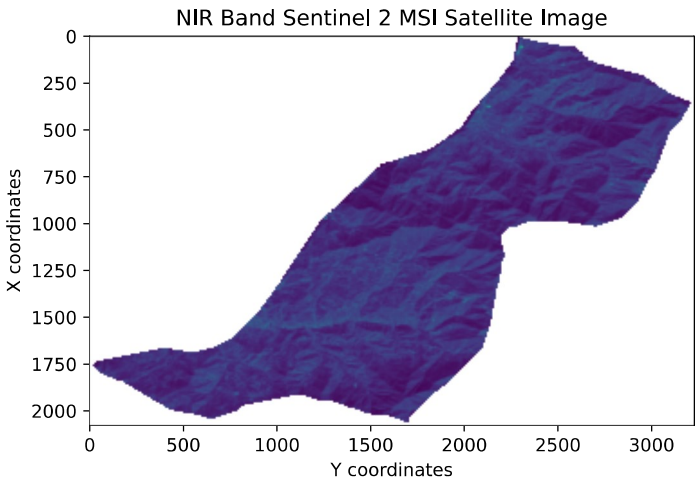
In [1]:
```
### NDVI for Sentinel 2 MSI
```

In [2]:
```
import matplotlib.pyplot as plt
from skimage import *
import numpy as np
from skimage.viewer import ImageViewer
from tifffile import *
import matplotlib.patches as mpatches
from skimage import io
import imagecodecs
```

In [3]:
```
#Read red and nir bands

red =io.imread('test_images_tiff/S2_RED.tif')
nir = io.imread('test_images_tiff/S2_NIR.tif')
```

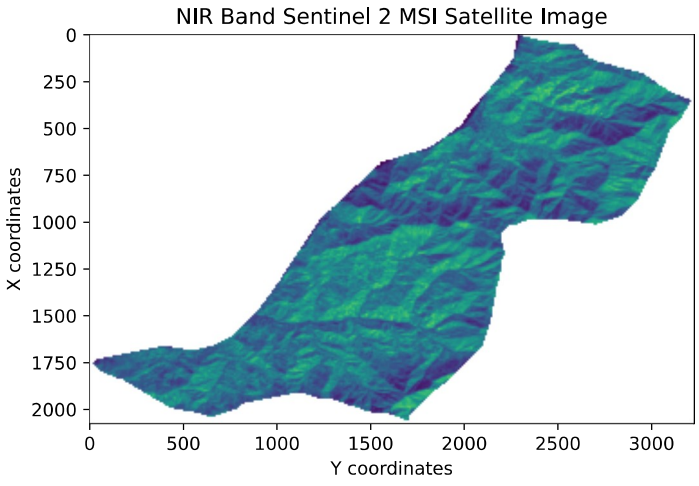
In [4]:
```
#plot red band

plt.title("NIR Band Sentinel 2 MSI Satellite Image")
plt.xlabel("Y coordinates")
plt.ylabel("X coordinates")
plt.imshow(red)
```

Out[4]: <matplotlib.image.AxesImage at 0x7f2d916f56a0>



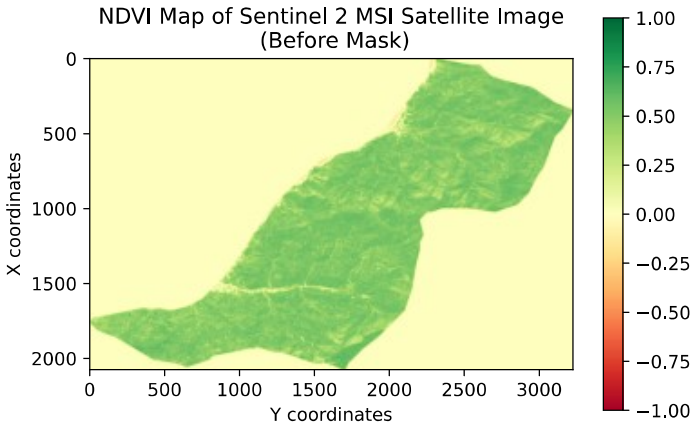
In [5]:
```
#plot nir band
plt.title("NIR Band Sentinel 2 MSI Satellite Image")
plt.xlabel("Y coordinates")
plt.ylabel("X coordinates")
plt.imshow(nir)
```

