find\_ans.py

import numpy as np  
import pandas as pd  
  
  
data = pd.read\_csv('find\_ans.csv' , encoding='gb18030' , index\_col=0)  
indicator = data.columns.tolist()   
project = data.index.tolist()   
value = data.values  
print(indicator)  
print(project)  
print(value)  
data.head()  
  
  
  
def std\_data ( value , flag ):  
 for i in range(len(indicator)):  
 # print(flag[i])  
 if flag[i] == '+':  
 value[: , i] = (value[: , i] - np.min(value[: , i] , axis=0)) / (  
 np.max(value[: , i] , axis=0) - np.min(value[: , i] , axis=0)) + 0.001  
 elif flag[i] == '-':  
 value[: , i] = (np.max(value[: , i] , axis=0) - value[: , i]) / (  
 np.max(value[: , i] , axis=0) - np.min(value[: , i] , axis=0)) + 0.001  
 # print(value)  
 return value  
  
  
  
flag = ["-" , "+" , "-" , "+" , "+" , "+"]   
std\_value = std\_data(value , flag)  
print(std\_value)  
std\_value.round(3)  
DF = pd.DataFrame(std\_value)  
DF.to\_csv('value.csv')  
  
  
  
def cal\_weight ( indicator , project , value ):  
 p = np.array([[0.0 for i in range(len(indicator))] for i in range(len(project))])  
 # print(p)  
 for i in range(len(indicator)):  
 p[: , i] = value[: , i] / np.sum(value[: , i] , axis=0)  
  
 e = -1 / np.log(len(project)) \* sum(p \* np.log(p))   
 g = 1 - e   
 w = g / sum(g)   
 return w  
  
  
w = cal\_weight(indicator , project , std\_value)  
w = pd.DataFrame(w , index=data.columns , columns=['weight'])  
print("#######weight:#######")  
print(w)  
w.to\_csv('weight.csv')  
score = np.dot(std\_value , w).round(2)  
score = pd.DataFrame(score , index=data.index , columns=['score']).sort\_values(by=['score'] , ascending=False)  
score.to\_csv('score.csv')  
print(score)

Read\_data.py

from osgeo import gdal  
import pandas as pd  
import numpy as np  
  
gdal.AllRegister()  
filename\_list = ['World\_Atlas\_2015.tif', 'GEDI04\_B\_MW019MW138\_02\_002\_05\_R01000M\_MU.tif' ,  
 'gpw\_v4\_population\_density\_rev11\_2015\_2pt5\_min.tif' ]  
find\_list = {'Nanjing Xinjiekou area': [32.0 , 118.0] , 'Fengxian District': [30.0 , 121.0] ,  
 'Yellowstone Park': [44.3 , -110.3] , 'Fresno city': [37.0 , -119.0] , 'Chesterbrook': [41.1 , -77.1] ,  
 'Manhattan': [40.0 , -74.0] , 'jiuzhaigou': [33.1 , 104.1] , 'Xiaogang Village': [32.2 , 117.7]}  
  
# for key in find\_list.keys():  
# print('find ans of ' + key)  
for filename in filename\_list:  
 filePath = r'D:\ICM2023\python\_code\NewWorldAtlas/' + filename  
 dataset = gdal.Open(filePath)  
 adfGeoTransform = dataset.GetGeoTransform()  
 nXSize = dataset.RasterXSize # x  
 nYSize = dataset.RasterYSize # y  
 print(nXSize , nYSize)  
 im\_data = dataset.ReadAsArray(0 , 0 , nXSize , nYSize)  
 index = [] # loc  
 columns = [] # iloc  
 for j in range(nYSize):  
 lat = adfGeoTransform[3] + j \* adfGeoTransform[5]  
 index.append(lat)  
 for i in range(nXSize):  
 lon = adfGeoTransform[0] + i \* adfGeoTransform[1]  
 columns.append(lon)  
 data = pd.DataFrame(im\_data , index=index , columns=columns)  
 data.where(data >= 0 , data - data , inplace=True)  
 print(filename + ' read done')  
 for i in data.index:  
 if -90 < i < 90:  
 pass  
 else:  
 i = i / 100000  
  
 data.rename(index={i: int(i)} , inplace=True)  
  
 for j in data.columns:  
 if -180 < j < 180:  
 pass  
 else:  
 j = j / 100000  
  
 data.rename(columns={j: int(j)} , inplace=True)  
  
 print(data)  
  
 for key in find\_list.keys():  
 print('find ans of ' + key)  
 x = find\_list[key][0]  
 y = find\_list[key][1]  
 key\_data = data.loc[(x,y)]  
 print(key\_data)  
 key\_data.to\_csv(key+filename+'.csv')