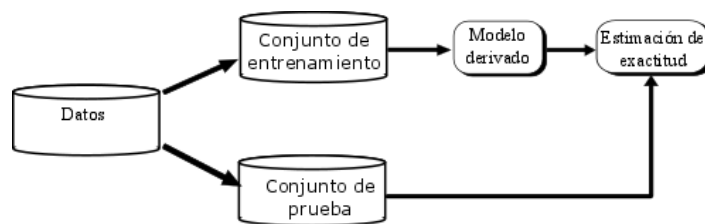


Figura 9: Método de retención o holdout method para validar un modelo. Fuente: Han et al. (2006).

3.13. Validación Cruzada

Consiste en dividir los datos en varios subconjuntos, posteriormente se selecciona uno de ellos como prueba y con los demás lo utiliza para entrenar el modelo de clasificación. El proceso se repite hasta evaluar cada subconjunto de datos. Los resultados de cada iteración se almacenan en una tabla para luego promediarlos y obtener la eficiencia de la predicción. El error que presenta este método tiene menor variabilidad aunque un mayor coste computacional, Reche (2013).

Conceptualmente sería como lo presenta la Figura 10:

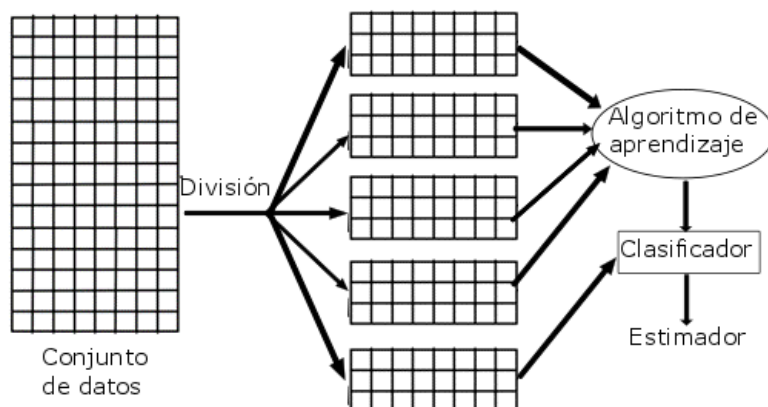


Figura 10: Método de la validación cruzada. Fuente: Bramer (2007).

3.14. Aplicación de redes neuronales en entidades financieras

Pérez y Fernández (2007), mencionan las ventajas y las desventajas del uso de las redes neuronales artificiales y su capacidad para estimar modelos no lineales. Además muestran la aplicación de las redes neuronales a la cuantificación del riesgo de crédito, se cruzan variables de cada cliente, crédito y comportamiento contra la variable default (fallidos y no fallidos). Dentro de las desventajas, es que la comparan con una caja negra, porque resuelve el problema pero es difícil saber cómo lo hizo, por ejemplo en un problema de clasificación, se obtienen los resultados sin saber que cálculos o procedimientos se realizaron.

Gil (2006) resumió algunas aplicaciones empresariales de las redes neuronales como por ejemplo, en banca credit scoring: riesgo en función de un historial, predicción del comportamiento de nuevos clientes, selección de buenas clases de riesgo y riesgo operacional (financiamiento con capitales propios). Las ventajas frente a otras técnicas son: el manejo de gran cantidad de variables y datos, disponen de capacidad de aprendizaje (automática) y elimina la necesidad de adaptación de los sistemas expertos.

Rayo *et al.* (2010), definen un modelo de credit scoring, que aplica redes neuronales en el ámbito financiero, detallando que es una metodología catalogada dentro de las técnicas no

paramétricas, que trata de imitar al sistema nervioso de modo que contribuyen a la construcción de sistemas con cierto grado de inteligencia.

Dueñas *et al.* (2010) con la ayuda de redes neuronales analizaron que el principal riesgo que afronta cualquier institución financiera es que los deudores no estén dispuestos o no tengan las condiciones de pagar sus préstamos. Concluye que el uso de redes neuronales con el algoritmo de aprendizaje backpropagation invierte un tiempo considerable en su entrenamiento, pero sus resultados son más satisfactorios.

IV. MATERIALES Y MÉTODOS

4.1. De oficina y cómputo

Para el desarrollo de este estudio se utilizó un equipo de cómputo con servicio de Internet, además programas ofimáticos (hoja de cálculo, procesadores de texto, presentaciones y notepad), programa estadístico de acceso libre: R versión 3.2.2; programas de licencia libre (netbeans, jdk de java), el gestor de base de datos mysql; programa de inteligencia artificial neuroph; materiales de escritorio y cómputo en general.

4.2. Datos del departamento de marketing de la entidad financiera

Este trabajo de tesis utilizó los datos de un banco, que con más de 100 años en el mercado, que ofrece sus servicios en todo el Perú y en algunos países de América, como: Bolivia, Panamá, USA, y anexos en Colombia y Chile. Uno de los servicios del banco es la compra de deuda en otros bancos de sus clientes. Este servicio es solo a nivel nacional del Perú, además su presencia en el mercado ocupa el 24% de las colocaciones (créditos otorgados a clientes) que realizan los bancos a nivel nacional. El segmento al cual se dirige este producto es a la banca minorista, es decir a clientes naturales, a los cuales se les otorga un crédito personal a sola firma sustentando sus ingresos.

Población: Reporte con un historial de solicitudes desde el mes de enero a junio del 2014.

En el reporte obtenido se tienen datos desde enero 2014 hasta junio 2014, con un promedio que oscila de 450 a 500 solicitudes por mes. En seis meses se tuvo aproximadamente 3000 unidades de análisis.

| Variable | Tipo | Detalle | Códigos |
|---------------|----------|-------------------------------------|---|
| Producto | Factor | Tipo de servicio | Tarjeta =1, crédito= 2 |
| Cita | Factor | Presencia del cliente en la entidad | Con cita=1, sin cita=2 |
| Banco | Factor | 47 instituciones | 1, 2, 3, ..., 47 |
| Observaciones | Factor | Detalle la solicitud | CLIENTE_NO_ASISTIO=1 CUENTA_INCORRECTA =2 MONTO_MAYOR =3 OTROS =4 PAGO_CONFORME =5 PAGO_IMPORTE =6 |
| Importe | Numérica | Valor de la deuda | Variable normalizada |
| Estado_4 | Factor | 4 Clases | Abono, amortizado, cancelado, rechazado |
| Estado_2 | Factor | 2 Clases | Aceptada, Rechazada |

Cuadro 1: Variables utilizadas en la investigación.

El Cuadro 1 presenta las variables utilizadas para el análisis y estas son: Producto (puede ser un crédito, una tarjeta), Cita (si un cliente debe asistir o no en el momento de la solicitud), Banco (47 instituciones financieras donde se envía la solicitud de compra codificadas del 1 al 47), observaciones (motivo por el cual se acepta o rechaza la solicitud), Importe (valor en soles del producto), Estado_4 (variable respuesta de 4 categorías) y la variable respuesta Estado_2 (variable respuesta de 2 categorías).

4.3. Métodos

Para el análisis de datos se aplicó minería de datos, en el pre-procesamiento se realizó la limpieza, el análisis de datos perdidos y transformación de los mismos. Como el objetivo es clasificar se utilizó redes neuronales, en el post-procesamiento se evaluó el desempeño de la red neuronal entrenada. Para la construcción de la red neuronal se tomaron en cuenta los siguientes aspectos: arquitectura, número de neuronas de la capa de entrada, número de neuronas de la capa de salida, número de capas ocultas, número de neuronas en la capa escondida, valor neto del perceptron, función de transferencia, factor de aprendizaje,

producir regla de aprendizaje y algoritmo de aprendizaje propagación hacia atrás (backpropagation).

- **Arquitectura**

Formada por 3 capas: capa de entrada, capa de salida y la capa escondida.

- **Número de neuronas de la capa de entrada**

Depende del número de variables independientes.

- **Número de neuronas de la capa de salida**

Depende del número de categorías o clases que tenga la variable respuesta.

- **Número de capas ocultas**

Según Masters (1993) no existe razón teórica para usar más de dos capas ocultas, la mayoría de los problemas prácticos se resuelven con una capa oculta, si se utiliza un gran número de neuronas en la capa oculta y no se soluciona el problema satisfactoriamente, debe usarse una segunda capa para reducir el número de neuronas en cada capa oculta.

- **Número de neuronas en la capa escondida**

La elección de un número apropiado de neuronas ocultas es muy importante, si se utiliza pocas se tendría pocos recursos para resolver el problema y el uso de demasiadas neuronas aumentaría el tiempo de entrenamiento además de causar un sobre ajuste, por eso para obtener una aproximación del número adecuado de neuronas en la capa oculta se utiliza la regla de la pirámide incremental, Masters (1993).

$$h = \overline{mp} \quad (4.1)$$

donde p por lo general es el número de variables predictoras, m es el número de neuronas de salida y h es el número de neuronas en la capa escondida.

- **Valor neto del perceptrón:**

El valor neto del perceptrón es calculado en la función 4.2, que es la suma de los productos de cada conexión y balance por sus pesos aleatorios, Michie *et al.* (1994).

$$n = \sum_{i=0}^{i=q} (w_i p_i + b) \quad (4.2)$$

w_i Son los pesos asociados a cada conexión en la red neuronal.

p_i Son las neuronas de entradas que corresponde a las variables de entradas.

Y b es el balance en la red neuronal.

- **Producir función de transferencia**

También llamada función umbral o de activación, está presente a partir de las neuronas en la capa de salida hacia adelante. Es la encargada de calcular el nivel del estado de activación generando una señal excitadora (valores muy altos) o inhibidora (valores muy pequeños) a partir de la entrada neta de cada neurona, los resultados en esta función siempre van a ser positivos, comprendidos entre 0 a 1. Mackay (2003).

$$f(n) = \frac{1}{(1+e^{-n})} \quad (4.3)$$

$$a = f(n) = f(\sum_{i=0}^{i=q} (w_i p_i + b)) \quad (4.4)$$

Para simplificar cálculos en lo posterior aplica la primera derivada a la función de activación:

$$f'(n) = f(n)(1 - f(n)) \quad (4.5)$$

- **Producir factor de aprendizaje**

El factor de aprendizaje, es la velocidad de aprendizaje de la red, por medio del ensayo y error siempre debe estar comprendido entre 0.4 y 1, porque si es muy alto el valor hará que

el progreso del entrenamiento sea más rápido y por ende puede no producir una convergencia. Masters (1993).

- **Producir regla de aprendizaje**

(Regla de corrección de errores para aprendizaje supervisado) Minimizar el error cuadrado total. Sea “y” el resultado generado por la red y “d” el resultado esperado. El principio básico de la regla de error- corrección, es la señal de error (d-y). Para lograr este aprendizaje se utiliza el algoritmo de backpropagation. Mateo (2012).

- **Algoritmo de aprendizaje propagación hacia atrás (backpropagation)**

Mateo (2012), menciona que el conocimiento de la red neuronal está en los pesos, y que para conseguir un buen ajuste, se realiza la actualización de estos valores en forma secuencial con los siguientes pasos:

- 1. Inicialización.**

Como no se tiene información a priori, se inicializan los pesos y balances de cada conexión con valores aleatorios entre 0 y 1. Alexandridis (2013).

