### **https://lh6.googleusercontent.com/xrUWl_Cfqys5q10qXK6AJDBmDXUjGzi54G2GaZpuhnmqNGBZbn-eTJJDbDXs55PozMe0peK_jTO_Z7n5jBC3y0f6Wbvjf49mB2YCEjvkqV5l5E-a3c1m-VNvkNPEc31QprBDxvkU**

**Proyecto Final Sistemas Inteligentes Computacionales**

**Selección de variables para redes neuronales mlp**

**John Eder Posada Zapata - Jorge Luis Ordoñez Ospina - Hans Rivera Londoño**

**Universidad Nacional de Colombia**

**Manizales - Caldas**

**2020**

# Índice

# 

[**Índice**](#_2u32wqns3sck) **1**

[**Problemática**](#_rz8uxv9iosmw) **2**

[**Solución propuesta**](#_qtkt3c1ac2f2) **2**

[**Diseño y funcionamiento**](#_xu9k0ou4p9k2) **2**

[Preselec.py](#_xn2znzrlhcu) 2

[AgProyecto.py](#_nqesthlrdy08) 2

[nn.py](#_g53zsydoaq65) 3

[main.py](#_gcp9ivu7mern) 4

[**Pruebas**](#_iqizjg47gyip) **5**

[**Requerimientos**](#_vbaocaxsuw12) **10**

[**Uso**](#_shaq1b2jctvf) **11**

[**Restricciones**](#_1u9vwqor7xgf) **11**

[**Bibliografía**](#_wu7n8v4l2em) **12**

# 

# Problemática

De acuerdo a diferentes análisis realizados en redes neuronales se ha determinado que en ocasiones es difícil seleccionar las variables más importantes que deben ingresar a la red neuronal para obtener un mejor resultado en su proceso de entrenamiento para la clasificación de clases.

# Solución propuesta

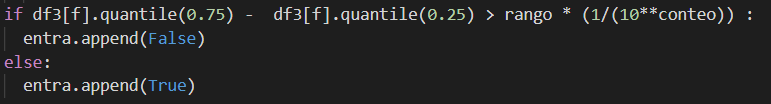
Se ha creado este proyecto el cual permite aplicar algoritmos genéticos para la clasificación de variables que serán utilizadas para un mejor entrenamiento de una red neuronal y además esta red permite ser aplicada a diferentes problemas donde la solución es una clasificación binaria; adicionalmente este proyecto está desarrollado de tal manera que permite la parametrización de las diferentes características tanto del algoritmo genético como de la red neuronal.

# Diseño y funcionamiento

A través de un esquema de varios archivos se mostrará el funcionamiento de este proyecto explicando cada uno por separado de la siguiente manera:

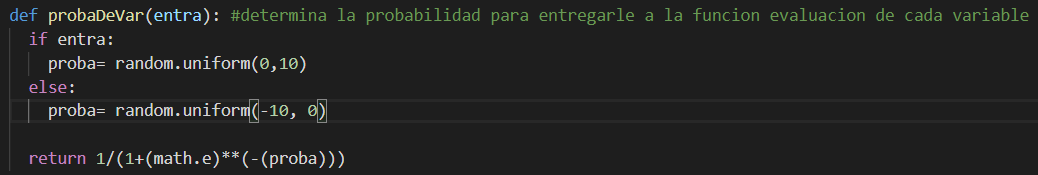
## Preselec.py

En este archivo lo que hace es generar un valor booleano para cada variable que será almacenado dentro de una lista que posteriormente se usará para generar una probabilidad cuando sea ingresado al algoritmo genético, teniendo en cuenta el nivel de variación de cada una de las columnas del data frame. Esto se realiza tomando la resta entre sus cuartiles 1 y 3 (percentil 25 y 75) y comparándolo con el rango multiplicado por 10^ (- dígitos de la mediana de la variable), y con base a esto si los datos tienen una gran variación se les retornará un valor de False, puesto que, se le consideran datos dispersos y poco significativos. Para las columnas que esto no sea así se retornará True ya que se consideran variables importantes para el siguiente paso en el algoritmo genético.

