**Nombre:** Alegre Flores Samuel Alejandro

**CI:** 12391101

**5. Del dataset elegido, migre el mismo a WEKA y utilice cuatro técnicas de preprocesamiento (realice la captura de pantallas de estos por fases). Explique la razón de aplicar estas técnicas.**

**Índice:**

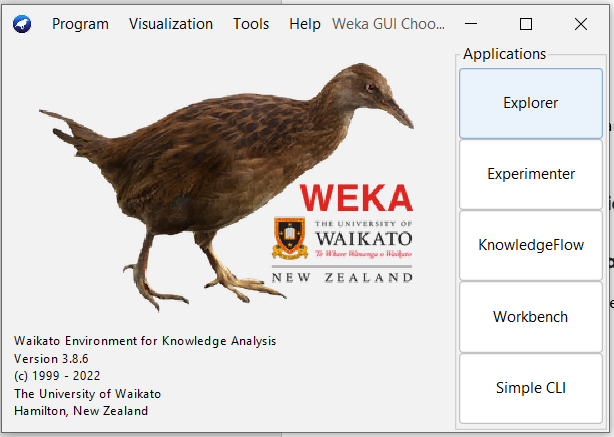
[Preprocesamiento 1: ReplaceMissingValues 2](#_Toc147428941)

[Preprocesamiento 2: Discretizer 5](#_Toc147428942)

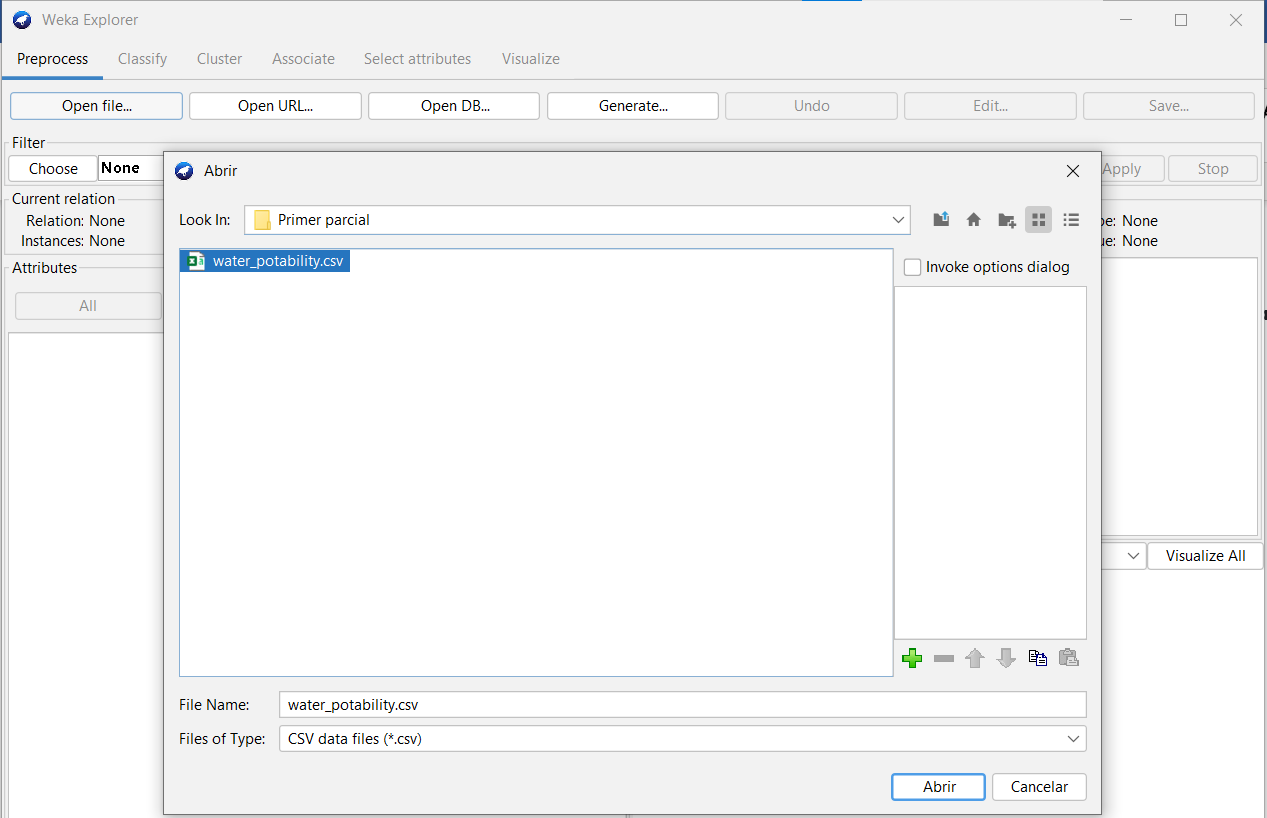
[Preprocesamiento 3: Normalize 8](#_Toc147428943)

