Візуалізація даних

**Лабораторна робота №8**

# Візуалізація зв’язків у соціальній мережі Instagram

Аналіз зв’язків у соціальних мережах є корисним і ефективним інструментом для виявлення основної специфіки людських відносин у соціальних групах. Ці інструменти дозволяють визначати кількісні та якісні параметри утворених зв’язків.

Одною з найпопулярніших соціальних мереж є Instagram. Для взаємодії з соціально мережею використовується спеціальний API. Саме за його допомогою можливо отримати всю необхідну інформацію.

Для python розроблено неофіційний Instagram API, щоб дати доступ до всіх функцій Instagram (наприклад, відслідковувати підписчиків, завантажувати фото та відео тощо). Сторінка на github: <https://github.com/LevPasha/Instagram-API-python> . Детальна інструкція з встановлення також знаходиться за вказаною адресою.

Для більшості випадків встановити Instagram API можна за допомогою наступної команди:

pip install InstagramApi

Приклад використання Instagram API наведено у лістингу 1.

Лістинг 1 – Приклад використання Instagram API

#!/usr/bin/env python

# -\*- coding: utf-8 -\*-

**from** InstagramAPI **import** InstagramAPI

**from** time **import** sleep

**import** networkx **as** nx

**import** matplotlib**.**pyplot **as** plt

api **=** InstagramAPI**(**"your\_login"**,** "your\_password"**)**

myFollowings **=** **[]**

myFollowingsNames **=** **[]**

G **=** nx**.**Graph**()**

**if** api**.**login**():**

**print(**"You login as:"**,** api**.**username**)**

api**.**getSelfUserFeed**()**

api**.**getUserFollowings**(**api**.**username\_id**)**

G**.**add\_node**(**api**.**username**,** label**=**api**.**username**)**

**for** following **in** api**.**LastJson**[**'users'**]:**

myFollowingsNames**.**append**(**following**[**'username'**])**

myFollowings**.**append**({**"username"**:** following**[**"username"**],** "id"**:** following**[**"pk"**]})**

G**.**add\_node**(**following**[**'username'**],** label**=**following**[**'full\_name'**])**

G**.**add\_edge**(**api**.**username**,** following**[**'username'**])**

**else:**

**print(**"cannot to connect"**)**

dictionary **=** **{}**

**print** myFollowings

i **=** 0

**for** person **in** myFollowings**:**

**print(**person**)**

listFollowers **=** **[]**

api**.**getUserFollowings**(**person**[**'id'**])**

**for** following **in** api**.**LastJson**[**'users'**]:**

**if** following**[**'username'**]** **in** myFollowingsNames**:**

G**.**add\_node**(**following**[**'username'**],** label**=**following**[**'full\_name'**])**

G**.**add\_edge**(**person**[**'username'**],** following**[**'username'**])**

i **+=** 1

sleep**(**1**)**

**print(**"Saving..."**)**

nx**.**write\_gexf**(**G**,** "InstaFriends.gexf"**)**

**print(**"Drawing..."**)**

nx**.**draw\_spring**(**G**,** with\_labels**=True,** font\_weight**=**'bold'**,** font\_size**=**5**)**

plt**.**savefig**(**'InstaGraf.png'**,** dpi**=**600**)**

plt**.**show**()**

Результати роботи програми представлено на рисунках 1 та 2.

