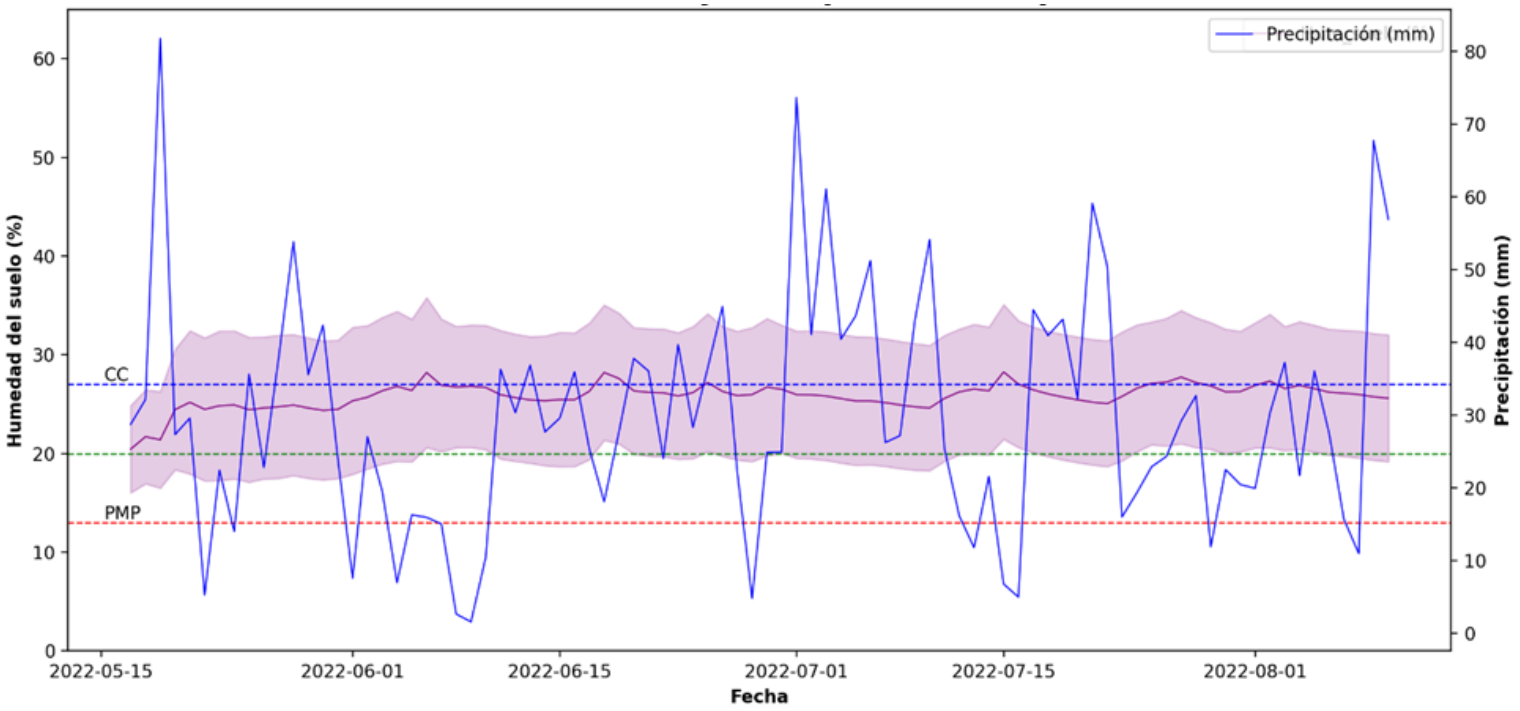

AgTech 19043: Digitalización de la Agricultura de Pequeña Escala

Producto 10. Nota técnica de rutinas de procesamiento y análisis de datos

Oscar Hernan Estrada Vargas

2023

FONTAGRO

Códigos JEL: Q16

FONTAGRO (Fondo Regional de Tecnología Agropecuaria) es un mecanismo único de cooperación técnica entre países de América Latina, el Caribe y España, que promueve la competitividad y la seguridad alimentaria. Las opiniones expresadas en esta publicación son de los autores y no necesariamente reflejan el punto de vista del Banco Interamericano de Desarrollo (BID), del Instituto Interamericano de Cooperación para la Agricultura (IICA), FONTAGRO, de sus directorios ejecutivos ni de los países que representan.

El presente documento ha sido preparado por Oscar Hernan Estrada Vargas.

Esta publicación puede solicitarse a:
FONTAGRO
Correo electrónico: fontagro@fontagro.org
www.fontagro.org

Tabla de Contenidos

Listado de Figuras

Resumen

La descripción y análisis de las bases de datos provenientes de los sensores de humedad de suelo instalados durante el año 2022 en Colombia, Honduras y Nicaragua, en el marco del proyecto “Digitalización de la Agricultura a Pequeña Escala” financiado por FONTAGRO, fueron realizados usando rutinas de programación en Python. Para el análisis descriptivo se generaron gráficos lineales en forma de series temporales, donde se relacionaron las curvas de los datos de humedad y precipitación, esta última obtenida de fuentes de información secundaria (Copernicus). Se graficaron también los datos de humedad del suelo agrupados por tipo de cobertura vegetal usada en el lote, como series de tiempo, para observar diferencias. Posteriormente se realizó un análisis de aprendizaje automático de bosque aleatorio para encontrar patrones y relaciones entre los datos colectados, graficando las relevancias de las variables y sus dependencias parciales, sin embargo, el desempeño del modelo no fue suficiente debido a la poca variabilidad en los datos y la baja correspondencia entre las variables.

Palabras Clave: Humedad del suelo, sensor, precipitación, temperatura, serie temporal, rutina, Python, machine learning.

Abstract

The description and analysis of the databases from the soil moisture sensors installed during the year 2022 in Colombia, Honduras and Nicaragua, within the framework of the "Digitalization of Small-Scale Agriculture" project financed by FONTAGRO, were carried out using programming routines in Python. For the descriptive analysis, linear graphs were generated in the form of time series, where the curves of the humidity and precipitation data were related, the latter obtained from secondary information sources (Copernicus). Soil moisture data grouped by type of plant cover used in the plot, as time series, were also plotted to observe differences. Subsequently, a machine learning (random forest) analysis was carried out to find patterns and relationships between the collected data, plotting the relevance of the variables and their partial dependencies, however, the performance of the model was not sufficient, due to the low variability in the data and the low correspondence between the variables.

Keywords: soil moisture, sensor, precipitation, temperature, time series, routine, Python, machine learning.

Introducción

El proyecto Agtech 19043 de Digitalización de la agricultura de pequeña escala desarrolló una solución tecnológica para medir humedad de suelo, que es robusta, de bajo costo, y alta usabilidad. La validación del dispositivo se hizo en campo, con productores, mediante una metodología participativa. Como parte de esta metodología se recolectó información de las prácticas agronómicas implementadas con los productores, adicionalmente a la información de humedad de suelo recolectada por los dispositivos. Este documento presenta la rutina de procesamiento de los datos obtenidos de los dispositivos, información de las prácticas de manejo, e información secundaria, en un esfuerzo por generar conocimiento adicional que pueda ser compartido con los productores. Se presenta tanto los códigos utilizados en el análisis, como los resultados obtenidos. En la medida de los posibles, se presenta la rutina de procesamiento como si fuera un guion (script). Los datos utilizados están disponibles en el repositorio digital del proyecto, que puede ser accedido en este enlace: https://github.com/gonzalezdeleon19/-Proyecto-Fontagro-Digitalizaci-n-de-la-agricultura-a-peque-a-escala

Datos

Las bases de datos analizadas contienen la información de humedad de suelo de los dispositivos instalados en Colombia, Honduras y Nicaragua, y de las prácticas de manejo en los cultivos donde se instalaron los dispositivos, además de los datos de las variables climáticas provenientes de fuentes de información secundarias. Los datos están comprendidos en el periodo de tiempo entre mayo y noviembre del año 2022. Los datos de humedad son datos horarios para cada sitio, y para el caso de los datos de clima, son datos diarios extraídos según las coordenadas donde se instalaron los dispositivos. Es importante mencionar que los datos colectados por los sensores de humedad de suelos son generados en formato CSV.

Software

Los datos fueron procesados y analizados utilizando el lenguaje de programación Python, versión 3.10.9, a través del entorno de desarrollo Spyder versión 5.4.2.

Procesamiento y análisis

Las rutinas de procesamiento están enfocadas en la depuración, corrección y transformación de la información, para obtener una base de datos unificada, en un formato adecuado para las siguientes etapas del proceso. A partir de esta base de datos se inició la fase de análisis, la cual se

dividió en una primera tarea de análisis exploratorio, para entender el comportamiento de los datos, y una segunda tarea de análisis con técnicas de aprendizaje automático (mejor conocido como Machine Learning), para encontrar patrones y correlaciones entre los datos de humedad y las demás variables.

Rutinas

Procesamiento y análisis exploratorio

Paso 1: el primer bloque de este script carga las librerías de Python necesarias para el procesamiento y la visualización de los datos, y define el directorio de trabajo:

```
"""
Created on Wed Feb  8 10:26:29 2023
@author: OEstrada
"""
import os
import numpy as np
import pandas as pd
import rasterio
from rasterio.plot import show
import geopandas as gpd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import missingno as msno
import matplotlib.pyplot as plt
import seaborn as sns
os.chdir(r'D:\OneDrive - CGIAR\CIAT\[2022] FONTAGRO')
```

Paso 2: a continuación, se procesa la base de datos de los sensores instalados en Colombia, para lo cual se lee la hoja de Excel que contiene dicha información, se filtran las variables de interés y se extraen las coordenadas de ubicación de los sensores:

```
data_col = pd.read_excel('Consolidado Promedio Diario_Trespaises + Datos manejo.xlsx',
            sheet_name='Colombia_')
names = list(data_col)
names
cols = ['Codigo_Sensor', 'Fecha', 'WVC_promedio', 'WVC_Max', 'WVC_Min', 'Precip_Chirps', 'Precip_Coper',
      'Radiacion_Acumula', 'Humedad_Relativa', 'Temperatura', 'Latitud_Y', 'Longitud_X',
      'Altitud', 'Cultivo', 'practica_consevacion_agua', 'Tipo_riego ', 'Practica_drenaje',
      'Practicas_conservacion_suelo', 'Coberturas_vegetal', 'Rendimiento']
data_col = data_col[cols]
data_col = data_col.rename(columns={'Tipo_riego ':'Tipo_riego'})
```

```
data_col.insert(1,'Pais', 'Colombia')
# Extraer coordenadas por sensor
textura_col = data_col.dropna(subset='Longitud_X')
textura_col = textura_col.drop_duplicates(subset=['Codigo_Sensor'])
textura_col = textura_col[['Codigo_Sensor', 'Longitud_X', 'Latitud_Y']]
```

Las coordenadas se utilizan también para extraer los datos de contenidos de arenas, limos y arcillas de 3 archivos tipo raster (.tif), descargados de la plataforma SoilGrids (https://soilgrids.org/) para el área de estudio:

```
coordenadas = textura_col.loc[:,['Longitud_X', 'Latitud_Y']].to_numpy().tolist()
feature = ['Arenas', 'Arcillas', 'Limos']
for i in feature:
    raster = rasterio.open(f'{i}.tif')
    show(raster)
    valores = pd.Series(dato for dato in raster.sample(coordenadas, indexes=1))
    textura_col[i] = 0
    for j in valores.index:
        textura_col[i].iloc[j]=valores[j]
    raster.close()
```

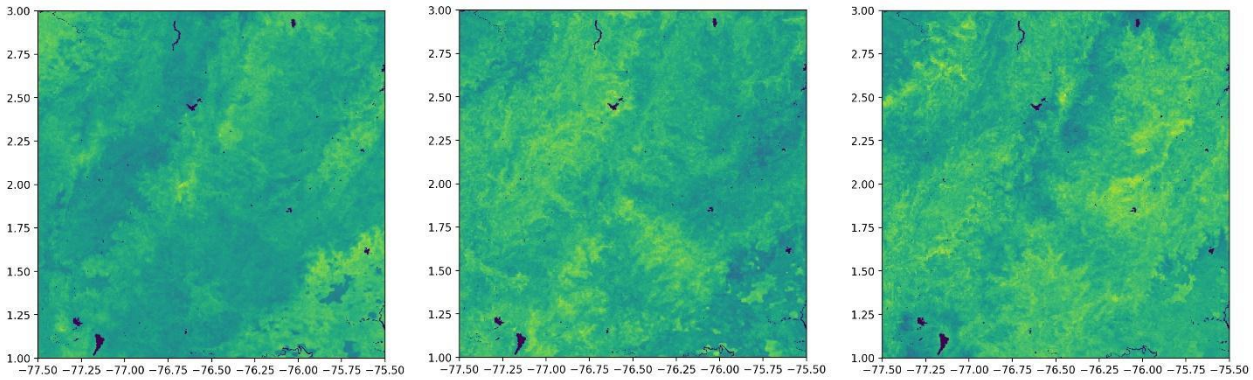

Figura 1. Rasters de contenido de arenas, limos y arcillas para el departamento del Cauca - Colombia (SoilGrids).

