

Maestría en Inteligencia Artificial Aplicada

Curso: Ciencia y analítica de datos

Tecnológico de Monterrey

Prof. Titular: María de la Paz Rico Fernández
Prof. Tutor: Orlando Figón Cruz.

RETO: SEMAFORO AGUA POTABLE MEXICO

Genaro Ramos Higuera - A00351269

Gerardo Aaron Castañeda Jaramillo - A01137646

AGUAS SUBTERRANEAS

Análisis, Limpieza y Pipeline

- 1068 registros, 57 columnas
- Columnas a eliminar:
 - 'SDT_mg/L' (columna vacía)
 - 'CALIDAD_SDT_ra' (informacion de uso agricola)
 - 'CUMPLE_CON_SDT_ra' (informacion de uso agricola)
 - 'PERIODO' (todas del año 2020)
 - 'CONTAMINANTES' (string valor repetido)
- Se eliminaron 14 filas (1.31%) con valores nulos para no afectar el semáforo



6 NA values in numeric variable: COND_mS/cm

	COND_mS/cm	CALIDAD_COND	CUMPLE_CON_COND	SEMAFORO
0	NaN	NaN	ND	Rojo
1	NaN	NaN	ND	Verde
2	NaN	NaN	ND	Rojo
3	NaN	NaN	ND	Amarillo
4	NaN	NaN	ND	Verde
5	NaN	NaN	ND	Verde

2 NA values in numeric variable: SDT_M_mg/L

	SDT_M_mg/L	CALIDAD_SDT	CUMPLE_CON_SDT	SEMAFORO
0	NaN	NaN	ND	Rojo
1	NaN	NaN	ND	Amarillo

1 NA values in numeric variable: DUR_mg/L

- Se elimino '<', y se resto una decimal de la misma magnitud para poder convertir de object a numeric las numéricas

```
for col in asub[num_nom].columns:
    i = 0
    for i in range(0,len(asub[col].index)):
        val = ''
        if '<' in str(asub[col][i]):
            val = asub[col][i].replace('<', '')
            dec = abs(decimal.Decimal(val).as_tuple().exponent) + 1
            val = float(val) - 1/np.power(10,dec)
            asub[col].replace(asub[col][i],val,inplace=True)

for col in asub[num_nom].columns:
    asub[col] = pd.to_numeric(asub[col])
```

Data columns (total 57 columns):

# Column	Non-Null Count	Type
0	CLAVE	1068 non-null object
1	SETTO	1068 non-null object
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_CONDUCC	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_salín	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
35	CALIDAD_PB	1068 non-null object
36	MN_TOT_mg/L	1068 non-null object
37	CALIDAD_MN	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_salín	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

AGUAS SUBTERRANEAS

Análisis, Limpieza y Pipeline



- Se utilizó `replace()` en variables categóricas de calidad para mantener sus valores ordinales. Siendo 1 la mejor calidad, y N su categoría que represente la mayor contaminación. Y `OrdinalEncoder()` para las demás.
- Se aplicaron transformaciones a las variables numéricas para intentar eliminar el sesgo positivo. Se decidió por raíz cubica.
- Se aplicó una escala de `MinMaxScaler((1,10))` para valores numéricos

```
scaled_features = MinMaxScaler((1,10)).fit_transform(asub[num_nom_cal].values)
num_scaled = pd.DataFrame(scaled_features, columns = num_nom_cal)
```

```
for col in num_nom_cal:
    trans_asub[col] = num_scaled[col]
```

```
trans_asub[num_nom_cal]
```

- Convertimos el 'SEMAFORO' de Verde, Amarillo y Rojo a 1, 2 y 3

```
asub['SEMAFORO'].replace({'Verde':1,'Amarillo':2,'Rojo':3},inplace=True)
```

