Devolución

1. Hola Iván, comencemos por el código de descarga del dataset, al tenerlo de esta manera, requiere que el usuario que quiera revisar el trabajo suba el archivo csv al colab:

# cargar los datos

diabetes\_dataset = pd.read\_csv('diabetes.csv')

data = diabetes\_dataset.values

Te recomiendo subir el archivo csv a su repositorio de github y obtener la ruta del archivo csv e implementar el siguiente código de descarga, adaptarlo a la ruta del archivo a usar:

if os.access('drug200.csv', os.F\_OK) is False:

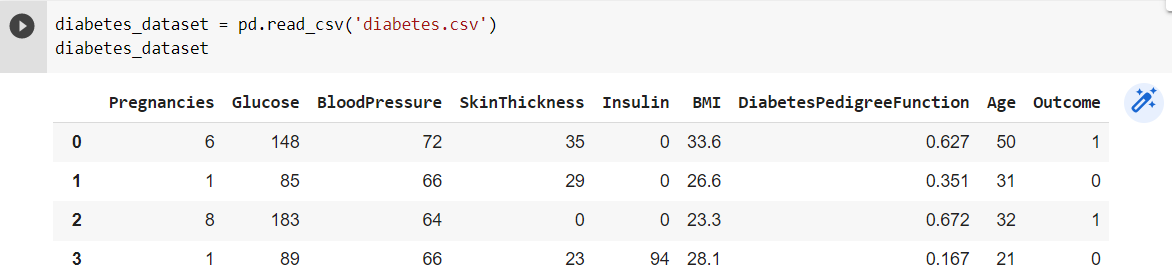
    if platform.system() == 'Windows':

        !curl https://raw.githubusercontent.com/InoveAlumnos/dataset\_analytics\_python/master/drug200.csv > drug200.csv

    else:

        !wget drug200.csv https://raw.githubusercontent.com/InoveAlumnos/dataset\_analytics\_python/master/drug200.csv

1. Al obtener la información, es preferible trabajarla en un dataframe para obtener tanto los valores como las columnas y poder visualizar la info. También, para usar algunos métodos de pandas.



1. Luego verificar si hay datos faltantes es muy importante, de esta manera solicita una descripción estadística básica y agrega dos filas más para contabilizar si hay datos faltantes.

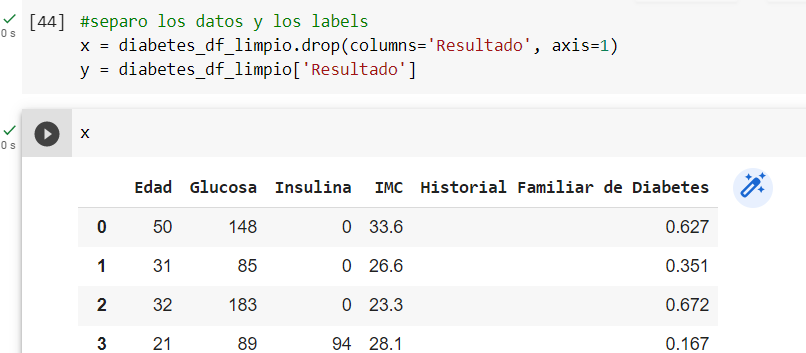
des = diabetes\_dataset.describe()

des.loc['Nan'] = diabetes\_dataset.isna().sum()

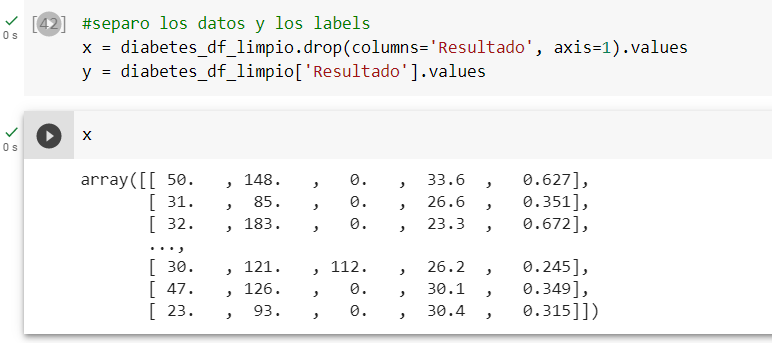
des.loc['%Nan'] = (diabetes\_dataset.isna().mean())\*100

des

1. Para la separación de valores para x e y, se necesita los valores, si se trabaja de esta manera, tanto x como y trae los valores como los nombres de las columnas, pero los algoritmos necesitan los valores solamente en una matriz, para ello ajusto esto:



A esto:



1. IMPORTANTE: Estamos ante un caso de clasificación donde las opciones de salida serian: diabético, no diabético
2. En este caso, los algoritmos a implementar serian: randomforest y knn(vecinos cercanos)