Módulo 2 - Práctica 4

[**Descubrimientos**](#_v7uzum5ph0fc) **2**

[Variables Discretas](#_jtrgkqklkrju) 2

[Variables Continuas](#_fwmzksqtkk8e) 2

[**Árbol de Decisión**](#_njlc9i8pbc9m) **4**

[**Hiper Parámetros**](#_rvj88utu5pkl) **5**

[**Red Neuronal**](#_6f7dqa5payqb) **6**

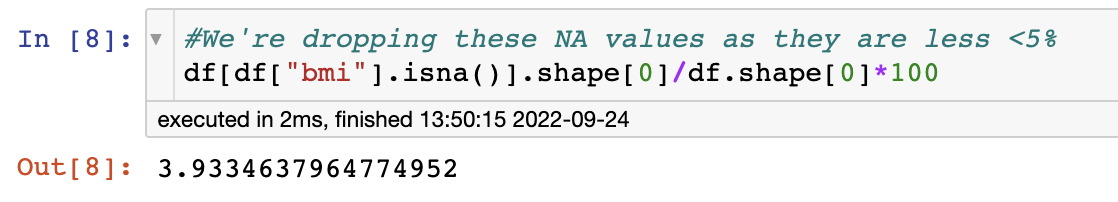
[Hiper Parámetros](#_zi88kaj05t8y) 6

[**Conclusiones**](#_myie25wuyjlx) **7**

# Descubrimientos

Al realizar esta práctica, se nos pidió hacer un modelo de clasificación para determinar si una persona iba a tener un “derrame cerebral” dadas ciertas características de Salud.

Para poder realizar esto, tenemos que ver que nuestros datos estén de buena manera. A lo que nos dimos cuenta que el “bmi” tenía valores nulos, los cales representaban menos del 5% del total, a lo que procedimos a borrarlos.



Fuera de eso, nuestro dataset no tenía grandes problemas con sus datos.

## Variables Discretas

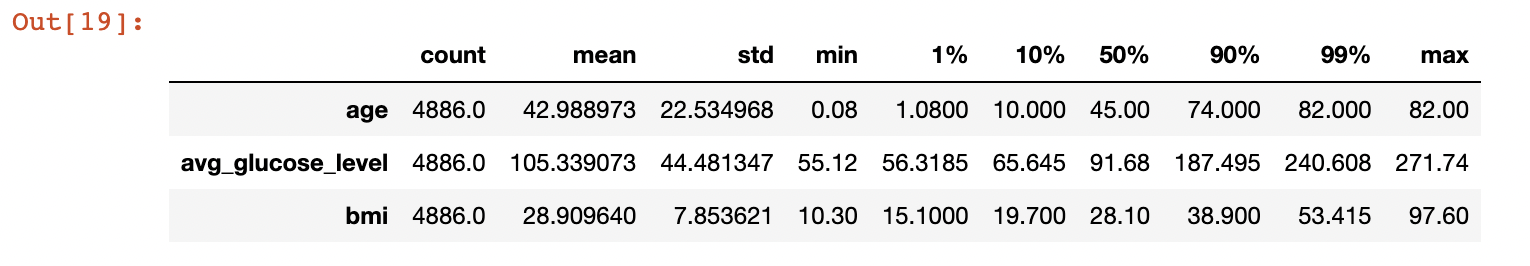
Para la Exploración y procesamiento de variables discretas, nos dimos cuenta de dos puntos:

* En el género del usuario, tenemos un 0.02% de datos que tienen un género de otros, los eliminamos.
* Tenemos un 0.4% de los datos que en la variable “work\_type” los usuarios nunca trabajaron. Eliminamos esas coincidencias

