

Introduction to Image Processing

Definition:

Image processing involves manipulating and analyzing digital images using algorithms to improve their quality, extract information, or automate tasks.

How it Works:

Image processing algorithms can perform a variety of tasks such as enhancing contrast, correcting distortions, and identifying patterns within the image data.

Types of Image Processing:

- **Digital Image Enhancement:**

Enhancing the visual aspects of an image to make it more pleasing or easier to interpret.

- **Image Restoration:**

Restoring the original appearance of an image that has been degraded by noise, blur, or other factors.

- **Image Segmentation:**

Dividing an image into multiple segments to simplify its representation and facilitate analysis.

Applications:

Image processing is used in various fields, including medical imaging, remote sensing, computer vision, and photography. It helps in improving image quality, diagnosing diseases, and enhancing user experiences with visual content.

Image processing continues to be an integral part of modern technology, with advancements leading to more sophisticated and automated methods for handling visual data.

Popular Python Libraries for Image Processing

OpenCV: An open-source library developed by Intel, OpenCV is widely used for computer vision tasks such as face detection, object detection, and image segmentation. It's known for its efficiency

and vast access to algorithms.

Scikit-Image: Part of the Scikit-learn family, Scikit-Image is used for various computer vision tasks. It's open-source, easy to use, and offers algorithms for segmentation, color space manipulation, and feature detection.

SciPy: A Python package that is useful for a variety of mathematical problems and procedures, including multidimensional image processing.

Pillow/PIL: Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL (Python Imaging Library) adds image processing capabilities to your Python interpreter.

Mahotas: A library for image processing and computer vision in Python. It contains traditional image processing functionality such as filtering and morphological operations.

Matplotlib: Though primarily a plotting library, Matplotlib can also be used for basic image processing tasks, such as displaying images and changing their properties.

NumPy: Essential for image processing, NumPy supports large, multi-dimensional arrays and matrices, which are fundamental for image manipulation.

▼ Understanding Images in Remote Sensing

What is an Image?

An image is a two-dimensional function, $\mathbf{F}(\mathbf{X}, \mathbf{y})$, where \mathbf{X} and \mathbf{y} are spatial coordinates. The amplitude of \mathbf{F} at any pair of coordinates (x, y) is called the intensity or brightness of the image at that point. When these values are finite, we have a digital image composed of pixels.

Types of Images Used in Remote Sensing:

Remote sensing utilizes various types of images, each serving a specific purpose:

- **Analog Images:**

Traditional photographs captured using film cameras, often used for aerial photography.

- **Digital Images:**

Captured through electronic sensors on platforms like satellites, these images are used for a wide range of applications, from environmental monitoring to urban planning.

- **Panchromatic Images:**

Black and white imagery capturing data in a single band of the electromagnetic spectrum.

- **Multispectral Images:**

Capture data at several specific wavelengths across the electromagnetic spectrum, including both visible and invisible light.

- **Hyperspectral Images:**

Similar to multispectral, but with many narrow spectral bands, providing detailed information for each pixel.

- **Radar Images:**

Utilize synthetic aperture radar (SAR) to capture images that can penetrate clouds and vegetation, useful for day and night imaging.

Applications of Remote Sensing Images:

These images are pivotal in various sectors, including:

- Environmental Monitoring: Assessing changes like deforestation and climate impact.
- Agriculture: Monitoring crop health and predicting yields.
- Urban Planning: Mapping and analyzing urban growth and infrastructure.
- Disaster Management: Providing critical information during natural disasters.

Remote sensing images are integral to our understanding of the Earth's surface and atmosphere, aiding in decision-making and scientific research.

- ✓ Example: Landsat 8 Hyperspectral Images

- ✓ Import

```
import numpy as np
import matplotlib.pyplot as plt
import skimage as skm
```

- ✓ Load the Images

```
blue = skm.io.imread("Blue.tif")
green = skm.io.imread("Green.tif")
red = skm.io.imread("Red.tif")
nir = skm.io.imread("NIR.tif")
swir1 = skm.io.imread("SWIR1.tif")
swir2 = skm.io.imread("SWIR2.tif")
```

or

```
landsat = skm.io.imread("*.tif")
```

```
len(landsat)
```

```
→ 6
```

```
blue
```

```
→ array([[-64, -71, -76, ..., 176, 174, 175],
         [-57, -34, -59, ..., 178, 173, 174],
         [-40, -3, -83, ..., 176, 173, 173],
         ...,
         [154, 137, 175, ..., 432, 418, 456],
         [179, 143, 205, ..., 454, 379, 311],
         [188, 169, 201, ..., 305, 185, 70]], dtype=int16)
```

or

```
landsat[0]
```

```
→ array([[-64, -71, -76, ..., 176, 174, 175],
         [-57, -34, -59, ..., 178, 173, 174],
         [-40, -3, -83, ..., 176, 173, 173],
         ...,
         [154, 137, 175, ..., 432, 418, 456],
         [179, 143, 205, ..., 454, 379, 311],
         [188, 169, 201, ..., 305, 185, 70]], dtype=int16)
```

```
blue.shape
```

```
→ (998, 998)
```

```
B2 = np.array(blue).astype(float)
B3 = np.array(green).astype(float)
B4 = np.array(red).astype(float)
B5 = np.array(nir).astype(float)
B6 = np.array(swir1).astype(float)
B7 = np.array(swir2).astype(float)
```

B2

```
→ array([[-64., -71., -76., ..., 176., 174., 175.],
       [-57., -34., -59., ..., 178., 173., 174.],
       [-40., -3., -83., ..., 176., 173., 173.],
       ...,
       [154., 137., 175., ..., 432., 418., 456.],
       [179., 143., 205., ..., 454., 379., 311.],
       [188., 169., 201., ..., 305., 185., 70.]])
```

B2[0][0]

```
→ -64.0
```

B2[1][1]

```
→ -34.0
```

▼ .ndim

B2.ndim

```
→ 2
```

▼ .size

B2.size

```
→ 996004
```

▼ .dtype

B2.dtype

```
→ dtype('float64')
```

▼ Layer Stacking

```
stack = np.stack([B4,B3,B2], axis=2)
```

▼ Display the Image

```
plt.imshow(stack)
plt.title('Landsat 8')
plt.axis('off')
plt.show()
```

→ Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

Landsat 8



OOPS!

The message you're seeing, "Clipping input data to the valid range for imshow with RGB data ([0...1] for floats or [0...255] for integers)," is a warning from the matplotlib library in Python. It typically occurs when you're trying to display an image using `imshow` and the data type of the image array is not as expected.

Here's what's happening:

`imshow` expects the RGB values to be within the range of [0...1] if the data type is floating-point (floats), and [0...255] if the data type is integers. If your image data has values outside these ranges, `imshow` will automatically clip them to fit within the valid range. This can happen if the image data is processed or scaled incorrectly, leading to values that are too high or too low. To fix this, you should ensure that your image data is correctly scaled before displaying it. If you're working with floating-point data, you can normalize your image array by dividing by the maximum value (usually 255):

▼ Normalize

Min-Max scaler

The **Min-Max scaler** is a preprocessing technique used to normalize the range of features in your dataset. It scales the data to a default range of 0 to 1 but can be adjusted to any specific range you define.

Equation

The transformation is given by:

$$X_{\text{std}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

$$X_{\text{scaled}} = X_{\text{std}} \times (\text{max} - \text{min}) + \text{min}$$

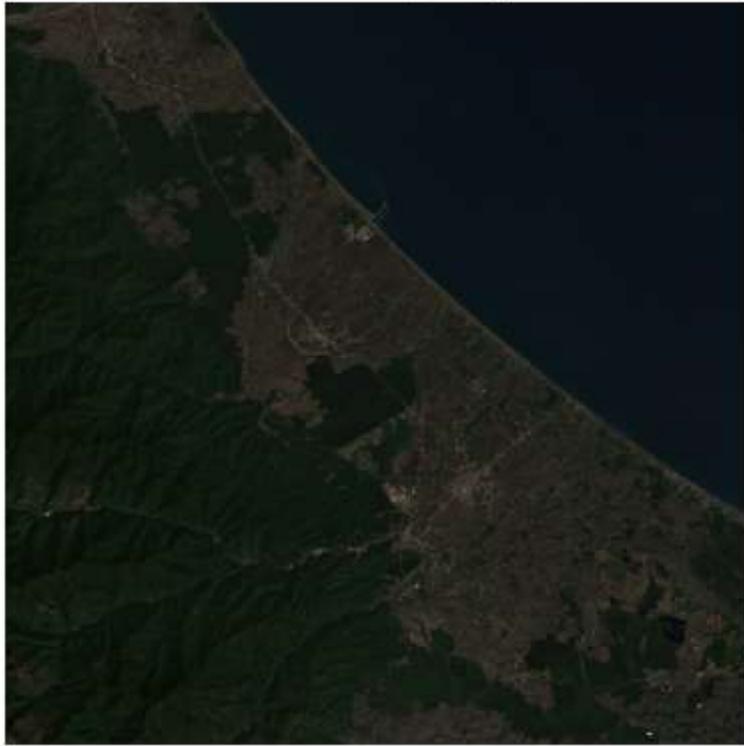
```
NB7 = (B7-np.min(B7))/(np.max(B7)-np.min(B7))
NB6 = (B6-np.min(B6))/(np.max(B6)-np.min(B6))
NB5 = (B5-np.min(B5))/(np.max(B5)-np.min(B5))
NB4 = (B4-np.min(B4))/(np.max(B4)-np.min(B4))
NB3 = (B3-np.min(B3))/(np.max(B3)-np.min(B3))
NB2 = (B2-np.min(B2))/(np.max(B2)-np.min(B2))
```

```
Nstack = np.stack([NB4,NB3,NB2], axis=2)
```

```
plt.imshow(Nstack)
plt.title('Landsat 8(RGB)')
plt.axis('off')
plt.show()
```



