

In [3]: `pip install earthpy`

Requirement already satisfied: earthpy in d:\anaconda\lib\site-packages (0.9.4)
Requirement already satisfied: geopandas in d:\anaconda\lib\site-packages (from earthpy) (1.1.1)
Requirement already satisfied: matplotlib>=2.0.0 in d:\anaconda\lib\site-packages (from earthpy) (3.9.2)
Requirement already satisfied: numpy>=1.14.0 in d:\anaconda\lib\site-packages (from earthpy) (1.26.4)
Requirement already satisfied: rasterio in d:\anaconda\lib\site-packages (from earthpy) (1.4.3)
Requirement already satisfied: scikit-image in d:\anaconda\lib\site-packages (from earthpy) (0.24.0)
Requirement already satisfied: requests in d:\anaconda\lib\site-packages (from earthpy) (2.32.3)
Requirement already satisfied: contourpy>=1.0.1 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (1.2.0)
Requirement already satisfied: cycloper>=0.10 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (1.4.4)
Requirement already satisfied: packaging>=20.0 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (24.1)
Requirement already satisfied: pillow>=8 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in d:\anaconda\lib\site-packages (from matplotlib>=2.0.0->earthpy) (2.9.0.post0)
Requirement already satisfied: pyogrio>=0.7.2 in d:\anaconda\lib\site-packages (from geopandas->earthpy) (0.11.1)
Requirement already satisfied: pandas>=2.0.0 in d:\anaconda\lib\site-packages (from geopandas->earthpy) (2.2.2)
Requirement already satisfied: pyproj>=3.5.0 in d:\anaconda\lib\site-packages (from geopandas->earthpy) (3.7.1)
Requirement already satisfied: shapely>=2.0.0 in d:\anaconda\lib\site-packages (from geopandas->earthpy) (2.1.1)
Requirement already satisfied: affine in d:\anaconda\lib\site-packages (from rasterio->earthpy) (2.4.0)
Requirement already satisfied: attrs in d:\anaconda\lib\site-packages (from rasterio->earthpy) (23.1.0)
Requirement already satisfied: certifi in d:\anaconda\lib\site-packages (from rasterio->earthpy) (2025.4.26)
Requirement already satisfied: click>=4.0 in d:\anaconda\lib\site-packages (from rasterio->earthpy) (8.1.7)
Requirement already satisfied: cligj>=0.5 in d:\anaconda\lib\site-packages (from rasterio->earthpy) (0.7.2)
Requirement already satisfied: click-plugins in d:\anaconda\lib\site-packages (from rasterio->earthpy) (1.1.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in d:\anaconda\lib\site-packages (from requests->earthpy) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in d:\anaconda\lib\site-packages (from requests->earthpy) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in d:\anaconda\lib\site-packages (from requests->earthpy) (2.2.3)
Requirement already satisfied: scipy>=1.9 in d:\anaconda\lib\site-packages (from scikit-image->earthpy) (1.13.1)
Requirement already satisfied: networkx>=2.8 in d:\anaconda\lib\site-packages (from scikit-image->earthpy) (3.3)
Requirement already satisfied: imageio>=2.33 in d:\anaconda\lib\site-packages (fr

```

om scikit-image->earthpy) (2.33.1)
Requirement already satisfied: tifffile>=2022.8.12 in d:\anaconda\lib\site-packag
es (from scikit-image->earthpy) (2023.4.12)
Requirement already satisfied: lazy-loader>=0.4 in d:\anaconda\lib\site-packages
(from scikit-image->earthpy) (0.4)
Requirement already satisfied: colorama in d:\anaconda\lib\site-packages (from cl
ick>=4.0->rasterio->earthpy) (0.4.6)
Requirement already satisfied: pytz>=2020.1 in d:\anaconda\lib\site-packages (fro
m pandas>=2.0.0->geopandas->earthpy) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in d:\anaconda\lib\site-packages (f
rom pandas>=2.0.0->geopandas->earthpy) (2023.3)
Requirement already satisfied: six>=1.5 in d:\anaconda\lib\site-packages (from py
thon-dateutil>=2.7->matplotlib>=2.0.0->earthpy) (1.16.0)
Note: you may need to restart the kernel to use updated packages.

