## Introduction

The “*Assignment EGM 722”* GitHub repository is a remote sensing toolbox for Python. The repository contains two Jupyter Notebook scripts; *NDVI\_Script* and *NDMI\_Script*. The former is designed to conduct a normalised difference vegetation index (NDVI) on satellite imagery from the Landsat collection (Landsat 4, 5, 7, or 8). The latter conducts a normalised difference moisture index (NDMI) on satellite imagery from the Landsat collection (Landsat 4, 5, 7, or 8). The user must download a Landsat image (as separate bands) to their local machine. It is recommended that the user replicates the *environment.yml* file from the repository, and then stores the images in the same folder as their own environment. It is important that the user removes band 8 from Landsat 8 (the panchromatic band) from their raw file, and it is recommended that they do the same with bands 10, 11 and 12 (Landsat 8 specific) as they can impact the band ordering when layer stacking an image.

Both scripts will load in the raw bands and layer stack them into a single image. This image can then be plotted (the user can specify their bands of interest for the plot). Each script contains a function to perform the analysis of interest (NDVI and NDMI). The analysis is subsequently run, and the results are categorised into customisable classes, allowing the user to display their results to highlight their specific interests. The raw and classified NDVI/NDMI can then be saved directly as .PNG files from the Jupyter Notebooks interface and used in scientific reports, for example.

## Setup and Installation:

The scripts and the accompanying and relevant metadata (such as the *environment.yml* file and the *README*) are available for download through the following link to the GitHub repository (<https://github.com/hjr26/Assignment_EGM722>). The two scripts are presented as individual Jupyter Notebooks which explain the code and the theory behind it. The *environment.yml* file allows the users to recreate the environment and automatically install the packages and dependencies that are required to perform the analysis using a graphical user interface of the user’s choosing (Anaconda Navigator is recommended). The main dependencies required for the script to run and their primary purpose as a Python package are listed below in Table 1.

Table 1: A table describing the main python packages required for the scripts and their major purpose.

|  |  |
| --- | --- |
| Python Package (REF) | Main Use |
| *Earthpy* | Facilitating working with raster and vector data |
| *Glob* | Returning all file paths that match a specific pattern |
| *Matplotlib* | Creating visualisations |
| *Numpy* | Support for arrays, matrices and mathematical functions |
| *Tifffile* | Working with GeoTIFF files |

Note that the “*Earthpy*” package may require manual installation. It is not listed in the main packages in Anaconda Navigator, for example. By launching the command prompt from their GUI, however, the user can download and install this package using the *Pip Install Earthpy* command, for example. Jupyter Notebook can subsequently be installed on the environment and launched from the GUI, and the user can navigate to the NDVI and NDMI scripts and run or edit them.

Both scripts use a Landsat 8 satellite image as an example walkthrough. This image is stored in the same directory as the user’s environment in a folder called *“Image”*, although the user may customise this in the script should they desire it. The test image was captured over northwest Brazil from 2022, which can be accessed and downloaded free of charge from the USGS Earth Explorer (USGS 2022). The test image’s code is “LC08\_L1TP\_229063\_20211031\_20211109\_01\_T1”, but the user can replace this with any Landsat image from Landsat 4, 5, 7 or 8 satellites.

## Methods:

As mentioned, both scripts are designed to perform a remote sensing analysis on Landsat imagery. The Landsat program is a satellite imagery acquisition mission operated by NASA, in conjunction with the USGS. It has been active since 1972 and has launched many satellites into orbit. These are described below in Table 2:

Table 2: The number of spectral bands for some common Landsat satellites.

|  |  |
| --- | --- |
| Satellite | Number of Spectral Bands |
| Landsat 1, 2, 3 | 4 |
| Landsat 4, 5 | 7 |
| Landsat 7 | 8 |
| Landsat 8 | 11 |

Landsat satellites have global coverage and a high temporal resolution (e.g. Landsat 8 has a revisit time of 16 days). Coupling this with the historical archive of 50 years data, Landsat imagery is an obvious tool for researchers to perform large scale analyses on large areas over time without the requirement for extensive and expensive in-situ visits. As such, uses of Landsat imagery are numerous, but common uses include land cover mapping (reviewed by Phiri and Morgenroth 2017), deforestation (e.g. Souza *et al.* 2013) and flood mapping (e.g. Wang *et al.* 2002). Landsat images can be downloaded from the USGS Earth Explorer Portal as individual bands (tar.gz file type).

