Segunda Entrega del Proyecto

Por:
Edwin David Duque Grajales cc. 1017255650 Ing Ambiental
Peter Daniel Garrido Rodriguez cc. 1085952515 Ing. Industrial
Jheison Andres Benavides Rincon cc.1015466242 Ing. Civil

Materia:
Introducción a la Inteligencia Artificial para las Ingenierías

Profesor:
Raúl Ramos Pollan

Universidad de Antioquia
Facultad de Ingeniería
Medellín 2023

INTRODUCCIÓN

La inteligencia artificial es una herramienta que nos permite modelar y procesar información que se puede utilizar para resolver problemáticas y desafíos que se presentan diariamente, en este caso los algoritmos de Machine Learning nos permiten procesar información para crear modelos de aprendizaje automático que utilicen datos de emisiones de fuente abierta (de observaciones del satélite Sentinel-5P) para predecir las emisiones de carbono. La aplicación de esta herramienta a buen término plantea una opción para interpretar las observaciones de otros satélites para las mismas variables o con otros términos de referencia , donde se hagan estimaciones , cálculos, proyecciones y análisis de los fenómenos de las emisiones de carbono y los datos observados.

Planteamiento del problema

Analizar los fenómenos de la naturaleza en los últimos años se ha vuelto una prioridad para aprender del pasado y en base a esto realizar estimaciones y proyecciones del posible futuro. Las emisiones de carbono en la actualidad son de interés científico, económico y social, ya que identificar las emisiones presentes en el entorno y las variables que las ocasionan facilita plantear y mejorar soluciones para reducirlas hasta el punto que su reducción ayude con la problemática del calentamiento global y al mismo tiempo implementar medidas puede mejorar la economía de varios sectores por su valor monetario en bolsa o programas que se encargan de reducir huellas de carbono a cambio de recursos económicos, ambientales y sociales.

Dataset

El dataset a utilizar proviene de una competencia de Kaggle en la cual se proporcionan datos que provienen de la selección de aproximadamente 497 ubicaciones únicas de múltiples áreas de Ruanda, distribuidas entre tierras agrícolas, ciudades y plantas de energía.

Se extrajeron siete características principales semanalmente de Sentinel-5P desde enero de 2019 hasta noviembre de 2022. Cada característica (dióxido de azufre, monóxido de carbono, etc.) contiene subcaracterísticas cómo column number density, qué es la densidad de la columna vertical a nivel del suelo, calculada utilizando la técnica DOAS. Puede leer más sobre cada característica en los enlaces a continuación, incluido cómo se miden y las definiciones de las variables. Se le proporcionan los valores de estas características en el conjunto de prueba y su objetivo de predecir las emisiones de CO2 utilizando información de tiempo además de estas características.

- Dióxido de azufre - COPERNICUS/S5P/NRTI/L3_SO2
- Monóxido de carbono - COPERNICUS/S5P/NRTI/L3_CO
- Dióxido de nitrógeno - COPERNICUS/S5P/NRTI/L3_NO2
- Formaldehído - COPERNICUS/S5P/NRTI/L3_HCHO
- Índice de aerosoles UV - COPERNICUS/S5P/NRTI/L3_AER_AI
- Ozono - COPERNICUS/S5P/NRTI/L3_O3
- Nube - COPERNICUS/S5P/OFFL/L3_CLOUD

Inicialmente se hizo énfasis en la exploración y limpieza de los datos suministrados. En la exploración identificamos que el tamaño del dataset presenta 79023 observaciones y 76 características para cada observación.

```
[ ]  # Tamaño del dataset
     print (d.shape)

     (79023, 76)
```

secuencialmente identificamos los valores estadísticos de los datos del estudio

```
#Valores estadisticos de las columnas
d._get_numeric_data().describe().T
```

	count	mean	std	min	25%	50%	75%	max
latitude	79023.0	-1.891072	0.694522	-3.299000	-2.451000	-1.882000	-1.303000	-0.510000
longitude	79023.0	29.880155	0.810375	28.228000	29.262000	29.883000	30.471000	31.532000
year	79023.0	2020.000000	0.816502	2019.000000	2019.000000	2020.000000	2021.000000	2021.000000
week_no	79023.0	26.000000	15.297155	0.000000	13.000000	26.000000	39.000000	52.000000
SulphurDioxide_SO2_column_number_density	64414.0	0.000048	0.000272	-0.000996	-0.000096	0.000024	0.000153	0.004191
...	...	...	...	...	...	...	...	...
Cloud_sensor_azimuth_angle	78539.0	-10.784832	30.374462	-102.739731	-30.309170	-12.673914	9.402202	78.223037
Cloud_sensor_zenith_angle	78539.0	40.436976	6.428216	2.998873	35.829907	41.119630	44.446272	65.951248
Cloud_solar_azimuth_angle	78539.0	-86.800583	37.837269	-153.464211	-125.991158	-84.644352	-48.132701	-22.653170
Cloud_solar_zenith_angle	78539.0	27.925981	4.403835	10.818288	24.686763	28.333630	31.499883	42.060436
emission	79023.0	81.940552	144.299648	0.000000	9.797995	45.593445	109.549595	3167.768000

