Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia Artificial
Recibido: 02/11/2020 | Aceptado: 22/01/2021 | Publicado: 30/03/2021

Clasificador de estrellas de Neutrones con una red neuronal multicapa utilizando R

Neutron star classifier with a multilayer neural network using R

Luis Angel Aliaga Marica [1*], Edilson Wanser Herrera Villa [2], José Mejía Huayhua [3], Lizette Quispe Flores [4]

[1] Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú. laliagam@unsa.edu.pe
[2] Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú. eherrerav@unsa.edu.pe
[3] Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú. jmejia@unsa.edu.pe
[4] Universidad Nacional de San Agustín de Arequipa, Arequipa, Perú. lquispeflore@unsa.edu.pe

[*] Autor para correspondencia: laliagam@unsa.edu.pe

Resumen
En este trabajo lo que se realizará es analizar el ejercicio "Clasificador de estrellas de Neutrones" para ello lo primero se expondrá una breve introducción de nuestro ejercicio planteado seguidamente realizaremos los conceptos básicos de un red neuronal ya que es el escogido para la resolución del presente ejercicio, pero este está clasificado por redes neuronales artificiales según la topología red y redes según el método de aprendizaje, donde se ha visto por conveniente realizarlo con la red neuronal multicapa – perceptrón multicapa, después se tendrá la limpieza de datos, transformación de casos, selección de casos, selección de un lenguaje de datos así mismo los paquetes, librerías framework que se utilizará, para seguidamente realizar la ejecución de la técnica de entrenamiento, modelo entrenado fase de comprobación , análisis de los resultados y análisis del cliente; finalmente llegar a las conclusiones.

Palabras clave: estrellas, multicapa, neuronas, perceptrón, red.

Abstract
In this work what will be done is to analyze the exercise "Neutron star classifier" for this, the first thing will be presented a brief introduction of our proposed exercise, then we will carry out the basic concepts of a neural network since it is the one chosen for the resolution of this exercise, but this is classified by artificial neural networks according to the network topology and networks according to the learning method, where it has been seen to be convenient to do it with the multilayer neural network - multilayer perceptron, then you will have the data cleaning, transformation of cases, selection of cases, selection of a data language as well as the packages, framework libraries that will be used, to then carry out the execution of the training technique, trained model, verification phase, analysis of the results and analysis of the client; finally come to conclusions.

Keywords: *multilayer, neurons, network, perceptron, stars.*

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

Introducción

Un púlsar del acrónimo en inglés de *pulsating star* (estrella que emite radiación muy intensa a intervalos cortos y regulares) es una estrella de neutrones que emite radiación periódica. Los púlsares poseen un intenso campo magnético que induce la emisión de estos pulsos de radiación electromagnética a intervalos regulares relacionados con el periodo de rotación del objeto. Las estrellas de neutrones pueden girar sobre sí mismas hasta varios cientos de veces por segundo; un punto de su superficie puede estar moviéndose a velocidades de hasta 70 000 km/s. Se pretende realizar un software que incluirá en el próximo telescopio que lance al espacio con el objetivo de identificar a las estrellas de neutrones de este tipo. Para esto cuenta con una base de datos histórica de 17 898 observaciones de estrellas en las que se clasifican en pulsares o no. En dicha base de datos se tiene como atributos de cada observación:

- Promedio de la radiación emitida.
- Desviación estándar de la radiación emitida.
- Índice de expansión de su masa.
- Índice del aumento de la radiación emitida.
- Promedio de la velocidad de rotación.
- Desviación estándar de la velocidad de rotación.
- Índice de ruido que posee la emisión de radio frecuencia.
- Fuerza centrífuga generada durante la rotación.

Materiales y métodos o Metodología computacional

Redes Neuronales Artificiales

Como lo menciona [1], las redes neuronales artificiales han recibido un interés particular como una tecnología para minería de datos, puesto que ofrece los medios para modelar de manera efectiva y eficiente problemas grandes y complejos. Son modelos computacionales inspirados en las neuronas biológicas, y que están conformadas por un conjunto de unidades de cómputo básico (neuronas) las cuales están conectadas entre ellas de múltiples maneras. Estas conexiones estarán definidas por unos pesos los cuales determinarán la fuerza o importancia de dichas conexiones, y durante el proceso de aprendizaje o entrenamiento de la red, serán estos pesos los que se ajustarán con el fin de producir la salida adecuada según la entrada que se aplique a la red.

