|  |  |  |
| --- | --- | --- |
| **#** | **MATERIA:** INTELIGENCIA ARTIFICIAL **SIGLA:** INF-354 | **T** |
| **BRIYAN JULIO TORREZ VARGAS** |

**PROYECTO INF-354**

**Link carpeta contenedora:** [**https://github.com/oblitus1010/Proyecto-INF-354-Briyan-Torrez.git**](https://github.com/oblitus1010/Proyecto-INF-354-Briyan-Torrez.git)

**Introducción**

Estamos viviendo una época de vulnerabilidad en el marco de la salud debido al COVID 19, a pesar de que ya estamos en una relativa etapa final, ya que la cura por las distintas potencias mundiales está en aparente auge. En un principio a comienzos del año 2020 comenzaron a verse brotes del virus en la población, fue desde ese momento que se fueron recopilando datos por día del Nro de infectados por departamento en Bolivia.

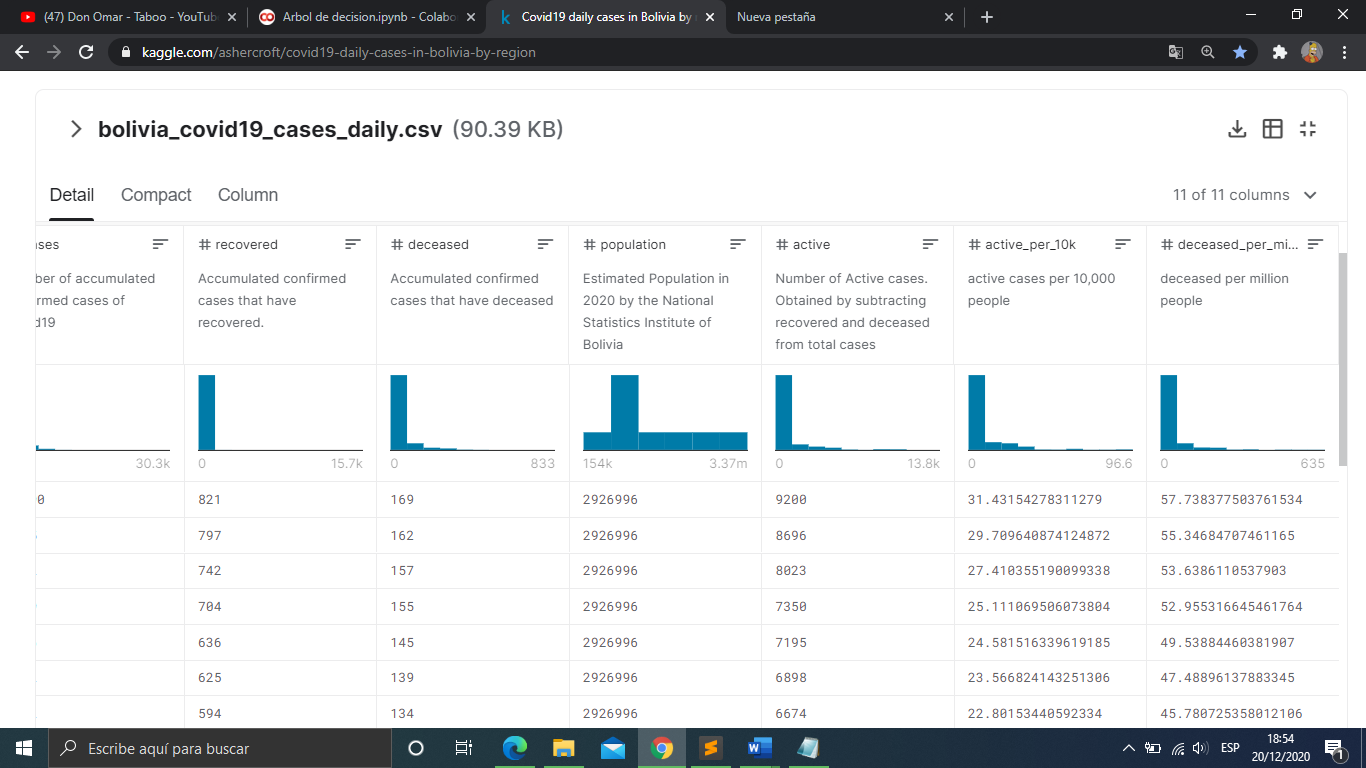
Bolivia tiene 9 departamentos: La paz, Cochabamba, Oruro, Santa Cruz, Beni, Pando, Sucre, Potosí, Tarija, siendo Sucre la capital del país.

