import networkx as nx

import matplotlib.pyplot as plt

import random as r

def create\_graph():

g=nx.Graph()

g.add\_nodes\_from(range(1,101))

return g

def add\_bmi(G):

for ele in G.nodes():

G.node[ele]['name']=r.randint(15,40)

G.node[ele]['type']='person'

def get\_attr(G):

d=dict()

for var in G.nodes():

d[var]=G.node[var]['name']

return d

def get\_size(G):

arr=list()

for ele in G.nodes():

if G.node[ele]['type']=='person':

arr.append(G.node[ele]['name'] \*10)

else:

arr.append(1000)

return arr

def foci\_nodes(G):

index=G.number\_of\_nodes()

index=index+1

data=['Yoga Club','Eatery','Gymnastics','Seminar','Directorate']

for ele in range(0,5):

G.add\_node(index)

G.node[index]['type']='foci'

G.node[index]['name']=data[ele]

index=index+1

def get\_nodes(G):

arr1=list()

arr2=list()

for ele in G.nodes():

if(G.node[ele]['type']=='person'):

arr1.append(ele)

else :

arr2.append(ele)

return arr1,arr2

def foci\_grp(G):

p,f=get\_nodes(G)

for ele in p:

ran=r.choice(f)

G.add\_edge(ran,ele)

def get\_color(G):

c=list()

for ele in G.nodes():

if G.node[ele]['type']=='person':

if G.node[ele]['name']==15:

c.append('green')

elif G.node[ele]['name']==40:

c.append('yellow')

else :

c.append('blue')

else:

c.append('red')

return c

def visualize(G):

d=get\_attr(G)

size=get\_size(G)

color=get\_color(G)

nx.draw(G,labels=d,node\_size=size,node\_color=color)

plt.show()

def imm\_neigh(u,v,G):

n1=G.neighbors(u)

n2=G.neighbors(v)

n1=set(n1)

n2=set(n2)

return len(n1&n2)

def closure(G):

arr=list()

for u in G.nodes():

for v in G.nodes():

if (u!=v):

if(G.node[v]['type']=='person' or G.node[u]['type']=='person'):

k=imm\_neigh(u,v,G)

prob=0.05

p=1-(1-prob)\*\*k

arr.append([u,v,p])

for ele in arr:

u=ele[0]

v=ele[1]

p=ele[2]

no=r.uniform(0,1)

if(no<p):

G.add\_edge(u,v)

def homophily(G):

p,f=get\_nodes(G)

for u in p:

for v in p:

if (u!=v):

diff=abs(G.node[u]['name']-G.node[v]['name'])

prob=float(1)/(diff+1000)

if(r.uniform(0,1)<prob):

G.add\_edge(u,v)

def influence(G):

p,f=get\_nodes(G)

for each in f:

if(G.node[each]['type']=='Eatery'):

for ele in G.neighbors(each):

if(G.node[ele]['name']!=40):

get=G.node[ele]['name']+1

G.node[ele]['name']=get

if(G.node[each]['type']=='Gymnastics'):

for ele in G.neighbors(each):

if(G.node[ele]['name']!=15):

get=G.node[ele]['name']-1

G.node[ele]['name']=get

G=create\_graph()

add\_bmi(G)

foci\_nodes(G)

foci\_grp(G)

visualize(G)

for var in range(5):

homophily(G)

closure(G)

influence(G)

visualize(G)

nx.write\_gml(G,"testing.gml")