****

**Universidad de Panamá**

**Facultad de Informática, Electrónica y Comunicación**

**Laboratorio de Inteligencia Artificial**

**Integrantes:**

Roderick Aparicio 8-916-593

Isaac Bethancourt 8-917-2263

Alan Castro 8-912-1890

Jesús De Gracia 8-1086-1646

Guillermo Espino 8-925-2235

Daniel González 8-907-1404

Allan Marín EC-20-12127

Elvin Marín EC-20-12126

David Morán 2-741-87

Lourdes Moreno 8-920-640

Kaiser Obaldía 8-898-703

Yenny Ortega 8-923-1263

Fermín Prováz 8-932-1661

Kevin Rueda 8-910-1800

Efraín Samudio 8-772-1697

Krisbel Sanjur 2-738-644

Alexander Santana 8-933-1167

Stephanie Tejeira 8-924-2239

Víctor Valenzuela 8-931-2246

Ana Villarreal 8-917-2049

**Profesor:**

**Ariel Vernaza**

**Fecha:**

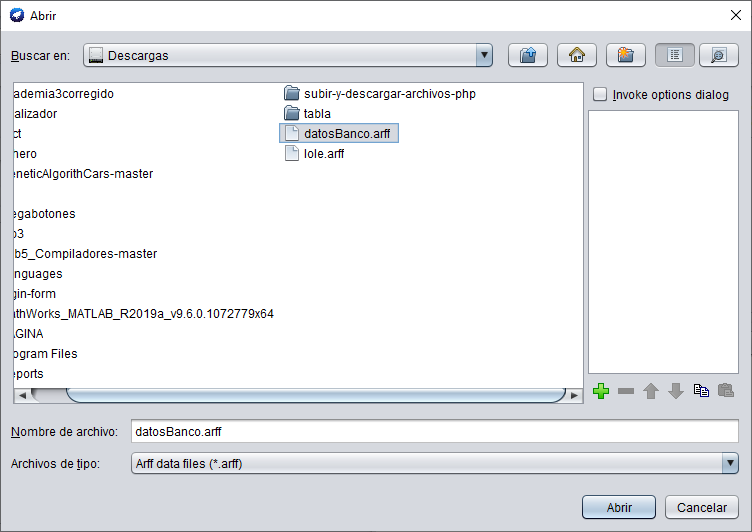
**25 de julio del 2020**

**Introducción**

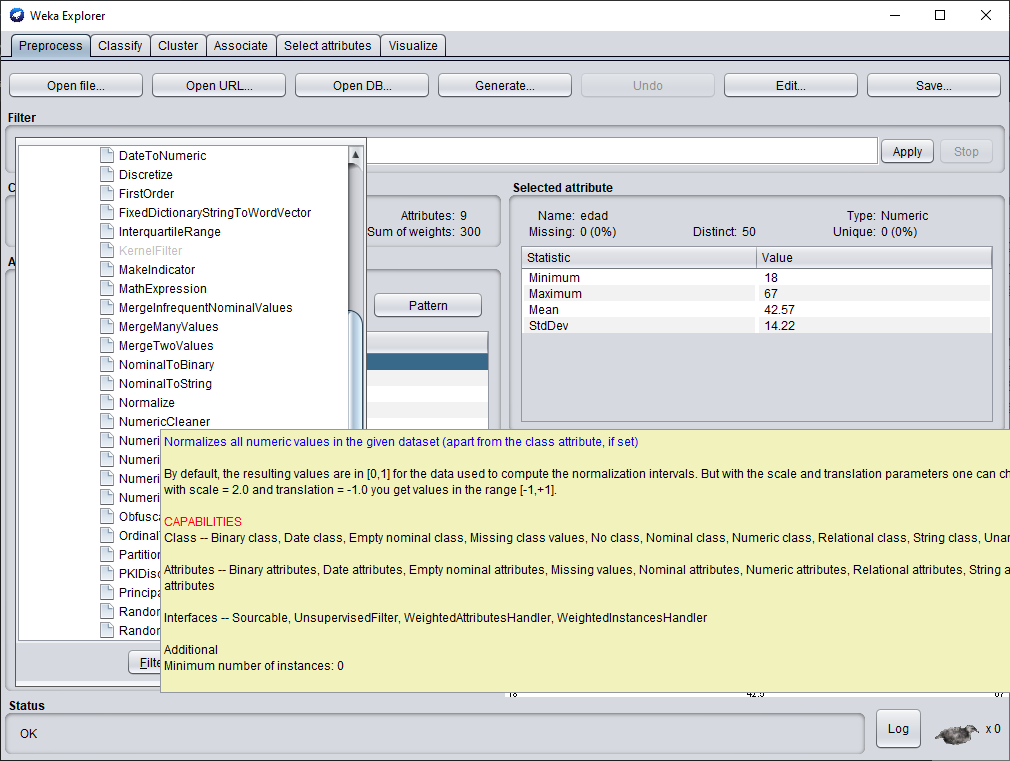
En este laboratorio trabajaremos con un archivo que posee datos de un banco, para determinar únicamente a partir de los datos si un nuevo cliente puede ser considerado como “bueno” para el banco. Así, se pretende obtener una RNA (Red Neur) que pueda ayudarnos a determinar si un cliente será “bueno” o “malo” para el banco y poder orientar la publicidad a los clientes “buenos”.