Landsat 8(RGB)



▼ Spectral indices

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm
from skimage import io
from skimage import exposure
import skimage as skm
from skimage import exposure, img_as_float
```

```

image1 = io.imread('Landsat8/Blue.tif')
image2 = io.imread('Landsat8/Green.tif')
image3 = io.imread('Landsat8/Red.tif')
image4 = io.imread('Landsat8//NIR.tif')
image5 = io.imread('Landsat8/SWIR1.tif')
image6 = io.imread('Landsat8/SWIR2.tif')
B1 = np.array(image1).astype(float)
B2 = np.array(image2).astype(float)
B3 = np.array(image3).astype(float)
B4 = np.array(image4).astype(float)
B5 = np.array(image5).astype(float)
B6 = np.array(image6).astype(float)
NB1 = (B1-np.min(B1))/(np.max(B1)-np.min(B1))
NB2 = (B2-np.min(B2))/(np.max(B2)-np.min(B2))
NB3 = (B3-np.min(B3))/(np.max(B3)-np.min(B3))
NB4 = (B4-np.min(B4))/(np.max(B4)-np.min(B4))
NB5 = (B5-np.min(B5))/(np.max(B5)-np.min(B5))
NB6 = (B6-np.min(B6))/(np.max(B6)-np.min(B6))

```

▼ Normalized difference vegetation index (NDVI)

The normalized difference vegetation index is a widely-used metric for quantifying the health and density of vegetation using sensor data. It is calculated from spectrometric data at two specific bands: red and near-infrared. The spectrometric data is usually sourced from remote sensors, such as satellites.

Equation:

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}$$

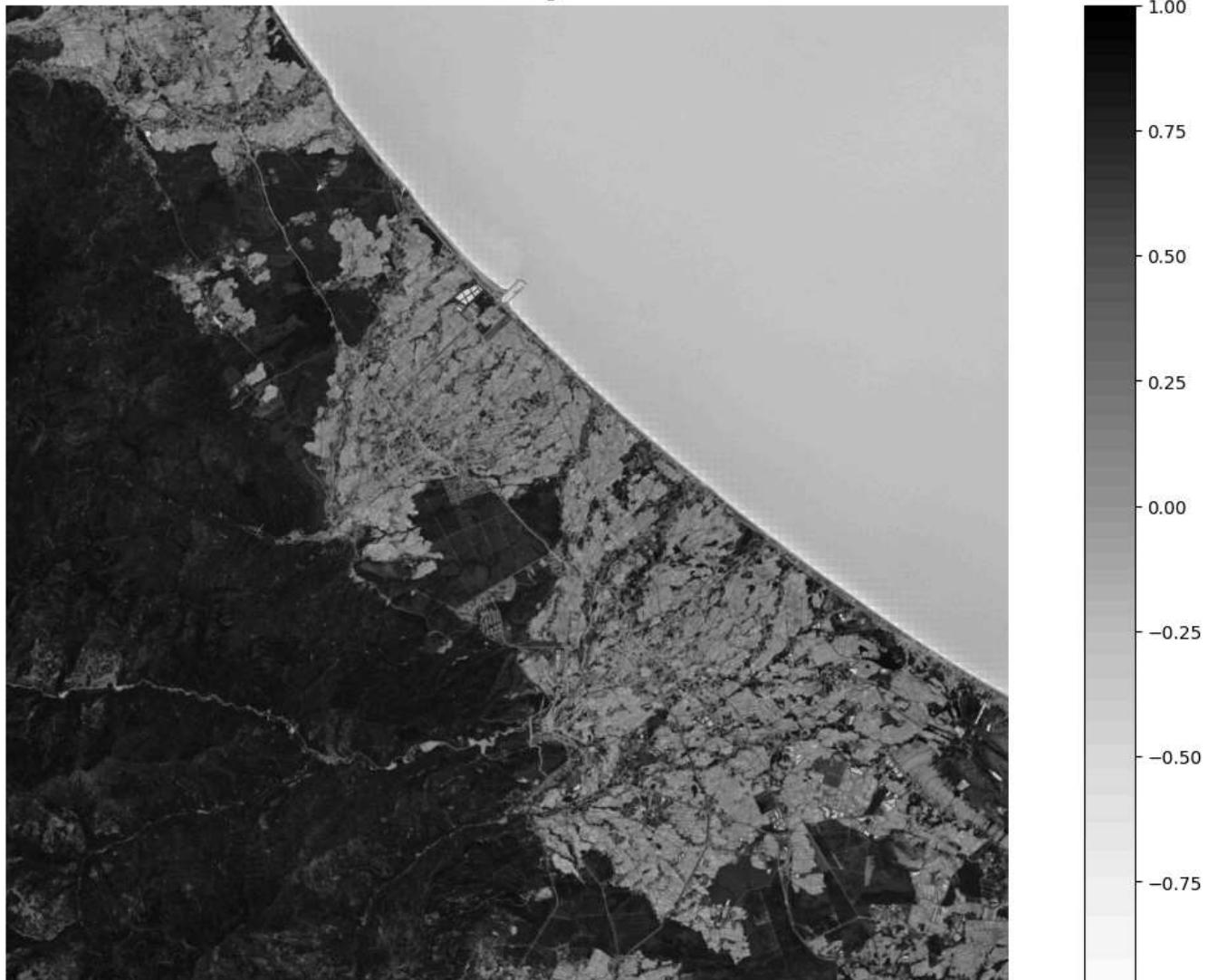
```

NDVI = (NB4-NB3)/(NB4+NB3)
plt.figure(figsize=(15,10))
plt.imshow(NDVI, cmap=cm.Greys)
plt.title('NDVI Image')
plt.colorbar()
plt.axis('off')
plt.show()

```



NDVI Image



▼ Normalized Difference Water Index (NDWI)

Normalized Difference Water Index (NDWI) may refer to one of at least two remote sensing-derived indexes related to liquid water: One is used to monitor changes in water content of leaves, using near-infrared (NIR) and short-wave infrared (SWIR) wavelengths, proposed by Gao in 1996:

$$\text{NDWI} = \frac{(X_{nir} - X_{swir})}{(X_{nir} + X_{swir})}$$

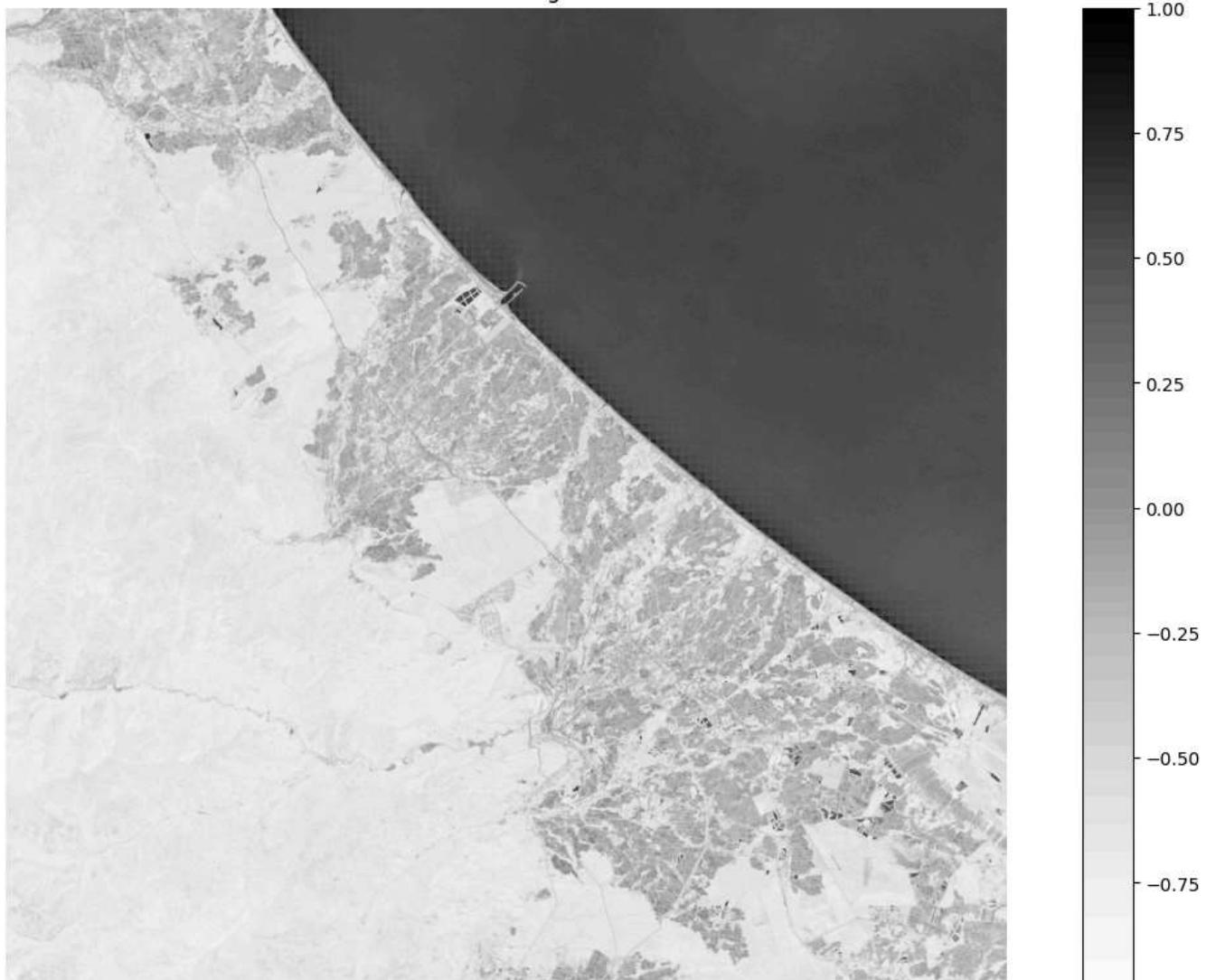
Another is used to monitor changes related to water content in water bodies, using green and NIR wavelengths, defined by McFeeters (1996):

$$\text{NDWI} = \frac{(X_{green} - X_{nir})}{(X_{green} + X_{nir})}$$

