```
import pandas as pd
import os

!pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (1.26.4)
Requirement already satisfied: pandas>=1.2 in c:\users\jaime\appdata\
local\programs\python\python312\lib\site-packages (from seaborn)
(2.2.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\
jaime\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.2.1)
Requirement already satisfied: cycler>=0.10 in c:\users\jaime\appdata\
local\programs\python\python312\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\jaime\
appdata\roaming\python\python312\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (24.0)
Requirement already satisfied: pillow>=8 in c:\users\jaime\appdata\
local\programs\python\python312\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\jaime\appdata\
local\programs\python\python312\lib\site-packages (from pandas>=1.2-
>seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in c:\users\jaime\appdata\
local\programs\python\python312\lib\site-packages (from python-
dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
```

```
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   - -------------------------------------- 10.2/294.9 kB ? eta
-:--:--
   ----- ---------------------------------- 41.0/294.9 kB 653.6 kB/s
eta 0:00:01
   ---------------------------------------- 294.9/294.9 kB 3.0 MB/s
eta 0:00:00
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2


[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Importamos las librerias necesarias

```
# Importar librerías
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os
#import missingno as msno
import math
import seaborn as sns
#import statsmodels.api as sm
```

Localizamos donde estan los archivos por medio de la libreria os para mayor robustez del proceso

```
current_directory = os.getcwd()
print(current_directory)

parent_directory=os.path.split(current_directory)[0]
parent_directory
```

```
c:\Users\jaime\OneDrive\Documentos\trabajo\dataknow\prueba_tecnica\
punto6
```

```
'c:\\Users\\jaime\\OneDrive\\Documentos\\trabajo\\dataknow\\
prueba_tecnica'
```

Asignamos la ruta de los archivos y los cargamos.

```
train_file =
os.path.join(parent_directory,'Prueba_Tecnica','Datos3' ,'train.csv')
test_file =
os.path.join(parent_directory,'Prueba_Tecnica','Datos3' ,'test.csv')
```

```
# Cargar los datos
train_data = pd.read_csv(train_file)
test_data = pd.read_csv(test_file)

pd.set_option('display.max_columns',None)
train_data
```

```
               id  FRAUDE      VALOR  HORA_AUX  Dist_max_NAL
Canal1  \
0     9000000001       1       0.00        13        659.13  ATM_INT

1     9000000002       1       0.00        17        594.77  ATM_INT

2     9000000003       1       0.00        13        659.13  ATM_INT

3     9000000004       1       0.00        13        659.13  ATM_INT

4     9000000005       1       0.00         0          1.00  ATM_INT

...          ...     ...        ...       ...           ...      ...

2960   622529101       1  993430.04        19        180.65      POS

2961  2043206272       0    9957.05        10        318.50      POS

2962  2943206272       0    9957.05        10        318.50      POS

2963  3136302872       0  996191.64        15        234.42      POS

2964  1953178702       1  999276.60        16          1.00  ATM_INT


         FECHA COD_PAIS    CANAL  DIASEM  DIAMES   FECHA_VIN
OFICINA_VIN  \
0     20150501       US  ATM_INT       5       1  20120306.0
392.0
1     20150515       US  ATM_INT       5      15  20050415.0
716.0
2     20150501       US  ATM_INT       5       1  20120306.0
392.0
3     20150501       US  ATM_INT       5       1  20120306.0
392.0
4     20150510       CR  ATM_INT       0      10  20141009.0
788.0
...        ...      ...      ...     ...     ...         ...
...
2960  20150519       US      POS       2      19  19740401.0
442.0
2961  20150524       US      POS       0      24  19970616.0
611.0
2962  20150524       US      POS       0      24  19970616.0
```

```
611.0
2963  20150513       US      MCI       3      13  20000609.0
534.0
2964  20150520       CR  ATM_INT       3      20  20050630.0
661.0

     SEXO       SEGMENTO  EDAD     INGRESOS     EGRESOS  NROPAISES  \
0       M  Personal Plus  29.0    1200000.0   1200000.0          1
1       M  Personal Plus  29.0    5643700.0    500000.0          1
2       M  Personal Plus  29.0    1200000.0   1200000.0          1
3       M  Personal Plus  29.0    1200000.0   1200000.0          1
4       M       Personal  25.0          0.0         0.0          1
...   ...            ...   ...          ...         ...        ...
2960    F   Preferencial  48.0  103918285.0  95475378.0          4
2961    F   Preferencial  35.0   23625000.0   5000000.0          3
2962    F   Preferencial  35.0   23625000.0   5000000.0          3
2963    F           PYME  34.0   56666000.0  37600750.0          1
2964    F  Personal Plus  29.0   12853000.0   6156000.0          1

      Dist_Sum_INTER  Dist_Mean_INTER  Dist_Max_INTER  NROCIUDADES  \
0                NaN              NaN             NaN            6
1                NaN              NaN             NaN            5
2                NaN              NaN             NaN            6
3                NaN              NaN             NaN            6
4                NaN              NaN             NaN            1
...              ...              ...             ...          ...
2960         8944.83          2236.21         3646.67            4
2961        27648.32          3949.76         4552.41           11
2962        27648.32          3949.76         4552.41           11
2963             NaN              NaN             NaN            3
2964             NaN              NaN             NaN            1

      Dist_Mean_NAL  Dist_HOY  Dist_sum_NAL
0            474.94   4552.41       5224.36
1            289.99   4552.41       2029.90
2            474.94   4552.41       5224.36
3            474.94   4552.41       5224.36
4               NaN   1482.35          1.00
...             ...       ...           ...
2960          96.86   4552.41        484.30
2961          82.67   4552.41       2810.75
2962          82.67   4552.41       2810.75
2963         219.46   4552.41       1316.79
2964            NaN   1482.35          1.00

[2965 rows x 26 columns]
```

Observamos el numero de filas no nulas mediante la función info().