75 rows × 8 columns

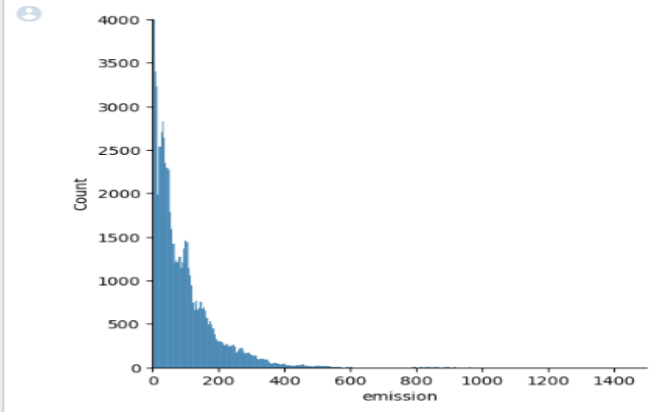
Esta operación nos permite identificar de manera clara los datos y nos obliga a tomar decisiones para la limpieza de datos. El paso que sigue es identificar los datos faltantes que presenta nuestro dataset y presentarlo en porcentaje con el fin de identificar la cantidad de datos faltantes que tiene nuestro dataset. Esto es un punto que nos va permitir filtrar datos cuando definamos las columnas categóricas del proyecto.

```
#Porcentaje de datos faltantes para cada variable del dataset
total          = d.isnull().sum().sort_values(ascending=False)
percent        = (d.isnull().sum()/d.isnull().count()*100).sort_values(ascending=False)
missing_train1 = pd.concat([total,percent],axis=1,keys=["Total","Percent"])
missing_train1
```

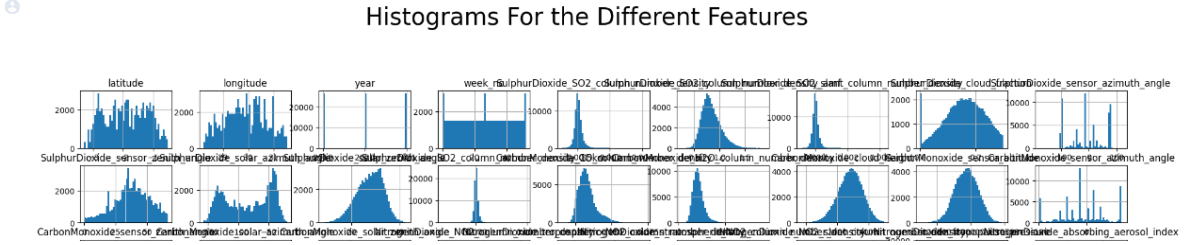
	Total	Percent
UvAerosolLayerHeight_aerosol_height	78584	99.444466
UvAerosolLayerHeight_solar_zenith_angle	78584	99.444466
UvAerosolLayerHeight_solar_azimuth_angle	78584	99.444466
UvAerosolLayerHeight_sensor_azimuth_angle	78584	99.444466
UvAerosolLayerHeight_aerosol_pressure	78584	99.444466
...	...	...

Ya con la información de las variables y la clasificación de los datos de nuestro dataset se realiza una representación de la variable principal y de las variables de las diferentes columnas para identificar el comportamiento de los datos y más adelante identificar los datos faltantes que son significativos para el comportamiento de las variables.

```
#Distribucion de la variable objetivo
sns.displot(d['emission'])
plt.xlim(0, 1500)
plt.ylim(0, 4000)
plt.show()
```



```
#Histogramas con los comportamientos de las diferentes columnas
d.hist(bins=50, figsize=(24,15))
plt.suptitle('Histograms For the Different Features', fontsize=30)
plt.show()
```



Para darle poder de significancia a los datos del dataset hacemos la matriz de correlaciones de las variables involucradas con la variable objetivo en este caso la emisión de carbono.

```
#Correlaciones de las variables con la variable objetivo
target_corr_df = pd.DataFrame(corr_matrix["emission"].sort_values(ascending=False))
target_corr_df
```

	emission
emission	1.000000
longitude	0.102746
UvAerosolLayerHeight_aerosol_height	0.069008
Cloud_surface_albedo	0.046587
Formaldehyde_tropospheric_HCHO_column_number_density_amf	0.040263
...	...
Formaldehyde_tropospheric_HCHO_column_number_density	-0.033333
NitrogenDioxide_solar_azimuth_angle	-0.033417
CarbonMonoxide_CO_column_number_density	-0.041328
CarbonMonoxide_H2O_column_number_density	-0.043217
UvAerosolLayerHeight_aerosol_pressure	-0.068138

75 rows × 1 columns

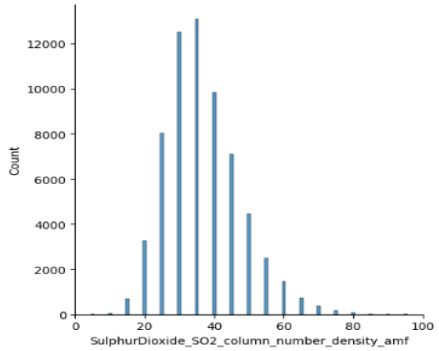
Después de esta exploración de datos, procedemos a limpiar los datos que tengan faltantes, sin embargo se definen las columnas categóricas y observamos su comportamiento gráfico, este procedimiento se realiza para las columnas categóricas que están establecidas.

