Predicting Ground Behavior from Backscatter Trends

Environment preparation and librarys import

```
!jupyter nbconvert --to html --no-input --no-prompt ().ipynb
```

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

```
pip install earthengine-api
```

```
Requirement already satisfied: earthengine-api in /usr/local/lib/python3.12/dist-packages (1.5.24)
Requirement already satisfied: google-cloud-storage in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (2.19.0)
Requirement already satisfied: google-api-python-client>=1.12.1 in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (
Requirement already satisfied: google-auth>=1.4.1 in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (2.38.0)
Requirement already satisfied: google-auth-httplib2>=0.0.3 in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (0.2.0
Requirement already satisfied: httplib2<1dev,>=0.9.2 in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (0.31.0)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from earthengine-api) (2.32.4)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.2.*,!=2.3.0,<3.0.0,>=1.31.5 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: uritemplate<5,>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from google-api-python-client>=1.1
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.4.1->eartheng
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.4.1->earthengi
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.12/dist-packages (from google-auth>=1.4.1->earthengine-api)
Requirement already satisfied: pyparsing<4,>=3.0.4 in /usr/local/lib/python3.12/dist-packages (from httplib2<1dev,>=0.9.2->eartheng
Requirement already satisfied: google-cloud-core<3.0dev,>=2.3.0 in /usr/local/lib/python3.12/dist-packages (from google-cloud-stora
Requirement already satisfied: google-resumable-media>=2.7.2 in /usr/local/lib/python3.12/dist-packages (from google-cloud-storage-
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /usr/local/lib/python3.12/dist-packages (from google-cloud-storage->ea
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->earthengine-api)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->earthengine-api) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->earthengine-api) (2.5.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->earthengine-api) (2025
Requirement already satisfied: googleapis-common-protos<2.0.0,>=1.56.2 in /usr/local/lib/python3.12/dist-packages (from google-api-
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<7.0.0,>=3.19.5 in /
Requirement already satisfied: proto-plus<2.0.0,>=1.22.3 in /usr/local/lib/python3.12/dist-packages (from google-api-core!=2.0.*,!=
Requirement already satisfied: pyasn1<0.7.0,>=0.6.1 in /usr/local/lib/python3.12/dist-packages (from pyasn1-modules>=0.2.1->google-
```

```
pip install earthaccess
```

```
  Downloading tenacity-9.1.2-py3-none-any.whl.metadata (1.2 kB)
Collecting tinynetrc>=1.3.1 (from earthaccess)
  Downloading tinynetrc-1.3.1-py2.py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from earthaccess) (4.15.0)
Collecting bounded-pool-executor (from pqdm>=0.1->earthaccess)
  Downloading bounded_pool_executor-0.0.3-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from pqdm>=0.1->earthaccess) (4.67.1)
Requirement already satisfied: python-dateutil<3.0.0,>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from python-cmr>=0.10.0-
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests>=2.26->earthacc
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests>=2.26->earthaccess) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests>=2.26->earthaccess) (
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests>=2.26->earthaccess) (
Collecting aiobotocore<3.0.0,>=2.5.4 (from s3fs>=2025.2->earthaccess)
  Downloading aiobotocore-2.24.2-py3-none-any.whl.metadata (25 kB)
Collecting fsspec>=2025.2 (from earthaccess)
  Downloading fsspec-2025.9.0-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from s3fs>=2025.2->earthacc
Collecting aioitertools<1.0.0,>=0.5.1 (from aiobotocore<3.0.0,>=2.5.4->s3fs>=2025.2->earthaccess)
  Downloading aioitertools-0.12.0-py3-none-any.whl.metadata (3.8 kB)
Collecting botocore<1.40.19,>=1.40.15 (from aiobotocore<3.0.0,>=2.5.4->s3fs>=2025.2->earthaccess)
```

```
Downloading pqdm-0.2.0-py2.py3-none-any.whl (6.8 kB)
Downloading python_cmr-0.13.0-py3-none-any.whl (14 kB)
Downloading s3fs-2025.9.0-py3-none-any.whl (30 kB)
Downloading fsspec-2025.9.0-py3-none-any.whl (199 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 199.3/199.3 kB 7.7 MB/s eta 0:00:00
Downloading tenacity-9.1.2-py3-none-any.whl (28 kB)
Downloading tinynetrc-1.3.1-py2.py3-none-any.whl (3.9 kB)
Downloading aiobotocore-2.24.2-py3-none-any.whl (85 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 85.4/85.4 kB 4.8 MB/s eta 0:00:00
Downloading bounded_pool_executor-0.0.3-py3-none-any.whl (3.4 kB)
Downloading aioitertools-0.12.0-py3-none-any.whl (24 kB)
Downloading botocore-1.40.18-py3-none-any.whl (14.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.0/14.0 MB 98.1 MB/s eta 0:00:00
Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Installing collected packages: tinynetrc, bounded-pool-executor, tenacity, pqdm, multimethod, jmespath, fsspec, aioitertools, pyt
  Attempting uninstall: tenacity
    Found existing installation: tenacity 8.5.0
    Uninstalling tenacity-8.5.0:
      Successfully uninstalled tenacity-8.5.0
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2025.3.0
    Uninstalling fsspec-2025.3.0:
      Successfully uninstalled fsspec-2025.3.0
```

```
!pip install geopandas shapely fiona pyproj
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.12/dist-packages (1.1.1)
Requirement already satisfied: shapely in /usr/local/lib/python3.12/dist-packages (2.1.2)
Collecting fiona
  Downloading fiona-1.10.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (56 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.6/56.6 kB 274.7 kB/s eta 0:00:00
Requirement already satisfied: pyproj in /usr/local/lib/python3.12/dist-packages (3.7.2)
Requirement already satisfied: numpy>=1.24 in /usr/local/lib/python3.12/dist-packages (from geopandas) (2.0.2)
Requirement already satisfied: pyogrio>=0.7.2 in /usr/local/lib/python3.12/dist-packages (from geopandas) (0.11.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.12/dist-packages (from geopandas) (25.0)
Requirement already satisfied: pandas>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from geopandas) (2.2.2)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.12/dist-packages (from fiona) (25.3.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from fiona) (2025.8.3)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.12/dist-packages (from fiona) (8.3.0)
Collecting click-plugins>=1.0 (from fiona)
  Downloading click_plugins-1.1.1.2-py2.py3-none-any.whl.metadata (6.5 kB)
Collecting cligj>=0.5 (from fiona)
  Downloading cligj-0.7.2-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->geopandas) (2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->geopandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas>=2.0.0->geopandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas>=2.0.0->geo
Downloading fiona-1.10.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.2/17.2 MB 23.2 MB/s eta 0:00:00
Downloading click_plugins-1.1.1.2-py2.py3-none-any.whl (11 kB)
Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Installing collected packages: cligj, click-plugins, fiona
Successfully installed click-plugins-1.1.1.2 cligj-0.7.2 fiona-1.10.1
```

```
!pip install contextily
```

```
Collecting contextily
  Downloading contextily-1.6.2-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: geopy in /usr/local/lib/python3.12/dist-packages (from contextily) (2.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from contextily) (3.10.0)
Collecting mercantile (from contextily)
  Downloading mercantile-1.2.1-py3-none-any.whl.metadata (4.8 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packages (from contextily) (11.3.0)
Collecting rasterio (from contextily)
  Downloading rasterio-1.4.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.1 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from contextily) (2.32.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from contextily) (1.5.2)
Requirement already satisfied: xyzservices in /usr/local/lib/python3.12/dist-packages (from contextily) (2025.4.0)
Requirement already satisfied: geographiclib<3,>=1.52 in /usr/local/lib/python3.12/dist-packages (from geopy->contextily) (2.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (1.4.9)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (2.9.0
Requirement already satisfied: click>=3.0 in /usr/local/lib/python3.12/dist-packages (from mercantile->contextily) (8.3.0)
Collecting affine (from rasterio->contextily)
```

```
  Downloading affine-2.4.0-py3-none-any.whl.metadata (4.0 kB)
Requirement already satisfied: attrs in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (25.3.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (2025.8.3)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (0.7.2)
Requirement already satisfied: click-plugins in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (1.1.1.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (3.4
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (2.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib->contexti
Downloading contextily-1.6.2-py3-none-any.whl (17 kB)
Downloading mercantile-1.2.1-py3-none-any.whl (14 kB)
Downloading rasterio-1.4.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (22.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 22.3/22.3 MB 5.6 MB/s eta 0:00:00
Downloading affine-2.4.0-py3-none-any.whl (15 kB)
Installing collected packages: mercantile, affine, rasterio, contextily
Successfully installed affine-2.4.0 contextily-1.6.2 mercantile-1.2.1 rasterio-1.4.3
```

```
pip install contextily
```

```
Collecting contextily
  Downloading contextily-1.6.2-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: geopy in /usr/local/lib/python3.12/dist-packages (from contextily) (2.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (from contextily) (3.10.0)
Collecting mercantile (from contextily)
  Downloading mercantile-1.2.1-py3-none-any.whl.metadata (4.8 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.12/dist-packages (from contextily) (11.3.0)
Requirement already satisfied: rasterio in /usr/local/lib/python3.12/dist-packages (from contextily) (1.4.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from contextily) (2.32.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from contextily) (1.5.2)
Requirement already satisfied: xyzservices in /usr/local/lib/python3.12/dist-packages (from contextily) (2025.4.0)
Requirement already satisfied: geographiclib<3,>=1.52 in /usr/local/lib/python3.12/dist-packages (from geopy->contextily) (2.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (1.4.9)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib->contextily) (2.9.0
Requirement already satisfied: click>=3.0 in /usr/local/lib/python3.12/dist-packages (from mercantile->contextily) (8.3.0)
Requirement already satisfied: affine in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (2.4.0)
Requirement already satisfied: attrs in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (25.3.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (2025.8.3)
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (0.7.2)
Requirement already satisfied: click-plugins in /usr/local/lib/python3.12/dist-packages (from rasterio->contextily) (1.1.1.2)
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (3.4
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->contextily) (2.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib->contexti
Downloading contextily-1.6.2-py3-none-any.whl (17 kB)
Downloading mercantile-1.2.1-py3-none-any.whl (14 kB)
Installing collected packages: mercantile, contextily
Successfully installed contextily-1.6.2 mercantile-1.2.1
```

