Instituto Tecnológico y de Estudios Superiores de Monterrey

Reto 1: Limpieza, análisis, visualización y kmeans

Arturo Eduardo Loperena GutierrezA01793641

Karla Daniela Valenzuela Gomez A00819192

```
import pandas as pd
import numpy as np
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import math
import matplotlib.pyplot as plt
import requests, zipfile
from io import BytesIO
from imblearn.metrics import geometric_mean_score, classification_report_imbalanced
from google.colab import drive
from sklearn.model_selection import learning_curve, validation_curve
from sklearn.preprocessing import QuantileTransformer
from sklearn.preprocessing import power_transform
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report, make_scorer
from sklearn.model_selection import  cross_validate,  RepeatedStratifiedKFold
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder, StandardScaler
from sklearn.preprocessing import FunctionTransformer
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import OneHotEncoder
```

```
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Saved successfully!

```
GridSearchCV
from sklearn.dummy import DummyRegressor
from sklearn.linear_model import LinearRegression
from sklearn.compose import TransformedTargetRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import RepeatedKFold
from shapely.geometry import Point
! pip install qeds fiona geopandas xgboost gensim folium pyLDAvis descartes
import geopandas as gpd
```

```
Requirement already satisfied: descartes in /usr/local/lib/python3.7/dist-packages (1.1.0)
Requirement already satisfied: quandl in /usr/local/lib/python3.7/dist-packages (from qeds) (3.7.0)
Requirement already satisfied: plotly in /usr/local/lib/python3.7/dist-packages (from qeds) (5.5.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from qeds) (3.2.2)
Requirement already satisfied: pandas-datareader in /usr/local/lib/python3.7/dist-packages (from qeds) (0.9.0)
Requirement already satisfied: pyarrow in /usr/local/lib/python3.7/dist-packages (from qeds) (6.0.1)
Requirement already satisfied: quantecon in /usr/local/lib/python3.7/dist-packages (from qeds) (0.5.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from qeds) (1.3.5)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from qeds) (1.21.6)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from qeds) (2.23.0)
Requirement already satisfied: openpyxl in /usr/local/lib/python3.7/dist-packages (from qeds) (3.0.10)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages (from qeds) (1.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from qeds) (1.7.3)

Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (from qeds) (0.11.2)
Requirement already satisfied: statsmodels in /usr/local/lib/python3.7/dist-packages (from qeds) (0.12.2)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.7/dist-packages (from fiona) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.7/dist-packages (from fiona) (0.7.2)
Requirement already satisfied: six>=1.7 in /usr/local/lib/python3.7/dist-packages (from fiona) (1.15.0)
Requirement already satisfied: munch in /usr/local/lib/python3.7/dist-packages (from fiona) (2.5.0)
Requirement already satisfied: click>=4.0 in /usr/local/lib/python3.7/dist-packages (from fiona) (7.1.2)
Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages (from fiona) (2022.9.24)
Requirement already satisfied: attrs>=17 in /usr/local/lib/python3.7/dist-packages (from fiona) (22.1.0)
```

```
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from fiona) (57.4.0)
Requirement already satisfied: shapely>=1.6 in /usr/local/lib/python3.7/dist-packages (from geopandas) (1.8.5.post1
Requirement already satisfied: pyproj>=2.2.0 in /usr/local/lib/python3.7/dist-packages (from geopandas) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas->qeds)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->qeds) (2022.6)
Requirement already satisfied: smart-open>=1.2.1 in /usr/local/lib/python3.7/dist-packages (from gensim) (5.2.1)
Requirement already satisfied: jinja2>=2.9 in /usr/local/lib/python3.7/dist-packages (from folium) (2.11.3)
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.7/dist-packages (from folium) (0.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2>=2.9->folium
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (from pyLDAvis) (1.2.0)
Requirement already satisfied: sklearn in /usr/local/lib/python3.7/dist-packages (from pyLDAvis) (0.0.post1)
Requirement already satisfied: funcy in /usr/local/lib/python3.7/dist-packages (from pyLDAvis) (1.17)
Requirement already satisfied: future in /usr/local/lib/python3.7/dist-packages (from pyLDAvis) (0.16.0)
Requirement already satisfied: numexpr in /usr/local/lib/python3.7/dist-packages (from pyLDAvis) (2.8.4)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->qeds)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->qeds) (0.11
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.7/dist-packages (from kiwisolver>=1.0.1-
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.7/dist-packages (from openpyxl->qeds) (1.1.0)
Requirement already satisfied: lxml in /usr/local/lib/python3.7/dist-packages (from pandas-datareader->qeds) (4.9.1
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->qeds) (3
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->qeds) (2.10)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.7/dist-packages (from plotly->qeds) (8.1.0
Requirement already satisfied: more-itertools in /usr/local/lib/python3.7/dist-packages (from quandl->qeds) (9.0.0)
Requirement already satisfied: inflection>=0.3.1 in /usr/local/lib/python3.7/dist-packages (from quandl->qeds) (0.5
Requirement already satisfied: sympy in /usr/local/lib/python3.7/dist-packages (from quantecon->qeds) (1.7.1)
Requirement already satisfied: numba in /usr/local/lib/python3.7/dist-packages (from quantecon->qeds) (0.56.4)
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dist-packages (from numba->quantecon-
Requirement already satisfied: llvmlite<0.40,>=0.39.0dev0 in /usr/local/lib/python3.7/dist-packages (from numba->qu
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata->numba-
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from scikit-learn->q


Requirement already satisfied: patsy>=0.5 in /usr/local/lib/python3.7/dist-packages (from statsmodels->qeds) (0.5.3
```

Saved successfully!

```
url = 'http://201.116.60.46/Datos_de_calidad_del_agua_de_5000_sitios_de_monitoreo.zip'
req = requests.get(url)
zipfile.ZipFile(BytesIO(req.content)).extractall('unzipped_zip/')
```

```
df_sub=pd.read_csv('unzipped_zip/Datos_de_calidad_del_agua_2020/Datos_de_calidad_del_agua_de_sitios_de_monitoreo_de_aguas_sul
df_sub.head()
```

Saved successfully!

	CLAVE	SITIO	ORGANISMO_DE_CUENCA	ESTADO	MUNICIPIO	ACUIFERO	SUBTIPO	LONGIT
			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	ASIENTOS	VALLE DE CHICALOTE	POZO	-102.022
1	DLAGU6516	POZO R013 CAÑADA HONDA	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	VALLE DE CHICALOTE	POZO	-102.200
2	DLAGU7	POZO COSIO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	COSIO	VALLE DE AGUASCALIENTES	POZO	-102.288
3	DLAGU9	POZO EL SALITRILLO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	RINCON DE ROMOS	VALLE DE AGUASCALIENTES	POZO	-102.294
4	DLBAJ107	RANCHO EL TECOLOTE	PENINSULA DE BAJA CALIFORNIA	BAJA CALIFORNIA SUR	LA PAZ	TODOS SANTOS	POZO	-110.244

5 rows × 57 columns