El dataset que se obtuvo tiene que ver con el contexto mencionado, en esta se tiene registro por día del numero de infectados. Se tiene por columnas con datos diarios lo siguiente:

* **Descripción de las columnas del dataset**

1. Date (fecha): esta columna nos indica las diferentes fechas desde el inicio de la pandemia hasta el 20 de julio.
2. Región: Departamentos donde se presentaron casos.
3. Cases (casos): El número de casos confirmados existentes hasta la fecha.
4. Recovered (recuperados): El número de personas recuperadas al final del día.
5. Deceased (muertes): El numero de personas fallecidas al final del día.
6. Population (poblacion): Numero de personas en la región correspondiente.
7. Active (activos): El número de personas infectadas restantes.
8. active\_per\_10k (activos x 10k): Cantidad de pacientes activos hay por cada 10.000 personas.
9. deceased\_per\_million (muertes x 1 millon): El número de fallecidos hay por cada millón de personas.
10. Suceptible (suceptibles): El numero de personas que aún no se contagiaron.
11. Removed (removido): Cuantos casos descartados existen.

Para una mayor comodidad se cambió los nombres de las columnas a español para facilitar la comprensión del dataset.



**Planteamiento del problema**

La inteligencia artificial se encuentra en pleno auge siendo uno de los campos mas estudiados, pero a su vez es realmente complejo, por la gran cantidad de conceptos matemáticos que forman parte del conjunto del área. Inciar puede resultar difícil, si no se tiene una base fuerte en las matemáticas, pero eso no debe detener el interés por entender de este campo complejo pero interesante. Es por ello que surge la pregunta, ¿Cómo implementar una inteligencia artificial en Python y no morir en el intento?

**Objetivos**

**Objetivo General**

Entender y desarrollar una IA en Python para obtener una base con la cual se pueda profundizar acerca de conceptos fundamentales del tema.

**Objetivo Especifico**

* Entender el dataset e identificar los algoritmos de preprocesamiento adecuados.
* Identificar columnas dependientes e independientes para el análisis de los datos.
* Realizar al menos 3 modelos de IA.
* Utilizar Google colab.

**Preprocesamiento**

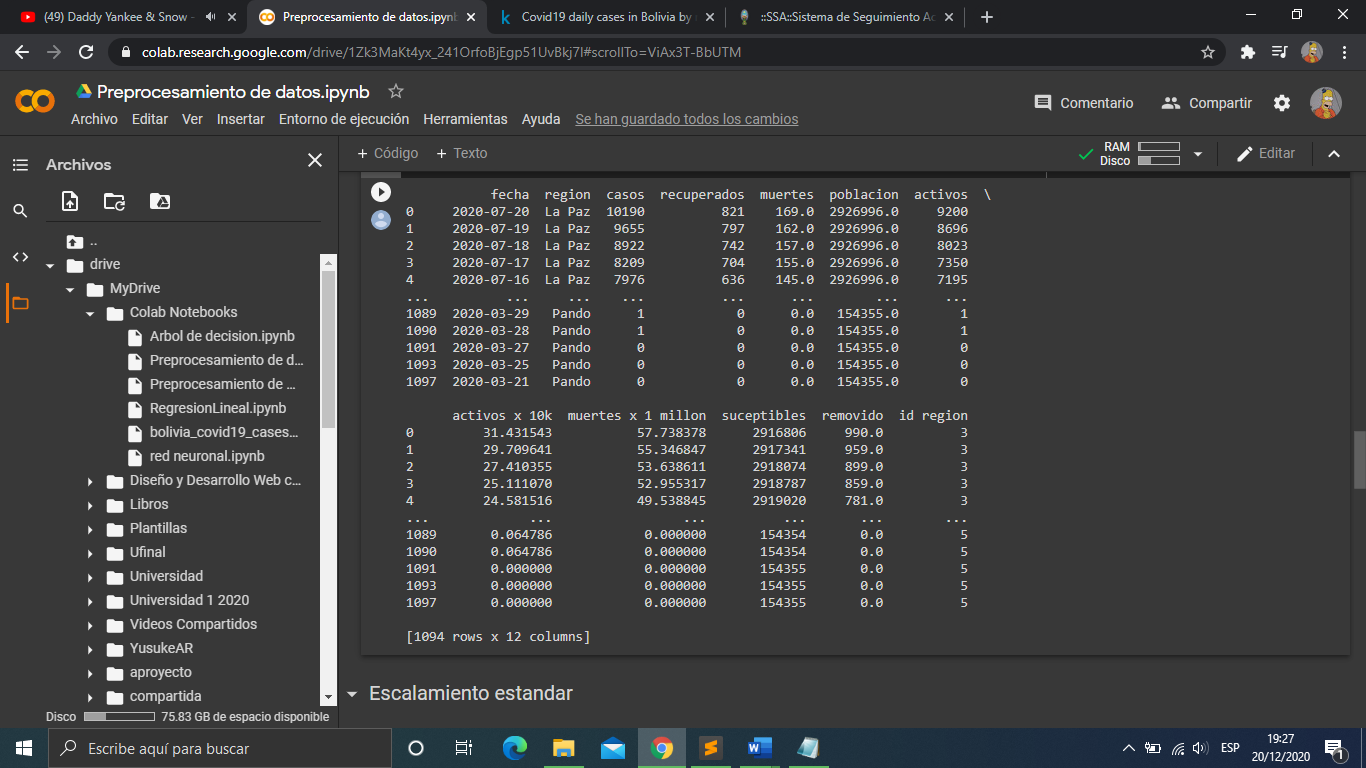
En primera instancia como se indica se eligió el dataset de <https://www.kaggle.com/ashercroft/covid19-daily-cases-in-bolivia-by-region> ya que tiene que ver con la situación del país en la actual coyuntura.