Para ello, a partir de todas las opciones de análisis y clasificación utilizando diferentes algoritmos, se deberá realizar un estudio detallado en el que se explique de forma amplia y clara (utilizando gráficos, reduciendo datos) cómo podemos utilizar diferentes algoritmos de clasificación en este problema en concreto.

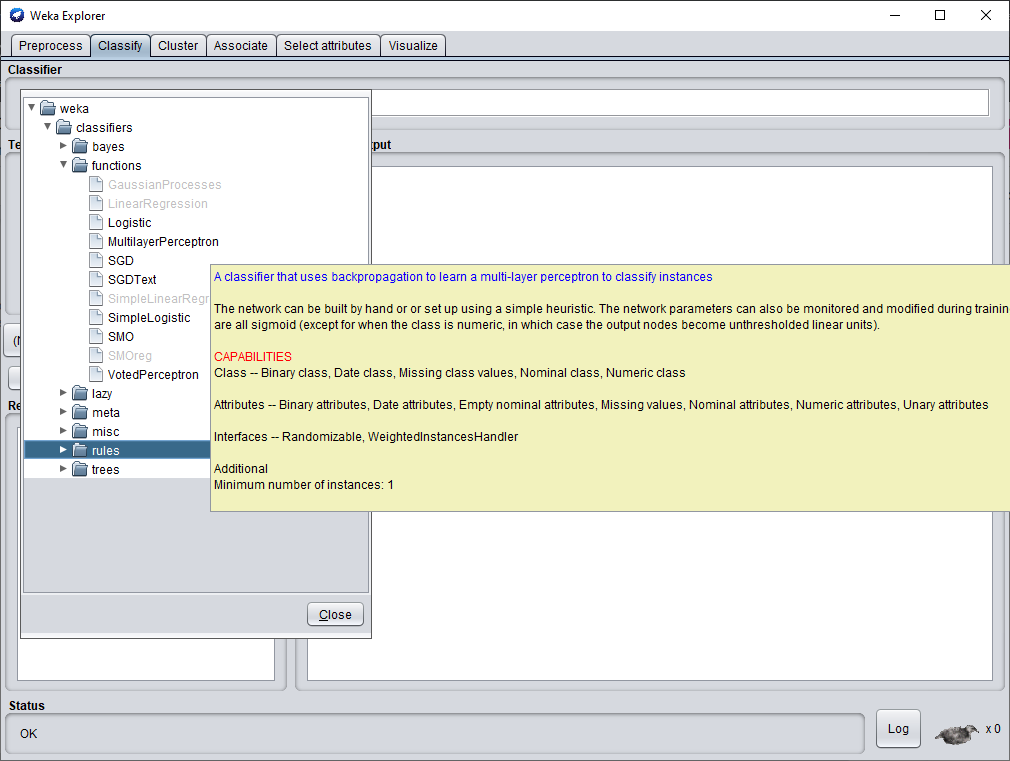
Utilizando el Simulador Weka



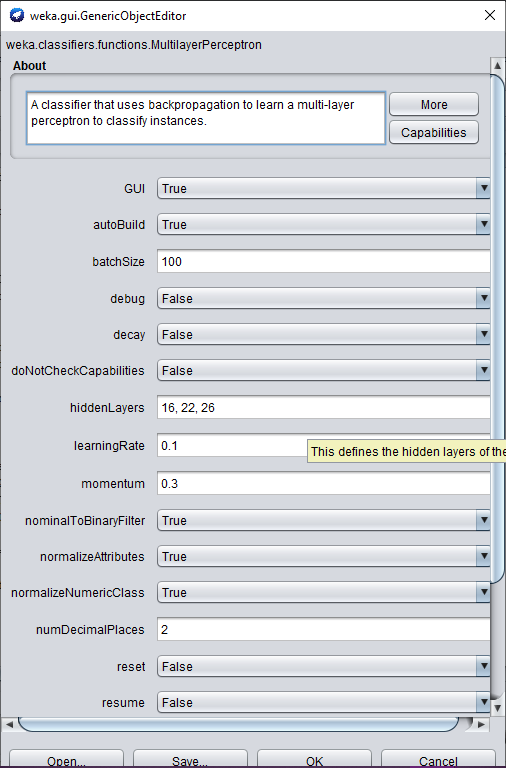
Subimos el archivo a weka para poder ir limpiando la data proporcionada.



Utilizamos la Función Normalize para limpiar la Data.

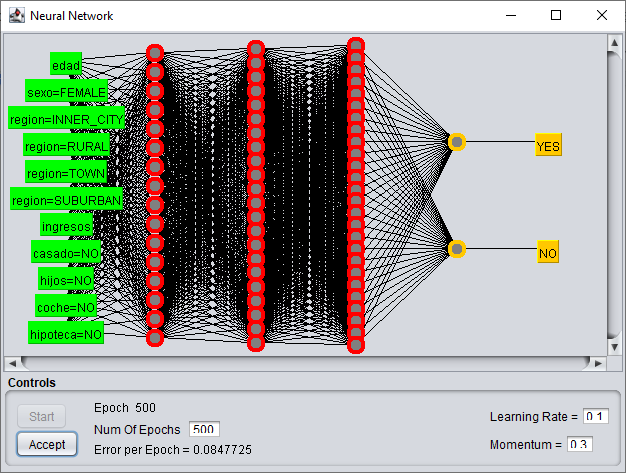


Clasificamos la data utilizando la función Multiplayer Perceptron.