- Y finalmente nos quedamos con la siguiente división de variables:

```
#definimos variables numéricas #16
num_nom_geo = ['LONGITUD','LATITUD']
num_nom_cal = ['ALC_mg/L','COND_ms/cm','SDT_mg/L','FLUO_mg/L','DUR_mg/L','CF_NMP/100_mL','NN03_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_TOT_mg/L']

#definimos variables categóricas #16
cat_nom = ['ORGANISMO_DE_CUENCA','SUBTIPO']
cat_nom_cal = ['CALIDAD_COND','CALIDAD_ALC','CALIDAD_SDT','CALIDAD_FLUO','CALIDAD_DUR','CALIDAD_CF','CALIDAD_NN03','CALIDAD_AS',
               'CALIDAD_CD','CALIDAD_CR','CALIDAD_HG','CALIDAD_PB','CALIDAD_MN','CALIDAD_FE']

#definimos variables binarias #14
bin_nom = ['CUMPLE_CON_ALC','CUMPLE_CON_COND','CUMPLE_CON_SDT','CUMPLE_CON_FLUO','CUMPLE_CON_DUR','CUMPLE_CON_CF','CUMPLE_CON_NN03','CUMPLE_CON_AS',
           'CUMPLE_CON_CR','CUMPLE_CON_HG','CUMPLE_CON_PB','CUMPLE_CON_MN','CUMPLE_CON_FE']

#VARIABLE CATEGORICA DE SALIDA Y #1
y_nom = ['SEMAFORO']
```

AGUAS SUPERFICIALES

Análisis, Limpieza y Pipeline

- 4141 registros, 54 columnas
- Columnas a eliminar:
 - 'TOX_D_48_FON_UT' (columna vacía)
 - 'CALIDAD_TOX_D_48_FON' (columna vacía)
 - 'TOX_FIS_FON_15_UT' (columna vacía)
 - 'CALIDAD_TOX_FIS_FON_15' (columna vacía)
 - 'PERIODO' (todas del año 2020)
 - 'CONTAMINANTES' (string valor repetido)
- Todas las columnas con valores nulos. Se imputaron valores numéricos mínimos.
- Se elimino '<', y se resto una decimal de la misma magnitud para poder convertir de object a numeric

```
min_vals = asup[temp_num_nom].min()
i = 0
for col in asup[temp_num_nom].columns:
    asup[col].replace(np.nan, min_vals[i], inplace=True)
    i = i + 1
```

```
for col in asup[temp_num_nom].columns:
    i = 0
    for i in range(0, len(asup[col].index)):
        val = ''
        if '<' in str(asup[col][i]):
            val = asup[col][i].replace('<', '')
            dec = abs(decimal.Decimal(val).as_tuple().exponent) + 1
            val = float(val) - 1/np.power(10, dec)
            asup[col].replace(asup[col][i], val, inplace=True)

for col in asup[temp_num_nom].columns:
    asup[col] = pd.to_numeric(asup[col])
```

- Para su respectivo valor categórico, se imputo un string 'ND' para los valores NA.

```
for col in asup[temp_cat_nom].columns:
    asup[col].replace(np.nan, 'ND', inplace=True)
```

- Se utilizo replace() en variables categóricas de calidad para mantener sus valores ordinales. Siendo 1 la mejor calidad, N su categoría que represente la mayor contaminación, y 0 el remplazo de 'ND'. Y OrdinalEncoder() para las demás.

```
for col in cal_nom[:11]:
    asup[col].replace({'Excelente':1, 'Buena calidad':2, 'Aceptable':3, 'Contaminada':4, 'Fuertemente contaminada':5, 'ND':0}, inplace=True)
for col in cal_nom[10:]:
    asup[col].replace({'No Tóxico':1, 'Toxicidad baja':2, 'Toxicidad moderada':3, 'Toxicidad alta':4, 'ND':0}, inplace=True)
```

