Consumer Mit Lab Evidencia 2 Equipo2

En la elaboración de este código para hacer una limpia de datos, para tener un archivo el cual estuviese libre de nulos de primera instancia para poder hacer que este código funcione correctamente como primer paso de la elaboración se necesito importar las librerías pertinentes para hacer que el código funcione tal como se necesita.

```
#Importamos las librerías pandas, numpy y matplotlib respectivamente
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as special
from scipy.optimize import curve_fit
import seaborn as sns
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

Una vez que tenemos las librerías importadas cargamos nuestro archivo para empezar a trabajar sobre el

```
#Cargar archivo csv desde equipo
from google.colab import files
files.upload()
```

Y cargamos el archivo de consumer_mit_lift_lab.xlsx

```
#Carga desde un archivo .csv sin indice
Retailer= pd.read_excel('consumer_mit_lift_lab.xlsx')
```

Para así poder trabajar en él, lo primero que hacemos es verificar cual es la información que tiene nuestro Data Frame para así saber sobre qué variables vamos a tener que trabajar para poder eliminar los nulos del modelo.

```
#Verificamos información del Dataframe
Retailer.info()
```

Después se realiza una copia del adat para así poder trabajar sobre ella al momento de estar eliminando los datos nulos y dejando un archivo libre de estos:

```
#Creamos copia del dataframe
data1=Retailer.copy()
data1
```

Al haber terminado está fase, comienza la fase del procesamiento de datos bajo métodos de sustitución para poder limpiar los datos nulos e innecesarios que se encuentran dentro del Data Frame, primer se identifica cuales son los valores nulos que se tiene dentro del Data Frame:

```
#Identificar valores nulos por columna
valores_nulos=data1.isnull().sum()
valores_nulos
```

Luego utilizamos un método con el cual sustituiremos los valores nulos que están por detrás de valores no nulos por ese mismo valor que buscamos sustituir:

```
#Utilizamos un método con el que sustituimos los valores nulos que estén por detrás de valores no nulos por ese mismo valor
data2= data1.fillna(method="bfill")
data2
```

Nuevamente corroboramos los valores nulos para poder determinar si hay que volver a eliminar datos o la tarea quedó completa:

```
#Corroboramos valores nulos
valores_nulos=data2.isnull().sum()
valores_nulos
```

Bajo este segundo análisis aun se determinó que existen valores nulos dentro del Data Frame, así que debido a que aún quedan valores nulos en el data frame, utilizamos el método "ffill" para cambiar los valores nulos que estén delante de un valores no nulo para cambiarlo por el mismo valor

```
#Debido a que todavia nos quedan unos cuantos valores nulos en el dataframe, utilizamos el metodo "ffill" para cambiar los valores nulos que estén delante de un valores no nulo para cambiarlo por el mismo valor
data3= data2.fillna(method='ffill')
data3
```

Nuevamente corroboramos los valores nulos para poder determinar si hay que volver a eliminar datos o la tarea quedó completa:

```
#Corroboramos valores nulos
valores_nulos=data3.isnull().sum()
valores_nulos
```

Y en caso de que resten unos cuantos valores nulos en el data frame, utilizamos el método de cambiar los valores nulos por un string en concreto para seguridad

```
#En caso de que resten unos cuantos valores nulos en el dataframe, utilizamos el metodo de cambiar los valores nulos por un string en concreto para seguridad
data4= data3.fillna('no valido')
data4
```

Ya por último buscamos la información que tenemos del data que trabajamos para saber sobre qué variables trabajaremos.

```
data4.info()
```

Nuevamente realizamos una copia al data para trabajar sobre ella

```
#Creamos otra copia del dataframe
data5=data4.copy()
```

Realizamos un filtro por filas donde se incluyan únicamente los valores cuantitativos, para que de este modo, la selección de datos sea más específicas enfocada en los valores y datos que realmente son importantes

```
#Realizamos un filtro por filas donde se incluyan únicamente los valores cuantitativos
cuantitativas=data5.iloc[:,[12,26,27,28,29,30,31]]      #filas      no consecutivas
cuantitativas
```

Así mismo realizamos un filtro por filas donde se incluyan únicamente los valores cualitativas

```
#Realizamos un filtro por filas donde se incluyan únicamente los valores cualitativas
cualitativas=data2.iloc[:,[0,1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,32,33,34,35]] #filas no consecutivas
cualitativas
```

Corroboramos cuales son los valores nulos que tenemos en el Data Frame

```
#Corroboramos valores nulos
valores_nulos=cuantitativas.isnull().sum()
valores_nulos
```

Una vez que tenemos esto aplicamos la desviación estándar para así encontrar los valores externos que tenemos en nuestro Data Frame

```
#Aplicamos desviación estándar para encontrar valores extremos
y=cuantitativas
Limite_Superior= y.mean() +3*y.std()
Limite_Inferior= y.mean() -3*y.std()
print("Limite superior permitido", Limite_Superior)
print("Limite inferior permitido", Limite_Inferior)
```

Así podemos encontrar los Outliers que tiene el Data Frame

```
#Encontramos Outliers del Dataframe
outliers= cuantitativas[(y>Limite_Superior)|(y<Limite_Inferior)]
outliers
```

Así mismo como obtener ya los datos limpios del Data Frame

```
#Obtenemos datos limpios
data5= cuantitativas[(y<=Limite_Superior)&(y>=Limite_Inferior)]
data5
```

Posteriormente se revisa si dentro del Data Frame aún existen valores atípicos los cuales puedan llegar a afectar al archivo limpio

```
#Revisamos valores atípicos (nulos) del dataframe4
valores_nulos=data5.isnull().sum()
valores_nulos
```

Por lo que se reemplazan los valores atípicos con “mean”, realizando una copia del Data Frame para trabajar sobre esta versión limpia

```
#Reemplazamos valores atípicos (nulos) del dataframe con "mean"
#Realizamos una copia del dataframe
data_clean=data5.copy()
data_clean=data_clean.fillna(round(data3.mean(),1))
data_clean
```

Por lo que se corrobora nuevamente que el Data este limpio

```
#Corroboramos valores nulos del dataframe LIMPIO
valores_nulos=data_clean.isnull().sum()
valores_nulos
```

Así es que unimos las columnas de los datos cuantitativos con los cualitativos dentro del mismo Data Frame

```
#Unimos las columnas cuantitativas y cualitativas en un mismo dataframe
Datos_limpios = pd.concat([cualitativas, data_clean], axis=1)
Datos_limpios
```

Para poder convertir nuestro Data en un archivo .csv en el cual ya tendremos los valores limpios, listos para sus regresiones e interpretaciones

```
#Corroboramos valores nulos del dataframe LIMPIO
valores_nulos=data_clean.isnull().sum().sum()
valores_nulos
```

Por lo que por última vez se realiza un conteo de nulos, en el cual se espera que el resultado sea=0 para así poder descargar el archivo en formato .csv y utilizarlo sin ni outliers.

```
#descargar archivo filtrado en csv
from google.colab import files

files.download("Datos_limpios_Consumer_Mit_Lab_Evidencia.csv")
```