- 2. Presentación de la muestra de entrenamiento.**

Elegir un patrón de entrada de los datos $X^{(\mu)}$ a ser analizados. Cada variable ingresará a una neurona en la capa de entrada.

- 3. Fase forward.**

Propagar la señal hacia adelante a través de la red, calculando las entradas netas y función de activación de cada perceptrón en la red neuronal.

Calcular δ_i^L el error de aprendizaje en la capa resultante $o_1 = y_i^L$, que es la diferencia de los valores observados y los esperados.

$$\delta_i^L = a'(h_i^L)[d_i^L - y_i^L] \quad (4.6)$$

donde h_i^l representa el input a la red en la i -ésima unidad en la l -ésima capa y a' es la derivada de la función de activación a .

4. Fase backward.

El error de aprendizaje marca el camino más adecuado para la actualización de los pesos que al final del entrenamiento producirá una respuesta satisfactoria. Se calculan los deltas para las capas precedentes propagando los errores hacia atrás.

$$\delta_i^l = a'(h_i^l) \sum_j w_{ij}^{l+1} \delta_j^{l+1} \quad (4.7)$$

Para $l=(L-1)$ capas.

Se realiza la modificación de cada peso utilizando la función 4.8.

$$\Delta w_{ji}^l = n \delta_i^l y_j^{l-1} \quad (4.8)$$

El método de aprendizaje que es el gradiente descendiente es la variación que se produce en el tiempo del valor de los pesos, propagando el error desde la capa final hasta la capa de entrada. Donde n es el factor de aprendizaje, que determina la velocidad con que aprende la red neuronal.

$$w_{ji}^l = w_{ji}^l + \Delta w_{ji}^l \quad (4.9)$$

La ecuación 4.9 calcula el nuevo valor del peso al sumar el incremento Δw al valor del peso actual.

5. Iteración.

Se repiten los pasos 3 y 4 con la muestra de entrenamiento hasta que el algoritmo ajuste los pesos de la red para minimizar el error cuadrático medio del error de aprendizaje hasta un decrecimiento menor a 10^{-6} .

V. RESULTADOS Y DISCUSIÓN

Para la aplicación se utilizó datos proporcionados por el departamento de marketing de una entidad financiera que brinda el servicio de compra de deuda a la banca minorista en todo el Perú y estos datos fueron pre procesadas por el autor. El pre proceso de los datos consistió en integrar en una sola matriz de datos cada uno de los reportes por mes, luego se realizó una codificación de las variables categóricas (producto, cita, observaciones, bancos). También se realizó un análisis descriptivo de cada una de las variables determinando la frecuencia de cada una. Por medio de un análisis gráficos se identificaron algunos datos perdidos como la cantidad de datos perdidos era mínima se procedió a con la eliminación de registros, resultando 3018 casos para el análisis. Además se aplicó una transformación del importe utilizando z-score para minimizar el efecto de los valores atípicos y para que el rango de los valores no sea muy diferente. Se realizó dos análisis uno cuando la variable respuesta es de cuatro clases y otro cuando es de dos clases.

Para por aplicar el método de redes neuronales se realizó la transformación de las variables respuesta a variables dummy según sus niveles, consistió en crear columnas adicionales en la matriz de datos, el número de columnas para la variable Estado_4 fue cuatro, perteneciendo a cada una de las cuatro clases y el número de columnas de la variable Estado_2 fue dos perteneciendo a cada una de las dos clases, esta nuevas columnas se completan con ceros y unos. En número uno en las columnas agregadas indica que esa observación pertenece a dicha clase y las demás columnas se completan con ceros, para esto se programó una función que se encuentra en el anexo 2.

| REGISTRO | VARIABLES |
|----------|-----------|
| 3025 | 7 |

Cuadro 2: Dimensión de los datos.

El Cuadro 2 presenta el total de registros y el número de variables de los datos utilizados, los cuales constan de 3025 filas y 7 columnas.

| BANCOS | CITA | OBSERVACIONES |
|---------------|---------------|------------------------|
| BC_1 :554 | con cita: 21 | CLIENTE_NO_ASISTIO: 87 |
| BC_2 :547 | CON CITA:1438 | CUENTA_INCORRECTA : 84 |
| BC_3 :313 | sin cita: 20 | MONTO_MAYOR : 328 |
| BC_4 :225 | SIN CITA:1546 | OTROS : 99 |
| BC_5 :218 | | PAGO_CONFORME :1428 |
| BC_6 :195 | | PAGO_IMPORTE : 999 |
| ...BC_47 :973 | | |

Cuadro 3: Resumen de los datos de cada variables.

El Cuadro 3 presenta en resumen la frecuencia de cada categoría en las variables banco. Como se puede observar todas las variables son factores, la variable banco tienen 47 instituciones y esta codificada del 1 al 47 esta frecuencia indica que a ciertos bancos se envían más solicitudes que a otras. En la variable cita tiene una proporción parecida en los trámites donde se necesita la presencia del cliente para proceder a la compra, además se observa que existe una inconsistencia de datos donde el programa está reconociendo cuatro categorías donde en realidad son dos esto se debe a las mayúsculas por lo tanto se realizó una limpieza de datos. La variable observaciones presenta 6 niveles donde las de mayor frecuencia son el pago conforme, y las de menor son cuando el cliente no asistió a la cita.

| IMPORTE | PRODUCTO | Estado4 | Estado2 |
|----------------|--------------|-----------------|----------------|
| Min. : 500 | CREDITO:2248 | ABONO A TC: 456 | ACEPTADO :2318 |
| 1st Qu.: 6773 | TARJETA: 777 | AMORTIZADO: 593 | RECHAZADO: 701 |
| Median : 14292 | | CANCELADO :1269 | NA's : 6 |
| Mean : 19647 | | RECHAZADO : 701 | |
| 3rd Qu.: 26482 | | NA's : 6 | |
| Max. :143087 | | | |
| NA's :1 | | | |

Cuadro 4: Resumen de los datos de cada variables.

El Cuadro 4 presenta en resumen la frecuencia de las variables categóricas y el detalle de importe que es numérica. En el importe se tiene que el valor mínimo es 500 y el máximo 143087, donde puede existir una dispersión de estos datos por lo que será conveniente una transformación en esta variable. En producto lo que más se solicita es la compra de créditos. El estado cuatro es la variable respuesta cuanto tiene cuatro niveles y un detalle más amplio al indicar que las solicitudes canceladas tienen una mayor proporción seguida de las rechaza, amortizadas y abonadas. Estado dos es la variable respuesta cuando tienen dos niveles donde detalla que en mayor proporción las solicitudes de compra han sido

aprobadas. Además se observa que existen datos perdidos por lo que se hizo un análisis de la ausencia de estos datos.

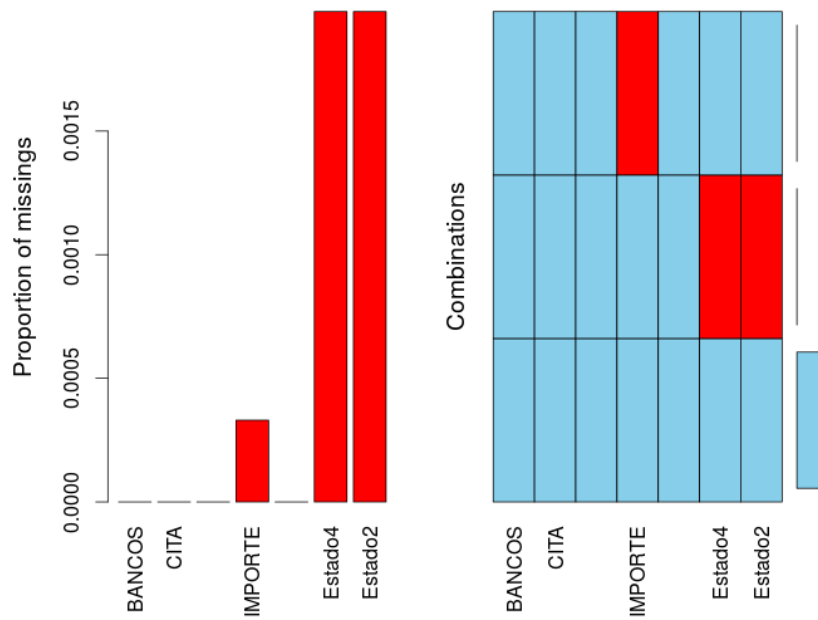


Figura 11: Análisis de los valores perdidos.

En la Figura 11 se aprecia que las variables que presentan un pequeño porcentaje de valores perdidos son el Importe, Estado4, Estado2 en total suman menos del 0.25% de valores perdidos. El resto de las variables consideradas no presentan valores perdidos.

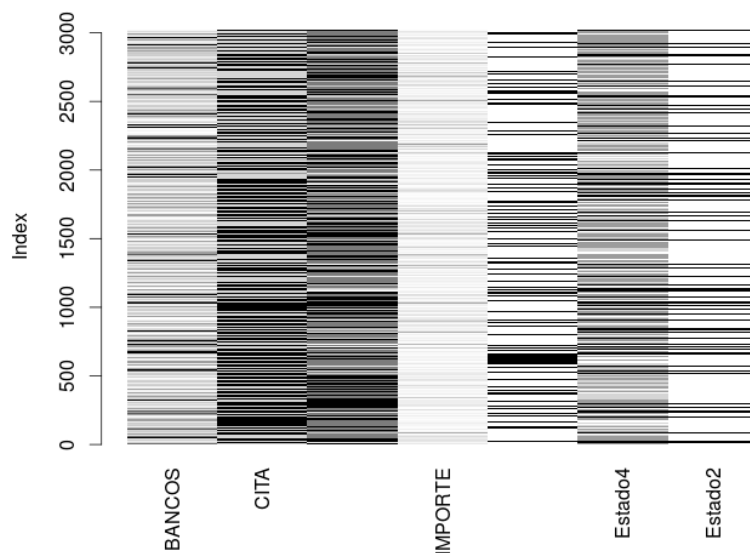


Figura 12: Análisis de los valores perdidos

En la Figura 12 es otra alternativa para presentar y realizar un análisis más general de los valores perdidos también se observa donde se presenta la mayor concentración de los datos según sus índices. Como la presencia de datos perdidos es mínima no se distinguen con claridad su ubicación.

| IMPORTE | Estado4 | Estado2 |
|----------------|----------------|----------------|
| 4 | 6 | 7 |

Cuadro 5: Columna donde existen datos perdidos.

El Cuadro 5 muestra el número de la columna de la matriz de los datos, donde presenta los datos perdidos.

| %Datos perdidos |
|------------------------|
| 0.23% |

Cuadro 6: Porcentaje de valores perdidos en el total de los datos.

En el Cuadro 6 está el porcentaje de datos perdidos, como es un porcentaje pequeño se puede optar por omitir estos datos.

| Variable | Cantidad |
|-----------------|-----------------|
| IMPORTE | 1 |
| Estado4 | 6 |
| Estado2 | 6 |

Cuadro 7: Números de datos perdidos por variables.