The code will firstly load all image bands under a common name using the “*Glob*” package (e.g. LC08\_L1TP\_229063\_20211031\_20211109\_01\_T1\_B\*). The “*B\**” in the code will load all images that have the specified name (i.e. B1, B2, etc.; Landsat imagery downloads as individual bands with the common name in each band). The bands are then stacked together to create a multispectral image using the “*Earthpy”* package. This same package can be used to plot the stacked image, and the user can customise the bands displayed to highlight their preference (e.g. false colour vs true colour composites). Two functions are then defined (one per script) to calculate the NDVI and NDMI and classify the results. The user can input the number of classes they desire, the range of values within each class, and the colour that they want to represent the class. These functions make use of the “*Numpy*”, “*Earthpy*”, and “*Matplotlib*” packages.

NDVI (normalised difference vegetation index) is a commonly used remote sensing tool to analyse vegetation health in an area of interest (Carlson and Ripley 1997). NDVI is calculated using the near infrared (NIR) and red band, as plant chlorophyll content reduces in the red band while increasing in the NIR band (Pettorelli *et al.* 2005), using the following equation:

NDVI values can range from -1 to 1, where values approaching -1 indicate an absence of vegetation, and values approaching 1 indicate an abundance of healthy vegetation (Myeni *et al.* 1995).

NDMI (normalised difference moisture index) is another commonly used remote sensing tool. It’s primary use is to inform on vegetation water content (Xu 2006). It is calculated using the shortwave infrared (SWIR) band and NIR band, as water absorbs light in both bands (Ji *et al.* 2009), using the following equation:

NDMI values can range from -1 to 1, where values approaching -1 indicate an absence of water, and values approaching 1 indicate a high presence of water (Taloor *et al.* 2021).

## Expected Results:

The expected result from two scripts are as follows.

The *NDVI\_Script* will produce two plots which the user can save; a raw NDVI plot (e.g. Figure 1, below) and a classification of the NDVI values (e.g. Figure 2, below) for their image of interest.

Map

Description automatically generated

Figure 1: an example of the raw NDVI output PNG from the NDVI\_Script. This example image is captured in 2022 of northwest Brazil from Landsat 8. The darker green values indicate healthier vegetation, the orange to red values indicate non-vegetation.

A picture containing text, stationary, envelope, picture frame

Description automatically generated

Figure 2: An example of the classified NDVI. The colours on the image correspond to the classes in the legend (the user will define these.

Similarly, the *NDMI\_Script* will produce two plots which the user can save; a raw NDMI plot (e.g. Figure 3, below) and a classification of the NDMI values (e.g. Figure 4, below) for their image of interest.

A picture containing text, envelope

Description automatically generated

Figure 3: an example of the raw NDMI output PNG from the NDMI\_Script. This example image is captured in 2022 of northwest Brazil from Landsat 8. The darker green values indicate vegetation with high water content, the white to purple values are representative of vegetation with low water content.

A picture containing text, stationary, envelope

Description automatically generated

Figure 4: An example of the classified NDMI. The colours on the image correspond to the classes in the legend (the user will define these.

## Troubleshooting:

Some troubleshooting options for possible issues are described below. If the user encounters an issue which is not described herein, it is recommended that the user comments on the GitHub repository for additional solutions.