```
NDWI = (NB2 - NB4)/(NB2 + NB4)
plt.figure(figsize=(15,10))
plt.imshow(NDWI, cmap=cm.Greys)
plt.title('NDWI Image')
plt.colorbar()
plt.axis('off')
plt.show()
```



NDWI Image



❖ Soil Adjusted Vegetation Index (SAVI)

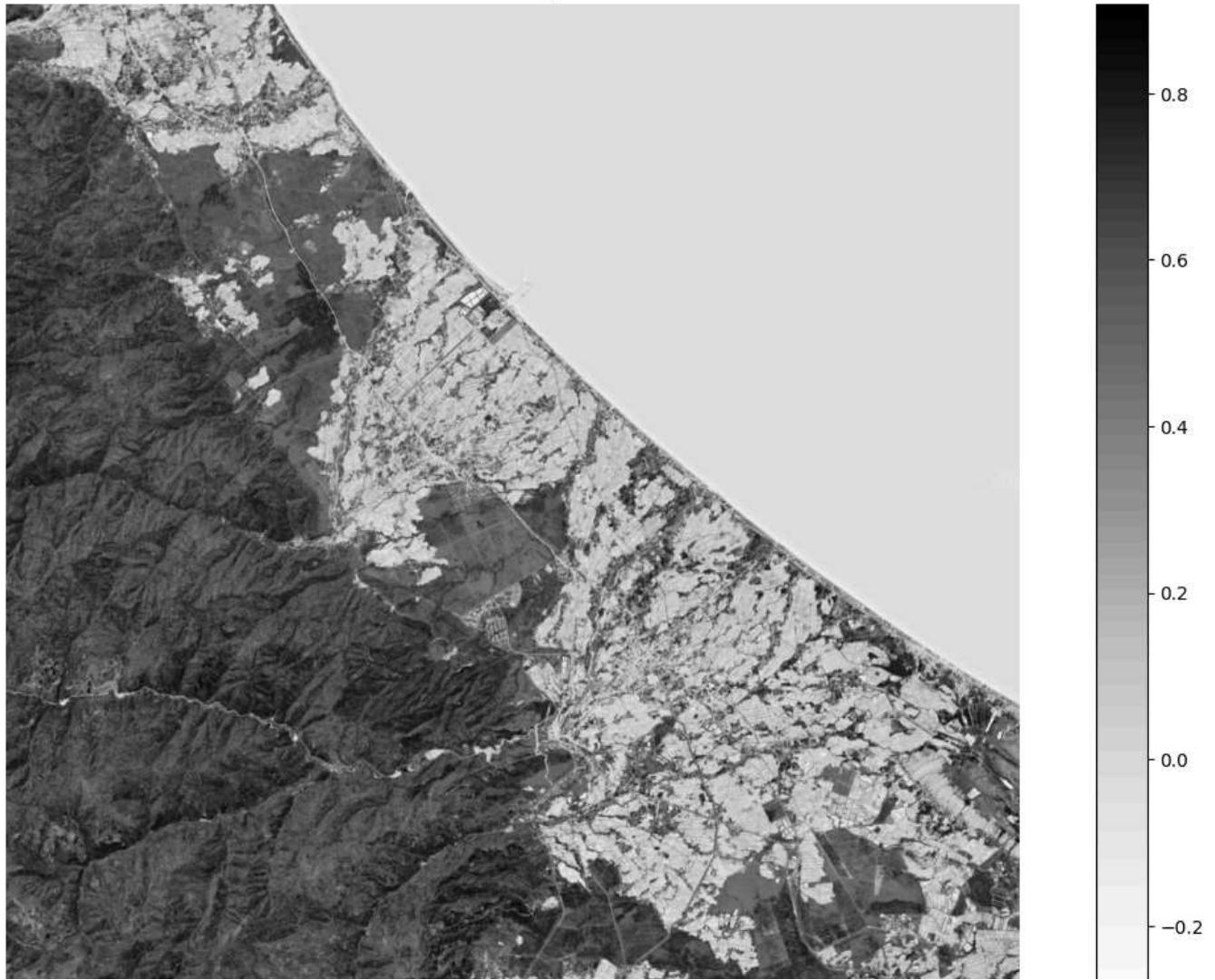
Soil Adjusted Vegetation Index (SAVI) is used to correct Normalized Difference Vegetation Index (NDVI) for the influence of soil brightness in areas where vegetative cover is low.

$$\text{SAVI} = \frac{(NIR - Red)}{(NIR + Red + L)} \times (1 + L)$$

```
SAVI = ((NB4-NB3)/(NB4+NB3+0.5))*(1+0.5)
plt.figure(figsize=(15,10))
plt.imshow(SAVI, cmap=cm.Greys)
plt.title('SAVI Image')
plt.colorbar()
plt.axis('off')
plt.show()
```



SAVI Image



▼ Urban Index (UI)

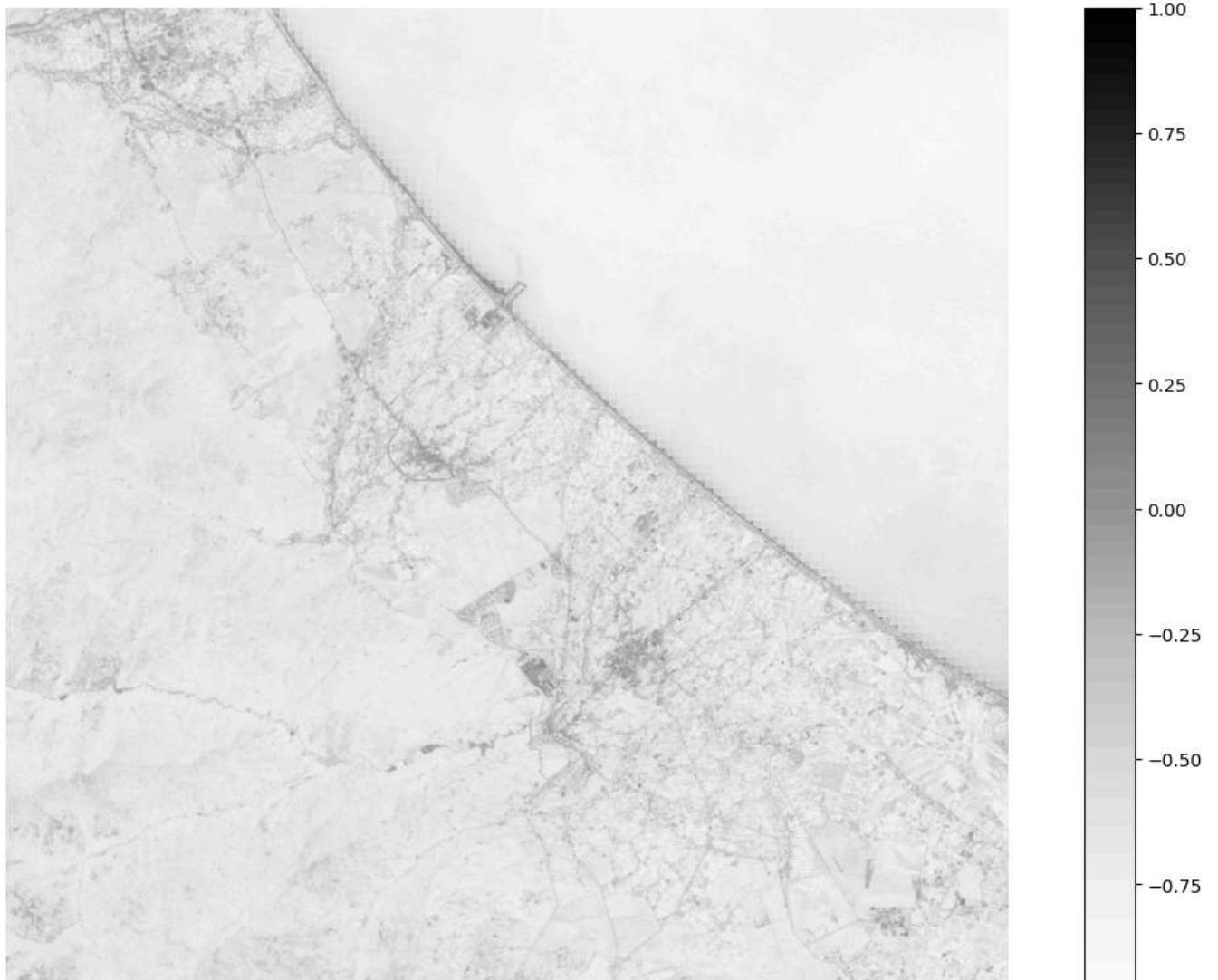
Urban remote sensing indices are generally used to distinguish built-up areas based on spectral values with a low level of accuracy because of high degree of homogeneity with the bare land and wetland areas.

$$\text{NDBI} = \frac{(SWIR - NIR)}{(SWIR + NIR)}$$