* **Eliminación de filas que tengan valores nulos (NaN)**

Esto podemos permitírnoslo porque el dataset es grande, pero no es recomendable en dataset con pocos datos, en esa en mejor usar la media de las columnas para reemplazar los campos vacíos.

* **Eliminación de filas duplicadas**
* **Preprocesamiento LabelEncoder**

Esto permite etiquetar una columna asignándole un numero según al grupo al que pertenezca.



* **Escalamiento estándar**

Estandariza los datos eliminando la media tal que su varianza sea 1. Ponemos todas nuestras características en la misma escala para que ninguna esté dominada por otra (Tiende a llevar la media a 0 y la desviación estándar a 1)

* **Normalización**

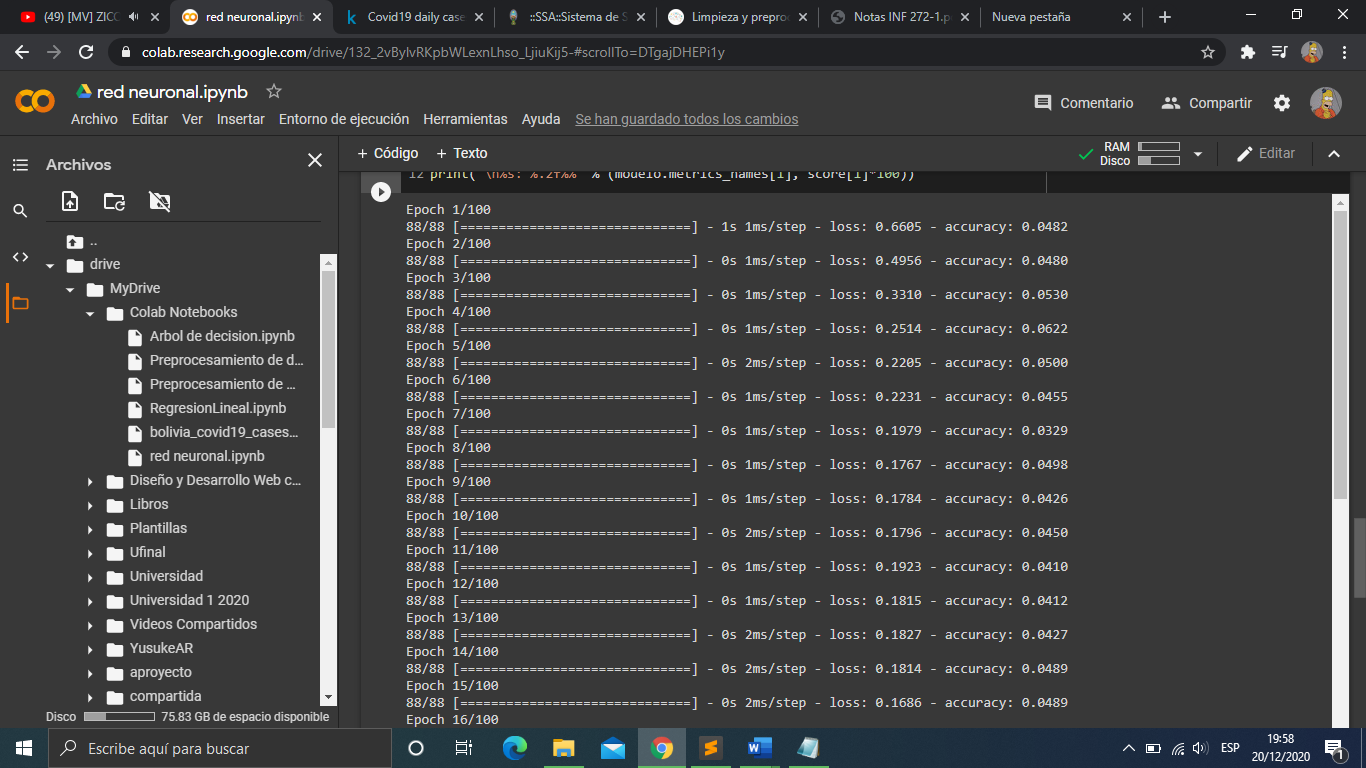
La normalización implica transformar o convertir el conjunto de datos en una distribución normal. Algunos algoritmos como Máquinas Vectores de Soporte convergen mucho más rápido en los datos normalizados, por lo que tiene sentido normalizar los datos para obtener mejores resultados.

