Tecnológico de Monterrey

Reporte preparación de la base de datos (Lastmile_Delivery)

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

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

```
[2] #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()
```

```
[4] #leemos el archivo
    data1 = pd.read_excel("2_ lastmile_delivery_operations_mit_lift_lab.xlsx")
    data1
```

El primer paso para comenzar con el código es importar e instalar las librerías necesarias en el código a posterior, después de eso se necesita subir el archivo y hacer que el código lo lea.

```
#vemos su informacion
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 390 entries, 0 to 389
Data columns (total 30 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   _record_id                              390 non-null    object
 1   _title                                  380 non-null    object
 2   _server_updated_at                      390 non-null    object
 3   _created_by                             390 non-null    object
 4   _updated_by                             390 non-null    object
 5   _geometry                               390 non-null    object
 6   _latitude                               390 non-null    float64
 7   _longitude                              390 non-null    float64
 8   arrival_of_the_freight_vehicle          390 non-null    object
 9   plates                                  389 non-null    object
 10  company_if_visible                      235 non-null    object
 11  visit_purpose                           390 non-null    object
 12  type_of_vehicle                         390 non-null    object
 13  number_of_operators                     390 non-null    int64
 14  refrigerated_truck                      390 non-null    object
 15  type_of_cargo                           389 non-null    object
 16  picture_of_the_parked_freight_vehicle   390 non-null    object
 17  departure_of_the_freight_vehicle        390 non-null    object
 18  where_was_the_vehicle_parked            390 non-null    object
 19  while_parked_was_the_engine_running     390 non-null    object
 20  used_traffic_cone                       390 non-null    object
 21  vehicles_unloading_door                 390 non-null    object
 22  number_of_available_trolleys            390 non-null    int64
 23  serving_customer                        98 non-null     float64
 24  garage_blocking                         390 non-null    object
 25  accident                                390 non-null    object
 26  describe_the_accident                   0 non-null      float64
 27  noise                                   390 non-null    object
 28  traffic_congestion                      390 non-null    object
 29  maximum_number_of_vehicles_in_the_traffic_jam  6 non-null      float64
dtypes: float64(5), int64(2), object(23)
```

```
#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
```

	number_of_operators	number_of_available_trolleys	_latitude	_longitude
0	1.0	0.0	19.021376	-98.260392
1	1.0	0.0	19.021111	-98.260568
2	2.0	0.0	19.035221	-98.267035
3	2.0	0.0	19.081790	-98.298561
4	1.0	0.0	19.081771	-98.298589
...	...	...	...	...
385	1.0	0.0	19.043612	-98.194933
386	1.0	0.0	19.043607	-98.194856
387	1.0	1.0	19.043336	-98.194503
388	2.0	0.0	19.043576	-98.194873
389	2.0	1.0	19.043590	-98.194901

390 rows × 4 columns

```
#Verificamos los valores nulos
valores_nulos=data1.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
```

```
[7] #reemplazamos los valores nulos con la palabra nulo
data=data1.copy()
data["_title"] =data["_title"].fillna("nulo")
data["company_if_visible"] =data["company_if_visible"].fillna("nulo")
data["plates"] =data["plates"].fillna("nulo")
data["type_of_cargo"] =data["type_of_cargo"].fillna("nulo")
data["serving_customer"] =data["serving_customer"].fillna("nulo")
data["describe_the_accident"] =data["describe_the_accident"].fillna("nulo")
data["maximum_number_of_vehicles_in_the_traffic_jam"] =data["maximum_number_of_vehicles_in_the_traffic_jam"].fillna("nulo")
```

