Group Name : Dream Crushers

Name : Aiman Lameesa
Email: st122876@ait.asia
Country :Bangladesh
College: Asian Institute of Technology
Specialization (Data Science, NLP, Data Analyst): Data Science

“Customer Segmentation” dataset is selected for my group project.

Data Cleansing and Transformation:

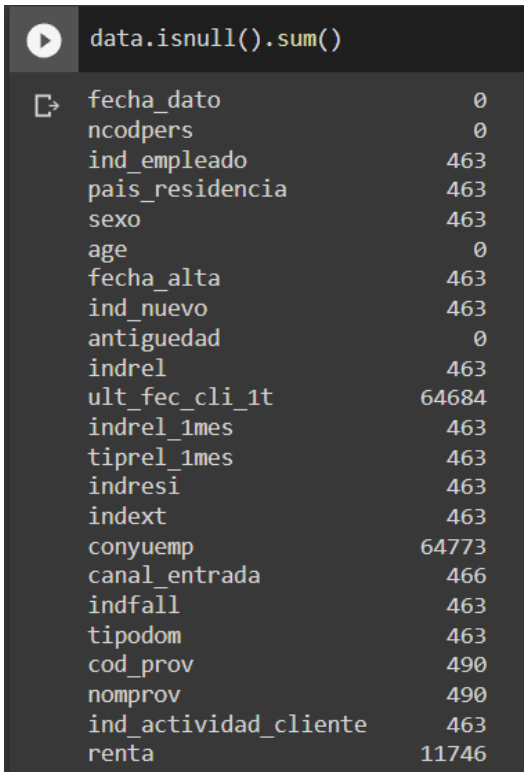
Handling NA values:

The steps applied to handle NA values in the dataset are given below:

1. **Checking Null Values:** First, the sum of null values for each column is checked. For this step, I used the following line of code:

   ```
   data.isnull().sum()
   ```

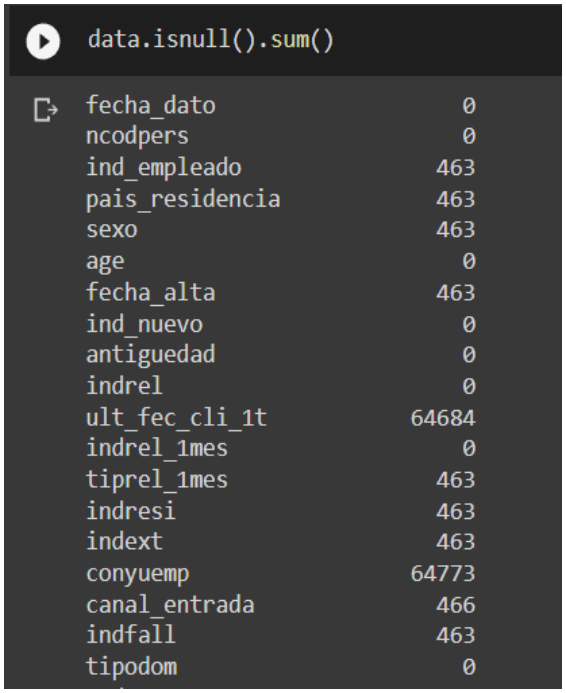
   Thus, the total number of null values of each column can be obtained:



2. **Replacing Null Values using Median based Approach:** Then median based approach is used to replace the null values in the dataframe.

```
data.fillna(data.mean(), inplace = True)
```

Again, the total number of null values of each column is obtained:



The null values in numerical columns e.g **“ind_nuevo”** and **“indrel”** are removed. Then null values in categorical columns or columns that contain object type data remain after this step.