En este punto se clasifica la textura de suelo en la ubicación del sensor a partir de los porcentajes de las partículas del suelo (para el caso de Colombia, comparando preliminarmente estos porcentajes extraídos de SoilGrids, se identificó que todos los sensores se ubicaron en suelos con textura Franco Arcillosa, por lo que solo fue necesario evaluar este caso en el script), se crea una columna en el set de datos con esta variable y se guardan las coordenadas para graficar posteriormente las ubicaciones:

```
# Asignar tipo de textura dependiendo de los porcentajes (x10) segun clasificacion USDA
textura_col.loc[(textura_col['Arenas'] > 200) & (textura_col['Arenas'] <= 450) &
```

```
        (textura_col['Arcillas'] > 270) & (textura_col['Arcillas'] <= 400) &
        (textura_col['Limos'] > 150) & (textura_col['Limos'] <= 520),
        ['Textura_15cm']] = 'FrancoArcilloso'
# Union de columna Textura a los datos
data_col = pd.merge(data_col, textura_col[['Codigo_Sensor','Textura_15cm']], on='Codigo_Sensor', how='left')
# Coordenadas para graficar
coord_col = data_col.drop_duplicates(subset=['Codigo_Sensor'])
```

Paso 3: el siguiente bloque de código se encarga de procesar la base de datos provenientes de Honduras, leyendo la hoja de Excel que los contiene, filtrando las variables de interés y extrayendo las coordenadas de ubicación de los sensores:

```
data_hon = pd.read_excel('Consolidado Promedio Diario_Trespaises + Datos manejo.xlsx',
            sheet_name='Honduras')
names = list(data_hon)
names
cols = ['Codigo_Sensor', 'Fecha', 'WVC_promedio', 'WVC_Max', 'WVC_Min', 'Precip_Chirps', 'Precip_Coper',
    'Radiacion_Acumula', 'Humedad_Relativa', 'Temperatura', 'Profundidad_instalacion',
    'Textura_15cm', 'Textura_30 cm', 'Pendiente', 'Cobertura', 'Cobertura.1',
    'Latitud', 'Longitud', 'Altitud', 'Cultivo', 'practica_consevacion_agua',
    'Tipo _riego', 'Practica_drenaje', 'Practicas_conservacion_suelo',
    'Coberturas_vegetal', 'Siembra_Cama.1', 'Rendimiento']
data_hon = data_hon[cols]
data_hon = data_hon.rename(columns={'Tipo _riego':'Tipo_riego'})
data_hon.insert(1,'Pais', 'Honduras')
# Coordenadas para graficar
coord_hon = data_hon.drop_duplicates(subset=['Codigo_Sensor'])
```

Paso 4: la siguiente sección en el script procesa la base de datos provenientes de Nicaragua, leyendo la hoja de Excel que los contiene, filtrando las variables de interés y extrayendo las coordenadas de ubicación de los sensores:

```
data_nic = pd.read_excel('Consolidado Promedio Diario_Trespaises + Datos manejo.xlsx',
            sheet_name='Nicaragua')
names = list(data_nic)
names
cols = ['Codigo_Sensor', 'Fecha', 'WVC_promedio', 'WVC_Max', 'WVC_Min', 'Precip_Chirps', 'Precip_Coper',
    'Radiacion_Acumula', 'Humedad_Relativa', 'Temperatura', 'Profundidad_instalacion',
    'Textura_15cm', 'Textura_30 cm', 'Pendiente', 'Cobertura', 'Cobertura.1',
    'Siembra_Cama', 'Latitud ', 'Longitud', 'Altitud', 'Cultivo', 'practica_consevacion_agua',
    'tipo_riego', 'Practica_drenaje', 'criterios_riego', 'Practicas_conservacion_suelo',
    'Coberturas_vegetal', 'Rendimiento']
data_nic = data_nic[cols]
data_nic = data_nic.rename(columns={'tipo_riego':'Tipo_riego'})
data_nic.insert(1, 'Pais', 'Nicaragua')
# Coordenadas para graficar
coord_nic = data_nic.drop_duplicates(subset=['Codigo_Sensor'])
```

Posteriormente se descartan las variables innecesarias a partir de este punto del entorno de programación:

```
del cols, coordenadas, feature, i, j, names, raster, textura_col, valores
```

Paso 5: la siguiente etapa del script de procesamiento consiste en unificar en un solo conjunto de datos la información de los tres países, eligiendo las variables que se tendrán en cuenta para la exploración:

```
# Describir datos
print(data_col.describe())
print(data_hon.describe())
print(data_nic.describe())

# Concatenar los datos en un solo dataframe
data = pd.concat([data_col[['Codigo_Sensor', 'Pais', 'Fecha', 'Precip_Chirps', 'Precip_Coper',
                  'Temperatura', 'Textura_15cm', 'Tipo_riego', 'Practica_drenaje',
                  'Coberturas_vegetal', 'Cultivo', 'WVC_promedio']],
            data_hon[['Codigo_Sensor', 'Pais', 'Fecha', 'Precip_Chirps', 'Precip_Coper',
                  'Temperatura', 'Profundidad_instalacion', 'Textura_15cm',
                  'Tipo_riego', 'Practica_drenaje', 'Coberturas_vegetal',
                  'Cultivo', 'WVC_promedio']],
            data_nic[['Codigo_Sensor', 'Pais', 'Fecha', 'Precip_Chirps', 'Precip_Coper',
                  'Temperatura', 'Profundidad_instalacion', 'Textura_15cm',
                  'Tipo_riego', 'Practica_drenaje', 'Coberturas_vegetal',
                  'Cultivo', 'WVC_promedio']]])
```

Paso 6: luego se realiza un proceso de limpieza de datos basado en el reporte recibido por los consultores en campo, descartando totalmente la información proveniente de los sensores que se averiaron y eliminando los días que presentaron lecturas atípicas en los demás sensores (valores inferiores a 5% o por encima de 60% de humedad):

```
# Eliminar informacion de sensores con fallos
remover = ['M216', 'M224', 'M229', 'M231', 'M233', 'M235', 'M165', 'M206','M181',
      'M172', 'M188', 'M192', 'M196', 'M197']
data = data[~data['Codigo_Sensor'].isin(remover)]
# Recortar dias a sensores con lecturas erroneas
sens_filtro = data[data['Codigo_Sensor'] == 'M159']
data = data.drop(sens_filtro[sens_filtro['Fecha'] > '2022-08-18'].index)
sens_filtro = data[data['Codigo_Sensor'] == 'M162']
data = data.drop(sens_filtro[sens_filtro['Fecha'] > '2022-07-31'].index)
# Eliminar registros de humedad superiores a 60%
data = data[data['WVC_promedio'] <= 60]
```

Paso 7: la siguiente sección del código se encarga de seleccionar la mejor fuente de información

para los datos de precipitación, realizando un análisis de regresión lineal simple entre los valores de humedad y los datos provenientes de CHIRPS y Copernicus:

```
# Evaluar cual fuente de precipitacion descartar mediante regresion lineal
data_test = data.dropna(subset=['Precip_Chirps', 'Precip_Coper', 'WVC_promedio'])
X = np.array(data_test['Precip_Chirps']).reshape(-1, 1)
y = np.array(data_test['WVC_promedio']).reshape(-1, 1)
# Evaluacion de CHIRPS
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
model = LinearRegression()
model.fit(X_train, y_train)
predicciones = model.predict(X_test)
print(f'RMSE CHIRPS: {mean_squared_error(y_test, predicciones, squared=False)}')
# Evaluacion de Copernicus
X = np.array(data_test['Precip_Coper']).reshape(-1, 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=1)
model = LinearRegression()
model.fit(X_train, y_train)
predicciones = model.predict(X_test)
print(f'RMSE COPERNICUS: {mean_squared_error(y_test, predicciones, squared=False)}')
# Descartar precipitacion de Chirps
data = data.drop('Precip_Chirps', axis=1)
```

Paso 8: posteriormente se realiza un ajuste a los nombres de las variables y se eliminan los registros con información faltante, examinando visualmente la composición de los datos mediante la librería “missingno”:

```
# Cambiar los nombres de las variables
data.columns = ['Cod_sens', 'Pais', 'Fecha', 'Precip', 'Temp', 'Text_15cm',
        'Riego', 'Drenaje', 'Cobertura', 'Cultivo', 'Hum_sens', 'Prof_sens']
# Eliminar registros con valores faltantes, excepto en 'Prof_sens'
msno.matrix(data);
```

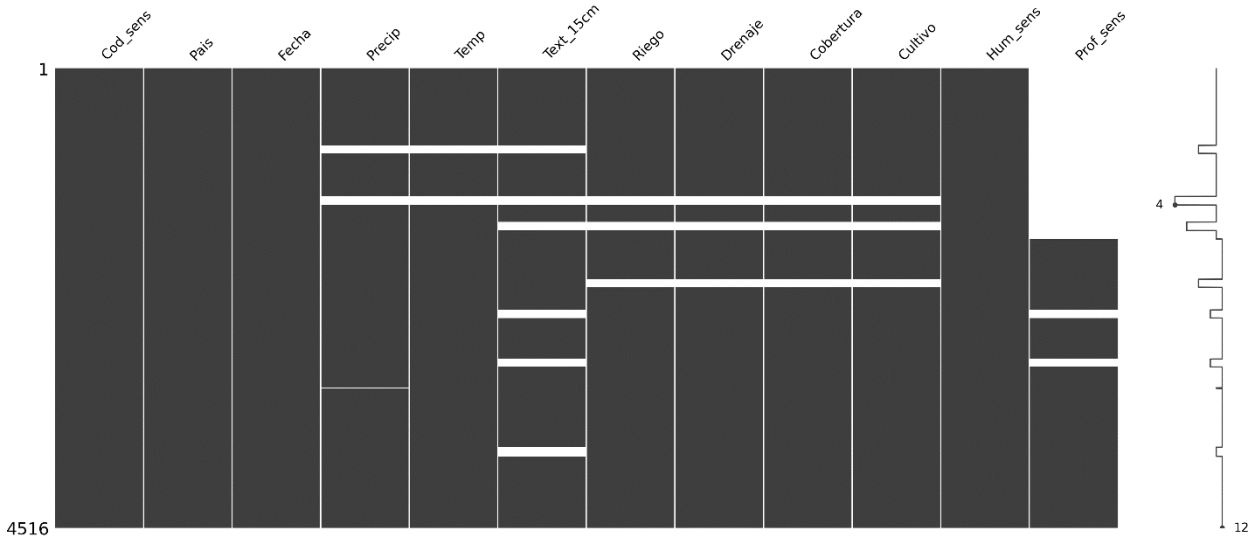

Figura 2. Visualización del set de datos para identificar valores faltantes.

Se realiza la eliminación de registros exceptuando la variable “Profundidad del sensor”, ya que para Colombia no se registró esta información:

```
data = data.dropna(subset=['Cod_sens', 'Pais', 'Fecha', 'Precip', 'Temp', 'Text_15cm',
        'Riego', 'Drenaje','Cobertura', 'Cultivo', 'Hum_sens'])
msno.matrix(data);
```

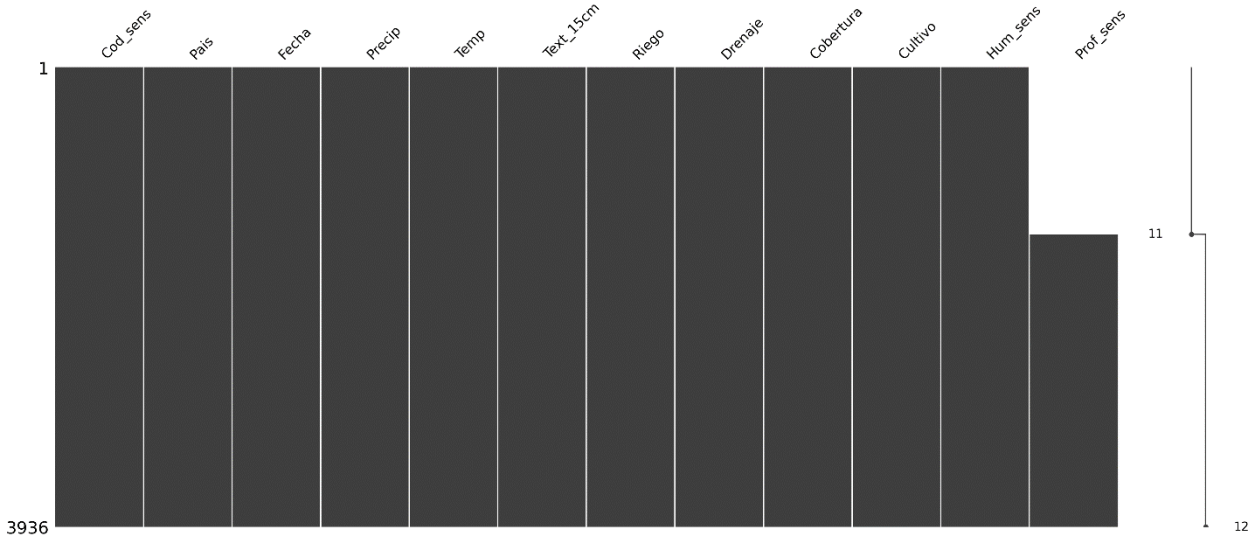

Figura 3. Visualización del set de datos después de eliminar valores faltantes.