Conection to Earth Data

```
import ee
ee.Authenticate()
ee.Initialize(project="spaceappsnasa-474214")
```

```



import earthaccess

# Te pedirá login en una ventanita/console output en Colab
auth = earthaccess.login(strategy="interactive")

# Ejemplo: buscar Sentinel-1 en Palín
results = earthaccess.search_data(
    short_name="SENTINEL-1A_SLC",
 bounding_box=(-90.7594, 14.4433, -90.6594, 14.5433),  #Santa María de Jesús, Sacatepéquez, Guatemala
    temporal=("2025-07-01", "2025-10-4")
)

print("Granules encontrados:", len(results))
```

```
Enter your Earthdata Login username: majoooo
Enter your Earthdata password: ··········
Granules encontrados: 14
```

```
import earthaccess

# Te pedirá login en una ventanita/console output en Colab
auth = earthaccess.login(strategy="interactive")

# Ejemplo: buscar Sentinel-1 en Palín
results = earthaccess.search_data(
    short_name="SENTINEL-1A_SLC",
 bounding_box=(-90.7594, 14.4433, -90.6594, 14.5433),  #Santa María de Jesús, Sacatepéquez, Guatemala
    temporal=("2025-09-01", "2025-10-4")
)

print("Granules encontrados:", len(results))
```

```
Granules encontrados: 5
```

```
save_dir = "/content/drive/MyDrive/data"

local_files = earthaccess.download(results, save_dir)

print("Downloaded", len(local_files), "granules into:", save_dir)
```

```
QUEUEING TASKS | : 100%                                              14/14 [00:00<00:00, 396.88it/s]
PROCESSING TASKS | : 100%                                              14/14 [07:56<00:00, 25.09s/it]
COLLECTING RESULTS | : 100%                                              14/14 [00:00<00:00, 479.35it/s]
Downloaded 14 granules into: /content/drive/MyDrive/data
```

Close

```
## What This Is?

High above Guatemala, the Sentinel-1 satellite passes silently over
of Volcán de Agua. From space, it captures echoes from the ground —
reflections that carry stories of movement, moisture, and change.

This notebook brings those echoes to life through two connected expl
The first looks back, tracing how the surface of Santa María de Jesú
changed in recent months. The second looks forward, using those same
forecast how the land might evolve in the near future. Together, the
satellite data into a narrative of a living landscape — one that mov
breathes, and occasionally warns us before we can see it with our ow
```

What This Is?