```python
# Ver información de los datos
print(train_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2965 entries, 0 to 2964
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   id               2965 non-null   int64  
 1   FRAUDE           2965 non-null   int64  
 2   VALOR            2965 non-null   float64
 3   HORA_AUX         2965 non-null   int64  
 4   Dist_max_NAL     2965 non-null   float64
 5   Canal1           2965 non-null   object 
 6   FECHA            2965 non-null   int64  
 7   COD_PAIS         2965 non-null   object 
 8   CANAL            2965 non-null   object 
 9   DIASEM           2965 non-null   int64  
 10  DIAMES           2965 non-null   int64  
 11  FECHA_VIN        2941 non-null   float64
 12  OFICINA_VIN      2941 non-null   float64
 13  SEXO             2910 non-null   object 
 14  SEGMENTO         2941 non-null   object 
 15  EDAD             2941 non-null   float64
 16  INGRESOS         2941 non-null   float64
 17  EGRESOS          2941 non-null   float64
 18  NROPAISES        2965 non-null   int64  
 19  Dist_Sum_INTER   1418 non-null   float64
 20  Dist_Mean_INTER  1418 non-null   float64
 21  Dist_Max_INTER   1418 non-null   float64
 22  NROCIUDADES      2965 non-null   int64  
 23  Dist_Mean_NAL    2508 non-null   float64
 24  Dist_HOY         2965 non-null   float64
 25  Dist_sum_NAL     2965 non-null   float64
dtypes: float64(13), int64(8), object(5)
memory usage: 602.4+ KB
None
```

Establecemos la cantidad de nulos por cada columna y que porcentaje representan

```python
#Conteo de la cantidad de nulos por cada columna
regs=train_data.shape[0]
df_transform=train_data.copy()
for col in train_data.columns:
    vals=train_data[col].value_counts()
    values=vals.shape[0]
    nas=train_data[train_data[col].isna()].shape[0]
    porcentaje_na=nas*100/regs
    if porcentaje_na>=10:
```

```
        text=f'elimienando columna: {col}'
        df_transform.drop(columns={col},inplace=True)
    else:
        text=f'Quitando nulos de {col}'
        df_transform.dropna(subset=col,inplace=True)

    print(f'-variable {col} tiene {values} valores único, {nas}
valores nulos ({nas*100/regs:.2f})%.')
```

```
-variable id tiene 2888 valores único, 0 valores nulos (0.00)%.
-variable FRAUDE tiene 2 valores único, 0 valores nulos (0.00)%.
-variable VALOR tiene 2259 valores único, 0 valores nulos (0.00)%.
-variable HORA_AUX tiene 24 valores único, 0 valores nulos (0.00)%.
-variable Dist_max_NAL tiene 264 valores único, 0 valores nulos
(0.00)%.
-variable Canal1 tiene 2 valores único, 0 valores nulos (0.00)%.
-variable FECHA tiene 31 valores único, 0 valores nulos (0.00)%.
-variable COD_PAIS tiene 29 valores único, 0 valores nulos (0.00)%.
-variable CANAL tiene 3 valores único, 0 valores nulos (0.00)%.
-variable DIASEM tiene 7 valores único, 0 valores nulos (0.00)%.
-variable DIAMES tiene 31 valores único, 0 valores nulos (0.00)%.
-variable FECHA_VIN tiene 1025 valores único, 0 valores nulos (0.00)%.
-variable OFICINA_VIN tiene 373 valores único, 0 valores nulos
(0.00)%.
-variable SEXO tiene 2 valores único, 0 valores nulos (0.00)%.
-variable SEGMENTO tiene 6 valores único, 0 valores nulos (0.00)%.
-variable EDAD tiene 62 valores único, 0 valores nulos (0.00)%.
-variable INGRESOS tiene 500 valores único, 0 valores nulos (0.00)%.
-variable EGRESOS tiene 193 valores único, 0 valores nulos (0.00)%.
-variable NROPAISES tiene 8 valores único, 0 valores nulos (0.00)%.
-variable Dist_Sum_INTER tiene 209 valores único, 0 valores nulos
(0.00)%.
-variable Dist_Mean_INTER tiene 182 valores único, 0 valores nulos
(0.00)%.
-variable Dist_Max_INTER tiene 62 valores único, 0 valores nulos
(0.00)%.
-variable NROCIUDADES tiene 18 valores único, 0 valores nulos (0.00)%.
-variable Dist_Mean_NAL tiene 772 valores único, 0 valores nulos
(0.00)%.
-variable Dist_HOY tiene 48 valores único, 0 valores nulos (0.00)%.
-variable Dist_sum_NAL tiene 841 valores único, 0 valores nulos
(0.00)%.
```

Observamos la cantidad de subniveles que hay por cada columna, para visualizar si hay algunas columna para descartar en el analisis