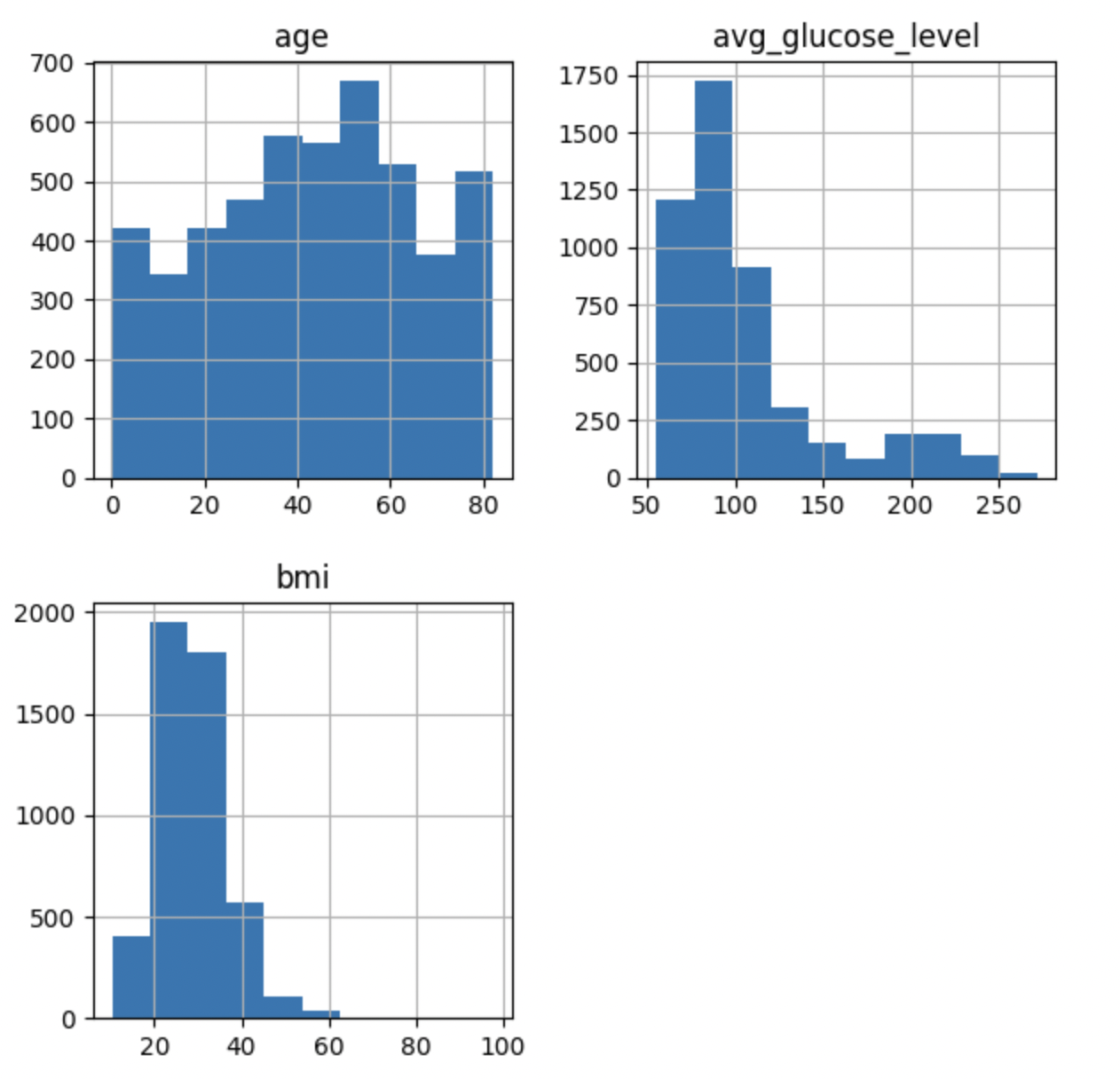
Para el resto de nuestras variables categóricas, procedimos a crear variables dummy, las cuales nos ayudarían de una manera sencilla a hacer nuestro modelo.