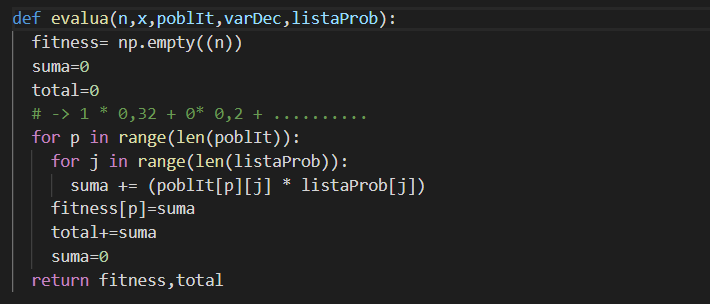


## AgProyecto.py

En este archivo lo primero que se realiza es utilizar el vector de valores booleanos generado en el archivo preselec.py y generar una probabilidad entre un rango 0 y 1, utilizando la función sigmoidea, para cada columna dependiendo del valor booleano generado, con base a esto se generará una probabilidad entre [0, 0.5) si su valor es False y (0.5, 1] si es True.



Luego de ser creada esta lista de porcentaje será enviado a la función evalúa que la utilizará para calcular el fitness que se utilizará para determinar cuál será nuestro mejor individuo que en este caso no dirá cuáles son las mejores columnas de nuestro data frame para utilizarse después en la red neuronal.

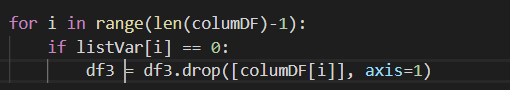


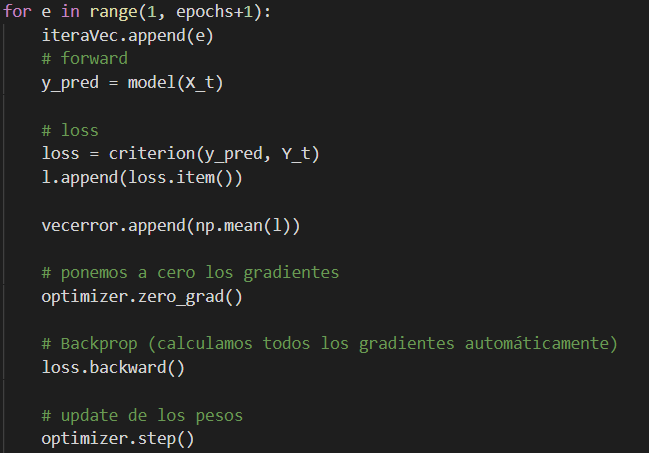
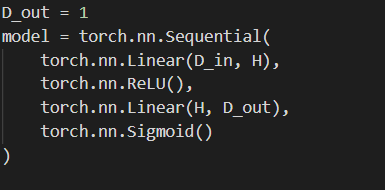
Ejemplo de individuo.



## nn.py

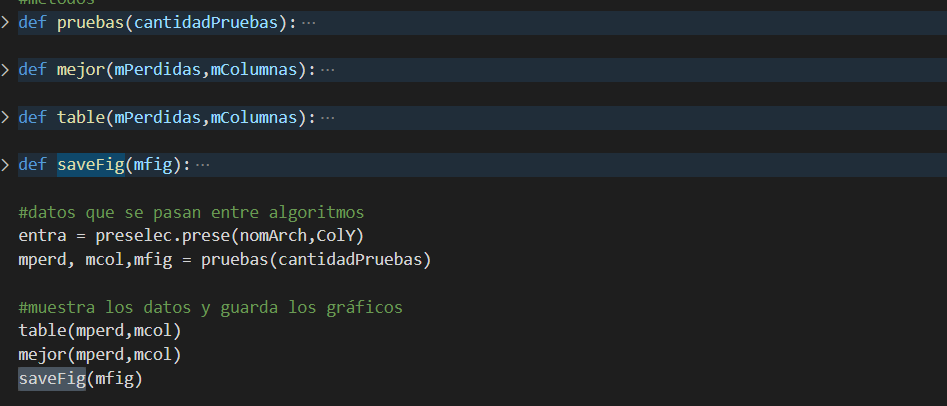
En este archivo se utiliza la lista entregada por el algoritmo genético que determinó cuáles serán las columnas que se usarán en el entrenamiento. A través del framework pytorch se realizará la creación de la red neuronal para una clasificación binaria que utilizara las columnas anteriormente seleccionadas.