El Cuadro 7 presenta la cantidad de datos perdidos organizados por variables.

| REGISTRO | VARIABLES |
|-----------------|------------------|
| 3018 | 7 |

Cuadro 8: Cantidad de datos sin valores perdidos.

Como es un porcentaje pequeño se procedió a omitir estos valores, quedando 3018 registros y 7 variables como indica el Cuadro 8.

| Transformación Max Min (Importe) | | | | | |
|----------------------------------|---------|---------|--------|---------|------|
| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| 0 | 0.04402 | 0.09683 | 0.1343 | 0.1822 | 1 |
| Normalización Z (Importe) | | | | | |
| Min. | 1st Qu. | Median | Mean | 3rd Qu. | Max. |
| -1.1 | -0.7163 | -0.2974 | 0 | 0.3797 | 6.87 |

Cuadro 9: Transformaciones aplicadas a la variable Importe.

Como la variable Importe presenta valores atípicos se procedió a aplicar las transformaciones presentadas en el Cuadro 9. Con esto se quiere minimizar la influencia de estos valores en los modelos, la primera transformación es de máximos - mínimos y la segunda Z-score.

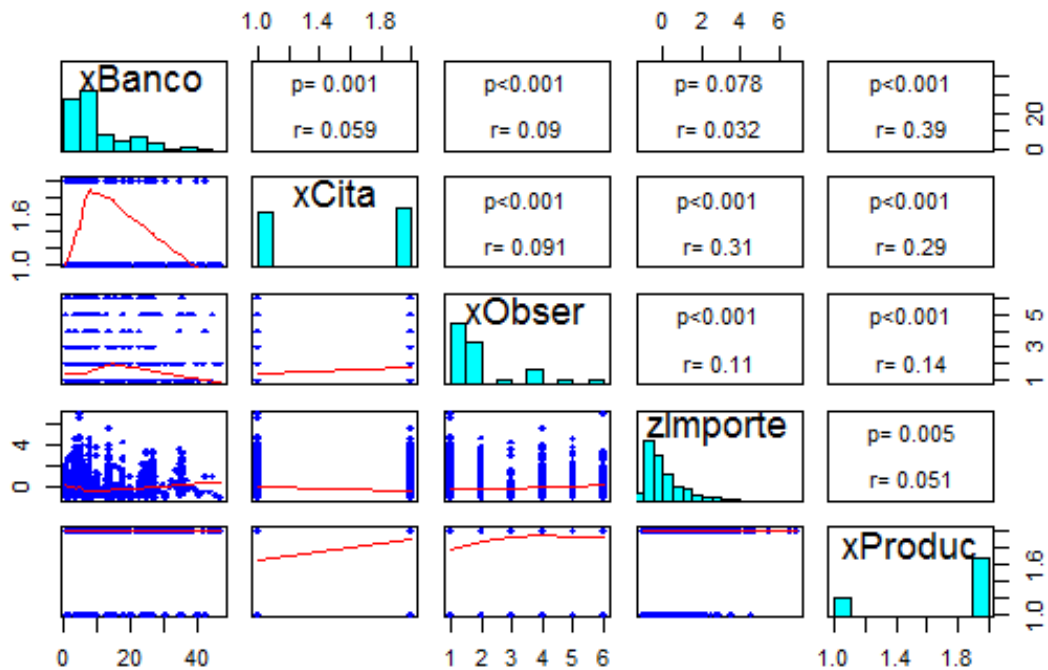


Figura 13: Matriz de correlación.

En la Figura 13 se aprecia la correlación entre las diferentes variables predictoras. Las correlaciones observadas son leves en todos los casos.

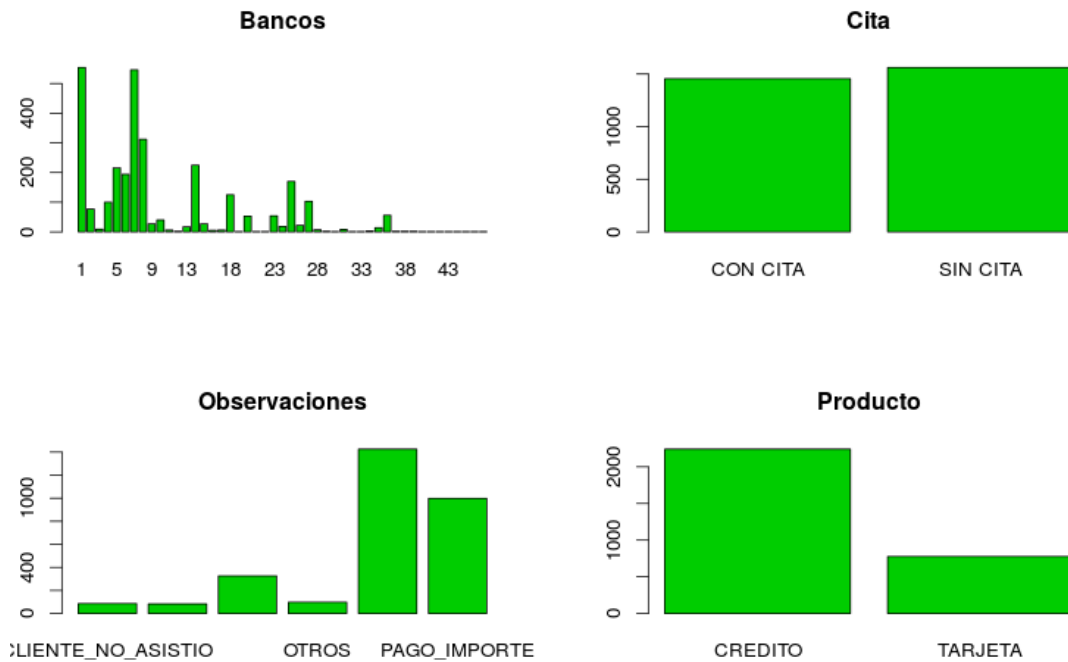


Figura 14: Gráficos exploratorios de las variables consideradas.

En la Figura 14 se aprecia que entre las 47 instituciones financieras codificadas del 1 al 47 a las que se envían solicitudes de compra de deuda hay mayor cantidad en unas que en otras, los trámites se realizan en forma personal casi en la misma magnitud que sin la presencia del cliente. Las mayores observaciones son por pago conforme y pago de importe, las de menor son por número de cuenta incorrecta, el cliente no asiste a la cita y porque el monto a pagar aumentó. Los productos más solicitados son los créditos y los menos solicitados son las tarjetas.

Se puede analizar que los bancos más grandes son los que presentan mayor frecuencia de pagos con mayor éxito y aquellos bancos más pequeños presentan menor frecuencia de pago puesto que son los que requieren la presencia del cliente, pues el cliente debe realizar trámites previos de pre cancelación.

Solicitan más compra de deuda de sus créditos en otras entidades que de las deudas en tarjetas de créditos, pero la deuda de tarjetas se la puede comprar con otro medio de compra que posee la empresa llamado balance transfer que es la compra de deuda de una tarjeta con otra tarjeta de crédito.

Considerando todas las entidades por igual en la compra de deuda, son los bancos chicos los que rechazan casi en su totalidad y algunos bancos grandes también, pero se debe tener en cuenta que en el universo de los 47 bancos son 3 los bancos más grandes que se ven compensados por no presentar muchos rechazos, pero son los que presentan mayor

frecuencia de pagos y en su mayoría solo se rechaza por deuda mayor, pero no por que quieran q el cliente este presente.

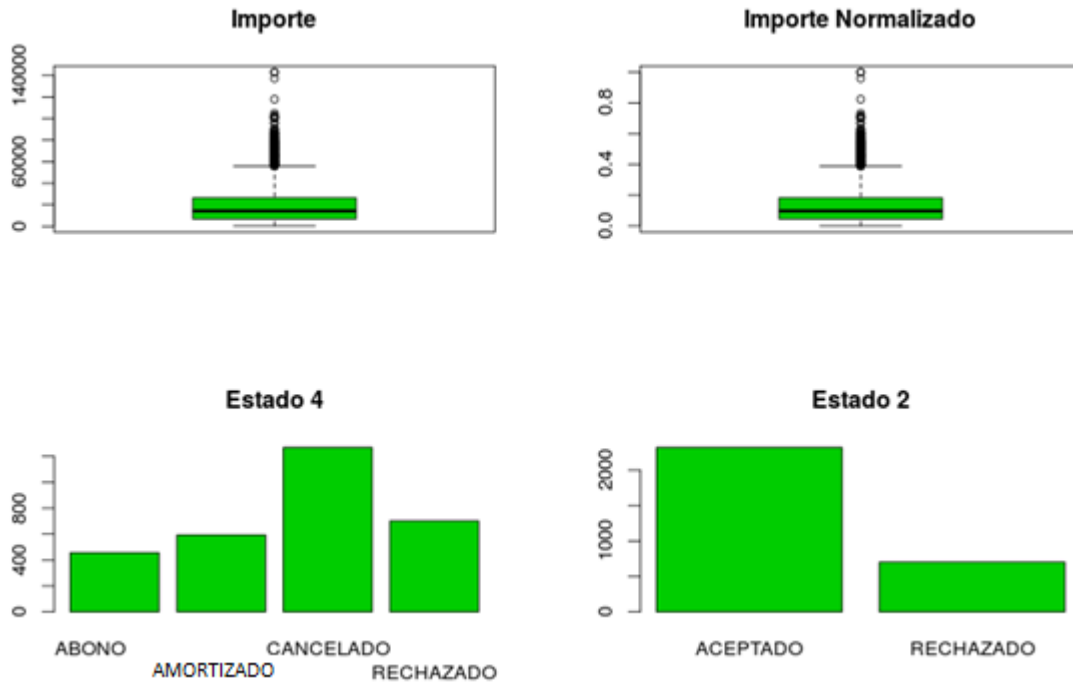


Figura 15: Gráficos exploratorios de las variables consideradas.

En la Figura 15 se aprecia la variable Importe donde se observa la presencia de outliers, para manejar un poco la influencia de estos valores se realizó la transformación por medio de la normalización Z-score, esto ayuda a minimizar la diferencia de estos valores. Además en la clase de cuatro categorías, la mayoría de las solicitudes se han cancelado en su totalidad y en menor porcentaje se han abonado. En la clase de dos categorías, las solicitudes en mayor porcentaje se han aceptado o aprobado.

| Muestra de datos para NNET (4 CLASES) | | | | | | | | |
|---------------------------------------|-------|--------|-------------|---------|----|----|----|----|
| xBanco | xCita | xObser | zImporte | xProduc | U1 | U2 | U3 | U4 |
| 1 | 1 | 1 | 0.05486998 | 1 | 1 | 0 | 0 | 0 |
| 2 | 1 | 1 | -0.01951322 | 2 | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | 1.39010201 | 2 | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | -0.37004668 | 2 | 0 | 1 | 0 | 0 |
| 2 | 1 | 1 | -0.86524474 | 2 | 0 | 1 | 0 | 0 |
| 3 | 1 | 2 | 0.38109645 | 2 | 0 | 1 | 0 | 0 |

Cuadro 10: Tabla de variables independientes y variable dependiente con cuatro clases.

