Lastmile Delivery Operations MIT lift lab Evidencia 2

EQUIPO 2

Para comenzar con el preprocesamiento de estos datos es importante exportar las librerías que utilizaremos para la limpieza de los datos en general para posteriormente remplazar los valores nulos y observar que todos los datos sean congruentes con ayuda de los Outliers .

```
#Importamos las librerias 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
```

Cargamos el archivo que fuimos llenando con la recolección de datos a lo largo de este proyecto, tales datos fueron recolectados con la ayuda de la app FULCROM.

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

```
#Carga desde un archivo excel sin indice
data= pd.read_excel('lastmile_delivery_operations_mit_lift_lab.xlsx')
```

Una vez cargado el archivo xlxs procedemos rectificar con el código mostrado a continuación la cantidad de valores nulos es cada una de las variables mostradas.

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

```
_record_id                                    0
_title                                       10
_server_updated_at                            0
_created_by                                   0
_updated_by                                   0
_geometry                                     0
_latitude                                     0
_longitude                                    0
arrival_of_the_freight_vehicle                0
plates                                        1
company_if_visible                          155
visit_purpose                                 0
type_of_vehicle                               0
number_of_operators                           0
refrigerated_truck                            0
type_of_cargo                                 1
picture_of_the_parked_freight_vehicle         0
departure_of_the_freight_vehicle              0
where_was_the_vehicle_parked                  0
while_parked_was_the_engine_running           0
used_traffic_cone                             0
vehicles_unloading_door                       0
number_of_available_trolleys                  0
serving_customer                            292
garage_blocking                               0
accident                                      0
describe_the_accident                       390
noise                                         0
traffic_congestion                            0
maximum_number_of_vehicles_in_the_traffic_jam 384
dtype: int64
```

Comenzaremos con la limpieza de los datos del archivo descartando las columnas que después de un previo análisis se consideran innecesarias.

```
#Eliminar columnas innecesarias
data2=data.drop(["_record_id", "picture_of_the_parked_freight_vehicle","serving_customer","describe_the_accident","maximum_number_of_vehicles_in_the_t
data2.head()
```

Usaremos un filtro para poder quedarnos con las variables cuantitativas.

```
#Filtro por columnas, de acuerdo a la información obtenida arriba obtenemos las cuantitativas
cuant=data2.iloc[ : , [12,20]]  #columnas no consecutivas
cuant
```

Aplicaremos una segunda limpieza de columnas, pero ahora específicamente para las que fueron filtradas como variables cuantitativas.

```
#Eliminar columnas innecesarias para crear el subdf de las cualitativas
cualit=data2.drop(["_latitude", "_latitude","number_of_operators","number_of_available_trolleys"], axis=1) #axis1=columnas y axis0=filas
cualit
```

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

Rectificamos las variables que aun tienen valores nulos para poder volver a filtrar o aplicar un método de sustitución de valores nulos.

```
_title                                10
_server_updated_at                     0
_created_by                            0
_updated_by                            0
_geometry                              0
_longitude                             0
arrival_of_the_freight_vehicle         0
plates                                 1
company_if_visible                   155
visit_purpose                          0
type_of_vehicle                        0
refrigerated_truck                     0
type_of_cargo                          1
departure_of_the_freight_vehicle       0
where_was_the_vehicle_parked           0
while_parked_was_the_engine_running    0
used_traffic_cone                      0
vehicles_unloading_door                0
garage_blocking                        0
accident                               0
noise                                  0
traffic_congestion                     0
dtype: int64
```

Ya que observamos cuales son las variables que aun tienen nulos aplicaremos el comando ffill y bfill para poder replicar los datos que tenemos arriba y replicarlos abajo en un caso y los que tenemos abajo replicarlos arriba en otro caso, posterior a esto nos quedaría una variable que tiene muchos valores nulos como lo es la variable compañía visible debido a que hay una gran variedad de compañías optamos por mejor crear una nueva categoría dentro de esa variable remplazando los nulos por un dato llamado compañía desconocida, esto con la ayuda del comando cualit .

```
#Sustituir valores nulos por valores no nulos hacia adelante "forward fill"("ffill")
cualit["_title"]=cualit["_title"].fillna(method="ffill")
cualit["plates"]=cualit["plates"].fillna(method="bfill")
cualit["type_of_cargo"]=cualit["type_of_cargo"].fillna(method="bfill")
#Sustituir valores nulos por un string en concreto
cualit["company_if_visible"]=cualit["company_if_visible"].fillna("Sin registro")
cualit
```

```
valores_nulos=cualit.isnull().sum()
valores_nulos
```

Una vez finalizada la sustitución de nulos rectificamos cuantos son los que quedan y podemos observar que ya no queda ninguna variable con valores nulos.

```
_title                                0
_server_updated_at                    0
_created_by                           0
_updated_by                           0
_geometry                             0
_longitude                            0
arrival_of_the_freight_vehicle        0
plates                                0
company_if_visible                    0
visit_purpose                         0
type_of_vehicle                       0
refrigerated_truck                    0
type_of_cargo                         0
departure_of_the_freight_vehicle      0
where_was_the_vehicle_parked          0
while_parked_was_the_engine_running   0
used_traffic_cone                     0
vehicles_unloading_door               0
garage_blocking                       0
accident                              0
noise                                 0
traffic_congestion                    0
dtype: int64
```

Aplicamos desviación estándar para observar como se comportan nuestros datos.

```
#Método aplicando desviación estandar. Encuentro los valores extremos
y=cuant
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)
```

```
Limite superior permitido number_of_operators            4.009477
number_of_available_trolleys    2.962660
dtype: float64
Limite inferior permitido number_of_operators           -0.753067
number_of_available_trolleys   -1.767788
dtype: float64
```

Posterior a esto sacamos Outliers para ver qué tan congruentes son los datos sustituidos que antes eran nulos en el dataframe.

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

	number_of_operators	number_of_available_trolleys
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN
...	...	...
385	NaN	NaN
386	NaN	NaN
387	NaN	NaN
388	NaN	NaN
389	NaN	NaN

Ya que al hacer la operación nos da un numero demasiado elevado, casi el total de entradas, no se hará el procedimiento de Outliers.

```
valores_nulos=outliers.isnull().sum()
valores_nulos
```

```
number_of_operators            387
number_of_available_trolleys   386
dtype: int64
```

Corroboramos si ya no tenemos ningún valor nulo en nuestro data frame

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

```
0
```

Convertimos el DataFrame a un archivo CSV y posteriormente lo preparamos para la descarga

```
#Convertir DataFrame a CSV
LMLEvidencia.to_csv("LMLEvidencia.csv")
```

```
#descargar archivo filtrado en csv
from google.colab import files
files.download("LMLEvidencia.csv")
```