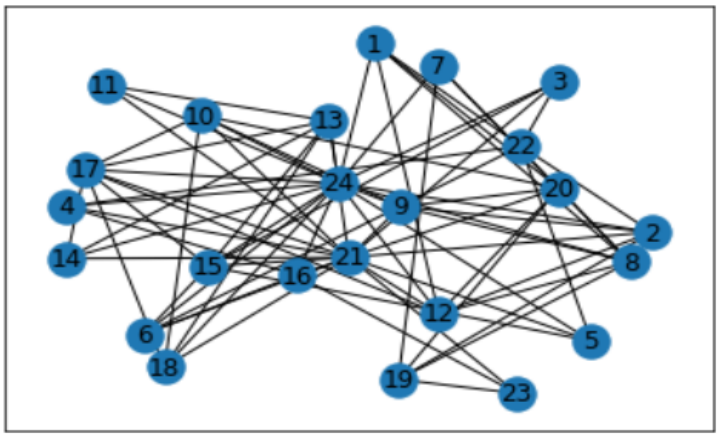
گزارش پروژه 2

زهراحیدری و فاطمه رضوی

در ابتدا همانند پروژه قبلی یک فایل تکست از داده های مجموعه اول میسازیم و با استفاده از دستوراتی که قبلا هم از آن استفاده کردیم نود ها و وزن هرکدام رو به برنامه معرفی میکنیم با این تفاوت که حتما باید گراف بدون جهت باشد و سپس شکل آن را رسم میکنیم.

```
:
import networkx as nx
import pandas as pd
import random
import numpy as np
```

```
:
df=pd.read_csv('arcs.txt', sep=' ', names=['n1','n2','Weight'])
G=nx.from_pandas_edgelist(df,'n1','n2',create_using=nx.Graph)
k_pos=nx.spring_layout(G, k=2.0)
nx.draw_networkx(G,k_pos)
```



و سپس در ادامه یک تابع همانند جزوه مینوسیم(مدل آبشاری) که بیاید تشخیص دهد که نود هایی ک فعال شده توانایی فعال کردن چ نود هایی در کنار همیسایه خود را دارد با این تفاوت که یک ورودی دیگر به اسم seeds اضافه میکنیم که بعدا یک لیستی برای گرفتن آنهایی ک فعال میشوند باشد و به لیست اضافه کند.

```python
def IC(G_nx,seeds,Probability):
    for i in G_nx.nodes():
        if i in seeds:
            G_nx.nodes[i]['infected']=True
        else:
            G_nx.nodes[i]['infected']=False
        G_nx.nodes[i]['try']=False
    c=0
    new_active=True
    while new_active:
        new_active=False
        for v in G_nx.nodes():
            if G_nx.nodes[v]['infected']==True and G_nx.nodes[v]['try']==False:
                for w in nx.neighbors(G_nx,v):
                    if G_nx.nodes[w]['infected']==False:
                        x=random.random()
                        if x<Probability:
                            G_nx.nodes[w]['infected']=True
                            new_active=True
                            c+=1
                G_nx.nodes[v]['try']=True
    return c
```

در ادامه دوتابع به اسم مونت کارلو وgreed hill climbing داریم که در مونت کارلو میایم تکرار هایی که برای تابع لازم است رو انجام میدهیم (repeat) سپس در تابع بعدی الگوریتم تپه نوردی رو با استفاده از شبه کدی که در جزوه ها قرار داشت پیاده سازی میکنیم و میایم در هر بار max spread , max node را به عنوان خروجی در نظر میگیریم که ببنیم در هرمرحله چ تغییراتی افتاده و سپس برحسب مقدار k ، k نود را به عنوان خروجی پاس میدهیم.

```python
def monte_carlo(G_nx,seeds,Probability,repeats):
    sum_IC = 0
    for i in range(repeats):
        sum_IC += IC(G_nx,seeds,Probability)
    return sum_IC / repeats

def greedy_hill_climbing(G_nx,k,probability=0.1,repeats=10):
    S = set()
    N = set(G_nx.nodes())
    for i in range(k):
        max_spread = -1
        max_node = None
        for n in N-S:
            spread = monte_carlo(G_nx,S.union({n}),probability,repeats)-monte_carlo(G_nx,S,probability,repeats)
            if spread > max_spread:
                max_spread = spread
                max_node = n
                print("Max Spread:", max_spread ,"\t\t\t Max Node:",max_node)
        S.add(max_node)
    return S
```

و از تابع greed hill climbing خروجی میگیریم تا مقادیر رو برای ما انتخاب کند.

```
: greedy_hill_climbing(G, 3 ,0.2 ,50)
```

```
Max Spread: 4.58                        Max Node: 1
Max Spread: 6.04                        Max Node: 2
Max Spread: 6.78                        Max Node: 4
Max Spread: 6.8                         Max Node: 6
Max Spread: 7.4                         Max Node: 9
Max Spread: 7.58                        Max Node: 10
Max Spread: 8.2                         Max Node: 12
Max Spread: 9.48                        Max Node: 21
Max Spread: 10.02                       Max Node: 24
Max Spread: 2.0                         Max Node: 1
Max Spread: 2.040000000000001                   Max Node: 6
Max Spread: 2.6400000000000006                  Max Node: 13
Max Spread: 2.74                        Max Node: 21
Max Spread: 0.5400000000000009                  Max Node: 1
Max Spread: 0.7200000000000006                  Max Node: 5
Max Spread: 1.0999999999999996                  Max Node: 8
Max Spread: 1.2599999999999998                  Max Node: 10
Max Spread: 1.8599999999999994                  Max Node: 12

: {12, 21, 24}
```

در 50 بار تکرار و k=3 و احتمال 0.2 به ما همچین خروجی میدهد که سه نود 12 و 21 و 24 بهینه ترین نود ها در این نتیجه هستن.

کد پروژه در آدرس گیت هاب قابل مشاهده میباشد.

https://github.com/Zahraheidari1/Social-Network