**Universidad “Mayor de San Andrés”**

**FACULTAD DE CIENCIAS PURAS Y NATURALES**



**DESAFÍO 1**

Integración por trapecio

**Universitario:** Carlo Wilder Fernando

**Carrera:** Informática

**Docente:** Lic. Brígida Carvajal Blanco

**Paralelo:** Curso verano 2020

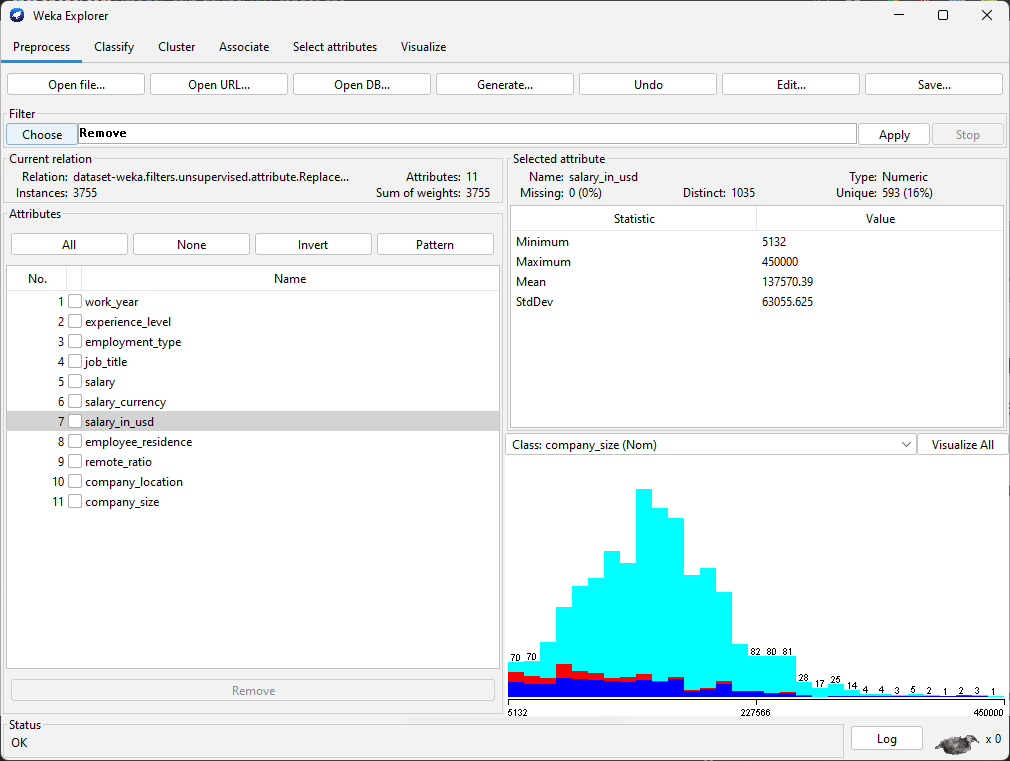
**Fecha:**  25 de enero de 2021

**Ci:** 9204488LP

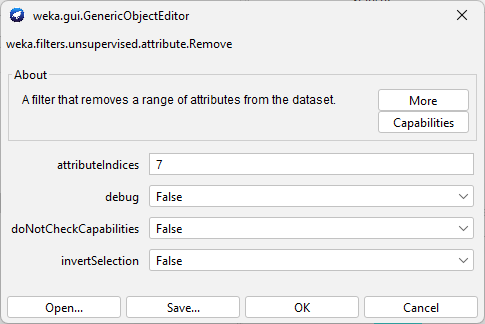
**La Paz - Bolivia**

1. <https://github.com/Will-Carlo/examen354/tree/main/1>
2. Del dataset anterior realice en WEKA, cuatro técnicas de preprocesamiento (realice la captura de pantallas de estos por fases). Explique la razón de aplicar estas técnicas.
   1. Eliminación de instancias o atributos: utilizaremos esta técnica para eliminar instancias o atributos que no son útiles o relevantes para nuestro problema. Utilizaremos el filtro "Remove" para eliminar instancias o atributos.