[Preprocesamiento 4: RemoveDuplicates 11](#_Toc147428944)

Paso 1: Abrimos weka y seleccionamos el dataset en formato .csv



* Entramos a la opción Explorer



* Seleccionamos el archivo y damos click en abrir, una vez abierto guardamos otra vez en formato .arff y volvemos a abrir este archivo para trabajar con este.

Aplicación

Descripción generada automáticamente con confianza media

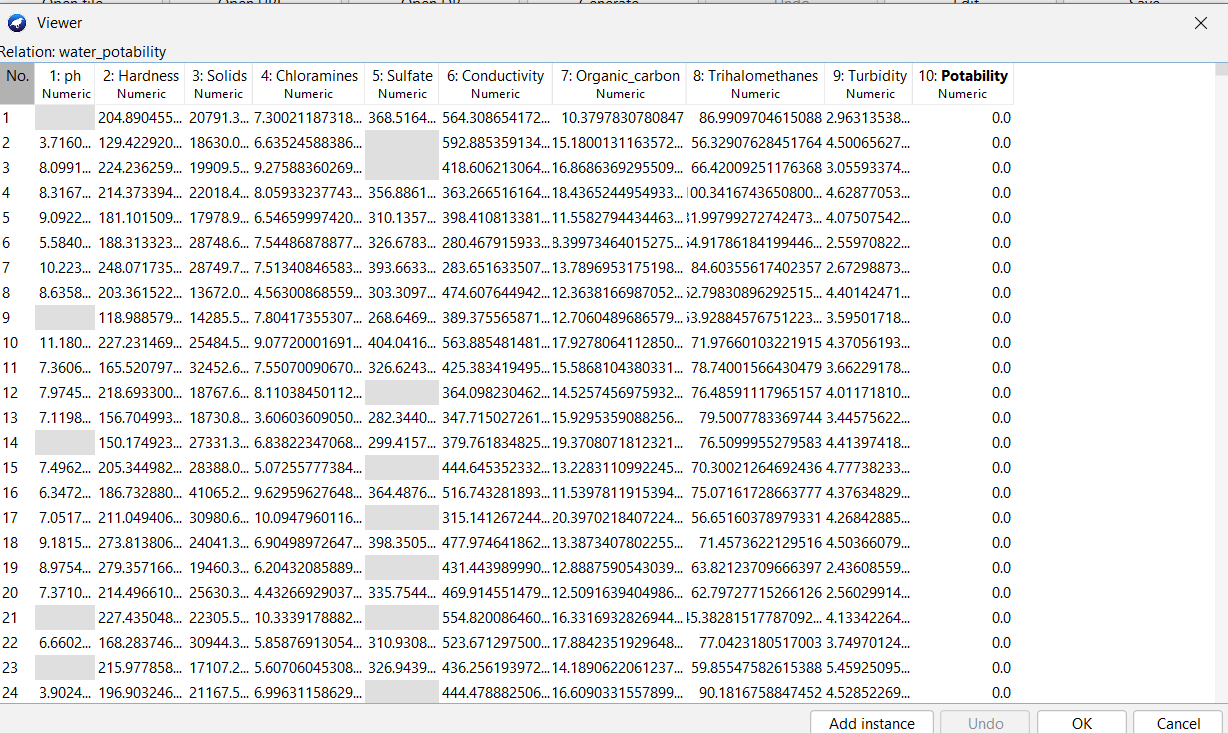
Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

# **Preprocesamiento 1: ReplaceMissingValues**

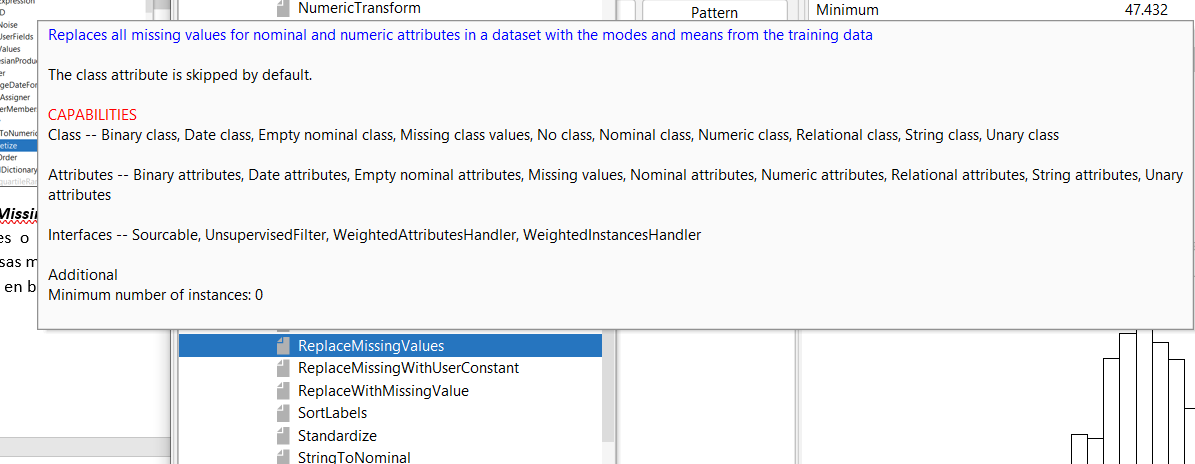
Paso 1: Al dar click en el botón edit podemos percatarnos que el dataset tiene varios valores nulos en sus columnas:

Texto

Descripción generada automáticamente con confianza media

Paso 2: Ahora nos dirigimos a choose>filters>unsupervised>replaceMissingValues

Elegimos no supervisado porque en el aprendizaje no supervisado, tienes un conjunto de datos sin etiquetas, es decir, solo tienes datos de entrada sin información sobre las salidas deseadas.

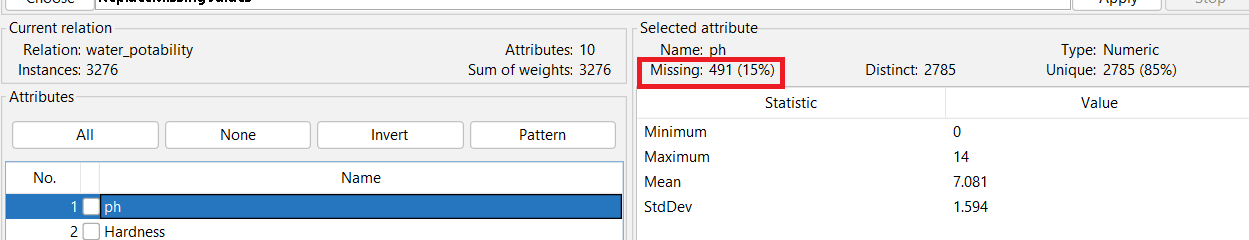


La función ***replaceMissingValues*** se utiliza en el preprocesamiento de datos para manejar y tratar los valores faltantes o ausentes en un conjunto de datos. Estos valores faltantes pueden ser denotados de diversas maneras en un conjunto de datos, como "NaN" (no es un número), "N/A" (no aplicable), espacios en blanco, valores nulos, etc.