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

Estructura de los modelos de redes neuronales: Las unidades de procesamiento se organizan en capas. Hay tres partes normalmente en una red neuronal: una capa de entrada, con unidades que representan los campos de entrada; una o varias capas ocultas; y una capa de salida, con una unidad o unidades que representa el campo o los campos de destino. Las unidades se conectan con fuerzas de conexión variables (o ponderaciones). Los datos de entrada se presentan en la primera capa, y los valores se propagan desde cada neurona hasta cada neurona de la capa siguiente. al final, se envía un resultado desde la capa de salida [2].

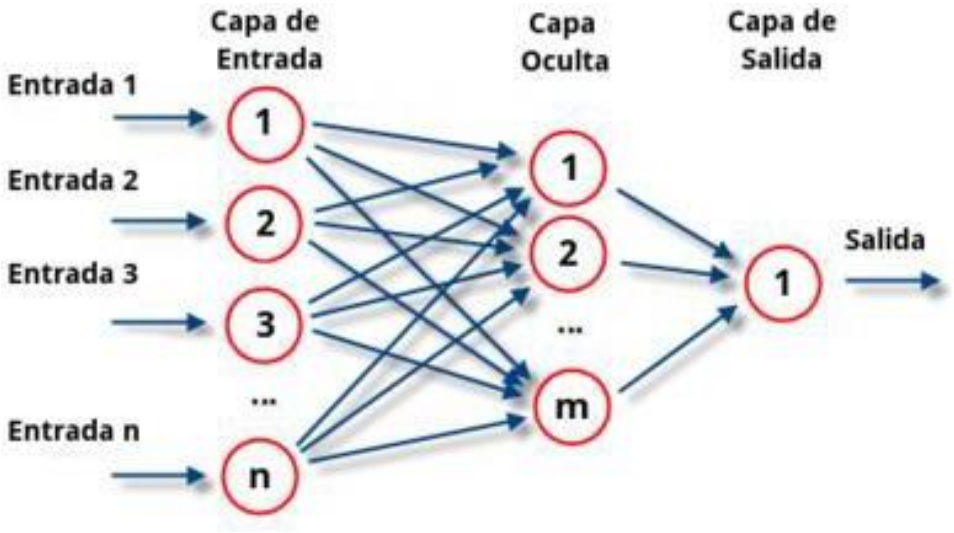

Figura 1: Estructura de una red neuronal.

Clasificación de redes neuronales artificiales según la topología red: Existen varios como lo menciona [3] en este caso solo se nombrarán a algunos:

- Red neuronal Monocapa-Perceptrón simple: Esto corresponde a una red neuronal más simple, está compuesta por una capa de neuronas que proyectan las entradas a una capa de neuronas de salida donde se realizan los diferentes cálculos.

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

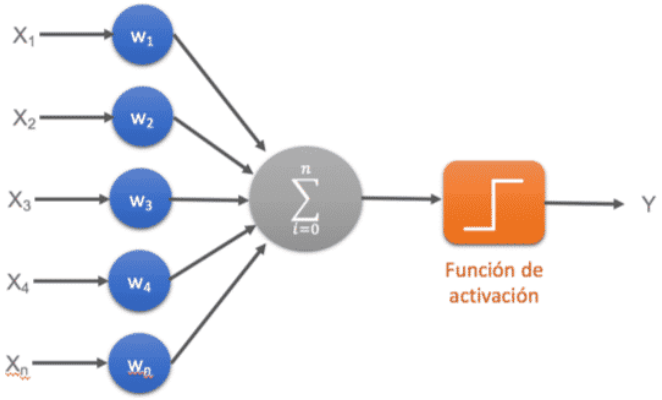

Figura 2: Gráfica de una red neuronal Monocapa-Perceptron simple.

- Red neuronal Multicapa- Perceptrón multicapa: Es una generalización de la red neuronal monocapa, la diferencia reside en que mientras la red neuronal monocapa está compuesta por una capa de neuronas de entrada y una capa de neuronas de salida, esta dispone de un conjunto de capas intermedias (capas ocultas) entre la capa de entrada y la de salida. Dependiendo del número de conexiones que presente la red esta puede estar total o parcialmente conectada.

