Tecnológico de Monterrey

Tecnológico de Monterrey

Reporte preparación de la base de datos (Consumer)

Escuela: Instituto Tecnológico de Estudios Superiores de Monterrey

Materia: Desarrollo de proyectos y análisis de datos

Profesor: Alfredo García Suárez

Nivel Académico: Profesional

Ciudad: Puebla

Autores

Omar Eduardo Pelcastre Reyes	A01735985@tec.mx
Saúl Jesús Cuervo Méndez	A01735937@tec.mx
Juan José Lara García	A01736667@tec.mx
Cristian Marino Gutiérrez Jiménez	A01736337@tec.mx
Kevin Vergara Lara	A01735970@tec.mx
Marco Ivan Olalde Gonzalez	A01733378@tec.mx

Etapa 2: Preparación de la base de datos

Como primer paso, instalamos las librerías a utilizar, cargamos el archivo excel a utilizar, que en este caso fue el de Consumer.

```
%pip install funpymodeling
%pip install pandas
```

```
#importamos papalerias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from funpymodeling.exploratory import freq_tbl
import seaborn as sns
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
```

```
from google.colab import files
files.upload()
```

Identificamos los valores nulos que tenía este archivo, y se sabe que estos valores se pueden sustituir por diferentes métodos, en este caso nosotros ocupamos el método fillna, que este lo que hace es sustituir los valores por un str en concreto, se escogió la palabra “Nulo” para rellenar dichos espacios vacíos.

```
#reemplazamos los valores nulos con la palabra nulo
data=data1.copy()
data["300_did_not_find"] =data["300_did_not_find"].fillna("nulo")
data["284_additional_products_store"] =data["284_additional_products_store"].fillna("nulo")
data["305_electricity_bill_store"] =data["305_electricity_bill_store"].fillna("nulo")
data["306_water_bill_store"] =data["306_water_bill_store"].fillna("nulo")
data["308_topups_in_store"] =data["308_topups_in_store"].fillna("nulo")
data["304_additional_services_store"] =data["304_additional_services_store"].fillna("nulo")
data["storefront_picture_just_once_if_possible"] =data["storefront_picture_just_once_if_possible"].fillna("nulo")
data["309_wiretransfers_in_store"] =data["309_wiretransfers_in_store"].fillna("nulo")
```

```
#verificamos valores nulos
valores_nulos=data.isnull().sum()
valores_nulos
```

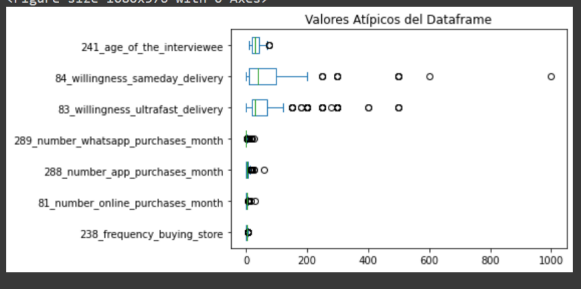
```
#aplicamos filtro
filtro1=data.iloc[ : , [0,1,2,3,4,7,8,9,10,11,13,14,15,16,17,18,19,20,21,22,23,24,25,32,33,34,35]] #obj
filtro2=data.iloc[ : , [12,26,27,28,29,30,31]]    #int
filtro3=data.iloc[ : , [5,6]]     #float
```

Al ser una base de datos con muchas columnas, se recurre a la utilización de filtro por columnas, con el fin de que la sumatoria de valores en todo el dataset sea igual a 0.

