Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

Tipo de artículo: Artículos originales
Temática: Inteligencia artificial
Recibido: 15/10/2021 | Aceptado: 27/11/2021 | Publicado: 30/03/2022

Identificadores persistentes:
ARK: ark:/42411/s8/a51
PURL: 42411/s8/a51

Sistema automático para calificación de vino mediante Redes Neuronales.

Automatic system for wine qualification through Neural Networks

Diego Richard Rivera Demanuel [1], Cleofe Huamani Huancara [2*], Yimy Alfredo Charca Ccama [3]

[1] Universidad Nacional de San Agustín. Arequipa, Perú. driverad@unsa.edu.pe
[2] Universidad Nacional de San Agustín. Arequipa, Perú. chuamanihu@unsa.edu.pe
[3] Universidad Nacional de San Agustín. Arequipa, Perú. ycharca@unsa.edu.pe

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

Resumen
Tratamiento de datos para la calificación de vinos, este informe detalla el proceso seguido, en donde se utilizó el lenguaje de programación Phyton, para el análisis de los datos del dataset, se utilizó el servidor Google Colab para ejecutar los algoritmos en la nube ya que el equipo considero que la velocidad de análisis de datos en google colab es más rápido. Las redes neuronales tienen capacidad de aprender y realizar tareas basadas en un entrenamiento inicial llamado aprendizaje adaptativo y además de que son tolerantes a los fallos.

Palabras clave: Redes neuronales, Tratamiento de datos, Datos masivos.

Abstract
Treatment of data for the qualification of wines, this report details the process followed, where the Python programming language was used, for the analysis of the data of the dataset, the Google Colab server was used to execute the algorithms in the cloud since the team considered that the speed of data analysis in Google Collab is faster. Neural networks have the ability to learn and perform tasks based on an initial training called adaptive learning and are also fault-tolerant.

Keywords: *Neural networks, Data processing, Big data.*

Introducción

Tratamiento de datos es un término utilizado cuando se trata de analizar datos en su mayoría masivos, existen muchos lenguajes de programación con librerías para el tratamiento de datos, y técnicas a utilizar, en este trabajo se presenta un informe sobre un sistema de calificación de vinos para una empresa productora de vino que desea un sistema informático que le permita calificar los vinos que esta produce, se consideró también de que los errores del sistema son mínimos considerando de que el porcentaje de errores es casi nula, en [13] recomienda el uso de la arquitectura de redes neuronales de la Inteligencia artificial donde se emplea el uso del lenguaje de programación Phyton y Jupyther Notebook como IDE. Se evaluaron los datos del Dataset proporcionado por el docente de la asignatura, el sistema permite al usuario poder definir una cantidad aleatoria de los casos del dataset para poder usarlos en el entrenamiento y definir otra cantidad aleatoria para poder utilizarlos en el entrenamiento, también se tomó en cuenta de que los casos utilizados en el entrenamiento no podrán ser utilizados en la validación del entrenamiento para luego mostrar las medidas de los errores.

El desarrollo del sistema requiere análisis de datos masivos, donde es necesario transformar la dataset, seleccionar un lenguaje de programación y buscar las librerías que trabajan con las redes neuronales, así lo afirma [14].

Materiales y métodos o Metodología computacional

En este trabajo de investigación se utilizaron los datos de una empresa productora de vino, la dataset fue proporcionado por el docente del curso de Inteligencia Artificial, las principales herramientas son el lenguaje de programación Python con los respectivos paquetes y Jupyther notebook como IDE principal, se utilizó redes neuronales como metodología computacional, consiste en un conjunto de unidades a las que se le llaman neuronas artificiales que van conectadas entre sí para transmitir las señales. En [12] afirma que en esta metodología la información que se utiliza como entrada atraviesa la red neuronal para ser sometida a diversas operaciones que tienen como fin producir valores de salida es así que el Sistema Automático califica los vinos que la empresa produce.