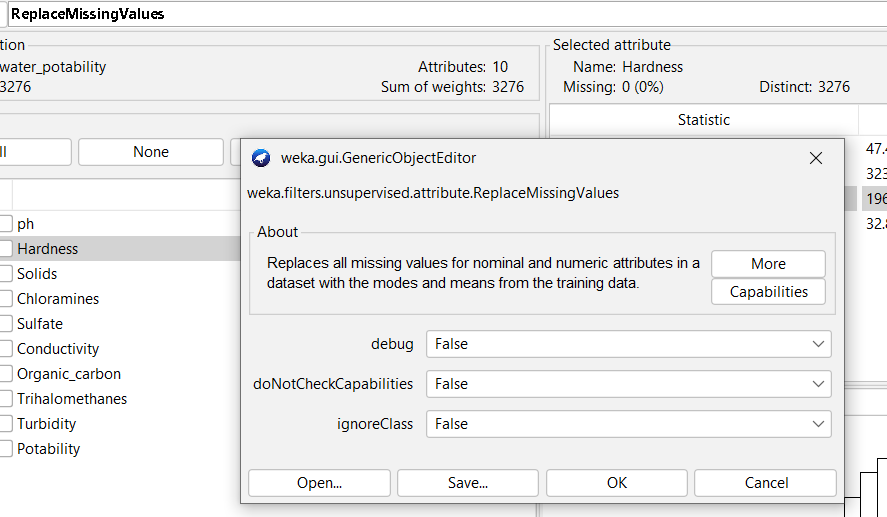
La razón principal para usar replaceMissingValues o técnicas similares es que los algoritmos de aprendizaje automático pueden no funcionar correctamente si se les alimenta con datos que contienen valores faltantes. Estos valores faltantes pueden causar problemas durante el entrenamiento y la predicción de modelos, ya que los algoritmos pueden no saber cómo manejarlos y pueden llevar a resultados incorrectos.

Paso 3: Configuramos el preprocesamiento replaceMissingValues dando click en la barra superior donde aparece el nombre del preprocesamiento:

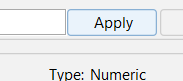
Podemos ver que el atributo ph tiene 15% de valores nulos:



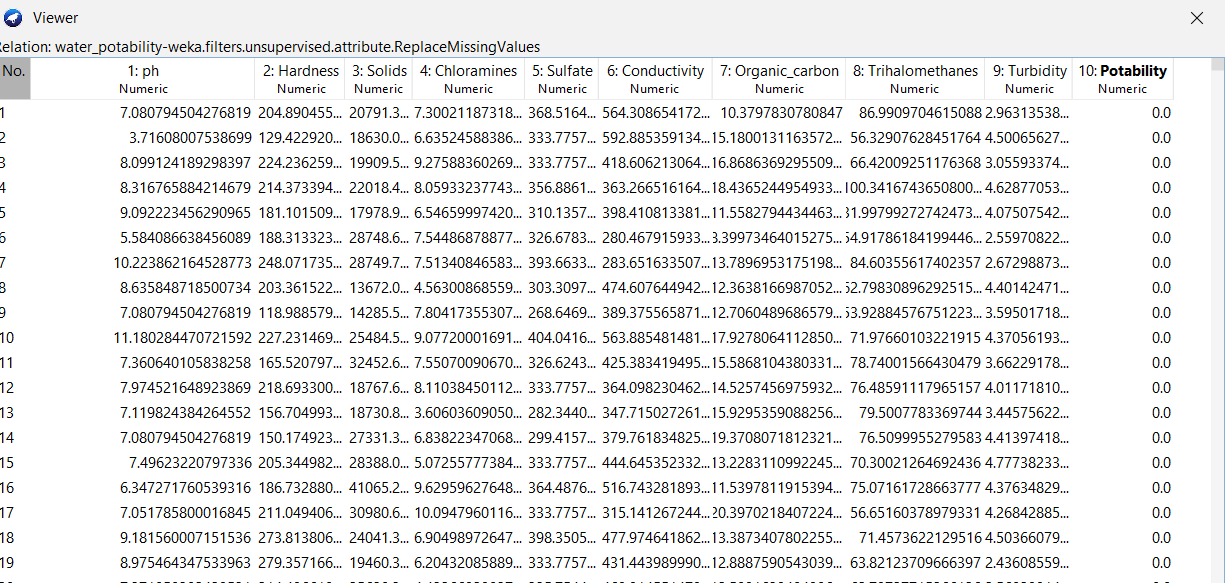
Configuramos el preprocesamiento para que reemplace con la media los valores perdidos:



Le damos en apply



Vemos nuestros datos y observamos que reemplazo los valores vacíos con la media en cada columna:



# **Preprocesamiento 2: Discretizer**

Discretizer se refiere a una técnica de preprocesamiento de datos utilizada en aprendizaje automático y análisis de datos para convertir variables numéricas continuas en variables discretas o categóricas*. Usaremos discretizer porque tenemos datos continuos que casi no se repiten y son más de 3000 datos, para realizar un mejor análisis dividiré los datos de cada columna en 10 rangos.*

Paso 1: Ahora nos dirigimos a choose>filters>unsupervised>discretize

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Paso 2: Configuramos el discretize para que coloque 10 rangos a nuestros datos (bins).