El segundo paso a seguir consiste en realizar las gráficas para encontrar nuestros valores atípicos (outliers y concluir con la limpieza de nuestros datos).

```
[14] #hacemos una grafica de valores atipicos
     fig = plt.figure(figsize =(15, 8))
     filtro2.plot(kind='box', vert=False)
     plt.title("Valores Atípicos del Dataframe")
     plt.show()
```

```
<Figure size 1080x576 with 0 Axes>
```



Para este caso en particular, utilizaremos el método intercuartilico, la cual se basa en delimitar los límites permitidos y que entran en un rango aceptable para que los datos mostrados no correspondan a un outlier.

```
#hacemos uso del metodo de cuartiles
y=filtro2

percentile25=y.quantile(0.25) #Q1
percentile75=y.quantile(0.75) #Q3
iqr= percentile75 - percentile25

Limite_Superior_iqr= percentile75 + 1.5*iqr
Limite_Inferior_iqr= percentile25 - 1.5*iqr
print("Limite superior permitido", Limite_Superior_iqr)
print("Limite inferior permitido", Limite_Inferior_iqr)
```

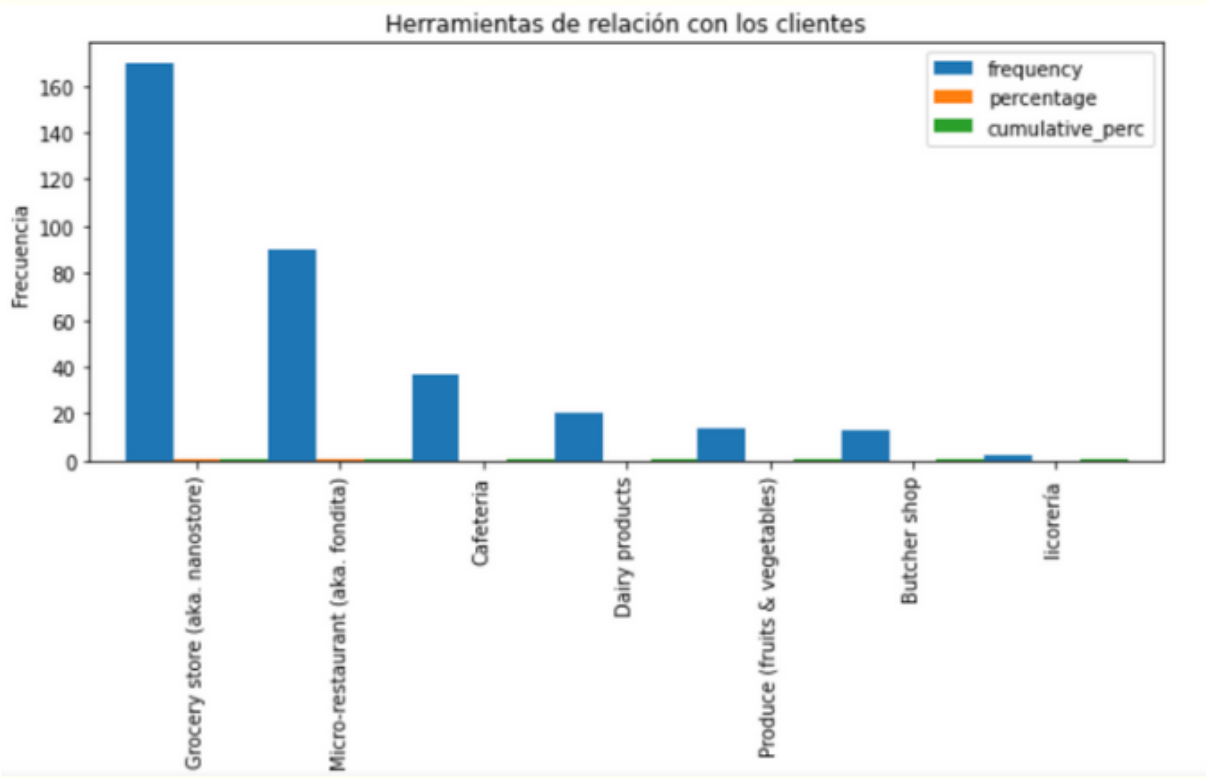
```
[12] #mostramos los limites
     outliers_iqr= filtro2[(y<=Limite_Superior_iqr)&(y>=Limite_Inferior_iqr)]
     outliers_iqr
```