## main.py

En este archivo se realiza la unión de los archivos anteriormente mencionados, y poder obtener una serie de resultados que serán mostrados por consola y a través de las gráficas que poseen el No. de iteración (época) y el error cuadrático medio. Las gráficas serán guardadas en un archivo pdf que se localizará dentro de la carpeta raíz con el nombre de graf.pdf.

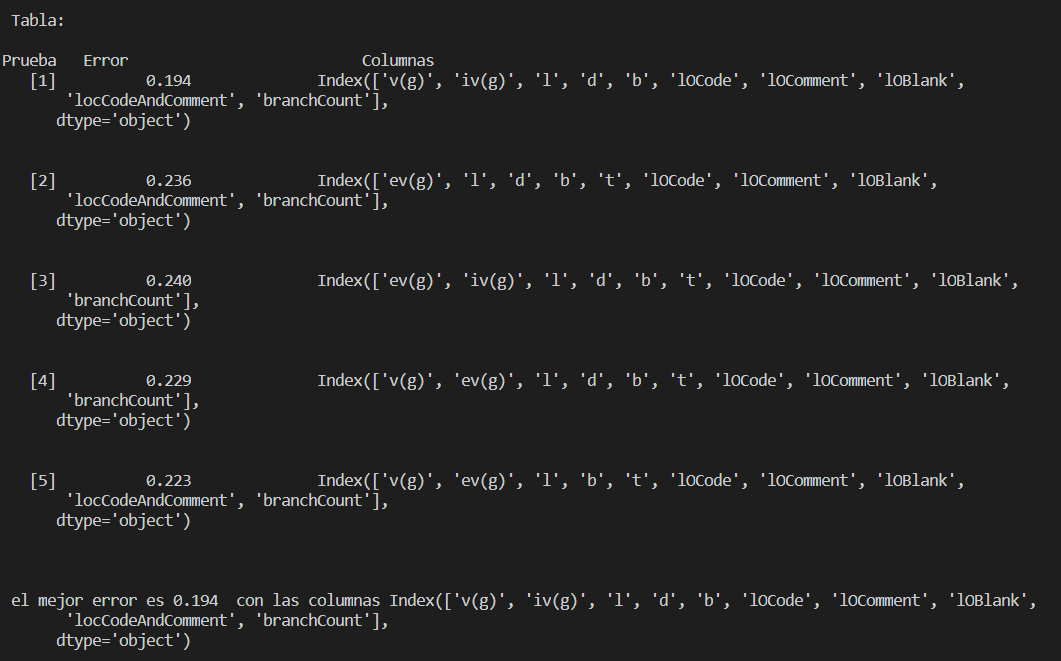


# Pruebas

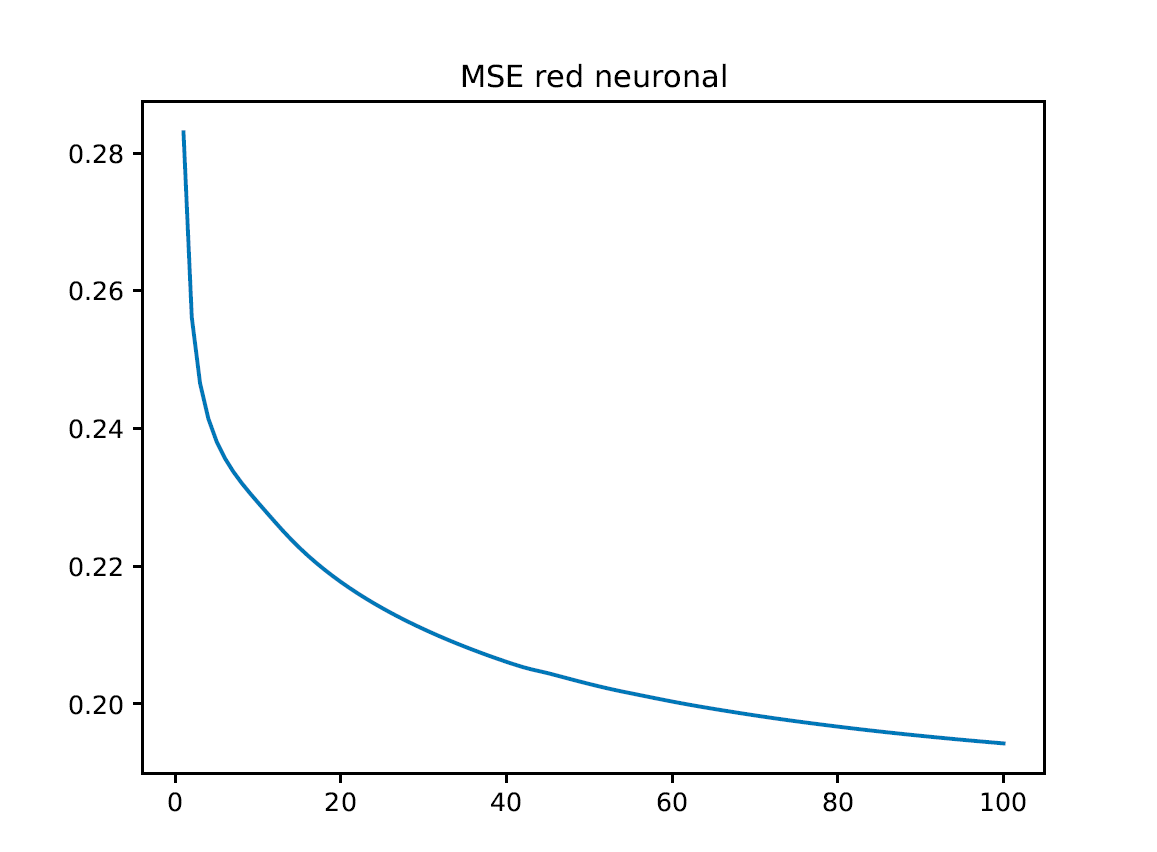