Adicionalmente a esto se utilizó Google Colab que es un entorno portátil de Jupyter que tiene un uso gratuito y se ejecuta completamente en la nube, una característica resaltante de Google Colab es que no requiere configuración y los miembros del equipo pueden editar simultáneamente los cuadernos que se crean, además admite muchas bibliotecas de aprendizaje automático populares que se pueden cargar fácilmente en su computadora portátil, Google Colab se utilizó para pruebas de entrenamiento, optimización de subparámetros y ejecuciones.

Decidimos utilizar la plataforma de Google Colab porque consideramos que el análisis de datos nos tomaría más tiempo en nuestras computadoras personales.

También se escogió la arquitectura del tipo Perceptrón Multicapa que evoluciona del perceptrón simple en donde incorpora capas de neuronas ocultas, con esto consigue representar funciones no lineales, además está compuesto por por una capa de entrada, una capa de salida y n capas ocultas entremedias. En [15] afirma que el perceptrón multicapa tiene salidas disjuntas pero relacionadas entre sí, de tal manera que la salida de una neurona es la entrada de la siguiente.

Descripción general de la aplicación.

Una empresa productora de vino desea hacer un sistema informático que le permita realizar la calificación de los vinos que esta produce. En dicho sistema desean utilizar inteligencia artificial para tal efecto. Para esto cuentan con una base de calificaciones históricas hechas por expertos con un total de 6 497 casos. Cada caso cuenta con los siguientes atributos: tipo de vino (blanco o tinto), nivel fijo de la acidez, nivel variable de la acidez, nivel de ácido cítrico, nivel de azúcar residual, cantidad de cloruros, nivel libre de dióxido de azufre, nivel de dióxido de azufre, densidad, pH, nivel de sulfatos, grado de alcohol y la calificación del vino (1 a 10).

Se desea que el sistema sea capaz de calificar un vino en la escala de 1 a 10 dando los valores de los atributos antes mencionados. Para esta tarea utilice el 75% la base de calificaciones anteriores para entrenar el modelo y el 25% de lecturas restantes para comprobar el entrenamiento de forma aleatoria; tenga en cuenta que estos parámetros (el porciento de entrenamiento y el de validación del mismo) son suministrados por el usuario. El sistema puede equivocarse como máximo 1 vez entre 10000 veces y el tiempo con que se cuenta para dicho entrenamiento es ilimitado. También debe ser posible guardar el modelo una vez entrenado en un fichero y poder cargarlo en otro momento. El usuario podrá cargar el fichero donde se encuentran las calificaciones históricas, definir los parámetros del entrenamiento y visualizar el error una vez ya entrenado el modelo. También se le podrá suministrar un nuevo conjunto de atributos al modelo ya entrenado y este debe ser capaz de calificar el vino.

Selección y justificación del modelo.

Para la selección del modelo con cual se desarrolló el sistema, evaluamos los aspectos de los modelos aprendidos en clase, en donde resaltamos al modelo de redes neuronales como metodología computacional que consiste en un conjunto de unidades a las que se le llaman neuronas artificiales que van conectadas entre sí para transmitir las señales. En esta

metodología la información que se utiliza como entrada atraviesa la red neuronal para ser sometida a diversas operaciones que tienen como fin producir valores de salida. En [12] menciona que las redes neuronales tienen capacidad de aprender y realizar tareas basadas en un entrenamiento inicial llamado aprendizaje adaptativo. De esta forma, la máquina puede aprender a llevar a cabo ciertas tareas mediante el entrenamiento. Las redes neuronales permiten organizar por sí mismas lo aprendido, mientras que el aprendizaje es la modificación de cada elemento procesado, la auto organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico, [14] lo especifica.

Además, las redes neuronales tienen la capacidad de ser tolerantes a fallos los primeros métodos computacionales con esta capacidad inherente fueron las redes neuronales con esta característica si se produce un fallo en un número no muy grande de neuronas, el sistema no sufre una caída repentina, aunque el comportamiento si se ve influenciado.