```
cols_cat = train_data.select_dtypes(include=['object'])
```

```
for col in cols_cat:
  print(f'Columna {col}: {train_data[col].nunique()} subniveles')
```

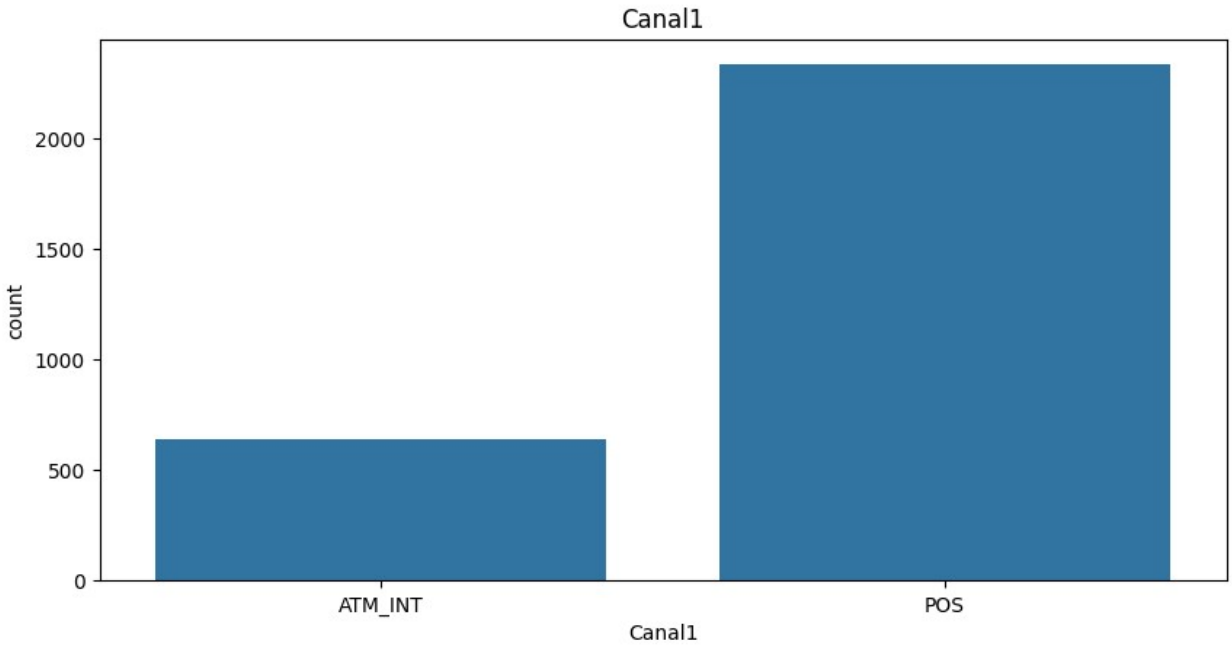
```
Columna Canal1: 2 subniveles
Columna COD_PAIS: 29 subniveles
Columna CANAL: 3 subniveles
Columna SEXO: 2 subniveles
Columna SEGMENTO: 6 subniveles
```

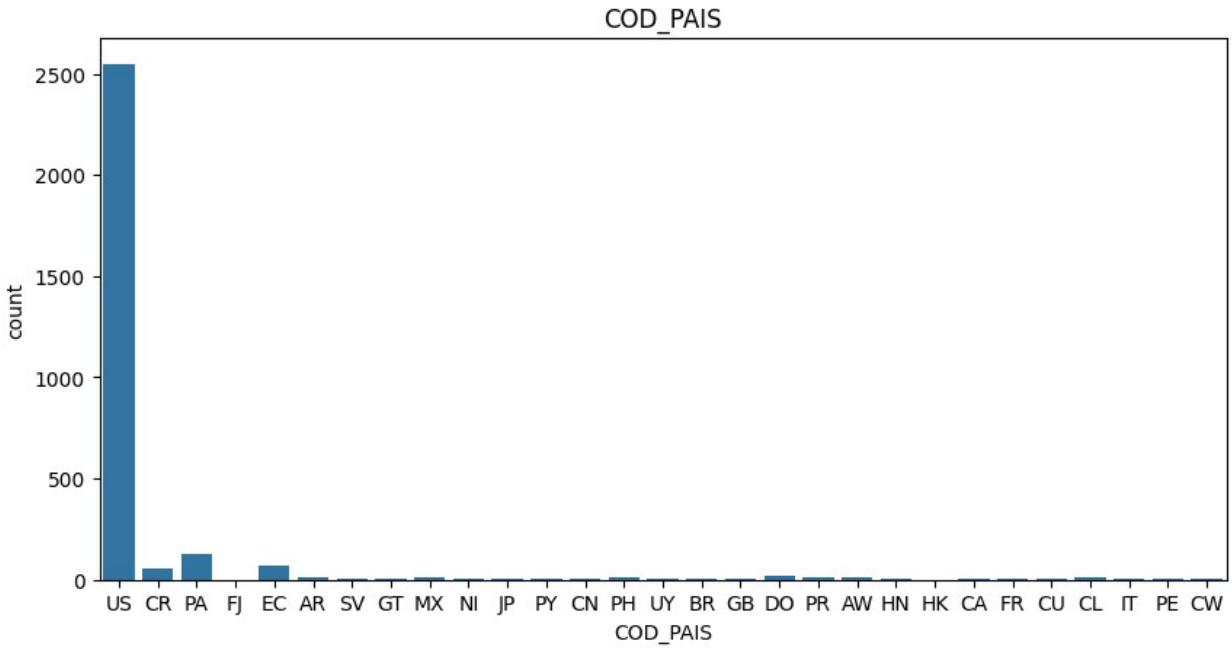
Grafico de barras de frencuencia

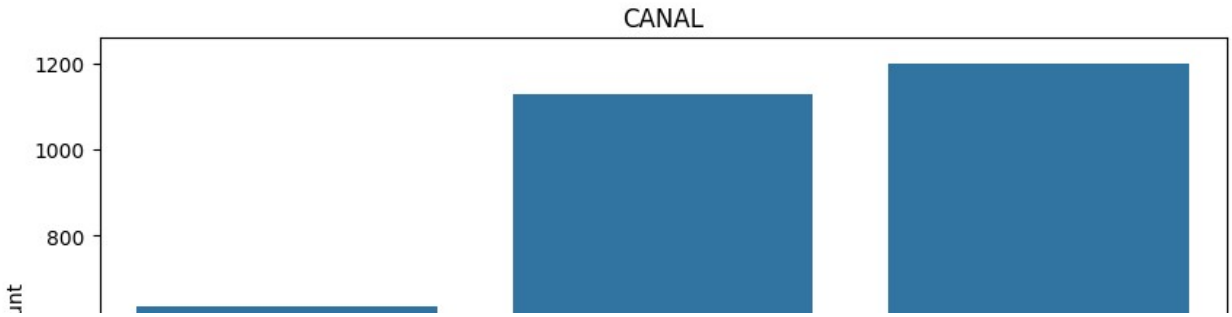
```
fig, ax = plt.subplots(nrows=5, ncols=1, figsize=(10,30))
fig.subplots_adjust(hspace=0.3)

for i, col in enumerate(cols_cat):
  sns.countplot(x=col, data=train_data, ax=ax[i])
  ax[i].set_title(col)
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
```

```
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\3464803981.py:7:
UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\3464803981.py:7:
UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\3464803981.py:7:
UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\3464803981.py:7:
UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\3464803981.py:7:
UserWarning: set_ticklabels() should only be used with a fixed number
of ticks, i.e. after set_ticks() or using a FixedLocator.
  ax[i].set_xticklabels(ax[i].get_xticklabels(),rotation=0)
```


Canal1
2000
1500
1000
500
0
count
ATM_INT
POS
Canal1


COD_PAIS
2500
2000
1500
1000
500
0
count
US CR PA FJ EC AR SV GT MX NI JP PY CN PH UY BR GB DO PR AW HN HK CA FR CU CL IT PE CW
COD_PAIS


CANAL
1200
1000
800

Descripción estadistica del dataframe train_data

```
train_data.describe()
```

```
                  id       FRAUDE          VALOR     HORA_AUX
Dist_max_NAL  \
count  2.965000e+03  2965.000000  2.965000e+03  2965.000000
2965.000000
mean   6.890938e+09     0.246543  5.035695e+05    14.960877
314.656739
std    9.739700e+09     0.431071  9.859497e+05     6.348607
295.142673
min    2.364560e+06     0.000000  0.000000e+00     0.000000
1.000000
25%    2.552997e+09     0.000000  9.016001e+04    12.000000
24.830000
50%    6.142884e+09     0.000000  2.435912e+05    16.000000
243.620000
75%    9.000000e+09     0.000000  5.058190e+05    20.000000
594.770000
max    9.330050e+10     1.000000  2.001406e+07    23.000000
1310.460000

              FECHA       DIASEM       DIAMES     FECHA_VIN
OFICINA_VIN  \
count  2.965000e+03  2965.000000  2965.000000  2.941000e+03
2941.000000
mean   2.015051e+07     3.143002    13.492411  2.000920e+07
416.426386
std    9.134641e+00     2.092284     9.134641  9.260427e+04
290.263484
min    2.015050e+07     0.000000     1.000000  1.911111e+07
1.000000
25%    2.015050e+07     1.000000     4.000000  1.995102e+07
168.000000
50%    2.015052e+07     3.000000    15.000000  2.001123e+07
360.000000
75%    2.015052e+07     5.000000    21.000000  2.008081e+07
659.000000
max    2.015053e+07     6.000000    31.000000  2.015043e+07
967.000000

              EDAD      INGRESOS       EGRESOS    NROPAISES
Dist_Sum_INTER  \
count  2941.000000  2.941000e+03  2.941000e+03  2965.000000
1418.000000
mean     40.010541  1.449104e+07  8.506309e+06     1.765936
17354.692891
std      12.976492  5.637311e+07  6.179161e+07     1.042219
25168.020649
```

```
min       0.000000  0.000000e+00  0.000000e+00     1.000000
904.810000
25%      31.000000  2.500000e+06  5.000000e+05     1.000000
6474.200000
50%      38.000000  5.800000e+06  1.800000e+06     1.000000
9104.820000
75%      47.000000  1.274000e+07  4.500000e+06     2.000000
21376.445000
max     133.000000  1.940070e+09  1.600000e+09     9.000000
758837.940000

       Dist_Mean_INTER  Dist_Max_INTER  NROCIUDADES  Dist_Mean_NAL  \
count      1418.000000     1418.000000  2965.000000    2508.000000
mean       4144.323540     4985.442313     3.943676     196.589282
std        1794.829357     2655.081718     2.750021     192.026206
min         904.810000      904.810000     1.000000       4.480000
25%        3178.210000     4552.410000     2.000000      60.800000
50%        4552.410000     4552.410000     3.000000     127.700000
75%        4552.410000     4552.410000     5.000000     269.082500
max       16328.810000    17780.330000    20.000000    1217.570000

           Dist_HOY  Dist_sum_NAL
count   2965.000000   2965.000000
mean    4379.826287   1765.212887
std     1779.739070   2398.666844
min        0.000000      1.000000
25%     4552.410000    139.870000
50%     4552.410000    836.080000
75%     4552.410000   2533.440000
max    21991.200000  18832.060000
```

Busqueda de outliers por medio de diagrama de caja

```
train_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2965 entries, 0 to 2964
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               2965 non-null   int64
 1   FRAUDE           2965 non-null   int64
 2   VALOR            2965 non-null   float64
 3   HORA_AUX         2965 non-null   int64
 4   Dist_max_NAL     2965 non-null   float64
 5   Canal1           2965 non-null   object
 6   FECHA            2965 non-null   int64
 7   COD_PAIS         2965 non-null   object
 8   CANAL            2965 non-null   object
```

```
 9   DIASEM           2965 non-null   int64
 10  DIAMES           2965 non-null   int64
 11  FECHA_VIN        2941 non-null   float64
 12  OFICINA_VIN      2941 non-null   float64
 13  SEXO             2910 non-null   object
 14  SEGMENTO         2941 non-null   object
 15  EDAD             2941 non-null   float64
 16  INGRESOS         2941 non-null   float64
 17  EGRESOS          2941 non-null   float64
 18  NROPAISES        2965 non-null   int64
 19  Dist_Sum_INTER   1418 non-null   float64
 20  Dist_Mean_INTER  1418 non-null   float64
 21  Dist_Max_INTER   1418 non-null   float64
 22  NROCIUDADES      2965 non-null   int64
 23  Dist_Mean_NAL    2508 non-null   float64
 24  Dist_HOY         2965 non-null   float64
 25  Dist_sum_NAL     2965 non-null   float64
dtypes: float64(13), int64(8), object(5)
memory usage: 602.4+ KB
```

```
train_data.columns
```

```
Index(['id', 'FRAUDE', 'VALOR', 'HORA_AUX', 'Dist_max_NAL', 'Canal1',
'FECHA',
       'COD_PAIS', 'CANAL', 'DIASEM', 'DIAMES', 'FECHA_VIN',
'OFICINA_VIN',
       'SEXO', 'SEGMENTO', 'EDAD', 'INGRESOS', 'EGRESOS', 'NROPAISES',
       'Dist_Sum_INTER', 'Dist_Mean_INTER', 'Dist_Max_INTER',
'NROCIUDADES',
       'Dist_Mean_NAL', 'Dist_HOY', 'Dist_sum_NAL'],
      dtype='object')
```

```
# Crear el boxplot
plt.figure(figsize=(14, 40))  # Ajusta el tamaño de la gráfica

# Boxplot para la Columna1
plt.subplot(16, 1, 1)
sns.boxplot(x='FRAUDE', data=train_data)
plt.title("Fraude")

# Boxplot para la Columna2
plt.subplot(16, 1, 2)
sns.boxplot(x='VALOR', data=train_data)
plt.title("Valor")

# Boxplot para la Columna3
plt.subplot(16, 1, 3)
sns.boxplot(x='HORA_AUX', data=train_data)
plt.title("Hora de la transacción")
```

```
# Boxplot para la Columna4
plt.subplot(16, 1, 4)
sns.boxplot(x='Dist_max_NAL', data=train_data)
plt.title("Distancia maxima recorrida a nivel nacional")

# Boxplot para la Columna5
plt.subplot(16, 1, 5)
sns.boxplot(x='EDAD', data=train_data)
plt.title("Edad")

# Boxplot para la Columna6
plt.subplot(16, 1, 6)
sns.boxplot(x='INGRESOS', data=train_data)
plt.title("Ingresos")

# Boxplot para la Columna7
plt.subplot(16, 1, 7)
sns.boxplot(x='EGRESOS', data=train_data)
plt.title("Egresos")

# Boxplot para la Columna8
plt.subplot(16, 1, 8)
sns.boxplot(x='EGRESOS', data=train_data)
plt.title("Egresos")

# Boxplot para la Columna9
plt.subplot(16, 1, 9)
sns.boxplot(x='NROPAISES', data=train_data)
plt.title("Numero de paises visitados")

# Boxplot para la Columna10
plt.subplot(16, 1, 10)
sns.boxplot(x='Dist_Sum_INTER', data=train_data)
plt.title("Sumatoria de distancia recorrida a nivel internacional (en
millas)")

# Boxplot para la Columna11
plt.subplot(16, 1, 11)
sns.boxplot(x='Dist_Mean_INTER', data=train_data)
plt.title("Promedio de distancia recorrida a nivel internacional (en
millas)")

# Boxplot para la Columna12
plt.subplot(16, 1, 12)
sns.boxplot(x='Dist_Max_INTER', data=train_data)
plt.title("Distancia maxima recorrida a nivel internacional (en
millas)")

# Boxplot para la Columna13
plt.subplot(16, 1, 13)
```

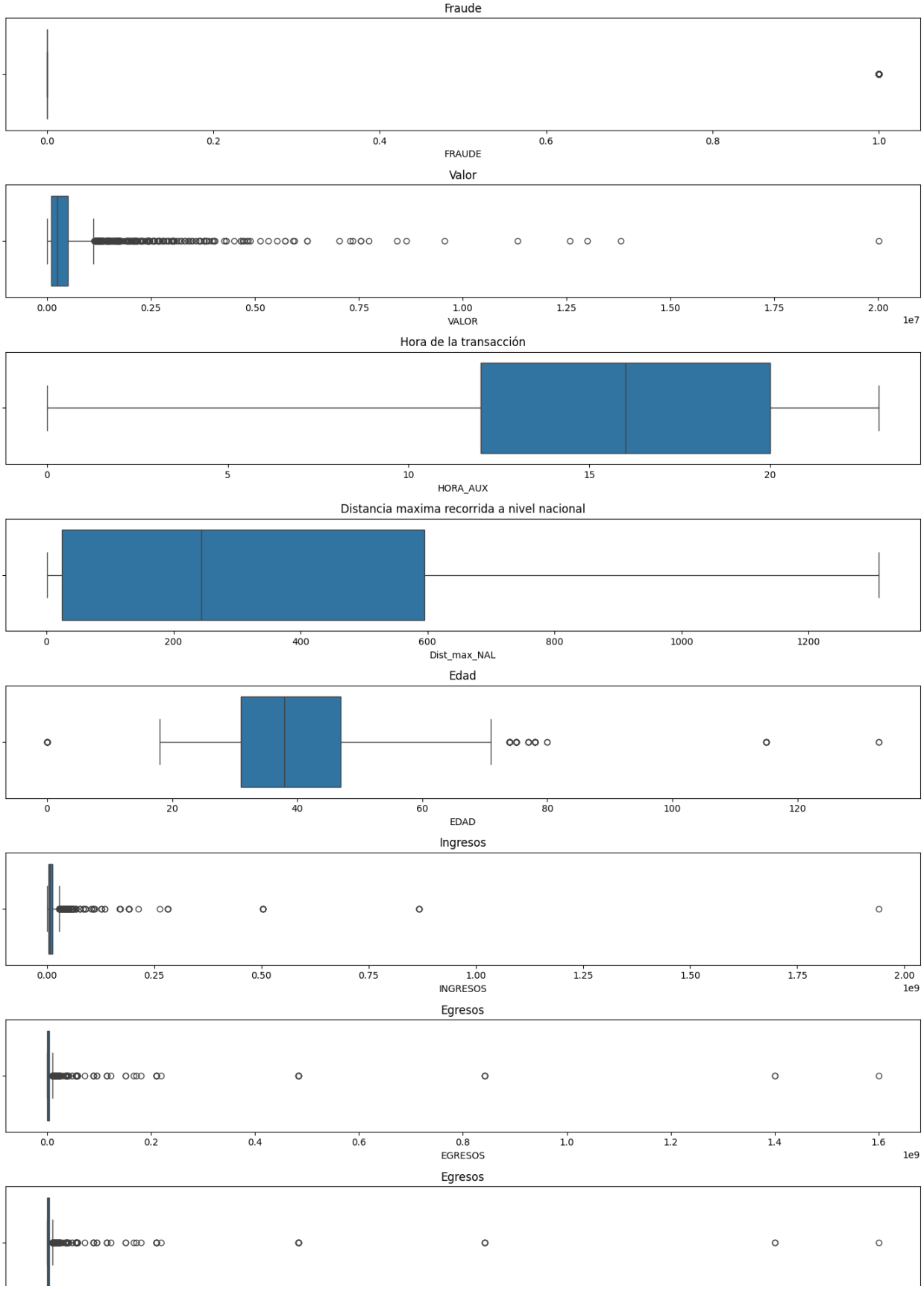
```
sns.boxplot(x='NROCIUDADES', data=train_data)
plt.title("Numero de ciudades nacionales visitadas")

# Boxplot para la Columna14
plt.subplot(16, 1, 14)
sns.boxplot(x='Dist_Mean_NAL', data=train_data)
plt.title("Distancia máxima recorrida a nivel nacional (en millas)")

# Boxplot para la Columna15
plt.subplot(16, 1, 15)
sns.boxplot(x='Dist_HOY', data=train_data)
plt.title("Diferencia entre la ultima transacción presente realizada y
la transacción que esta realizando el dia de hoy")

# Boxplot para la Columna16
plt.subplot(16, 1, 16)
sns.boxplot(x='Dist_sum_NAL', data=train_data)
plt.title("Sumatoria de distacia recorrida a nivel nacional (en
millas)")

# Ajustar el diseño y mostrar la gráfica
plt.tight_layout()
plt.show()
```


Fraude
0.0
0.2
0.4
0.6
0.8
1.0
FRAUDE
Valor
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
VALOR
1e7
Hora de la transacción
0
5
10
15
20
HORA_AUX
Distancia maxima recorrida a nivel nacional
0
200
400
600
800
1000
1200
Dist_max_NAL
Edad
0
20
40
60
80
100
120
EDAD
Ingresos
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
INGRESOS
1e9
Egresos
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
EGRESOS
1e9
Egresos

Eliminacion e imputacion de valores segun el porcentaje de nulos y la relevancia de la variable para el analisis

```
train_data = train_data.dropna(subset=['FECHA_VIN', 'OFICINA_VIN',
'SEGMENTO', 'INGRESOS', 'EGRESOS'])
train_data['EDAD'] =
train_data['EDAD'].fillna(train_data['EDAD'].median())
train_data['SEXO'] =
train_data['SEXO'].fillna(train_data['SEXO'].mode()[0])
train_data['Dist_Sum_INTER'] = train_data['Dist_Sum_INTER'].fillna(0)
train_data['Dist_Mean_INTER'] =
train_data['Dist_Mean_INTER'].fillna(0)
train_data['Dist_Max_INTER'] = train_data['Dist_Max_INTER'].fillna(0)
train_data['Dist_Mean_NAL'] = train_data['Dist_Mean_NAL'].fillna(0)
```

```
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['EDAD'] =
train_data['EDAD'].fillna(train_data['EDAD'].median())
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['SEXO'] =
train_data['SEXO'].fillna(train_data['SEXO'].mode()[0])
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:4:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['Dist_Sum_INTER'] =
train_data['Dist_Sum_INTER'].fillna(0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:5:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['Dist_Mean_INTER'] =
train_data['Dist_Mean_INTER'].fillna(0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:6:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['Dist_Max_INTER'] =
train_data['Dist_Max_INTER'].fillna(0)
C:\Users\jaime\AppData\Local\Temp\ipykernel_3692\1449264375.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  train_data['Dist_Mean_NAL'] = train_data['Dist_Mean_NAL'].fillna(0)
```

```
#Conteo de la cantidad de nulos por cada columna
regs=train_data.shape[0]
df_transform=train_data.copy()
for col in train_data.columns:
    vals=train_data[col].value_counts()
    values=vals.shape[0]
    nas=train_data[train_data[col].isna()].shape[0]
    porcentaje_na=nas*100/regs
    if porcentaje_na>=10:
        text=f'elimienando columna: {col}'
        df_transform.drop(columns={col},inplace=True)
    else:
        text=f'Quitando nulos de {col}'
        df_transform.dropna(subset=col,inplace=True)

    print(f'-variable {col} tiene {values} valores único, {nas}
valores nulos ({nas*100/regs:.2f})%.')
```

```
-variable id tiene 2888 valores único, 0 valores nulos (0.00)%.
-variable FRAUDE tiene 2 valores único, 0 valores nulos (0.00)%.
-variable VALOR tiene 2259 valores único, 0 valores nulos (0.00)%.
-variable HORA_AUX tiene 24 valores único, 0 valores nulos (0.00)%.
-variable Dist_max_NAL tiene 264 valores único, 0 valores nulos
(0.00)%.
-variable Canal1 tiene 2 valores único, 0 valores nulos (0.00)%.
```

```
-variable FECHA tiene 31 valores único, 0 valores nulos (0.00)%.
-variable COD_PAIS tiene 29 valores único, 0 valores nulos (0.00)%.
-variable CANAL tiene 3 valores único, 0 valores nulos (0.00)%.
-variable DIASEM tiene 7 valores único, 0 valores nulos (0.00)%.
-variable DIAMES tiene 31 valores único, 0 valores nulos (0.00)%.
-variable FECHA_VIN tiene 1025 valores único, 0 valores nulos (0.00)%.
-variable OFICINA_VIN tiene 373 valores único, 0 valores nulos
(0.00)%.
-variable SEXO tiene 2 valores único, 0 valores nulos (0.00)%.
-variable SEGMENTO tiene 6 valores único, 0 valores nulos (0.00)%.
-variable EDAD tiene 62 valores único, 0 valores nulos (0.00)%.
-variable INGRESOS tiene 500 valores único, 0 valores nulos (0.00)%.
-variable EGRESOS tiene 193 valores único, 0 valores nulos (0.00)%.
-variable NROPAISES tiene 8 valores único, 0 valores nulos (0.00)%.
-variable Dist_Sum_INTER tiene 209 valores único, 0 valores nulos
(0.00)%.
-variable Dist_Mean_INTER tiene 182 valores único, 0 valores nulos
(0.00)%.
-variable Dist_Max_INTER tiene 62 valores único, 0 valores nulos
(0.00)%.
-variable NROCIUDADES tiene 18 valores único, 0 valores nulos (0.00)%.
-variable Dist_Mean_NAL tiene 772 valores único, 0 valores nulos
(0.00)%.
-variable Dist_HOY tiene 48 valores único, 0 valores nulos (0.00)%.
-variable Dist_sum_NAL tiene 841 valores único, 0 valores nulos
(0.00)%.

train_data.info()

<class 'pandas.core.frame.DataFrame'>
Index: 2941 entries, 0 to 2964
Data columns (total 26 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   id               2941 non-null   int64
 1   FRAUDE           2941 non-null   int64
 2   VALOR            2941 non-null   float64
 3   HORA_AUX         2941 non-null   int64
 4   Dist_max_NAL     2941 non-null   float64
 5   Canal1           2941 non-null   object
 6   FECHA            2941 non-null   int64
 7   COD_PAIS         2941 non-null   object
 8   CANAL            2941 non-null   object
 9   DIASEM           2941 non-null   int64
 10  DIAMES           2941 non-null   int64
 11  FECHA_VIN        2941 non-null   float64
 12  OFICINA_VIN      2941 non-null   float64
 13  SEXO             2941 non-null   object
 14  SEGMENTO         2941 non-null   object
 15  EDAD             2941 non-null   float64
```

```
 16  INGRESOS         2941 non-null   float64
 17  EGRESOS          2941 non-null   float64
 18  NROPAISES        2941 non-null   int64
 19  Dist_Sum_INTER   2941 non-null   float64
 20  Dist_Mean_INTER  2941 non-null   float64
 21  Dist_Max_INTER   2941 non-null   float64
 22  NROCIUDADES      2941 non-null   int64
 23  Dist_Mean_NAL    2941 non-null   float64
 24  Dist_HOY         2941 non-null   float64
 25  Dist_sum_NAL     2941 non-null   float64
dtypes: float64(13), int64(8), object(5)
memory usage: 620.4+ KB
```

Entrenamiento del modelo

```
!pip install -U scikit-learn
```

```
Collecting scikit-learn
  Downloading scikit_learn-1.5.2-cp312-cp312-win_amd64.whl.metadata
(13 kB)
Requirement already satisfied: numpy>=1.19.5 in c:\users\jaime\
appdata\local\programs\python\python312\lib\site-packages (from
scikit-learn) (1.26.4)
Collecting scipy>=1.6.0 (from scikit-learn)
  Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl.metadata (60 kB)
     ---------------------------------------- 0.0/60.8 kB ? eta
-:--:--
     ------ --------------------------------- 10.2/60.8 kB ? eta
-:--:--
     ---------------------------------------- 60.8/60.8 kB 1.6 MB/s
eta 0:00:00
Collecting joblib>=1.2.0 (from scikit-learn)
  Downloading joblib-1.4.2-py3-none-any.whl.metadata (5.4 kB)
Collecting threadpoolctl>=3.1.0 (from scikit-learn)
  Downloading threadpoolctl-3.5.0-py3-none-any.whl.metadata (13 kB)
Downloading scikit_learn-1.5.2-cp312-cp312-win_amd64.whl (11.0 MB)
   ---------------------------------------- 0.0/11.0 MB ? eta -:--:--
   ---- ----------------------------------- 1.2/11.0 MB 37.1 MB/s eta
0:00:01
   ------------ --------------------------- 3.7/11.0 MB 46.5 MB/s eta
0:00:01
   ---------------------- ----------------- 6.1/11.0 MB 55.4 MB/s eta
0:00:01
   ------------------------------ --------- 8.5/11.0 MB 54.1 MB/s eta
0:00:01
   ---------------------------------------  11.0/11.0 MB 73.1 MB/s eta
0:00:01
   ---------------------------------------- 11.0/11.0 MB 59.5 MB/s eta
0:00:00
```