## Variables Continuas

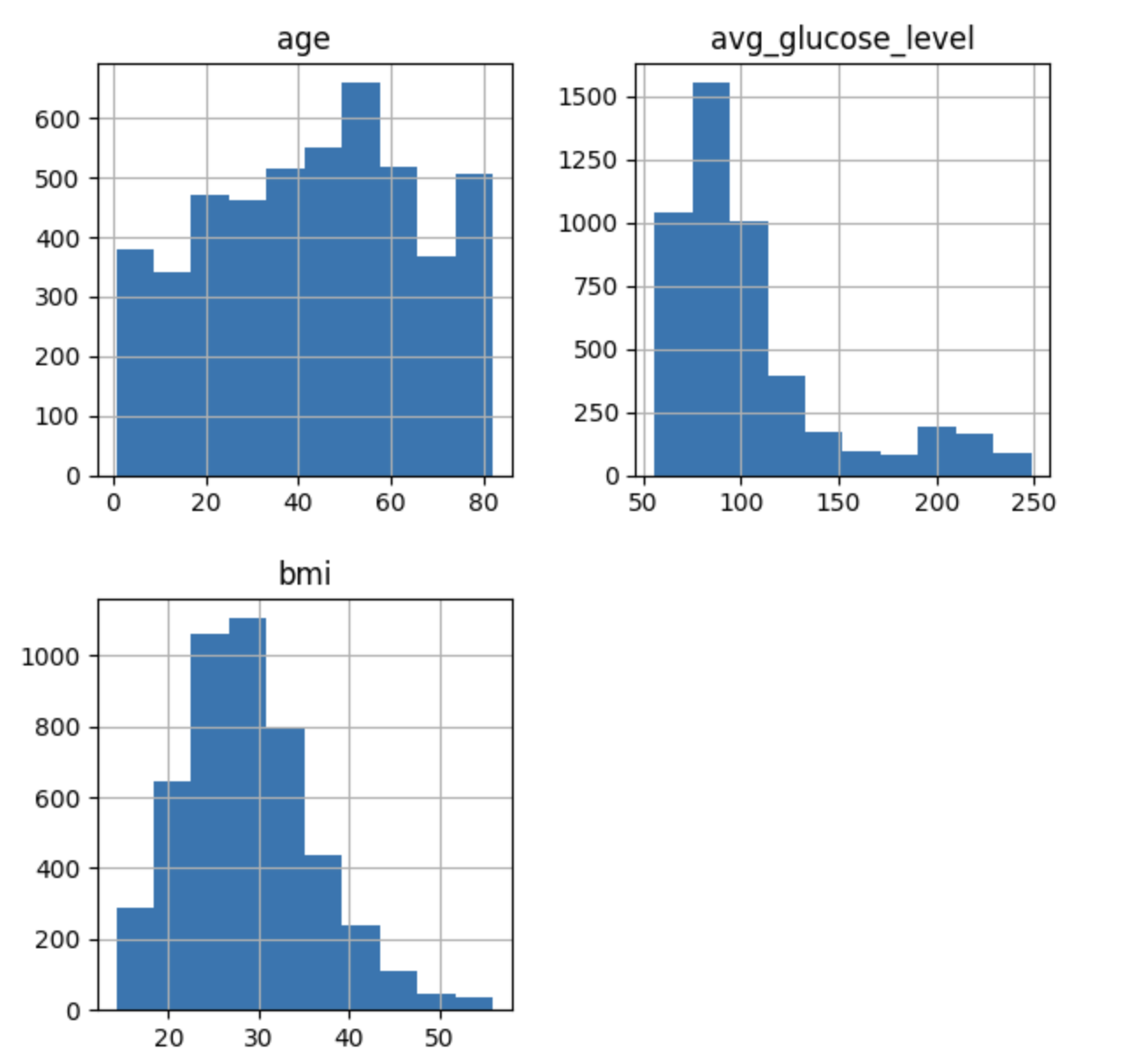
Para las variables continuas, hicimos un poco más de tratamiento. Al hacer el análisis de percentiles, nos dimos cuenta que hay algo de diferencia entre el 99% percentil y el valor máximo del BMI.



Al analizar de una forma más gráfica, podemos darnos cuenta que, aunque las distribuciones no se ven mal, las podemos mejorar.