3. **Mode based Approach for Handling Null Values in Categorical Columns:** To remove null values in categorical columns, mode based approach is used.

```
data.fillna(data.select_dtypes(include='object').mode().iloc[0],
inplace=True)
```

Again, the total number of null values of each column is checked for verification:

```
data.isnull().sum()
```

```
fecha_dato                0
ncodpers                  0
ind_empleado              0
pais_residencia           0
sexo                      0
age                       0
fecha_alta                0
ind_nuevo                 0
antiguedad                0
indrel                    0
ult_fec_cli_1t            0
indrel_1mes               0
tiprel_1mes               0
indresi                   0
indext                    0
conyuemp              64773
canal_entrada             0
indfall                   0
tipodom                   0
cod_prov                  0
nomprov                   0
```

The null values in categorical columns e.g **“ind_empleado”** and **“pais_residencial”** are removed. The null values in the **“conyuemp”** column only remain after this step.

4. **Filling Null Values with Unknown Class:** In the dataset, the **“conyuemp”** column is fully empty. Since there are neither numeric nor categorical values in this column, the above method does not work for this column. That is why this column is filled with “Unknown” class.

```
data['conyuemp'] = data['conyuemp'].fillna("Unknown")
```

Lastly, the total number of null values of each column is checked if there is any null values left in any of the columns:

```
data.isnull().sum()
```

```
fecha_dato              0
ncodpers                0
ind_empleado            0
pais_residencia         0
sexo                    0
age                     0
fecha_alta              0
ind_nuevo               0
antiguedad              0
indrel                  0
ult_fec_cli_1t          0
indrel_1mes             0
tiprel_1mes             0
indresi                 0
indext                  0
conyuemp                0
canal_entrada           0
indfall                 0
tipodom                 0
cod_prov                0
nomprov                 0
ind_actividad_cliente   0
renta                   0
```

Here it can be obtained that the dataframe does not have any null values in any of the columns. Thus, the methods can be successfully applied to handle the null values in this data.

Handling Outliers:

The steps applied to handle outliers in the dataset are given below:

1. **Detecting Outliers using Inter Quantile Range (IQR):** In order to remove outliers from the data, first I obtained the outliers for each feature in the dataset. For this step, I used the following code:

```
Q1 = data.quantile(0.25)

Q3 = data.quantile(0.75)

IQR = Q3 - Q1

print(IQR)
```

The following result is obtained in this step:

```
ncodpers                63294.00
ind_nuevo                   0.00
indrel                      0.00
indrel_1mes                 0.00
tipodom                     0.00
cod_prov                   24.00
ind_actividad_cliente       1.00
renta                   52935.87
ind_ahor_fin_ult1           0.00
ind_aval_fin_ult1           0.00
ind_cco_fin_ult1            0.00
ind_cder_fin_ult1           0.00
ind_cno_fin_ult1            0.00
ind_ctju_fin_ult1           0.00
ind_ctma_fin_ult1           0.00
ind_ctop_fin_ult1           0.00
```

It is obtained that there are outliers in a few columns, i.e, **“ncodpers”**, **“renta”** etc . The obtained outliers are removed in the next step.

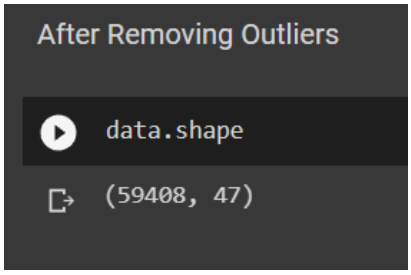
2. **Removing Outliers:** To remove the outliers, first the outliers are replaced with null values. Then the null values are removed from the dataframe. I used the following code to replace the outliers with null values, for example, for the **“ncodpers”** column:

```
for x in ['ncodpers']:

    q75,q25 = np.percentile(data.loc[:,x],[75,25])

    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)

    min = q25-(1.5*intr_qr)

    data.loc[data[x] < min,x] = np.nan

    data.loc[data[x] > max,x] = np.nan
```

Then I used the following line of code to remove the null values from the dataframe:

```
data = data.dropna(axis = 0)
```

Finally the dataframe is checked, where it is confirmed that the number of rows has ben reduced to 59408 from 64773.



Thus, the outliers in the data can be removed successfully after detecting them and replacing them with null values.

Github Repo Link:

aimanlameesa/Week-9 (github.com)