High above Guatemala, the Sentinel-1 satellite passes silently over the slopes of Volcán de Agua. From space, it captures echoes from the ground — faint reflections that carry stories of movement, moisture, and change.

This notebook brings those echoes to life through two connected explorations. The first looks back, tracing how the surface of Santa María de Jesús has changed in recent months. The second looks forward, using those same signals to forecast how the land might evolve in the near future. Together, they turn raw satellite data into a narrative of a living landscape — one that moves, breathes, and occasionally warns us before we can see it with our own eyes.

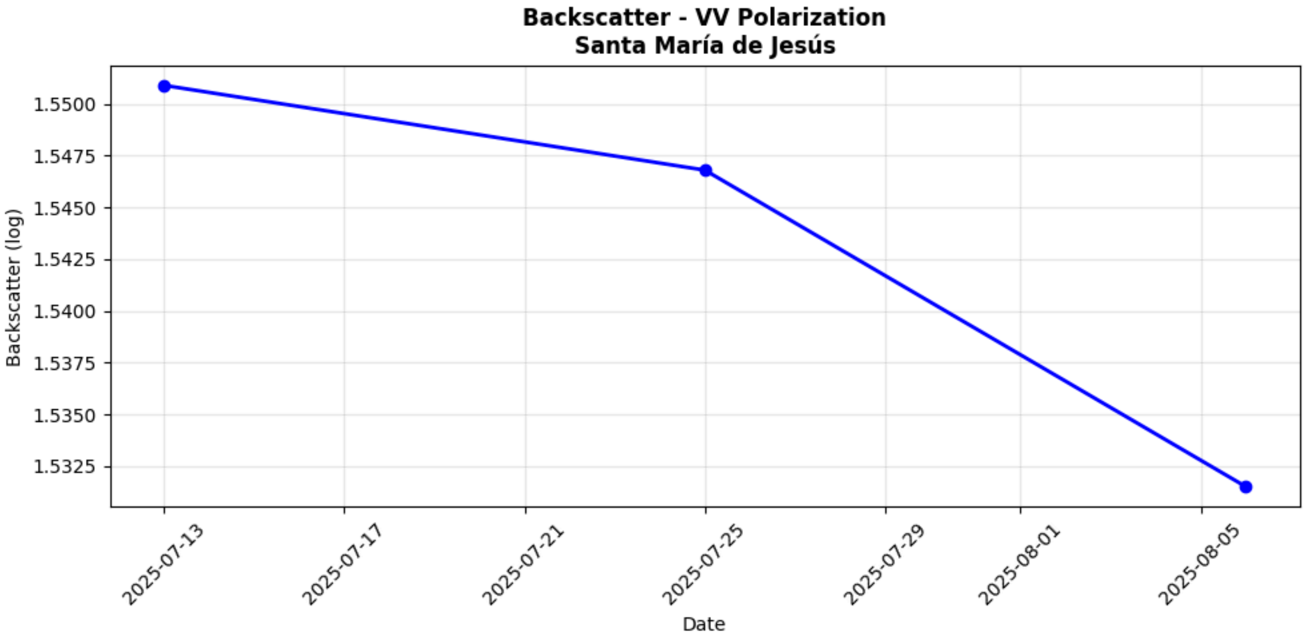
```
import requests
import pandas as pd
from shapely.geometry import Point
import geopandas as gpd

# USGS API query
url = ("https://earthquake.usgs.gov/fdsnws/event/1/query?format=geojson"
       "&starttime=2025-04-01&endtime=2025-10-04"
       "&minlatitude=14.3&maxlatitude=14.6"
       "&minlongitude=-90.8&maxlongitude=-90.6")

# Get JSON directly
resp = requests.get(url)
data = resp.json()

# Parse features
records = []
for f in data["features"]:
    coords = f["geometry"]["coordinates"]  # [lon, lat, depth]
    coords = f["geometry"]["coordinates"]  # [lon, lat, depth]
    props = f["properties"]
    records.append({
        "mag": props.get("mag"),
        "time": props.get("time"),
        "place": props.get("place"),
        "geometry": Point(coords[0], coords[1])
    })

# Convert to GeoDataFrame
gdf = gpd.GeoDataFrame(records, crs="EPSG:4326")

print("Earthquakes loaded:", len(gdf))
print(gdf.head())
```

```
Earthquakes loaded: 13
   mag           time                                       place  \
0  3.5  1752379426618   2 km E of Santa María de Jesús, Guatemala
1  4.0  1752136175006                                        None
2  3.5  1752121604745             4 km SW of Amatitlán, Guatemala
3  3.4  1752042767191  4 km SE of Santa María de Jesús, Guatemala
4  3.7  1752041597610            4 km SSW of Amatitlán, Guatemala

                   geometry
0  POINT (-90.6886 14.4978)
1  POINT (-90.6664 14.5484)
2    POINT (-90.6657 14.45)
3    POINT (-90.68 14.4667)
4  POINT (-90.6549 14.4468)
```