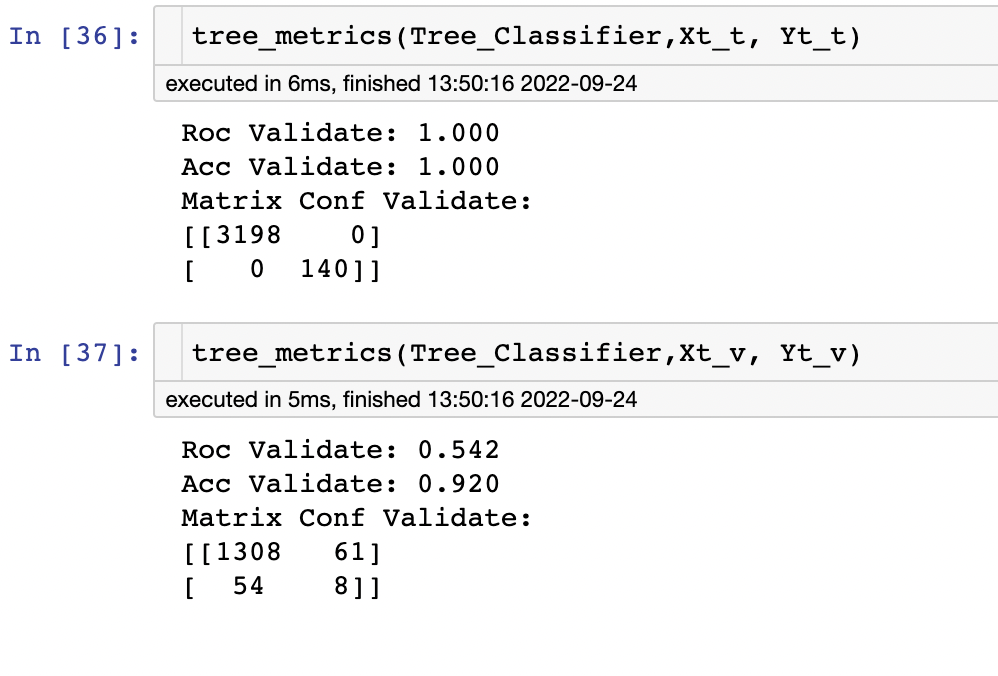
Para poder hacer esto, quitamos los outliers que tengamos dado la cerca percentil. Para este ejercicio, ocupamos 0.005 y el 0.995 percentil. Luego de quitar esos datos, obtenemos las siguientes distribuciones.



# Árbol de Decisión

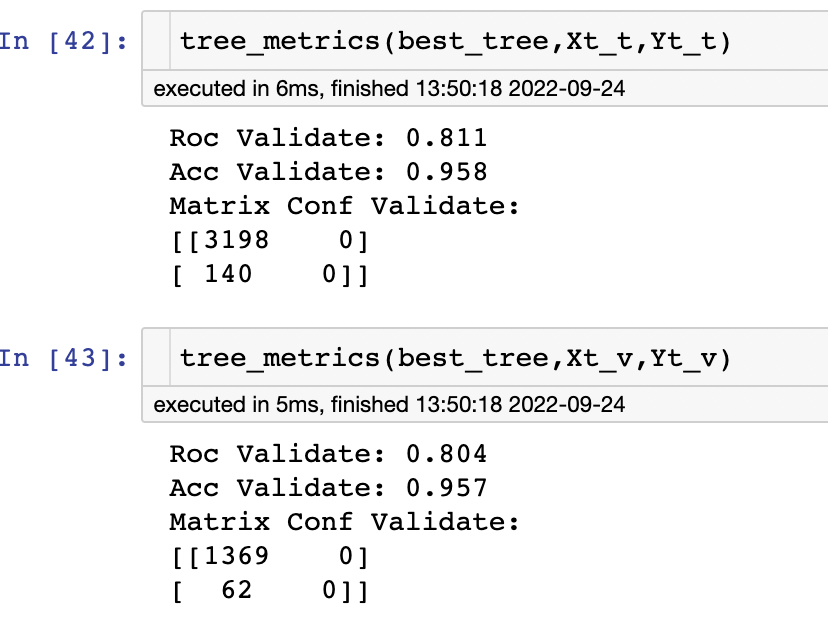
Para poder hacer el árbol de decisión, elegimos usar el *DecisionTreeClassifier* que se encuentra dentro de la librería Scikit-Learn. Luego escalamos nuestras variables continuas.