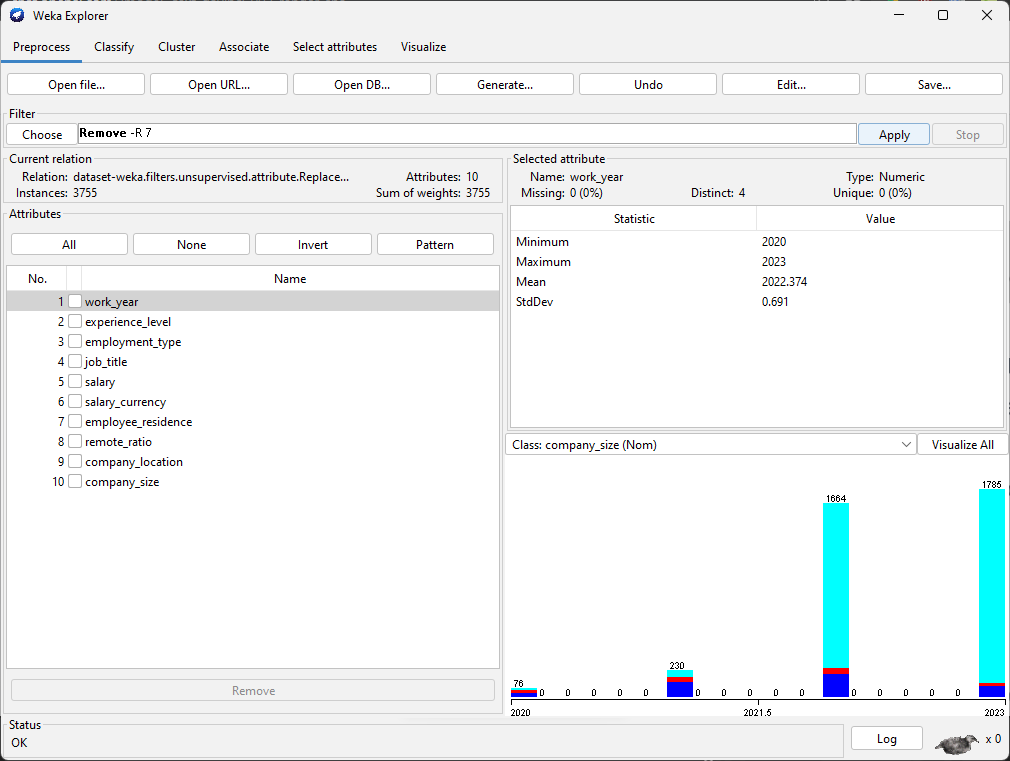
Antes: eliminaremos el atributo ‘salary\_in\_usd’



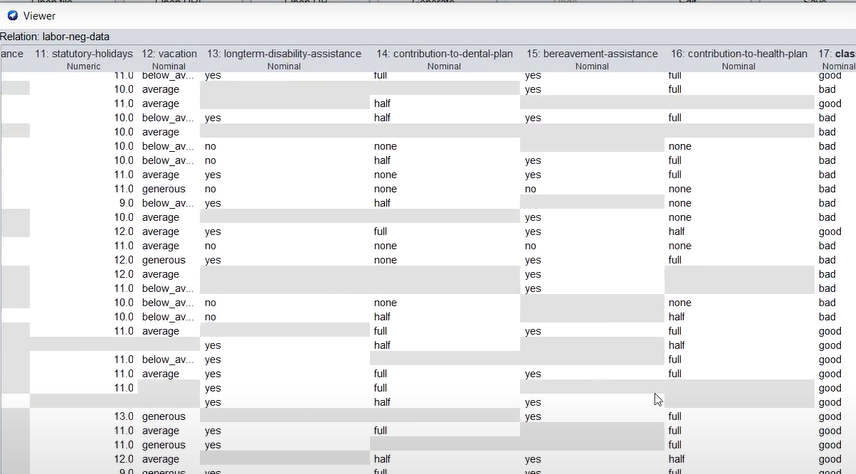
Seleccionamos la columna 7:



Después:

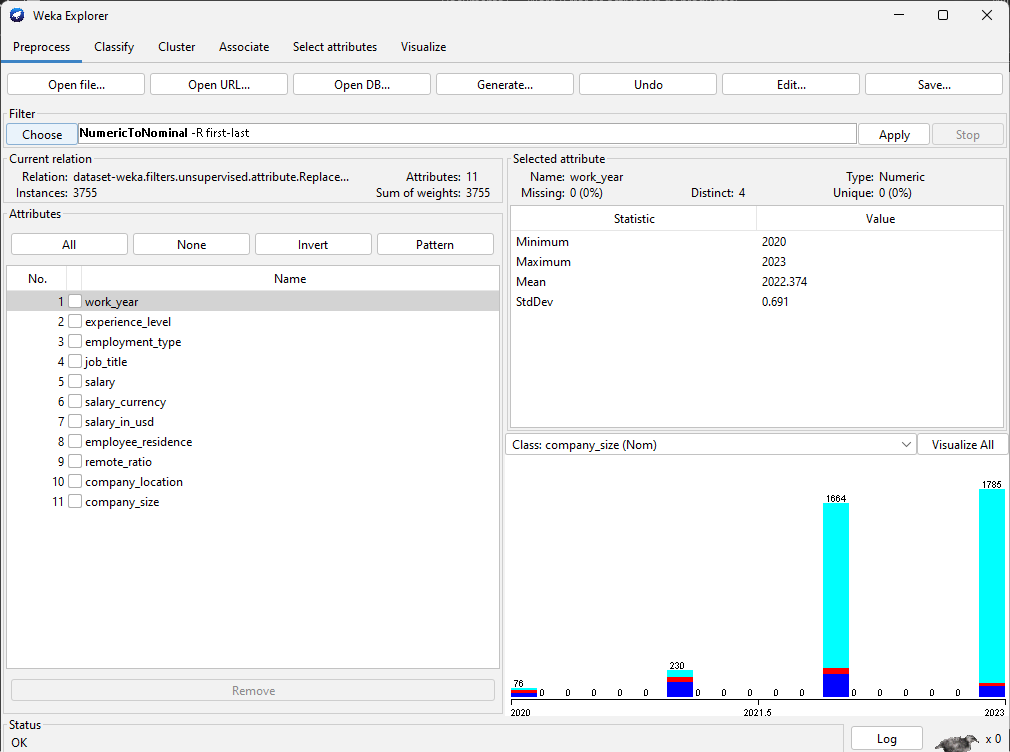


* 1. Usamos Weka para la limpieza de datos: Como vemos a continuación vemos la falta de información, usamos el filtro “ReplaceMissingValues”. Luego podemos aplicar reglas como el OneR para ver los cambios.