```

```

In [7]: import os
import numpy as np
import pandas as pd

```

```

In [11]: from sklearn import model_selection
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, classification_rep
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import cohen_kappa_score

```

```

In [13]: import rasterio as rio
import earthpy.plot as ep
import cartopy.feature as cfeature

#from matplotlib_scalebar.scalebar import ScaleBar
from matplotlib import pyplot as plt
from matplotlib import cm
from matplotlib.colors import ListedColormap

```

```

In [15]: os.getcwd()

```

```

Out[15]: 'C:\\Users\\User'

```

```

In [18]: os.chdir('D:/Assignment')

```

```

In [21]: data=pd.read_csv('sample_kolkata - sample_kolkata.csv')
data

```

Out[21]:

	B2	B3	B4	B5	B6	B7	MNDWI	NDBI	NDVI	LULC	S
0	9335	8513	8199	10561	10483	9265	-0.103706	-0.003707	0.125906	Built-up	
1	9473	8620	8467	10446	10632	9463	-0.104509	0.008824	0.104637	Built-up	
2	9354	8313	8154	9530	10483	9590	-0.115450	0.047619	0.077810	Built-up	
3	9458	8556	8830	9389	9895	9221	-0.072571	0.026239	0.030682	Built-up	
4	9697	8681	8535	10033	11246	10498	-0.128720	0.057005	0.080676	Built-up	
...
2495	8652	7693	6811	6163	5321	5188	0.182265	-0.073319	-0.049946	water	
2496	8568	7635	6746	6121	5299	5173	0.180609	-0.071979	-0.048574	water	
2497	8614	7643	6794	6132	5344	5213	0.177023	-0.068665	-0.051215	water	
2498	8750	7892	7019	6298	5446	5285	0.183386	-0.072548	-0.054141	water	
2499	8751	7879	7028	6220	5454	5268	0.181880	-0.065616	-0.060990	water	

2500 rows × 11 columns



```
In [24]: x=data.drop(['LULC', 'SYMBOL'],axis=1)
x=x[:].values
y=data.iloc[:,-1]
y=y[:].values
```

```
In [27]: print(y)
print(y.shape)
```

```
[0 0 0 ... 0 0 0]
(2500,)
```

```
In [30]: print(x)
print(x.shape)
```

```
[[ 9.33500e+03  8.51300e+03  8.19900e+03 ... -1.03706e-01 -3.70700e-03
  1.25906e-01]
 [ 9.47300e+03  8.62000e+03  8.46700e+03 ... -1.04509e-01  8.82400e-03
  1.04637e-01]
 [ 9.35400e+03  8.31300e+03  8.15400e+03 ... -1.15450e-01  4.76190e-02
  7.78100e-02]
 ...
 [ 8.61400e+03  7.64300e+03  6.79400e+03 ...  1.77023e-01 -6.86650e-02
 -5.12150e-02]
 [ 8.75000e+03  7.89200e+03  7.01900e+03 ...  1.83386e-01 -7.25480e-02
 -5.41410e-02]
 [ 8.75100e+03  7.87900e+03  7.02800e+03 ...  1.81880e-01 -6.56160e-02
 -6.09900e-02]]
(2500, 9)
```

```
In [33]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.30,stratify=y,
```

```
In [36]: x_train.shape
```

```
Out[36]: (1750, 9)
```

```
In [40]: y_test.shape
```

```
Out[40]: (750,)
```

```
In [43]: scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

```
In [46]: lr = LogisticRegression(random_state=42)
lr.fit(x_train_scaled, y_train)

y_test_pred_LR = lr.predict(x_test_scaled)
y_train_pred_LR = lr.predict(x_train_scaled)

test_accuracy_LR = accuracy_score(y_test, y_test_pred_LR)
train_accuracy_LR = accuracy_score(y_train, y_train_pred_LR)

print("Test Accuracy:", test_accuracy_LR)
print("Train Accuracy:", train_accuracy_LR)

print("Classification Report:\n", classification_report(y_test, y_test_pred_LR))
```

Test Accuracy: 0.9893333333333333

Train Accuracy: 0.9954285714285714

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	450
1	0.98	0.99	0.99	300
accuracy			0.99	750
macro avg	0.99	0.99	0.99	750
weighted avg	0.99	0.99	0.99	750