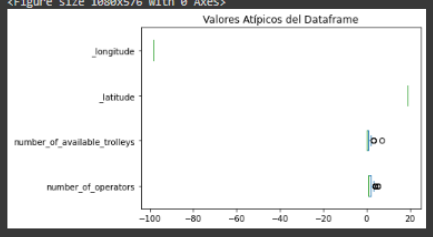
Posteriormente, se ve la información de las columnas del archivo, esto para poder ver cuántas columnas se tienen, sus nombres y el tipo de dato que se maneja en cada una de estas columnas, también observamos la cantidad de nulos que hay

en cada columna, y los nulos cuyos datos sean de tipo objeto (textos), se les sustituyen esos nulos por la palabra nulo.

```
[64] #aplicamos filtro
     filtro1=data.iloc[ : , [0,1,2,3,4,5,8,9,10,11,12,14,15,16,17,18,19,20,21,24,25,27,28]] #obj
     filtro2=data.iloc[ : , [13,22,6,7]]    #valores numericos, sin la columna 23, 26 y 29 ya que todos o casi todos son nulos y por lo tanto no las usaremos
```

```
#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 1000x576 with 0 Axes>
```




```
[66] #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)
```

```
Limite superior permitido number_of_operators             3.500000
number_of_available_trolleys     2.500000
_latitude                       19.101536
_longitude                     -98.146671
dtype: float64
Limite inferior permitido number_of_operators            -0.500000
number_of_available_trolleys    -1.500000
_latitude                       18.970004
_longitude                     -98.330490
dtype: float64
```

Lo siguiente fue realizar un filtrado de datos, en los cuales separamos los datos en objetos y numéricos (float y enteros), además de eliminar tres columnas ya que en

ellas la mayoría de los datos eran valores nulos.

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

	number_of_operators	number_of_available_trolleys	_latitude	_longitude
0	1.0	0.0	19.021376	-98.260392
1	1.0	0.0	19.021111	-98.260568
2	2.0	0.0	19.035221	-98.267035
3	2.0	0.0	19.081790	-98.298561
4	1.0	0.0	19.081771	-98.298589
...	...	...	...	...
385	1.0	0.0	19.043612	-98.194933
386	1.0	0.0	19.043607	-98.194856
387	1.0	1.0	19.043336	-98.194503
388	2.0	0.0	19.043576	-98.194873
389	2.0	1.0	19.043590	-98.194901

390 rows × 4 columns

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

```
number_of_operators            0
number_of_available_trolleys   0
_latitude                      0
_longitude                     0
dtype: int64
```

```
#concatenamos
Lastmile_customer_limpios = pd.concat([filtro1, filtro2], axis=1)
Lastmile_customer_limpios
```