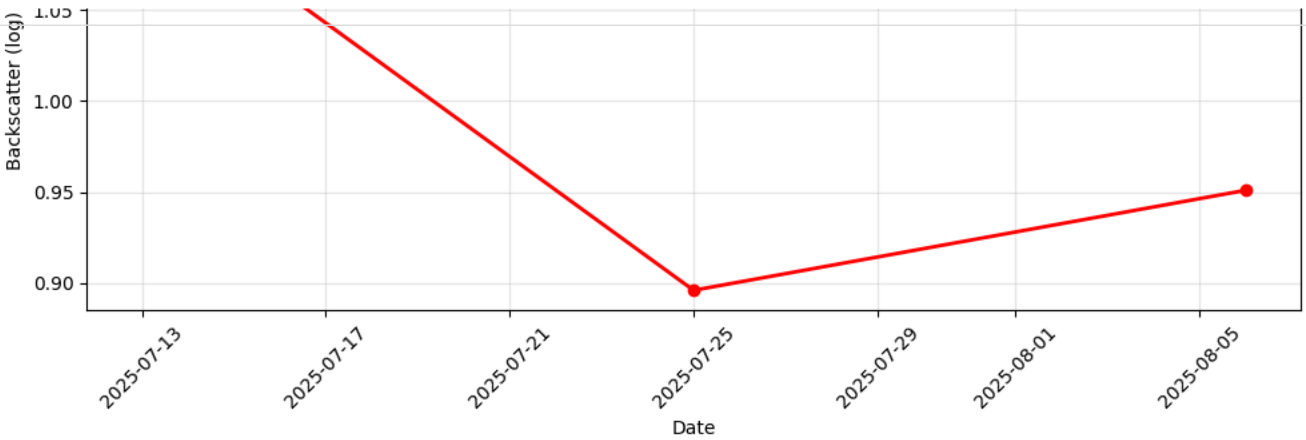
Monitoring Surface Changes Using Sentinel-1 Backscatter in Santa María de Jesús

```
import rasterio
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from datetime import datetime
import glob
import os
import warnings
warnings.filterwarnings('ignore')

from google.colab import drive
drive.mount('/content/drive')

measurement_path = "/content/drive/MyDrive/Measurement"
print(f"📁 Processing images in: {measurement_path}")

def fast_load_data():
    print("🚀 LOADING DATA...")

    tiff_files = glob.glob(os.path.join(measurement_path, "*.tiff")) + \
                 glob.glob(os.path.join(measurement_path, "*.tif"))

    print(f"📊 Found {len(tiff_files)} files")

    images_by_date = {}
    processed = 0

    for tiff_path in tiff_files[:15]:
        try:
            name = os.path.basename(tiff_path)
            print(f"   📥 {name[:30]}...")

            if '20250701' in name: date_str = '2025-07-01'
            elif '20250713' in name: date_str = '2025-07-13'
            elif '20250717' in name: date_str = '2025-07-17'
            elif '20250725' in name: date_str = '2025-07-25'
            elif '20250806' in name: date_str = '2025-08-06'
            elif '20250810' in name: date_str = '2025-08-10'
            elif '20250822' in name: date_str = '2025-08-22'
            else: continue

            if 'vv' in name.lower(): polarization = 'VV'
            elif 'vh' in name.lower(): polarization = 'VH'
            else: continue

            with rasterio.open(tiff_path) as src:
                data = src.read(1, out_shape=(1, src.height//20, src.width//20))[0]
                if np.iscomplexobj(data):
                    data = np.abs(data)
                if np.nanmin(data) >= 0:
                    data = np.log1p(data)

            if date_str not in images_by_date:
                images_by_date[date_str] = {}

            images_by_date[date_str][polarization] = {
                'data': data,
                'date': datetime.strptime(date_str, '%Y-%m-%d')
            }

            processed += 1
            print(f"     ✅ {date_str} | {polarization}")

        except Exception as e:
            print(f"     ❌ Error: {str(e)[:50]}...")
            continue

    print(f"\n✅ LOAD COMPLETE: {processed} files processed")
    return images_by_date

def generate_individual_plots(images_by_date):
    if not images_by_date:
        print("❌ NO DATA AVAILABLE")
        return

    print("\n🎯 GENERATING PLOTS...")

    vv_dates, vv_means = [], []
    vh_dates, vh_means = [], []

    for date_str, pols in sorted(images_by_date.items()):
        if 'VV' in pols:
            vv_dates.append(pols['VV']['date'])
            vv_means.append(np.nanmean(pols['VV']['data']))
        if 'VH' in pols:
            vh_dates.append(pols['VH']['date'])
            vh_means.append(np.nanmean(pols['VH']['data']))

    print(f"📅 VV data: {len(vv_dates)} dates")
    print(f"📅 VH data: {len(vh_dates)} dates")

    if vv_dates:
        print("\n1️⃣ VV PLOT...")
        plt.figure(figsize=(10, 5))
        plt.plot(vv_dates, vv_means, 'bo-', linewidth=2, markersize=6)
        plt.title('Backscatter - VV Polarization\nSanta María de Jesús', weight='bold')
        plt.xlabel('Date')
        plt.ylabel('Backscatter (log)')
        plt.grid(True, alpha=0.3)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
        print(f"   📈 VV: {np.mean(vv_means):.3f} ± {np.std(vv_means):.3f}")

    if vh_dates:
        print("\n2️⃣ VH PLOT...")
        plt.figure(figsize=(10, 5))
        plt.plot(vh_dates, vh_means, 'ro-', linewidth=2, markersize=6)
        plt.title('Backscatter - VH Polarization\nSanta María de Jesús', weight='bold')
        plt.xlabel('Date')
        plt.ylabel('Backscatter (log)')
        plt.grid(True, alpha=0.3)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
        print(f"   📈 VH: {np.mean(vh_means):.3f} ± {np.std(vh_means):.3f}")

    if vv_dates and vh_dates:
        print("\n3️⃣ VV vs VH COMPARISON...")
        common_dates, vv_common, vh_common = [], [], []
        for date_str, pols in sorted(images_by_date.items()):
            if 'VV' in pols and 'VH' in pols:
                common_dates.append(pols['VV']['date'])
                vv_common.append(np.nanmean(pols['VV']['data']))
                vh_common.append(np.nanmean(pols['VH']['data']))

        if common_dates:
            plt.figure(figsize=(10, 5))
            plt.plot(common_dates, vv_common, 'bo-', label='VV', linewidth=2)
            plt.plot(common_dates, vh_common, 'ro-', label='VH', linewidth=2)
            plt.title('Comparison: VV vs VH Backscatter\nSanta María de Jesús', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('Backscatter (log)')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()
            print(f"   🔄 Common dates: {len(common_dates)}")

    if vv_dates and vh_dates:
        print("\n4️⃣  VV/VH RATIO...")
        ratios, ratio_dates = [], []

        for date_str, pols in sorted(images_by_date.items()):
            if 'VV' in pols and 'VH' in pols:
                vv_val = np.nanmean(pols['VV']['data'])
                vh_val = np.nanmean(pols['VH']['data'])
                if vh_val > 0:
                    ratios.append(vv_val / vh_val)
                    ratio_dates.append(pols['VV']['date'])

        if ratios:
            plt.figure(figsize=(10, 4))
            plt.plot(ratio_dates, ratios, 'go-', linewidth=2, markersize=5)
            plt.axhline(y=np.mean(ratios), color='red', linestyle='--',
                        label=f'Average: {np.mean(ratios):.2f}')
            plt.title('VV/VH Ratio - Change Indicator\nSanta María de Jesús', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('VV/VH Ratio')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()
            print(f"   📊 Average ratio: {np.mean(ratios):.3f}")

    print("\n" + "="*50)
    print("📋 FINAL SUMMARY")
    print("="*50)

    if vv_dates:
        trend_vv = "↗️ INCREASING" if vv_means[-1] > vv_means[0] else "↘️ DECREASING"
        print(f"VV: {len(vv_dates)} images | Trend: {trend_vv}")
    if vh_dates:
        trend_vh = "↗️ INCREASING" if vh_means[-1] > vh_means[0] else "↘️ DECREASING"
        print(f"VH: {len(vh_dates)} images | Trend: {trend_vh}")

print("🚀 STARTING COMPLETE ANALYSIS")
print("="*40)

data = fast_load_data()
generate_individual_plots(data)

print("\n✅ ANALYSIS COMPLETED SUCCESSFULLY")
```