El Cuadro 10 presenta una muestra de la transformación de los datos. Para empezar el análisis de redes neuronales, se transforman las variables asignando códigos en banco, cita, observaciones, productos. En la variable Importe como es cuantitativa se optó por aplicar una normalización z-score para minimizar el efecto de outliers y la variable respuesta de cuatro categorías se convirtió en variable dummy.

| Muestra de datos para NNET (2 CLASES) | | | | | | |
|---------------------------------------|-------|--------|-------------|---------|----|----|
| xBanco | xCita | xObser | zImporte | xProduc | U1 | U2 |
| 1 | 1 | 1 | 0.05486998 | 1 | 1 | 0 |
| 2 | 1 | 1 | -0.01951322 | 2 | 1 | 0 |
| 2 | 1 | 1 | 1.39010201 | 2 | 1 | 0 |
| 2 | 1 | 1 | -0.37004668 | 2 | 1 | 0 |
| 2 | 1 | 1 | -0.86524474 | 2 | 1 | 0 |
| 3 | 1 | 2 | 0.38109645 | 2 | 1 | 0 |

Cuadro 11: Tabla de variables independientes y variable dependiente con dos clases.

El Cuadro 11 presenta una muestra de la transformación de los datos. Para empezar el análisis de redes neuronales, se transforman las variables asignando códigos en banco, cita, observaciones, productos. La variable importe como es cuantitativa se optó por aplicar una normalización z-score para minimizar el efecto de outliers y la variable respuesta de dos categorías se convirtió en variable dummy.

| Muestra de datos para REG-LOG (4 CLASES) | | | | | |
|--|-------|--------|-------------|---------|------------|
| xBanco | xCita | xObser | zImporte | xProduc | Estado4 |
| 1 | 1 | 1 | 0.05486998 | 1 | ABONO A TC |
| 2 | 1 | 1 | -0.01951322 | 2 | CANCELADO |
| 2 | 1 | 1 | 1.39010201 | 2 | CANCELADO |
| 2 | 1 | 1 | -0.37004668 | 2 | CANCELADO |
| 2 | 1 | 1 | -0.86524474 | 2 | CANCELADO |
| 3 | 1 | 2 | 0.38109645 | 2 | CANCELADO |

Cuadro 12: Tabla de variables independientes y variable dependiente con cuatro clases.

El Cuadro 12 presenta una muestra de la transformación de los datos. Para empezar el análisis de regresión logística se transforman las variables asignando códigos en banco, cita, observaciones, productos. El importe como es numérico se optó por aplicar una normalización para minimizar el efecto de outliers y la variable respuesta de cuatro categorías mantuvo su naturaleza.

| Muestra de datos para REG-LOG (2 CLASES) | | | | | |
|---|--------------|---------------|-----------------|----------------|----------------|
| xBanco | xCita | xObser | zImporte | xProduc | Estado2 |
| 1 | 1 | 1 | 0.05486998 | 1 | ACEPTADO |
| 2 | 1 | 1 | -0.01951322 | 2 | ACEPTADO |
| 2 | 1 | 1 | 1.39010201 | 2 | ACEPTADO |
| 2 | 1 | 1 | -0.37004668 | 2 | ACEPTADO |
| 2 | 1 | 1 | -0.86524474 | 2 | ACEPTADO |
| 3 | 1 | 2 | 0.38109645 | 2 | ACEPTADO |

Cuadro 13: Tabla de variables independientes y variable dependiente con dos clases.

El Cuadro 13 presenta una muestra de la transformación de los datos, para empezar el análisis de regresión logística se transforman las variables asignando códigos en banco, cita, observaciones, productos. El importe como es numérico se optó por aplicar una normalización para minimizar el efecto de outliers y la variable respuesta de dos categorías mantuvo su naturaleza.

La red neuronal entrenada usando el 70% de los casos de la base de datos original produjo un error de clasificación de 19.23% cuando la variable respuesta tiene 4 categorías y 4.31% cuando la variable respuesta tiene dos categorías en los datos de prueba (30% de los casos de los datos original). Las especificaciones utilizadas para el entrenamiento de la red neuronal fueron:

- **Tipo de red neuronal.**

Se utilizó una red neuronal multicapa porque la variable respuesta presentó más de dos niveles, estructurado en tres capas: entrada, oculta, y salida.

- **Número de Capas.**

La red neuronal multicapa entrenada para cuatro clases presenta la siguiente estructura: la capa de entrada estuvo formada por un número de neuronas que corresponden al número de variables que en este caso fueron cinco importe, bancos, producto, cita y observaciones. En la capa escondida se establece que el número adecuado de neuronas es también de siete, de acuerdo a la fórmula de pirámide incremental detallada en la metodología, el número de neuronas en la capa de salida está dada por el número de clases en la variable respuesta,

por lo cual quedó fijado en cuatro. La red neuronal multicapa entrenada para dos clases presenta las siguientes características: la capa de entrada estuvo formada por un número de neuronas que corresponden al número de variables que en este caso fueron cinco. En la capa escondida se establece que el número adecuado de neuronas es también de siete, de acuerdo a la fórmula de pirámide; el número de neuronas en la capa de salida está dado por el número de clases en la variable respuesta, por lo cual quedó fijado en dos.

- **Regla de Aprendizaje.**

La regla de aprendizaje que se aplicó fue la de minimizar el error cuadrado total $E = \frac{1}{2} \sum_{k=1}^M \delta^2 p k$, el cual se calcula con la diferencia entre las observaciones y los valores esperados, este resultado es obtenido realizando muchas iteraciones con ayuda del algoritmo de aprendizaje. El criterio de parada de las iteraciones es que el error tenga un valor menor a 10^{-6} .

- **Algoritmo de Aprendizaje.**

El algoritmo de aprendizaje fue el de retropropagación o backpropagation, el cual conjuntamente con el error realiza un elevado número de iteraciones retrocediendo y actualizando los pesos para recalculan el error. En este caso se determinó el valor de parada en 200 iteraciones porque el error de aprendizaje ya no disminuía su valor realizando más iteraciones, indicando que la red estaba entrenada.

- **Factor de aprendizaje.**

El factor de aprendizaje usado fue establecido de forma aleatoria por el programa y es un valor pequeño entre 0.4 a 1 como se explicó en la metodología.

- **Datos.**

Los datos del banco que contiene una variable respuesta con cuatro clases y adicionalmente una variable respuesta con dos clases, fue organizada de manera que se

dicotomizó la pertenencia a cada una de las clases y se dividió los datos en dos partes una para el entrenamiento (70% de los casos) y la otra para la validación del modelo (30% de los casos).

- **Error.**

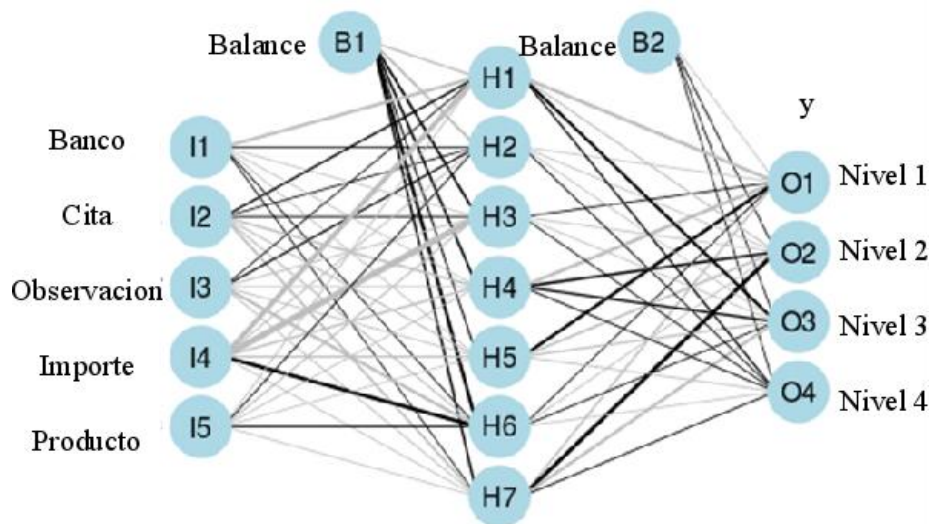
El error de clasificación obtenido al aplicar a la base de datos de prueba la red neuronal multicapa entrenada con las especificaciones ya señaladas fue de 22.89% y 4.31% para red de cuatro y dos salidas respectivamente. Al probar con más de siete neuronas se obtuvo errores de clasificaciones similares o mayores, con un mayor costo computacional.

- **Eficiencia de predicción.**

El error obtenido por el método de retención es muy variable, por ese motivo se optó por evaluar los modelos con el método de validación cruzada, donde se obtuvieron los resultados 76.57% y 77.29% de error de buena clasificación, para la red neuronal multicapa y regresión logística respectivamente.

5.1. Resultados

Figura 16: Estructura de la red neuronal entrenada utilizando un perceptrón multicapa.



La Figura 16 detalla cinco covariables de interés en los datos, las cuales se conectan luego con las siete neuronas que componen la capa escondida. A la derecha se aprecian las cuatro neuronas que componen la capa de salida; cuyo número obedece a las cuatro clases posibles en los datos (correspondientes a los cuatro estados la compra de deuda y que fueron dicotomizadas en cuatro variables de respuesta en la etapa de preparación de los datos).

| Peso de 74 conexiones de NNET 4 CLASES | | | | | |
|--|-----------|-----------|-----------|-----------|-----------|
| [1] | -6.17597 | -15.02852 | 5.4851391 | 0.274102 | -28.82457 |
| [6] | -1.351265 | -11.92623 | 0.0301862 | 0.1350623 | 4.1762946 |
| [11] | -1.901041 | 0.6065296 | 9.1395301 | -0.044413 | 1.2567955 |
| [16] | -2.332141 | -40.15844 | -3.760133 | 8.292965 | -0.032286 |
| [21] | -7.706837 | -0.871019 | -6.86192 | -0.559529 | 2.9112099 |
| [26] | -0.027786 | -2.559204 | -0.834555 | -5.414711 | -0.429358 |
| [31] | 16.176338 | 0.0421703 | -3.503203 | -15.11004 | 18.4796 |
| [36] | 2.4264206 | 7.6422921 | 0.0032628 | -9.512172 | -1.288871 |
| [41] | -0.191627 | -0.018386 | -0.01214 | -16.26504 | -0.005489 |
| [46] | 0.7303474 | -17.28206 | 15.148734 | 0.1155885 | -6.581789 |
| [51] | 0.7352504 | -2.676759 | -0.723865 | -0.517136 | 9.2985119 |
| [56] | -8.507859 | -0.720657 | 20.630595 | 0.2651863 | 13.112968 |
| [61] | -0.260157 | -0.223608 | 6.9728119 | -5.443033 | 0.6154246 |
| [66] | -14.93403 | 0.014476 | 4.3192041 | 0.9856242 | 0.0133368 |
| [71] | 1.1387612 | -1.389424 | -0.002501 | 0.8262761 | |

Cuadro 14: 74 pesos correspondientes a cada conexión de la red neuronal.