```
UI = (NB6 - NB4)/(NB6 + NB4)
plt.figure(figsize=(15,10))
plt.imshow(UI, cmap=cm.Greys)
plt.title('Urban Index')
plt.colorbar()
plt.axis('off')
plt.show()
```



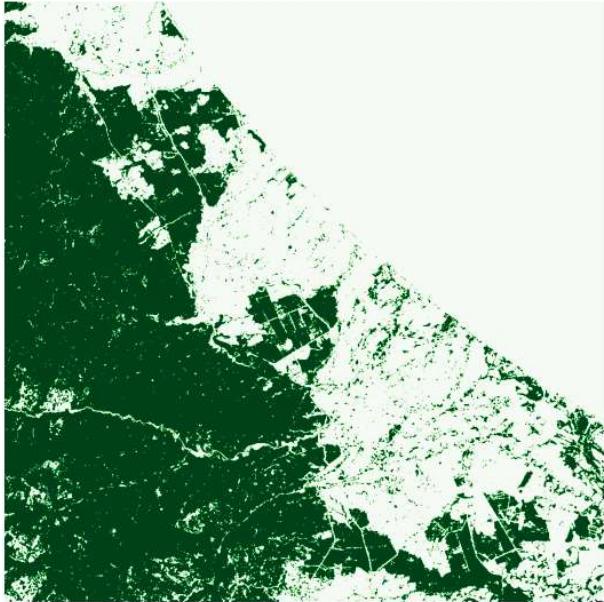
Urban Index



```
lai = 0.68
veg = NDVI > lai
non_veg = NDVI <= lai
water = 0.2
waterno = NDWI > water
non_waterno = NDWI <= water
soilveg = 0.44
soil = SAVI > soilveg
non_soil = SAVI <= soilveg
urbaniz = -0.28
urban = UI > urbaniz
non_urban = UI <= urbaniz
fig , ax = plt.subplots(nrows=2, ncols=2, figsize=(15,15))
ax[0,0].imshow(veg, cmap=cm.Greens)
ax[0,0].set_title('NDVI')
ax[0,0].axis('off')
#####
ax[0,1].imshow(waterno, cmap=cm.Blues)
ax[0,1].set_title('NDWI')
ax[0,1].axis('off')
#####
ax[1,0].imshow(urban, cmap=cm.Greys)
ax[1,0].set_title('UI')
ax[1,0].axis('off')
#####
ax[1,1].imshow(soil, cmap=cm.Oranges)
ax[1,1].set_title('SAVI')
ax[1,1].axis('off')
plt.show()
```



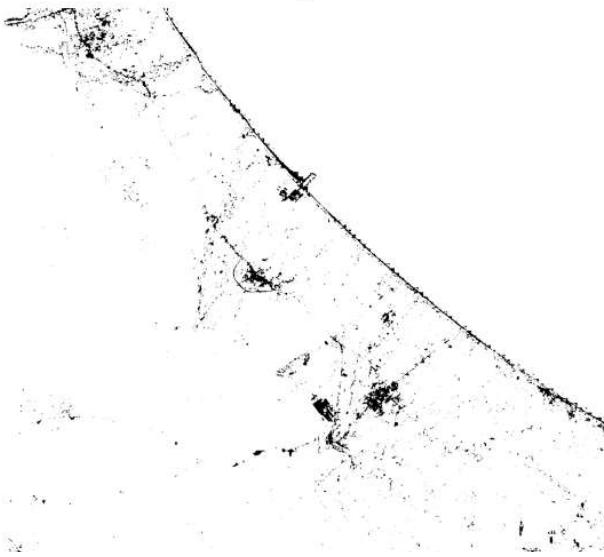
NDVI



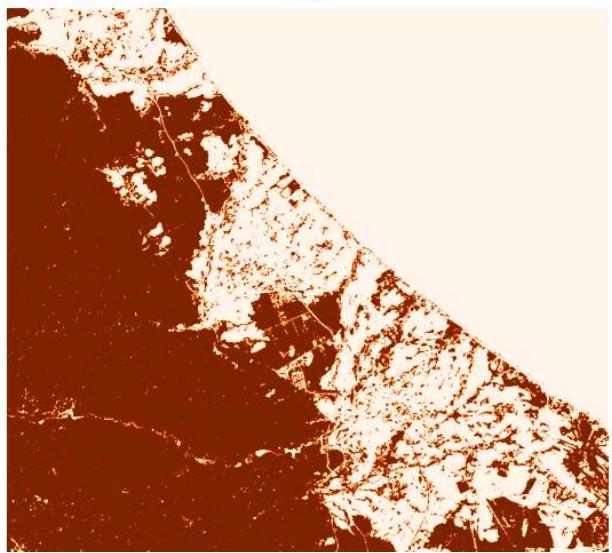
NDWI



UI



SAVI



▼ Image Enhancement

Image enhancement is a process in digital image processing aimed at improving the visual appearance of an image or to convert the image to a form better suited for analysis by humans or machines. This involves amplifying certain features of an image to make it more appealing or easier to interpret.

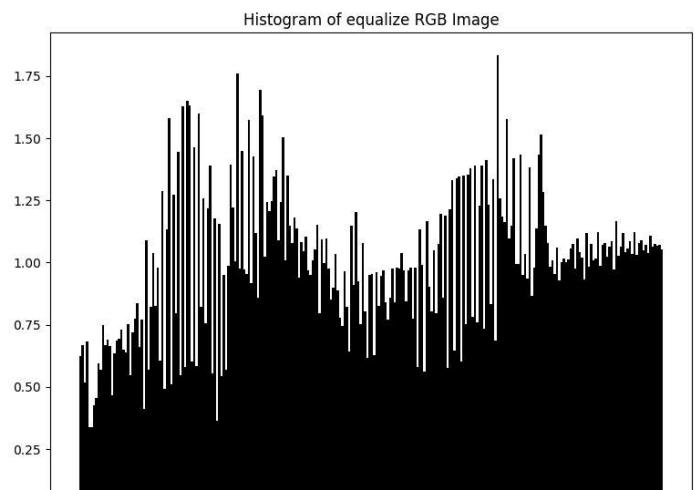
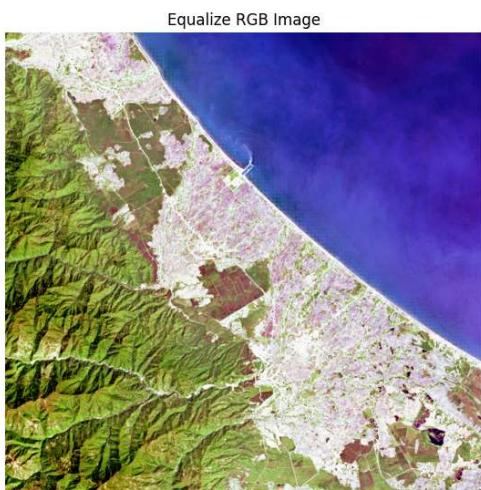
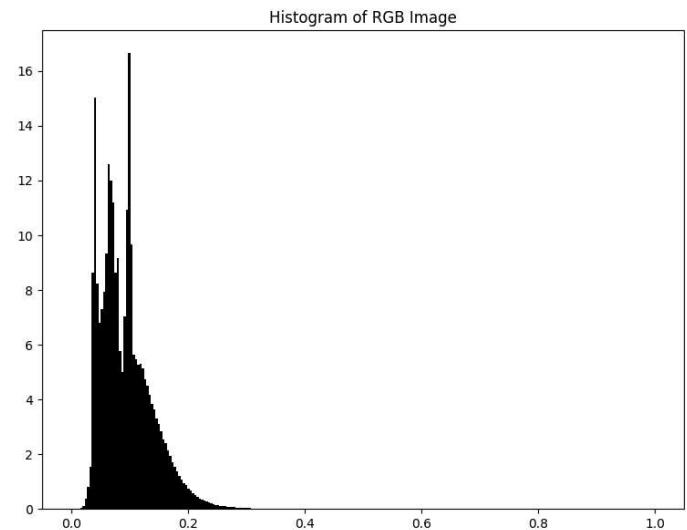
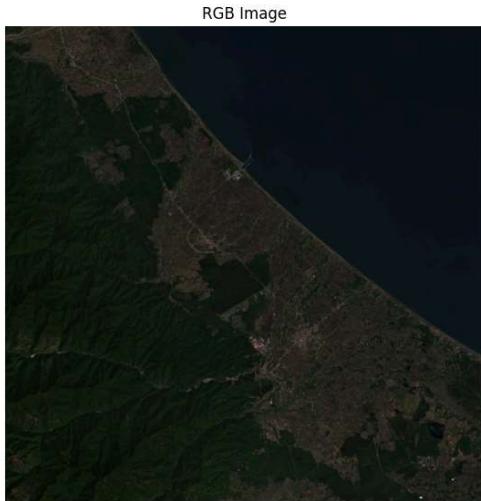
```
from skimage import exposure, img_as_float
```

❖ Histogram Equalization

It spreads out the most frequent intensity values in an image, effectively “stretching” the histogram of pixel values. This often results in an image with enhanced contrast and can be particularly useful for images with backgrounds and foregrounds that are both bright or both dark.

```
equalizeEnNB2 = exposure.equalize_hist(NB2, nbins=256, mask=None)
equalizeEnNB3 = exposure.equalize_hist(NB3, nbins=256, mask=None)
equalizeEnNB4 = exposure.equalize_hist(NB4, nbins=256, mask=None)
equalizeNstack = np.stack([equalizeEnNB4,equalizeEnNB3,equalizeEnNB2], axis=2)

fig , ax = plt.subplots(nrows=2, ncols=2, figsize=(20,15))
ax[0,0].imshow(Nstack)
ax[0,0].set_title('RGB Image')
ax[0,0].axis('off')
ax[0,1].hist(Nstack.ravel(), bins=256, density=True, histtype='bar', color='black')
ax[0,1].ticklabel_format(style='plain')
ax[0,1].set_title('Histogram of RGB Image')
#####
ax[1,0].imshow(equalizeNstack, cmap='gray')
ax[1,0].set_title('Equalize RGB Image')
ax[1,0].axis('off')
ax[1,1].hist(equalizeNstack.ravel(), bins=256, density=True, histtype='bar', color='black')
ax[1,1].ticklabel_format(style='plain')
ax[1,1].set_title('Histogram of equalize RGB Image')
plt.show()
```



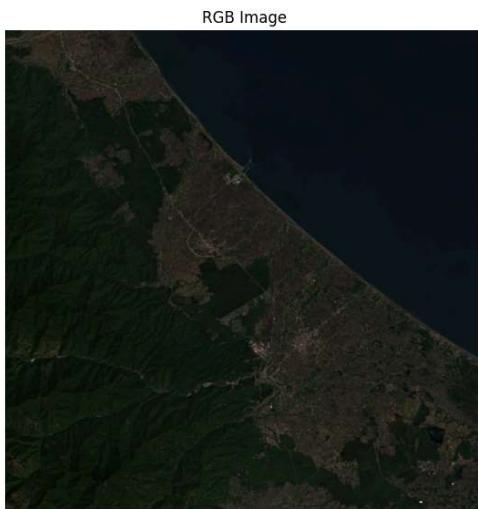
▼ Gamma Correction

Gamma Correction in image processing, which is a nonlinear operation that can be used to encode and decode luminance or tristimulus values in images

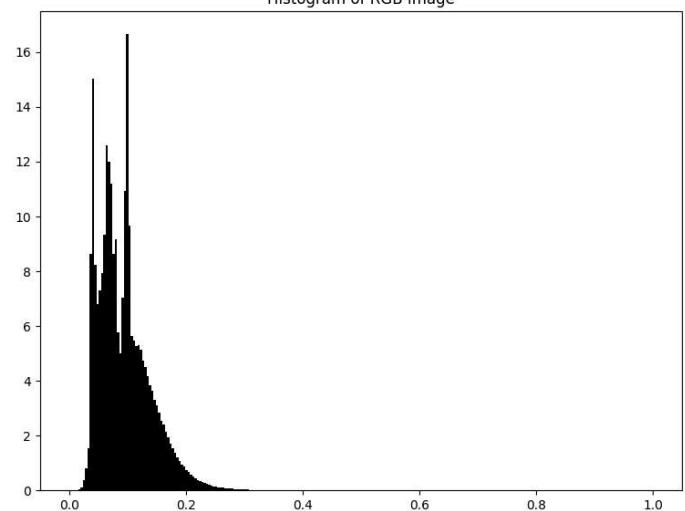
```
GEnNB2 = exposure.adjust_gamma(NB2 , gamma=0.5, gain=1)
GEnNB3 = exposure.adjust_gamma(NB3 , gamma=0.5, gain=1)
GEnNB4 = exposure.adjust_gamma(NB4 , gamma=0.5, gain=1)
GNstack = np.stack([GEnNB4, GEnNB3, GEnNB2], axis=2)
```

```
fig , ax = plt.subplots(nrows=2, ncols=2, figsize=(20,15))
ax[0,0].imshow(Nstack)
ax[0,0].set_title('RGB Image')
ax[0,0].axis('off')
ax[0,1].hist(Nstack.ravel(), bins=256, density=True, histtype='bar', color='black')
ax[0,1].ticklabel_format(style='plain')
ax[0,1].set_title('Histogram of RGB Image')
#####
ax[1,0].imshow(GNstack, cmap='gray')
ax[1,0].set_title('Gamma RGB Image')
ax[1,0].axis('off')
ax[1,1].hist(GNstack.ravel(), bins=256, density=True, histtype='bar', color='black')
ax[1,1].ticklabel_format(style='plain')
ax[1,1].set_title('Histogram of Gamma RGB Image')
```

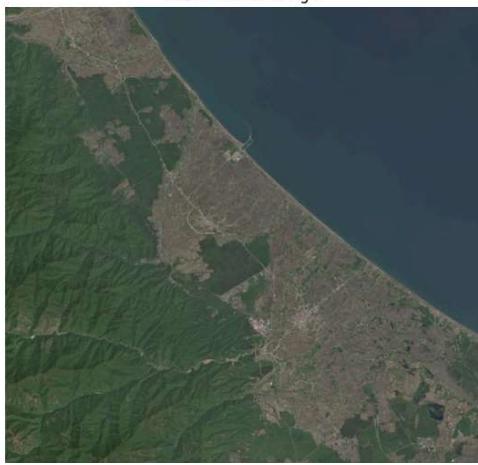
→ Text(0.5, 1.0, 'Histogram of Gamma RGB Image')



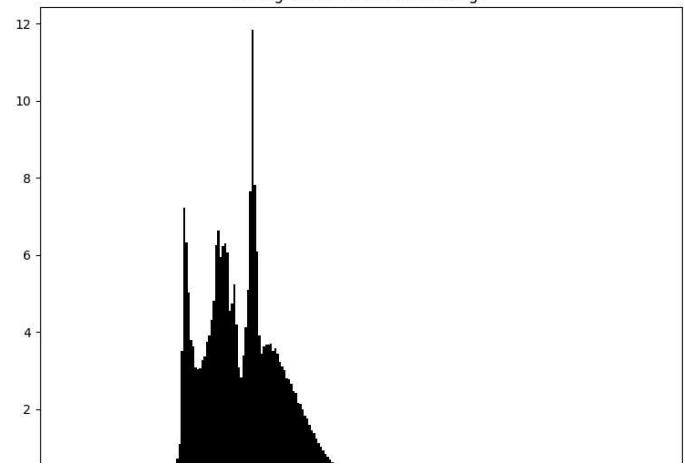
Histogram of RGB Image



RGB Image



Histogram of Gamma RGB Image



❖ k-means clustering

K-means is an unsupervised learning algorithm used in machine learning for clustering analysis. In unsupervised learning, algorithms are used to group unlabelled datasets, and k-means is particularly effective for partitioning a dataset into distinct clusters based on similarity. Here's a high-level overview of how the k-means algorithm works:

Initialization:

Choose the number of clusters, (K), and select (K) random points as cluster centers or centroids.

Assignment:

Assign each data point to the nearest centroid, forming (K) clusters.

Update:

Calculate the new centroids (mean) of the clusters by taking the average of all data points in the cluster.

Iteration:

Repeat the assignment and update steps until the centroids no longer change significantly, indicating that the clusters are stable.

The goal of k-means is to minimize the within-cluster variance, which is the sum of the squared distances between each data point and its corresponding centroid. This process iteratively refines the positions of the centroids, resulting in a set of clusters where each data point is closer to its own cluster's centroid than to any other.

K-means is widely used because it's simple, efficient, and can handle large datasets. However, it does have some limitations, such as sensitivity to the initial placement of centroids and difficulty in clustering data with varying shapes and densities. It's also important to predefine the value of (K), which may not be straightforward and often requires additional methods like the elbow method to estimate the optimal number of clusters.

```
from sklearn.cluster import KMeans  
from skimage import io
```

```
image = io.imread('NIR.tif')  
image.shape
```

→ (998, 998)

```
row = image.shape[0]
col = image.shape[1]
X = image.reshape((row*col, 1))
```

```
X.shape
```

```
→ (996004, 1)
```

```
X
```

```
→ array([[3311],
       [3527],
       [3537],
       ...,
       [2986],
       [2712],
       [2843]], dtype=int16)
```

```
kmeans = KMeans(n_clusters=4 , random_state=0)
kmeans.fit(X)
```

```
→ KMeans
KMeans(n_clusters=4, random_state=0)
```

```
cluster_centers = kmeans.cluster_centers_
cluster_labels = kmeans.labels_
```