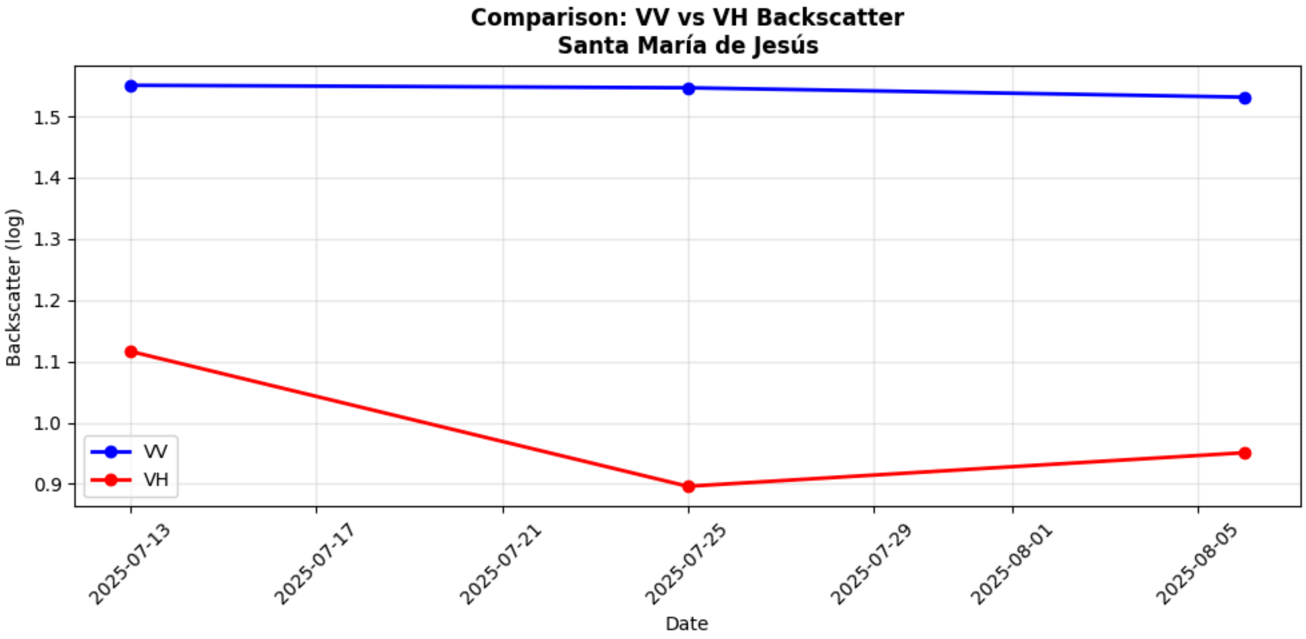
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
📁 Processing images in: /content/drive/MyDrive/Measurement
🚀 STARTING COMPLETE ANALYSIS
========================================
🚀 LOADING DATA...
📊 Found 64 files
   📥 s1a-iw1-slc-vh-20250725t001438...
     ✅ 2025-07-25 | VH
   📥 s1a-iw2-slc-vh-20250725t001436...
     ✅ 2025-07-25 | VH
   📥 s1a-iw1-slc-vv-20250725t001438...
     ✅ 2025-07-25 | VV
   📥 s1a-iw2-slc-vv-20250725t001436...
     ❌ Error: s1a-iw2-slc-vv-20250725t001436-20250725t001502-060...
   📥 Copia de s1a-iw2-slc-vh-202507...
     ✅ 2025-07-13 | VH
   📥 Copia de s1a-iw3-slc-vv-202507...
     ✅ 2025-07-13 | VV
   📥 Copia de s1a-iw3-slc-vh-202507...
     ✅ 2025-07-13 | VH
   📥 Copia de s1a-iw2-slc-vv-202507...
     ✅ 2025-07-13 | VV
   📥 Copia de s1a-iw1-slc-vv-202507...
     ✅ 2025-07-13 | VV
   📥 Copia de s1a-iw1-slc-vh-202507...
     ✅ 2025-07-13 | VH
   📥 Copia de s1a-iw3-slc-vv-202508...
     ✅ 2025-08-06 | VV
   📥 Copia de s1a-iw3-slc-vh-202508...
     ✅ 2025-08-06 | VH
   📥 Copia de s1a-iw2-slc-vv-202508...
     ✅ 2025-08-06 | VV
   📥 Copia de s1a-iw2-slc-vh-202508...
     ✅ 2025-08-06 | VH
   📥 Copia de s1a-iw1-slc-vv-202508...
     ✅ 2025-08-06 | VV