El Cuadro 14 contiene los valores de 74 pesos ajustados, que corresponden a las 67 conexiones. El número de conexiones se obtiene con lo siguiente $6 \times 7 + 8 \times 4 = 74$. Seis es la suma del número de neuronas de entradas más el balance, siete es el número de neuronas en la capa escondida, ocho es el número de neuronas en la capa escondida más el balance y cuatro es el número de neuronas de salida.

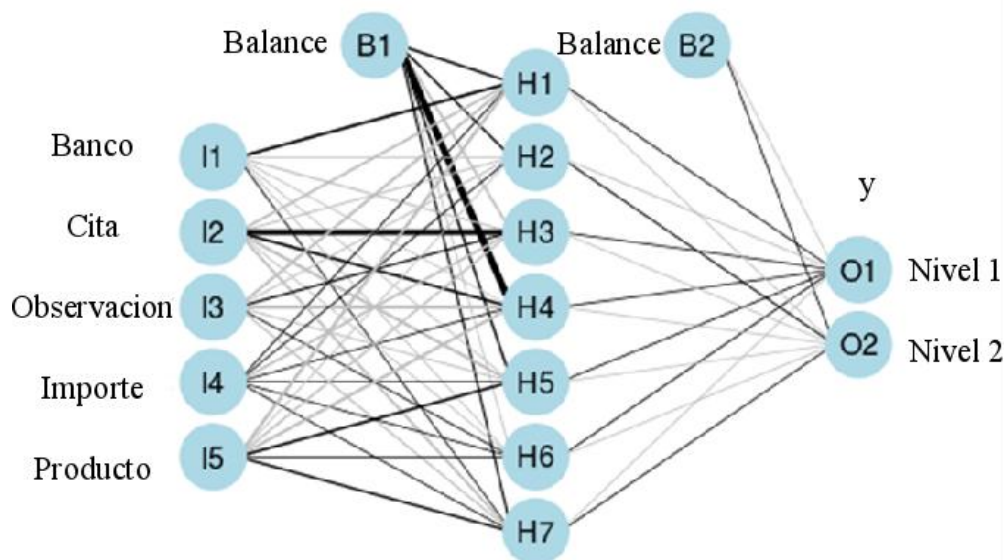


Figura 17: Estructura de la red neuronal entrenada utilizando un perceptron multicapa.

La Figura 17 detalla cinco covariables de interés en los datos, las cuales se conectan luego con las siete neuronas que componen la capa escondida. A la derecha se aprecian las 4 neuronas que componen la capa de salida; cuyo número obedece a las dos clases posibles en los datos [correspondientes a los dos estados (cancelado o rechazado) la compra de deuda y que fueron dicotomizadas en dos variables de respuesta en la etapa de preparación de los datos].

| Pesos de 58 conexiones de NNET 2 CLASES | | | | | | |
|---|-----------|-----------|-----------|-----------|-----------|-------------|
| [1] | 21.810689 | 35.421534 | -31.52306 | -37.07453 | 0.4632949 | -46.6508431 |
| [7] | 17.170267 | -0.052626 | -9.041923 | -8.804695 | 0.1894927 | -2.08827644 |
| [13] | -28.36105 | -12.21718 | 102.13449 | 13.270462 | -29.54525 | -20.6606785 |
| [19] | 134.55401 | -0.043475 | 30.553752 | -55.6042 | 0.6302303 | -40.8449342 |
| [25] | 20.541888 | -0.049161 | -41.74452 | -7.596351 | 0.1846254 | 28.2341493 |
| [31] | -1.485228 | -0.394727 | -0.530641 | 0.0408867 | 0.18208 | 0.20366143 |
| [37] | 13.937341 | 0.0230601 | -31.12563 | -13.82409 | 0.4148403 | 20.9181058 |
| [43] | -0.166721 | 0.1640252 | -5.311244 | 0.0596754 | 0.9403807 | 0.9638138 |
| [49] | 1.2221765 | -0.897811 | 1.1661182 | -0.163428 | 5.3025574 | -0.05974483 |
| [55] | -0.940365 | -0.963736 | -1.211192 | 0.8975989 | | |

Cuadro 15: 58 pesos correspondientes a cada conexión de la red neuronal.

El Cuadro 15 contiene los valores de 58 pesos ajustados, que corresponden a las 58 conexiones. El número de conexiones se obtiene con lo siguiente $6 \times 7 + 8 \times 2 = 58$. Seis es la suma del número de neuronas de entradas más el balance, siete es el número de neuronas en la capa escondida, ocho es el número de neuronas en la capa escondida más el balance y dos es el número de neuronas de salida.

| Muestra de predicciones de NNET 4 CLASES | | | | |
|--|-----------|-----------|-----------|-----------|
| | X1 | X2 | X3 | X4 |
| 2761 | 0.679909 | -0.001651 | 0.323016 | 0.0039275 |
| 2828 | 0.4992736 | 0.0538622 | 0.4497669 | 0.0027843 |
| 863 | -0.019044 | 0.7323172 | 0.2896301 | -0.004312 |
| 2504 | 0.0566789 | 0.6804923 | 0.2589888 | 0.0052166 |
| 1935 | -0.010007 | 0.7071269 | 0.3093846 | -0.00827 |
| 1565 | -0.064105 | 0.8014477 | 0.2553252 | 0.0005444 |

Cuadro 16: Tabla de predicciones con la variable de cuatro clases.

El Cuadro 16 muestra los valores de las predicciones de una red neuronal para cuatro clases, indicando la clase que pertenece cada individuo analizado.

| Matriz de Confusión | | | | |
|----------------------------------|----------|----------|----------|----------|
| cd | 1 | 2 | 3 | 4 |
| 1 | 83 | 1 | 17 | 0 |
| 2 | 8 | 239 | 70 | 24 |
| 3 | 13 | 3 | 27 | 0 |
| 4 | 0 | 1 | 1 | 116 |
| Error método de retención | | | | |
| 22.89% | | | | |

Cuadro 17: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados.

El Cuadro 17 presenta los elementos que están bien y mal clasificados en cada clase, analizados con la red neuronal entrenada para cuatro clases. Utilizando el método de retención para validar el modelo, se obtuvo un error de clasificación de 22.89%.

| Muestra de predicciones de NNET 2 CLASES | | |
|---|-----------|--------------|
| | X | X.1 |
| 2761 | 1.0054189 | -0.005406071 |
| 2828 | 1.0056943 | -0.005679061 |
| 863 | 1.0021592 | -0.002409381 |
| 2504 | 0.9857717 | 0.014160489 |
| 1935 | 1.0014592 | -0.001390662 |
| 1565 | 0.9909624 | 0.008972877 |

Cuadro 18: Tabla de predicciones con la variable de dos clases.

El Cuadro 18 muestra los valores de las predicciones de una red neuronal para dos clases, indicando la clase que pertenece cada individuo analizado.

| Matriz de Confusión | | |
|----------------------------------|----------|----------|
| cd | 1 | 2 |
| 1 | 462 | 25 |
| 2 | 1 | 115 |
| Error método de retención | | |
| 4.31% | | |

Cuadro 19: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados.

El Cuadro 19 presenta los elementos que están bien y mal clasificados en cada clase, analizados con la red neuronal entrenada para dos clases. Utilizando el método de retención para validar el modelo, se obtuvo un error de clasificación de 4.31%.

| Importancia de las Variables | |
|-------------------------------------|------------------|
| Overall | Variables |
| 110.84356 | xObser4 |
| 101.91012 | xObser3 |
| 76.75369 | xObser5 |
| 55.90841 | zImport |
| 32.43394 | xObser6 |
| 22.48857 | xCita4 |

Cuadro 20: Tabla de importancia de las variables.

El cuadro 20 presenta la importancia de las variables, en las primeras posiciones se tiene tres niveles de la variable observación y en el cuarto lugar la variable importe normalizada.

| Matriz de Confusión | | | | |
|----------------------------------|-------------------|-------------------|------------------|------------------|
| preds2 | ABONO A TC | AMORTIZADO | CANCELADO | RECHAZADO |
| ABONO A TC | 85 | 18 | 1 | 0 |
| AMORTIZADO | 11 | 26 | 3 | 0 |
| CANCELADO | 8 | 70 | 239 | 24 |
| RECHAZADO | 0 | 1 | 1 | 116 |
| Error método de retención | | | | |
| 22.72% | | | | |

Cuadro 21: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados.

El Cuadro 21 presenta los elementos que están bien y mal clasificados en cada clase, analizados con la regresión logística construida para cuatro clases. Utilizando el método de retención para validar el modelo, se obtuvo un error de clasificación de 22.72%.

| Matriz de Confusión | | |
|----------------------------------|-----------------|------------------|
| preds2 | ACEPTADO | RECHAZADO |
| ACEPTADO | 461 | 24 |
| RECHAZADO | 2 | 116 |
| Error método de retención | | |
| 4.31% | | |

Cuadro 22: Tabla de la matriz de confusión, representa los individuos bien y mal clasificados.

El Cuadro 22 presenta los elementos que están bien y mal clasificados en cada clase, analizados con la regresión logística construida para dos clases. Utilizando el método de retención para validar el modelo, se obtuvo un error de clasificación de 4.31%.

5.2. Resultados consolidados

| Resultados Consolidados | |
|--------------------------------|--------|
| 4 CLASES | |
| nnet4 | 22.89% |
| reglog4 | 22.72% |
| 2CLASES | |
| nnet2 | 4.31% |
| reglog2 | 4.31% |

Cuadro 23: Tabla de resultados consolidados.

El Cuadro 23 presenta un consolidado de los resultados del error de clasificación para cada técnica utilizada en la clasificación, obtenidos con el método de retención. Se puede observar que la diferencia de los porcentajes en ambas técnicas es mínima.

5.3. Validación Cruzada.

| Muestra de datos para NNET (4 CLASES) | | | | | | | | | |
|---------------------------------------|-------|--------|-----------|---------|----|----|----|----|-------|
| xBanco | xCita | xObser | zImporte | xProduc | U1 | U2 | U3 | U4 | kfold |
| 1 | 1 | 1 | 0.05487 | 1 | 1 | 0 | 0 | 0 | 10 |
| 2 | 1 | 1 | -0.019513 | 2 | 0 | 1 | 0 | 0 | 3 |
| 2 | 1 | 1 | 1.390102 | 2 | 0 | 1 | 0 | 0 | 1 |
| 2 | 1 | 1 | -0.370047 | 2 | 0 | 1 | 0 | 0 | 1 |
| 2 | 1 | 1 | -0.865245 | 2 | 0 | 1 | 0 | 0 | 1 |
| 3 | 1 | 2 | 0.3810965 | 2 | 0 | 1 | 0 | 0 | 6 |

Cuadro 24: Datos preparados para aplicar la validación cruzada en la red neuronal multicapa.

El Cuadro 24 presenta los datos preparados para aplicar el método de validación cruzada, para evaluar los modelos se puede observar que a los datos se le añade una variable kfold, que representa a los k grupos que son asignados datos de forma aleatoria, para luego construir modelos y evaluarlos de forma repetitiva.