```
cluster_centers
```

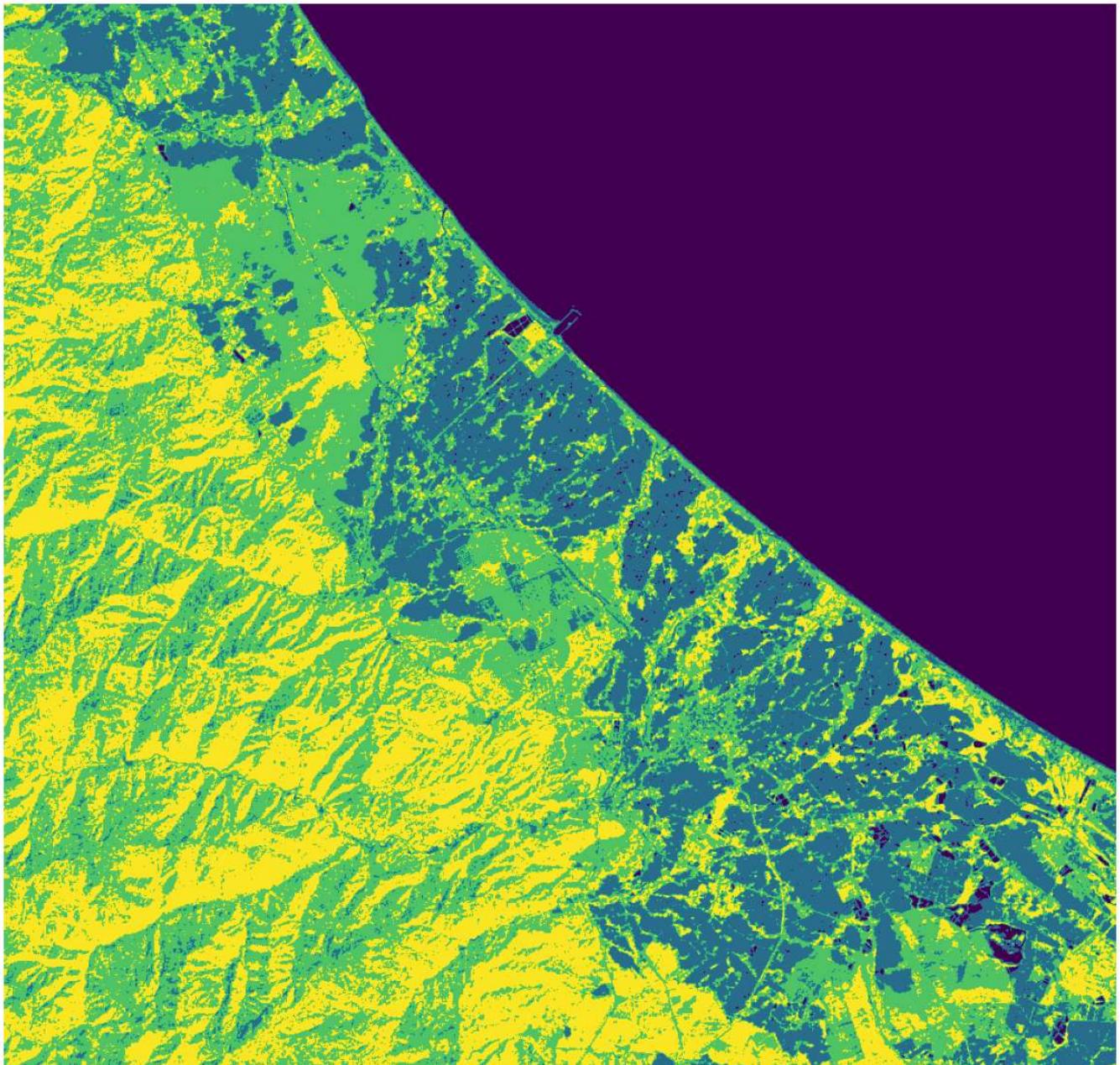
```
→ array([[1499.75884581],
       [2751.75957124],
       [3721.88881973],
       [173.49399968]])
```

```
cluster_labels
```

```
→ array([2, 2, 2, ..., 1, 1, 1])
```

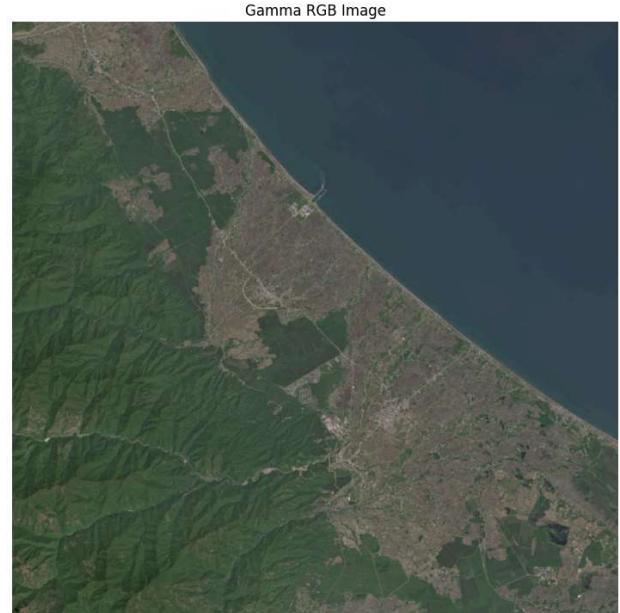
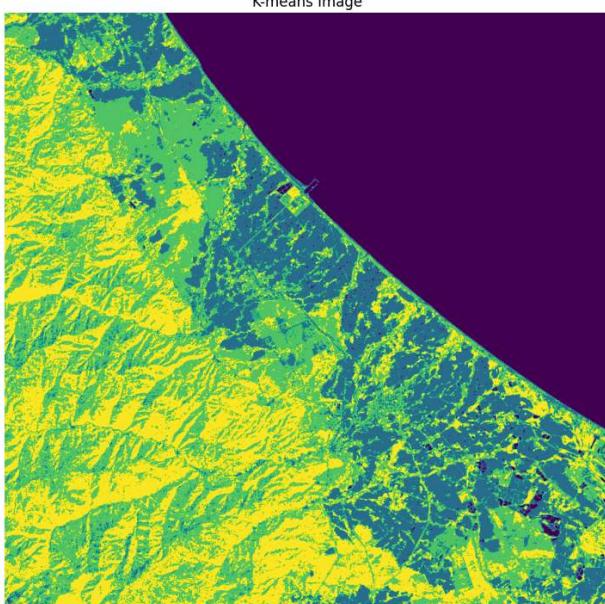
```
clusterd_image= cluster_centers[cluster_labels].reshape(row, col)
```

```
plt.figure(figsize=(15,15))
plt.imshow(clusterd_image)
plt.axis('off')
plt.show()
```



```
fig, ax = plt.subplots(nrows=1, ncols=2, figsize=(20,15))
ax[0].imshow(clusterd_image)
ax[0].set_title('K-means Image')
ax[0].axis('off')
ax[1].imshow(GNstack, cmap='gray')
ax[1].set_title('Gamma RGB Image')
ax[1].axis('off')
```

→ (-0.5, 997.5, 997.5, -0.5)



❖ Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful and versatile supervised machine learning algorithm used for both classification and regression tasks, though it is primarily known for classification. Here's an overview of SVM:

Objective: The main goal of SVM is to find the optimal hyperplane that separates different classes in the feature space with the maximum margin possible¹.

Core Idea:

SVM classifies data by finding the optimal hyperplane which maximizes the margin between different classes in an N-dimensional space.

Hyperplane:

This is the decision boundary that separates different classes. The dimension of the hyperplane depends on the number of input features.

Support Vectors:

These are the data points nearest to the hyperplane, which influence its position. SVM aims to maximize the distance (margin) from these points to the hyperplane

Margin:

It's the distance between the hyperplane and the nearest data point from either class. A larger margin is associated with a lower generalization error of the classifier

Kernels:

SVM can handle linear and non-linear data. For non-linear data, it uses kernel functions to transform the input space into a higher-dimensional space where a linear separator can be used.

Robustness to Outliers:

SVM is known for its robustness to outliers. It focuses on the support vectors and the margin, which helps in ignoring the outliers while finding the hyperplane1.

Applications:

SVM is used in various domains like text classification, image recognition, spam detection, and more due to its effectiveness in high-dimensional spaces and its ability to handle nonlinear relationships.

SVM's ability to find the maximum separating hyperplane by considering the closest points of different classes makes it a strong model, especially when dealing with complex datasets.

```
import skimage as skm
from skimage import io
import matplotlib.pyplot as plt
import cv2
import numpy as np
from sklearn import svm
import pandas as pd
import itertools

collection = io.imread_collection('Landsat8/*.tif')

BB = collection[0]
BG = collection[1]
BR = collection[3]

NBR = (BR - np.min(BR))/(np.max(BR) - np.min(BR))
NBG = (BG - np.min(BG))/(np.max(BG) - np.min(BG))
NBB = (BB - np.min(BB))/(np.max(BB) - np.min(BB))

rgb = np.stack([NBR, NBG, NBB], axis=2)
```

```
width = collection[0].shape[0]
height = collection[0].shape[1]
```

width

```
→ 998
```

height

```
→ 998
```

```
all_images = io.concatenate_images(collection).transpose()
all_image_reshape = all_images.reshape((height*width, len(collection)))
```

all_image_reshape

```
→ array([[ -64,   386,  3311,   163,  1695,   844],
       [ -57,   329,  2843,   191,  1517,   794],
       [ -40,   429,  2769,   236,  1772,   955],
       ...,
       [ 456,   701,  1649,   796,   921,   531],
       [ 311,   635,  2440,   631,  1350,   779],
       [  70,   452,  2843,   342,  1276,   668]], dtype=int16)
```

```
class_num = 4
sample_num = 10
```

```
columns = ['Band{}'.format(i+1) for i in range(len(collection))]
classes_df = pd.DataFrame(columns=columns)
```

columns

```
→ ['Band1', 'Band2', 'Band3', 'Band4', 'Band5', 'Band6']
```

classes_df

```
→ Band1 Band2 Band3 Band4 Band5 Band6
```

```
targets = [[i+1]*sample_num for i in range(class_num)]
merged = list(itertools.chain(*targets))
classes_df['Target'] = merged
```

targets

```
→ [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1],  
 [2, 2, 2, 2, 2, 2, 2, 2, 2, 2],  
 [3, 3, 3, 3, 3, 3, 3, 3, 3, 3],  
 [4, 4, 4, 4, 4, 4, 4, 4, 4, 4]]
```

merged

```
→ [1,  
 1,  
 1,  
 1,  
 1,  
 1,  
 1,  
 1,  
 1,  
 1,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 2,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 3,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4,  
 4]
```

classes_df

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	NaN	NaN	NaN	NaN	NaN	NaN	1
1	NaN	NaN	NaN	NaN	NaN	NaN	1
2	NaN	NaN	NaN	NaN	NaN	NaN	1
3	NaN	NaN	NaN	NaN	NaN	NaN	1
4	NaN	NaN	NaN	NaN	NaN	NaN	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3

30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