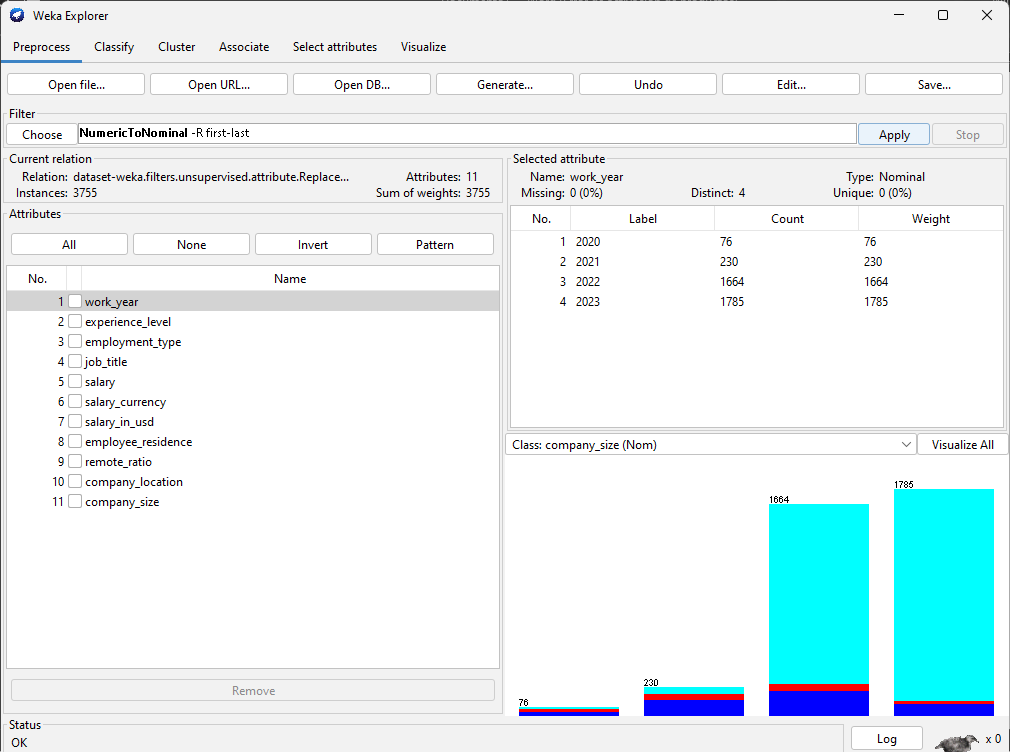


* 1. Transformación de datos: esta técnica la utilizamos para transformar los datos en una forma más adecuada para su procesamiento. Utilizaremos el filtro "NumericToNominal" para convertir los datos numéricos en datos nominales.

Antes:

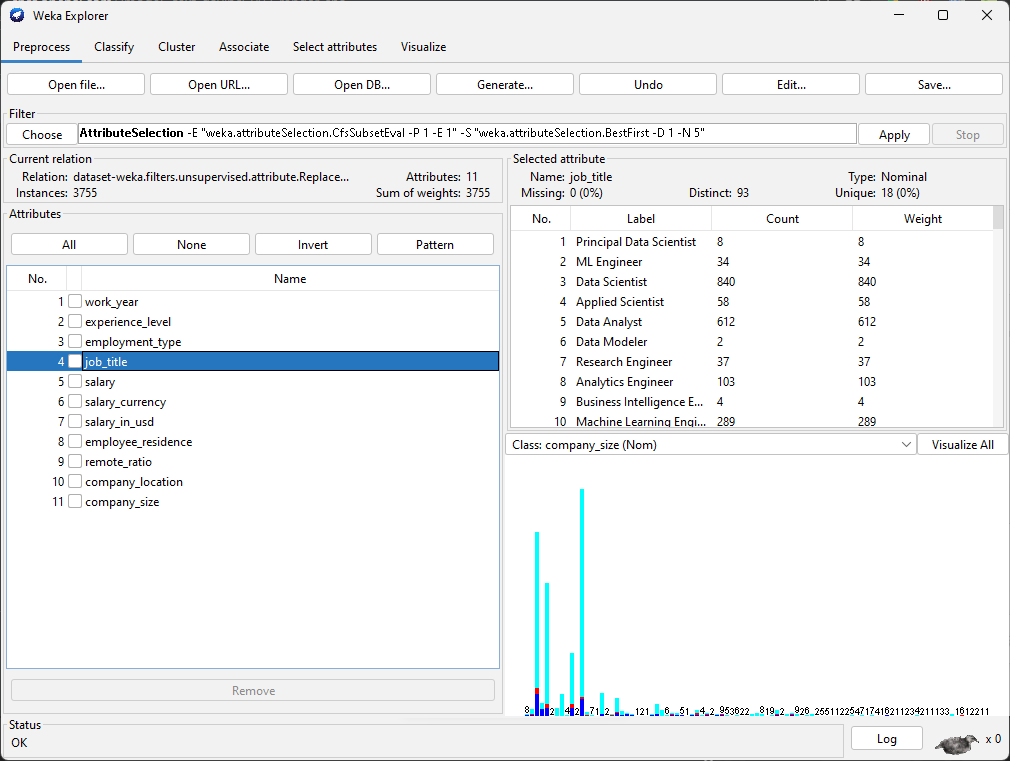


Después:

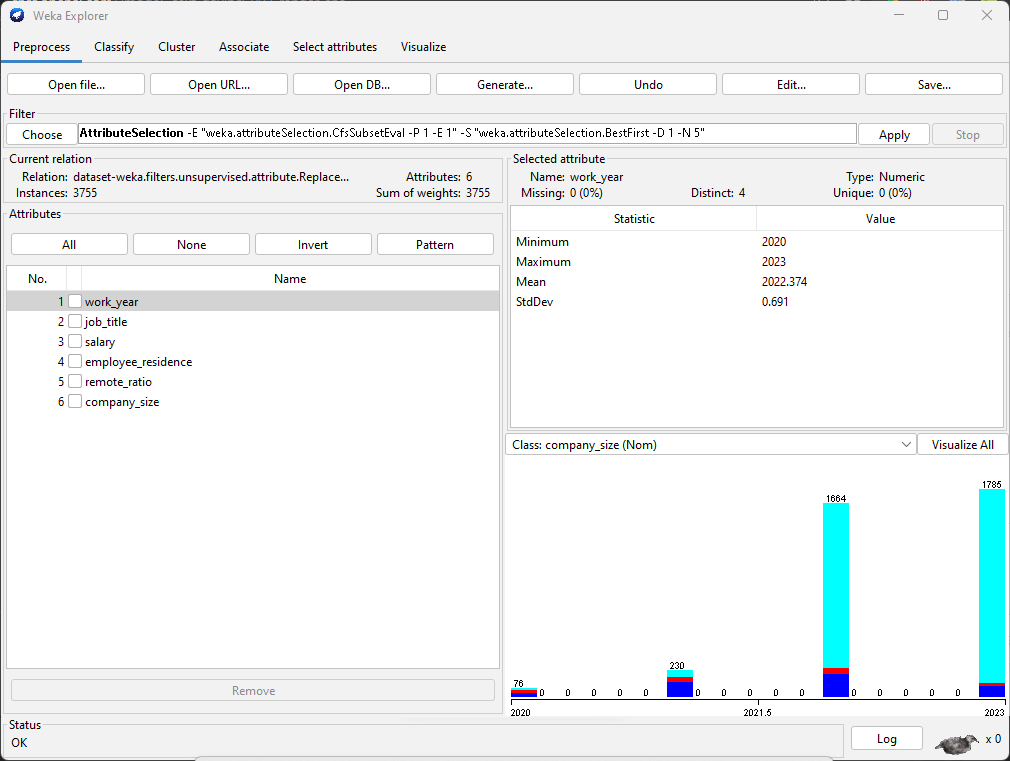


* 1. Selección de atributos: esta técnica la utilizaremos para seleccionar los atributos más relevantes para el problema que se está resolviendo. Utilizaremos el filtro "AttributeSelection" para seleccionar los atributos utilizando diferentes criterios, como la ganancia de información o la correlación.

Antes:



Después:



1. Del dataset anterior realice en PYTHON, tres algoritmos de preprocesamiento. Explique la razón de aplicar estas técnicas.

Implementaremos en Python utilizando las bibliotecas Pandas y Scikit-learn.

* 1. **Limpieza de datos faltantes:**

Es común encontrar datasets incompletos, en donde faltan valores en algunas celdas. Estos valores faltantes pueden afectar negativamente el rendimiento del modelo de aprendizaje automático. Por lo tanto, es importante realizar una limpieza de los datos faltantes.

En Python, se puede utilizar la función `dropna()` de Pandas para eliminar filas con valores nulos. Por ejemplo, si el conjunto de datos se almacena en un DataFrame de Pandas llamado `df`, se puede limpiar utilizando la siguiente línea de código:

* df.dropna(inplace=True)
  1. **Normalización de datos:**

En muchos casos, las características o variables del dataset pueden tener diferentes escalas y rangos. Por ejemplo, algunas características pueden estar en el rango de 0 a 1, mientras que otras pueden estar en el rango de 0 a 1000. Esto puede generar problemas en el rendimiento del modelo de aprendizaje automático, por lo que se requiere normalizar los datos.

En Python, se puede utilizar la función `MinMaxScaler` de Scikit-learn para normalizar los datos. Por ejemplo, si el conjunto de datos se almacena en un DataFrame de Pandas llamado `df`, se puede normalizar utilizando la siguiente línea de código:

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df\_normalized = scaler.fit\_transform(df)

* 1. **Codificación de variables categóricas:**

En algunos casos, las características o variables pueden ser categóricas y tener valores como 'Sí' o 'No', o 'Hombre' o 'Mujer'. Los modelos de aprendizaje automático no pueden trabajar con valores categóricos, por lo que se debe realizar una codificación de estas variables para que sean representadas por números.

En Python, se puede utilizar la función `get\_dummies()` de Pandas para codificar variables categóricas. Por ejemplo, si el conjunto de datos se almacena en un DataFrame de Pandas llamado `df`, se puede codificar las variables categóricas utilizando la siguiente línea de código:

df\_encoded = pd.get\_dummies(df)

En resumen, la limpieza de los datos faltantes, la normalización de los datos y la codificación de las variables categóricas son técnicas comunes de preprocesamiento de datos que pueden mejorar el rendimiento del modelo de aprendizaje automático.