```
df = df_sub
```

```
df
```

	CLAVE	SITIO	ORGANISMO_DE_CUENCA	ESTADO	MUNICIPIO	ACUIFERO	SUBTIPO
0	DLAGU6	POZO SAN GIL	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	ASIENTOS	VALLE DE CHICALOTE	POZO
		13 DA HONDA	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	VALLE DE CHICALOTE	POZO
2	DLAGU7	POZO COSIO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	COSIO	VALLE DE AGUASCALIENTES	POZO
3	DLAGU9	POZO EL SALITRILLO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	RINCON DE ROMOS	VALLE DE AGUASCALIENTES	POZO
4	DLBAJ107	RANCHO EL TECOLOTE	PENINSULA DE BAJA CALIFORNIA	BAJA CALIFORNIA SUR	LA PAZ	TODOS SANTOS	POZO
...	...	...	...	...	...	...	...
1063	OCRBR5101M1	L-310 (COMUNIDAD SAN MANUEL)	RIO BRAVO	NUEVO LEON	LINARES	CITRICOLA SUR	POZO
1064	OCRBR5102M1	L-305 (EJIDO OJO DE AGUA LAS CRUCESITAS)	RIO BRAVO	NUEVO LEON	LINARES	CITRICOLA SUR	POZO
1065	OCRBR5105M2	HACIENDA MEXIQUITO POZO 01	RIO BRAVO	NUEVO LEON	CADEREYTA JIMENEZ	CITRICOLA NORTE	POZO
		COMUNIDAD				NAVIDAD POTOSI	

Saved successfully!

```
df.describe()
```

	LONGITUD	LATITUD	PERIODO	ALC_mg/L	CONDUCT_mS/cm	SDT_mg/L
count	1068.000000	1068.000000	1068.0	1064.000000	1062.000000	0.0
			2020.0	235.633759	1138.953013	NaN
			0.0	116.874291	1245.563674	NaN
min	-116.664250	14.561150	2020.0	26.640000	50.400000	NaN
25%	-105.388865	20.212055	2020.0	164.000000	501.750000	NaN

Saved successfully!

```
df.info()
```

```
 2   ORGANISMO_DE_CUENCA   1068 non-null   object
 3   ESTADO                1068 non-null   object
 4   MUNICIPIO             1068 non-null   object
 5   ACUIFERO              1068 non-null   object
 6   SUBTIPO               1068 non-null   object
 7   LONGITUD              1068 non-null   float64
 8   LATITUD               1068 non-null   float64
 9   PERIODO               1068 non-null   int64
 10  ALC_mg/L              1064 non-null   float64
 11  CALIDAD_ALC           1064 non-null   object
 12  CONDUCT_mS/cm         1062 non-null   float64
 13  CALIDAD_CONDUC        1062 non-null   object
 14  SDT_mg/L              0 non-null      float64
 15  SDT_M_mg/L            1066 non-null   object
 16  CALIDAD_SDT_ra        1066 non-null   object
 17  CALIDAD_SDT_salin     1066 non-null   object
 18  FLUORUROS_mg/L        1068 non-null   object
 19  CALIDAD_FLUO          1068 non-null   object
 20  DUR_mg/L              1067 non-null   object
 21  CALIDAD_DUR           1067 non-null   object
 22  COLI_FEC_NMP/100_mL   1068 non-null   object

 23  CALIDAD_COLI_FEC      1068 non-null   object
 24  N_NO3_mg/L            1067 non-null   object
 25  CALIDAD_N_NO3         1067 non-null   object
 26  AS_TOT_mg/L           1068 non-null   object
 27  CALIDAD_AS            1068 non-null   object
 28  CD_TOT_mg/L           1068 non-null   object
 29  CALIDAD_CD            1068 non-null   object
```

```
 30  CR_TOT_mg/L           1068 non-null   object
 31  CALIDAD_CR            1068 non-null   object
 32  HG_TOT_mg/L           1068 non-null   object
 33  CALIDAD_HG            1068 non-null   object
 34  PB_TOT_mg/L           1068 non-null   object
                           1068 non-null   object
                           1068 non-null   object
                           1068 non-null   object
 38  FE_TOT_mg/L           1068 non-null   object
 39  CALIDAD_FE            1068 non-null   object
 40  SEMAFORO              1068 non-null   object
 41  CONTAMINANTES         634 non-null    object
 42  CUMPLE_CON_ALC        1068 non-null   object
 43  CUMPLE_CON_COND       1068 non-null   object
 44  CUMPLE_CON_SDT_ra     1068 non-null   object
 45  CUMPLE_CON_SDT_salin  1068 non-null   object
 46  CUMPLE_CON_FLUO       1068 non-null   object
 47  CUMPLE_CON_DUR        1068 non-null   object
 48  CUMPLE_CON_CF         1068 non-null   object
 49  CUMPLE_CON_NO3        1068 non-null   object
 50  CUMPLE_CON_AS         1068 non-null   object
 51  CUMPLE_CON_CD         1068 non-null   object
 52  CUMPLE_CON_CR         1068 non-null   object
 53  CUMPLE_CON_HG         1068 non-null   object
 54  CUMPLE_CON_PB         1068 non-null   object
 55  CUMPLE_CON_MN         1068 non-null   object
 56  CUMPLE_CON_FE         1068 non-null   object
dtypes: float64(5), int64(1), object(51)
memory usage: 475.7+ KB
```

Saved successfully!

```
df.shape
```

```
(1068, 57)
```

```
df.columns
```

```
Index(['CLAVE', 'SITIO', 'ORGANISMO_DE_CUENCA', 'ESTADO', 'MUNICIPIO',
       'ACUIFERO', 'SUBTIPO', 'LONGITUD', 'LATITUD', 'PERIODO', 'ALC_mg/L',
       'CALIDAD_ALC', 'CONDUCT_mS/cm', 'CALIDAD_CONDUC', 'SDT_mg/L',
       'SDT_M_mg/L', 'CALIDAD_SDT_ra', 'CALIDAD_SDT_salin', 'FLUORUROS_mg/L',
```

```
       'CALIDAD_FLUO', 'DUR_mg/L', 'CALIDAD_DUR', 'COLI_FEC_NMP/100_mL',
       'CALIDAD_COLI_FEC', 'N_NO3_mg/L', 'CALIDAD_N_NO3', 'AS_TOT_mg/L',
       'CALIDAD_AS', 'CD_TOT_mg/L', 'CALIDAD_CD', 'CR_TOT_mg/L', 'CALIDAD_CR',
       'HG_TOT_mg/L', 'CALIDAD_HG', 'PB_TOT_mg/L', 'CALIDAD_PB', 'MN_TOT_mg/L',
       'CALIDAD_MN', 'FE_TOT_mg/L', 'CALIDAD_FE', 'SEMAFORO', 'CONTAMINANTES',
       'CUMPLE_CON_COND', 'CUMPLE_CON_SDT_ra',
       'CUMPLE_CON_SDT_salin', 'CUMPLE_CON_FLUO', 'CUMPLE_CON_DUR',
       'CUMPLE_CON_CF', 'CUMPLE_CON_NO3', 'CUMPLE_CON_AS', 'CUMPLE_CON_CD',
       'CUMPLE_CON_CR', 'CUMPLE_CON_HG', 'CUMPLE_CON_PB', 'CUMPLE_CON_MN',
       'CUMPLE_CON_FE'],
      dtype='object')
```

Saved successfully!

