Tam Doan

Spatial-DS

Program-2

import pygame

import random

from dbscan import \*

import sys,os

import pprint as pp

Resolution = 2000

def calculate\_mbrs(points, epsilon, min\_pts):

"""

Find clusters using DBscan and then create a list of bounding rectangles

to return.

"""

mbrs = []

clusters = dbscan(points, epsilon, min\_pts)

"""

Using list index value to iterate over the clusters dictionary

Does same as above

"""

for id in range(len(clusters)-1):

xs = []

ys = []

for p in clusters[id]:

xs.append(p[0])

ys.append(p[1])

max\_x = max(xs)

max\_y = max(ys)

min\_x = min(xs)

min\_y = min(ys)

mbrs.append([(min\_x,min\_y),(max\_x,min\_y),(max\_x,max\_y),(min\_x,max\_y),(min\_x,min\_y)])

return mbrs

def clean\_area(screen,origin,width,height,color):

"""

Prints a color rectangle (typically white) to "erase" an area on the screen.

Could be used to erase a small area, or the entire screen.

"""

ox,oy = origin

points = [(ox,oy),(ox+width,oy),(ox+width,oy+height),(ox,oy+height),(ox,oy)]

pygame.draw.polygon(screen, color, points, 0)

background\_colour = (255,255,255)

black = (0,0,0)

Manhatten = (194,35,38)

Queens = (243,115,56)

StatenIsland = (253,182,50)

Bronx = (2,120,120)

Brooklyn = (128,22,56)

(width, height) = (Resolution , Resolution)

screen = pygame.display.set\_mode((width, height))

pygame.display.set\_caption('Simple Line')

screen.fill(background\_colour)

pygame.display.flip()

epsilon = 20

min\_pts = 5.0

points = []

num\_points = 500

for i in range(num\_points):

x = random.randint(10,width-10)

y = random.randint(10,height-10)

points.append((x,y))

mbrs = calculate\_mbrs(points, epsilon, min\_pts)

running = True

DIRPATH = os.path.dirname(os.path.realpath(\_\_file\_\_))

keys = []

crimes = []

MaxX = 1067226

MaxY = 271820

MinX = 913357

MinY = 121250

got\_keys = False

with open(DIRPATH+'/'+'filtered\_crimes\_bronx.csv') as f:

for line in f:

line = ''.join(x if i % 2 == 0 else x.replace(',', ':') for i, x in enumerate(line.split('"')))

line = line.strip().split(',')

if not got\_keys:

keys = line

got\_keys = True

continue

crimes.append(line)

pointListBronx = []

for crime in crimes:

if (crime[19] != "") and (crime[20] != "" ):

x1 = int ( ( int(crime[19]) - MinX ) / ( MaxX - MinX ) \* Resolution )

y1 = Resolution - int ( ( int(crime[20]) - MinY ) / ( MaxY - MinY ) \* Resolution )

p = (x1,y1)

pointListBronx.append(p)

#######################################################3

got\_keys = False

crimes.clear()

with open(DIRPATH+'/'+'filtered\_crimes\_brooklyn.csv') as f:

for line in f:

line = ''.join(x if i % 2 == 0 else x.replace(',', ':') for i, x in enumerate(line.split('"')))

line = line.strip().split(',')

if not got\_keys:

keys = line

got\_keys = True

continue

crimes.append(line)

pointListBrooklyn = []

for crime in crimes:

if (crime[19] != "") and (crime[20] != "" ):

x1 = int ( ( int(crime[19]) - MinX ) / ( MaxX - MinX ) \* Resolution )

y1 = Resolution - int ( ( int(crime[20]) - MinY ) / ( MaxY - MinY ) \* Resolution )

p = (x1,y1)

pointListBrooklyn.append(p)

#########################################################

got\_keys = False

crimes.clear()

with open(DIRPATH+'/'+'filtered\_crimes\_manhattan.csv') as f:

for line in f:

line = ''.join(x if i % 2 == 0 else x.replace(',', ':') for i, x in enumerate(line.split('"')))

line = line.strip().split(',')

if not got\_keys:

keys = line

got\_keys = True

continue

crimes.append(line)

pointListManhattan = []

for crime in crimes:

if (crime[19] != "") and (crime[20] != "" ):

x1 = int ( ( int(crime[19]) - MinX ) / ( MaxX - MinX ) \* Resolution )

y1 = Resolution - int ( ( int(crime[20]) - MinY ) / ( MaxY - MinY ) \* Resolution )

p = (x1,y1)

pointListManhattan.append(p)

######################################################

got\_keys = False

crimes.clear()

with open(DIRPATH+'/'+'filtered\_crimes\_queens.csv') as f:

for line in f:

line = ''.join(x if i % 2 == 0 else x.replace(',', ':') for i, x in enumerate(line.split('"')))

line = line.strip().split(',')

if not got\_keys:

keys = line

got\_keys = True

continue

crimes.append(line)

pointListQueens = []

for crime in crimes:

if (crime[19] != "") and (crime[20] != "" ):

x1 = int ( ( int(crime[19]) - MinX ) / ( MaxX - MinX ) \* Resolution )

y1 = Resolution - int ( ( int(crime[20]) - MinY ) / ( MaxY - MinY ) \* Resolution )

p = (x1,y1)

pointListQueens.append(p)

######################################################

got\_keys = False

crimes.clear()

with open(DIRPATH+'/'+'filtered\_crimes\_staten\_island.csv') as f:

for line in f:

line = ''.join(x if i % 2 == 0 else x.replace(',', ':') for i, x in enumerate(line.split('"')))

line = line.strip().split(',')

if not got\_keys:

keys = line

got\_keys = True

continue

crimes.append(line)

pointListStatenIsland = []

for crime in crimes:

if (crime[19] != "") and (crime[20] != "" ):

x1 = int ( ( int(crime[19]) - MinX ) / ( MaxX - MinX ) \* Resolution )

y1 = Resolution - int ( ( int(crime[20]) - MinY ) / ( MaxY - MinY ) \* Resolution )

p = (x1,y1)

pointListStatenIsland.append(p)

######################################################

for point in pointListBronx:

pygame.draw.circle(screen, Bronx, point, 3, 0)

for point in pointListBrooklyn:

pygame.draw.circle(screen, Brooklyn, point, 3, 0)

for point in pointListManhattan:

pygame.draw.circle(screen, Manhatten, point, 3, 0)

for point in pointListQueens:

pygame.draw.circle(screen, Queens, point, 3, 0)

for point in pointListStatenIsland:

pygame.draw.circle(screen, StatenIsland, point, 3, 0)

for event in pygame.event.get():

if event.type == pygame.QUIT:

running = False

if event.type == pygame.MOUSEBUTTONDOWN:

clean\_area(screen,(0,0),width,height,(255,255,255))

points.append(event.pos)

mbrs = calculate\_mbrs(points, epsilon, min\_pts)

pygame.image.save(screen , "temp.png")

#pygame.display.flip()