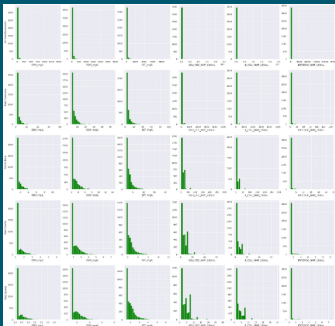
#	Column	Count	Dtype
0	CLAVE	3493 non-null	object
1	SITIO	3493 non-null	object
2	ORGANISMO_DE_CUENCA	3493 non-null	object
3	ESTADO	3493 non-null	object
4	MUNICIPIO	3493 non-null	object
5	CUENCA	3492 non-null	object
6	CUERPO DE AGUA	3479 non-null	object
7	TIPO	3493 non-null	object
8	SUBTIPO	3479 non-null	object
9	LONGITUD	3493 non-null	float64
10	LATITUD	3493 non-null	float64
11	PERIODO	3493 non-null	float64
12	DBO_mg/L	2581 non-null	object
13	CALIDAD_DBO	2581 non-null	object
14	DQO_mg/L	2581 non-null	object
15	CALIDAD_DQO	2581 non-null	object
16	SST_mg/L	3489 non-null	object
17	CALIDAD_SST	3489 non-null	object
18	COLI_FEC_NMP_100mL	2582 non-null	object
19	CALIDAD_COLI_FEC	2582 non-null	object
20	E_COLI_NMP_100mL	2582 non-null	object
21	CALIDAD_E_COLI	2582 non-null	object
22	ENTEROCO_NMP_100mL	984 non-null	object
23	CALIDAD_ENTEROC	984 non-null	object
24	OD_PORC	1797 non-null	object
25	CALIDAD_OD_PORC	1797 non-null	object
26	OD_PORC_SUP	1619 non-null	object
27	CALIDAD_OD_PORC_SUP	1619 non-null	object
28	OD_PORC_MED	487 non-null	object
29	CALIDAD_OD_PORC_MED	487 non-null	object
30	OD_PORC_FON	946 non-null	object
31	CALIDAD_OD_PORC_FON	946 non-null	object
32	TOX_D_48_UT	1816 non-null	object
33	CALIDAD_TOX_D_48	1816 non-null	object
34	TOX_V_15_UT	1819 non-null	object
35	CALIDAD_TOX_V_15	1819 non-null	object
36	TOX_D_48_SUP_UT	762 non-null	object
37	CALIDAD_TOX_D_48_SUP	762 non-null	object
38	TOX_D_48_FON_UT	0 non-null	float64
39	CALIDAD_TOX_D_48_FON	0 non-null	float64
40	TOX_FIS_SUP_15_UT	1674 non-null	object
41	CALIDAD_TOX_FIS_SUP_15	1674 non-null	object
42	TOX_FIS_FON_15_UT	0 non-null	float64
43	CALIDAD_TOX_FIS_FON_15	0 non-null	float64
44	SEMAFORO	3493 non-null	object
45	CONTAMINANTES	2226 non-null	object
46	CUMPLE_CON_DBO	3493 non-null	object
47	CUMPLE_CON_DQO	3493 non-null	object
48	CUMPLE_CON_SST	3493 non-null	object
49	CUMPLE_CON_CF	3493 non-null	object
50	CUMPLE_CON_E_COLI	3493 non-null	object
51	CUMPLE_CON_ENTEROC	3493 non-null	object
52	CUMPLE_CON_OD	3493 non-null	object
53	CUMPLE_CON_TOX	3493 non-null	object
54	GRUPO	3493 non-null	object



AGUAS SUPERFICIALES

Análisis, Limpieza y Pipeline

- Se aplicaron transformaciones a las variables numéricas para intentar eliminar el sesgo positivo. Se decidió por raíz cubica. Pero es claro el sesgo creado por imputar valores mínimos.



- Se aplico una escala de MinMaxScaler((1,10)) para valores numéricos

```
scaled_features = MinMaxScaler((1,10)).fit_transform(asup[num_nom_cal].values)
num_scaled = pd.DataFrame(scaled_features, columns = num_nom_cal)

for col in num_nom_cal:
    trans_asup[col] = num_scaled[col]

trans_asup[num_nom_cal]
```

- Convertimos el 'SEMAFORO' de Verde, Amarillo y Rojo a 1, 2 y 3

```
asup['SEMAFORO'].replace({'Verde':1,'Amarillo':2,'Rojo':3},inplace=True)
```

- Se eliminaron 13 columnas (1.32%) con valores nulos en 'SUBTIPO', para evitar afectar al semáforo imputando valores.

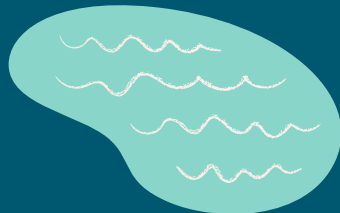
```
row_na = list()
for idx,row in asup[['SUBTIPO']].iterrows():
    if any(row.isnull()):
        row_na.append(idx)
asup.drop(row_na, inplace=True)
asup.reset_index(drop=True,inplace=True)
```