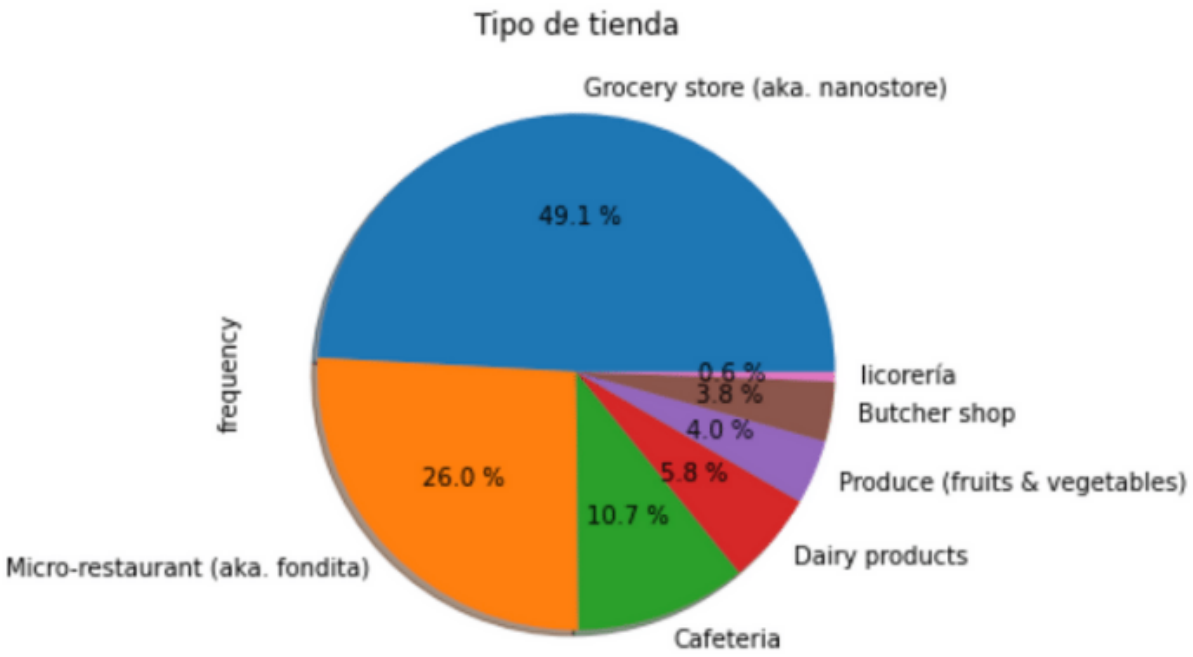
```
#Reemplazamos valores atipicos (nulos) del dataframe con "mean"
#Realizamos una copia del dataframe
Valores_finales= outliers_iqr.copy()
Valores_finales=Valores_finales.fillna(round(outliers_iqr.mean(),1))
Valores_finales
```

Concatenamos los valores limpios para convertirlos a un nuevo archivo csv y poder descargarlo.

```
[14] #vemos los valores nulos
     valores_nulos=Valores_finales.isnull().sum()
     valores_nulos

[15] #concatemos
     Lastmile_customer_limpios = pd.concat([filtro1, filtro2, filtro3], axis=1)
     Lastmile_customer_limpios

     Lastmile_customer_limpios.info()

[ ]  #Convertir DataFrame a CSV
     Lastmile_customer_limpios.to_csv("2_ consumer_mit_lift_lab.csv")

[ ]  #Descargar archivo filtrado en csv
     from google.colab import files
     files.download("2_ consumer_mit_lift_lab.csv")
```

Una vez limpios los datos, podemos extraerlos y realizar las gráficas correspondientes, gráficas de barras, histogramas, gráfica de pastel y gráficas de área.




Tipo de tienda
Grocery store (aka. nanostore)
49.1 %
frequency
0.6 %
licorería
3.8 %
Butcher shop
4.0 %
Produce (fruits & vegetables)
5.8 %
Dairy products
10.7 %
Cafeteria
26.0 %
Micro-restaurant (aka. fondita)