✅ LOAD COMPLETE: 14 files processed

🎯 GENERATING PLOTS...
📅 VV data: 3 dates
📅 VH data: 3 dates

1️⃣  VV PLOT...
```



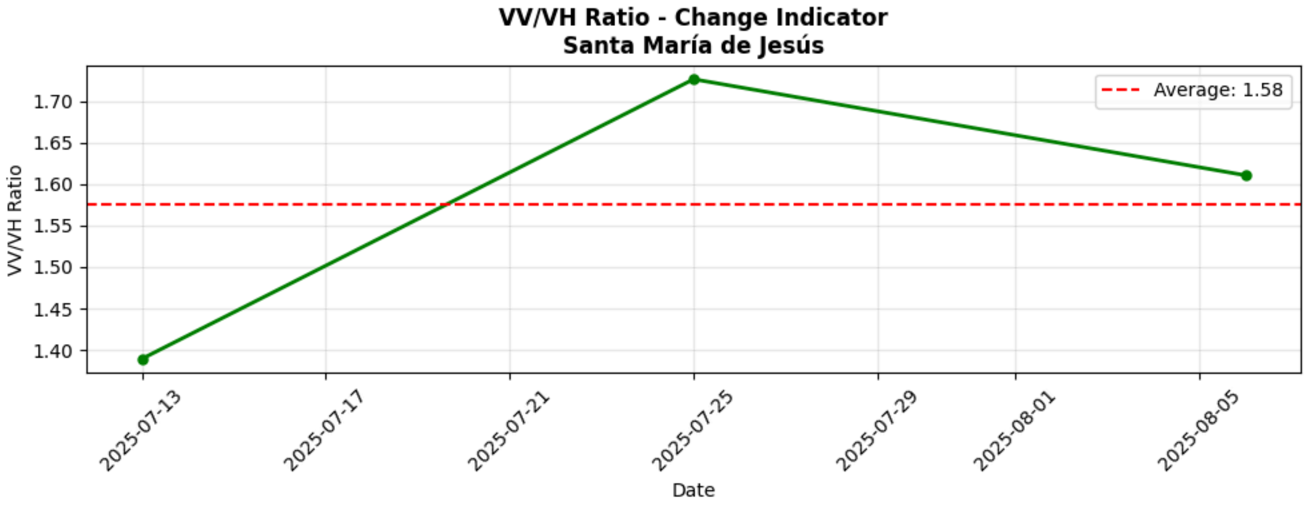
```
   📈 VV: 1.543 ± 0.008

2️⃣  VH PLOT...
```



```
📈 VH: 0.988 ± 0.094

3️⃣ VV vs VH COMPARISON...
```



```
🔄 Common dates: 3

4️⃣ VV/VH RATIO...
```



```
📊 Average ratio: 1.575

==================================================
📋 FINAL SUMMARY
==================================================
VV: 3 images | Trend: ↘️ DECREASING
VH: 3 images | Trend: ↘️ DECREASING

✅ ANALYSIS COMPLETED SUCCESSFULLY
```

By plotting this, we can observe how the land’s surface responds differently in each polarization. In Santa María de Jesús, we notice VV values remain relatively stable, while VH shows more variability. This indicates subtle changes in vegetation, moisture, or soil properties.

Overall, these visualizations help detect surface and environmental changes over time, which is crucial for disaster risk monitoring, agriculture, and ecological assessments.

Ground Movement Forecast - Prediction

In the highlands of Guatemala, just below the slopes of Volcán de Agua, lies the town of Santa María de Jesús — a community that has long lived under the quiet tension of the earth beneath it. Movements here are rarely visible to the naked eye, yet they tell stories of pressure, moisture, and time.

Our team wanted to understand those stories. Using radar data from the Sentinel-1 satellites, we analyzed the Earth’s surface through its reflections — the VV and VH backscatter signals. These signals reveal how rough or moist the ground is, and when they change, they hint that the terrain itself may be shifting.

```
import rasterio
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from datetime import datetime, timedelta
import glob
import os
import warnings
warnings.filterwarnings('ignore')

from google.colab import drive
drive.mount('/content/drive')

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA

measurement_path = "/content/drive/MyDrive/Measurement"
print(f"Processing images in: {measurement_path}")

def fast_load_data():
    print("LOADING DATA...")

    tiff_files = glob.glob(os.path.join(measurement_path, "*.tiff")) + \
                 glob.glob(os.path.join(measurement_path, "*.tif"))

    print(f"Found {len(tiff_files)} files")

    images_by_date = {}
    processed = 0

    for tiff_path in tiff_files[:15]:
        try:
            name = os.path.basename(tiff_path)
            print(f"   {name[:30]}...")

            if '20250701' in name: date_str = '2025-07-01'
            elif '20250713' in name: date_str = '2025-07-13'
            elif '20250717' in name: date_str = '2025-07-17'
            elif '20250725' in name: date_str = '2025-07-25'
            elif '20250806' in name: date_str = '2025-08-06'
            elif '20250810' in name: date_str = '2025-08-10'
            elif '20250822' in name: date_str = '2025-08-22'
            else: continue

            if 'vv' in name.lower(): polarization = 'VV'
            elif 'vh' in name.lower(): polarization = 'VH'
            else: continue

            with rasterio.open(tiff_path) as src:
                data = src.read(1, out_shape=(1, src.height//20, src.width//20))[0]
                if np.iscomplexobj(data):
                    data = np.abs(data)
                if np.nanmin(data) >= 0:
                    data = np.log1p(data)

            if date_str not in images_by_date:
                images_by_date[date_str] = {}

            images_by_date[date_str][polarization] = {
                'data': data,
                'date': datetime.strptime(date_str, '%Y-%m-%d')
            }

            processed += 1
            print(f"     {date_str} | {polarization}")