```
In [56]: img=rio.open('compkol.tif')
```

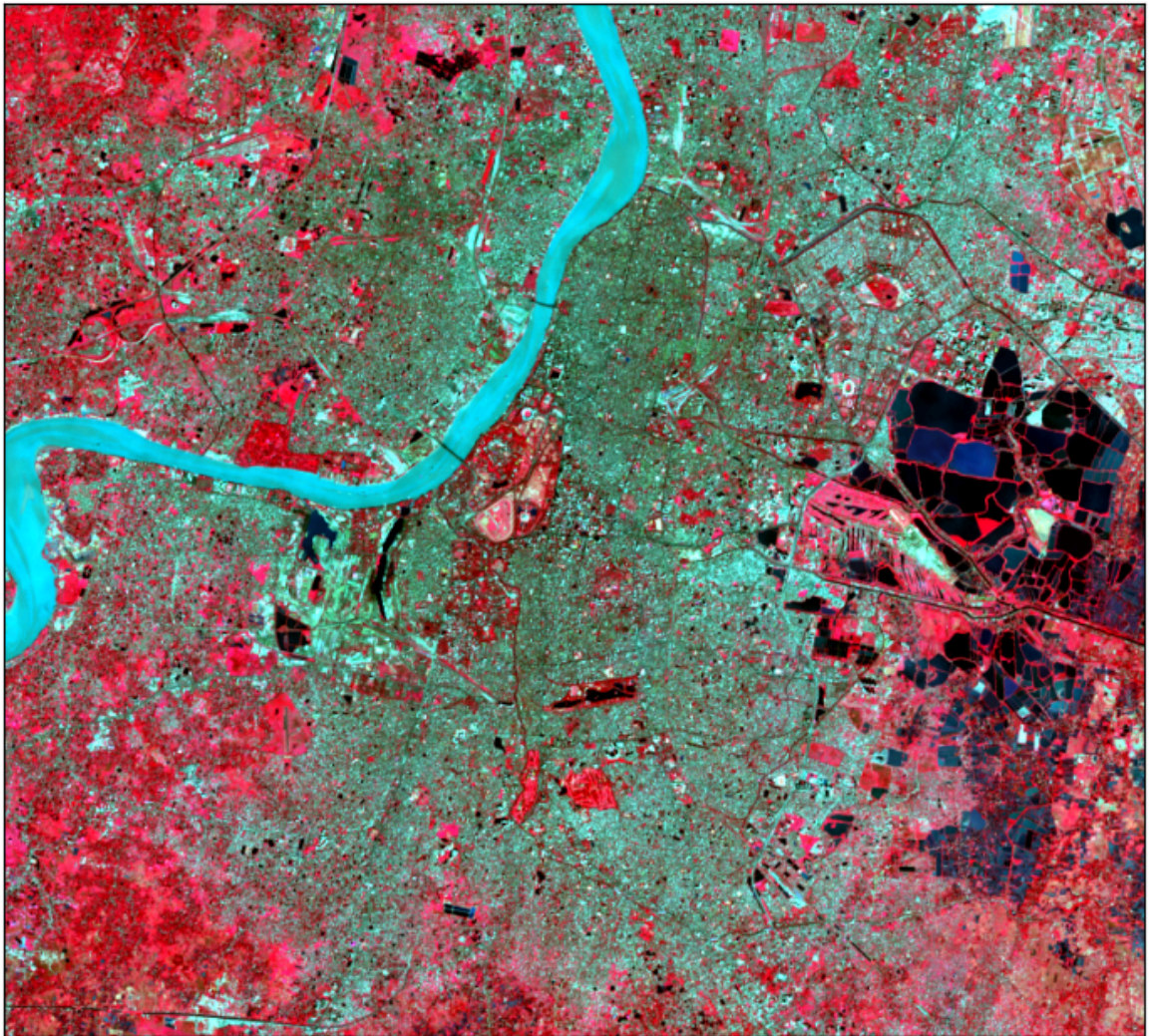
```
In [58]: img=img.read()
print(img.shape)
print(img.dtype)
```

```
(9, 726, 802)
```

```
float32
```

```
In [61]: ep.plot_rgb(img,rgb=[3,2,1],stretch=True,figsize=(10,8),title='RGB image of Kolk
#plt.savefig('RGB_1400.jpg', dpi=1000, format='jpg')
plt.show()
```

RGB image of Kolkata



