INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

Segunda entrega: Predicción de las emisiones de CO2

Presentado a: Raul
Ramos Pollan

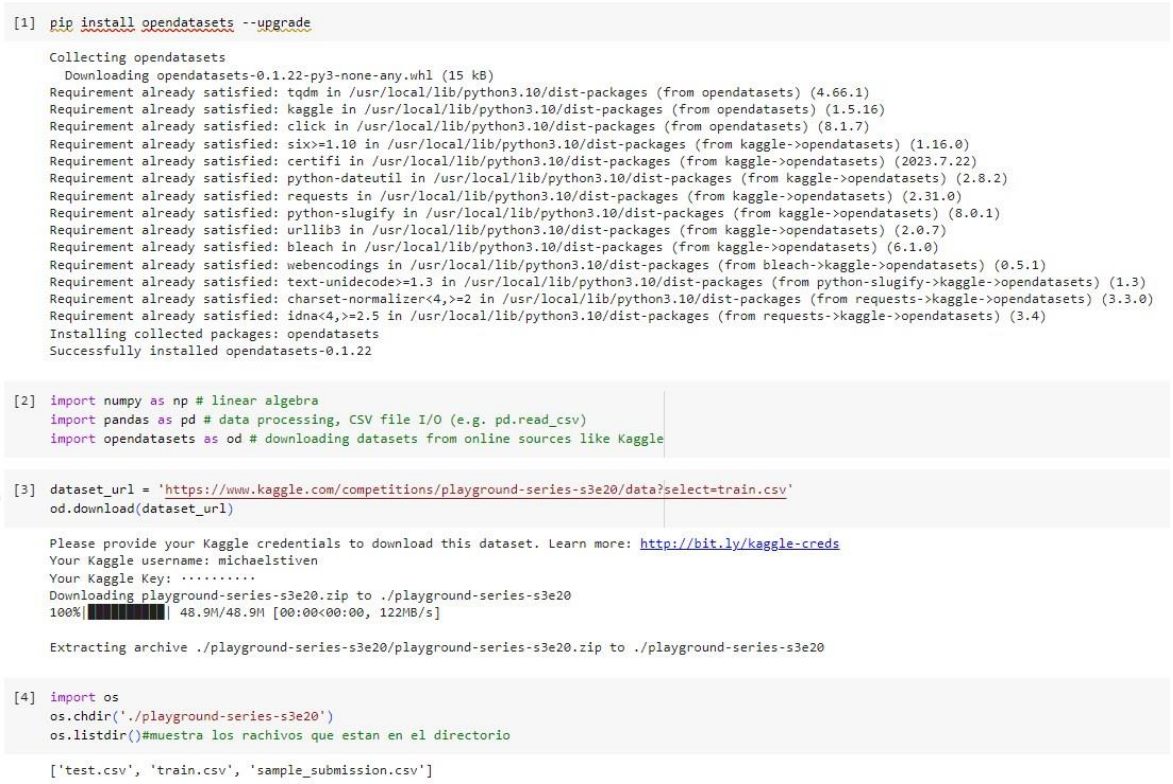
Presentado por:
Michael Stiven Zapata Giraldo
1017172358
Estudiante

Daniel Andres Vergara de Leon
1002388181
Estudiante

Universidad De Antioquia
Facultad de Ingeniería
Medellín
2023

Nuestro objetivo con este proyecto es predecir las emisiones de CO2 en lugares sobre todo industriales, con muestras como el dióxido de azufre y monóxido de carbono. Este objetivo se inicia a partir de utilizar factores de tiempo y técnicas como la DOAS (**Differential Optical Absorption Spectroscopy).**

Lo primero que se hizo en esta etapa fue la conexión del dataset de Kaggle con el Google Collab. En Kaggle se descargó el archivo tipo JSON que contiene el usuario y la contraseña para poder empezar a trabajar en Google Collab y poder acceder a los datos. En el Collab se instaló la librería opendatasets y luego la importamos. Después de eso se agregó el link de la competencia a otro módulo del Collab para poder iniciar sesión con el usuario y la clave suministrados en archivo JSON que previamente se descargó. En el módulo de "os" agregamos el link de Kaggle perteneciendo al tema de **Predict CO2 Emissions in Rwanda,** y nos muestra los archivos que hay en ese directorio. Todo esto se evidencia en la **figura 1**.



```
[1] pip install opendatasets --upgrade

    Collecting opendatasets
      Downloading opendatasets-0.1.22-py3-none-any.whl (15 kB)
    Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from opendatasets) (4.66.1)
    Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (from opendatasets) (1.5.16)
    Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from opendatasets) (8.1.7)
    Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (1.16.0)
    Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (2023.7.22)
    Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (2.8.2)
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (2.31.0)
    Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (8.0.1)
    Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (2.0.7)
    Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle->opendatasets) (6.1.0)
    Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle->opendatasets) (0.5.1)
    Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle->opendatasets) (1.3)
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle->opendatasets) (3.3.0)
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle->opendatasets) (3.4)
    Installing collected packages: opendatasets
    Successfully installed opendatasets-0.1.22

[2] import numpy as np # linear algebra
    import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
    import opendatasets as od # downloading datasets from online sources like Kaggle

[3] dataset_url = 'https://www.kaggle.com/competitions/playground-series-s3e20/data?select=train.csv'
    od.download(dataset_url)

    Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
    Your Kaggle username: michaelstiven
    Your Kaggle Key: ··········
    Downloading playground-series-s3e20.zip to ./playground-series-s3e20
    100%|██████████| 48.9M/48.9M [00:00<00:00, 122MB/s]

    Extracting archive ./playground-series-s3e20/playground-series-s3e20.zip to ./playground-series-s3e20

[4] import os
    os.chdir('./playground-series-s3e20')
    os.listdir()#muestra los rachivos que estan en el directorio

    ['test.csv', 'train.csv', 'sample_submission.csv']
```

Figura 1. Módulos de los datos cargados desde Kaggle.