        except Exception as e:
            print(f"     Error: {str(e)[:50]}...")
            continue

    print(f"LOAD COMPLETE: {processed} files processed")
    return images_by_date

def prepare_time_series(vv_dates, vv_means, vh_dates, vh_means):
    if vv_dates:
        vv_df = pd.DataFrame({
            'date': vv_dates,
            'value': vv_means
        })
        vv_df['date'] = pd.to_datetime(vv_df['date'])
        vv_df.set_index('date', inplace=True)
        vv_df = vv_df.asfreq('D', method='ffill')
    else:
        vv_df = pd.DataFrame()

    if vh_dates:
        vh_df = pd.DataFrame({
            'date': vh_dates,
            'value': vh_means
        })
        vh_df['date'] = pd.to_datetime(vh_df['date'])
        vh_df.set_index('date', inplace=True)
        vh_df = vh_df.asfreq('D', method='ffill')
    else:
        vh_df = pd.DataFrame()

    return vv_df, vh_df

def forecast_series(df, steps=6, method='linear'):
    if df.empty:
        return None, None

    last_date = df.index[-1]
    future_dates = pd.date_range(start=last_date + timedelta(days=1), periods=steps, freq='MS')

    if method == 'linear':
        X = np.arange(len(df)).reshape(-1, 1)
        y = df['value'].values
        model = LinearRegression()
        model.fit(X, y)

        future_X = np.arange(len(df), len(df) + steps).reshape(-1, 1)
        predictions = model.predict(future_X)

        pred_df = pd.DataFrame({'date': future_dates, 'predicted_value': predictions})
        pred_df.set_index('date', inplace=True)

        full_X = np.arange(len(df) + steps).reshape(-1, 1)
        full_y = np.concatenate([y, predictions])

        return full_y, pred_df

    elif method == 'arima':
        try:
            model = ARIMA(df['value'], order=(1,1,1))
            model_fit = model.fit()
            forecast = model_fit.forecast(steps=steps)
            pred_df = pd.DataFrame({'date': future_dates, 'predicted_value': forecast})
            pred_df.set_index('date', inplace=True)

            return None, pred_df
        except Exception as e:
            print(f"ARIMA failed: {e}. Using linear fallback.")
            return forecast_series(df, steps, 'linear')

    return None, None

def generate_forecast_plots(vv_df, vh_df):
    if vv_df.empty and vh_df.empty:
        print("NO DATA FOR FORECASTING")
        return

    steps = 6

    print(f"GENERATING FORECASTS FOR {steps} MONTHS...")

    if not vv_df.empty:
        print("\nVV FORECAST...")
        full_vv, vv_pred = forecast_series(vv_df, steps, 'linear')
        if full_vv is not None:
            plt.figure(figsize=(12, 6))
            hist_dates = vv_df.index
            plt.plot(hist_dates, vv_df['value'], 'bo-', linewidth=2, markersize=6, label='Historical VV')
            pred_dates = vv_pred.index if vv_pred is not None else pd.date_range(start=vv_df.index[-1] + timedelta(days=1), perioc
            if vv_pred is not None:
                plt.plot(pred_dates, vv_pred['predicted_value'], 'b--', linewidth=2, label='Predicted VV')
                pred_std = np.std(vv_df['value']) * np.sqrt(np.arange(1, steps+1))
                plt.fill_between(pred_dates,
                                vv_pred['predicted_value'] - pred_std,
                                vv_pred['predicted_value'] + pred_std,
                                alpha=0.3, color='blue', label='Uncertainty')

            plt.title('VV Backscatter Forecast - Santa María de Jesús (Next 6 Months)', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('Backscatter (log)')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()

            trend_pred = "INCREASING" if vv_pred['predicted_value'].iloc[-1] > vv_df['value'].iloc[-1] else "DECREASING"
            print(f"VV Forecast Trend: {trend_pred} | Final pred: {vv_pred['predicted_value'].iloc[-1]:.3f}")

    if not vh_df.empty:
        print("\nVH FORECAST...")
        full_vh, vh_pred = forecast_series(vh_df, steps, 'linear')
        if full_vh is not None:
            plt.figure(figsize=(12, 6))
            hist_dates = vh_df.index
            plt.plot(hist_dates, vh_df['value'], 'ro-', linewidth=2, markersize=6, label='Historical VH')
            pred_dates = vh_pred.index if vh_pred is not None else pd.date_range(start=vh_df.index[-1] + timedelta(days=1), perioc
            if vh_pred is not None:
                plt.plot(pred_dates, vh_pred['predicted_value'], 'r--', linewidth=2, label='Predicted VH')
                pred_std = np.std(vh_df['value']) * np.sqrt(np.arange(1, steps+1))
                plt.fill_between(pred_dates,
                                vh_pred['predicted_value'] - pred_std,
                                vh_pred['predicted_value'] + pred_std,
                                alpha=0.3, color='red', label='Uncertainty')

            plt.title('VH Backscatter Forecast - Santa María de Jesús (Next 6 Months)', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('Backscatter (log)')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()

            trend_pred = "INCREASING" if vh_pred['predicted_value'].iloc[-1] > vh_df['value'].iloc[-1] else "DECREASING"
            print(f"VH Forecast Trend: {trend_pred} | Final pred: {vh_pred['predicted_value'].iloc[-1]:.3f}")

    if not vv_df.empty and not vh_df.empty:
        print("\nVV/VH RATIO FORECAST...")
        common_dates = vv_df.index.intersection(vh_df.index)
        if len(common_dates) > 1:
            ratios_hist = []
            for date in common_dates:
                vv_val = vv_df.loc[date, 'value']
                vh_val = vh_df.loc[date, 'value']
                if vh_val > 0:
                    ratios_hist.append(vv_val / vh_val)

            if ratios_hist:
                ratio_df = pd.DataFrame({
                    'date': list(common_dates)[:len(ratios_hist)],
                    'value': ratios_hist
                })
                ratio_df['date'] = pd.to_datetime(ratio_df['date'])
                ratio_df.set_index('date', inplace=True)