Otra de las prioridades principales de las redes neuronales es que pueden operar en tiempo real, las redes neuronales trabajan mediante conexiones en paralelo, lo que permite grandes velocidades de transmisión y respuesta instantáneas. Por último, una de las ventajas que destaca es la facilidad de inserción en la tecnología existente.

Lenguaje de Programación y herramientas

Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional.

Python tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único, lo cual daría una enorme ventaja al desarrollador y hacen un aliado perfecto para la Inteligencia Artificial. [1]

Permite plasmar ideas complejas con unas pocas líneas de código, lo que no es possible con otros lenguajes. Existen bibliotecas como «Keras» y «TensorFlow", que contienen mucha información sobre las funcionalidades del aprendizaje automático.

¿Por qué Phyton?

Python es un lenguaje que todo el mundo debería conocer. En [16] menciona su sintaxis simple, clara y sencilla; el tipado dinámico, el gestor de memoria, la gran cantidad de librerías disponibles y la potencia del lenguaje y algunas características más hacen que desarrollar una aplicación en Python sea sencillo y muy.

La sintaxis de Python es tan sencilla y cercana al lenguaje natural que los programas elaborados en Python parecen pseudocódigo. Por este motivo se dice que es uno de los mejores lenguajes para comenzar a programar. Algunos casos de éxito en el uso de Python son Google, Yahoo, la NASA, Industrias Light & Magic, y todas las distribuciones Linux, en las que Python cada vez representa un tanto por ciento mayor de los programas con disponibilidad.

Anaconda

Anaconda es una Suite de código abierto que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la Ciencia de datos con Python. [7]

Características:

- Anaconda Navigator es una interfaz gráfica de usuario GUI bastante sencilla, pero con un potencial enorme.
- Puede gestionar de manera avanzada paquetes relacionados a la Ciencia de datos con Python desde la terminal.
- Permite compilar Python en código de máquina para una ejecución rápida.

Jupyter notebook

Es una aplicación web de código abierto, desarrollada utilizando lenguaje HTML agnóstico que permite crear, compartir y editar documentos, en los que se puede ejecutar código python, hacer anotaciones, insertar ecuaciones, visualizar resultados y documentar funcionalidades. Entre sus usos están: la limpieza y transformación de datos, la simulación numérica, el modelado estadístico, el aprendizaje automático, etc. [11]

Librerías y framework

Para el desarrollo de la aplicación se procedió a la instalación de librerías correspondientes a la herramienta elegida, el cual facilitará el tratamiento de los datos, y por ende el entrenamiento de este mismo.

Keras

En [17] nos dice que el mundo de las redes neuronales está en auge. El hecho de simular el cerebro humano en un ordenador, parece ser una de las tecnologías más prometedores de la informática, pero aún no se ha conseguido, aunque mediante algoritmos de machine learning, ya es posible entrenar máquinas para que aprendan de forma parecida a como lo haría nuestro cerebro.

Keras es un framework de alto nivel para el aprendizaje, escrito en Python y capaz de correr sobre los frameworks TensorFlow, CNTK, o Theano. Fue desarrollado con el objeto de facilitar un proceso de experimentación rápida. [4]

- Para instalar Keras desde la terminal de anaconda, ejecutamos el siguiente comando, conda install keras.

Pandas

En [19] no dice que Pandas es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para hacer que el trabajo con datos "relacionales" o "etiquetados" sea fácil e intuitivo. Pretende ser el elemento fundamental de alto nivel para realizar análisis de datos prácticos y del mundo real en Python, además de que proporciona estructuras de datos similares a los dataframes de R. Pandas depende de Numpy, la librería que añade un potente tipo matricial a Python. Los principales tipos de datos que pueden representarse con pandas son:

Datos tabulares con columnas de tipo heterogéneo con etiquetas en columnas y filas y Series temporales. [9]

- Para instalar Pandas desde la terminal de anaconda, ejecutamos el siguiente comando, conda install pandas.

Numpy

NumPy es una extensión de Python, es de código abierto, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. [8]

Es una biblioteca de Python para trabajar con arreglos multidimensionales lo que hace que el principal tipo de dato es el arreglo o array, también nos permite trabajar con la semántica de matrices y nos ofrece muchas funciones útiles para el procesamiento de números. [20]

- Para instalar Numpy desde la terminal de anaconda, ejecutamos el siguiente comando, conda install numpy.

Matplotlib

Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy. Proporciona una API, pylab, diseñada para recordar a la de MATLAB.[10].

Matplotlib.pyplot es una colección de funciones estilo comandos que hacen que matplotlib sea similar a matlab. Cada función de pyplot hace algunos cambios a la figura, funciones como crear figura, crear un área de impresión en una figura, imprimir líneas en el área de impresión, agregar etiquetas a las figuras, etc. [21]

- Para instalar Matplotlib desde la terminal de anaconda, ejecutamos el siguiente comando, conda install matplotlib.

Descripción de los procesos

Para tener una red neuronal artificial que funcione según la aplicación que se desee, se debe realizar antes de nada un aprendizaje con unos patrones conocidos. Por lo que para utilizar una red neuronal artificial, el primer paso es el entrenamiento de la misma. En él, se presentan un conjunto de entradas y salidas conocidas, que se propagan a través de una red hasta poder obtener unas ciertas salidas (feedforward), para que después se puedan modificar los pesos y bias de manera que la salida coincida con la deseada (backpropagation). El segundo paso trata de verificar la red mediante un conjunto de entradas y salidas también conocidas, y de la misma forma haciendo feedforward, pero sin necesidad de backpropagation, dado que en principio la red ya está entrenada correctamente. [2]

Por lo que para completar estos pasos ya mencionados se deben de seguir una serie de procesos.

Tratamiento de datos.

El procesamiento de datos es parte fundamental de cualquier proceso de aprendizaje automático. En algunos casos, busca corregir deficiencias en los datos que puedan dañar el aprendizaje, como omisiones, ruido y valores extremos. En otros casos, se persigue adaptar los datos al modelo que se pretende entrenar para simplificar u optimizar el proceso. [3]

En la dataset proporcionado, se tenían campos anómalos que a simple vista se podía ver, sin embargo, para el tratamiento de estos se tuvo que elegir diferentes métodos, de acuerdo a la herramienta elegida, dado que son más de 6000 datos. Por lo que se procedió a la transformación de estos.

Transformación de datos

Para este apartado se seguirá una secuencia de pasos:

- El primer paso que haremos será identificar las variables dependientes e independientes.

Variables Dependientes:

x_0: Tipo

x_1: Acidez fija

x_2: Acidez variable

x_3: Ácido cítrico

x_4: Azúcar residual

x_5: Cloruros

x_6: Dióxido de azufre libre

x_7: Nivel de dióxido de azufre

x_8: Densidad

x_9: pH

x_10: Sulfatos

x_11: Grado de alcohol

Variables Independientes:

y : Calidad

- El segundo paso que haremos es: identificar nuestros campos o variables categóricas y la existencia de valores outliers en dichos campos.

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

En la dataset que se nos proporcionó, se identificó sólo un campo categórico, “Tipo”, por lo que se le aplicó el preprocesamiento, convertir variables categóricas en variables numéricas.

También se identificaron 3 campos con valores outliers, “Densidad”, “Cloruros”, “Acidez variable”, por lo que de la misma forma se realizó el preprocesamiento.

Para ver la distribución de valores de dichos campos, se generó un histograma de cada uno de ellos. Esto con la función hist(), después se generó el diagrama de caja y bigotes, esto para detectar los outliers.

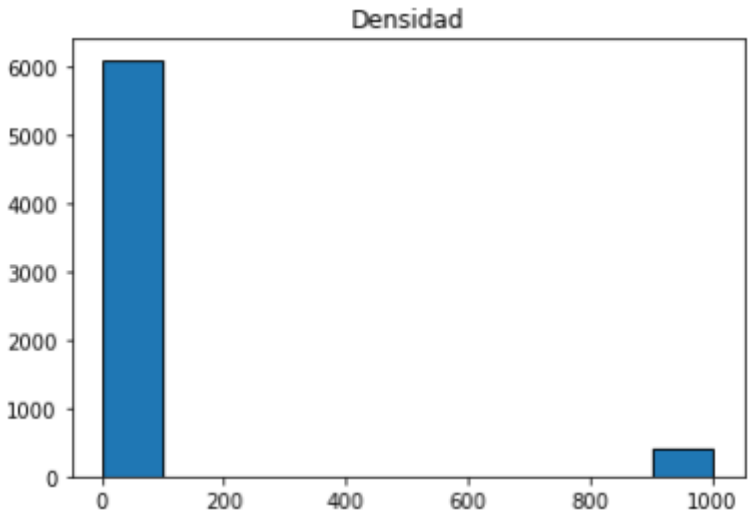

Figura 1. Histograma Densidad

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

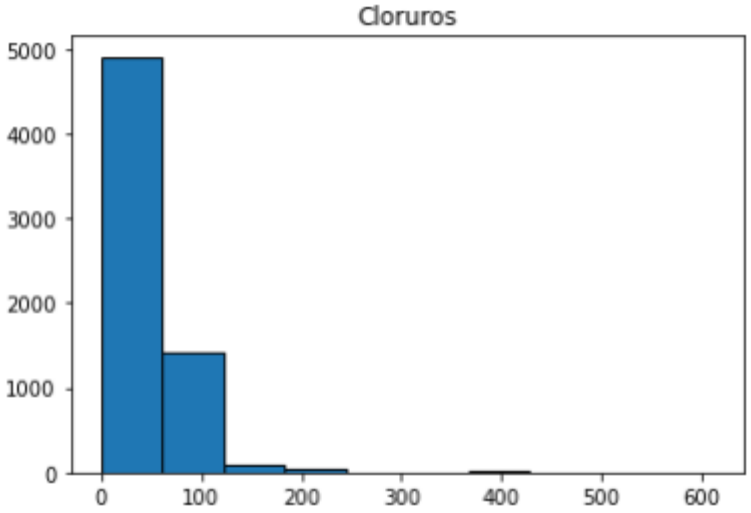

Figura 2. Histograma Cloruros

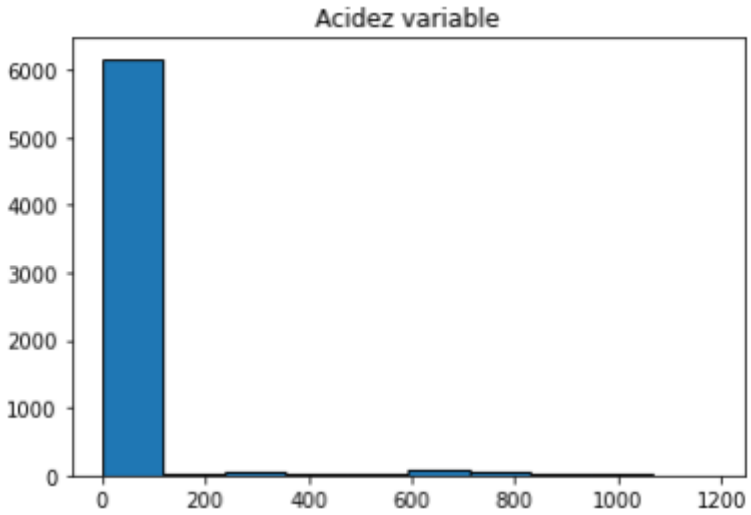

Figura 3. Histograma Acidez variable

En la figura 1, 2 y 3; se puede observar una cantidad de datos atípicos.

Para la identificación de outliers en la dataset, utilizamos el gráfico de caja y bigotes. el cual se formará con la función boxplot() del paquete matplotlib.

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

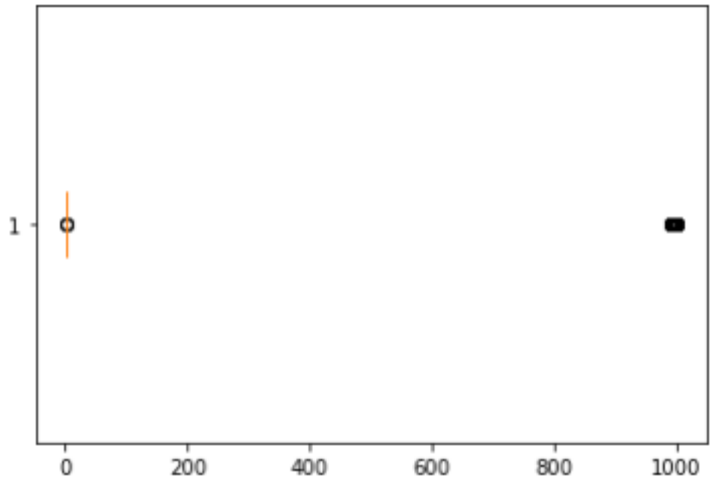

Figura 4. Diagrama de Caja y Bigotes Densidad.

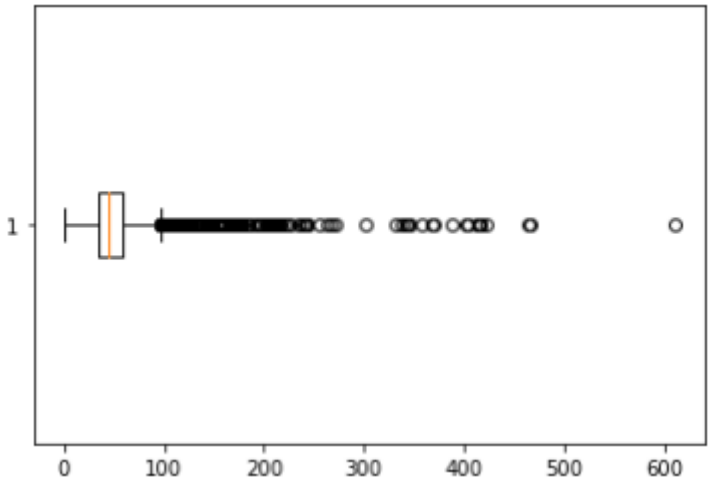

Figura 5. Diagrama de Caja y Bigotes Cloruro.

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

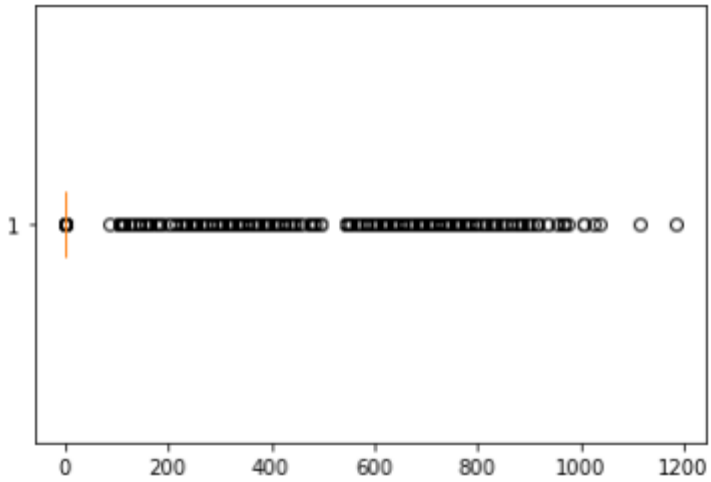

Figura 6. Diagrama de Caja y Bigotes Acidez variable.

La identificación de los valores outliers se realizó de la siguiente manera:

- Se obtuvo los valores del primer cuartil, tercer cuartil, rango inter cuartil, y mediana, Todo esto para hallar los valores inferior y superior del bigote, y seguido de esto reemplazar los valores outliers, hallados.

```
Q1=dataset['Variable'].quantile(0.25)
Q3=dataset['Variable'].quantile(0.75)
IQR=Q3-Q1
mediana=dataset['Variable'].median()
```

Figura 7. Obtención de valores.

- El siguiente paso, para hallar los valores inferior y superior del bigote

```
B_inf=(Q1 -1.5 * IQR)
B_sup=(Q3 +1.5 * IQR)
```

Figura 8. Obtención de bigotes.

- Como paso final para el tratamiento de datos, procedemos a reemplazar los valores outliers con la mediana, de estos.

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

```
dataset['Densidad']=np.where(ubicacion_outliers,mediana,dataset['Densidad'])
```

Figura 9. Reemplazo outliers por mediana.

Entrenamiento y Validación

- **Procesamiento de Datos**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Figura 10. Implementación de librerías.

```
%matplotlib inline
df=pd.read_csv('dataset_train.csv',engine='python', index_col=0)
df.head()
```

Figura 11. Importación del dataset.

```
x=df[['Tipo','Acidez fija','Acidez variable','Ácido cítrico','Azucar residual','Cloruros','Dióxido de azufre libre','Nivel de di
    ,'Densidad','pH', 'Sulfatos','Grado de alcohol' ]]
x
```

Figura 12. Variables independientes.

```
y=df[('Calidad')]
y
```

Figura 13. Variable dependiente.

Revista Innovación y Software
Vol. 3, No. 1, Mes Marzo-Agosto, 2022
ISSN: 2708-0935
Pág. 30-46
https://revistas.ulasalle.edu.pe/innosoft

```
model=Sequential()
model.add(Dense(48, input_dim=12, activation='relu' ))
model.add(Dense(64, activation='relu'))
model.add(Dense(10, activation='softmax'))
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
model.fit(x, y, epochs=10000, batch_size=64)
```

Figura 14. Aplicación de capas.

- **Entrenamiento**

Para poder entrenar el modelo ya establecido se necesitó de 10000 iteraciones con las cuales se pudo llegar al resultado de una precisión del 81.75%.

```
scores = model.evaluate(x, y)
print("\n%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

```
153/153 [==============================] - 0s 979us/step - loss: 0.4490 - accuracy: 0.8175

accuracy: 81.75%
```

Figura 15. Entrenamiento.

- **Validación**

Posteriormente se cargaron los datos de la dataset de prueba para poder evaluar nuestro modelo ya planteado con anterioridad.

```
scores_test = model.evaluate(x_test, y_test)
print("%s: %.2f%%" % (model.metrics_names[1], scores_test[1]*100))
```

```
51/51 [==============================] - 0s 946us/step - loss: 4.8618 - accuracy: 0.5606
accuracy: 56.06%
```

Figura 16. Validación.

Conclusiones

- Usualmente se usan redes neuronales para el tratamiento de datos.

- Las redes neuronales tienen la capacidad de ser tolerantes a fallos y son los primeros métodos computacionales con esta capacidad inherente
- La evolución de la tecnología con redes neuronales busca asimilarse a un cerebro humano con funciones de aprendizaje y así poder analizar una vasta cantidad de datos mucho más rápido y minimizando errores.
- Los algoritmos para el análisis de datos definen el resultado de un análisis en donde se califican los errores con el fin de reducirlos y hacer más eficiente al sistema.

Referencias

[1] Python.[Online]. Available: https://es.wikipedia.org/wiki/Python

[2] E. Lazarte, "Entrenamiento de una red neuronal hardware desde matlab (hardware in the loop)," Madrid, Febrero, 2017. Available: http://oa.upm.es/45516/1/TFG_EDUARDO_LASARTE_ZAPATA.pdf/.

[3] D. Peralta, A. Herrera, F. Herrera, "Un estudio sobre el preprocesamiento para Redes Neuronales Profundas y Aplicación sobre Reconocimiento de Dígitos Manuscritos,"September 2016 . [Online], Available: https://eusal.es/index.php/eusal/catalog/download/978-84-9012-632-5/5256/5486-1?inline=1

[4]Keras.[Online].Available: https://keras.io/

[5]Tensorflow.[Online]. Available:https://www.tensorflow.org/

[6]Machine Learnig con tensorflow y keras en python.[Online]. Available: https://www.luisllamas.es/machinelearning-con-tensorflow-y-keras-en-python/

[7] Anaconda Distribution: La Suite más completa para la Ciencia de datos con Python.[Online]. Available: https://blog.desdelinux.net/ciencia-de-datos-con-python/

[8] Numpy.[Online].Available: https://damianavila.github.io/Python-Cientifico-HCC/3_NumPy.html

[9]Bioinformatics at COMAV.[Online].Available https://bioinf.comav.upv.es/courses/linux/python/pandas.html

[10]Matplotlib. [Online].Available: https://es.wikipedia.org/wiki/Matplotlib.

[11]Jupyter notebook: documenta y ejecuta código desde el navegador.[Online].Avaliable https://blog.desdelinux.net/jupyter-notebook/

[12] D. Matich, "Redes Neuronales: Conceptos Básicos y Aplicaciones," Universidad Tecnológica Nacional – Facultad Regional Rosario Departamento de Ingeniería Química - Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ), Mar. 2021. [Online] Disponible en: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf

[13] J. Montaño, "Redes Neuronales Artificiales aplicadas al Análisis de Datos," UNIVERSITAT DE LES ILLES BALEARS - Facultad de Psicología, 2002. [Online] Disponible en:https://www.tesisenred.net/bitstream/handle/10803/9441/tjjmm1de1.pdf?sequence=1

[14] A. Serrano, E. Soria, J. Martín, "Redes neuronales artificiales," Escola Tecnica Superior de Enginyeria - Departament de Enginyeria Electronica, 2010, [Online] Disponible en: http://ocw.uv.es/ingenieria-y-arquitectura/1- 2/libro_ocw_libro_de_redes.pdf

[15] F. Lara, "Fundamentos de redes neuronales artificiales," Laboratorio de cibernética aplicada, centro de instrumentos UNAM, [Online] Disponible en: http://conceptos.sociales.unam.mx/conceptos_final/598trabajo.pdf

[16] R Gonzales, "Phyton para todos," [Online] Disponible en: http://www.utic.edu.py/citil/images/Manuales/Python_para_todos.pdf

[17] C. Antona, "Herramientas modernas en redes neuronales: La librería Keras," Universidad autónoma de Madrid, Ene. 2017, [Online] Disponible en: https://repositorio.uam.es/bitstream/handle/10486/677854/antona_cortes_carlos_tfg.pdf?sequence=1&isAllowed=y

[18] D. Conde, "Inteligencia Artificial con Tensor Flow para la predicción de comportamientos," Departamento de Ingeniería telemática – Escuela Técnica Superior de Ingeniería – Universidad de Sevilla, 2018, [Online] Disponible en: https://pdfs.semanticscholar.org/bf01/10c9822592997a8ecef35bb3963e86276c54.pdf

[19] Rip Tutorial, "Aprendizaje pandas," [Online] Disponible en: https://riptutorial.com/Download/pandas-es.pdf

[20] F. Batista, "Numpy + Scipy", [Online] Disponible en: http://www.taniquetil.com.ar/homedevel/presents/numsci.pdf

[21] J. Padilla, "Gráficas con la librería Matplotlib para Python," [Online] Disponible en: http://jpadilla.docentes.upbbga.edu.co/Logica_y_Algoritmia/Graficas%20con%20la%20libreria%20Matplotlib%20para%20Python.pdf