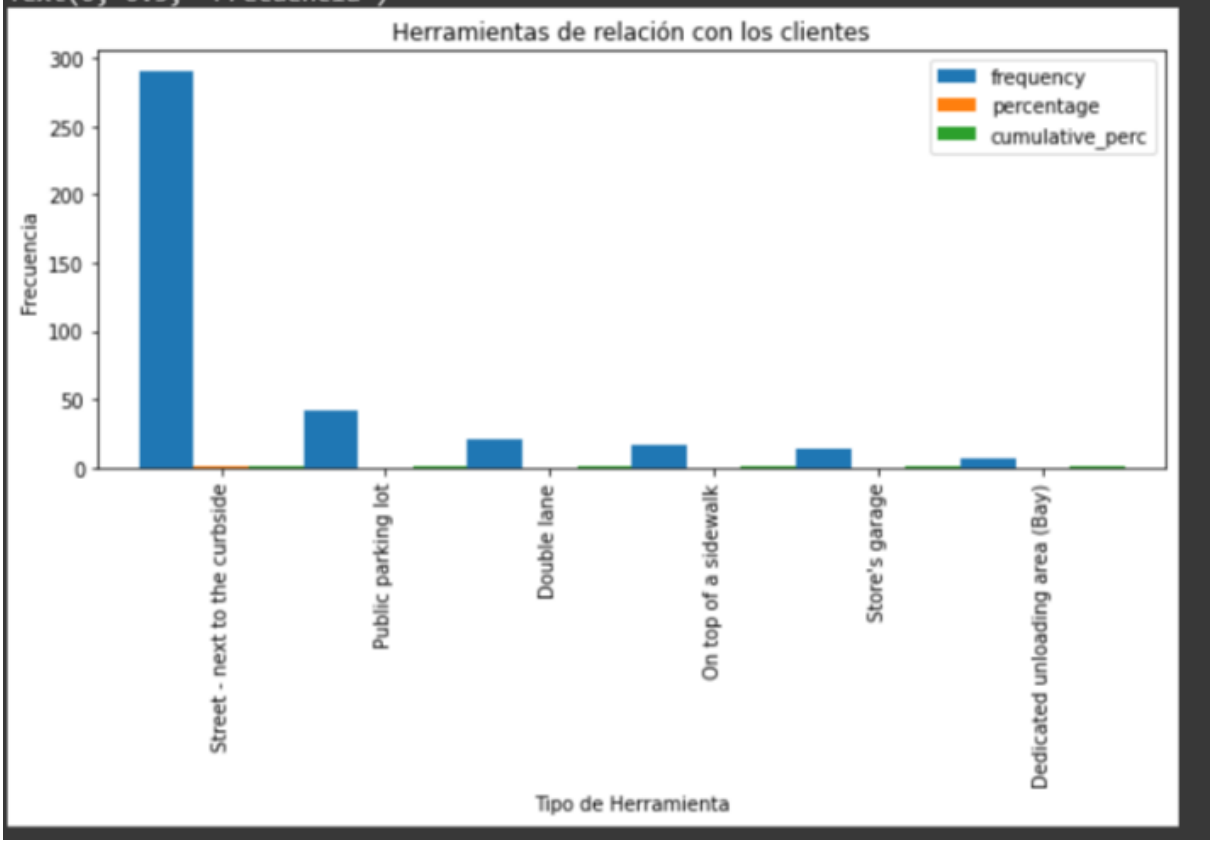
Después procedimos a identificar los valores nulos y una vez que los identificamos los cambiamos por la palabra nulo y concatenamos todos los filtros que teníamos para hacer un solo data frame que contenga todo.

```
#Obtengo un análisis univariado de las variables categóricas
freq_tbl(Lastmile_customer_limpios)

#Obtengo un análisis univariado de una variable categórica en específico
table1= freq_tbl(Lastmile_customer_limpios['where_was_the_vehicle_parked'])
table1

#Obtengo un filtro de los valores más reelevantes de la variables categórica seleccionada
Filtro= table1[table1['frequency']>1]
Filtro

#Ajusto el indice de mi dataframe
Filtro_index= Filtro.set_index('where_was_the_vehicle_parked')
Filtro_index
```

Lo primero es hacer una tabla de la frecuencia de cada columna, y posteriormente hacer solo una tabla que contenga la frecuencia de una sola columna.

```python
#Realizamos grafico de barras del dataframe filtrado
Filtro_index.plot(kind = 'bar', width=1, figsize=(10,4))
plt.title('Herramientas de relación con los clientes')
plt.xlabel('Tipo de Herramienta')
plt.ylabel('Frecuencia')
```