**Modelos y Pruebas**

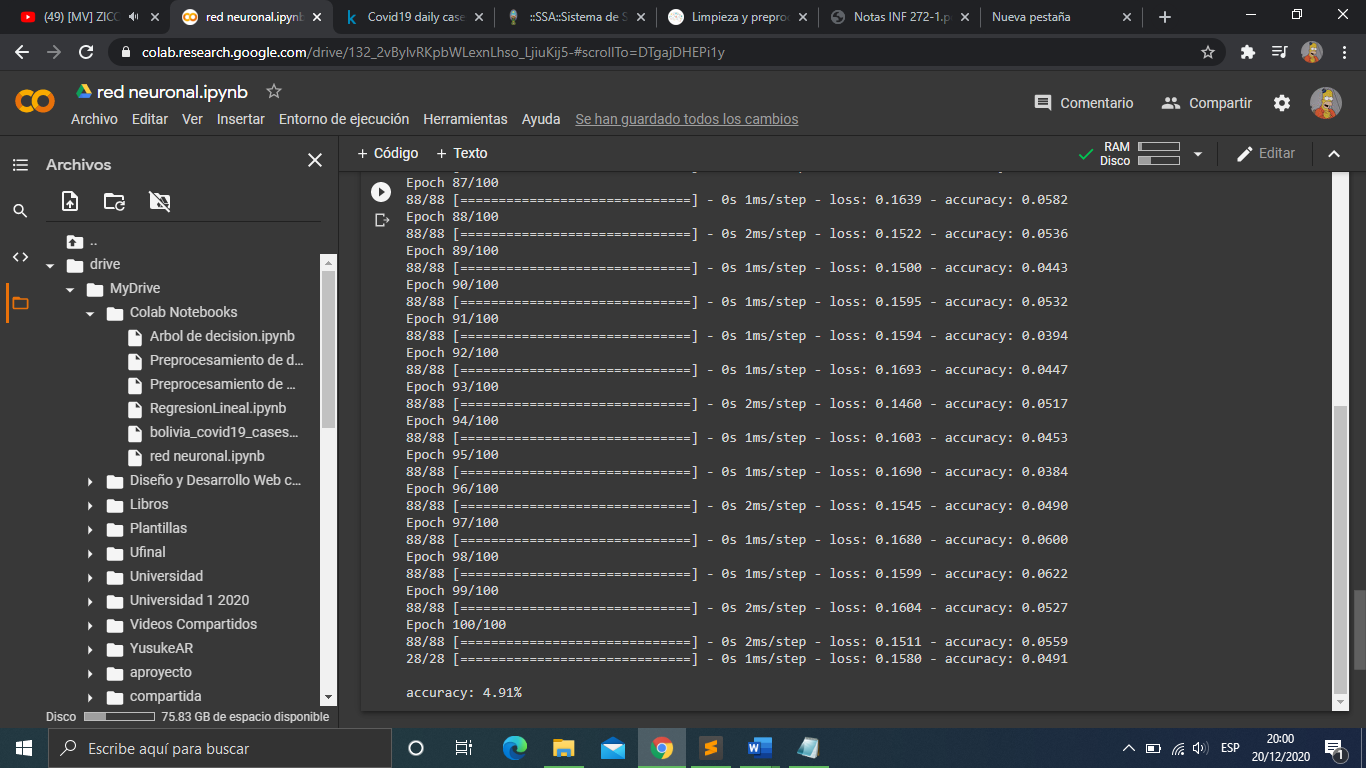
En este caso se realizará un modelo supervisado, para ello se utilizará:

* **Red neuronal**

La estructura será de 3>10>10>1 con 100 épocas, el tamaño de los datos de entrada será de 10.



Se seleccióno casos, recuperados, muertes para el analisis, de tal manera que se pueda predecir