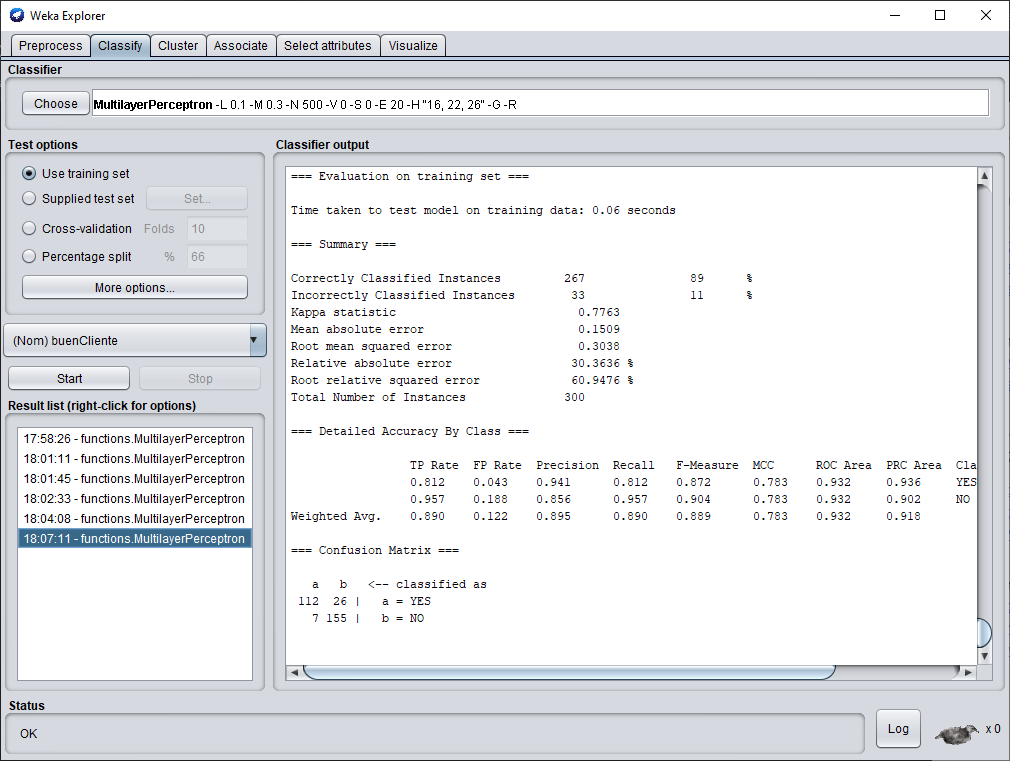
Se aplica los HiddenLayers para entrenar la data

En la siguiente imagen se muestran las capas de la red neuronal que estamos trabajando



Aquí vemos la época que son 500 en la que se entrena la red neuronal con un aprendizaje de 0.1 donde su impulso es de 0.3 y el error por época es de 0.0847725.

Esta es una red neuronal con 16 capa de entrada, 22 capa oculta y 26 capa de salida.



En esta imagen se puede ver la clasificación de la data donde se muestra lo siguiente:

**Instancias del conjunto de entrenamiento correctamente clasificadas.**

**Porcentaje de instancias correctamente clasificadas.**

**Falsos positivos y negativos con la matriz de confusión**

**Bibliografía**

* https://stackoverrun.com/es/q/10334347
* https://jakevdp.github.io/PythonDataScienceHandbook/05.02-introducing-scikit-learn.html
* https://www.researchgate.net/publication/292026781\_Neural\_networks\_learning\_improvement\_using\_the\_K-means\_clustering\_algorithm\_to\_detect\_network\_intrusions
* https://machinelearningmastery.com/clustering-algorithms-with-python/
* https://www.youtube.com/watch?v=w2wzVg0owxU
* https://youtu.be/s6PSSzeUMFk
* https://www.youtube.com/watch?v=mmB9RroFv8c
* https://youtu.be/EZOab1vkFmI
* https://github.com/Xpikuos/Inteligencia-Artificial/blob/master/K-Means/K-Means.ipynb
* https://youtu.be/PJbnP5xTeQc
* <https://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/>
* https://www.interactivechaos.com/manual/tutorial-de-machine-learning/perceptron-multicapa
* https://www.geeksforgeeks.org/confusion-matrix-machine-learning/
* https://colab.research.google.com/drive/1a07pzNtKLmoJhdvEp13KsTbQrtgYerok?usp=sharing
* https://www.youtube.com/watch?v=ikt0sny\_ImY
* https://youtu.be/PJbnP5xTeQc
* https://www.youtube.com/watch?v=PJbnP5xTeQc&feature=youtu.be
* https://www.youtube.com/watch?v=mmB9RroFv8c
* https://youtu.be/s6PSSzeUMFk