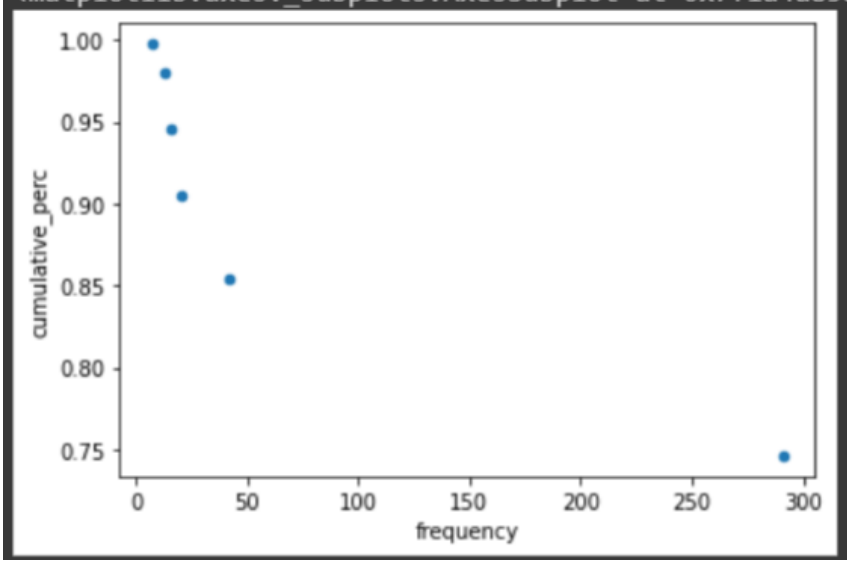
```
Text(0, 0.5, 'Frecuencia')
```



Procedimos a obtener la gráfica de frecuencia vs el cumulative perc

```
#Realizamos grafico de dispersión del dataframe filtrado
Filtro_index.plot("frequency", "cumulative_perc", kind="scatter")
```

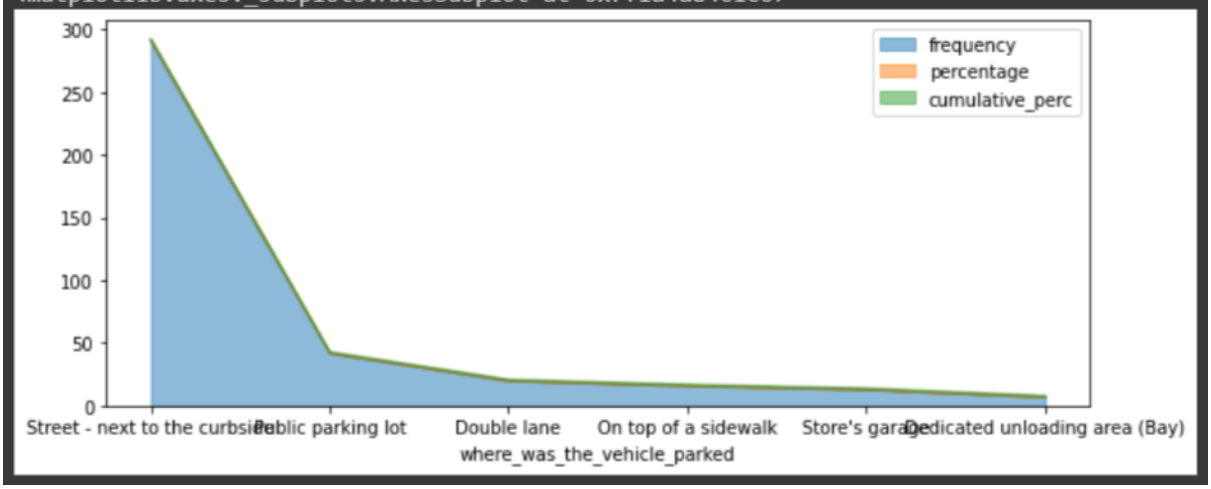
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a4a858520>
```



Después realizamos el gráfico del dataframe ya filtrado y obtuvimos la gráfica que se muestra en la parte inferior la cual nos muestra dónde es que el vehículo se estacionaba con mayor frecuencia, con estos datos era posible predecir dónde es que el vehículo se estacionaria cuando éste viniera dejar la mercancía

```
#Realizamos grafico de área del dataframe filtrado
Filtro_index.plot(kind='area', figsize=(10,4),alpha = 0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a4a84c1c0>
```



Luego obtenemos la gráfica de pastel del dataframe ya filtrado en la que se nos muestra dónde es que se estacionaba el vehículo repartidor y con que frecuencia lo hacía

```
#Realizamos grafico de pastel del dataframe filtrado
Filtro_index["frequency"].plot(kind='pie', figsize=(10,5), shadow=True, autopct="%0.1f %%")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1a4a766070>
```