                _, ratio_pred = forecast_series(ratio_df, steps, 'linear')
                if ratio_pred is not None:
                    plt.figure(figsize=(12, 6))
                    plt.plot(ratio_df.index, ratio_df['value'], 'go-', linewidth=2, markersize=6, label='Historical Ratio')
                    plt.plot(ratio_pred.index, ratio_pred['predicted_value'], 'g--', linewidth=2, label='Predicted Ratio')
                    pred_std = np.std(ratio_df['value']) * np.sqrt(np.arange(1, steps+1))
                    plt.fill_between(ratio_pred.index,
                                    ratio_pred['predicted_value'] - pred_std,
                                    ratio_pred['predicted_value'] + pred_std,
                                    alpha=0.3, color='green', label='Uncertainty')

                    plt.axhline(y=np.mean(ratios_hist), color='red', linestyle='--',
                                label=f'Historical Avg: {np.mean(ratios_hist):.2f}')
                    plt.title('VV/VH Ratio Forecast - Santa María de Jesús (Next 6 Months)', weight='bold')
                    plt.xlabel('Date')
                    plt.ylabel('VV/VH Ratio')
                    plt.legend()
                    plt.grid(True, alpha=0.3)
                    plt.xticks(rotation=45)
                    plt.tight_layout()
                    plt.show()

                    ratio_trend = "INCREASING" if ratio_pred['predicted_value'].iloc[-1] > ratio_df['value'].iloc[-1] else "DECREA
                    print(f"Ratio Forecast Trend: {ratio_trend} | Final pred: {ratio_pred['predicted_value'].iloc[-1]:.3f}")

    print("\nFORECAST SUMMARY")
    print("Predictions use linear regression (effective with limited data).")
    print("For higher accuracy, use longer historical data or advanced models like LSTM.")
    print("Uncertainty bands increase over time due to error propagation.")

def generate_individual_plots(images_by_date):
    if not images_by_date:
        print("NO DATA AVAILABLE")
        return

    print("GENERATING PLOTS...")

    vv_dates, vv_means = [], []
    vh_dates, vh_means = [], []

    for date_str, pols in sorted(images_by_date.items()):
        if 'VV' in pols:
            vv_dates.append(pols['VV']['date'])
            vv_means.append(np.nanmean(pols['VV']['data']))
        if 'VH' in pols:
            vh_dates.append(pols['VH']['date'])
            vh_means.append(np.nanmean(pols['VH']['data']))

    print(f"VV data: {len(vv_dates)} dates")
    print(f"VH data: {len(vh_dates)} dates")

    if vv_dates:
        plt.figure(figsize=(10, 5))
        plt.plot(vv_dates, vv_means, 'bo-', linewidth=2, markersize=6)
        plt.title('Backscatter - VV Polarization (Santa María de Jesús)', weight='bold')
        plt.xlabel('Date')
        plt.ylabel('Backscatter (log)')
        plt.grid(True, alpha=0.3)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
        print(f"VV: {np.mean(vv_means):.3f} ± {np.std(vv_means):.3f}")

    if vh_dates:
        plt.figure(figsize=(10, 5))
        plt.plot(vh_dates, vh_means, 'ro-', linewidth=2, markersize=6)
        plt.title('Backscatter - VH Polarization (Santa María de Jesús)', weight='bold')
        plt.xlabel('Date')
        plt.ylabel('Backscatter (log)')
        plt.grid(True, alpha=0.3)
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
        print(f"VH: {np.mean(vh_means):.3f} ± {np.std(vh_means):.3f}")

    if vv_dates and vh_dates:
        common_dates, vv_common, vh_common = [], [], []

        for date_str, pols in sorted(images_by_date.items()):
            if 'VV' in pols and 'VH' in pols:
                common_dates.append(pols['VV']['date'])
                vv_common.append(np.nanmean(pols['VV']['data']))
                vh_common.append(np.nanmean(pols['VH']['data']))

        if common_dates:
            plt.figure(figsize=(10, 5))
            plt.plot(common_dates, vv_common, 'bo-', label='VV', linewidth=2)
            plt.plot(common_dates, vh_common, 'ro-', label='VH', linewidth=2)
            plt.title('Comparison: VV vs VH Backscatter (Santa María de Jesús)', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('Backscatter (log)')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()
            print(f"Common dates: {len(common_dates)}")

    if vv_dates and vh_dates:
        ratios, ratio_dates = [], []

        for date_str, pols in sorted(images_by_date.items()):
            if 'VV' in pols and 'VH' in pols:
                vv_val = np.nanmean(pols['VV']['data'])
                vh_val = np.nanmean(pols['VH']['data'])
                if vh_val > 0:
                    ratios.append(vv_val / vh_val)
                    ratio_dates.append(pols['VV']['date'])

        if ratios:
            plt.figure(figsize=(10, 4))
            plt.plot(ratio_dates, ratios, 'go-', linewidth=2, markersize=5)
            plt.axhline(y=np.mean(ratios), color='red', linestyle='--',
                        label=f'Average: {np.mean(ratios):.2f}')
            plt.title('VV/VH Ratio - Change Indicator (Santa María de Jesús)', weight='bold')
            plt.xlabel('Date')
            plt.ylabel('VV/VH Ratio')
            plt.legend()
            plt.grid(True, alpha=0.3)
            plt.xticks(rotation=45)
            plt.tight_layout()
            plt.show()
            print(f"Average ratio: {np.mean(ratios):.3f}")

    print("\nFINAL SUMMARY")

    if vv_dates:
        trend_vv = "INCREASING" if vv_means[-1] > vv_means[0] else "DECREASING"
        print(f"VV: {len(vv_dates)} images | Trend: {trend_vv}")
    if vh_dates:
        trend_vh = "INCREASING" if vh_means[-1] > vh_means[0] else "DECREASING"
        print(f"VH: {len(vh_dates)} images | Trend: {trend_vh}")

    vv_df, vh_df = prepare_time_series(vv_dates, vv_means, vh_dates, vh_means)
    generate_forecast_plots(vv_df, vh_df)
```

```
if __name__ == '__main__':
    images_data = fast_load_data()
    generate_individual_plots(images_data)
```