Procedimos a entrenar y luego ver las métricas de nuestro árbol de decisión, lo cual nos dimos cuenta que no geeralizaba. Procedemos a hacer un randomized Search



## Hiper Parámetros

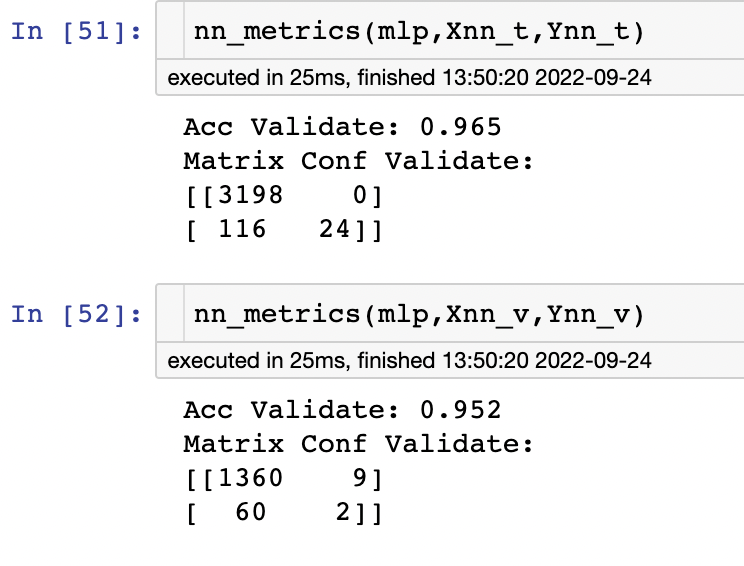
Como nos dimos cuenta que nuestro árbol inicial se sobre ajustaba, empezamos a usar una búsqueda con los mejores parámetros por medio de un Randomized Search. Una vez hecho esto, nuestro modelo sí mejoró, pero no parecía ser la mejor opción; ya que, teníamos falsos negativos considerables.



# Red Neuronal

Como nos dimos cuenta que nuestro árbol de decisión parecía ser una no tan buena solución, pasamos a hacer nuestra red neural. En este caso, usamos el Multilayer Perceptron Classifier (*MLPClassifier*) dentro de la paquetería de Scikit-learn.

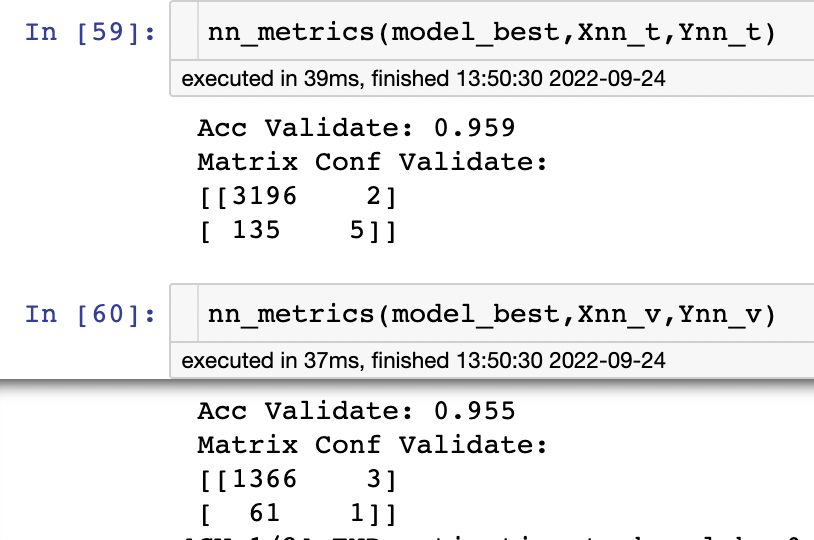
De igual manera que en nuestro árbol, para las redes neuronales escalamos nuestras variables continuas y procedimos a entrenar.



Con este modelo inicial, empezamos a tener mejores resultados que en el árbol, pero tenemos un número considerable de falsos positivos en nuestro set de validación, a lo cual hacemos nuestra búsqueda de hiper parámetros.

## Hiper Parámetros

Al hacer la búsqueda con *RandomizedSearch* con 1000 iteraciones, nos dimos cuenta que nuestro modelo no mejoraba considerablemente con respecto al modelo inicial. Pero nos quedamos con esta opción.



# Conclusiones

Nuestra mejor opción actual es usar una red neuronal dadas las métricas que se obtuvieron en los diferentes análisis. Para poder ir mejorando el modelo, en un futuro se podría:

* Crear más variables en el dataset.
* No borrar las respuestas que platicamos con anterioridad.