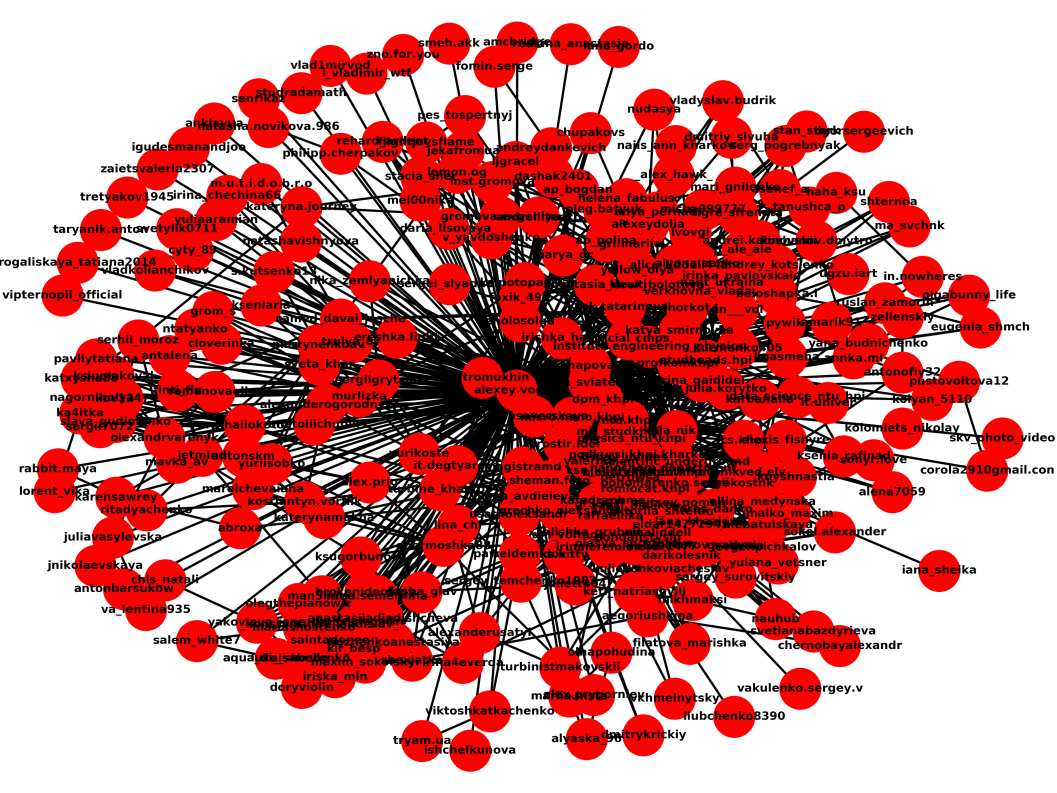


Рисунок 1 – Результат візуалізації графу за допомогою networkX

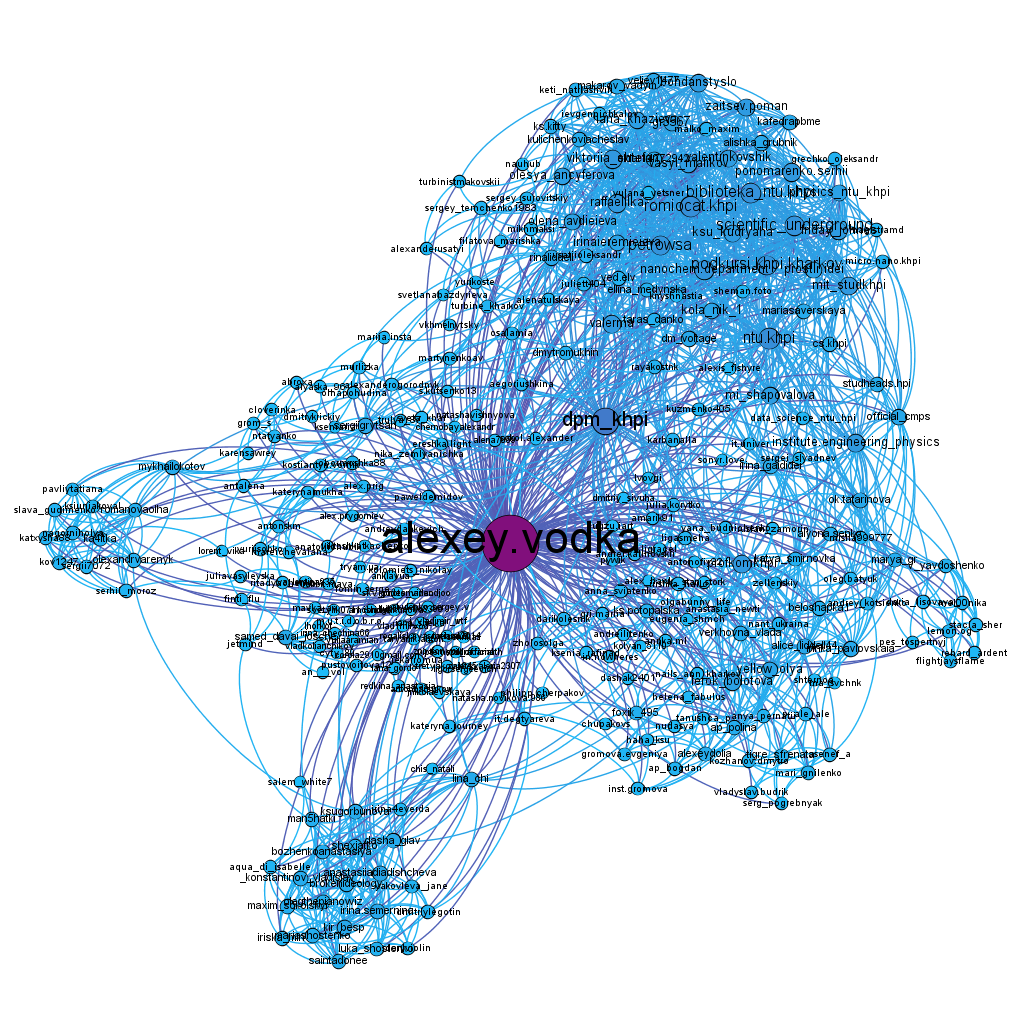


Рисунок 2 – Результат візуалізації графу за допомогою gephi

# Варіанти завдань

Побудувати граф ваших підписчиків у соціальній мережі Instagram. Провести візуалізацію графа за допомогою бібліотеки networkx та gephi. Також визначити за допомогою networkx або gephi характеристики згідно з варіантом:

**Варіант 1**

Кількість вузлів графу;

**Варіант 2**

Кількість дуг графу;

**Варіант 3**

Середню ступінь вершини;

**Варіант 4**

Діаметр графу;

**Варіант 5**

Густину графу;

**Варіант 6**

Середній коефіцієнт кластеризації;

**Варіант 7**

Середню довжину шляху;

**Варіант 8**

Модулярність графу;

**Варіант 9**

Відношення кількості дуг графу до кількості вузлів;