Utilizando el data frame Software Defect Prediction (<https://www.kaggle.com/semustafacevik/software-defect-prediction?select=about+JM1+Dataset.txt>). Se busca realizar la clasificación entre un software que tiene defectos y otro que no lo tiene.

En la siguiente imagen se muestra las columnas del data frame usado, con el nombre de todas sus columnas.

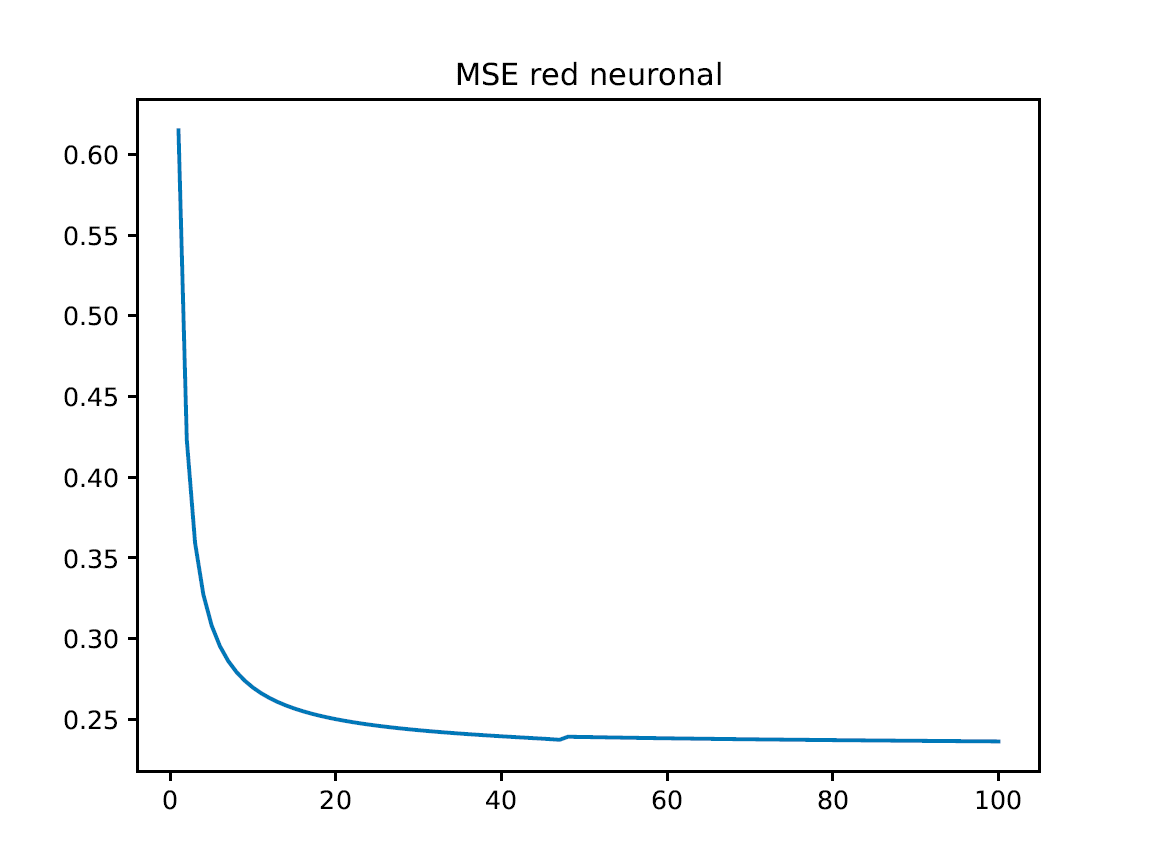
Luego de usar nuestras soluciones se han encontrado los siguientes resultados