```
df.isna().sum().sort_values(ascending=False)
```

```
SDT_mg/L                1068
CONTAMINANTES            434
CALIDAD_CONDUC             6
CONDUCT_mS/cm              6
ALC_mg/L                   4
CALIDAD_ALC                4
CALIDAD_SDT_ra             2
SDT_M_mg/L                 2
CALIDAD_SDT_salin          2
CALIDAD_N_NO3              1
CALIDAD_DUR                1
N_NO3_mg/L                 1
DUR_mg/L                   1
CUMPLE_CON_COND            0
CUMPLE_CON_ALC             0
SEMAFORO                   0
CALIDAD_FE                 0
FE_TOT_mg/L                0
CALIDAD_MN                 0
CUMPLE_CON_SDT_ra          0
CUMPLE_CON_SDT_salin       0
CLAVE                      0
CUMPLE_CON_FLUO            0
CUMPLE_CON_DUR             0
CALIDAD_PB                 0
CUMPLE_CON_CF              0
CUMPLE_CON_NO3             0
CUMPLE_CON_AS              0
CUMPLE_CON_CD              0
```

```
CUMPLE_CON_CD              0
CUMPLE_CON_CR              0
CUMPLE_CON_HG              0
CUMPLE_CON_PB              0
CUMPLE_CON_MN              0
```

Saved successfully!

```
CALIDAD_HG                 0
ORGANISMO_DE_CUENCA        0
ESTADO                     0
MUNICIPIO                  0
ACUIFERO                   0
SUBTIPO                    0
LONGITUD                   0
LATITUD                    0
PERIODO                    0
FLUORUROS_mg/L             0
CALIDAD_FLUO               0
COLI_FEC_NMP/100_mL        0
CALIDAD_COLI_FEC           0
AS_TOT_mg/L                0
CALIDAD_AS                 0
SITIO                      0
CALIDAD_CD                 0
CR_TOT_mg/L                0
CALIDAD_CR                 0
HG_TOT_mg/L                0
CUMPLE_CON_FE              0
dtype: int64
```

```
columnas_numericas = ['ALC_mg/L','CONDUCT_mS/cm','SDT_mg/L','SDT_M_mg/L','FLUORUROS_mg/L','DUR_mg/L','COLI_FEC_NMP/100_mL',
                      'N_NO3_mg/L','AS_TOT_mg/L','CD_TOT_mg/L','CR_TOT_mg/L','HG_TOT_mg/L','PB_TOT_mg/L','MN_TOT_mg/L','FE_T(
```

```
new_df = df[columnas_numericas]
```

```
new_df
```

	ALC_mg/L	CONDUCT_mS/cm	SDT_mg/L	SDT_M_mg/L	FLUORUROS_mg/L	DUR_mg/L	COLI_FEC_NMP/100_mL	N_NO3_mg/L	AS_
0	229.990	940.0	NaN	603.6	0.9766	213.732	<1.1	4.184656	
			NaN	445.4	0.9298	185.0514	<1.1	5.75011	
			NaN	342	1.8045	120.719	<1.1	1.449803	
3	327.000	686.0	NaN	478.6	1.1229	199.879	<1.1	1.258597	
4	309.885	1841.0	NaN	1179	0.2343	476.9872	291	15.672251	
...	...	...	...	...	...	...	...	...	
1063	231.045	2350.0	NaN	1545.8	<0.2	752.096	<1.1	14.615488	
1064	256.000	529.0	NaN	297	<0.2	273	<1.1	77.392	
1065	330.690	2600.0	NaN	1873	0.7574	660.2126	620	36.477104	
1066	193.140	873.0	NaN	690.6667	0.7108	406.368	<1.1	<0.02	
1067	263.070	817.0	NaN	495	0.4002	362.544	<1.1	0.811876	

Saved successfully!

```
for i in columnas_numericas:
  print('Valores unicos', new_df[i].value_counts())
```

```
0.00002      1
Name: HG_TOT_mg/L, Length: 61, dtype: int64
Valores unicos <0.005     1038
0.01225       1
0.00709       1
0.00596       1
0.046         1
0.005         1
0.00744       1
0.00644       1
0.00619       1
0.00703       1
0.0133        1
0.00734       1
0.00557       1
0.00777       1
0.01075       1
0.0116        1
```

```
0.0399        1

0.00556       1
0.00859       1
0.0086        1
```

Saved successfully!

```
0.00813       1
0.01117       1
0.0152        1
0.0219        1
0.0809        1
0.0135        1
0.049         1
0.0053        1
Name: PB_TOT_mg/L, dtype: int64
Valores unicos <0.0015    545
0.0017      12
0.0021      10
0.0016       9
0.003        8
          ...
0.0056       1
0.0193       1
0.00445      1
0.0208       1
0.0242       1
Name: MN_TOT_mg/L, Length: 362, dtype: int64
Valores unicos <0.025    401
0.0288      4
0.0492      4
0.0471      3
0.0564      3
          ...
0.1118      1
0.0565      1
0.3947      1
0.0858      1
0.1786      1
Name: FE_TOT_mg/L, Length: 615, dtype: int64
```

```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1068 entries, 0 to 1067
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   ALC_mg/L            1064 non-null   float64
 1   CONDUCT_mS/cm       1062 non-null   float64
 2   SDT_mg/L            0 non-null      float64
 3   SDT_M_mg/L          1066 non-null   object 
 4   FLUORUROS_mg/L      1068 non-null   object 
 5   DUR_mg/L            1067 non-null   object 
 6   COLI_FEC_NMP/100_mL  1068 non-null   object 
 7   N_NO3_mg/L          1067 non-null   object 
 8   AS_TOT_mg/L         1068 non-null   object 
 9   CD_TOT_mg/L         1068 non-null   object 
 10  CR_TOT_mg/L         1068 non-null   object 
 11  HG_TOT_mg/L         1068 non-null   object 
 12  PB_TOT_mg/L         1068 non-null   object 
 13  MN_TOT_mg/L         1068 non-null   object 
 14  FE_TOT_mg/L         1068 non-null   object 
dtypes: float64(3), object(12)
memory usage: 125.3+ KB
```



```
for i in columnas_numericas:
  new_df[i] = new_df[i].astype('str')
  new_df[i] = new_df[i].str.replace('<0.2','0.2')
  new_df[i] = new_df[i].str.replace('<20','20')
  new_df[i] = new_df[i].str.replace('<1.1','1.1')
  new_df[i] = new_df[i].str.replace('<0.02','0.02')
  new_df[i] = new_df[i].str.replace('<0.01','0.01')
  new_df[i] = new_df[i].str.replace('<0.003','0.003')
  new_df[i] = new_df[i].str.replace('<0.005','0.005')
  new_df[i] = new_df[i].str.replace('<0.0005','0.0005')
  new_df[i] = new_df[i].str.replace('<0.0015','0.0015')
  new_df[i] = new_df[i].str.replace('<0.025','0.025')
  new_df[i] = new_df[i].str.replace('<25','25')
  new_df[i] = new_df[i].astype('float')
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
  import sys
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:8: FutureWarning: The default value of regex will chan

                                                    ackages/ipykernel_launcher.py:8: SettingWithCopyWarning:
                                                    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#returni

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: FutureWarning: The default value of regex will chan
  if __name__ == '__main__':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:9: 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#returni
  if __name__ == '__main__':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: FutureWarning: The default value of regex will cha
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: 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#returni
  # Remove the CWD from sys.path while we load stuff.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: FutureWarning: The default value of regex will cha
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:11: 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#returni
  # This is added back by InteractiveShellApp.init_path()
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: FutureWarning: The default value of regex will cha
  if sys.path[0] == '':
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:12: 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#returni
  if sys.path[0] == '':
```

Saved successfully!

```
if sys.path[0] == '':

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:13: 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#returni

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: 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#returni
```

Saved successfully!

```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1068 entries, 0 to 1067
Data columns (total 15 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ALC_mg/L             1064 non-null   float64
 1   CONDUCT_mS/cm        1062 non-null   float64
 2   SDT_mg/L             0 non-null      float64
 3   SDT_M_mg/L           1066 non-null   float64
 4   FLUORUROS_mg/L       1068 non-null   float64
 5   DUR_mg/L             1067 non-null   float64
 6   COLI_FEC_NMP/100_mL  1068 non-null   float64
 7   N_NO3_mg/L           1067 non-null   float64
 8   AS_TOT_mg/L          1068 non-null   float64
 9   CD_TOT_mg/L          1068 non-null   float64
 10  CR_TOT_mg/L          1068 non-null   float64
 11  HG_TOT_mg/L          1068 non-null   float64
 12  PB_TOT_mg/L          1068 non-null   float64
 13  MN_TOT_mg/L          1068 non-null   float64
 14  FE_TOT_mg/L          1068 non-null   float64
dtypes: float64(15)
memory usage: 125.3 KB
```

```
new_df.drop('SDT_mg/L', axis=1, inplace=True)
```

```
/usr/local/lib/python3.7/dist-packages/pandas/core/frame.py:4913: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
```

Saved successfully!

ation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a

```
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1068 entries, 0 to 1067
Data columns (total 14 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   ALC_mg/L             1064 non-null   float64
 1   CONDUCT_mS/cm        1062 non-null   float64
 2   SDT_M_mg/L           1066 non-null   float64
 3   FLUORUROS_mg/L       1068 non-null   float64
 4   DUR_mg/L             1067 non-null   float64
 5   COLI_FEC_NMP/100_mL  1068 non-null   float64
 6   N_NO3_mg/L           1067 non-null   float64
 7   AS_TOT_mg/L          1068 non-null   float64
 8   CD_TOT_mg/L          1068 non-null   float64
 9   CR_TOT_mg/L          1068 non-null   float64
 10  HG_TOT_mg/L          1068 non-null   float64
 11  PB_TOT_mg/L          1068 non-null   float64
 12  MN_TOT_mg/L          1068 non-null   float64
 13  FE_TOT_mg/L          1068 non-null   float64
dtypes: float64(14)
memory usage: 116.9 KB
```

```
y = pd.DataFrame(df['SEMAFORO'])
```

```
y.value_counts()
```

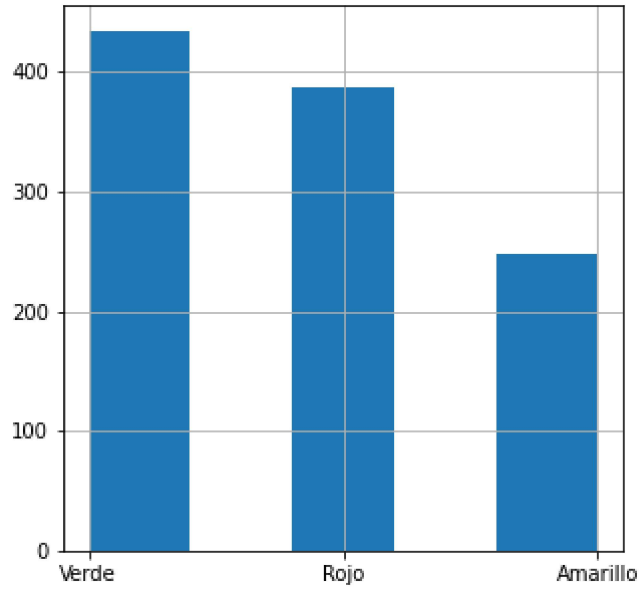
```
SEMAFORO
```

```
Verde       434
Rojo        387
Amarillo    247
dtype: int64
```

Saved successfully!

```
=(5,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2ce823610>
```



y

	SEMAFORO
0	Verde

Saved successfully!

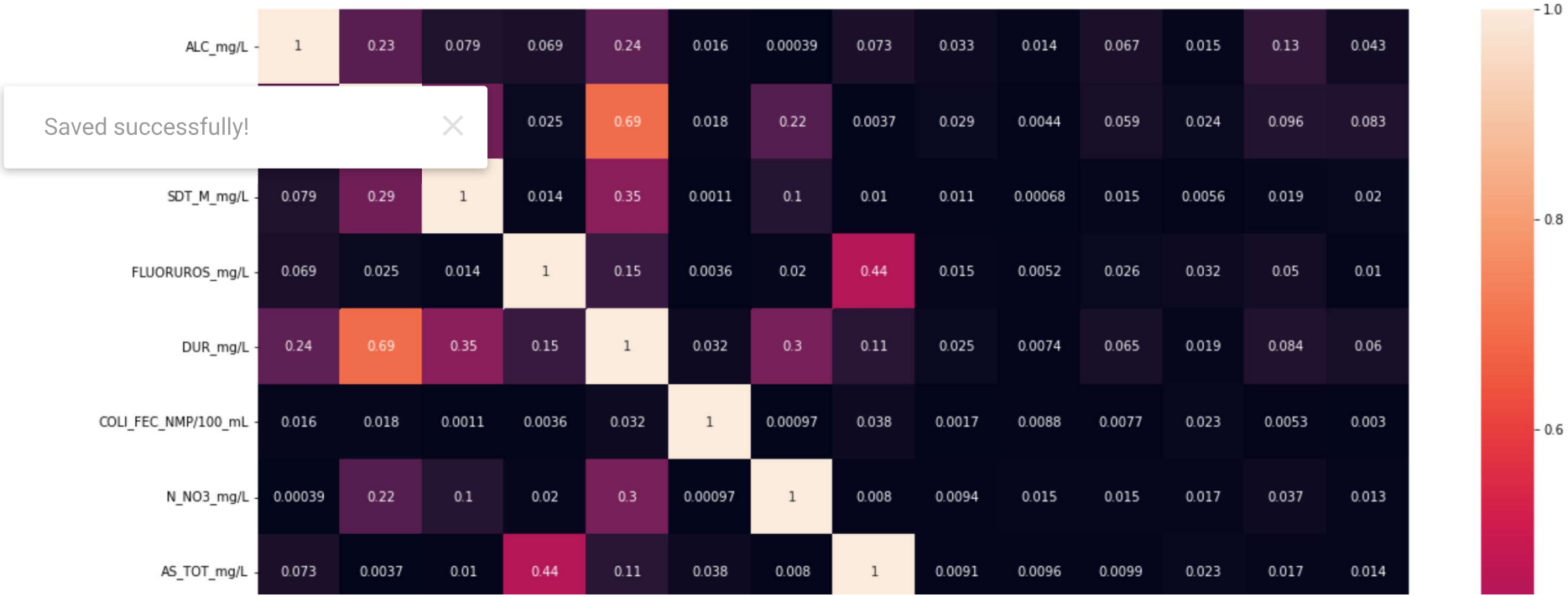
3	Verde
4	Rojo

```
new_df.corr()
```

	ALC_mg/L	CONDUCT_mS/cm	SDT_M_mg/L	FLUORUROS_mg/L	DUR_mg/L	COLI_FEC_NMP/100_mL	N_NO3_mg/
ALC_mg/L	1.000000	0.232003	0.079350	0.069056	0.243177	-0.016449	-0.0003
CONDUCT_mS/cm	0.232003	1.000000	0.286555	-0.025200	0.693146	0.017808	0.2195
SDT_M_mg/L	0.079350	0.286555	1.000000	-0.013798	0.347205	-0.001133	0.1017
FLUORUROS_mg/L	0.069056	-0.025200	-0.013798	1.000000	-0.149691	0.003564	-0.0197
DUR_mg/L	0.243177	0.693146	0.347205	-0.149691	1.000000	0.031727	0.3014
COLI_FEC_NMP/100_mL	-0.016449	0.017808	-0.001133	0.003564	0.031727	1.000000	-0.0009
N_NO3_mg/L	-0.000394	0.219512	0.101706	-0.019782	0.301468	-0.000969	1.0000
AS_TOT_mg/L	0.073299	-0.003722	-0.010157	0.444079	-0.106498	0.038151	-0.0080
CD_TOT_mg/L	0.032686	0.029040	0.010800	-0.015123	0.025002	-0.001656	-0.0093
CR_TOT_mg/L	-0.014282	0.004412	-0.000682	-0.005242	0.007438	-0.008840	-0.0151
HG_TOT_mg/L	0.067195	0.059093	0.015114	-0.026358	0.064839	-0.007661	0.0149
PB_TOT_mg/L	0.015064	0.024083	-0.005552	-0.032236	-0.018908	0.022510	-0.0165
MN_TOT_mg/L	0.129866	0.095955	0.018927	-0.049742	0.083822	-0.005326	-0.0368
FE_TOT_mg/L	0.043423	0.083181	0.020104	-0.009994	0.059775	0.003045	0.0132

```
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=(20,15))

sns.heatmap(new_df.corr().abs(), annot=True, cbar='Spectral', ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2ce99a0d0>
```




```
new_df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
ALC_mg/L	1064.0	235.633759	116.874291	26.6400	164.000000	215.527500	292.710000	1650.000000
		1138.953013	1245.563674	50.4000	501.750000	815.000000	1322.750000	18577.000000
		896.101567	2751.530590	25.0000	337.500000	550.400000	916.100000	82170.000000
FLUORUROS_mg/L	1068.0	1.075600	1.924278	0.2000	0.267175	0.503500	1.139850	34.803300
DUR_mg/L	1067.0	347.938073	359.669452	20.0000	121.194800	245.335800	453.930000	3810.692200

Saved successfully!

```
new_df
```

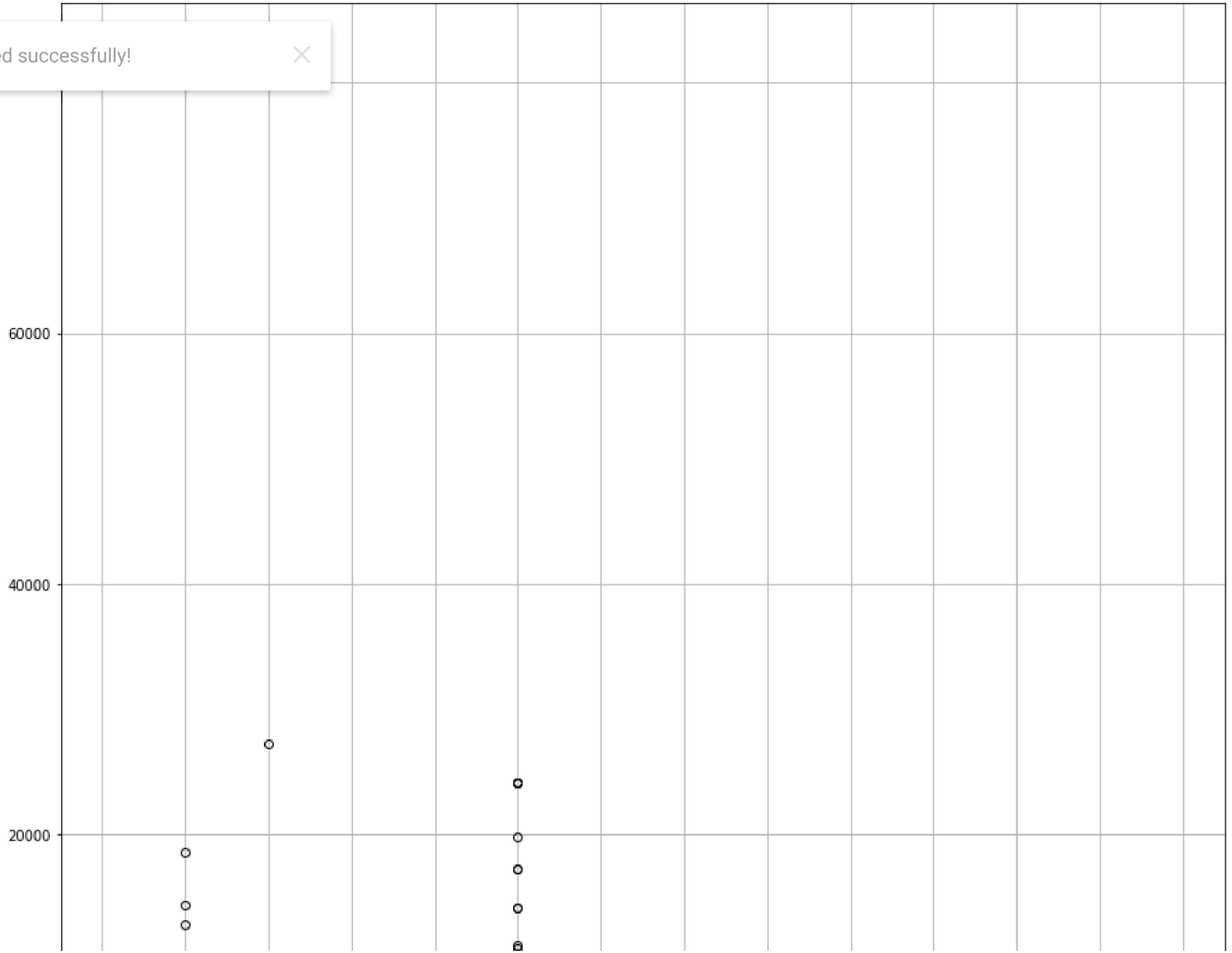
	ALC_mg/L	CONDUCT_mS/cm	SDT_M_mg/L	FLUORUROS_mg/L	DUR_mg/L	COLI_FEC_NMP/100_mL	N_NO3_mg/L	AS_TOT_mg/L
0	229.990	940.0	603.6000	0.9766	213.7320	1.1	4.184656	0.0161
1	231.990	608.0	445.4000	0.9298	185.0514	1.1	5.750110	0.0134
2	204.920	532.0	342.0000	1.8045	120.7190	1.1	1.449803	0.0370
3	327.000	686.0	478.6000	1.1229	199.8790	1.1	1.258597	0.0154
4	309.885	1841.0	1179.0000	0.2343	476.9872	291.0	15.672251	0.0100
...	...	...	...	...	...	...	...	...
1063	231.045	2350.0	1545.8000	0.2000	752.0960	1.1	14.615488	0.0100
1064	256.000	529.0	297.0000	0.2000	273.0000	1.1	77.392000	0.0100
1065	330.690	2600.0	1873.0000	0.7574	660.2126	620.0	36.477104	0.0100
1066	193.140	873.0	690.6667	0.7108	406.3680	1.1	0.020000	0.0100
1067	263.070	817.0	495.0000	0.4002	362.5440	1.1	0.811876	0.0100

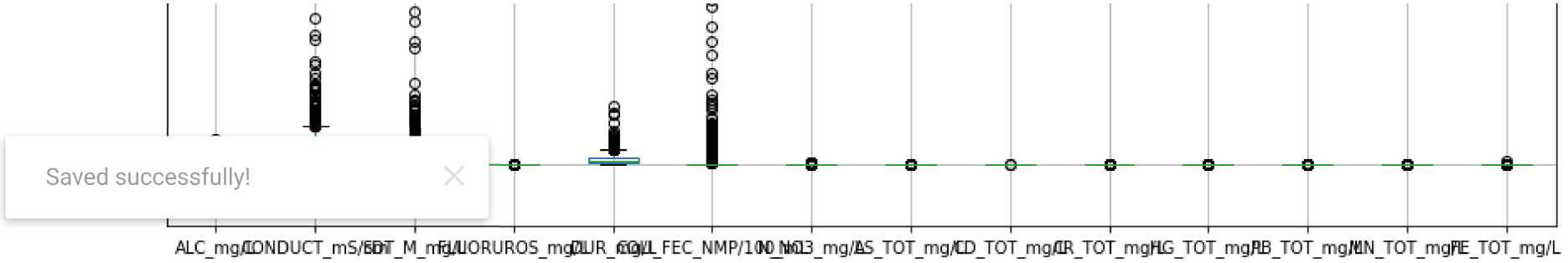
1068 rows × 14 columns

```
new_df.boxplot(figsize=(15,15))
```

```
/usr/local/lib/python3.7/dist-packages/matplotlib/cbook/__init__.py:1376: VisibleDeprecationWarning: Creating an nd
  X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2cea63e10>
```

Saved successfully!





Saved successfully!

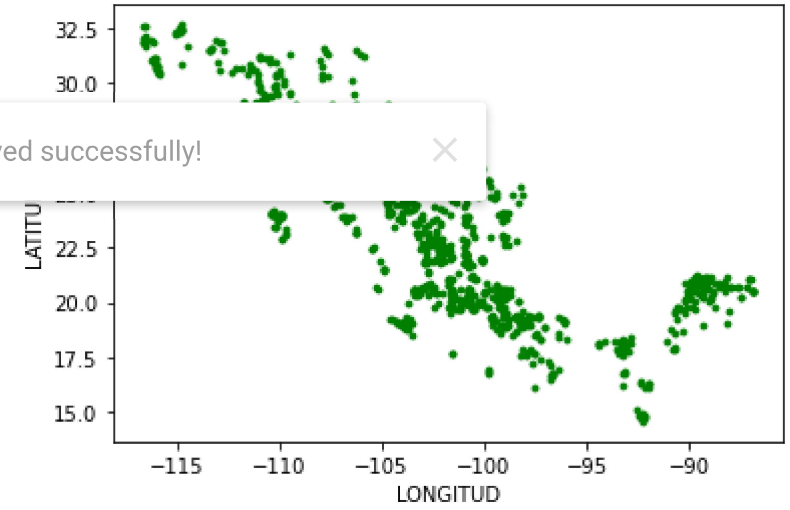
```
df_ubicacion = df[['LONGITUD','LATITUD']]
df_ubicacion
```

	LONGITUD	LATITUD
0	-102.02210	22.20887
1	-102.20075	21.99958
2	-102.28801	22.36685
3	-102.29449	22.18435
4	-110.24480	23.45138
...	...	...
1063	-99.54191	24.76036
1064	-99.70099	24.78280
1065	-99.82249	25.55197
1066	-100.32683	24.80118
1067	-100.73302	25.09380