Generamos algunas columnas categoricas

```
[ ] list_catgr = ["SulphurDioxide_SO2_column_number_density_amf", "CarbonMonoxide_CO_column_number_density", '
    d_01 = d.copy()

    for i in list_catgr:
      intervalo = np.linspace(d_01[i].min(), d_01[i].max(), 20).tolist()
      d_01[i] = pd.cut(d_01[i], bins=intervalo, labels=list(range(5,100,5)))
```

```
for i in list_catgr:
  sns.displot(d_01[i])
  plt.show()
```



Continuando se procede a segmentar esos datos de tal manera que escojamos aquellos que tengan un porcentaje de datos faltantes menor al 10%, esto con el fin de cumplir los

criterios establecidos del proyecto y utilizar los datos que más nos convengan para evaluar el modelo con respecto a las columnas categóricas.

Hora debemos rellenar los datos que siguen estando faltantes

```
[ ] #Porcentaje de datos faltantes para cada variable del dataset
    total          = d_02.isnull().sum().sort_values(ascending=False)
    percent        = (d_02.isnull().sum()/d_02.isnull().count()*100).sort_values(ascending=False)
    missing_train1 = pd.concat([total,percent],axis=1,keys=["Total","Percent"])
    missing_train1
```

	Total	Percent
Formaldehyde_tropospheric_HCHO_column_number_density	7278	9.209977
Formaldehyde_tropospheric_HCHO_column_number_density_amf	7277	9.208711
Formaldehyde_sensor_zenith_angle	7277	9.208711
Formaldehyde_solar_azimuth_angle	7277	9.208711
Formaldehyde_solar_zenith_angle	7277	9.208711
Formaldehyde_cloud_fraction	7277	9.208711
Formaldehyde_HCHO_slant_column_number_density	7277	9.208711

Para finalizar escogemos el parámetro para resolver los datos faltantes, lo cual se lleva a cabo reemplazando el dato faltante por la moda del grupo de datos, con el fin de representar los datos de tal manera que no cambien de manera significativa en estos casos y no representan cambios drásticos de acuerdo a la ubicación que se encuentran el dataset.

```
[ ] # Completar los valores faltantes en las columnas restantes con la moda
    for feature in d_02.columns:
        if d_02[feature].isnull().sum() > 0:
            mode_value = d_02[feature].mode()[0] # Calcular la moda de la columna
            d_02[feature].fillna(mode_value, inplace=True)

    print(d_02.head)
```

```
<bound method NDFrame.head of            ID_LAT_LON_YEAR_WEEK  latitude  longitude  year  week_no  \
0      ID_-0.510_29.290_2019_00    -0.510     29.290  2019        0
1      ID_-0.510_29.290_2019_01    -0.510     29.290  2019        1
2      ID_-0.510_29.290_2019_02    -0.510     29.290  2019        2
3      ID_-0.510_29.290_2019_03    -0.510     29.290  2019        3
4      ID_-0.510_29.290_2019_04    -0.510     29.290  2019        4
...                         ...       ...        ...   ...      ...
79018  ID_-3.299_30.301_2021_48    -3.299     30.301  2021       48
79019  ID_-3.299_30.301_2021_49    -3.299     30.301  2021       49
79020  ID_-3.299_30.301_2021_50    -3.299     30.301  2021       50
79021  ID_-3.299_30.301_2021_51    -3.299     30.301  2021       51
79022  ID_-3.299_30.301_2021_52    -3.299     30.301  2021       52
```

Bibliografía

- Communications. (2021, marzo 9). ¿Qué es el dióxido de carbono (CO2) y cómo impacta en el planeta? BBVA. https://www.bbva.com/es/sostenibilidad/que-es-el-dioxido-de-carbono-co2-y-como-impacta-en-el-planeta/
- Datacleaning Limpieza de datos: definición, importancia. (2022, abril 7). Formation Data Science | Datascientest.com. https://datascientest.com/es/datacleaning-limpieza-de-datos-definicion-tecnicas-importancia-en-data-science
- Differential Optical Absorption Spectroscopy, 2011. Recuperado el 22 de septiembre de 2023, de http://chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/http://repositorio.gestiondelriesgo.gov.co/bitstream/20.500.11762/20564/3/Fenomeno_nino-2016.pdf
- Predict CO2 emissions in Rwanda. (s/f). Kaggle.com. Recuperado el 22 de octubre de 2023, de https://www.kaggle.com/competitions/playground-series-s3e20/data
- Wikipedia contributors. (s/f). *Error absoluto medio*. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=Error_absoluto_medio&oldid=146234007