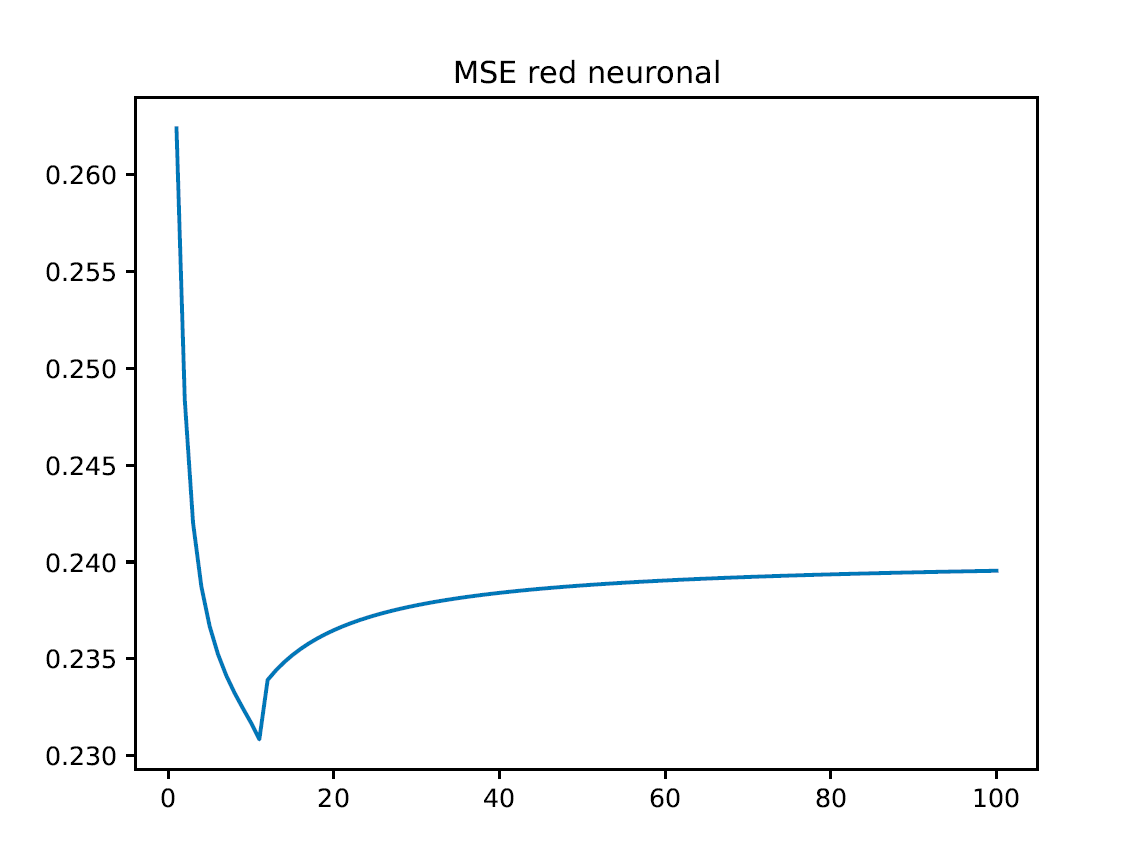
**Iteración 1**



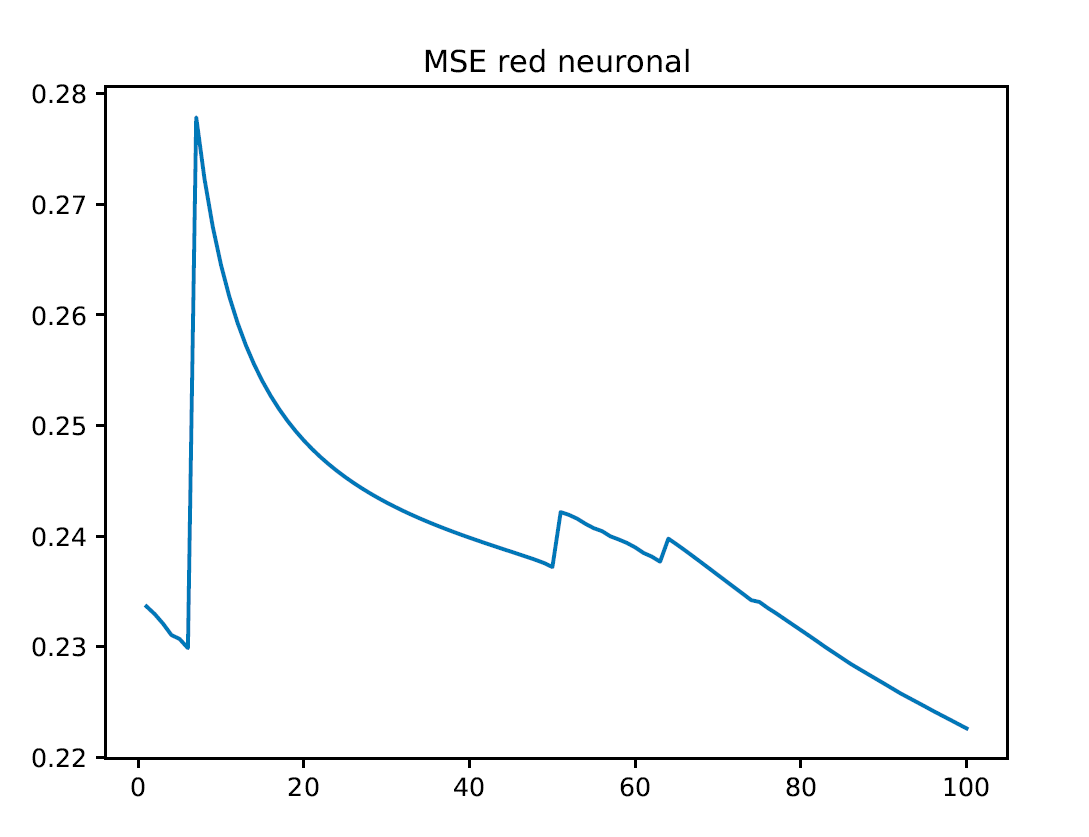
**Iteración 2**



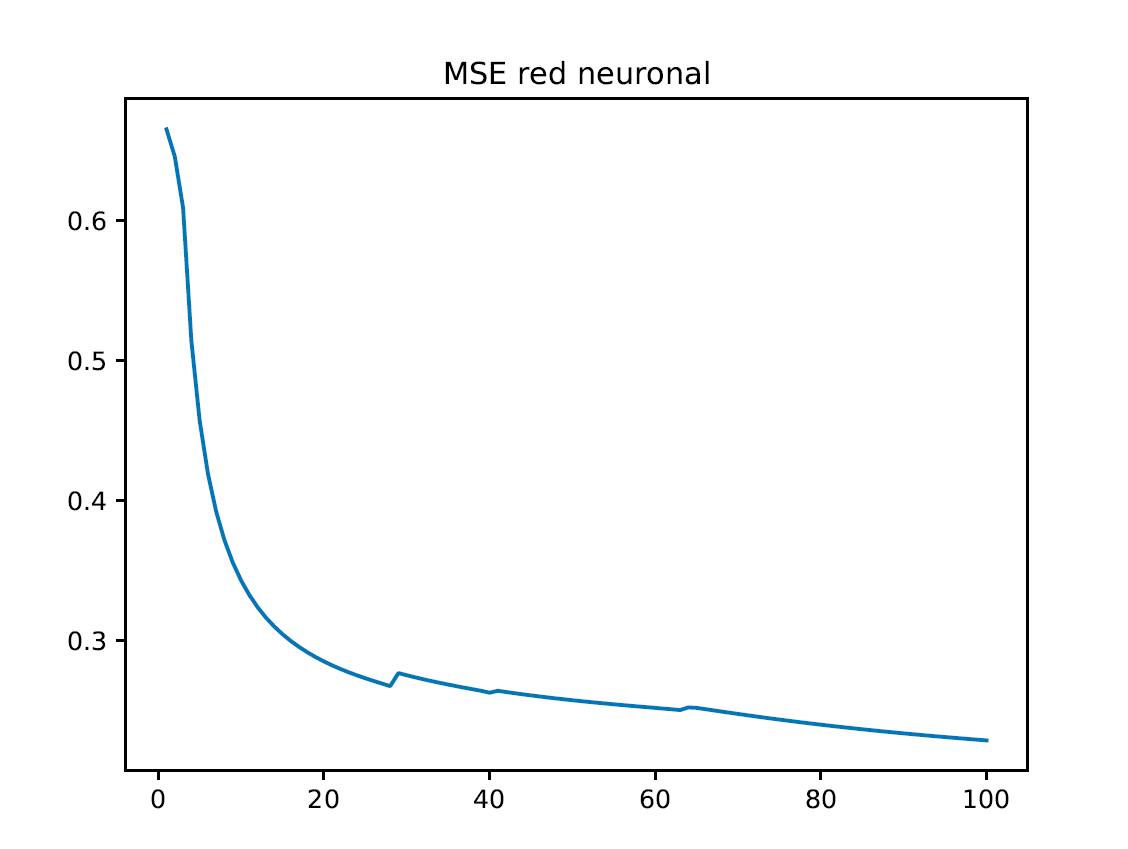
**Iteración 3**



**Iteración 4**



**Iteración 5**



# Requerimientos

1. Poseer Python 3.7 o superior.
2. Se necesitan las siguientes librerías para el funcionamiento correcto del proyecto:

* pytorch=1.7.0
* scikit-learn=0.23.2
* scipy=1.5.2
* pandas=1.1.1
* numpy=1.19.1
* numpy-base=1.19.1
* matplotlib=3.2.2
* matplotlib-base=3.2.2

Para instalar dichas librerías, se requiere ejecutar el siguiente comando:

py -m pip install -r requirements.txt , adicionalmente ejecutar

pip install torch==1.7.0+cpu torchvision==0.8.1+cpu torchaudio===0.7.0 -f https://download.pytorch.org/whl/torch\_stable.html

# Uso

Luego de cumplir con los requerimientos, se deje ejecutar el archivo main.py a través de la línea de comandos con la instrucción siguiente: Python main.py

Si desea cambiar algún parámetro, esto lo encontrará en el archivo main.py, allí encontrará una descripción de cada parámetro que es posible cambiar.

# Restricciones

Algunas restricciones que presenta son:

* Todas sus columnas deben de ser numéricas.
* El data frame no puede ser clustering, ya que nuestra red neuronal hace uso de la columna de Y real que son los valores que tienen en la vida real el sistema que se está modelando.