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Colocamos bins en 10, que ignore la clase (La última columna que es el resultado) y le damos en ok

**Antes:**

**Un dibujo de un barco

Descripción generada automáticamente con confianza media**

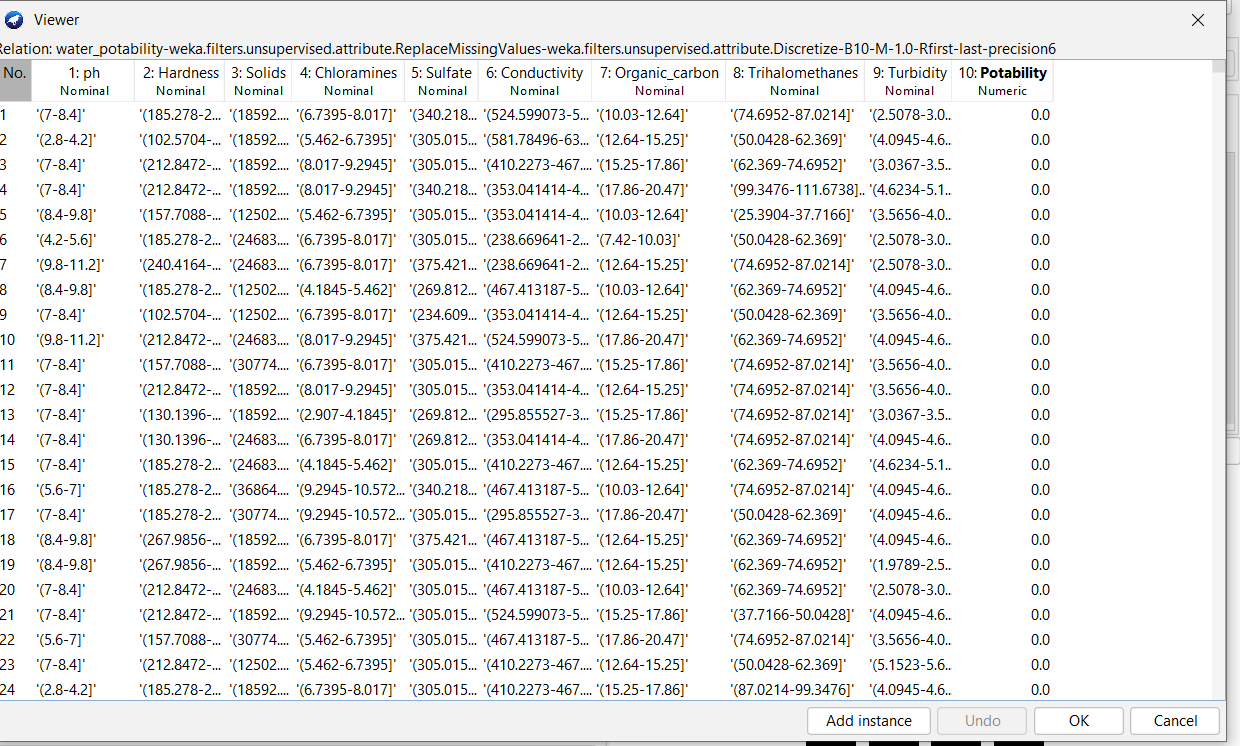
**Después:**

Imagen de la pantalla de un video juego

Descripción generada automáticamente con confianza baja

Podemos notar que son los mismos gráficos obtenidos en el ejercicio 1 del examen.

Los datos cambiaron se discretizaron, y se asemejan mucho a la distribución normal:



# **Preprocesamiento 3: Normalize**

La normalización de datos es esencial en el preprocesamiento de datos para garantizar que todas las características tengan una escala común. Esto ayuda a evitar que características con magnitudes diferentes dominen el proceso de entrenamiento de modelos de aprendizaje automático, asegura la convergencia eficiente de algoritmos y facilita la interpretación de los resultados.