```
i = 0
```

```
def mouseclick(event , x , y , flags, param):
    global i
    if event == cv2.EVENT_LBUTTONDOWN:
        if i < class_num * sample_num:
            mylist = []
            for j in collection:
                mylist.append(j[x][y])
            classes_df.iloc[i , 0:len(collection)] = mylist
            i = i+1
            print(classes_df)
    else:
        array = classes_df.values
        X = array[:, :len(collection)]
        Y = array[:, len(collection)].astype('int')

        clf = svm.SVC(gamma='scale')#Start Classifier
        clf.fit(X,Y)
        pred = clf.predict(all_image_reshape)
        plt.imshow(pred.reshape((height, width)).transpose())
        plt.show()
```

```
cv2.namedWindow('mouseClick', cv2.WINDOW_NORMAL)
cv2.setMouseCallback('mouseClick', mouseclick)

while True:
    cv2.imshow('mouseClick', rgb)
    if cv2.waitKey(20) == 27:
        break

cv2.destroyAllWindows()
```

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	NaN	NaN	NaN	NaN	NaN	NaN	1
2	NaN	NaN	NaN	NaN	NaN	NaN	1
3	NaN	NaN	NaN	NaN	NaN	NaN	1
4	NaN	NaN	NaN	NaN	NaN	NaN	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	NaN	NaN	NaN	NaN	NaN	NaN	1
3	NaN	NaN	NaN	NaN	NaN	NaN	1
4	NaN	NaN	NaN	NaN	NaN	NaN	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2

14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
--	-------	-------	-------	-------	-------	-------	--------

0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	NaN	NaN	NaN	NaN	NaN	NaN	1
4	NaN	NaN	NaN	NaN	NaN	NaN	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3

30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	NaN	NaN	NaN	NaN	NaN	NaN	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1

4	185	678	3373	597	2054	1184	1
5	NaN	NaN	NaN	NaN	NaN	NaN	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	NaN	NaN	NaN	NaN	NaN	NaN	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2

20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	NaN	NaN	NaN	NaN	NaN	NaN	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4

35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	NaN	NaN	NaN	NaN	NaN	NaN	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	NaN	NaN	NaN	NaN	NaN	NaN	1

10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	NaN	NaN	NaN	NaN	NaN	NaN	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3

25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	NaN	NaN	NaN	NaN	NaN	NaN	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

Band1	Band2	Band3	Band4	Band5	Band6	Target
-------	-------	-------	-------	-------	-------	--------

0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	NaN	NaN	NaN	NaN	NaN	NaN	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	NaN	NaN	NaN	NaN	NaN	NaN	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2

15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	NaN	NaN	NaN	NaN	NaN	NaN	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4

31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	NaN	NaN	NaN	NaN	NaN	NaN	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1

5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	NaN	NaN	NaN	NaN	NaN	NaN	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	NaN	NaN	NaN	NaN	NaN	NaN	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3

21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	NaN	NaN	NaN	NaN	NaN	NaN	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4

36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	NaN	NaN	NaN	NaN	NaN	NaN	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2

-	--	--	--	-	-	--	
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	NaN	NaN	NaN	NaN	NaN	NaN	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	NaN	NaN	NaN	NaN	NaN	NaN	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3

26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	NaN	NaN	NaN	NaN	NaN	NaN	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
--	-------	-------	-------	-------	-------	-------	--------

0	-12	534	3690	242	1804	874	1
---	-----	-----	------	-----	------	-----	---

-	-	-	-	-	-	-	-
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	NaN	NaN	NaN	NaN	NaN	NaN	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
-	-	-	-	-	-	-	-

16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	NaN	NaN	NaN	NaN	NaN	NaN	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	NaN	NaN	NaN	NaN	NaN	NaN	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4

32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	NaN	NaN	NaN	NaN	NaN	NaN	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
-	---	---	---	---	---	---	-

6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	NaN	NaN	NaN	NaN	NaN	NaN	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3

22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	NaN	NaN	NaN	NaN	NaN	NaN	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	NaN	NaN	NaN	NaN	NaN	NaN	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4

37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	NaN	NaN	NaN	NaN	NaN	NaN	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2

12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	NaN	NaN	NaN	NaN	NaN	NaN	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3

27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	NaN	NaN	NaN	NaN	NaN	NaN	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	NaN	NaN	NaN	NaN	NaN	NaN	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1

2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	NaN	NaN	NaN	NaN	NaN	NaN	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4
	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2

17	86	409	2455	342	1455	113	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	NaN	NaN	NaN	NaN	NaN	NaN	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4

33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	352	662	1505	782	745	476	4
36	NaN	NaN	NaN	NaN	NaN	NaN	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	352	662	1505	782	745	476	4
36	285	715	2362	710	1736	1112	4
37	NaN	NaN	NaN	NaN	NaN	NaN	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	26	572	4288	215	1781	838	1

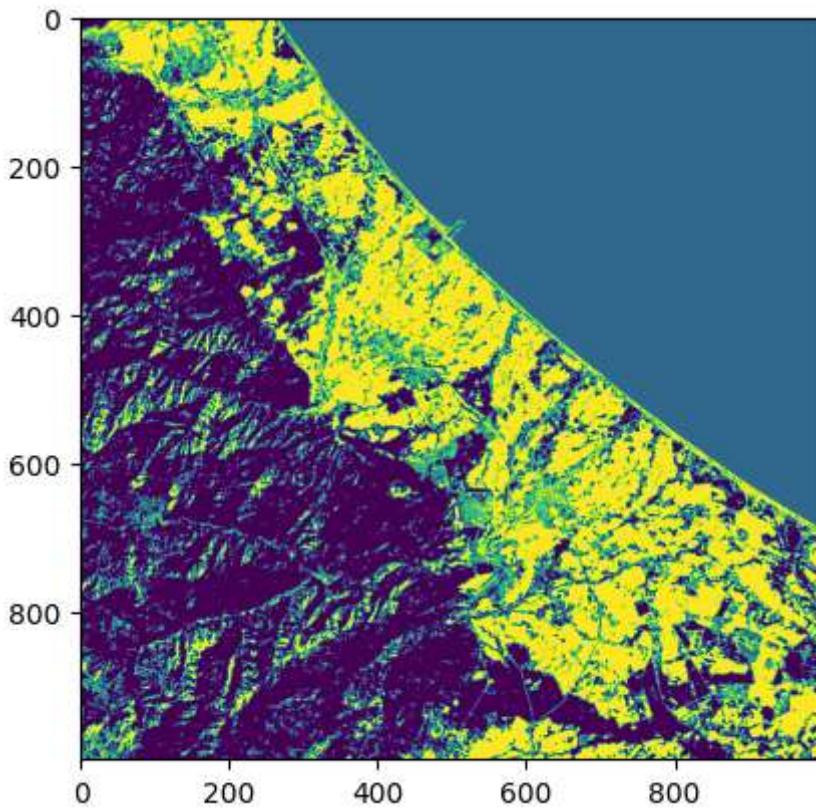
/	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	352	662	1505	782	745	476	4
36	285	715	2362	710	1736	1112	4
37	0	254	2065	231	1064	544	4
38	NaN	NaN	NaN	NaN	NaN	NaN	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3

23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	352	662	1505	782	745	476	4
36	285	715	2362	710	1736	1112	4
37	0	254	2065	231	1064	544	4
38	33	326	2269	264	1169	587	4
39	NaN	NaN	NaN	NaN	NaN	NaN	4

	Band1	Band2	Band3	Band4	Band5	Band6	Target
0	-12	534	3690	242	1804	874	1
1	-189	179	2946	66	1342	635	1
2	19	411	3537	250	1788	858	1
3	-120	306	2625	134	1358	672	1
4	185	678	3373	597	2054	1184	1
5	-25	521	3493	244	1801	899	1
6	-156	253	2669	107	1312	646	1
7	-36	532	4288	215	1781	838	1
8	-57	290	3341	144	1577	751	1
9	-22	529	4968	224	2054	932	1
10	369	663	2400	825	1910	1332	2
11	191	325	204	139	201	156	2
12	237	353	197	138	192	152	2
13	31	384	2564	234	1470	726	2
14	230	241	158	138	152	130	2
15	197	730	3581	518	2125	1253	2
16	22	442	3575	209	1735	809	2
17	86	409	2455	342	1455	773	2
18	235	560	1544	647	797	486	2
19	211	275	211	155	188	156	2
20	-22	449	2891	244	1629	837	3
21	150	532	3480	389	1986	1220	3
22	206	711	3876	540	2151	1221	3
23	108	488	3032	446	1917	1059	3
24	257	298	238	202	200	165	3
25	243	296	239	190	210	172	3
26	21	478	3827	231	1676	796	3
27	623	1037	2452	1087	2136	1615	3
28	24	372	2505	212	1558	790	3
29	209	216	142	110	142	121	3
30	226	543	1178	718	625	427	4
31	733	1053	1961	1250	2404	2339	4
32	90	469	2595	386	1486	866	4
33	296	676	1370	827	1150	846	4
34	228	501	1271	615	607	393	4
35	352	662	1505	782	745	476	4
36	285	715	2362	710	1736	1112	4
37	0	254	2065	231	1064	544	4
38	33	326	2269	264	1169	587	4

38	33	326	2269	264	1169	587	4
39	49	316	2386	297	1223	680	4