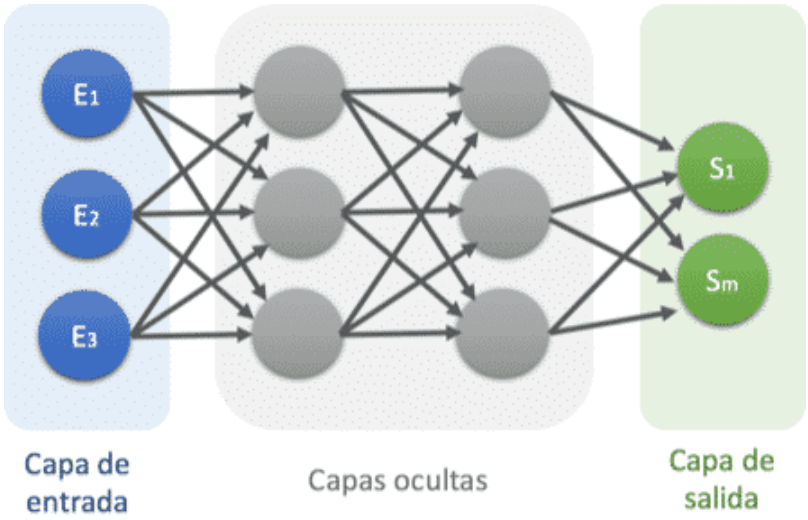

Figura 3: Gráfica de red neuronal multicapa.

- Red neuronal recurrente: No tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos, con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria.

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

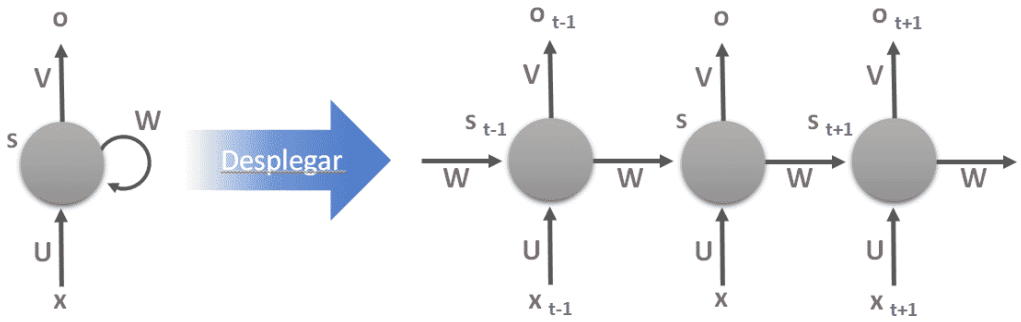

Figura 4: Grafica de red neuronal recurrente.

Clasificación de redes según el método de aprendizaje: Existen varios como lo menciona [4] pero en este caso solo se mencionarán algunos:

- Aprendizaje por corrección de error: Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los obtenidos en la salida. Ejemplos de algoritmos:
  - Perceptrón
  - Delta o Mínimo error cuadrado (LMS Error: Least Mean Squared)
  - Backpropagation o Programación hacia atrás (LMS multicapa)

- Aprendizaje estocástico: Consiste básicamente en realizar cambios aleatorios en los valores de los pesos y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad. Una red que utiliza este tipo de aprendizaje es la red Boltzman Machine, ideada por Hinton, Ackley y Sejnowski en 1984 y la red Cauchy Machine desarrollada por Szu en 1986 [4].

Resultados y discusión

Se utilizó como lenguaje de programación R a través del entorno de desarrollo como RStudio [5] o RStudioCloud [6]. Lo primero que se realizara es cargar el *dataset* con el siguiente comando:

```
#Forma1
dataset <- read.csv("pulsar_stars.csv")
#Forma2
data = read.csv("dataset.csv", fileEncoding = "Latin1", check.names = F)
```

Figura 5: Código para cargar el *dataset.*

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

El proceso de limpieza de datos permite identificar datos incompletos, incorrectos, inexactos, no pertinentes, etc. y luego substituir, modificar o eliminar estos datos sucios. Después de haber realizado el ingreso de datos se prosigue a analizar que las variables de entrada influyen en el resultado final, las imágenes analizadas son:

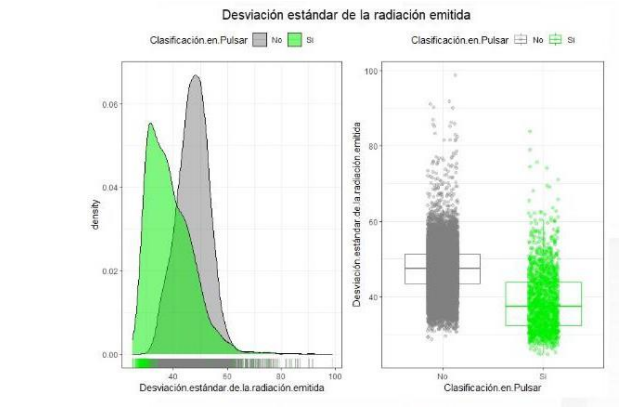

Figura 6: Gráfica de desviación estándar de la radiación emitida

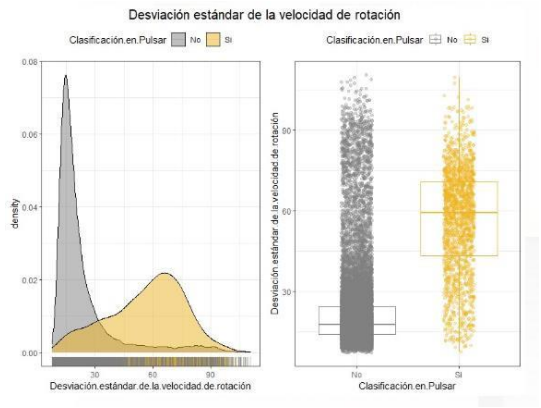

Figura 7: Gráfica de desviación estándar de la velocidad de rotación

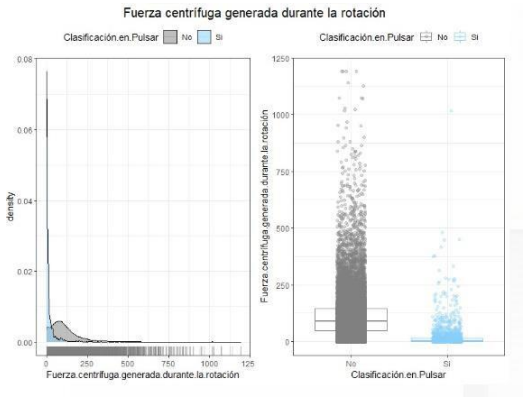

Figura 8: Grafica de fuerza centrífuga generada durante la rotación.

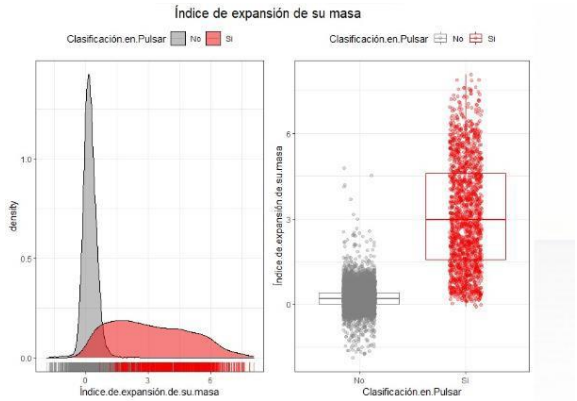

Figura 9: Grafica de índice de expansión de la masa.

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

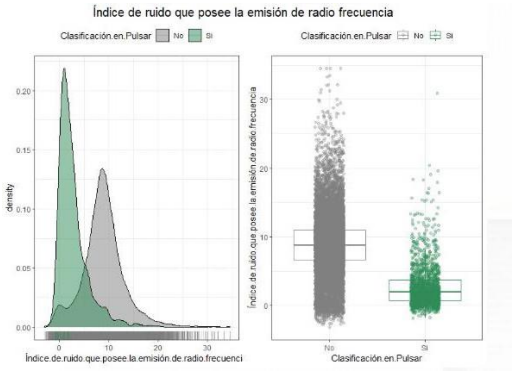

Figura 10: Gráfica de índice de ruido que posee la emisión de radio frecuencia.