La normalización puede ayudar a reducir el impacto de valores atípicos o extremos al escalar todos los datos dentro de un rango común. Esto puede hacer que el modelo sea más robusto a valores extremos.

Paso 1: Volvemos al estado inicial del dataset haciendo click en undo:

Texto

Descripción generada automáticamente con confianza baja

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Paso 2: Nos dirigimos a choose>filters>unsupervised>normalize y seleccionamos el filtro de preprocesamiento

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Configuramos para que la escala sea de 0 a 1, es decir nuestros datos estarán dentro de esa escala:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Datos antes de Normalizar:

Tabla

Descripción generada automáticamente

Imagen en blanco y negro de un barco

Descripción generada automáticamente con confianza media

Datos después de normalizar:

Texto

Descripción generada automáticamente

Un dibujo de un barco

Descripción generada automáticamente con confianza media

Los datos recopilados en un conjunto de datos pueden no ser lo suficientemente útiles para un algoritmo de minería de datos. A veces, los atributos seleccionados son atributos brutos que tienen un significado en el dominio original de donde se obtuvieron, o están diseñados para funcionar con el sistema operativo en el que se están utilizando actualmente. Por lo general, estos atributos originales

Texto, Carta

Descripción generada automáticamente con confianza media

Formula normalización

Para que funcionen mejor muchos algoritmos de Machine Learning usados en Data Science, hay que normalizar las variables de entrada al algoritmo. Normalizar significa, en este caso, comprimir o extender los valores de la variable para que estén en un rango definido.

# **Preprocesamiento 4: RemoveDuplicates**

"RemoveDuplicates" (Eliminar duplicados) se aplica para eliminar instancias duplicadas de un conjunto de datos. Esto significa que elimina las filas que tienen los mismos valores en todas sus características. Este filtro se utiliza principalmente para limpiar conjuntos de datos y garantizar que cada instancia sea única.

Eliminación de datos redundantes: Los datos duplicados no aportan información adicional y pueden inflar el tamaño del conjunto de datos innecesariamente. Al eliminar duplicados, se reduce la redundancia y se simplifican los datos.

Paso 1: Volvemos al estado inicial del dataset haciendo click en Undo:

Texto

Descripción generada automáticamente con confianza baja

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

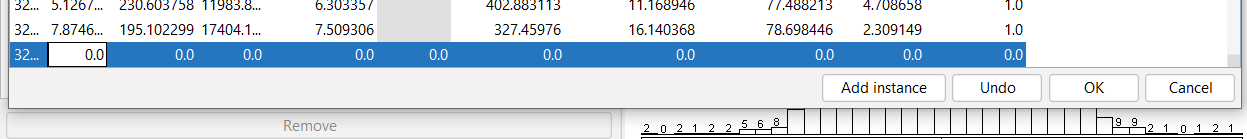
Paso 2: Nos dirigimos a choose>filters>unsupervised>instance>RemoveDuplicates

*Instance se refiere a las filas del dataset y attribute a las columnas.*

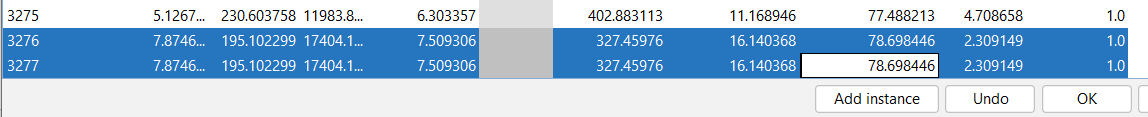
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Paso 3: Añadimos una instancia igual al final para mostrar eliminación:



Tenemos dos filas idénticas:



Paso 4: Aplicamos y configuramos el preprocesamiento RemoveDuplicates:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Paso 5: Aplicamos

Datos Antes:

Tabla

Descripción generada automáticamente

Datos Después:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Podemos observar que la fila duplicada se elimino correctamente. Este filtro es de mucha utilidad para eliminar filas duplicadas que pueden deberse a errores humanos a la hora de registrar los datos en el dataset. Y así reducir el número de filas conservando solo los datos importantes.