```
KeyboardInterrupt                                     Traceback (most recent call last)
Cell In[21], line 6
      4 while True:
      5     cv2.imshow('mouseClick', rgb)
----> 6     if cv2.waitKey(20) == 27:
      7         break
      9 cv2.destroyAllWindows()

KeyboardInterrupt:
```

▼ Last but not least: Introduction to N-dimensional data

When it comes to analyzing and interpreting the vast and complex world of climate data, Xarray stands out as a powerful tool tailored for handling N-dimensional datasets. This Python library is designed to make working with multi-dimensional arrays both simple and intuitive, particularly for those in the field of climate science. When it comes to analyzing and interpreting the vast and complex world of climate data, Xarray stands out as a powerful tool tailored for handling N-dimensional datasets. This Python library is designed to make working with multi-dimensional arrays both simple and intuitive, particularly for those in the field of climate science. Xarray

introduces labels in the form of dimensions, coordinates, and attributes on top of raw NumPy-like arrays, which is a game-changer for climate data analysis. It allows you to go beyond the traditional row and column format, enabling you to work with data more naturally and with less room for error.

Key Features:

- **Alignment and indexing:** Xarray's data structures align themselves along dimensions without having to keep track of the order of those dimensions, simplifying data manipulation and analysis.
- **Flexible data manipulation:** It integrates seamlessly with Pandas to offer a wide range of operations for data manipulation, from simple data aggregation to complex multi-step workflows.
- **I/O and computation:** Xarray supports direct serialization and I/O operations for a wide range of data formats, which is essential for dealing with large and diverse climate datasets.

Practical Applications:

- **Time-series analysis:** With Xarray, you can easily handle time-series data, perform resampling operations, and compute rolling statistics, which are common tasks in climate data analysis.
- **Spatial analysis:** Xarray facilitates working with spatial data, allowing you to slice and dice datasets based on geographical coordinates, and perform computations like zonal statistics.
- **Ensemble analysis:** Climate models often produce ensemble datasets. Xarray simplifies the process of combining and comparing these datasets to assess uncertainties and make predictions.

▼ Send your first API request

from <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-land-monthly-means?tab=overview> select your data

```
import cdsapi

c = cdsapi.Client()

c.retrieve(
    'reanalysis-era5-land-monthly-means',
    {
        'product_type': 'monthly_averaged_reanalysis',
        'variable': [
            'skin_temperature', 'snow_cover', 'snow_depth',
            'snow_evaporation', 'snowfall', 'snowmelt',
            'total_evaporation', 'total_precipitation',
        ],
        'year': [
            '2000', '2001', '2002',
            '2003', '2004', '2005',
            '2006', '2007', '2008',
            '2009', '2010', '2011',
            '2012', '2013', '2014',
            '2015', '2016', '2017',
            '2018', '2019', '2020',
            '2021',
        ],
        'month': [
            '01', '02', '03',
            '04', '05', '06',
            '07', '08', '09',
            '10', '11', '12',
        ],
        'time': '00:00',
        'area': [
            41.29, 44.69, 22.51,
            62.13,
        ],
        'format': 'netcdf.zip',
    },
    'download.netcdf.zip')
```

```
→ 2024-06-04 11:28:36,275 INFO Sending request to https://cds.climate.copernicus.eu/api/v2
2024-06-04 11:28:57,311 WARNING Connection error: [HTTPSConnectionPool(host='cds.climate.copernicus.eu'): Max retries exceeded with url: /api/v2/reanalysis-era5-land-monthly-means?product_type=monthly_averaged_reanalysis&variable=snow_depth&year=2000&month=01&time=00:00&area=[41.29, 44.69, 22.51, 62.13]&format=netcdf.zip (Caused by SSLError("bad handshake: unknown ca certificate"))]
2024-06-04 11:28:57,312 WARNING Retrying in 120 seconds
2024-06-04 11:30:57,312 INFO Retrying now...
2024-06-04 11:30:58,327 INFO Request is completed
2024-06-04 11:30:58,328 INFO Downloading https://download-0012-clone.copernicus-climate.eu/cache-compute-0012/cache/data1/adaptor.mars.internal-1717465898.6368284-7641-15-b814f915-320b-4a3c-9b38-749a66dea248.zip
```

After downloading data you can start your work

▼ Import

```
import xarray as xr  
ds = xr.open_dataset('data.nc')  
ds
```

→ xarray.Dataset

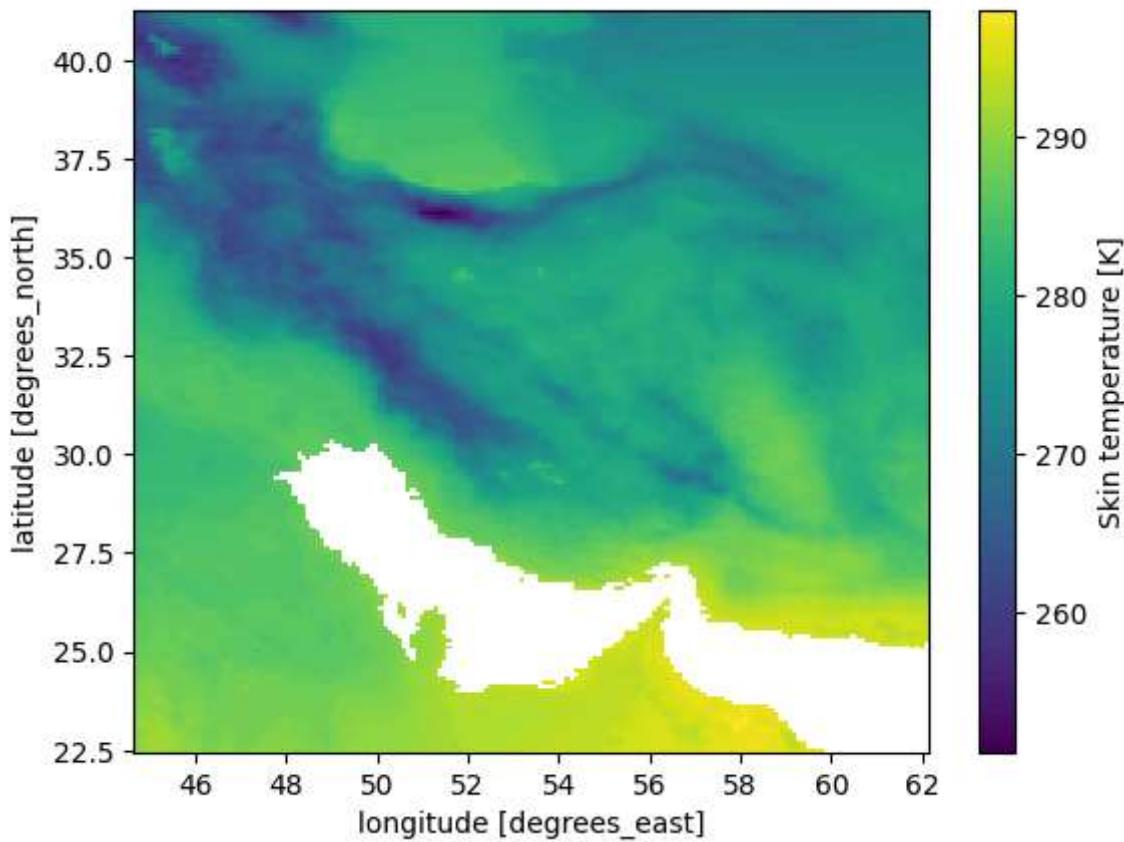
► Dimensions:	(longitude: 175, latitude: 188, time: 264)		
▼ Coordinates:			
longitude	(longitude)	float32	44.69 44.79 44.89 ... 61.99 62.09
latitude	(latitude)	float32	41.21 41.11 41.01 ... 22.61 22.51
time	(time)	datetime64[ns]	2000-01-01 ... 2021-12-01
▼ Data variables:			
skt	(time, latitude, longitude)	float64	...
snowc	(time, latitude, longitude)	float64	...
sde	(time, latitude, longitude)	float64	...
es	(time, latitude, longitude)	float64	...
sf	(time, latitude, longitude)	float64	...
smlt	(time, latitude, longitude)	float64	...
e	(time, latitude, longitude)	float64	...
tp	(time, latitude, longitude)	float64	...
► Indexes: (3)			
▼ Attributes:			
Conventions :	CF-1.6		
history :	2024-06-04 01:51:35 GMT by grib_to_ncdf-2.24.0: /opt/ecmwf/eccodes/bin/grib_to_ncdf -S param -o /cache/tmp/b814f915-320b-4a3c-9b38-749a66dea248-adaptor.mars.internal-1717465886.9057872-7641-15-tmp.nc /cache/tmp/b814f915-320b-4a3c-9b38-749a66dea248-adaptor.mars.internal-1717465452.6220074-7641-14-tmp.grib		

▼ Discovery

```
ds_skt = ds.skt.isel(time=0)  
ds_skt.plot()
```

```
[1]: <matplotlib.collections.QuadMesh at 0x1ba80e299a0>
```

time = 2000-01-01



▼ Calculate Change

```
temp_change = (ds.skt.sel(time='2020-01-01')
               - ds.skt.sel(time='2000-01-01'))
```

```
temp_change
```

```
→ xarray.DataArray 'skt' (latitude: 188, longitude: 175)
```

```
array([[ 7.37029687,  6.43320676,  5.56598254, ...,  0.57223491,
       0.54340136,  0.50015105],
       [ 7.65974127,  6.7204332 ,  5.8521     , ...,  0.52898459,
       0.50902291,  0.47575344],
```

```
temp_change.plot()
```

```
→ <matplotlib.collections.QuadMesh at 0x1baed41a7c0>
```