Paso 9: la siguiente sección del script realiza la conversión de la variable “Temperatura” de grados Kelvin a Celsius, corrige categorías que incluyen espacios en blanco o utilizan tildes o mayúsculas, y agrupa las opciones de textura de suelo para obtener solo 5 categorías:

```
# Convertir grados Kelvin a Celsius
data['Temp'] = data['Temp'] - 273.15
# Unificar categorias de cultivo
data['Cultivo'].value_counts()
dicc = {'Maíz': 'Maiz', 'Café': 'Cafe', 'Plátano': 'Platano', 'frijol': 'Frijol',
      'caña_panelera': 'Caña_Panelera'}
data = data.replace({'Cultivo': dicc}, regex=True)
# Modificacion de las categorias de drenaje
data['Drenaje'].value_counts()
data['Drenaje'] = data['Drenaje'].replace({'si': 'Si', 'no': 'No', ' ': 'No'})
# Modificacion de las categorias de cobertura vegetal
data['Cobertura'].value_counts()
data['Cobertura'] = data['Cobertura'].replace(
   {'si_vivas': 'Cob_vivas', 'si_muertas': 'Cob_muertas', 'no_realiza': 'Sin_cobert'})

# Unificar categorias en textura de suelo
data['Text_15cm'].value_counts()
data.loc[data['Text_15cm'].str.startswith('FrancoArci'), ['Text_15cm']] = 'FrancoArcilloso'
data.loc[data['Text_15cm'].str.startswith('FrancoAre'), ['Text_15cm']] = 'FrancoArenoso'
data.loc[data['Text_15cm'].str.startswith('Arcillo'), ['Text_15cm']] = 'Arcilloso+'
data.loc[data['Text_15cm'].str.startswith('Franco '), ['Text_15cm']] = 'Franco'
data.drop(data[data['Text_15cm'] == 'Limoso'].index, inplace = True)
```

En la agrupación por texturas, debido a que algunas categorías contenían pocos datos, se unificaron los tipos de textura Franco Arcilloso (Far), Franco Arcilloso Limoso (FArL) y Franco Arcilloso Arenoso (FArA) como Franco Arcilloso, de igual manera se unificaron como Arcilloso+ los tipos de textura Arcilloso Arenoso (ArA) y Arcilloso Limoso (ArL).

Paso 10: la etapa de procesamiento termina con la exploración de la base de datos final, la exportación de esta a formato “.csv” para posteriores análisis y la eliminación de variables innecesarias del entorno de programación:

```
data.info()
print(data.describe())
data['Cod_sens'].describe()
# Guardar base de datos para analisis de ML
data.to_csv('data_sens.csv', index=False)
del data_col, data_hon, data_nic, data_test, remover, sens_filtro, dicc, model
del predicciones, X, X_test, X_train, y, y_test, y_train
```

Paso 11: el siguiente bloque de código del script genera los gráficos para el análisis exploratorio de los datos, iniciando con la creación de la carpeta para guardar las imágenes:

```
# Definir ruta para guardar gráficos
path = os.getcwd()
ruta = path + '/Graficos/'
if not os.path.isdir(ruta):
   os.makedirs(ruta)
```

Paso 12: posteriormente se grafica la ubicación de los sensores en cada país, cargando los "shapefiles" de los territorios y ubicando los sensores con las coordenadas extraídas anteriormente en el script:

```
# Coordenadas Colombia
archivo_shapefile = r'D:\OneDrive - CGIAR\CIAT\[2022]
        FONTAGRO\Shapefiles\SUDAMERICA_ADM2\sudamerica_adm2.shp'
shapefile = gpd.read_file(archivo_shapefile)
shapefile = shapefile[(shapefile['ADM0'] == 'COLOMBIA') & (shapefile['ADM1'] == 'Cauca')]
fig, ax = plt.subplots(figsize=(10, 10))
shapefile.plot(ax=ax, edgecolor='black', facecolor='none')
ax.scatter(coord_col['Longitud_X'], coord_col['Latitud_Y'])
ax.set_title('Colombia - Cauca', size=16, fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '1_Sensores_Colombia.png')
plt.show();
```

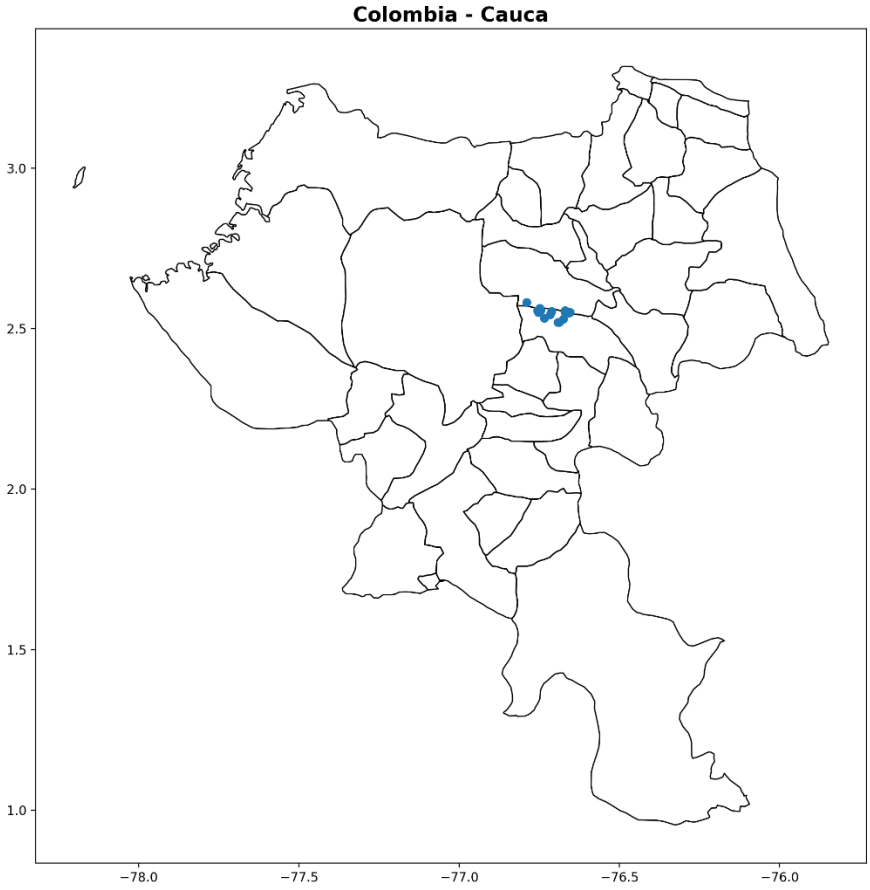

Figura 4. Ubicación de los sensores en el departamento del Cauca – Colombia.

```
# Coordenadas Honduras
archivo_shapefile = r'D:\OneDrive - CGIAR\CIAT\[2022] FONTAGRO\
        Shapefiles\hnd_adm_sinit_20161005_shp\hnd_admbnda_adm2_sinit_20161005.shp'
shapefile = gpd.read_file(archivo_shapefile)
shapefile = shapefile[shapefile['ADM1_ES'] == 'Francisco Morazan']
fig, ax = plt.subplots(figsize=(10, 10))
shapefile.plot(ax=ax, edgecolor='black', facecolor='none')
ax.scatter(coord_hon['Longitud'], coord_hon['Latitud'])
ax.set_title('Honduras - Francisco Morazán', size=16, fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '2_Sensores_Honduras.png')
plt.show();
```

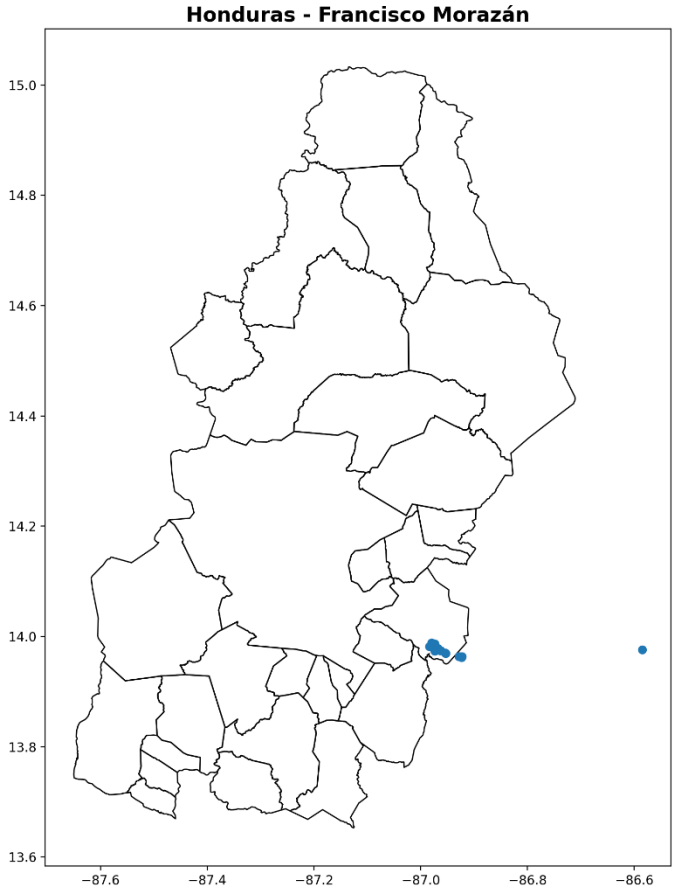

Figura 5. Ubicación de los sensores en el departamento de Francisco Morazán – Honduras.

En este gráfico se observa un punto fuera de la zona y obedece a un error de lectura del GPS que registró una longitud errónea para esta ubicación, la cual debe estar cerca a los demás sensores.

```
# Coordenadas Nicaragua
archivo_shapefile = r'D:\OneDrive - CGIAR\CIAT\[2022] FONTAGRO\
        Shapefiles\municipios_50k_nicaragua_2015\municipios_50k_nicaragua_2015.shp'
shapefile = gpd.read_file(archivo_shapefile)
shapefile = shapefile[(shapefile['departamen'] == 'JINOTEGA') | (shapefile['departamen'] == 'ESTELI') |
        (shapefile['departamen'] == 'NUEVA SEGOVIA')]
fig, ax = plt.subplots(figsize=(10, 10))
shapefile.plot(ax=ax, edgecolor='black', facecolor='none')
ax.scatter(coord_nic['Longitud'], coord_nic['Latitud '])
ax.set_title('Nicaragua - Estelí, Nueva Segovia y Jinotega', size=16, fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '3_Sensores_Nicaragua.png')
plt.show();
```

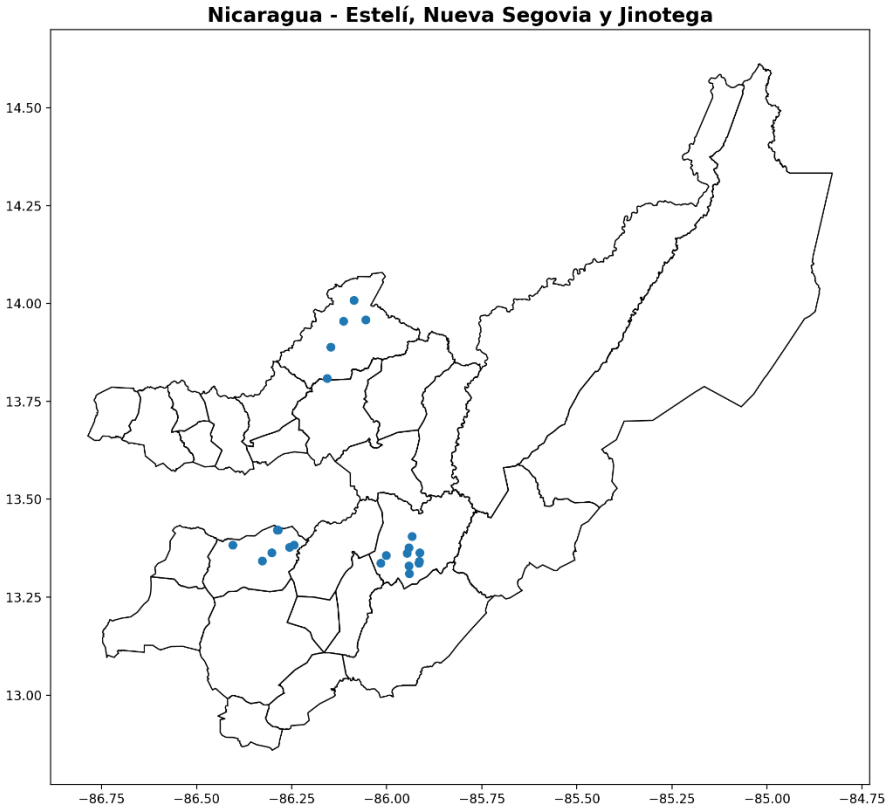

Figura 6. Ubicación de los sensores en los departamentos de Estelí, Nueva Segovia y Jinotega – Nicaragua.

Paso 13: se genera un gráfico conocido como matriz de correlaciones, para observar cómo se relacionan humedad del suelo, temperatura y precipitación:

```
# Grafico matriz de correlaciones
sns.pairplot(data[['Precip', 'Temp', 'Hum_sens']])
plt.tight_layout()
plt.savefig(ruta + '4_Matriz_corr.png')
plt.show();
del ax
```

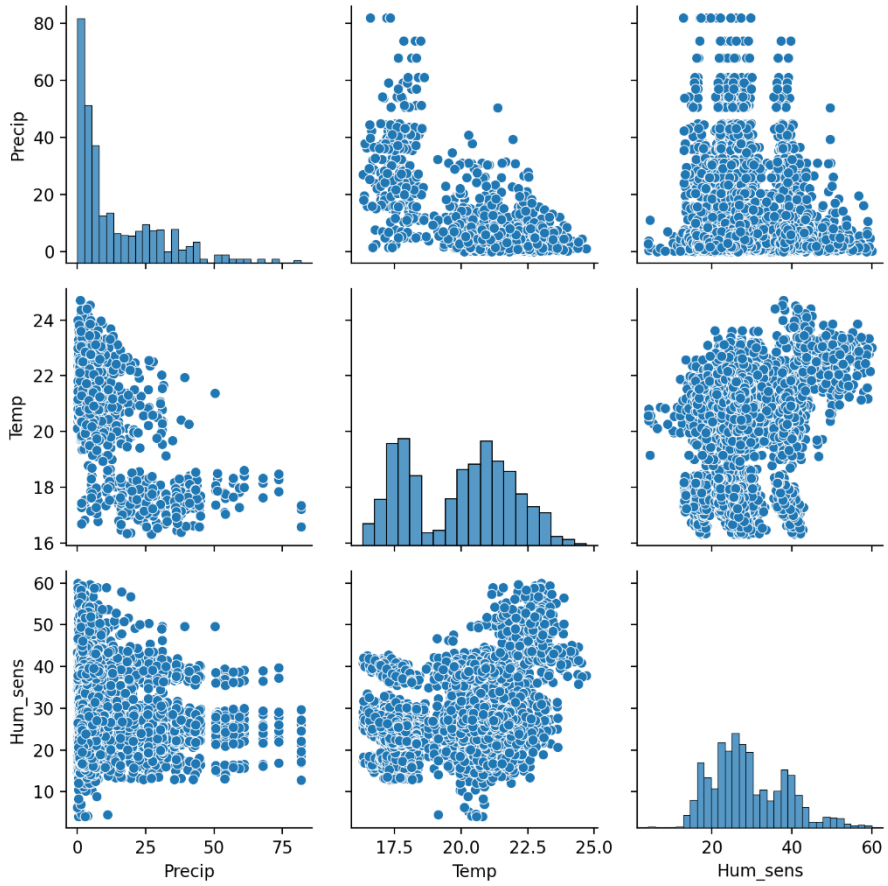

Figura 7. Matriz de correlaciones entre humedad del suelo, precipitación y temperatura.