- Y finalmente nos quedamos con la siguiente división de variables:

```
#definimos variables numéricas
num_nom_geo = ['LONGITUD','LATITUD']
num_nom_cal = ['DBO_mg/L','DQO_mg/L','SST_mg/L','COLI_FEC_NMP_100mL','E_COLI_NMP_100mL','ENTEROC_NMP_100mL','OD_PORC','OD_PORC_SUP',
               'OD_PORC_MED','OD_PORC_FON','TOX_D_48_UT','TOX_V_15_UT','TOX_D_48_SUP_UT','TOX_FIS_SUP_15_UT']

#definimos variables categóricas
cat_nom = ['ORGANISMO_DE_CUENCA','SUBTIPO','GRUPO']
cat_nom_cal = ['CALIDAD_DBO','CALIDAD_DQO','CALIDAD_SST','CALIDAD_COLI_FEC','CALIDAD_E_COLI','CALIDAD_ENTEROC','CALIDAD_OD_PORC','CALIDAD_OD_PORC_SUP',
               'CALIDAD_OD_PORC_MED','CALIDAD_OD_PORC_FON','CALIDAD_TOX_D_48','CALIDAD_TOX_V_15','CALIDAD_TOX_D_48_SUP','CALIDAD_TOX_FIS_SUP_15']

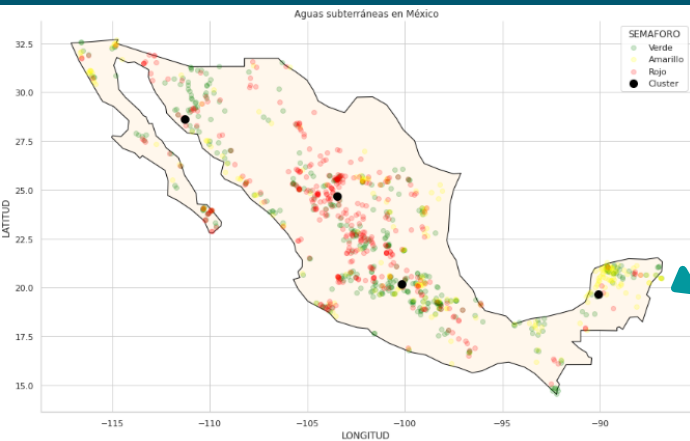
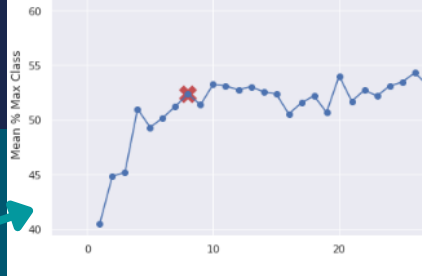
#definimos variables binarias
bin_nom = ['CUMPLE_CON_DQO','CUMPLE_CON_SST','CUMPLE_CON_CF','CUMPLE_CON_E_COLI','CUMPLE_CON_ENTEROC','CUMPLE_CON_OD','CUMPLE_CON_TOX']
#VARIABLE CATEGORICA DE SALIDA Y
y_nom = ['SEMAFORO']
```



AGUAS SUBTERRANEAS

KMeans

- Método de ELBOW arroja 4 clusters. Sin embargo, es en base a INERTIA. Buscamos una relación entre clusters y el semáforo.
- Iteramos N clusters vs. Promedio de la clase maxima en clusters. Observamos que el ultimo aumento significativo es en 8 clusters y que el promedio no es muy alto.



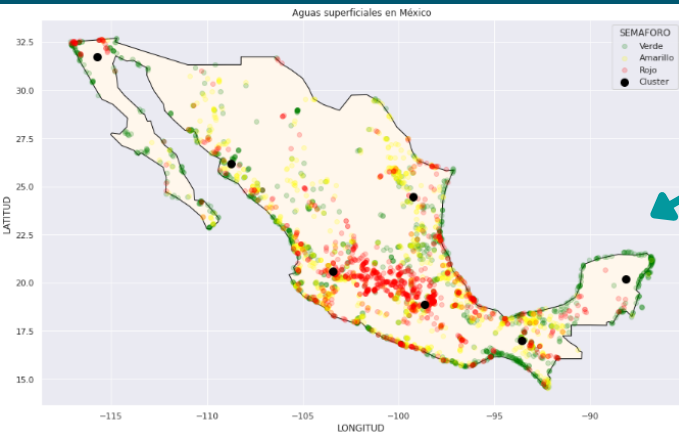
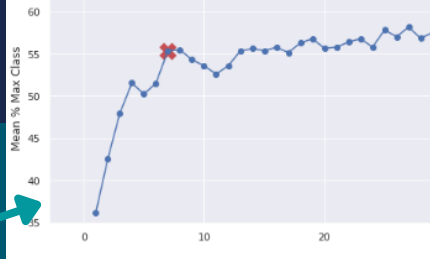
- No se observa ninguna clase con máximo particularmente alto en ningún cluster.
- Regresamos al numero optimo de 4 para INERTIA y graficamos.
- Concluimos que no existe relación significativa entre ubicación y semáforo.

Cluster 0
Class=1, n=60 (58.252%)
Class=3, n=30 (29.126%)
Class=2, n=13 (12.621%)
Cluster 1
Class=2, n=84 (52.830%)
Class=1, n=61 (38.365%)
Class=3, n=14 (8.805%)
Cluster 2
Class=1, n=99 (44.000%)
Class=3, n=100 (44.444%)
Class=2, n=26 (11.556%)
Cluster 3
Class=1, n=31 (29.524%)
Class=3, n=38 (36.190%)
Class=2, n=36 (34.286%)
Cluster 4
Class=1, n=86 (53.086%)
Class=2, n=38 (23.457%)
Class=3, n=38 (23.457%)
Cluster 5
Class=3, n=123 (68.333%)
Class=1, n=49 (27.222%)
Class=2, n=8 (4.444%)
Cluster 6
Class=3, n=15 (31.250%)
Class=2, n=22 (45.833%)
Class=1, n=11 (22.917%)
Cluster 7
Class=3, n=24 (33.333%)
Class=1, n=30 (41.667%)
Class=2, n=18 (25.000%)

AGUAS SUPERFICIALES

KMeans

- Método de ELBOW arroja 5 clusters. Sin embargo, es en base a INERTIA. Buscamos una relación entre clusters y el semáforo.
- Iteramos N clusters vs. Promedio de la clase maxima en clusters. Observamos que el ultimo aumento significativo es en 7 clusters y que el promedio no es muy alto.



- Se observa únicamente para el cluster 2 una clase con 88.62% en una clase.
- Graficamos los 7 clusters y observamos que el 2 esta en la península de Yucatán.
- Concluimos que para todo México no existe relación significativa entre ubicación y semáforo.

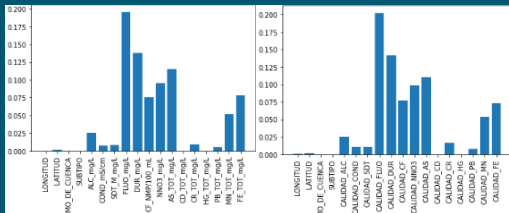
Cluster 0
Class=2, n=189 (47.607%)
Class=1, n=134 (33.753%)
Class=3, n=74 (18.640%)
Cluster 1
Class=1, n=87 (65.414%)
Class=2, n=5 (3.759%)
Class=3, n=41 (30.827%)
Cluster 2
Class=1, n=148 (88.623%)
Class=2, n=4 (2.395%)
Class=3, n=15 (8.982%)
Cluster 3
Class=2, n=182 (33.151%)
Class=1, n=268 (48.816%)
Class=3, n=99 (18.033%)
Cluster 4
Class=3, n=472 (47.059%)
Class=2, n=326 (32.502%)
Class=1, n=205 (20.439%)
Cluster 5
Class=3, n=333 (40.959%)
Class=1, n=205 (25.215%)
Class=2, n=275 (33.825%)
Cluster 6
Class=1, n=212 (50.839%)
Class=3, n=57 (13.669%)
Class=2, n=148 (35.492%)



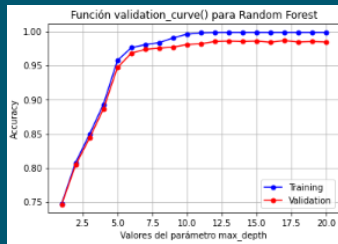
AGUAS SUBTERRANEAS

Clasificador Optimo

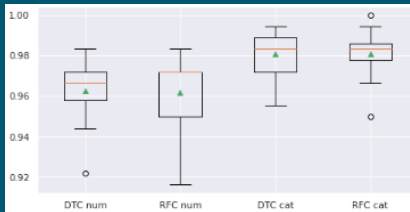
- Aplicando feature importances para los dos modelos, se encontraron las mismas variables con importancia nula en variables numéricas y categóricas. Por lo que no se utilizaron para el modelo.