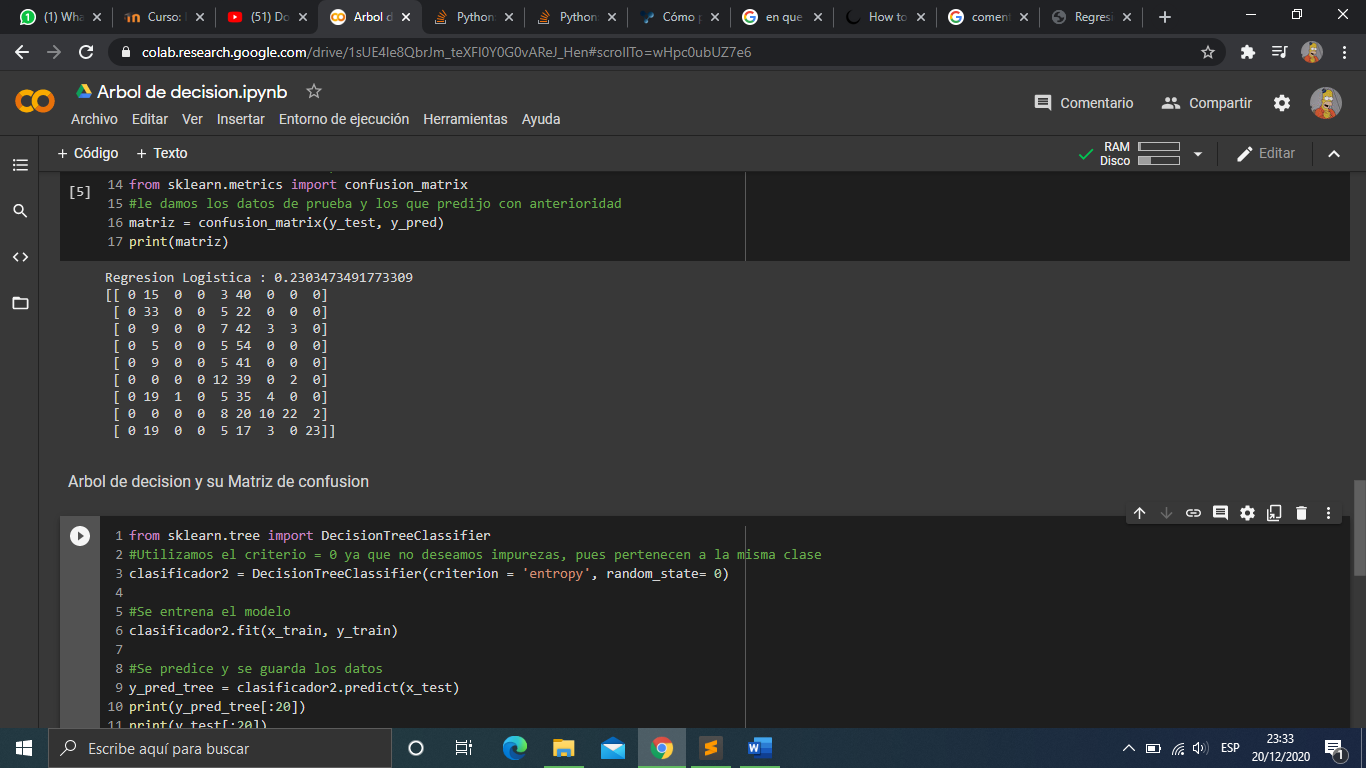
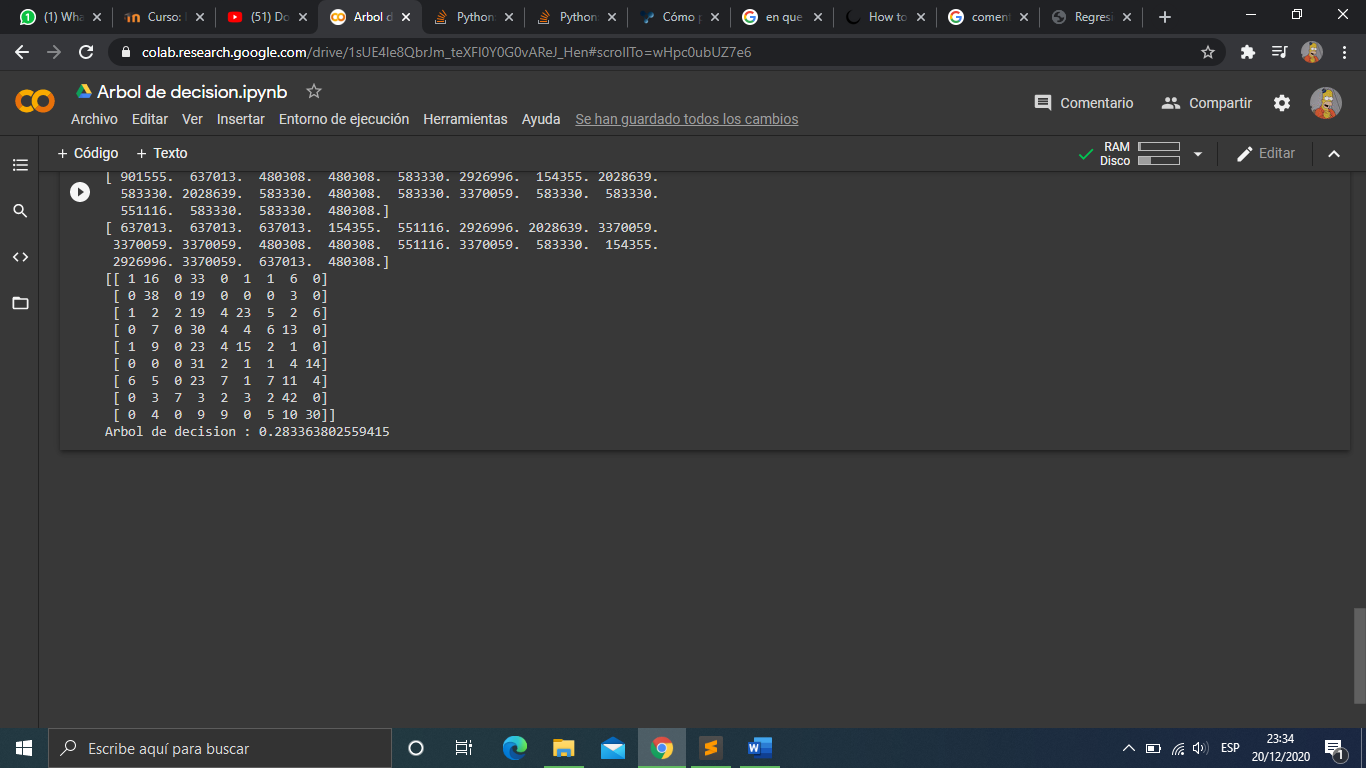


Se selecciono "casos, recuperados, muertes" como x y como y "activos", tal que pueda precedir y

* **Árbol de decisión**

Se realiza una comparación de la regresión logística con el árbol de decisión.

Se tiene como datos x e y, a recuperados y población respectivamente. Además se realiza un preprocesamiento de valores nulos.

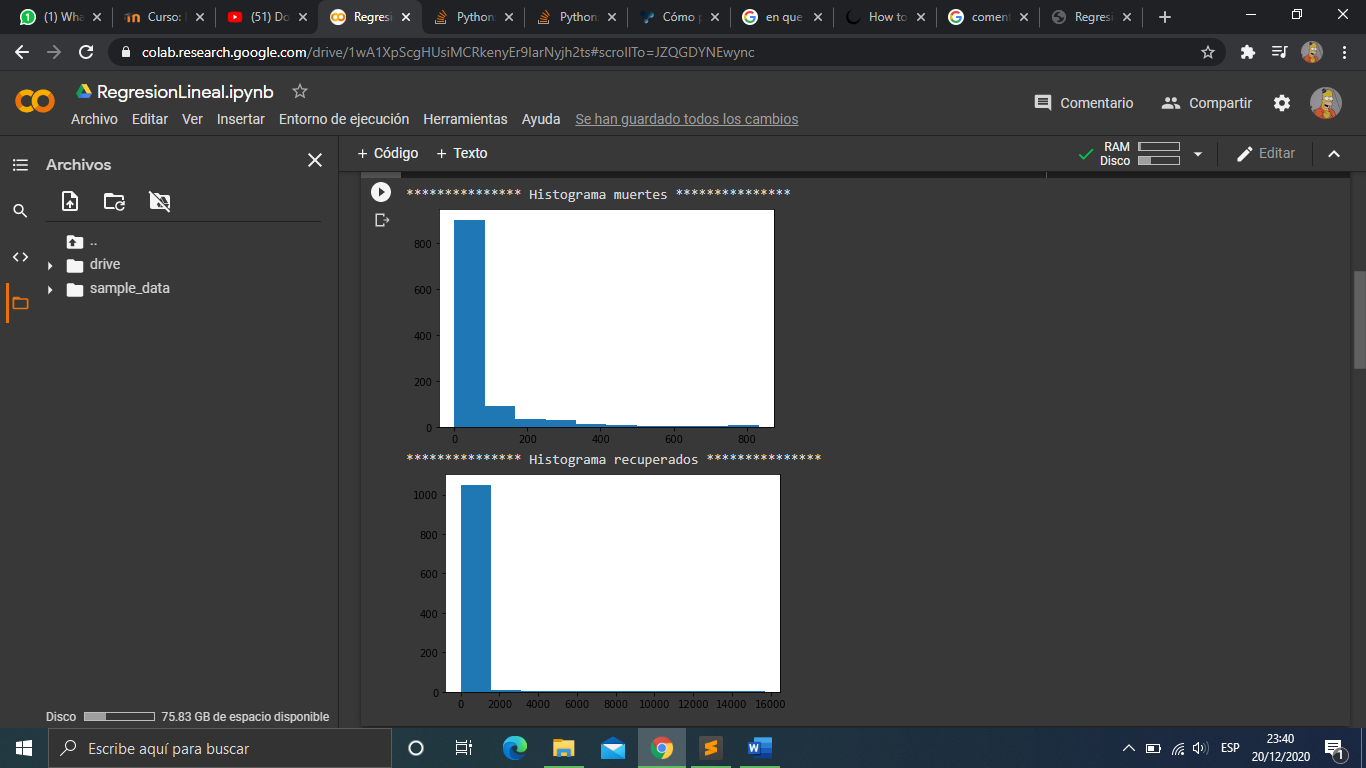
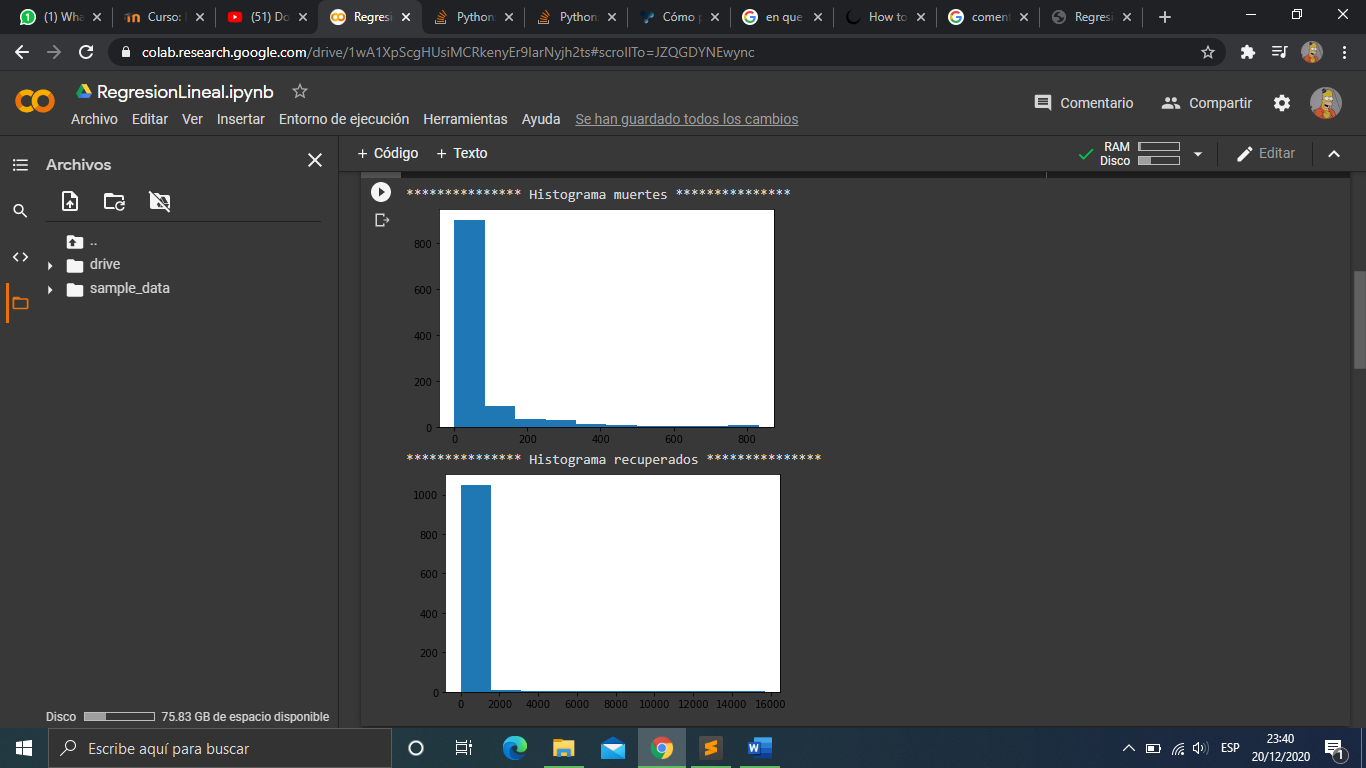
 

Podemos ver que respecto a la regresión logística y el árbol de decisión quien más porcentaje de predicción tiene es el árbol de decisión.

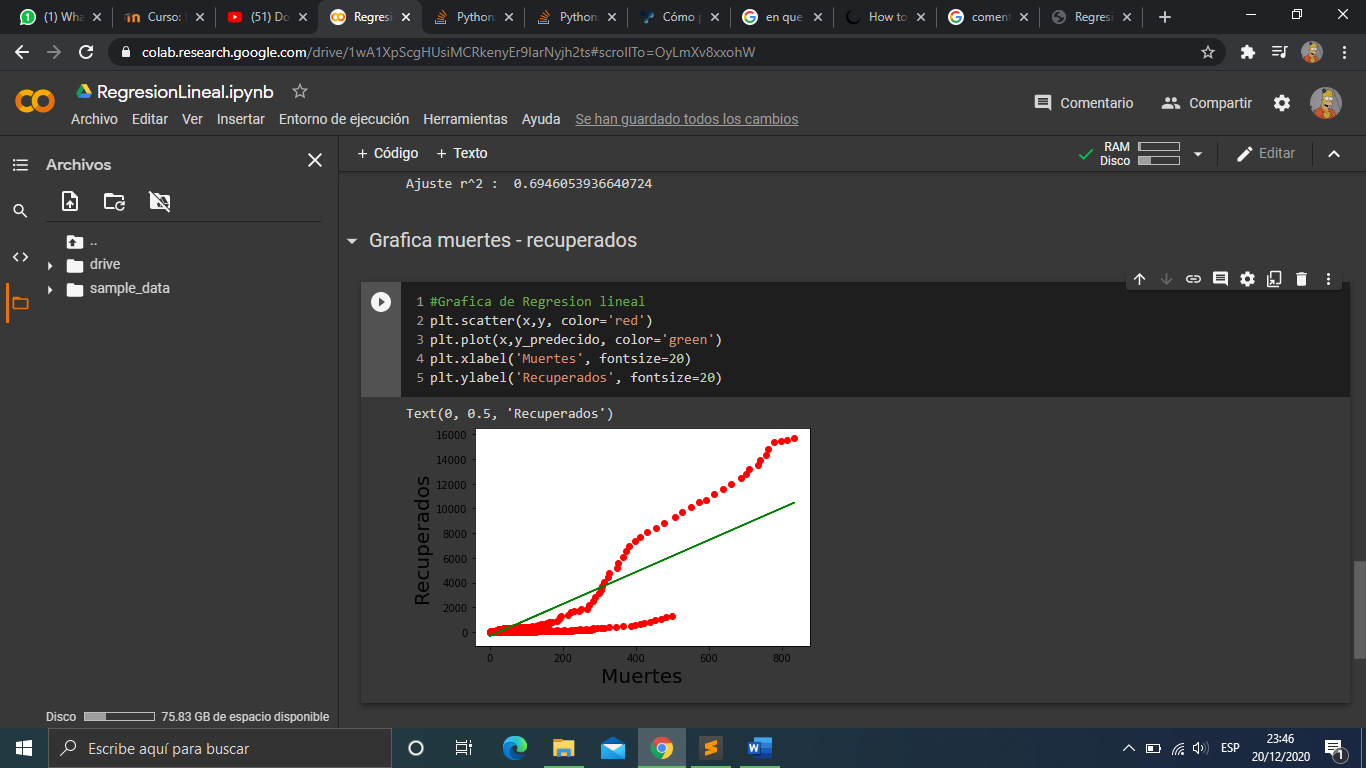
Además, podemos ver la matriz de confusión indicando en que puntos se tuvo más aciertos.

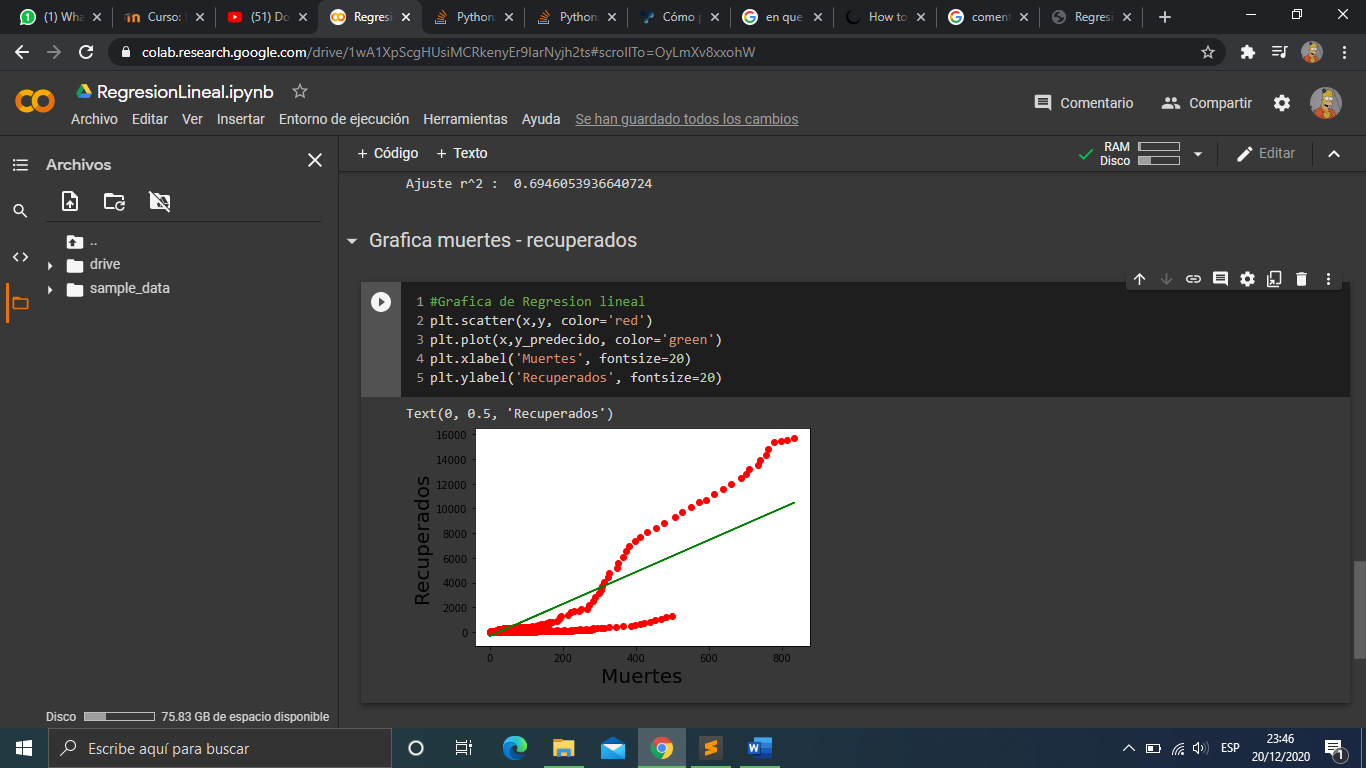
* **Regresión lineal**

Para este al igual que en todos se realizo una limpieza de valores nulos. Las columnas con las cuales se trabajará son muertes y recuperados.

Al aplicar el modelo de regresión lineal, veremos que:





Donde podemos apreciar el ajuste de R2 que esta por arriba de 50%, lo cual es un buen indicador de que el modelo se encuentra bien ajustado.

**Conclusiones**

Los modelos estudiados en este proyecto fueron muy interesantes, se pudo ver que es muy importante elegir que modelo aplicar en cada caso, ya que si se tuviera un dataset de predicción donde generalmente tiene una columna dependiente que determina la verdad o falsedad 1, 0 de las columnas independientes los algoritmos con función sigmoide irían a la perfección. Pero fuera de eso se pudo entender como funcionan los modelos que se usan en inteligencia artificial, claro esta que no a la perfección ya que sigue mucha práctica, puesto que este campo es muy extenso y profundamente complejo.