Paso 14: finalmente, mediante ciclos anidados, se generan y se guardan los gráficos de las series de tiempo para observar el comportamiento de la humedad de suelo promedio registrada por los sensores respecto la precipitación de la zona, agrupándolos por país y por textura de suelo. De igual manera se generaron gráficos de la humedad de suelo agrupándolos de acuerdo con el tipo de cobertura vegetal presente en los cultivos:

```
# Identificacion de paises
paises = pd.unique(data['Pais'])
i=1
# Ciclo FOR para generar los gráficos agrupando por país y textura de suelo
for pais in paises:
    data_sel = data[(data['Pais'] == pais) & (data['Riego'] == 'no_realiza')]
    texturas = pd.unique(data_sel['Text_15cm'])
    for tipo in texturas:
        # Definicion de valores de PMP y CC por textura para trazar las líneas en los gráficos
        if tipo == 'FrancoArcilloso':
            pmp = 13
```

```
    cc = 27
elif tipo == 'FrancoArenoso':
    pmp = 6
    cc = 14
elif tipo == 'Franco':
    pmp = 10
    cc = 22
elif tipo == 'Arcilloso+':
    pmp = 17
    cc = 35
 # Humedad sensor vs precipitacion
sns.lineplot(data=(data_sel[data_sel['Text_15cm']==tipo]), x='Fecha', y='Hum_sens',
        linewidth=.8, errorbar='sd', color='purple', label='Hum_suelo (%)')
plt.gca().set_ylim(bottom=0, top=65)
plt.xlabel('Fecha', fontweight='bold')
plt.ylabel('Humedad del suelo (%)', fontweight='bold')
plt.axhline(y=pmp, color='r', linestyle='--', linewidth=.9)
plt.annotate('PMP', xy=(data_sel['Fecha'].min(),pmp), xytext=(-15,2), textcoords='offset points')
plt.axhline(y=cc, color='b', linestyle='--', linewidth=.9)
plt.annotate('CC', xy=(data_sel['Fecha'].min(),cc), xytext=(-15,2), textcoords='offset points')
plt.axhline(y=((cc+pmp)/2), color='g', linestyle='--', linewidth=.9)
sns.lineplot(data=(data_sel[data_sel['Text_15cm']==tipo]), x='Fecha', y='Precip',
        linewidth=.8, errorbar=None, color='b', label='Precipitación (mm)', ax=plt.twinx())
plt.ylabel('Precipitación (mm)', fontweight='bold')
plt.legend();
plt.title(pais + ' - Humedad del suelo y Precipitación - Tipo: ' + tipo, size=18, fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '5_Hum_vs_Prec_' + str(i) + '_' + pais + '_' + tipo + '.png')
plt.show();
# Humedad sensor vs coberturas
sns.lineplot(data=(data_sel[data_sel['Text_15cm']==tipo]), x='Fecha', y='Hum_sens',
        linewidth=.8, errorbar='sd', hue='Cobertura')
plt.gca().set_ylim(bottom=0, top=65)
plt.xlabel('Fecha', fontweight='bold')
plt.ylabel('Humedad del suelo (%)', fontweight='bold')
plt.axhline(y=pmp, color='r', linestyle='--', linewidth=.9)
plt.annotate('PMP', xy=(data_sel['Fecha'].min(),pmp), xytext=(-15,2), textcoords='offset points')
plt.axhline(y=cc, color='b', linestyle='--', linewidth=.9)
plt.annotate('CC', xy=(data_sel['Fecha'].min(),cc), xytext=(-15,2), textcoords='offset points')
plt.axhline(y=((cc+pmp)/2), color='g', linestyle='--', linewidth=.9)
plt.legend();
plt.title(pais + ' - Humedad del suelo y Coberturas - Tipo: ' + tipo, size=18, fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '6_Cobertura_' + str(i) + '_' + pais + '_' + tipo + '.png')
i = i + 1
plt.show();
```

Para estos gráficos se trazaron las líneas de capacidad de campo “CC” (azul) y punto de marchitez permanente “PMP” (rojo) para cada tipo de textura de suelo, así como la línea media (verde)

entre los dos valores. Adicionalmente se representa la desviación estándar entre las lecturas de los sensores de humedad mediante la franja que se encuentra alrededor de la línea que representa esta variable (franja púrpura), si no se visualiza esta franja significa que solo se cuenta con las lecturas de un solo sensor para ese periodo de tiempo o textura específica.

Figuras de humedad de suelo versus precipitación

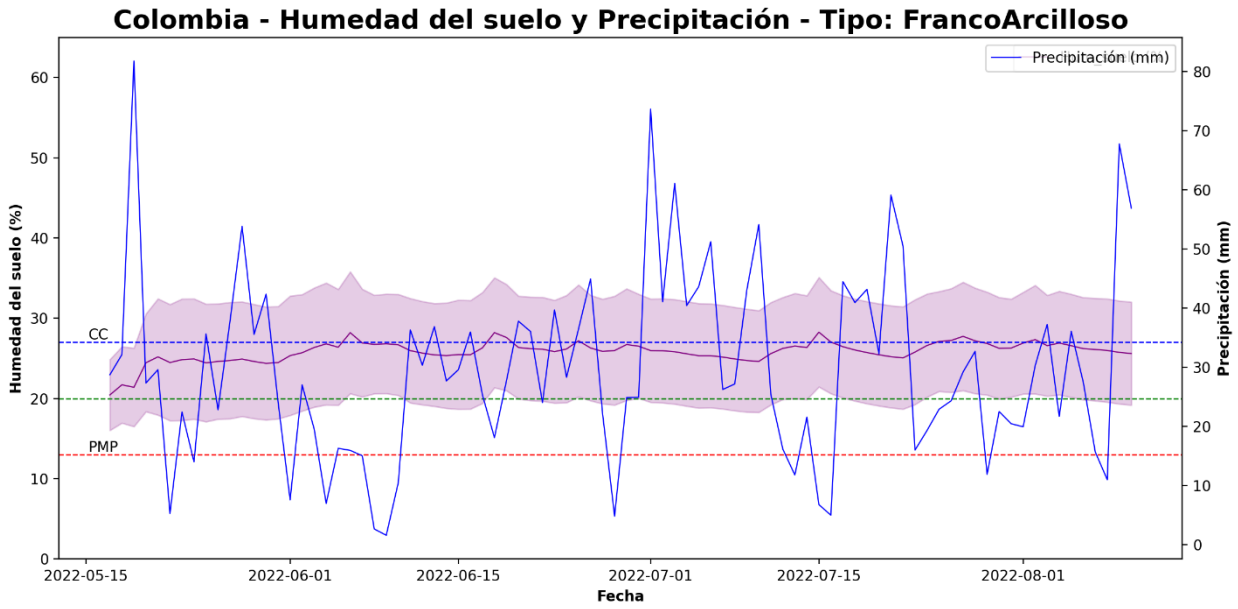

Figura 8. Relación entre humedad del suelo y precipitación – Colombia (Franco Arcilloso).

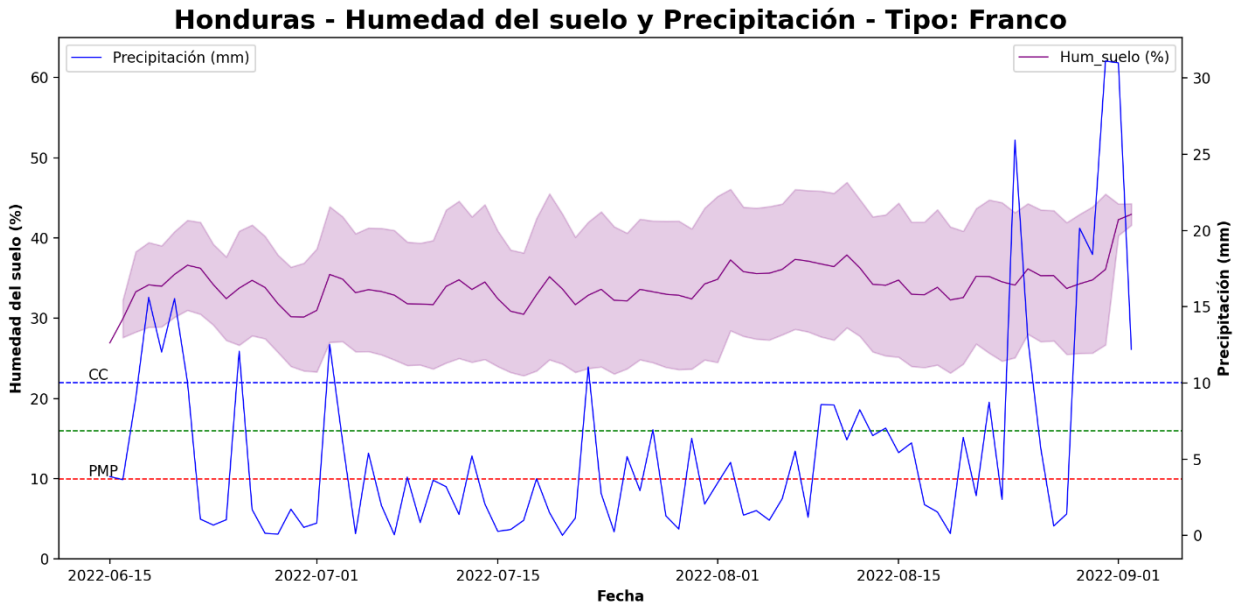

Figura 9. Relación entre humedad del suelo y precipitación – Honduras (Franco).

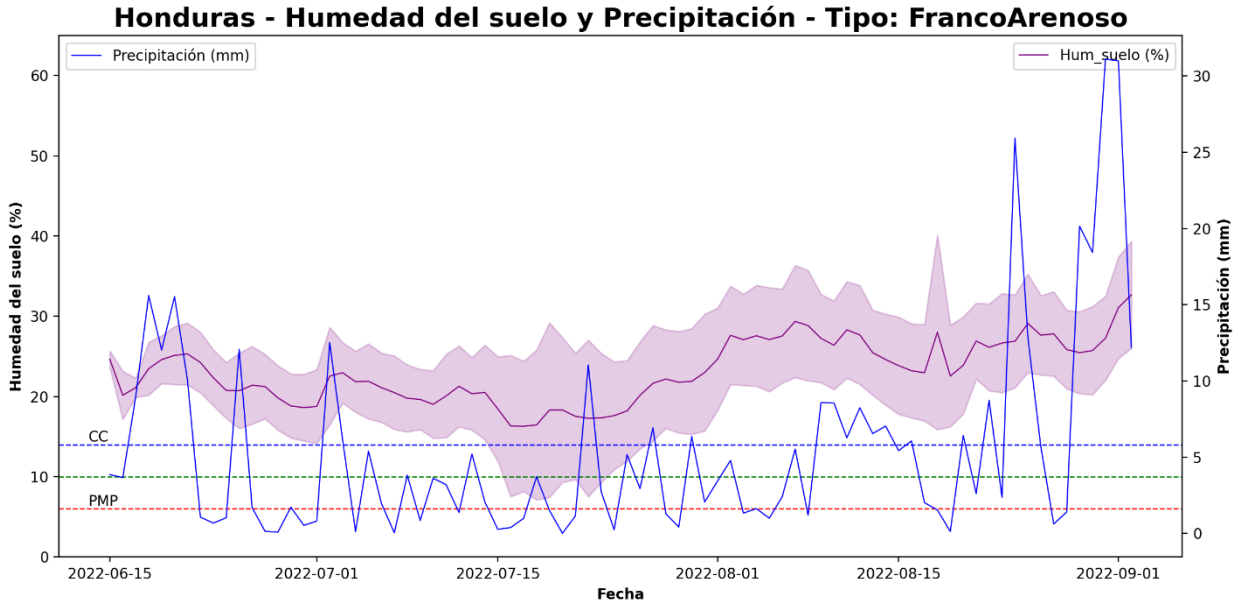

Figura 10. Relación entre humedad del suelo y precipitación – Honduras (Franco Arenoso).