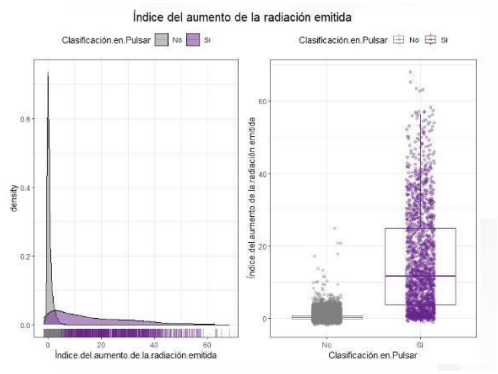

Figura 11: Gráfica de índice de aumento de la radiación emitida.

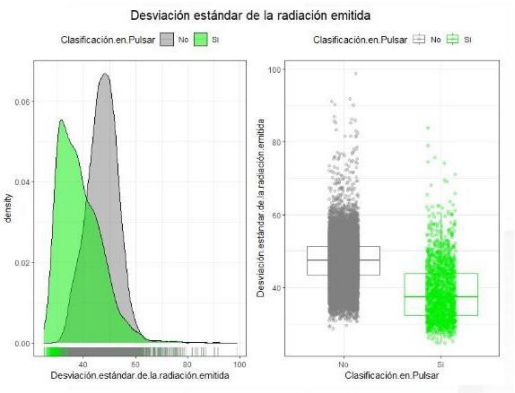

Figura 12: Gráfica de promedio de la velocidad de rotación.

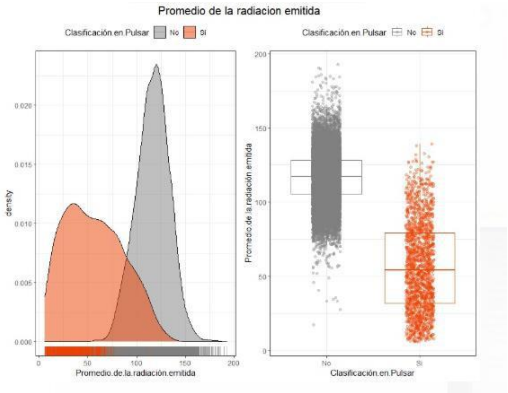

Figura 13: Gráfica de promedio de la radiación emitida.

Entonces se puede decir que las variables de entrada si influyen en el resultado final. No se cuenta con ninguna columna vacía ni fila vacía, tampoco se tiene que sustituir ninguna columna ya que todas son influyentes en el resultado final, para ello se comprueba que todas las columnas están totalmente llenas. A continuación, se muestran los datos estadísticos de cada una nuestras variables de entrada.

Una vez preparados los datos se procedió a utilizar la red neuronal artificial, para esto fue necesario descargar el paquete "h2o" [7], el cual es una librería de análisis predictivo y *machine learning* que incorpora la funcionalidad para crear RNA así como modelos de *deep learning.* Ahora podemos hacer uso de nuestro objeto clasificador mediante la función *h2o.deeplearning*.

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

```
> summary(data)
 Promedio.de.la.radiación.emitida Desviación.estándar.de.la.radiación.emitida Índice.de.expansión.de.su.masa
 Min.   :  5.812                  Min.   :24.77                               Min.   :-1.8760
 1st Qu.:100.930                  1st Qu.:42.38                               1st Qu.: 0.0271
 Median :115.078                  Median :46.95                               Median : 0.2232
 Mean   :111.080                  Mean   :46.55                               Mean   : 0.4779
 3rd Qu.:127.086                  3rd Qu.:51.02                               3rd Qu.: 0.4733
 Max.   :192.617                  Max.   :98.78                               Max.   : 8.0695
 Índice.del.aumento.de.la.radiación.emitida Promedio.de.la.velocidad.de.rotación
 Min.   :-1.7919                           Min.   :  0.2132
 1st Qu.:-0.1886                           1st Qu.:  1.9231
 Median : 0.1987                           Median :  2.8018
 Mean   : 1.7703                           Mean   : 12.6144
 3rd Qu.: 0.9278                           3rd Qu.:  5.4643
 Max.   :68.1016                           Max.   :223.3921
 Desviación.estándar.de.la.velocidad.de.rotación Índice.de.ruido.que.posee.la.emisión.de.radio.frecuencia
 Min.   :  7.37                                  Min.   :-3.139
 1st Qu.: 14.44                                  1st Qu.: 5.782
 Median : 18.46                                  Median : 8.434
 Mean   : 26.33                                  Mean   : 8.304
 3rd Qu.: 28.43                                  3rd Qu.:10.703
 Max.   :110.64                                  Max.   :34.540
 Fuerza.centrífuga.generada.durante.la.rotación Clasificación.en.Pulsar
 Min.   :  -1.977                               Length:17898
 1st Qu.:  34.961                               Class :character
 Median :  83.065                               Mode  :character
 Mean   : 104.858
 3rd Qu.: 139.309
 Max.   :1191.001
>
```

Figura 14: Gráfica de promedio de la velocidad de rotación.