One issue may occur when plotting the image. The user may produce an image which does not appear as expected despite having inputted what the user deems to be the correct bands. This can happen where the bands are in the wrong order, or where the wrong image bands were present in the local directory. For example, if Landsat 8 band 8 (panchromatic band) is in the repository the user will get an error while trying to import the bands using the glob function. The panchromatic band is a different size and cannot be present in the repository. Delete this band file and retry the import code. Following on from this, there is a line of code commented out which the user can run to test their band order “*print(image\_path)*”. The user may notice that bands 10 – 12 are included and are being ordered incorrectly. As they band names are strings, they order as band1, band10, band 11, etc. As mentioned above, the function is designed for a repository where these bands are not present. If the user wants them to be present, then can customise the order of their bands (manipulate the “*image\_path*” variable as desired). Note that the NDVI/NDMI function will not function correctly if the Red/NIR/SWIR bands are not in the correct band positions (these are described in the function). The user can of course edit the function if they do not want to remove the unnecessary bands from their directory.

Secondly, if the raw NDVI plot appears to have very few negative or positive values according to the continuous colour plot, the user may think that the function has not run correctly. To establish if this is the case, the user can change the minimum and maximum values (vmin/vmax) for the plot to very small values (e.g. -0.05 -> 0.05). This will highlight if very small negative values are present, for example. This is more an ‘issue’ relating to the geographic location of the image of interest, so it is recommended that the user well is aware of the expected values of their area of interest, as well as the theoretical basic behind the NDVI and NDMI.

Thirdly, the user may get an error wherein the classification does not run because the number of colours and named classes are not equal (e.g. user has specified 6 classes but only 5 colour). This is documented in the script, but the user must ensure that the number of class names specified matches the number of colours specified.

Finally, the user may encounter error messages where they are being informed that the one (or more) of the modules cannot be found. The user should ensure that they are using the correct environment in their GUI, and that it contains the packages required to run the scripts. As mentioned above, in the case that the missing package is “*Earthpy*”, the user must ensure to install it via the command prompt as it may not be a suggested package in the GUI (e.g. Anaconda Navigator). It is recommended that the user downloads the *environment.yml* file from the repository and creates their own environment by importing this one, as this will automatically install the required packages.

## References:

Carlson, T.N. and Ripley, D.A., 1997. On the relation between NDVI, fractional vegetation cover, and leaf area index. *Remote sensing of Environment*, *62*(3), pp.241-252.

Ji, L., Zhang, L. and Wylie, B., 2009. Analysis of dynamic thresholds for the normalized difference water index. *Photogrammetric Engineering & Remote Sensing*, *75*(11), pp.1307-1317.

Myneni, R.B., Hall, F.G., Sellers, P.J. and Marshak, A.L., 1995. The interpretation of spectral vegetation indexes. *IEEE Transactions on Geoscience and Remote Sensing*, *33*(2), pp.481-486

Pettorelli, N., Vik, J.O., Mysterud, A., Gaillard, J.M., Tucker, C.J. and Stenseth, N.C., 2005. Using the satellite-derived NDVI to assess ecological responses to environmental change. *Trends in ecology & evolution*, *20*(9), pp.503-510.

Phiri, D. and Morgenroth, J., 2017. Developments in Landsat land cover classification methods: A review. *Remote Sensing*, *9*(9), p.967.

Souza Jr, C.M., Siqueira, J.V., Sales, M.H., Fonseca, A.V., Ribeiro, J.G., Numata, I., Cochrane, M.A., Barber, C.P., Roberts, D.A. and Barlow, J., 2013. Ten-year Landsat classification of deforestation and forest degradation in the Brazilian Amazon. *Remote Sensing*, *5*(11), pp.5493-5513.

Taloor, A.K., Manhas, D.S. and Kothyari, G.C., 2021. Retrieval of land surface temperature, normalized difference moisture index, normalized difference water index of the Ravi basin using Landsat data. *Applied Computing and Geosciences*, *9*, p.100051.

U.S. Geological Survey, 2022, Earth Explorer, accessed April – May 2022 at URL <https://earthexplorer.usgs.gov/>

Wang, Y., Colby, J.D. and Mulcahy, K.A., 2002. An efficient method for mapping flood extent in a coastal floodplain using Landsat TM and DEM data. *International Journal of Remote Sensing*, *23*(18), pp.3681-3696.

Xu, H., 2006. Modification of normalised difference water index (NDWI) to enhance open water features in remotely sensed imagery. *International journal of remote sensing*, *27*(14), pp.3025-3033.