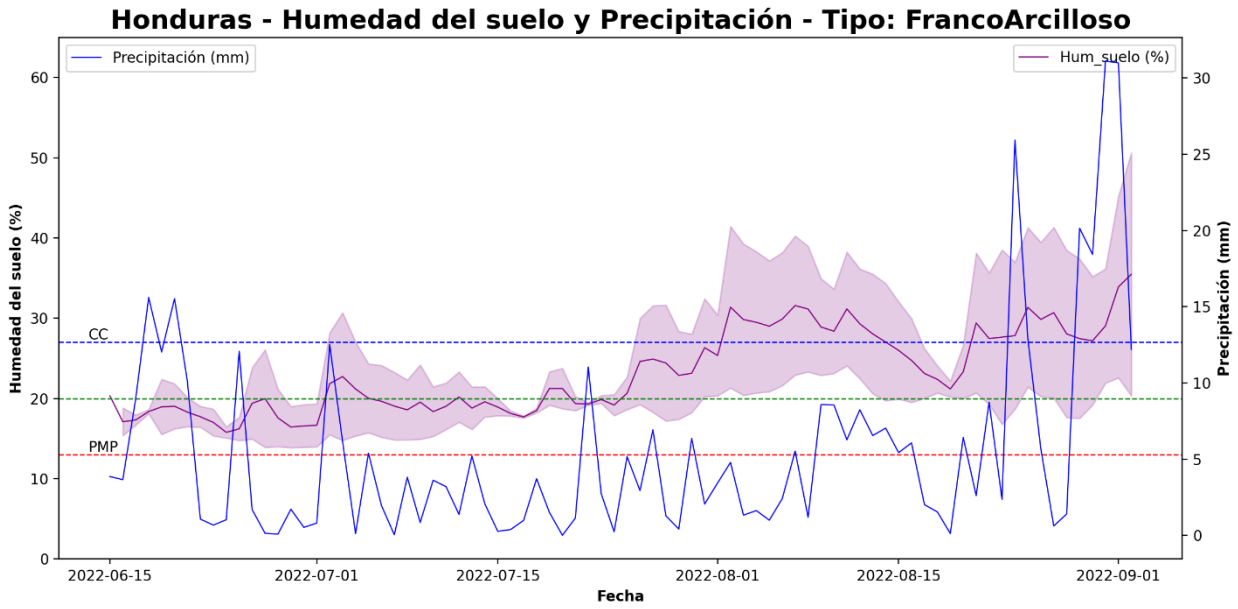

Figura 11. Relación entre humedad del suelo y precipitación – Honduras (Franco Arcilloso).

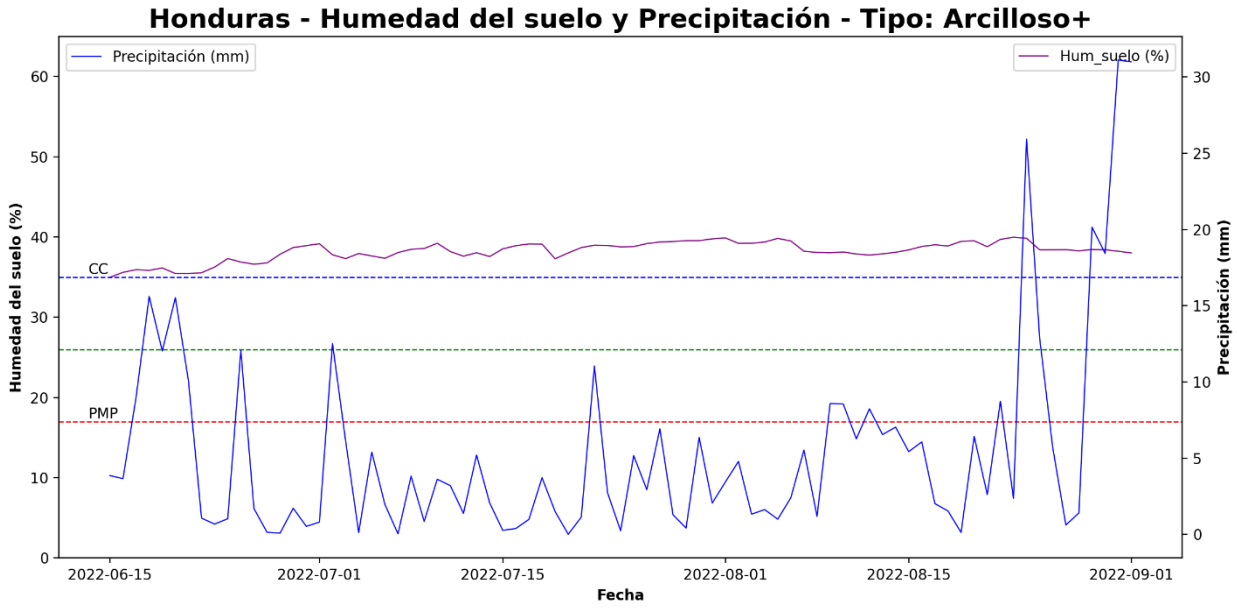

Figura 12. Relación entre humedad del suelo y precipitación – Honduras (Arcilloso+).

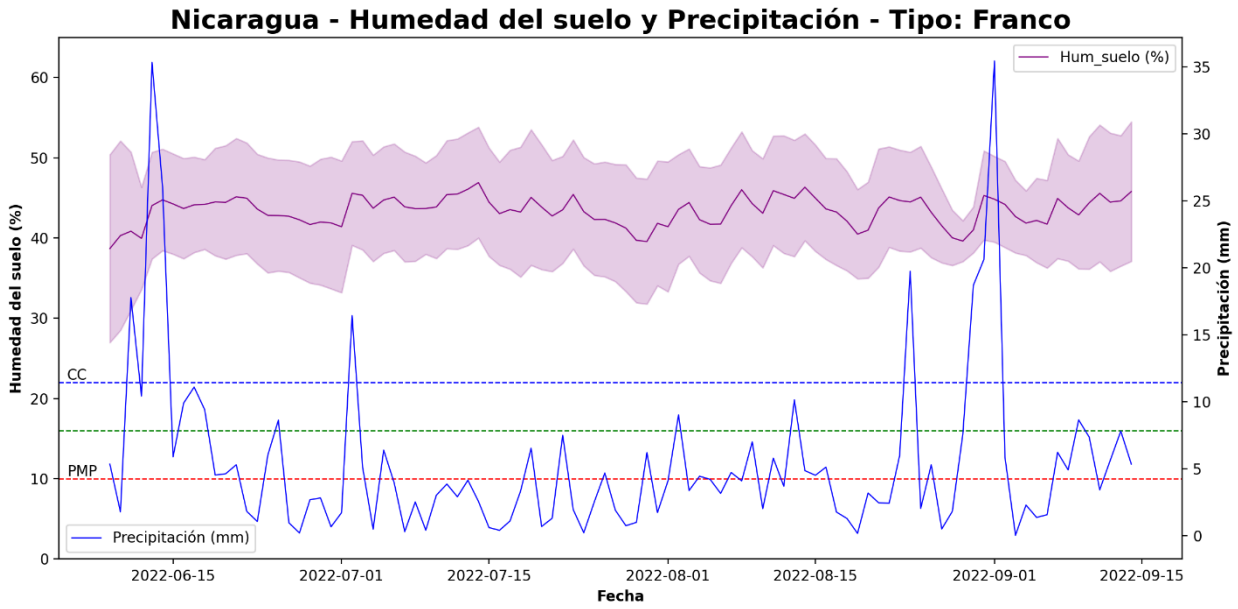

Figura 13. Relación entre humedad del suelo y precipitación – Nicaragua (Franco).

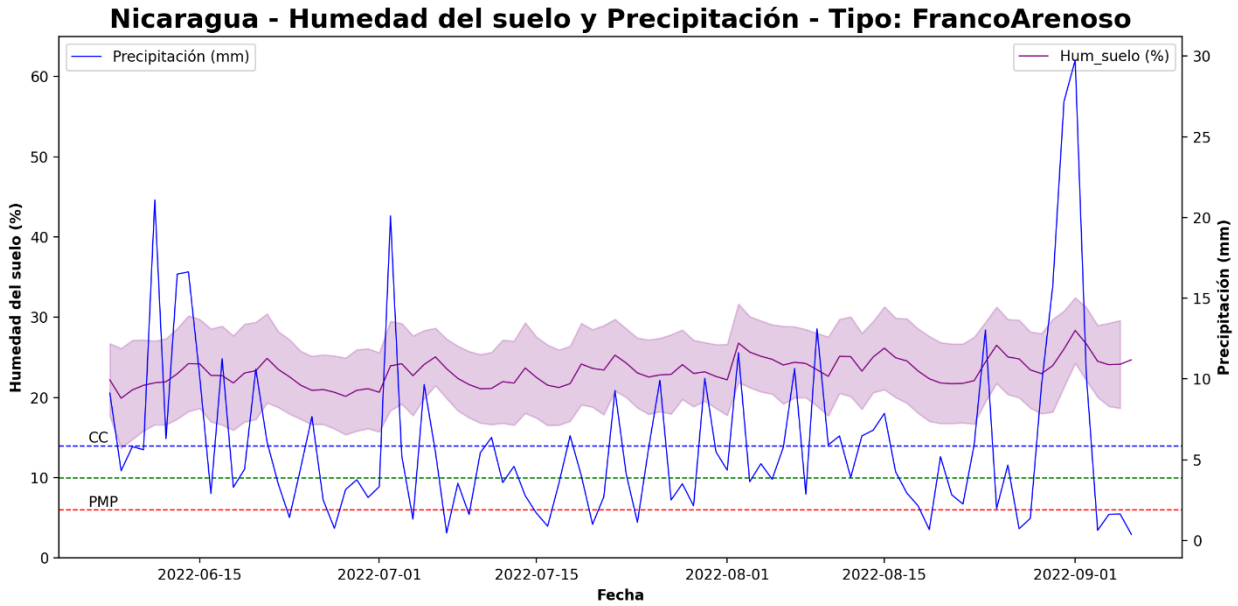

Figura 14. Relación entre humedad del suelo y precipitación – Nicaragua (Franco Arenoso).

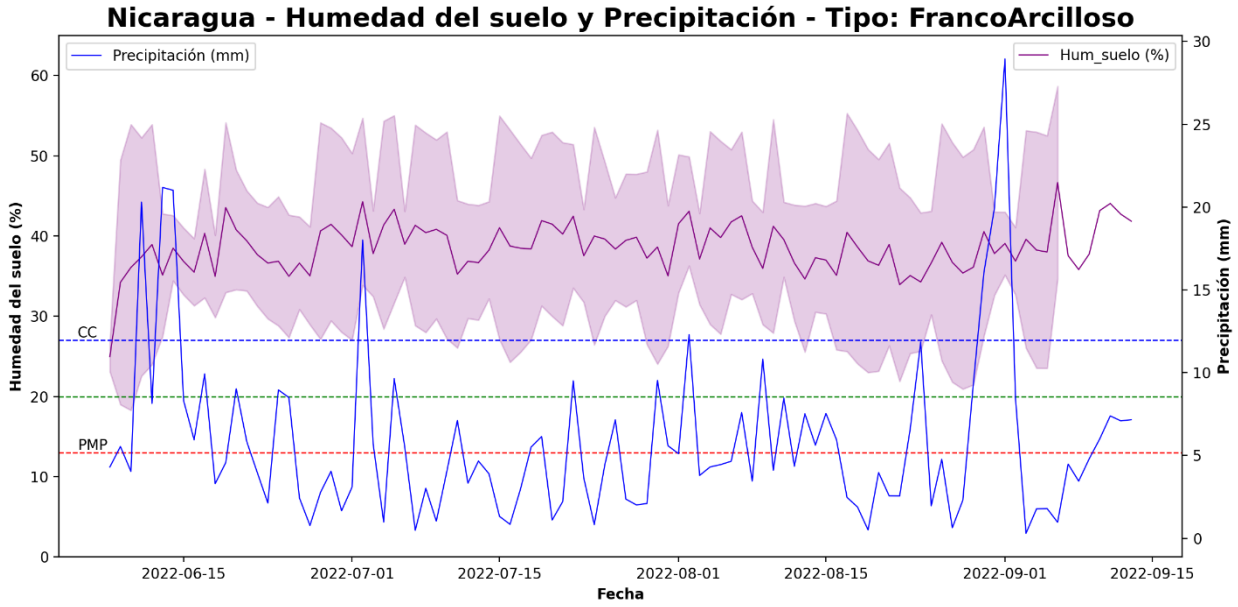

Figura 15. Relación entre humedad del suelo y precipitación – Nicaragua (Franco Arcilloso).

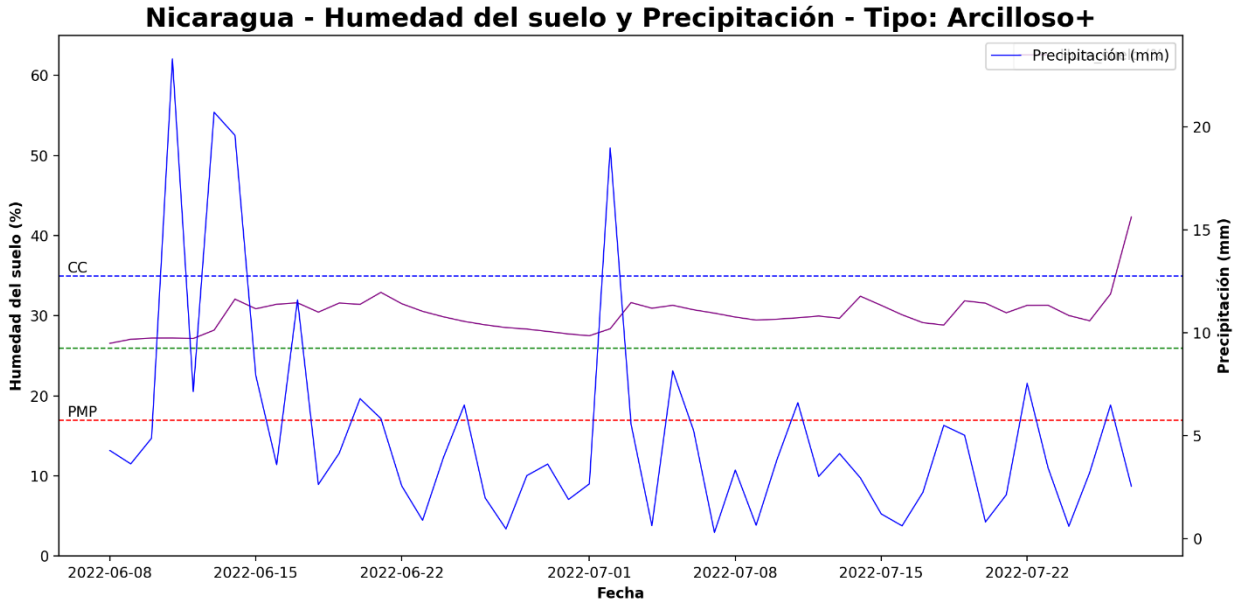

Figura 16. Relación entre humedad del suelo y precipitación – Nicaragua (Arcilloso+).