```
In [65]: Image=img.transpose([1,2,0])
         Image.shape
```

```
Out[65]: (726, 802, 9)
```

```
In [68]: new_Image=np.nan_to_num(Image)
```

```
In [71]: Bands=new_Image.reshape(Image.shape[0]*Image.shape[1],Image.shape[2])
         Bands.shape
```

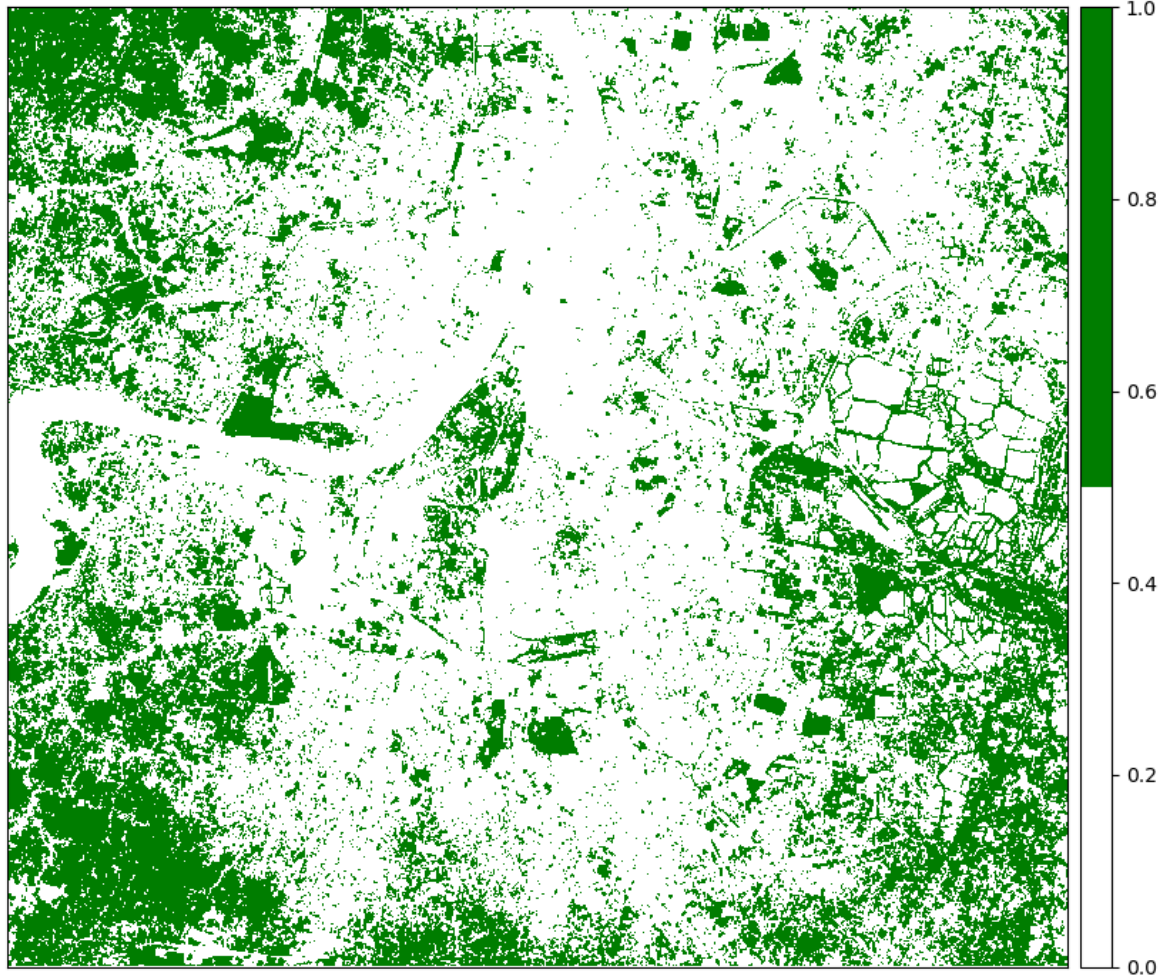
```
Out[71]: (582252, 9)
```

```
In [77]: Bands_scaled = scaler.transform(Bands)
```

```
In [80]: Pred_best_LR_Img=lr.predict(Bands_scaled)
         Pred_best_LR=Pred_best_LR_Img.reshape(Image.shape[0],Image.shape[1])
```

```
In [83]: cmap=ListedColormap(['white','green'])
         ep.plot_bands(Pred_best_LR,scale=False,figsize=(10,12),cmap=cmap,\
                       title='Green spaces of Kolkata municipality in 2021 using Logistic
#plt.savefig('PDF_plot.jpg', dpi=800, format='jpg')
plt.show()
```


Green spaces of Kolkata municipality in 2021 using Logistic Regression



In []: