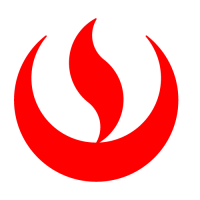
UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

CARRERA PROFESIONAL DE INGENIERÍA DE SOFTWARE



Asignatura:

SI404- INTELIGENCIA ARTIFICIAL

Sección: VW71

**Informe de Aplicación:**

**TRABAJO 02**

**Presentado por**

Sanchez Virto, Christian Fast

Zegarra Arones Frank Gustavo

Valdiviezo Chumbes Fernando Piero

Campus Monterrico,

Noviembre de 2020

**CONTENIDO**

[**I.** **Descripción del problema** 3](#_Toc55989738)

[**II.** **Descripción de recolección de datos** 3](#_Toc55989739)

[**III. Descripción de arquitectura de Red Neuronal** 3](#_Toc55989740)

[**IV.** **Detalles del código fuente** 3](#_Toc55989741)

[**V.** **Pruebas de uso y ejecución** 3](#_Toc55989742)

[**VI.** **Referencias** 3](#_Toc55989743)

# **Descripción del problema**

En ciertos lugares del mundo existen animales que están en peligro de extinción. Uno de estos animales es el pingüino, el cual vive en el continente Antártico. La causa de este problema es el calentamiento global que está afectando el hábitat de estos animales. Entre las especies de pingüinos que están en peligro de extinción se encuentran las de tipo Adelie, Gentoo y Chinstrap.

Por lo tanto, para salvarlos y ponerlos en cautiverio en un hábitat superficial en caso de su extinción total se utilizará un programa de clasificación de pingüinos que permita clasificarlos por especies. Por cada especie clasificada, se seleccionarán parejas para llevarlos a un hábitat superficial y puedan reproducirse con los cuidados necesarios y con ello evitar la pérdida de estas especies en el futuro.

# **Descripción de recolección de datos**

Los datos recopilados involucran una serie de características físicas de los pingüinos desde su especie hasta el tamaño de su pico y aleta. Con la información de estos datos, se implementará un algoritmo supervisado de inteligencia artificial. Entre la información recopilada se encuentra:

* Tipo de especie.
* Hábitat natural de convivencia.
* Longitud del pico.
* Profundidad del pico.
* Longitud de la aleta.
* Masa corporal.
* Sexo.

Luego de estas características ingresadas a la red neuronal, esta será capaz de entender el patrón de comportamiento brindando así un resultado preciso y esperado.

# **Descripción de arquitectura de Red Neuronal**

Para el desarrollo del algoritmo de inteligencia artificial se implementará una red neuronal artificial perceptrón multicapa el cual posee múltiples capas, entre ellas capa de entrada, oculta y salida además de su otra funcionalidad propagación hacia atrás.

La capa de entrada introduce a la red neuronal el patrón de comportamiento, en esta etapa no se realiza procedimiento lógico. La capa oculta es desarrollada mediante los datos obtenidos de la capa anterior, y las salidas componen a las siguientes capas posteriores conformado por la capa de salida que es el resultado final de toda la red neuronal.

Una de las características más representativos de este algoritmo es la propagación hacia atrás (back-propagation) que busca corregir los pesos de la red neuronal logrando así un resultado con mayor precisión para que luego de un entrenamiento con los datos de prueba, las salidas esperadas serán el tipo de especie del pingüino más cercanos al patrón de comportamiento ingresado y así solucionar el problema planteado anteriormente.

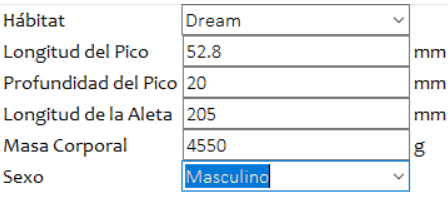
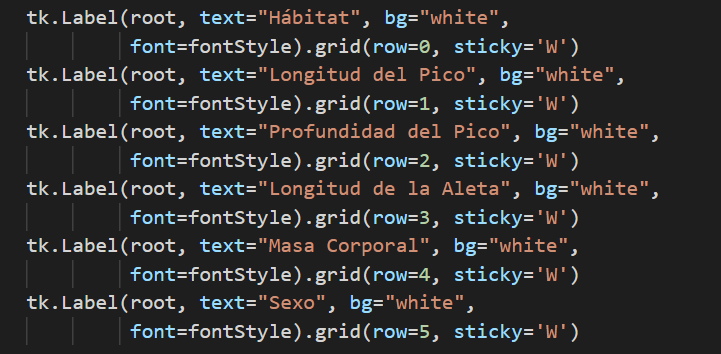
# **Detalles del código fuente**

**Librerías**

Las librerías que se utilizaron fueron:

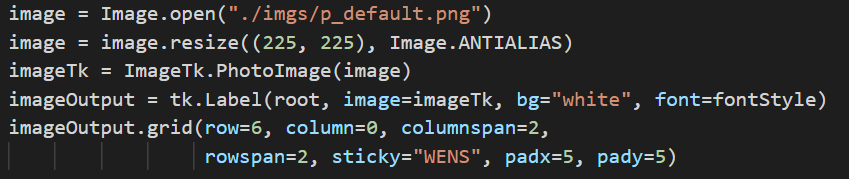
Tkinter

Librería que sirvió de apoyo para elaborar la interfaz gráfica mediante el uso de combo box, text box, labels, buttons etc. Ejemplo:

Pillow

Librería que sirvió para la lectura y escalamiento de las imágenes que se muestran en la interfaz. Cada imagen facilita la identificación de la especie de pingüino detectada por el algoritmo. Ejemplo:

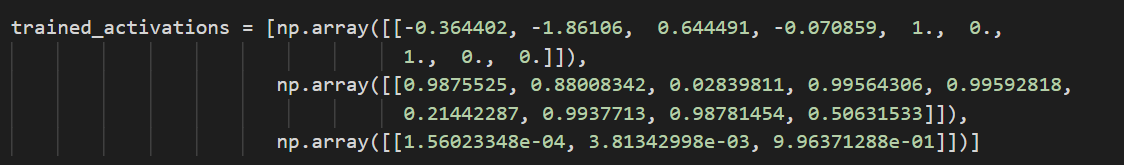
 

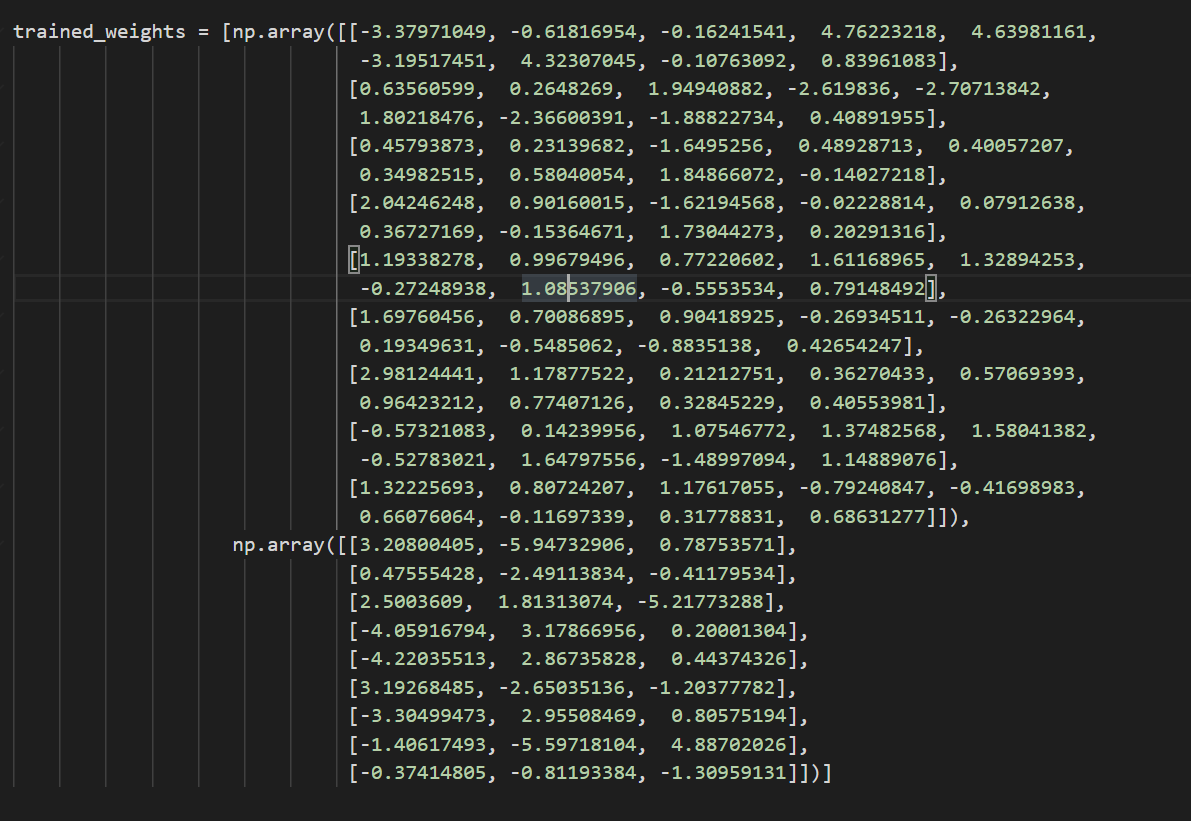
Pandas

Biblioteca escrita como extensión de NumPy para la manipulación y análisis de datos para el lenguaje de programación Python. Ofrece estructuras de datos y operaciones para manipular tablas numéricas y series temporales.

Numpy

Librería utilizada para cálculo de operaciones matemáticas realizadas por la red neuronal y para almacenar los datos que este arroja. Ejemplo:

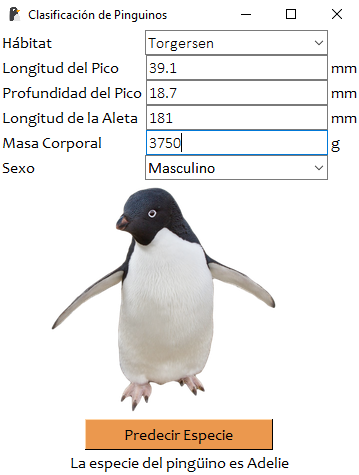




# **Pruebas de uso y ejecución**

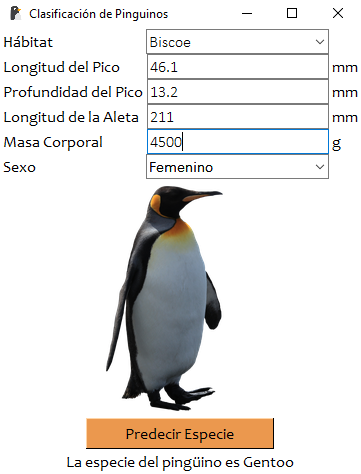
Captura 1

La aplicación muestra como resultado a un pingüino de la especie Adelie en caso de que este fuera el resultado obtenido. Incluye además una imagen representativa de esta especie de pingüino y un mensaje señalando el nombre de la especie.



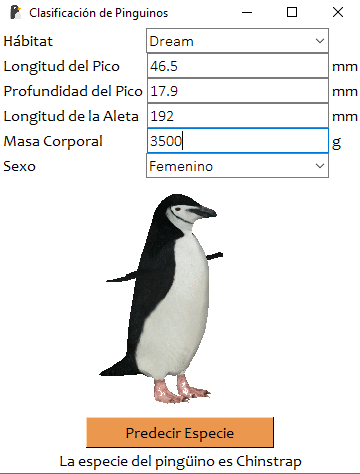
Captura 2

La aplicación muestra como resultado a un pingüino de la especie Gentoo en caso de que este fuera el resultado obtenido. Incluye además una imagen representativa de esta especie de pingüino y un mensaje señalando el nombre de la especie.



Captura 3

La aplicación muestra como resultado a un pingüino de la especie Chinstrap en caso de que este fuera el resultado obtenido. Incluye además una imagen representativa de esta especie de pingüino y un mensaje señalando el nombre de la especie.



Captura 4

En caso de que el resultado no se pueda predecir debido a ingresar parámetros no correspondientes a las características de un pingüino, la aplicación muestra una imagen default y el mensaje de que no se pudo determinar la especie.



# **Referencias**

LeNail, A. (2019). NN-SVG: *Publication-Ready Neural Network Architecture Schematics*. Journal of Open Source Software, 4(33), 747, Recuperado de https://doi.org/10.21105/joss.00747

Nielsen, M. (2019). *Neural Networks and Deep Learning*. Capítulo 1. Recuperado de http://neuralnetworksanddeeplearning.com/chap1.html

Velardo V. [The Sound of AI] (2020). *Training a neural network: Backward propagation and gradient descent*. [Video]. Youtube. https://youtu.be/ScL18goxsSg