En esta función el parámetro *activation* hace referencia a la función de activación que utilizará cada una de las neuronas que en este caso es Rectifier (rectificador). Mientras que *hidden* hace referencia a la cantidad de capas ocultas, así como neuronas en cada capa de ellas, en este caso se usaron dos capas ocultas con 7 neuronas en cada una de ellas. Por último, *epoch* hace referencia a la cantidad de veces que se le pasaran los datos de entrenamiento a fin de aplicar el algoritmo de aprendizaje.

```
> classifier = h2o.deeplearning(y = 'TipoEstrella',
+                               training_frame = as.h2o(training_set),
+                               activation = 'Rectifier',
+                               hidden = c(6, 6),
+                               epochs = 500,
+                               train_samples_per_iteration = -2)
```

Figura 15: Creación del modelo de red neuronal.

Una vez finalizado el entrenamiento, podemos realizar predicciones sobre el conjunto de validación de la siguiente manera:

Revista Innovación y Software
Vol. 2, No. 1, Mes Marzo-Agosto, 2021
ISSN: 2708-0935
Pág. 33-42
https://revistas.ulasalle.edu.pe/innosoft

```
> prob_pred <- h2o.predict(classifier, newdata = as.h2o(test_set))
  |======================================================================| 100%
  |======================================================================| 100%
```

Figura 16: Entrenamiento de la red neuronal.

Obteniendo así una matriz que nos indicará cuantos datos han sido predichos de manera correcta, dándonos las respuestas obtenidas vs las respuestas operadas, en este caso se obtuvieron 4388 respuestas correctas y 87 respuestas erradas, teniendo así un modelo que tiene una efectividad del 98.017%.

```
> cm <- table(y_test_set, y_pred)
> cm
          y_pred
y_test_set    0    1
         0 4022   43
         1   44  366
```

Figura 15: Resultados de la comprobación del entrenamiento.

Conclusiones

Después de desarrollar la investigación y la experimentación de este caso de estudio se llegó a las siguientes conclusiones:

- Las redes neuronales multicapa son muy útiles para problemas de clasificación.
- El lenguaje de programación R cuenta con muchas librerías muy útiles para el procesamiento de datos.
- Todas las variables de entrada influyen en la decisión de clasificar una estrella en pulsar o no, por ende, no pueden ser eliminadas.
- La efectividad obtenida con una red neural con una sola neurona en la capa oculta es mejor que con dos neuronas.
- El error menor lo presenta la red neuronal con una sola neurona en la capa oculta.
- La red neuronal falla más con al momento de clasificar a las que si son pulsares.

Referencias

[1] C. C. Aggarwal, “Neural networks and deep learning,” Springer, vol. 10, pp. 978–3, 2018.

[2] J. Schmidhuber, “Deep learning in neural networks: An overview,” Neural networks, vol. 61, pp. 85–117, 2015.

[3] V. Kreinovich, “From traditional neural networks to deep learning: towards mathematical foundations of empirical successes,” in Recent Developments and the New Direction in Soft-Computing Foundations and Applications, Springer, 2021, pp. 387–397.

[4] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton, “Optimizing deep learning hyper-parameters through an evolutionary algorithm,” in Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, 2015, pp. 1–5.

[5] J. Allaire, “RStudio: integrated development environment for R,” Boston, MA, vol. 770, p. 394, 2012.

[6] J. M. Elias, “Webinar sobre la docencia en línea con RStudio Cloud,” IDP: revista d’Internet, dret i política, no. 31, 2020.

[7] S. Aiello, E. Eckstrand, A. Fu, M. Landry, and P. Aboyoun, “Machine Learning with R and H2O,” H2O booklet, vol. 550, 2016.