- Se creó una validation curve variando max_depth para los 4 modelos. En base a ella se hicieron los gridsearchcv:



- Se graficaron en boxplot cada split del resultado del crossvalidation del gridsearch.



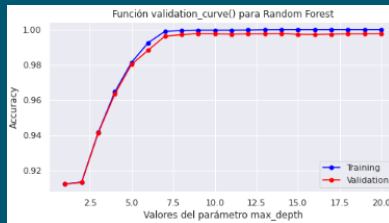
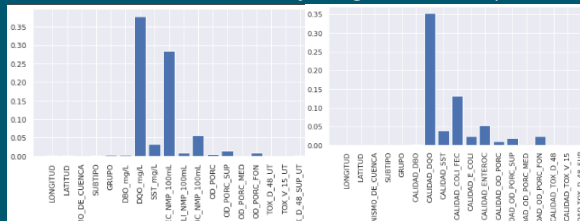
- Modelo óptimo fue **Random Forest con variables categóricas**. Tiene menor varianza en sus resultados, y arroja métricas con 100% para los valores de prueba:

```
Accuracy: 1.0  
F1-score: 1.0  
Precision: 1.0  
Recall: 1.0
```

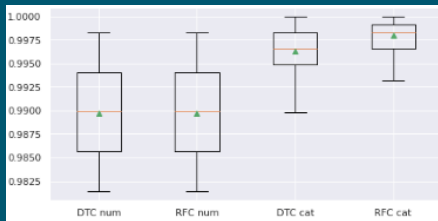

Clasificador Optimo



- Se creo una validation curve variando max_depth para los 4 modelos. En base a ella se hicieron los gridsearchcv:



- Se graficaron en boxplot cada split del resultado del crossvalidation del gridsearch.



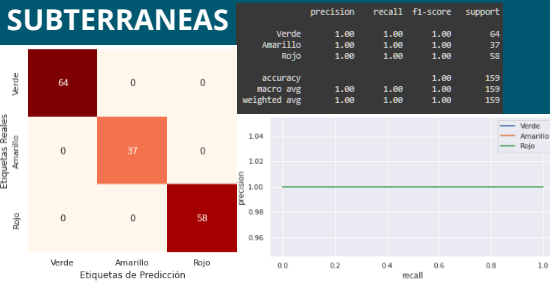
- Modelo optimo fue **Random Forest con variables categóricas**. Tiene menor varianza en sus resultados, y arroja métricas con 96.62% para los valores de prueba:

```
Accuracy: 0.9961685823754789
F1-score: 0.9961685823754789
Precision: 0.9961685823754789
Recall: 0.9961685823754789
```

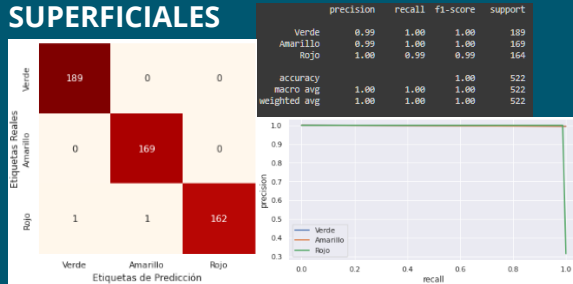
AGUAS SUBTERRANEAS Y SUPERFICIALES

Exactitud y Conclusiones

SUBTERRANEAS



SUPERFICIALES



Random Forest con datos de prueba:

- Reporte de clasificación arroja 100%
- 100% curva de precision-recall
- 0 errores en matriz de confusión

Random Forest con datos de prueba:

- Reporte de clasificación arroja 99%, casi 100%
- Curva de precision-recall cercana a 100%
- Matriz de confusión:
 - Errores de Clase Semáforo VERDE: FP=1, FN=0
 - Errores de Clase Semáforo AMARILLO: FP=1, FN=0
 - Errores de Clase Semáforo ROJO: FP=0, FN=2

- No existe relación considerable entre ubicación geográfica y calidad del agua en aguas subterráneas ni superficiales.
- El mejor modelo para aguas subterráneas fue Random Forest con las variables categóricas de calidad. Se obtuvo accuracy del 100% con datos de prueba
- El mejor modelo para aguas superficiales fue Random Forest con variables categóricas de calidad. Se obtuvo accuracy del 99.61% con datos de prueba