# Bibliografía

* PEDRO, J. (2020). El Perceptrón. Retrieved 29 November 2020, from <https://sensioai.com/blog/012_perceptron1>
* PEDRO, J. (2020). El Perceptrón Multicapa - Introducción. Retrieved 29 November 2020, from <https://sensioai.com/blog/023_mlp_backprop>
* Automatiza decisiones con Inteligencia Artificial en Excel (instructivo para novatos. Parte 1). (2020). Retrieved 29 November 2020, from <https://www.rankia.co/blog/comstar/3618510-automatiza-decisiones-inteligencia-artificial-excel-instructivo-para-novatos-parte-1>
* ¿Cómo seleccionar las variables adecuadas para tu modelo? | Máxima Formación. (2020). Retrieved 29 November 2020, from <https://www.maximaformacion.es/blog-dat/como-seleccionar-las-variables-adecuadas-para-tu-modelo/>
* Best subset selection — STATS 202. (2020). Retrieved 29 November 2020, from <https://web.stanford.edu/class/stats202/notes/Model-selection/Best-subset.html>
* Ana González, Vidal. (2020). Retrieved 29 November 2020, from <http://eio.usc.es/pub/mte/descargas/ProyectosFinMaster/Proyecto_1263.pdf>
* (2020). Retrieved 29 November 2020, from <https://www.datos.gov.co/browse?category=Ciencia%2C+Tecnolog%C3%ADa+e+Innovaci%C3%B3n&sortBy=newest>
* DIFERENCIA ENTRE CLASIFICACIÓN Y REGRESIÓN EN MACHINE LEARNING. (2020). Retrieved 29 November 2020, from <https://ingenierobeta.com/clasificacion-vs-regresion-machine-learning/#:~:text=La%20regresi%C3%B3n%20tiene%20el%20objetivo,clasificaci%C3%B3n%20los%20valores%20son%20discretos>
* Heras, J. (2020). Regularización Lasso L1, Ridge L2 y ElasticNet - IArtificial.net. Retrieved 29 November 2020, from <https://www.iartificial.net/regularizacion-lasso-l1-ridge-l2-y-elasticnet/#Regularizacion_Lasso_(L1)>
* Corral Sastre, Antonio, (2020). Retrieved 29 November 2020, from <https://riunet.upv.es/bitstream/handle/10251/111512/Corral%20-%20Desarrollo%20y%20Evaluaci%C3%B3n%20de%20Algoritmos%20Gen%C3%A9ticos%20Multiobjetivo.%20Aplicaci%C3%B3n%20al%20problema%20del%20viajante..pdf?sequence=1&isAllowed=y>
* Damián Jorge, Matich, (2020). Retrieved 29 November 2020, from <https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf>
* Tutorial básico de redes neurales con Python - VIDEO — gidahatari. (2020). Retrieved 29 November 2020, from <https://gidahatari.com/ih-es/tutorial-basico-de-redes-neurales-con-python-video>
* python-engineer/pytorchTutorial. (2020). Retrieved 29 November 2020, from <https://github.com/python-engineer/pytorchTutorial/blob/master/07_linear_regression.py>
* Sequential — PyTorch 1.7.0 documentation. (2020). Retrieved 29 November 2020, from <https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html>
* Deep Learning with PyTorch. (2020). Retrieved 29 November 2020, from <https://books.google.com.co/books?hl=es&lr=&id=DOlODwAAQBAJ&oi=fnd&pg=PP1&dq=using+sigmoid+for+neural+networks+in+pytorch&ots=koY_e2aDMj&sig=dBuY-tr2i0eqt9zRx8TPKbTaXv4#v=onepage&q=using%20sigmoid%20for%20neural%20networks%20in%20pytorch&f=false>
* Studies!), G. (2020). Pytorch Introduction | How to Build A Neural Network. Retrieved 29 November 2020, from <https://www.analyticsvidhya.com/blog/2019/01/guide-pytorch-neural-networks-case-studies/>
* Análisis de clasificación binaria | Interactive Chaos. (2020). Retrieved 29 November 2020, from <https://www.interactivechaos.com/manual/tutorial-de-deep-learning/analisis-de-clasificacion-binaria>
* (2020). Retrieved 29 November 2020, from <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/#:~:text=La%20funci%C3%B3n%20de%20activaci%C3%B3n%20se,con%20ello%20el%20coste%20computacional>
* torch. argmax — PyTorch 1.7.0 documentation. (2020). Retrieved 29 November 2020, from <https://pytorch.org/docs/stable/generated/torch.argmax.html>
* Multipage PDF — Matplotlib 3.1.0 documentation. (2020). Retrieved 29 November 2020, from <https://matplotlib.org/3.1.0/gallery/misc/multipage_pdf.html>
* Joshi, S. (2020). Cómo guardar parcelas como archivo PDF en Matplotlib. Retrieved 29 November 2020, from <https://www.delftstack.com/es/howto/matplotlib/how-to-save-plots-as-pdf-file-in-matplotlib/#:~:text=Los%20gr%C3%A1ficos%20generados%20a%20partir,PDF%2C%20utilizamos%20la%20clase%20PdfPages%20>
* Software Defect Prediction. (2020). Retrieved 29 November 2020, from <https://www.kaggle.com/semustafacevik/software-defect-prediction?select=about+JM1+Dataset.txt>