PROYECTO_ENTREGA 2

Es importante mencionar que nuestro proyecto consiste en la clasificación de 72 clases de cobertura del suelo, para lo cual se mapeo la cubierta terrestre de Europa (por ejemplo, Bare Land ,Grassland without tree/shrub cover, Coniferous woodland, Potatoes, Vineyards). proceso que se realizó en cinco años diferentes (2006, 2009, 2012, 2015, 2018). utilizando como insumo un conjunto de datos que incluye más de 42,237 muestras in situ y 416 características derivadas de datos de teledetección e imágenes satelitales.

En este caso se deberá predecir la clase de cobertura del suelo en un conjunto de pruebas que contiene alrededor de 42,271 muestras que no están presentes en el conjunto de datos de entrenamiento.

DESCRIPCIÓN DE LOS AVANCES

En nuestro caso lo primero que hicimos fue revisar la información proporcionada en la base de datos, para esto abrimos cada componente de la base de datos la cual está compuesta por los archivos train.csv, test.csv, sample_summision. Para entender de mejor el contenido de los archivos utilizamos la aplicación excel con la cual pudimos analizar la información en los archivos de la base de datos.

Para este informe creamos un notebook de colab para cada miembro del equipo y lo enlazamos con el drive correspondiente a traves del siguiente código:

```
from google.colab import drive

drive.mount('/content/drive')
```

En los respectivos drives creamos una carpeta llamada Entrega2 la cual iba a contener toda la información del proyecto. Al interior de la carpeta Entrega2 creamos otra carpeta llamada kaggel, en la cual descargamos la base de datos necesaria para el desarrollo de nuestro proyecto nombrado en la competencia de Kaggel como: oemc-hackathon-eu-land-cover-classification.

Luego subimos la información anterior al notebook, para ello fue necesario descargar las credenciales de la competición que elegimos en kaggel y subirla a la carpeta de nuestro drive para lo cual utilizamos el código

```
import os

os.environ['KAGGLE_CONFIG_DIR'] = '/content/drive/MyDrive/Entrega2/kaggle'
```

con el cual utilizamos la biblioteca Pandas para cargar y manipular datos de un archivo CSV ubicado en una ruta específica en Google Drive. luego usamos el código:

```
%cd /content/drive/MyDrive/Entrega2/kaggle
```

cambiar el directorio de trabajo actual a la ruta especificada. En este caso, la ruta especificada es: `/content/drive/MyDrive/Entrega2/kaggle`. y utilizamos `!ls` para mostrar los contenidos del directorio actual. para descargar los datos de una competición específica llamada "oemc-hackathon-eu-land-cover-classification" utilizamos el código:

```
!kaggle competitions download -c oemc-hackathon-eu-land-cover-classification
```

para crear un directorio llamado "land_cover" y luego mover un archivo llamado "oemc-hackathon-eu-land-cover-classification.zip" a ese directorio se utilizaron los comandos

```
!mkdir land_cover
```

```
!mv oemc-hackathon-eu-land-cover-classification.zip land_cover
```

luego descomprimimos los archivos zip que se descargaron en el directorio usamos el comando

```
!unzip oemc-hackathon-eu-land-cover-classification.zip
```

para leer los archivos utilizamos los comandos:

```
import pandas as pd
```

```
train = pd.read_csv('/content/drive/MyDrive/Entrega2/kaggle/land_cover/train.csv')
```

```
train
```

```
sample_submission = pd.read_csv('sample_submission.csv')
```

```
sample_submission
```

```
import pandas as pd
```

```
test = pd.read_csv('test.csv')
```

```
test
```

No hemos avanzado más porque estamos investigando como eliminar la información de celdas al azar, para cumplir con el requisito del 10% de datos faltantes.

Después de investigar tenemos una idea de como eliminar la información de algunas celdas al azar en un DataFrame de Pandas y es utilizando la función **sample** para seleccionar aleatoriamente las filas o columnas que deseemos y luego modificar sus valores. A través del siguiente código pero aun no lo logramos debemos investigar que error estamos cometiendo.

```
import pandas as pd

import numpy as np  # Importa NumPy para generar valores aleatorios


# Define la fracción de celdas que deseas eliminar (por ejemplo, 10%)

fraccion_a_eliminar = 0.1


# Calcula el número total de celdas en el DataFrame

total_celdas = train.size


# Calcula la cantidad de celdas a eliminar

celdas_a_eliminar = int(total_celdas * fraccion_a_eliminar)


# Genera una lista de índices aleatorios para seleccionar celdas al azar

indices_aleatorios = np.random.choice(train.index, size=celdas_a_eliminar, replace=False)


# Itera sobre los índices aleatorios y establece el valor en esas celdas como NaN (o cualquier otro valor que elijas)

for indice in indices_aleatorios:

    columna = np.random.choice(train.columns)  # Elije una columna aleatoriamente

    train.at[indice, columna] = np.nan  # Establece el valor como NaN
```