```
Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
   ---------------------------------------- 0.0/301.8 kB ? eta -:--:--
   ---------------------------------------- 301.8/301.8 kB ? eta
0:00:00
Downloading scipy-1.14.1-cp312-cp312-win_amd64.whl (44.5 MB)
   ---------------------------------------- 0.0/44.5 MB ? eta -:--:--
   -- ------------------------------------- 3.3/44.5 MB 70.4 MB/s eta
0:00:01
   ----- ---------------------------------- 5.7/44.5 MB 72.7 MB/s eta
0:00:01
   ------- -------------------------------- 8.4/44.5 MB 67.1 MB/s eta
0:00:01
   --------- ------------------------------ 11.0/44.5 MB 72.6 MB/s eta
0:00:01
   ----------- ---------------------------- 12.9/44.5 MB 65.6 MB/s eta
0:00:01
   ------------- -------------------------- 14.6/44.5 MB 59.5 MB/s eta
0:00:01
   --------------- ------------------------ 17.4/44.5 MB 59.5 MB/s eta
0:00:01
   ----------------- ---------------------- 19.9/44.5 MB 54.4 MB/s eta
0:00:01
   ------------------- -------------------- 22.3/44.5 MB 54.7 MB/s eta
0:00:01
   --------------------- ------------------ 24.3/44.5 MB 65.6 MB/s eta
0:00:01
   ----------------------- ---------------- 25.9/44.5 MB 65.2 MB/s eta
0:00:01
   ------------------------- -------------- 28.3/44.5 MB 54.7 MB/s eta
0:00:01
   --------------------------- ------------ 30.5/44.5 MB 50.4 MB/s eta
0:00:01
   ----------------------------- --------- 33.6/44.5 MB 50.4 MB/s eta
0:00:01
   ------------------------------- ------- 36.5/44.5 MB 65.6 MB/s eta
0:00:01
   --------------------------------- ------ 37.4/44.5 MB 65.6 MB/s eta
0:00:01
   ----------------------------------- ---- 39.0/44.5 MB 54.7 MB/s eta
0:00:01
   ------------------------------------- -- 42.3/44.5 MB 54.7 MB/s eta
0:00:01
   ---------------------------------------  44.5/44.5 MB 54.7 MB/s eta
0:00:01
   ---------------------------------------  44.5/44.5 MB 54.7 MB/s eta
0:00:01
   ---------------------------------------- 44.5/44.5 MB 36.4 MB/s eta
0:00:00
Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
```

```
Installing collected packages: threadpoolctl, scipy, joblib, scikit-
learn
Successfully installed joblib-1.4.2 scikit-learn-1.5.2 scipy-1.14.1
threadpoolctl-3.5.0


[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

División en datos de entrenamiento, validacion y evaluación

```
from sklearn.model_selection import train_test_split

# Dividir el dataset de entrenamiento
X = train_data.drop('FRAUDE', axis=1)
y = train_data['FRAUDE']

# Dividir en conjunto de entrenamiento y validación
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Contrucción del pipeline para el entranamiento y evalaución del modelo usando random forest y grid seach para la busqueda de los mejores hiperparametros

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import accuracy_score, f1_score, make_scorer

# Columnas categóricas
columnas_categoricas = ['Canal1', 'COD_PAIS', 'CANAL', 'SEXO',
'SEGMENTO']

# Preprocesamiento
preprocesamiento = ColumnTransformer(
    transformers=[
        # No preprocesamos las columnas numéricas
        ('categorico', Pipeline([
            ('onehotencoder', OneHotEncoder(handle_unknown='ignore',
sparse_output=False))  # Codificación one-hot sin salida esparcida
        ]), columnas_categoricas)
    ],
    remainder='passthrough'  # Deja pasar las columnas no
especificadas (en este caso, las numéricas)
)

# Pipeline con RandomForest
```

```
pipeline_rf = Pipeline([
    ('preprocesador', preprocesamiento),
    ('classifier', RandomForestClassifier(class_weight='balanced',
random_state=42))
])

# Configuración de la malla de búsqueda (hiperparámetros para ajustar)
malla_rf = {
    'classifier__n_estimators': [50, 100, 200],  # Número de árboles
    'classifier__max_depth': [None, 10, 20],  # Profundidad máxima del
árbol
    'classifier__min_samples_split': [2, 5, 10],  # Mínimo número de
muestras para dividir un nodo
    'classifier__min_samples_leaf': [1, 2, 4],  # Mínimo número de
muestras en una hoja
    'classifier__bootstrap': [True, False]  # Si utilizar o no
bootstrap
}

# Crea el GridSearchCV
grid_rf = GridSearchCV(pipeline_rf, param_grid=malla_rf,
                       cv=5,  # Validación cruzada de 5
                       scoring={'accuracy':
make_scorer(accuracy_score),
                                'f1': make_scorer(f1_score)},  #
Métricas de evaluación
                       refit='f1',  # Métrica principal para refit
                       return_train_score=True)

# Ajustar el modelo con GridSearch
grid_rf.fit(X_train, y_train)

# Mejores parámetros encontrados por GridSearch
print(f'Mejores Parametros: {grid_rf.best_params_}')
```

```
Mejores Parametros: {'classifier__bootstrap': False,
'classifier__max_depth': None, 'classifier__min_samples_leaf': 1,
'classifier__min_samples_split': 2, 'classifier__n_estimators': 200}
```

Evaluación del modelo

```
from sklearn.metrics import accuracy_score, f1_score

# Predecir y evaluar en el conjunto de validación
y_pred = grid_rf.predict(X_val)

print(f'Accuracy: {accuracy_score(y_val, y_pred)}')
print(f'F1-Score: {f1_score(y_val, y_pred)}')
```

```
Accuracy: 0.969439728353141
F1-Score: 0.9411764705882353
```

Generación del archivo de predicciones "predicciones_fraude.csv"

```
# Predecir en el archivo de prueba
predicciones_test = grid_rf.predict(test_data)

# Guardar las predicciones
output = pd.DataFrame({'id': test_data['id'], 'FRAUDE':
predicciones_test})
output.to_csv('predicciones_fraude.csv', index=False)

!pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\jaime\appdata\local\
programs\python\python312\lib\site-packages (1.26.4)


[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Matriz de confusión y comparación de las predicciones con los datos de validación

```
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score


# Comparar y_pred (predicciones) y y_val (valores reales)
matriz_confusion = confusion_matrix(y_val, y_pred)
accuracy = accuracy_score(y_val, y_pred)
f1 = f1_score(y_val, y_pred)

print("Matriz de confusión:")
print(matriz_confusion)

print(f"Accuracy: {accuracy}")
print(f"F1 Score: {f1}")
```

```
Matriz de confusión:
[[427  10]
 [  8 144]]
Accuracy: 0.969439728353141
F1 Score: 0.9411764705882353
```