1068 rows × 2 columns

```
df_ubicacion.plot.scatter('LONGITUD','LATITUD', s=8, c='g')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa2ced3dc50>
```




```
df_ubicacion
df_ubicacion["COORDENADAS"] = list(zip(df_ubicacion.LONGITUD, df_ubicacion.LATITUD))
df_ubicacion["COORDENADAS"] = df_ubicacion["COORDENADAS"].apply(Point)
df_ubicacion.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.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

ation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returni
```

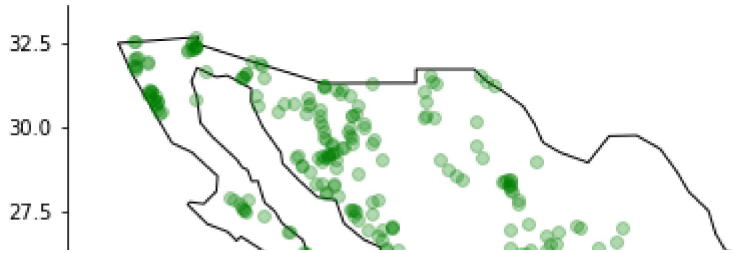


```
_ubicacion, geometry="COORDENADAS")

world = gpd.read_file(gpd.datasets.get_path("naturalearth_lowres"))

world = world.set_index("iso_a3")
world.name.unique()
fig, gax = plt.subplots(figsize=(10,10))

world.query("name == 'Mexico'").plot(ax=gax, edgecolor='black',color='white')

gax.set_xlabel('LATITUD')
gax.set_ylabel('LONGITUD')

gax.spines['top'].set_visible(False)
gax.spines['right'].set_visible(False)

puntos_en_mapa.plot(ax=gax, color='g', alpha = 0.3)
puntos_en_mapa
```

	LONGITUD	LATITUD	COORDENADAS
0	-102.02210	22.20887	POINT (-102.02210 22.20887)
			NT (-102.20075 21.99958)
			NT (-102.28801 22.36685)
3	-102.29449	22.18435	POINT (-102.29449 22.18435)
4	-110.24480	23.45138	POINT (-110.24480 23.45138)
...	...	...	...
1063	-99.54191	24.76036	POINT (-99.54191 24.76036)
1064	-99.70099	24.78280	POINT (-99.70099 24.78280)
1065	-99.82249	25.55197	POINT (-99.82249 25.55197)
1066	-100.32683	24.80118	POINT (-100.32683 24.80118)
1067	-100.73302	25.09380	POINT (-100.73302 25.09380)

1068 rows × 3 columns

Saved successfully!

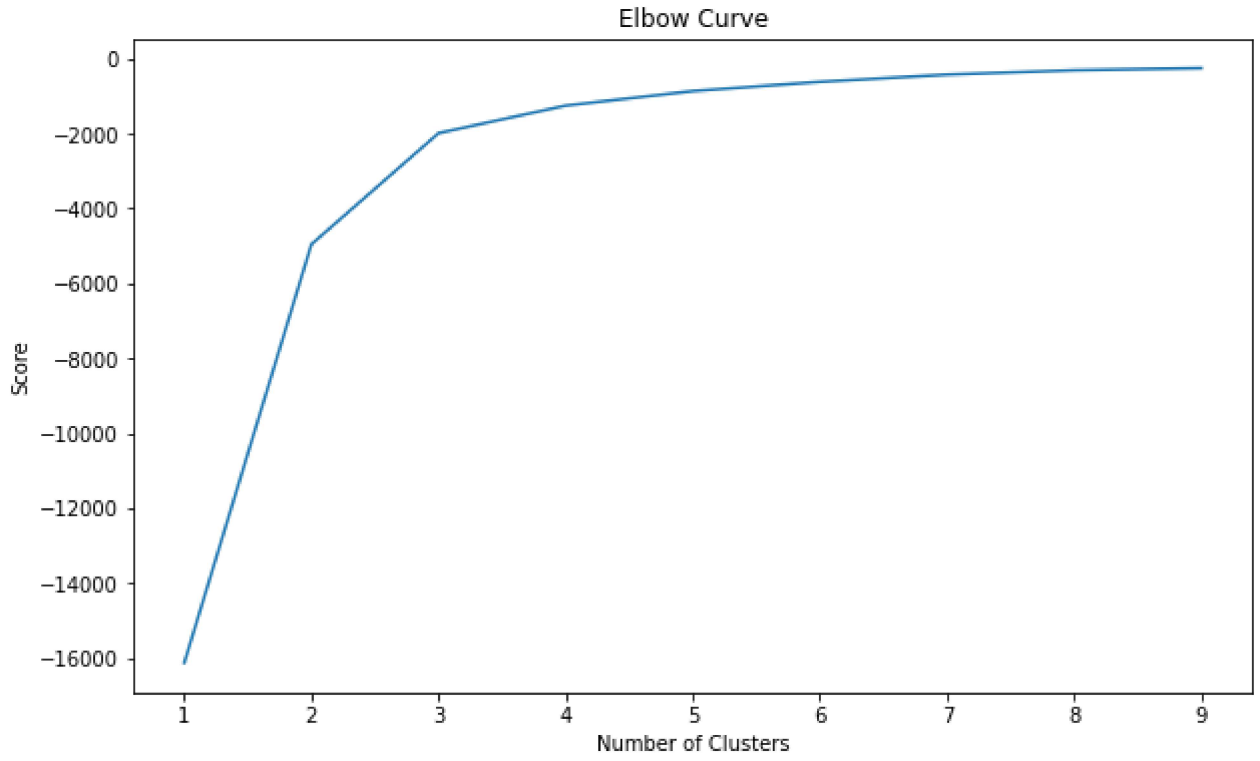



```
# info por color o ubicacion
from sklearn.cluster import KMeans


numero_de_clusters = range(1,10)
mi_kmeans = [KMeans(n_clusters=i) for i in numero_de_clusters]
Y_axis = df_ubicacion[['LATITUD']]
X_axis = df_ubicacion[['LONGITUD']]
calculo_kmeans = [mi_kmeans[i].fit(Y_axis).score(Y_axis) for i in range(len(mi_kmeans))]
```

```
plt.figure(figsize=(10,6))
plt.plot(numero_de_clusters, calculo_kmeans)
plt.xlabel('Number of Clusters')
```

Saved successfully!

```
plt.show()
```




```
X = df_sub[['LONGITUD', 'LATITUD']]

kmeans = KMeans(n_clusters=3).fit(X)
centroids = kmeans.cluster_centers_
labels = kmeans.predict(X)
C = kmeans.cluster_centers_
```

```
C_DF = pd.DataFrame(C)
C_DF["Coordinates"] = list(zip(C_DF[0], C_DF[1]))
C_DF["Coordinates"] = C_DF["Coordinates"].apply(Point)

                                                (C_DF, geometry="Coordinates")
```

Saved successfully!

	0	1	Coordinates
0	-110.794488	28.438202	POINT (-110.79449 28.43820)
1	-101.722127	22.254226	POINT (-101.72213 22.25423)
2	-90.728470	19.473362	POINT (-90.72847 19.47336)

```
df_sub['SEMAFORO'].value_counts()

     Verde       434
     Rojo        387
     Amarillo    247
     Name: SEMAFORO, dtype: int64


print(y.head())
print(df_ubicacion.head())

       SEMAFORO
     0    Verde
     1    Verde
     2     Rojo
     3    Verde
     4     Rojo
         LONGITUD   LATITUD                  COORDENADAS
     0 -102.02210  22.20887  POINT (-102.02210 22.20887)
     1 -102.20075  21.99958  POINT (-102.20075 21.99958)
     2 -102.28801  22.36685  POINT (-102.28801 22.36685)
     3 -102.29449  22.18435  POINT (-102.29449 22.18435)
     4 -110.24480  23.45138  POINT (-110.24480 23.45138)
```

```
y['SEMAPHORE'] = y['SEMAFORO'].replace(to_replace = "Verde", value = "green")
y['SEMAPHORE'].replace(to_replace = "Rojo", value = "red", inplace=True)
y['SEMAPHORE'].replace(to_replace = "Amarillo", value = "yellow", inplace=True)
y
```

Saved successfully!

0	Verde	green
1	Verde	green
2	Rojo	red
3	Verde	green
4	Rojo	red
...	...	...
1063	Rojo	red
1064	Rojo	red
1065	Rojo	red
1066	Verde	green
1067	Verde	green

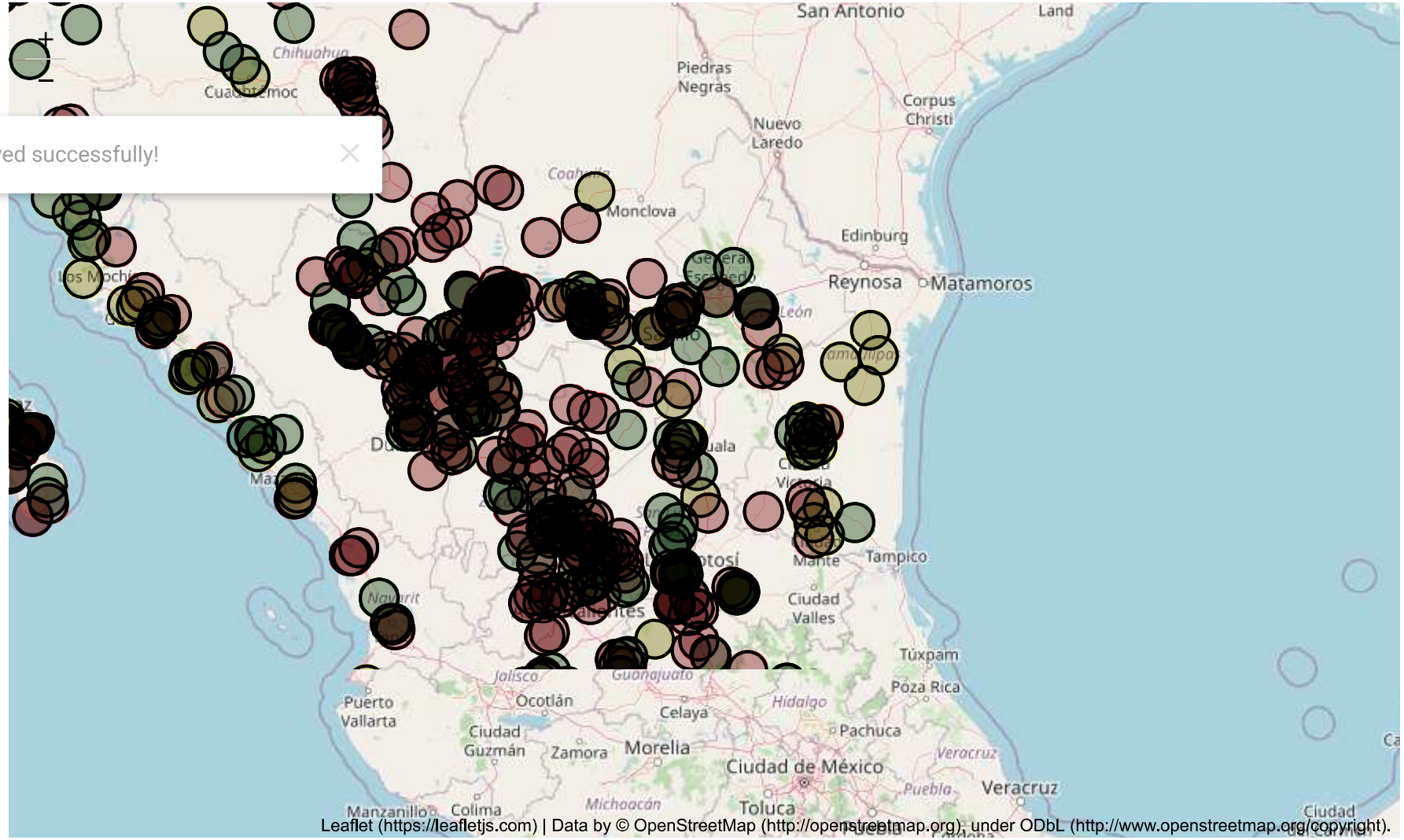
1068 rows × 2 columns

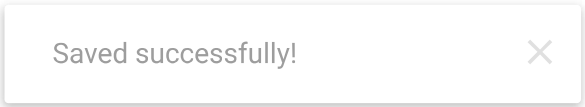
```
puntos_en_mapa['LATITUDYLONGITUD'] = puntos_en_mapa['LATITUD'] + puntos_en_mapa['LONGITUD']
diccionario_semaforo = dict(zip(puntos_en_mapa.LATITUDYLONGITUD, y.SEMAPHORE))
diccionario_semaforo

import folium
lat = puntos_en_mapa.iloc[0]['LATITUD']
lng = puntos_en_mapa.iloc[0]['LONGITUD']
map = folium.Map(location=[lng, lat], zoom_start=1)
for _, row in puntos_en_mapa.iterrows():
    folium.CircleMarker(
```

```
        location=[row["LATITUD"], row["LONGITUD"]],
        radius=12,
        weight=2,
        fill=True,
        fill_color=diccionario_semaforo[row["LATITUDYLONGITUD"]],
        color=diccionario_semaforo[row["LATITUDYLONGITUD"]]
    ).add_to(map)
color='black'
for _, row in puntos_en_mapa.iterrows():
    folium.CircleMarker(
        location=[row[1], row[0]],
        radius=12,
        weight=2,
        fill=True,
        fill_color=color,
        color=color
    ).add_to(map)
map
```

Saved successfully!

Saved successfully!





Colab paid products - Cancel contracts here

✓ 3s completed at 10:54 PM