Figuras de humedad de suelo agrupado por tipo de cobertura

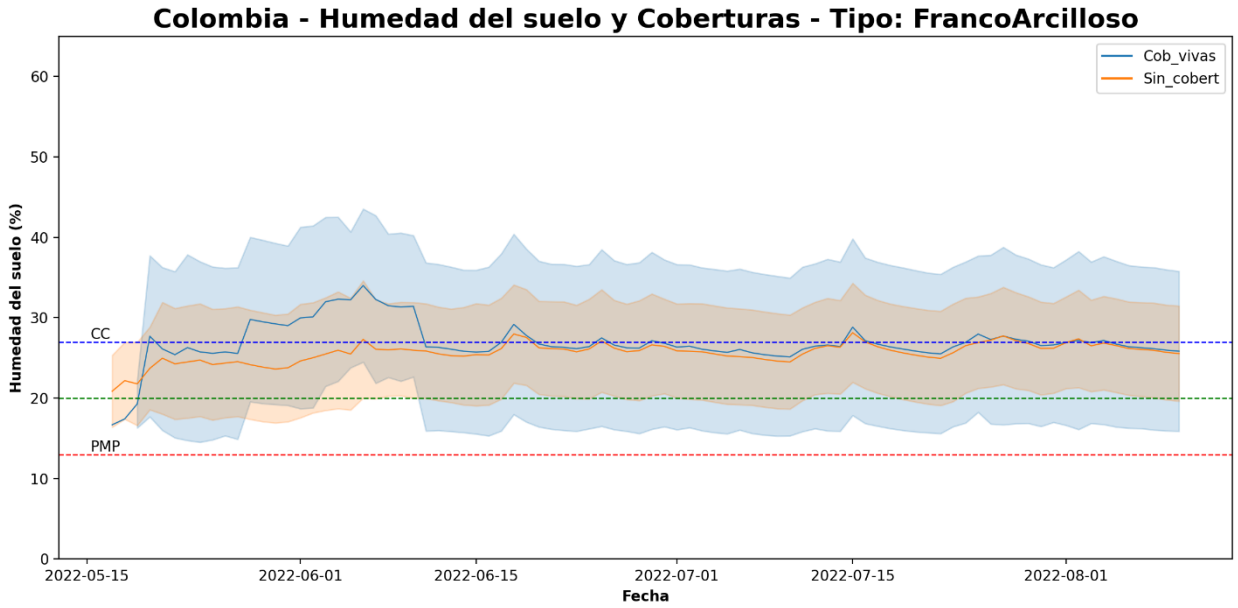

Figura 17. Relación entre humedad del suelo y el tipo de cobertura – Colombia (Franco Arcilloso).

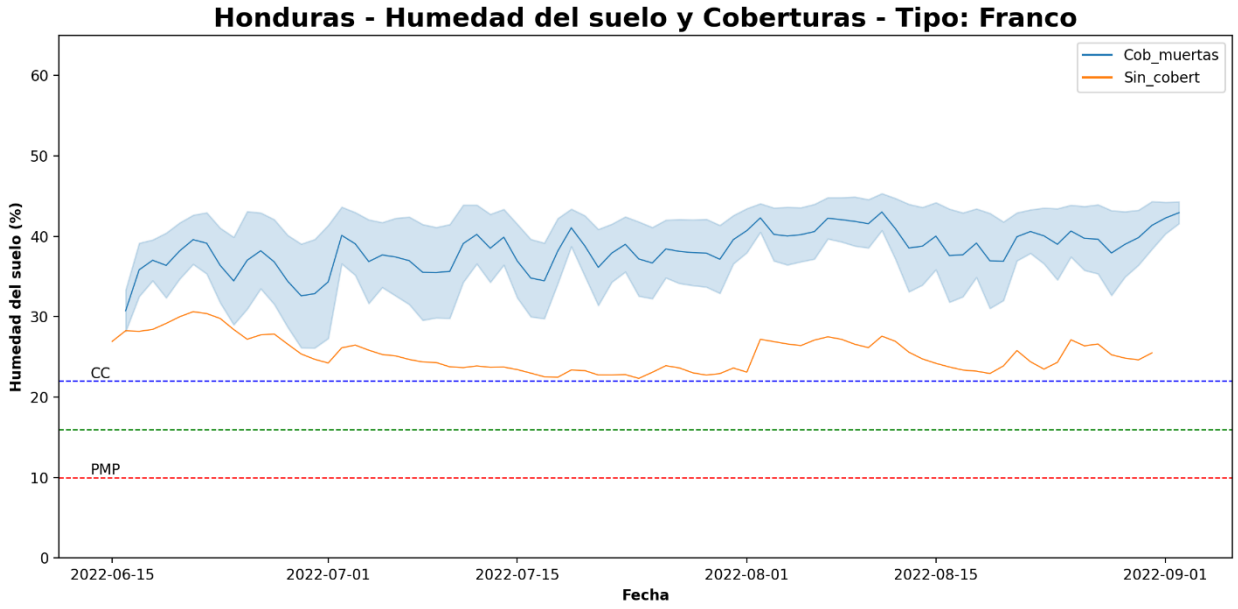

Figura 18. Relación entre humedad del suelo y el tipo de cobertura – Honduras (Franco).

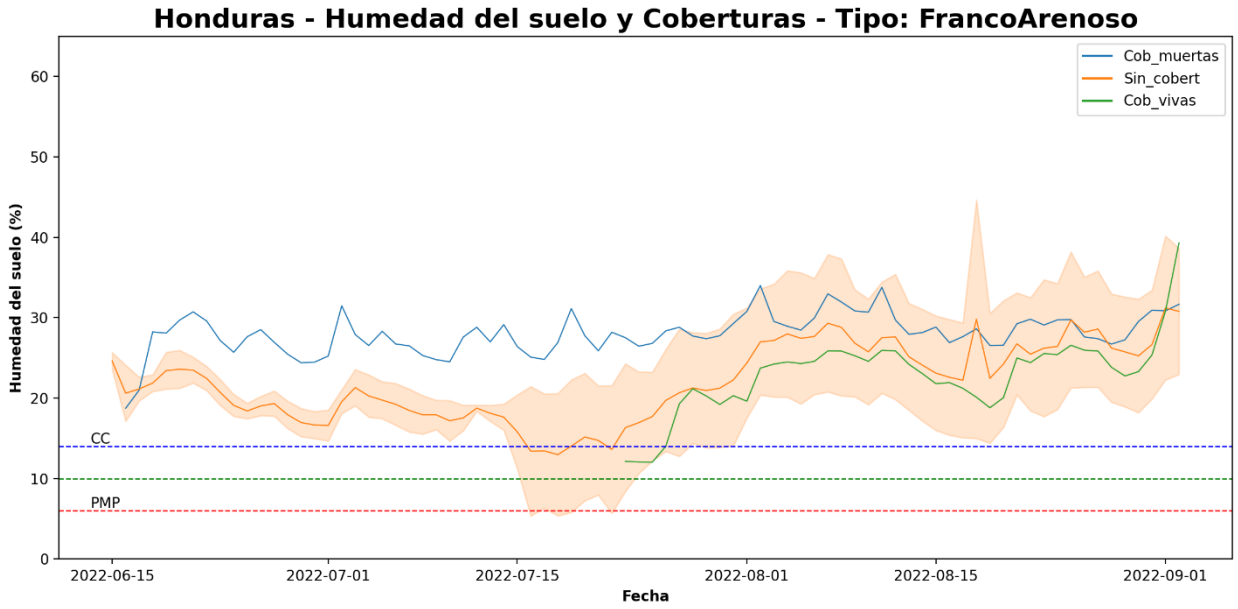

Figura 19. Relación entre humedad del suelo y el tipo de cobertura – Honduras (Franco Arenoso).

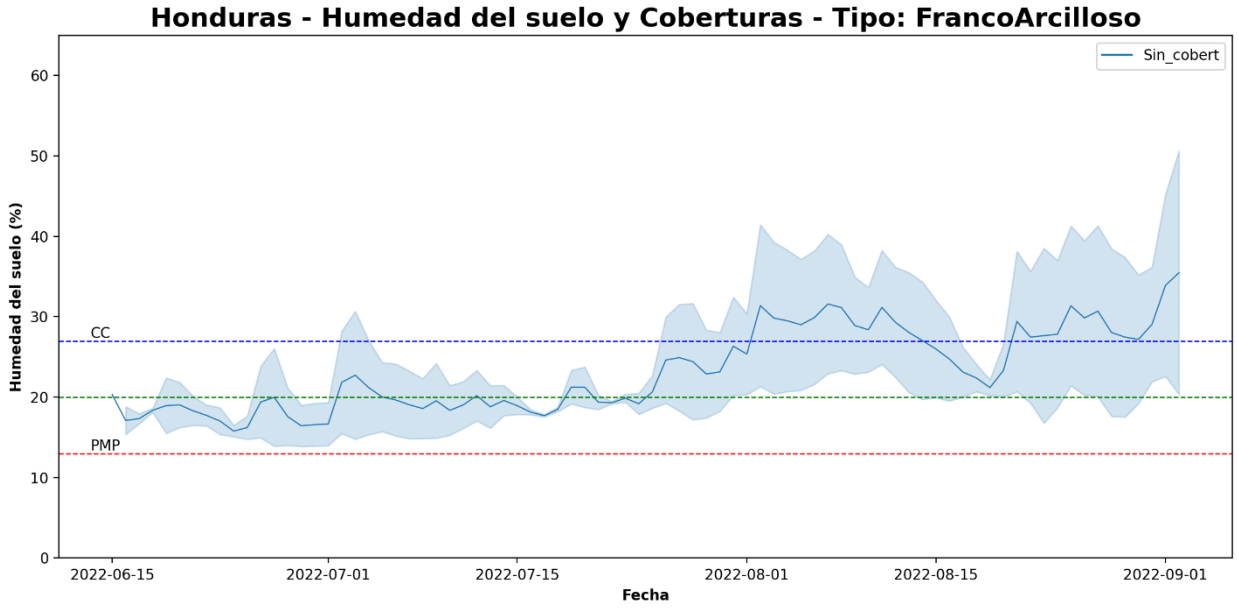

Figura 20. Relación entre humedad del suelo y el tipo de cobertura – Honduras (Franco).

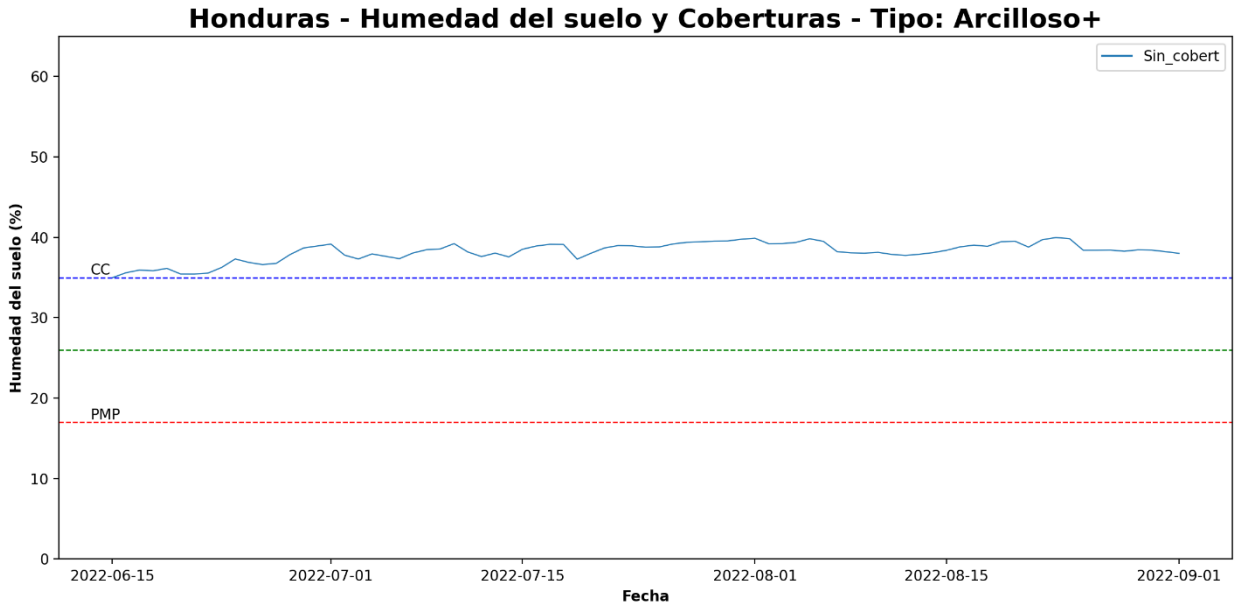

Figura 21. Relación entre humedad del suelo y el tipo de cobertura – Honduras (Arcilloso+).

