


# Práctico Pol-SAR Python

Armando Marino

## Ejecutar un cuaderno en Jupyter Notebook

El primer paso después de instalar Anaconda es abrir una sesión de Jupyter Notebook. Si está

usando Windows OS, debe ver un icono como este en su escritorio , o puede hacer clic en el menú de Inicio y buscar Jupyter Notebook entre las opciones.

Si está usando Mac OS, use Launchpad y haga clic en Anaconda-Navigator. Cuando Anaconda abra, cree un Jupyter Notebook.

Jupyter abrirá en un navegador. Una vez que se abra, cree una nueva carpeta a la que pueda cargar los scripts. Puede hacer esto usando los botones “New” y “Upload” en la parte superior derecha.



Debe empezar con el cuaderno ***PolSAR\_TimeSeries.ipynb***. En este archivo, va a ejecutar código para preprocesar los datos a un nivel útil para el siguiente script de aprendizaje automático (ML). Después de crear los cubos de datos (datacubes), por favor ejecute ***PolSAR\_TimeSeries\_ML\_QUIZ.ipynb***.

Notará que necesita llenar las partes del script donde aparece la palabra “QUIZ”. El archivo ***PolSAR\_TimeSeries\_ML.ipynb*** contiene las soluciones. Mi sugerencia es que intente resolver todos los QUIZZES primero y si tiene alguna dificultad, puede darle una mirada a las soluciones. Durante el video, hablaré de las soluciones.

Puede abrir el script haciendo clic en el archivo.

**Appying Machine Learning to PoISAR Time Series using Python (on Jupyter Notebook): Part 1**

Armando Marino, The University of Stirling, UK

In this script, you will process a time series of **Sentinel-1** Imagery © (Euopen Space Agency [ESA]) acquired in 2019. The 19 images were acquired over Angus, a region near Dundee, in Scotland. They are a time series of dual-polarimetric C-band images.

In the following practical, I have pre-processed the Sentinel-1 images to speed up the work and to make sure the files sizes are not too large for download. However, I encourage you to download the full images yourself (either from Copernicus ESA Data Hub or the Alaskan Satellite Facility) at your leisure. To get to the processing level we use in this practical, I had to calibrate the data, apply the orbit file, make a subset, produce the covariance matrix, multilook 4x1 to make the pixel more squared, co-registered the stack, and geocode the images. The images are saved in binary ENVI format that can be easily read by any GIS software.

This is **Part 1** of a two-part practical. You will not be able to run the second part if you have not done the first. I separated the scripts in two because Part 1 focuses on the pre-processing of the polarimetric time series and **Part 2** is applying machine learning methodologies to the pre-processed cubes.

A Google Earth image of the study area is below.

Hay varias instrucciones en la parte “Markdown” del código (o sea, las celdas azules) que explican cómo ejecutar el código. Las instrucciones dentro de las celdas verdes tienen un “In []” en la parte izquierda. Para ejecutar una celda, haga clic en este y presione Ctrl+Enter en Windows (o Shift+return en una Mac). Para asegurarse de estar definiendo todas las variables en el momento correcto, debe ejecutar las celdas en su secuencia.

Si hay un error en alguna celda, aparecerá en rojo, como en la siguiente captura de pantalla de cuando intenté ejecutar una celda subsiguiente sin haber ejecutado la celda anterior primero.

```
del T13Full_real, T13Full_imag

##### Loading T23 #####
fileT23_real = QUIZ
T23Full_real = QUIZ
fileT23_imag = QUIZ
T23Full_imag = QUIZ

T23Full = QUIZ
del QUIZ

-----
NameError                                Traceback (most recent call last)
<ipython-input-1-092a51d6979a> in <module>
      2 # First we need to identify the name of the file for the HH image
      3 fileT11 = "T11"
----> 4 T11Full = Open_ENVI_Image(path + fileT11, col, row, dtype)
      5 # Full stand for "Entire image"
      6 # Notice I am calling a function by writing its name and passing the parameters separated by a comma.

NameError: name 'Open_ENVI_Image' is not defined
```

Si en algún momento se queda atascado durante el práctico, por favor dele una mirada al archivo de soluciones para ver si las entiende. Si no entiende algo o desea saber más, por favor haga su pregunta durante la sesión para preguntas y respuestas o comuníquese conmigo directamente.

## Para ejecutar el código de Python en casa

Para poder ejecutar Python en casa, necesita instalarlo en su máquina. Mi sugerencia es que utilicen el instalador *Anaconda*, ya que este facilita las cosas (también cuando necesite instalar bibliotecas en el futuro al usar *conda*).

<https://www.anaconda.com/products/individual#download-section>

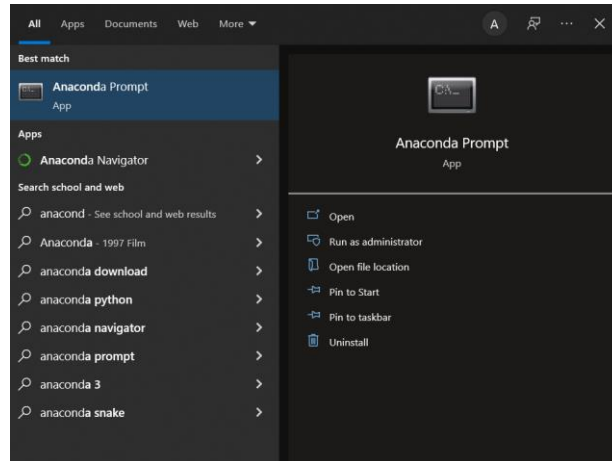


Mi sugerencia es que elijan la última versión (p.ej., 3.9) y no la versión 2.7, la que no es compatible con las últimas versiones 3.x.

Una vez que tenga Python instalado, necesitará un editor (para escribir y ejecutar el código). Puede utilizar Jupyter Notebook (como en el práctico) o el editor Spyder. Puede encontrar ambos en la instalación de Anaconda. Aunque Jupyter lo hallo mejor para enseñar, puede que usted quiera utilizar Spyder para desarrollar proyectos, pues permite depurar el código de manera más agilizada. Puede convertir el código del formato Jupyter al formato Spyder muy fácilmente al descargar el código.

Diseñé este práctico para que cualquiera lo pudiera ejecutar sin necesidad de instalar bibliotecas adicionales. Sin embargo, en el script doy la opción de usar una biblioteca adicional llamada Spectral. Puede intentar instalarla usando conda forge, abriendo Anaconda Prompt y escribiendo el texto sugerido por Spectral.

<https://anaconda.org/conda-forge/spectral>



Si no logra instalarla en este momento, está bien. Igual puede realizar el práctico sin esta biblioteca.