Out[5]: <matplotlib.image.AxesImage at 0x7f2d8c924970>



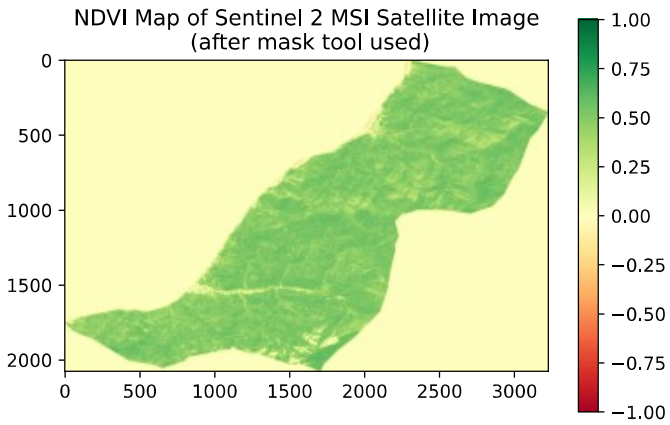
In [6]:
```
#import create_ndvi function in order to calculate ndvi
from utils import create_ndvi
ndvi=create_ndvi(nir_band=nir,red_band=red)
plt.title("NDVI Map of Sentinel 2 MSI Satellite Image\n (Before Mask)")
plt.xlabel("Y coordinates")
plt.ylabel("X coordinates")
plt.imshow(ndvi,cmap='RdYlGn',vmin=-1,vmax=1)
plt.colorbar()
```

Out[6]: <matplotlib.colorbar.Colorbar at 0x7f2d8c06a0a0>



In [7]:
```
#import mask function in order to mask water bodies
from utils import water_mask_ndvi_for_sentinel_2

img = water_mask_ndvi_for_sentinel_2(ndvi_band=ndvi,nir_band=nir)
plt.title("NDVI Map of Sentinel 2 MSI Satellite Image\n (after mask tool used)")
plt.imshow(img,cmap='RdYlGn',vmin=-1,vmax=1)
plt.colorbar()
```

Out[7]: <matplotlib.colorbar.Colorbar at 0x7f2d8c028c70>



In [8]:
```
#Create classes to have convenient legends and apply to results
ndvi_class_bins = [-np.inf, 0, 0.1, 0.25, 0.4, np.inf]
ndvi_landsat_class = np.digitize(ndvi, ndvi_class_bins)

# Apply the nodata mask to the newly classified NDVI data
ndvi_landsat_class = np.ma.masked_where(
    np.ma.getmask(ndvi), ndvi_landsat_class
)
np.unique(ndvi_landsat_class)
```

Out[8]:
```
masked_array(data=[1, 2, 3, 4, 5],
             mask=False,
       fill_value=999999)
```

In [9]:
```
# Define color map
nbr_colors = ["gray", "y", "yellowgreen", "g", "darkgreen"]
# Define class names
ndvi_cat_names = [
    "No Vegetation",
    "Bare Area",
    "Low Vegetation",
```

```
    "Moderate Vegetation",
    "High Vegetation",
]

# Get list of classes
classes = np.unique(ndvi_landsat_class)
classes = classes.tolist()
# The mask returns a value of none in the classes. remove that
classes = classes[0:5]

# Plot your data
fig, ax = plt.subplots(figsize=(12, 12))

im = ax.imshow(ndvi_landsat_class, cmap='RdYlGn')

no_veg_patch = mpatches.Patch(color='red', label='No Vegetation')
bare_patch = mpatches.Patch(color='y', label='Bare Area')
low_veg_patch = mpatches.Patch(color='yellowgreen', label='Low Vegetation')
mod_veg_patch = mpatches.Patch(color='g', label='Moderate Vegetation')
high_veg_patch= mpatches.Patch(color='darkgreen', label='High Vegetation')
no_data_patch= mpatches.Patch(color='orange', label='No Data & Water Bodies')


plt.legend(handles=[no_veg_patch,bare_patch,low_veg_patch,mod_veg_patch,high_veg_patch,no_data_patch], loc='upper left')
ax.set_title(
    "NDVI Map of Sentinel 2 MSI Satellite Image\n (After Mask Tool Used)",
    fontsize=14,
)

plt.xlabel("Y coordinates")
plt.ylabel("X coordinates")
plt.savefig('outputs/NDVI_Sentinel_2.png',format="png")
# Auto adjust subplot to fit figure size
plt.tight_layout()
```