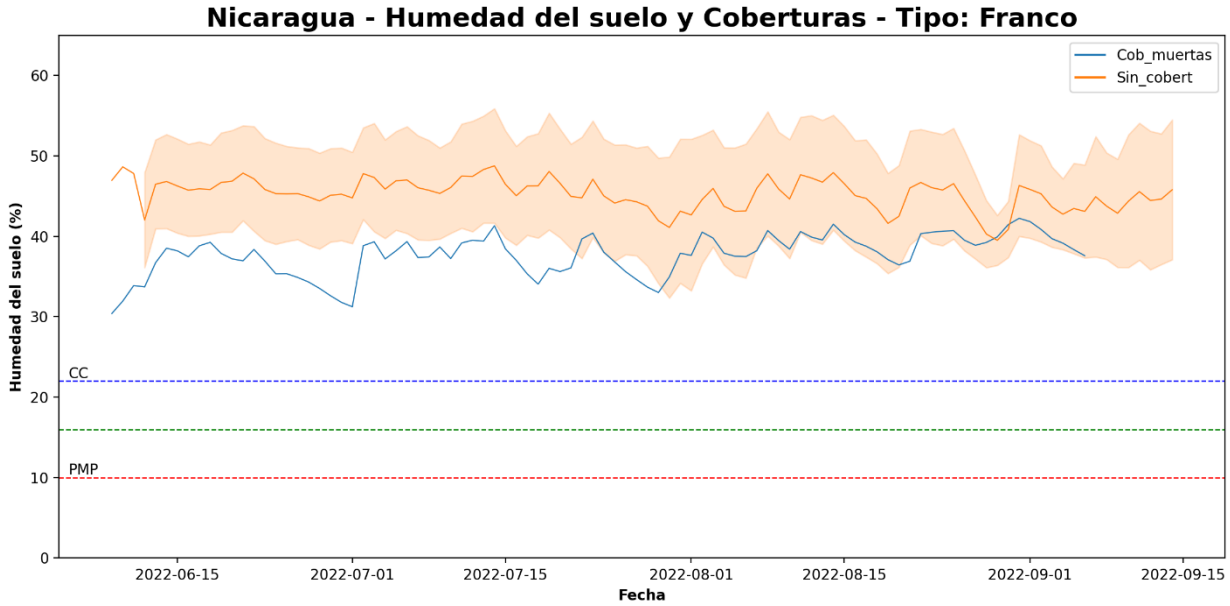

Figura 22. Relación entre humedad del suelo y el tipo de cobertura – Nicaragua (Franco).

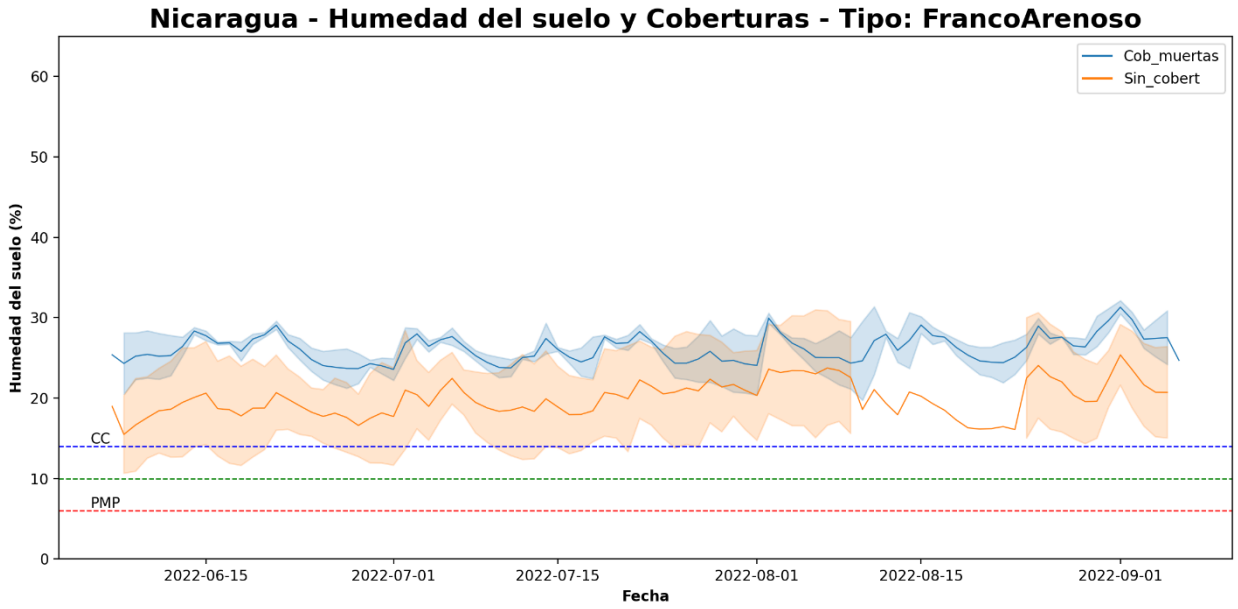

Figura 23. Relación entre humedad del suelo y el tipo de cobertura – Nicaragua (Franco Arenoso).

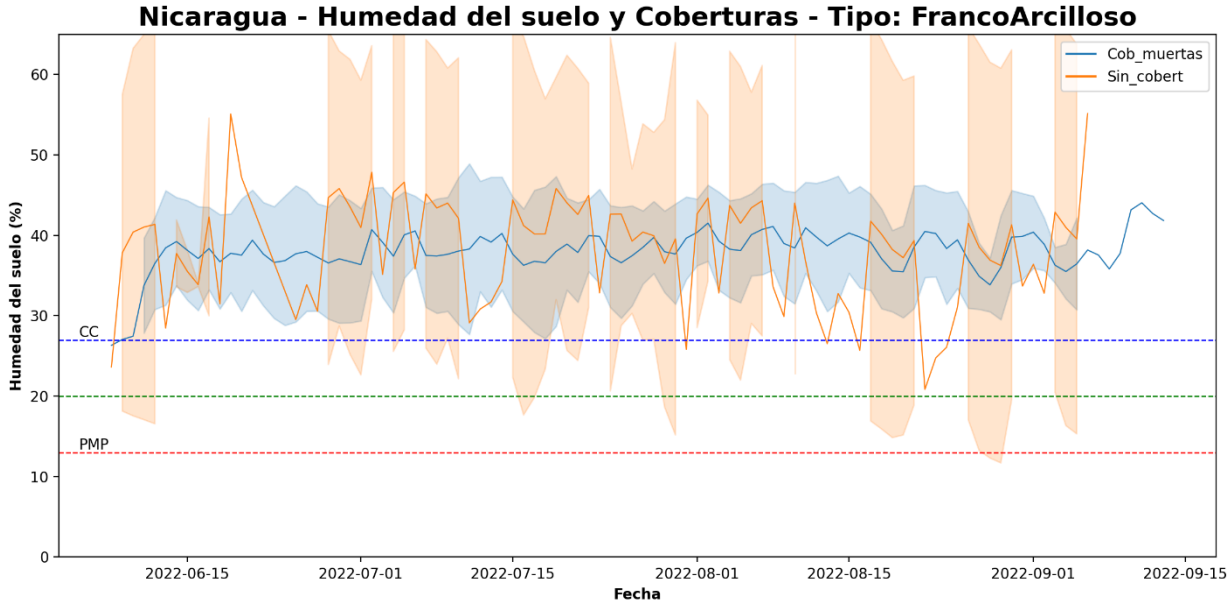

Figura 24. Relación entre humedad del suelo y el tipo de cobertura – Nicaragua (Franco Arcilloso).

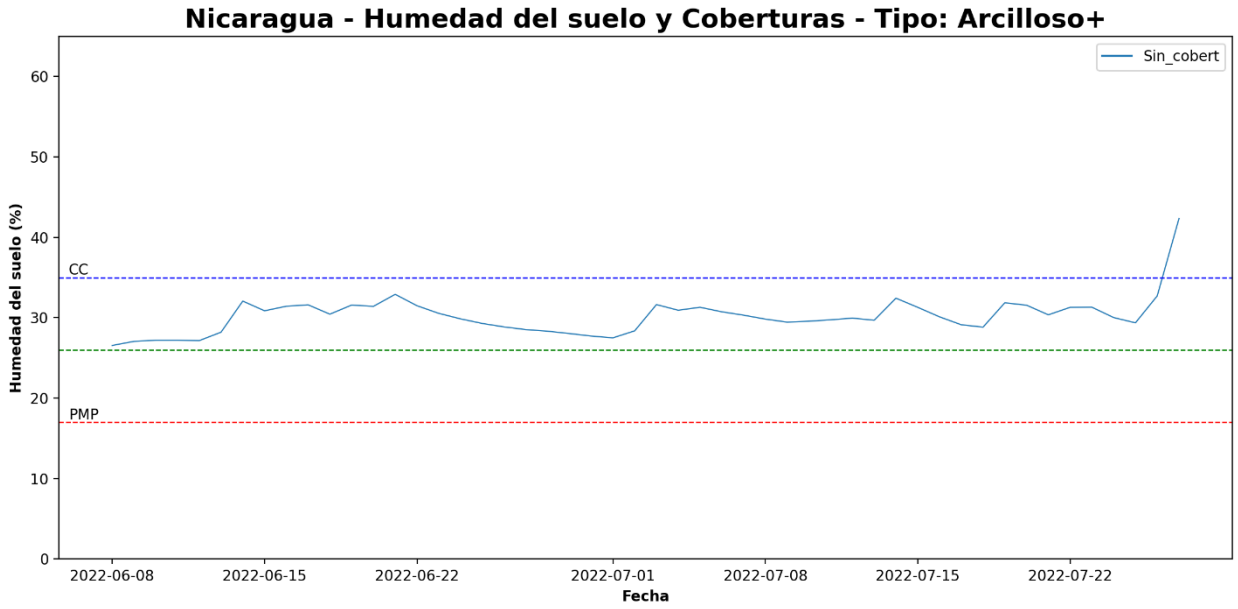

Figura 25. Relación entre humedad del suelo y el tipo de cobertura – Nicaragua (Arcilloso+).

En los gráficos de Humedad del suelo vs Precipitación para suelos Francos, FrancoArenosos y Arcillosos+ en Honduras, así como en los gráficos para suelos Francos, FrancoArenosos y FrancoArcillosos en Nicaragua, aunque se observa una correspondencia en las tendencias de las curvas, los suelos estuvieron en saturación (sobre la línea de capacidad de campo) durante todo el ciclo de cultivo, de acuerdo con lo registrado por los sensores y según los rangos para el tipo de textura. En algunas visitas de seguimiento en campo se pudo evidenciar esta saturación en varias ubicaciones, debido al exceso de lluvias que se presentó durante el semestre de evaluación, sin embargo, la variación en el tiempo de estas curvas, y las pendientes de los terrenos en los que se instalaron algunos de estos sensores hace pensar que existen otros factores que pudieron alterar los datos registrados.

Uno de estos factores podría ser la temperatura, ya que estudios realizados han identificado que las sondas de humedad de suelo capacitivas (como la que utilizan los dispositivos desarrollados en el proyecto) son sensibles a su variación, especialmente las que funcionan con frecuencias inferiores a los 100 MHz (Aranda, Tapia y Millán, 2022). Los estudios también indican que la mineralogía y granulometría del suelo afectan las mediciones, por lo que hacen énfasis en la necesidad de realizar la calibración de este tipo de sensores en cada clase textural de suelo para mejorar la precisión. Estos factores pudieron haber generado un sesgo en la información registrada, por lo que se recomienda, cuando se cuente con conocimiento técnico y tecnologías adicionales, incluir la medición de la temperatura, para realizar la corrección del dato de humedad, y realizar calibración específica para el tipo de suelo en el que se instalará el sensor.

Análisis de datos mediante técnicas de aprendizaje automático

Para encontrar correlaciones entre los valores de humedad registrados por los dispositivos y los datos de precipitación y temperatura, obtenidos para las zonas de estudio mediante la plataforma Copernicus, se construyó un modelo de bosques aleatorios, dada la capacidad y buen desempeño general de esta herramienta (Breiman, 2001), asignando las variables de precipitación y temperatura como entradas del modelo y el valor de humedad de suelo como variable de salida.