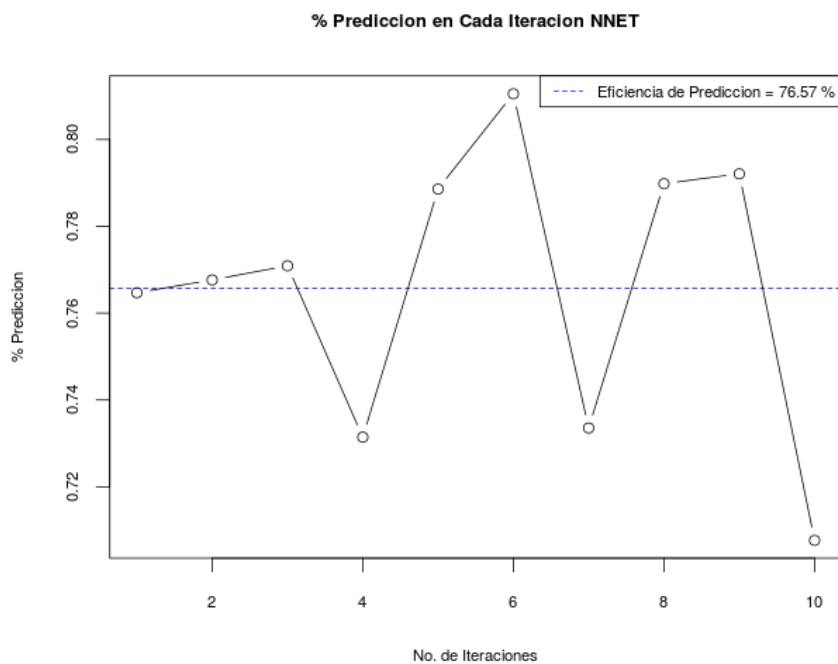


Figura 18: Eficiencia de predicción del método de Redes Neuronales Multicapa.

En la Figura 18 se observa la variabilidad del porcentaje de predicción al realizar el método por validación cruzada utilizando 10 particiones. Los 10 puntos graficados en la

figura corresponden a cada una de las estimaciones, al dejar una de las 10 particiones fuera de la conjunto de datos de entrenamiento y utilizarla como conjunto de datos de prueba. La línea azul corresponde a la media de esos 10 porcentajes de eficiencia de predicción y su valor se especifica en la leyenda.

| Muestra de datos para REG-LOG (4 CLASES) | | | | | | |
|--|-------|--------|-------------|---------|------------|-------|
| xBanco | xCita | xObser | zImporte | xProduc | Estado4 | Kfold |
| 1 | 1 | 1 | 0.05486998 | 1 | ABONO A TC | 6 |
| 2 | 1 | 1 | -0.01951322 | 2 | CANCELADO | 3 |
| 2 | 1 | 1 | 1.39010201 | 2 | CANCELADO | 8 |
| 2 | 1 | 1 | -0.37004668 | 2 | CANCELADO | 2 |
| 2 | 1 | 1 | -0.86524474 | 2 | CANCELADO | 4 |
| 3 | 1 | 2 | 0.38109645 | 2 | CANCELADO | 2 |

Cuadro 25: Datos preparados para aplicar la validación cruzada en la regresión logística multinomial.

El Cuadro 25 presenta los datos preparados para aplicar el método de validación cruzada, para evaluar los modelos se puede observar que a los datos se le añade una variable kfold, que representa a los k grupos que son asignados datos de forma aleatoria, para luego construir modelos y evaluarlos de forma repetitiva.

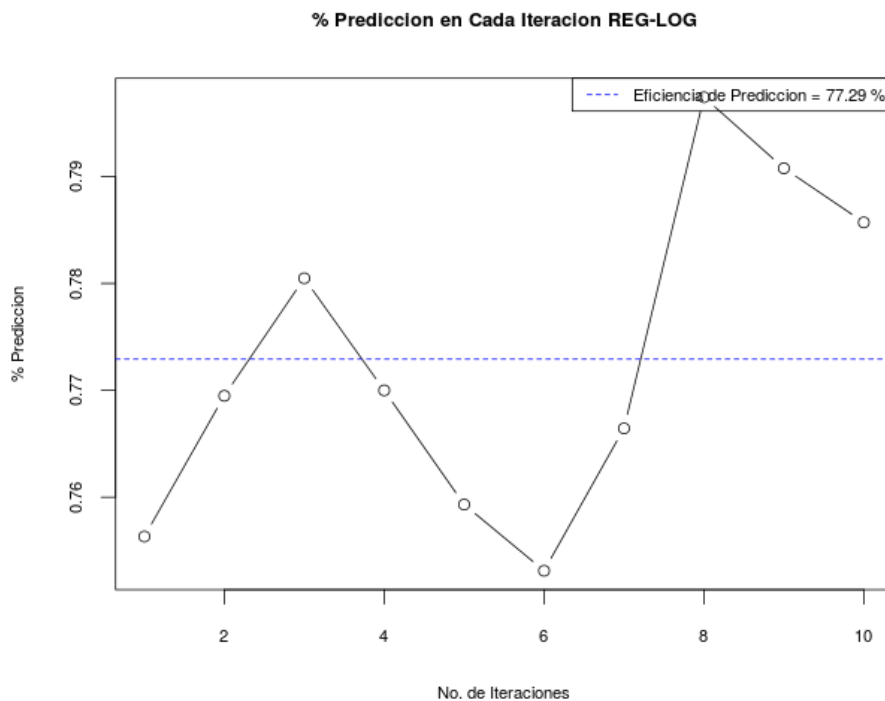


Figura 19: Eficiencia de predicción del método regresión logística.

En la Figura 19 se observa la variabilidad del porcentaje de predicción al realizar el método por validación cruzada utilizando 10 particiones. Los 10 puntos graficados en la figura corresponden a cada una de las estimaciones al dejar una de las 10 particiones fuera de la base de datos de entrenamiento y utilizarla como base de datos de prueba. La línea azul corresponde a la media de esos 10 porcentajes de eficiencia de predicción y su valor se especifica en la leyenda.

| Resultados consolidados 4 clases | |
|---|---------------------------------|
| Técnica | Eficiencia de predicción |
| NNET | 76.57% |
| REG-LOG | 77.29% |

Cuadro 26: Porcentaje de eficiencia de predicción consolidado.

El Cuadro 26 presenta un consolidado de los resultados de la eficiencia de predicción, para cada técnica utilizada en la clasificación, obtenidos con el método de validación cruzada. Se puede observar que la diferencia de los porcentajes en ambas técnicas es mínima. Este resultado es más confiable porque es menos variable, en comparación con el método de retención.

VI. CONCLUSIONES

- Se desarrolló la técnica de redes neuronales con el perceptrón multicapa y el algoritmo de retropropagación aplicado a la predicción de la compra de una deuda de una institución financiera a otra. Encontrándose que esta red neuronal multicapa aporta un error de correcta clasificación del 76.57% además se aplicó la regresión logística aportando un error de correcta clasificación del 77.29%, siendo este último un 0.72% mejor que el anterior método. En este proceso se aplicó minería de datos con sus tres etapas: pre procesamiento, análisis y pos procesamiento.
- Un aspecto importante dentro de la minería de datos es el preprocesamiento, con esto se tuvo mejor calidad e integridad de los mismos. En primera instancia se organizó los datos desagregados en una sola matriz, luego se analizó la presencia de datos perdidos que como eran muy pocos se optó por eliminarlos. En las variables categóricas se realizó una codificación para poder aplicar el análisis de redes neuronales con el programa y en la variable numérica se aplicó una transformación por normalización para minimizar el efecto de valores atípicos. Además se realizó un Análisis Exploratorio de Datos para tener una idea general de todas las variables.
- El uso de un perceptrón multicapa permitió analizar los datos al superar la limitación que impuso el uso de un perceptrón simple, pues utilizando más de una capa es posible contemplar cuatro clases posibles. Esta posibilidad de cuatro clases se expresó en que la red neuronal formada constó de cinco neuronas de entrada (las covariables), siete neuronas en la capa escondida (según la regla de pirámide incremental) y cuatro neuronas en la capa de salida, además funciona con clases linealmente separables. Esta topología conjuntamente con el algoritmo de retropropagación permitió el aprendizaje con sus diferentes etapas.
- Los resultados encontrados al momento de aplicar el método de redes neuronales en los datos de la entidad financiera para la clasificación el error con los datos de prueba fue de 22.89% cuando la variable respuesta tiene cuatro categorías y 43.11%

cuando la variable respuesta tiene dos categorías. El menor error de clasificación correspondió a considerar una capa escondida formada por el número aproximado de neuronas recomendado por la fórmula de pirámide incremental. Cuando se utilizó una capa oculta con un número diferente de neuronas se obtuvieron errores de clasificaciones mayores o un error similar pero con más costo computacional. Como el método de retención es muy variable se aplicó la validación cruzada, obteniendo una eficiencia de predicción del 76.57%.

VII. RECOMENDACIONES

- La aplicación que se realizó con las redes neuronales ha sido de clasificación, pero sería de mucho interés la aplicación de esta técnica en otras áreas como: Diseño de máquinas industriales, reconocimiento de firmas, reconocimiento de señales de sonar, entretenimiento y otras más.
- Aplicar un método de imputación en los valores perdidos como alternativa a la eliminación que se utilizó en el pre procesamiento, además aplicar un tipo distinto de transformación en la variable numérica con el que se puede manejar de mejor manera la presencia de los valores outliers y hacer un análisis exploratorio de datos más exhaustivo para poder tener una mejor calidad de los datos.
- Para hacer otras comparaciones con estos resultados haciendo mejoras continuas de los mismos se puede probar con diferentes arquitecturas que definan la topología de la red. Además se puede utilizar otro algoritmo de aprendizaje y establecer un nuevo criterio de parada. También se puede aumentar el número de datos para ajustar mejor el modelo. Asimismo optar por otra función de activación para cada neurona en las diferentes capas.
- Como alternativa a los métodos de validación utilizados en la investigación como el error por retención y validación cruzada. Se pueden aplicar métodos de evaluación de modelos con menos variabilidad en los resultados y poder pensar en métodos que realicen este proceso en conjunto con la técnica bootstrap.

VIII. REFERENCIAS BIBLIOGRÁFICAS

Alexandridis, A. K., & Zaprani, A. D. 2013. Wavelet neural networks: A practical guide. *Neural Networks*, 42, 1-27.

Barajas, F. H.; Morales, J. C. C. 2009. Comparación entre tres técnicas de clasificación. *Revista Colombiana de Estadística* 32(2):247-265.

Berry, M. J.; Linoff, G. S. 2004. *Data mining techniques: for marketing, sales, and customer relationship management*. Indianapolis, Indiana, John Wiley & Sons.

Bramer, M. 2007. *Principles of data mining* (Vol. 131). London: Springer.

Dean, J. 2014. *Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners*. Hoboken, New Jersey, John Wiley & Sons.

Dueñas, A. M. H.; Huaraz, V. C.; Sánchez, J. C. H. 2010. Modelo de Evaluación de Créditos Financieros basados en Redes Neuronales orientado a Edpymes. *Revista de investigación de Sistemas e Informática* 7(2):21-33.

Gareth, J. 2013. *An Introduction to Statistical Learning: with Applications in R*. Springer New York Heidelberg Dordrecht London, Springer.

Gil, F. R. 2006. La redes neuronales y la gerencia de riesgos. Gerencia de riesgos y seguros 23(94):43-54.

Han, J.; Kamber, M.; Pei, J. 2006. Data mining, southeast asia edition: Concepts and techniques. 225 Wyman Street, Waltham, MA 02451, USA, Morgan kaufmann.

Hastie, T.; Tibshirani, R.; Friedman, J.; Hastie, T.; Friedman, J.; Tibshirani, R. 2009. The elements of statistical learning. Stanford, California, Springer. v. 1, (2)

Hernández, C.; Rodríguez, J. E. R. 2013. Preprocesamiento de datos estructurados. Vínculos 4(2):27-48.

Kovalevski, L. 2011. Métodos de clasificación no paramétrica. Decimosextas Jornadas" Investigaciones en la Facultad" de Ciencias Económicas y Estadística, Universidad Nacional de Rosario, Rosario, Argentina.

Liu, B. 2007. Web data mining: exploring hyperlinks, contents, and usage data. Springer Heidelberg Dordrecht London New York, Springer Science & Business Media.

MacKay, D. J. 2003. Information theory, inference, and learning algorithms. Cambridge University Citeseer. (7)

Makhabel, B. 2015. Learning Data Mining with R, Develop key skills and techniques with R to create and customize data mining algorithms. Birmingham Mumbai, Packt Publishing.

Masters, T. 1993. Practical neural network recipes in C++. San Francisco, California, Morgan Kaufmann.

Mateo, F. 2012. Redes neuronales y preprocesado de variables para modelos y sensores en bioingeniería.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.

Michie, D.; Spiegelhalter, D. J.; Taylor, C. C. 1994. *Machine learning, neural and statistical classification*. New York, N.Y.: Ellis Horwood, Ellis Horwood Series in Artificial Intelligence.

Pérez, R. F. O.; Fernández, C. H. 2007. Las redes neuronales y la evaluación del riesgo de crédito. *Revista Ingenierías Universidad de Medellín* 6(10):77-91.

Pyle, D. 1999. *Data preparation for data mining*. Morgan Kaufmann. (1)

Rahm, E.; Do, H. H. 2000. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.* 23(4):3-13.

Rayo, C. S.; Lara, R. J.; Camino, B. D. 2010. Un Modelo de Credit Scoring para instituciones de microfinanzas en el marco de Basilea II. *Journal of Economics, Finance and Administrative Science* 15(28):89-124.

Reche, J. L. C. 2013. *Regresión logística. Tratamiento computacional con R*.

Rosenblatt, F. 1962. *Principles of neurodynamics*.

Rumelhart, D. 1986. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Nature* 323:533-536.

Seo, S. (2002). A review and comparison of methods for detecting outliers in univariate data sets (Doctoral dissertation, Faculty of Graduate School of Public Health in partial fulfillment of the requirements for the degree of Master of Science by Songwon Seo BS, Kyunghee University).

Widrow, B.; Hoff, M. E. 1994. Adaptive switching circuits (in IRE WESCON Convention Record 1960). SPIE MILESTONE SERIES MS 96:105-105.

Witten, I. H.; Frank, E. 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

IX. ANEXOS

9.1. ANEXO 1

Librerías en R.

```
###LIBRERIAS  
  
library(VIM)  
  
library(psych)  
  
#  
  
library(nnet)  
  
library(caret)  
  
library(Metrics)  
  
setwd("~/Escritorio/KDD TESIS/")  
  
source('nnet_plot_update.r')
```

9.2. ANEXO 2

Comandos de R para el preprocesamiento

```
#*****  
*****  
  
#                   *PREPROCESAMIENTO*  
  
#*****  
*****
```

```

# *LIMPIEZA DE DATOS

datos<-read.delim('clipboard',T)

#banco<-read.table("datosTesis.csv",header=T)

head(datos)

summary(datos)

dim(datos)

names(datos)

which(colSums(is.na(datos))!=0)

rmiss=which(rowSums(is.na(datos))!=0,arr.ind=T)

length(rmiss)*100/dim(datos)[1]

b<-aggr(datos)

b

matrixplot(datos)

# Eliminar casos

datos.cl=na.omit(datos) #DATOS LIMPIOS

dim(datos.cl)

dim(datos)

dt4<-datos.cl[-7]

names(dt4)

dt2<-datos.cl[-6]

names(dt2)

# *TRANSFORMACIÓN

```

```

NDataMt01<-function(datos,class){

  #FUNCIÓN PARA TRANSFORMAR UNA VARIABLE EN DUMMY

  #class es la columna donde esta la clase

  dt1<-datos[-class]

  nfila<-dim(dt1)[1]

  ncol<-dim(dt1)[2]

  #ndatos<-matrix(0, nfila, ncol)

  ndatos<-dt1

  #matriz respuesta en ceros y unos

  dclass<-unique(datos[class])

  nclass<-dim(dclass)[1]

  ncol1<-dim(dclass)[1]

  nfilas1<-dim(datos)[1]

  mtz01<-matrix(0, nfilas1, ncol1)

  for(i in 1:ncol1){

    for(j in 1:nfilas1){

      if(datos[j,class]==dclass[i,1]){

        mtz01[j,i]<-1

      }

    }

  }

  return(mtz01)

}

```

```
dt4Dummy<-NDataMt01(dt4,6)
```

```
View(cbind(dt4[6],dt4Dummy))
```

```
dt2Dummy<-NDataMt01(dt2,6)
```

```
View(cbind(dt4[6],dt2Dummy))
```

```
norNMAx<-function(datos){
```

```
  dt<-as.numeric(datos)
```

```
  nfila<-length(dt)
```

```
  nfila
```

```
  Xmin<-min(dt)
```

```
  Xmax<-max(dt)
```

```
  ndatos<-c()
```

```
  for(j in 1:nfila){
```

```
    ndatos[j]<--(dt[j]-Xmin)/(Xmax-Xmin)
```

```
  }
```

```
  return(ndatos)
```

```
}
```

```
ImporNor<-norNMAx(datos.cl[,4])
```

```
summary(ImporNor)
```

```
class(ImporNor)
```

```
#Z-score
```

```
library(reshape)
```



```

zImporte<-rescaler(x=datos.cl[4],type="sd")

#zImporte<-cbind(zImporte,datos.cl[4] )

summary(zImporte)

# * ANALISIS EXPLORATORIO DE DATOS

summary(datos.cl) #RESUMEN DE DATOS

describeBy(datos.cl[,-7],datos.cl[,7])

pairs.panels(datos.cl[,-
6],bg=c("red","yellow","blue","green")[datos.cl[,6]],pch=21,main="DEA
TESIS",hist.col="red") #to show changing the diagonal

pairs.panels(datos.cl,pch=".")

names(datos.cl)

summary(datos.cl[6])

par(mfrow=c(2,4))

plot(datos.cl[1],main="Bancos",col=3)

plot(datos.cl[2],main="Cita",col=3)

plot(datos.cl[3],main="Observaciones",col=3)

plot(datos.cl[5],main="Producto",col=3)

boxplot(datos.cl[4],main="Importe",col=3)

boxplot(zImporte,main="Importe Normalizado",col=3)

plot(datos.cl[6],main="Estado 4",col=3)

plot(datos.cl[7],main="Estado 2",col=3)

par(mfrow=c(1,1))

```

```
# *CODIFICACIÓN
```

```
codificar<-function(var){
```

```
  x<-var
```

```
  tipo<-unique(x)
```

```
  ntipo<-dim(tipo)[1]
```

```
  n<-dim(x)[1]
```

```
  x1<-rep(0,n)
```

```
  for(i in 1:n){
```

```
    for(j in 1:ntipo){
```

```
      if(x[i,1]==tipo[j,1]){
```

```
        x1[i]<-j
```

```
      }
```

```
    }
```

```
  }
```

```
  return(x1)
```

```
}
```

```
names(datos.cl)
```

```
xBanco<-codificar(datos.cl[1])
```

```
xCita<-codificar(datos.cl[2])
```

```
xObser<-codificar(datos.cl[3])
```

```
#zImporte
```

```

xProduc<-codificar(datos.cl[5])

# *INTEGRACIÓN

dnnet4<-cbind(xBanco,xCita,xObser,zImporte,xProduc,dt4Dummy)

head(dnnet4)

dim(dnnet4)

dnnet2<-cbind(xBanco,xCita,xObser,zImport,xProduc,dt2Dummy)

head(dnnet2)

dim(dnnet2)

dreglo4<-cbind(xBanco,xCita,xObser,zImport,xProduc,dt4[6])

head(dreglo4)

dreglo2<-cbind(xBanco,xCita,xObser,zImport,xProduc,dt2[6])

head(dreglo2)

#CREAR DATA

#write.csv(dnnet4, file = "DNNET4.csv",row.names = FALSE)

#write.csv(dnnet2, file = "DNNET2.csv",row.names = FALSE)

#write.csv(dreglo4, file = "DREGLO4.csv",row.names = FALSE)

#write.csv(dreglo2, file = "DREGLO2.csv",row.names = FALSE)

#write.csv(dnnet4BCMZX, file = "norDNNET4.csv",row.names = FALSE)

#write.csv(drelog4BC, file = "norDREGLO4.csv",row.names = FALSE)

```

9.3. ANEXO 3

Comandos de R para la Clasificación.

```
#####  
*****  
  
#           *CLASIFICACIÓN  
  
#####  
*****  
  
# * RED NEURONAL MULTICAPA  
  
  
  
#dnnet4BC<-read.table("DNNET4.csv",header=T,sep=",")  
dnnet4BC<-read.table("norDNNET4.csv",header=T,sep=",")  
#dnnet4BCMZX<-dnnet4BC  
set.seed(42)  
smp11<-sample(x=nrow(dnnet4BC),size=0.2*nrow(dnnet4BC))  
#dnnet4BCMZX$IMPORTE<-ImporNor  
summary(dnnet4BC)  
  
  
test<-dnnet4BC[smp11,]  
train<-dnnet4BC[-smp11,]  
head(dnnet4BC)  
dim(dnnet4BC)  
str(train)
```

```

#Iteraciones

#mod1 variables independientes ~ variables respuestas

mod1<-nnet(train[,1:5],train[,6:9],data=dat.in,size=7,linout=T,maxit = 200)

#help(nnet)

#Red Neuronal

plot.nnet(mod1)

pred<-predict(mod1,newdata = test)

head(pred)

cd<-c()

for(j in 1:dim(pred)[1]){

  maxv<-max(pred[j,])

  for(i in 1:dim(pred)[2])

    if(pred[j,i]==maxv){

      cd[j]<-i

    }

}

#cd

tcd<-c()

for(j in 1:dim(test)[1]){

  for(i in 6:dim(test)[2])

    #empieza del 6 porque hay esta la variable respuesta

    if(test[j,i]==1){

      tcd[j]<-(i-5)

      #se le resta 5 porque necesitamos los valores de la clase

```

```

    }
}
#tcd

error=mean(cd!=tcd)

print('ERROR POR MÉTODO DE RETENCIÓN')

error

#Calcular la matriz de confusión

print('MATRIZ DE CONFUSIÓN')

table(cd,tcd)

# * NNET CASOS DE EXITOS

dnnet2BC<-read.table("DNNET2.csv",header=T,sep=",")

set.seed(42)

smp1<-sample(x=nrow(dnnet2BC),size=0.2*nrow(dnnet2BC))

test<-dnnet2BC[smp1,]

train<-dnnet2BC[-smp1,]

head(dnnet2BC)

#Iteraciones

#mod1 variables independientes ~ variables respuestas

```

```

#set.seed(42)

mod1<-nnet(train[,1:5],train[,6:7],data=dat.in,size=7,linout=T,maxit = 600)

#help(nnet)

#Red Neuronal

plot.nnet(mod1)

pred<-predict(mod1,newdata = test)

head(pred)

cd<-c()

for(j in 1:dim(pred)[1]){

  maxv<-max(pred[j,])

  for(i in 1:dim(pred)[2])

    if(pred[j,i]==maxv){

      cd[j]<-i

    }

}

#cd

tcd<-c()

for(j in 1:dim(test)[1]){

  for(i in 6:dim(test)[2])

    #empieza del 6 porque hay esta la variable respuesta

    if(test[j,i]==1){

      tcd[j]<-(i-5)

      #se le resta 5 porque necesitamos los valores de la clase

    }

}

```

```

}

#tcd

error1=mean(cd!=tcd)

print('ERROR POR EL MÉTODO DE RETENCIÓN')

error1

#Calcular la matriz de confusi?n

print('MATRIZ DE CONFUSIÓN')

table(cd,tcd)

# * REGRESIÓN IOGISTICA MULTINOMIAL

#drelog4BC<-read.table("DREGLO4.csv",header=T,sep=",")
drelog4BC<-read.table("norDREGLO4.csv",header=T,sep=",")

head(drelog4BC)

str(drelog4BC)

drelog4BC$xProduc<-as.factor(drelog4BC$xProduc)

drelog4BC$xCita<-as.factor(drelog4BC$xCita)

drelog4BC$xObser<-as.factor(drelog4BC$xObser)

drelog4BC$Estado4<-as.factor(drelog4BC$Estado4)

drelog4BC$zImport<-ImporNor

summary(drelog4BC)

```



```

#unique(as.numeric(banco$ESTADO))

set.seed(42)

str(drelog4BC)

smp11<-sample(x=nrow(drelog4BC),size=0.2*nrow(drelog4BC))

test1<-drelog4BC[smp11,]

entrenamiento1<-drelog4BC[-smp11,]

dim(test1)#Porción de base de datos 20%

dim(entrenamiento1)#Porción de base de datos el 80%

head(entrenamiento1)

mod1<-multinom(entrenamiento1$Estado4~.,data=entrenamiento1,max=200,trace=T)

#Importancia de las variables

topModels<-varImp(mod1)

topModels$Variables<-row.names(topModels)

topModels<-topModels[order(-topModels$Overall),]

head(topModels)

#(Observaciones VS Predichas)

preds1<-predict(mod1,type='probs',newdata=test1)

preds2<-predict(mod1,type='class',newdata=test1)

error=mean(preds2!=test1$Estado4)

print('ERROR POR METODO DE RETENCIÓN')

error

```

```
#Calcular la matriz de confusión
```

```
print('MATRIZ DE CONFUSIÓN')
```

```
table(preds2,test1$Estado4)
```

```
## REGRESIÓN LOGÍSTICA CASOS DE EXITOS
```

```
drelog2BC<-read.table("DREGLO2.csv",header=T,sep=",")
```

```
head(drelog2BC)
```

```
drelog2BC$xProduc<-as.factor(drelog2BC$xProduc)
```

```
drelog2BC$xCita<-as.factor(drelog2BC$xCita)
```

```
drelog2BC$xObser<-as.factor(drelog2BC$xObser)
```

```
drelog2BC$Estado2<-as.factor(drelog2BC$Estado2)
```

```
#unique(as.numeric(banco$ESTADO))
```

```
str(drelog2BC)
```

```
set.seed(42)
```

```
smpl1<-sample(x=nrow(drelog2BC),size=0.2*nrow(drelog2BC))
```

```
test1<-drelog2BC[smpl1,]
```

```
entrenamiento1<-drelog2BC[-smpl1,]
```

```
dim(test1)#Porción de base de datos 20%
```

```
dim(entrenamiento1)#Porción de base de datos el 80%
```

```
head(entrenamiento1)
```

```
mod1<-multinom(entrenamiento1$Estado2~.,data=entrenamiento1,max=200,trace=T)
```

```

#Importancia de las variables

topModels<-varImp(mod1)

topModels$Variables<-row.names(topModels)

topModels<-topModels[order(-topModels$Overall),]

head(topModels)

#(Observaciones VS Predichas)

preds1<-predict(mod1,type='probs',newdata=test1)

preds2<-predict(mod1,type='class',newdata=test1)

error=mean(preds2!=test1$Estado2)

print('ERROR')

error

#Calcular la matriz de confusión

print('MATRIZ DE CONFUSIÓN')

table(preds2,test1$Estado2)

```

9.4. ANEXO 4

Comandos de R para el post procesamiento.

```

#*****
*****

#           *POS PROCESAMIENTO

```

```

#####
*****

# * VALIDACIÓN CRUZADA

##RED NEURONAL MULTICAPA

valCruzNNET<-function(datos){

Folds <-10

datos<-datos

datos$kfold <-sample(1:Folds,nrow(datos),replace=T)

head(datos)

Iteraciones <-data.frame(iteracion=NULL,aciertos=NULL)

for (k in 1:Folds)

{

test <-subset(datos,kfold==k) # Crea set TEST para i iteracion

train <-subset(datos,!kfold==k) # Crea set Entren. para i iteracion

Modelo <-nnet(train[,1:5],train[,6:9],data=datos,size=7,linout=T,maxit = 200)

names(test)

pred<-predict(Modelo,newdata = test)

cd<-c()

for(j in 1:dim(pred)[1]){

maxv<-max(pred[j,])

for(i in 1:dim(pred)[2])

```

```

if(pred[j,i]==maxv){
  cd[j]<-i
}
}
#cd
tcd<-c()
for(j in 1:dim(test)[1]){
  #for(i in 5:dim(test)[2])
  for(i in 6:9)
    #empieza del 6 porque hay esta la variable respuesta
    #termina en 9 porque la 10 es la variable kflod
    if(test[j,i]==1){
      tcd[j]<-(i-5)
      #se le resta 4 porque necesitamos los valores de la clase
    }
}
#tcd
Aciertos<-mean(cd==tcd)
Iteraciones <-rbind(Iteraciones,
  data.frame(iteracion=k,
    acierto=Aciertos)) # resultado en cada iteracion
}

# -----
# PASO Graficar resultado de las iteraciones

```

```

plot(Iteraciones,type="b",main="% Prediccion en Cada Iteracion NNET",
     cex.axis=.7,cex.lab=.7,cex.main=.8, xlab="No. de Iteraciones", ylab="% Prediccion")

abline(h=mean(Iteraciones$acierto),col="blue",lty = 2)

promedio <-format(mean(Iteraciones$acierto, na.rm=TRUE)*100,digits = 4)
legend("topright", legend = paste("Eficiencia de Prediccion =",promedio,"%"),
      col = "blue", lty = 2, lwd = 1,cex=.7,bg=NULL)

promedioNNET<-promedio
return(promedioNNET)
}

summary(dnnet4BC)
summary(dnnet4BCMZX)

efnnet<-valCruzNNET(dnnet4BC)
efnnet

# *REGRESIÓN LOGÍSTICA
valCruzRegLo<-function(datos){
head(datos)

Folds    <-10

datos$fold <-sample(1:Folds,nrow(datos),replace=T)

#####

# Regresion Logistica      #

```

```
#####

Iteraciones <-data.frame(iteracion=NULL,aciertos=NULL)

for (i in 1:Folds)

{

  Test      <-subset(datos,kfold==i)  # Crea set TEST para i iteracion

  Entrenamiento <-subset(datos,!kfold==i) # Crea set Entren. para i iteracion

  names(Test)

  Modelo<-multinom(Entrenamiento$Estado4~.,data=Entrenamiento[-
7],max=200,trace=T)

  Prediccion<-predict(Modelo,type='class',newdata=Test[,c(-6,-7)])

  names(Test)

  Aciertos=mean(Prediccion==Test$Estado4)

#Modelo <-lda(Sobrevivio~-PH,data=Entrenamiento)

#Prediccion <- predict(Modelo,Test[,-1])$class

#MC      <-table(Test["Sobrevivio"],Prediccion) # Matriz de Confusión

#Aciertos <-MC[1,1]/(MC[1,1]+MC[2,1])

Iteraciones <-rbind(Iteraciones,

                    data.frame(iteracion=i,

                                acierto=Aciertos)) # resultado en cada iteracion

}

# -----

# PASO 4: Graficar resultado de las iteraciones
```

```

plot(Iteraciones,type="b",main="% Prediccion en Cada Iteracion REG-LOG",
     cex.axis=.7,cex.lab=.7,cex.main=.8, xlab="No. de Iteraciones", ylab="% Prediccion")

abline(h=mean(Iteraciones$acierto),col="blue",lty = 2)

promedio <-format(mean(Iteraciones$acierto, na.rm=TRUE)*100,digits = 4)
legend("topright", legend = paste("Eficiencia de Prediccion =",promedio,"%"),
      col = "blue", lty = 2, lwd = 1,cex=.7,bg=NULL)

promedioRLOG<-promedio
return(promedioRLOG)
}

efReglo<-valCruzRegLo(drelog4BC)

efReglo

```

9.5. ANEXOS 5

Aplicaciones de Redes Neuronales en otras áreas.

Las Redes Neuronales son multipropósitos y su aplicación en otros campos es muy amplia y se la encuentra en:

- Diseño de máquinas industriales.
- Reconocimiento de firmas.
- Reconocimiento de señales de sonar.
- Predicción de indicadores económicos.

Concesión de préstamos.

Las Redes Neuronales son actualmente el sistema utilizado para el aprendizaje de un software por sí mismo.

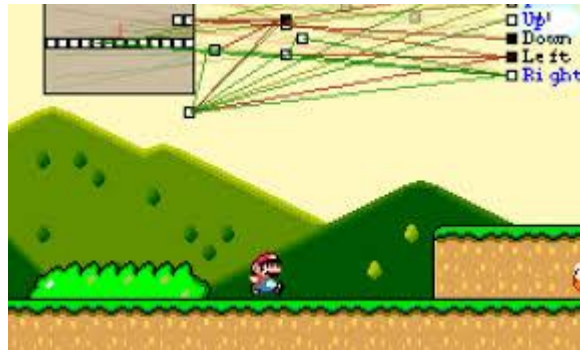


Figure 20: Aplicación de red neuronal utilizando el algoritmo NEAT (Neuro Evolución de Topologías Aumentativas).

La Figura 20 muestra el set de reglas que permiten a la pc jugar por sí misma el primer nivel del juego “Super Mario World”. La red neuronal construye las reglas necesarias que permitan maximizar el puntaje obtenido, que es una función del porcentaje de la superficie del nivel que se logra recorrer y la velocidad con la que se efectúa dicho recorrido. La aplicación corresponde al programa “MarI/O” y se refiere a una red neuronal entrenada por 24 horas.