Las características principales para hallar estas predicciones son siete:

- Sulphur Dioxide
- Carbon Monoxide
- Nitrogen Dioxide
- Formaldehyde
- UV Aerosol Index

- Ozone
- Cloud

Las subdivisiones de las anteriores características, se tratan como variables independientes para la predicción, estas subdivisiones suman en total 70 columnas La columna final corresponde a la tabla de predicción de emisión del CO2: emissio, para 71 columnas y al inicio del dataset están 5 columnas:

1. ID_LAT_LON_YEAR_WEEK
2. latitude
3. longitude
4. year
5. week_no

Para un total de 76 columnas, del dataset de Kaggle. Para el modelo de predicción se evitó la columna de **ID_LAT_LON_YEAR_WEEK**, pues es un string y para poder utilizar la función de regresión lineal se debe trabajar con datos numéricos, en este caso se trabajó con valores tipo float64, se evidencia en la **figura 3**.

```
[ ] Formaldehyde_sensor_zenith_angle                          float64
    Formaldehyde_sensor_azimuth_angle                         float64
    dtype: object

[ ] print('\nBloque 5 de Índice de aerosoles UV(UV Aerosol Index)):')
    data5.dtypes

    Bloque 5 de Índice de aerosoles UV(UV Aerosol Index)):
    UvAerosolIndex_absorbing_aerosol_index        float64
    UvAerosolIndex_sensor_altitude                float64
    UvAerosolIndex_sensor_azimuth_angle           float64
    UvAerosolIndex_sensor_zenith_angle            float64
    UvAerosolIndex_solar_azimuth_angle            float64
    UvAerosolIndex_solar_zenith_angle             float64
    UvAerosolLayerHeight_aerosol_height           float64
    UvAerosolLayerHeight_aerosol_pressure         float64
    UvAerosolLayerHeight_aerosol_optical_depth    float64
    UvAerosolLayerHeight_sensor_zenith_angle      float64
    UvAerosolLayerHeight_sensor_azimuth_angle     float64
    UvAerosolLayerHeight_solar_azimuth_angle      float64
    UvAerosolLayerHeight_solar_zenith_angle       float64
    dtype: object

[ ] print('\nBloque 6 de Ozono (Ozone)):')
    data6.dtypes

    Bloque 6 de Ozono (Ozone)):
    Ozone_O3_column_number_density          float64
    Ozone_O3_column_number_density_amf      float64
    Ozone_O3_slant_column_number_density    float64
    Ozone_O3_effective_temperature          float64
    Ozone_cloud_fraction                    float64
    Ozone_sensor_azimuth_angle              float64
    Ozone_sensor_zenith_angle               float64
    Ozone_solar_azimuth_angle               float64
    Ozone_solar_zenith_angle                float64
    dtype: object
```

Figura 2. Tipos de datos trabajados en el dataset de Kaggle.

Primer Modelo Predictivo (Regresión lineal)

```
[ ]  modelo=LinearRegression()
     modelo.fit(X=data8[variables_independientes],y=data8[variable_objetivo])
```

```
▾ LinearRegression
LinearRegression()
```

Figura 3. Modelo de regresión.

La **figura 3** muestra el método de regresión lineal para la predicción de emisiones de CO2, cuyo resultado se puede ver en la última columna de la **figura 4.** Esto es una primera propuesta de modelo, ya que se buscará otros métodos para acertar un modelo con mejores resultados y el error de medidas y comparaciones sea lo más mínimo posible.

```
[ ]  data8[['emission','Emission_predict']]
```

	emission	Emission_predict
0	3.750994	108.061534
1	4.025176	71.530307
2	4.231381	59.330956
3	4.305286	96.039428
4	4.347317	79.531064
...	...	...
79018	29.404171	69.712717
79019	29.186497	101.025614
79020	29.131205	102.425027
79021	28.125792	63.228106
79022	27.239302	78.245672

79023 rows × 2 columns

Figura 4. Primera propuesta del modelo de predicciones.

```
Root Mean Squared Error
142.25429866076868
```

Figura 5. Error cuadrático medio primer modelo predictivo.

En la **figura 5** se hace evidente que el error cuadrático medio se encuentra significativamente por encima del valor objetivo, lo que confirma la necesidad de explorar otro tipo de modelo predictivo con el fin de obtener resultados más precisos y satisfactorios.

Segundo Modelo Predictivo (Regresión de bosques aleatorios)

```
#Dividimos entre train set y de test set
X_train, X_test, y_train, y_test = train_test_split(X, y_true, test_size=0.2, random_state=42)

# Entrenando el Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```
# Haciendo prediccion en el test set
y_pred = rf_model.predict(X_test)
```

Figura 5. Modelo de Regresión de bosques aleatorios.

En la **figura 5** podemos ver el modelo de regresión de bosques aleatorios implementado en el collab de google, los resultados de este se verán en la **figura 6** haciendo uso del error cuadrático medio entre los datos medidos y los datos reales.

```
Root Mean Squared Error
126.68841504976845
```

Figura 6. Error cuadrático medio segundo modelo predictivo.

Al observar la **figura 6**, se puede notar que el segundo modelo predictivo ha mostrado una mejora con respecto a los resultados anteriores. A pesar de este avance, aún no se ha logrado alcanzar un valor aceptable de error cuadrático medio según los datos en consideración. Por consiguiente, se continuará la búsqueda de otros modelos predictivos con el fin de mejorar los resultados de manera significativa.