Se utilizaron solo estas variables ya que son las únicas de tipo cuantitativo relacionas con la humedad que tienen información diaria. La rutina de Python utilizada se describe a continuación:

Paso 1: en el primer bloque se cargan las librerías, se define la ruta de trabajo y se lee la base de datos en formato ".csv" generada en la rutina de procesamiento anterior:

```
"""
Created on Mon Mar  6 14:59:54 2023
@author: OEstrada
"""
import os
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.inspection import PartialDependenceDisplay
import matplotlib.pyplot as plt
import seaborn as sns
# Definir directorio de trabajo
ruta = r'D:\OneDrive - CGIAR\CIAT\[2022] FONTAGRO'
os.chdir(ruta)
# Leer base de datos
data = pd.read_csv('data_sens.csv')
```

Paso 2: a continuación, se aplica un filtro, conservando solamente los sensores instalados en los lotes que no tuvieron riego, para observar únicamente el efecto de la precipitación sobre los datos de humedad, y se realiza una exploración de número de sensores y días que colectaron datos:

```
# Filtrar por sensores sin riego
data = data[data['Riego'] == 'no_realiza']
# Explorar cuantos dias tienen los sensores y cuantos son sin riego
data['Cod_sens'].value_counts()
len(pd.unique(data['Cod_sens']))
```

Paso 3: se seleccionan solo las variables numéricas de interés y se elimina cualquier registro que tenga datos faltantes, ya que los modelos de aprendizaje automático no aceptan variables categóricas ni información incompleta. Adicionalmente se separan las variables como entradas y

salidas del modelo y se seleccionan los registros para el conjunto de datos de entrenamiento, así como para el conjunto de prueba:

```
# Seleccion de variables numericas
sel = data[['Precip', 'Temp', 'Hum_sens']]
# Eliminar posibles registros con datos faltantes
sel = sel.dropna()
# Seleccionar las variables de entrada y salida
x = sel.drop('Hum_sens', axis=1)
y = sel['Hum_sens']
# Separar datos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state=1)
```

Paso 4: en el bloque siguiente se crea el modelo de bosque aleatorio y se carga con los datos de entrenamiento, posteriormente se evalúa el desempeño del modelo a través de la métrica R^2 (coeficiente de determinación) sobre los datos de entrenamiento y sobre los datos de prueba, para verificar su capacidad de generalizar con datos nuevos. Se eligió esta métrica debido a que el enfoque del análisis es comprender la relación entre las variables y explicar la mayor proporción de varianza de la variable dependiente, contenido de humedad de suelo.

```
# Modelo RandomForest
modelo = RandomForestRegressor(random_state=1, n_jobs=-1)
modelo.fit(X_train, y_train)
# Evaluacion de desempeño del modelo
print('R2 para datos de entrenamiento: ' + str(modelo.score(X_train, y_train)*100))
print('R2 para datos de prueba: ' + str(modelo.score(X_test, y_test)*100))
```

El desempeño obtenido por el modelo fue el siguiente:

R^2 para datos de entrenamiento: 62.6 %
R^2 para datos de prueba: 10.7 %

Estos valores indican que el modelo no tiene un buen desempeño, ya que, aunque las variables de entrada en el grupo de datos de entrenamiento explican hasta un 62.6 % de la varianza observada en la variable de salida, humedad del suelo en este caso, al ingresar datos nuevos que el modelo no ha "visto" solo se obtuvo un 10.7 % de desempeño. Esto sucede principalmente por la poca variabilidad observada en los datos, y la ausencia de patrones de correspondencia marcados entre las variables analizadas. Por ejemplo, el suelo se mantuvo saturado la mayor parte del tiempo que los dispositivos estuvieron en campo, reduciendo significativamente la variación de la humedad en el suelo. Se debe recordar además que las variables de suelo (en el caso de Colombia) y las variables climáticas son datos provenientes de plataformas abiertas, es decir, son aproximaciones generadas a partir de modelos estadísticos e información real en campo. Se recomienda para futuros estudios aumentar el número de sensores/observaciones en el tiempo en el mismo sitio, así como usar datos que correspondan a información primaria

obtenida en campo debidamente procesada y corregida.

Paso 5: posteriormente, en la rutina se extraen del modelo los datos de relevancia de las variables de entrada y se grafican, para entender el aporte que hace cada variable de entrada sobre la variable de salida, es decir, sobre la humedad registrada por el sensor en este caso:

```
# Grafico de relevancia de variables para RandomForest
data_model = pd.concat([pd.Series(modelo.feature_names_in_),
pd.Series(modelo.feature_importances_ )], axis=1)
data_model.columns = ['Feature', 'Values']
data_model = data_model.sort_values('Values', ascending=False)
data_model = data_model.iloc[0:len(data_model['Feature'])]
sns.barplot(data=data_model, x='Feature', y='Values', color='#1f77b4')
plt.title('Relevancia de Variables', size=16, fontweight='bold')
plt.xticks(rotation=90)
plt.xlabel('Variable', fontweight='bold')
plt.ylabel('Relevancia media', fontweight='bold')
plt.tight_layout()
plt.savefig(ruta + '\Relevancias.png')
```

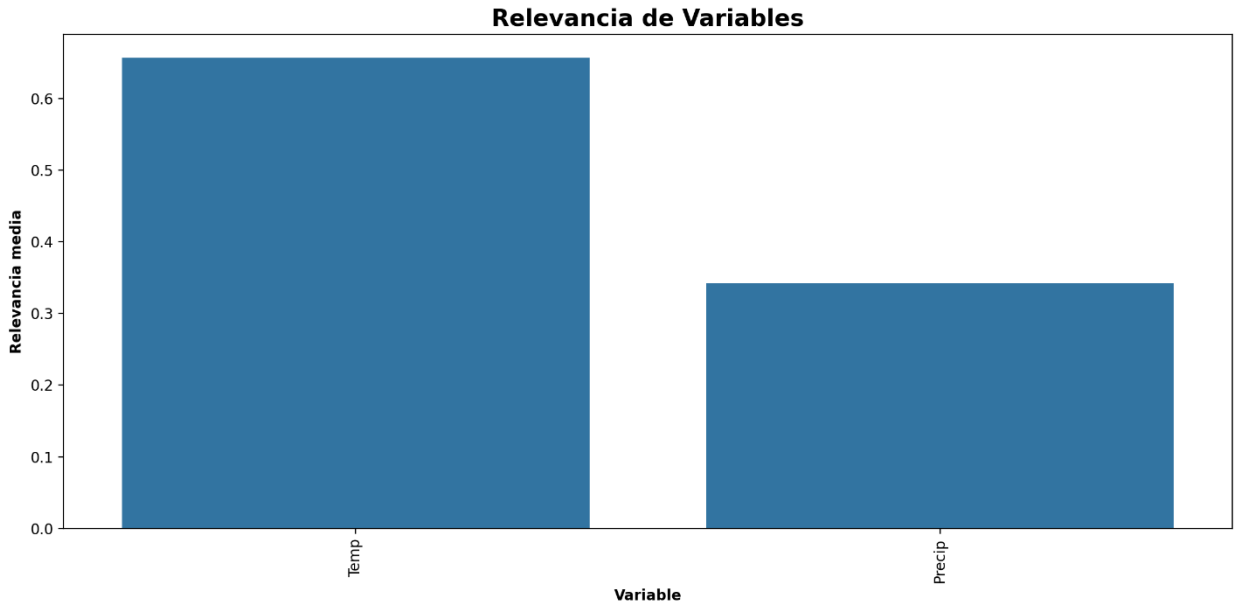

Figura 26. Relevancia de las variables de entrada del modelo de bosque aleatorio.

Relevancia para la variable Temperatura: 65.7 %
Relevancia para la variable Precipitación: 34.3 %

Se observa en el gráfico de relevancias de cada variable, un aporte de la temperatura en el modelo cercano al doble del aporte hecho por la precipitación, lo cual parece confirmar que esta variable

pudo haber sesgado la información de la humedad de suelo. Sin embargo, al tener un modelo con un desempeño general tan bajo, no se puede llegar a una conclusión definitiva sobre el aporte de las variables.

Paso 6: como parte final de la rutina, se extrae del modelo el comportamiento individual o "dependencia parcial" de cada variable, asumiendo una independencia de ésta respecto a las demás variables de entrada, y se grafica esta información mediante un ciclo FOR:

```
# Graficos de dependencias parciales
data_model = pd.concat([pd.Series(modelo.feature_names_in_),
                        pd.Series(modelo.feature_importances_ )], axis=1)
data_model.columns = ['Feature', 'Values']
data_model = data_model.sort_values('Values', ascending=False)
data_model = data_model.iloc[0:len(data_model['Feature'])]
for i in data_model['Feature']:
    PartialDependenceDisplay.from_estimator(modelo, X_train, [i])
    plt.title(i + ' - Dependencia parcial', size=16, fontweight='bold')
    plt.xlabel(i, fontweight='bold')
    plt.ylabel('Dependencia parcial', fontweight='bold')
    plt.tight_layout()
    plt.savefig(ruta + '\Depend_' + i + '.png' )
```

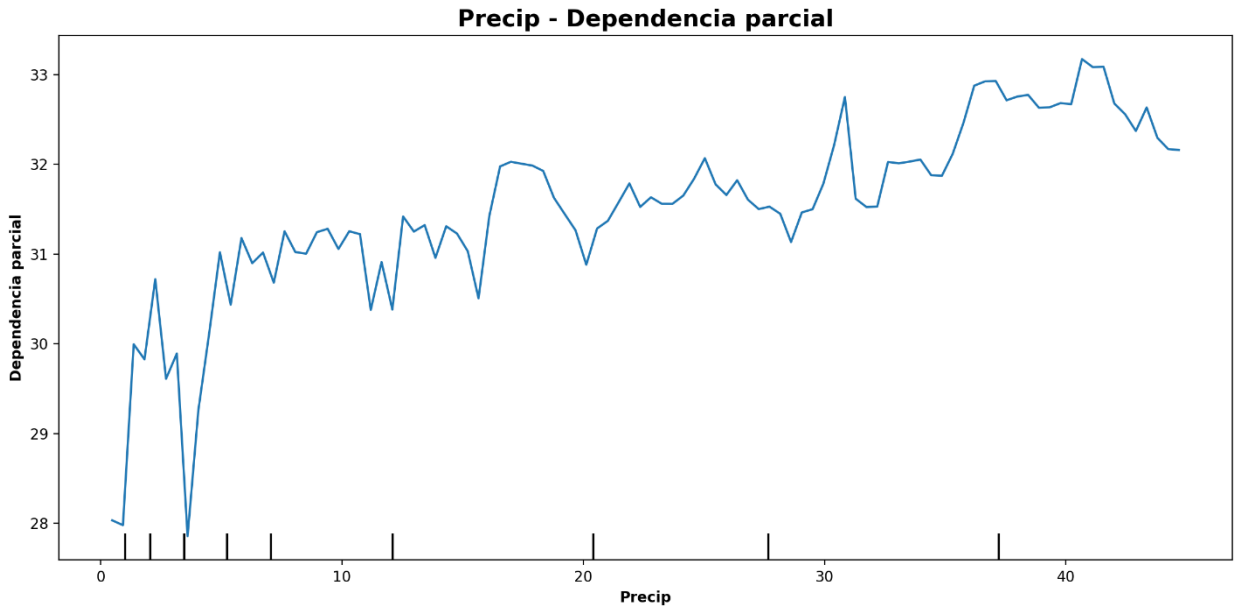

Figura 27. Dependencia parcial de la variable Precipitación respecto a la Humedad de suelo.

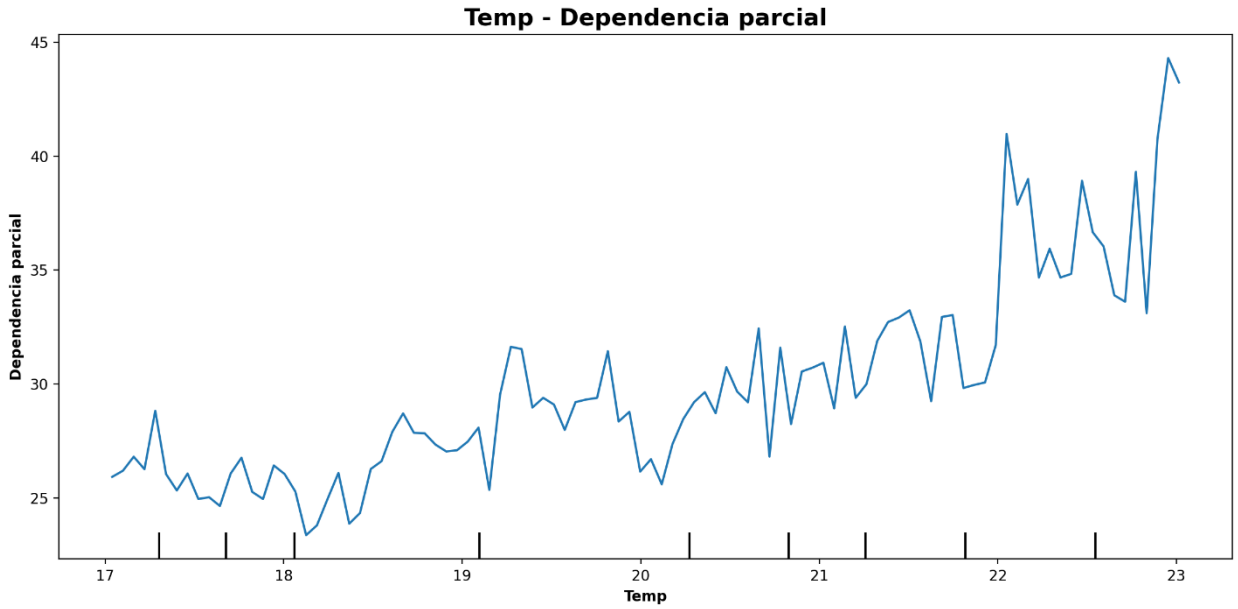

Figura 28. Dependencia parcial de la variable Temperatura respecto a la Humedad de suelo.

En el caso de la dependencia parcial para la precipitación, se observa una tendencia de incremento a medida que se incrementa la humedad del suelo, lo cual coincide con el comportamiento esperado. Sin embargo, en la figura de dependencia parcial de la temperatura también se observa una tendencia similar, lo cual no se debería presentar, pues en este caso se refuerza la idea de que esta variable influye sobre los registros de humedad. Por otro lado, cabe resaltar nuevamente que los resultados de este análisis de aprendizaje automático no tienen suficiente peso para considerarse determinantes, debido al bajo desempeño del modelo y a que los datos no guardan mayor correspondencia entre sí.

Referencias Bibliográficas

Aranda, D., Tapia, A. & Millán, P. (2022). Calibración y caracterización de sensores capacitivos de bajo coste para la monitorización de humedad de suelo. XLIII Jornadas de Automática. [Archivo PDF]. Universidad Loyola.

Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

Instituciones participantes

Secretaría Técnica Administrativa

FONTAGRO

Con el apoyo de:

BID
Banco Interamericano de Desarrollo

IICA

www.fontagro.org
Correo electrónico: fontagro@fontagro.org